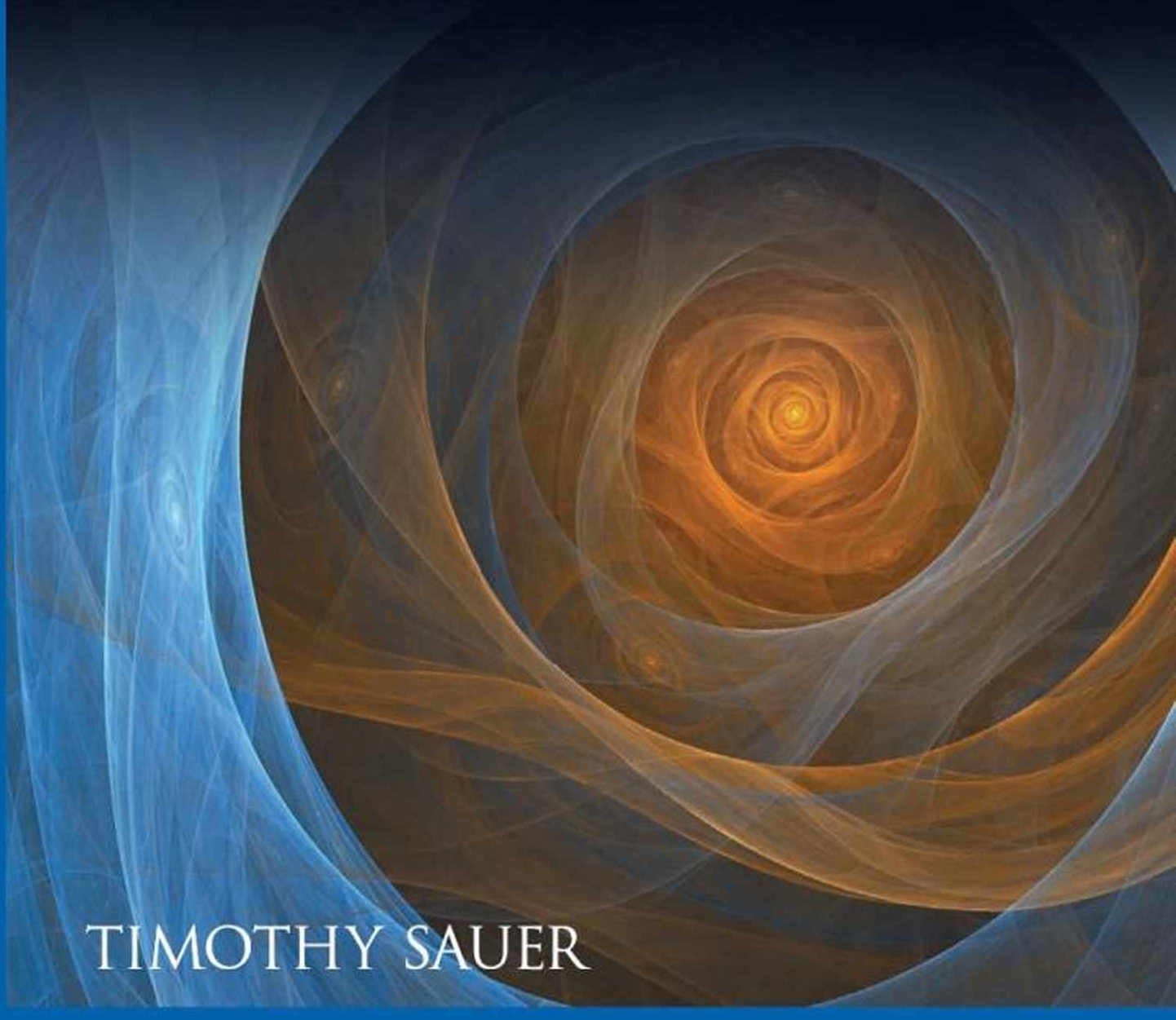
ANÁLISIS NUMÉRICO

SEGUNDA EDICIÓN



TIMOTHY SAUER

ALWAYS LEARNING

PEARSON

Análisis numérico

Análisis numérico

SEGUNDA EDICIÓN

Timothy Sauer
George Mason University

TRADUCCIÓN

Jesús Elmer Murrieta Murrieta
Maestro en Investigación de operaciones
ITESM, Campus Morelos

REVISIÓN TÉCNICA

Salvador García Burgos
Departamento de Ciencias Básicas
Facultad de Ingeniería
Universidad Nacional Autónoma de México

José Job Flores Godoy
Departamento de Física y Matemáticas
Universidad Iberoamericana

PEARSON

Datos de catalogación bibliográfica

SAUER, TIMOTHY
Análisis numérico. Segunda edición.

PEARSON EDUCACIÓN, México, 2013

ISBN 978-607-32-2059-0
Área: Matemáticas

Formato: 20 × 25.5 cm Páginas: 664

Authorized translation from the English language edition, entitled *NUMERICAL ANALYSIS*, 2nd. Edition, by *TIMOTHY SAUER*, published by Pearson Education, Inc., publishing as Pearson, Copyright © 2012. All rights reserved.
ISBN 9780321783677

Traducción autorizada de la edición en idioma inglés, titulada *NUMERICAL ANALYSIS*, 2a. edición, por *TIMOTHY SAUER*, publicada por Pearson Education, Inc., publicada como Pearson, Copyright © 2012. Todos los derechos reservados.

Esta edición en español es la única autorizada.

Edición en español

Dirección General:	Philip De la Vega
Dirección Educación Superior:	Mario Contreras
Editora Sponsor:	Gabriela López Ballesteros e-mail: gabriela.lopezballesteros@pearson.com
Editor de Desarrollo:	Bernardino Gutiérrez Hernández
Supervisor de Producción:	José D. Hernández Garduño
Gerencia Editorial Educación Superior Latinoamérica:	Marisa de Anta

SEGUNDA EDICIÓN, 2013

D.R. © 2013 por Pearson Educación de México, S.A. de C.V.
Atlacomulco 500-5o. piso
Col. Industrial Atoto
53519, Naucalpan de Juárez, Estado de México

Cámara Nacional de la Industria Editorial Mexicana. Reg. núm. 1031.

Reservados todos los derechos. Ni la totalidad ni parte de esta publicación pueden reproducirse, registrarse o transmitirse, por un sistema de recuperación de información, en ninguna forma ni por ningún medio, sea electrónico, mecánico, fotoquímico, magnético o electroóptico, por fotocopia, grabación o cualquier otro, sin permiso previo por escrito del editor.

El préstamo, alquiler o cualquier otra forma de cesión de uso de este ejemplar requerirá también la autorización del editor o de sus representantes.

ISBN VERSIÓN IMPRESA: 978-607-32-2059-0
ISBN VERSIÓN E-BOOK: 978-607-32-2060-6
ISBN E-CHAPTER: 978-607-32-2061-3

Impreso en México. *Printed in Mexico.*
1 2 3 4 5 6 7 8 9 0 - 16 15 14 13

PEARSON

Contenido

PREFACIO xiii

CAPÍTULO 0 Fundamentos 1

0.1 Evaluación de un polinomio 1
0.2 Números binarios 5
0.2.1 Decimal a binario 6
0.2.2 De binario a decimal 7
0.3 Representación del punto flotante de los números reales 8
0.3.1 Formatos de punto flotante 8
0.3.2 Representación en máquina 11
0.3.3 Suma de números de punto flotante 13
0.4 Pérdida de significancia 16
0.5 Repaso de cálculo 19
Software y lecturas adicionales 23

CAPÍTULO 1 Resolución de ecuaciones 24

1.1 El método de bisección 25
1.1.1 Confinamiento de una raíz 25
1.1.2 ¿Qué tan exacto y a qué velocidad? 28
1.2 Iteración de punto fijo 30
1.2.1 Puntos fijos de una función 31
1.2.2 Geometría de la iteración de punto fijo 33
1.2.3 Convergencia lineal de la iteración de punto fijo 34
1.2.4 Criterios de detención 40
1.3 Límites de exactitud 43
1.3.1 Error hacia adelante y hacia atrás 44
1.3.2 El polinomio de Wilkinson 47
1.3.3 Sensibilidad de la localización de raíces 48
1.4 Método de Newton 51
1.4.1 Convergencia cuadrática del método de Newton 53
1.4.2 Convergencia lineal del método de Newton 55
1.5 Localización de raíces sin derivadas 61
1.5.1 Método de la secante y sus variantes 61
1.5.2 Método de Brent 64

Comprobación en la realidad 1: Cinemática de la plataforma Stewart 67
Software y lecturas adicionales 69

CAPÍTULO 2 Sistemas de ecuaciones 71

2.1 Eliminación gaussiana 71
2.1.1 Eliminación gaussiana simple 72
2.1.2 Conteo de operaciones 74

2.2 La factorización LU 79
2.2.1 Forma matricial de la eliminación gaussiana 79
2.2.2 Sustitución hacia atrás con la factorización LU 81
2.2.3 Complejidad de la factorización LU 83
2.3 Fuentes de error 85
2.3.1 Error de magnificación y número de condición 86
2.3.2 Dominancia 91
2.4 La factorización PA = LU 95
2.4.1 Pivoteo parcial 95
2.4.2 Matrices de permutación 97
2.4.3 Factorización PA = LU 98

Comprobación en la realidad 2: La viga de Euler-Bernoulli 102

2.5 Métodos iterativos 106
2.5.1 Método de Jacobi 106
2.5.2 Método de Gauss-Seidel y SRS 108
2.5.3 Convergencia de los métodos iterativos 111
2.5.4 Cálculos de matrices dispersas 113
2.6 Métodos para matrices simétricas definidas positivas 117
2.6.1 Matrices simétricas definidas positivas 117
2.6.2 Factorización de Cholesky 119
2.6.3 Método del gradiente conjugado 121
2.6.4 Precondicionamiento 126
2.7 Sistemas de ecuaciones no lineales 130
2.7.1 Método de Newton multivariado 131
2.7.2 Método de Broyden 133
Software y lecturas adicionales 137

CAPÍTULO 3 Interpolación 138

3.1 Datos y funciones de interpolación 139
3.1.1 Interpolación de Lagrange 140
3.1.2 Diferencias divididas de Newton 141
3.1.3 ¿Cuántos polinomios de grado *d* pasan por *n* puntos? 144
3.1.4 Código para la interpolación 145
3.1.5 Representación de funciones mediante polinomios de aproximación 147
3.2 Error de interpolación 151
3.2.1 Fórmula del error en la interpolación 151
3.2.2 Demostración de la forma de Newton y la fórmula del error 153
3.2.3 Fenómeno de Runge 155
3.3 Interpolación de Chebyshev 158
3.3.1 Teorema de Chebyshev 158
3.3.2 Polinomios de Chebyshev 160
3.3.3 Cambio de intervalo 162
3.4 Splines cúbicas 166
3.4.1 Propiedades de las splines 167
3.4.2 Condiciones de extremo 173
3.5 Curvas de Bézier 179

Comprobación en la realidad 3: Fuentes a partir de las curvas de Bézier 183

Software y lecturas adicionales 187

CAPÍTULO 4 Mínimos cuadrados 188

4.1 Mínimos cuadrados y ecuaciones normales 188
- 4.1.1 Sistemas de ecuaciones inconsistentes 189
- 4.1.2 Modelos de ajuste a los datos 193
- 4.1.3 Condicionamiento de mínimos cuadrados 197

4.2 Exploración de modelos 201
- 4.2.1 Datos periódicos 201
- 4.2.2 Linealización de datos 203

4.3 Factorización QR 212
- 4.3.1 Ortogonalización de Gram-Schmidt y mínimos cuadrados 212
- 4.3.2 Ortogonalización de Gram-Schmidt modificado 218
- 4.3.3 Reflectores de Householder 220

4.4 Método del residuo mínimo generalizado (GMRES) 225
- 4.4.1 Métodos de Krylov 226
- 4.4.2 GMRES precondicionado 228

4.5 Mínimos cuadrados no lineales 230
- 4.5.1 Método de Gauss-Newton 230
- 4.5.2 Modelos con parámetros no lineales 233
- 4.5.3 Método de Levenberg-Marquardt 235

Comprobación en la realidad 4: GPS, condicionamiento y mínimos cuadrados no lineales 238

Software y lecturas adicionales 242

CAPÍTULO 5 Diferenciación e integración numérica 243

5.1 Diferenciación numérica 244
- 5.1.1 Fórmulas de las diferencias finitas 244
- 5.1.2 Error de redondeo 247
- 5.1.3 Extrapolación 249
- 5.1.4 Diferenciación e integración simbólica 250

5.2 Fórmulas de Newton-Cotes para la integración numérica 254
- 5.2.1 Regla del trapecio 255
- 5.2.2 Regla de Simpson 257
- 5.2.3 Fórmulas de Newton-Cotes compuestas 259
- 5.2.4 Métodos de Newton-Cotes abiertos 262

5.3 Integración de Romberg 265

5.4 Cuadratura adaptativa 269

5.5 Cuadratura gaussiana 273

Comprobación en la realidad 5: Control de movimiento en el modelado asistido por computadora 278

Software y lecturas adicionales 280

CAPÍTULO 6 Ecuaciones diferenciales ordinarias 281

6.1 Problemas de valor inicial 282
- 6.1.1 Método de Euler 283
- 6.1.2 Existencia, unicidad y continuidad de las soluciones 287
- 6.1.3 Ecuaciones lineales de primer orden 290

6.2 Análisis del error en la solución de PVI 293
- 6.2.1 Error de truncamiento local y total 293

6.2.2 Método explícito del trapecio 297
6.2.3 Métodos de Taylor 300
6.3 Sistemas de ecuaciones diferenciales ordinarias 303
6.3.1 Ecuaciones de orden superior 304
6.3.2 Simulación en computadora: el péndulo 305
6.3.3 Simulación en computadora: la mecánica orbital 309
6.4 Métodos y aplicaciones de Runge-Kutta 314
6.4.1 La familia Runge-Kutta 314
6.4.2 Simulación en computadora: la neurona de Hodgkin-Huxley 317
6.4.3 Simulación en computadora: las ecuaciones de Lorenz 319

Comprobación en la realidad 6: El puente Tacoma Narrows 322

6.5 Métodos con tamaño de paso variable 325
6.5.1 Pares integrados de Runge-Kutta 325
6.5.2 Métodos de cuarto y quinto orden 328
6.6 Métodos implícitos y ecuaciones rígidas 332
6.7 Métodos de varios pasos 336
6.7.1 Generación de métodos de varios pasos 336
6.7.2 Métodos de varios pasos explícitos 339
6.7.3 Métodos de varios pasos implícitos 342
Software y lecturas adicionales 347

CAPÍTULO 7 Problemas de valor de frontera 348

7.1 Método de disparo 349
7.1.1 Soluciones a problemas de valor de frontera 349
7.1.2 Implementación del método de disparo 352

Comprobación en la realidad 7: Deformación de un anillo circular 355

7.2 Métodos de diferencias finitas 357
7.2.1 Problemas de valor de frontera lineales 357
7.2.2 Problemas de valor de frontera no lineales 359
7.3 Colocación y el método del elemento finito 365
7.3.1 Colocación 365
7.3.2 Elementos finitos y el método de Galerkin 367
Software y lecturas adicionales 373

CAPÍTULO 8 Ecuaciones diferenciales parciales 374

8.1 Ecuaciones parabólicas 375
8.1.1 Método de las diferencias hacia adelante 375
8.1.2 Análisis de estabilidad del método de las diferencias hacia adelante 379
8.1.3 Método de la diferencia hacia atrás 380
8.1.4 Método de Crank-Nicolson 385
8.2 Ecuaciones hiperbólicas 393
8.2.1 La ecuación de onda 393
8.2.2 La condición CFL 395
8.3 Ecuaciones elípticas 398
8.3.1 Método de las diferencias finitas para ecuaciones elípticas 399

Comprobación en la realidad 8: Distribución del calor en una aleta de enfriamiento 403
8.3.2 Método del elemento finito para ecuaciones elípticas 406

8.4 Ecuaciones diferenciales parciales no lineales 417
8.4.1 Solucionador implícito de Newton 417
8.4.2 Ecuaciones no lineales en dos dimensiones espaciales 423
Software y lecturas adicionales 430

CAPÍTULO 9 Números aleatorios y sus aplicaciones 431

9.1 Números aleatorios 432
9.1.1 Números pseudoaleatorios 432
9.1.2 Números aleatorios exponenciales y normales 437
9.2 Simulación de Monte Carlo 440
9.2.1 Leyes de potencia para la estimación de Monte Carlo 440
9.2.2 Números cuasialeatorios 442
9.3 Movimiento browniano discreto y continuo 446
9.3.1 Caminatas aleatorias 447
9.3.2 Movimiento browniano continuo 449
9.4 Ecuaciones diferenciales estocásticas 452
9.4.1 Incorporación de la incertidumbre a las ecuaciones diferenciales 452
9.4.2 Métodos numéricos para EDE 456

Comprobación en la realidad 9: La fórmula de Black-Scholes 464
Software y lecturas adicionales 465

CAPÍTULO 10 Interpolación trigonométrica y la TRF 467

10.1 La transformada de Fourier 468
10.1.1 Aritmética compleja 468
10.1.2 Transformada discreta de Fourier 470
10.1.3 La transformada rápida de Fourier 473
10.2 Interpolación trigonométrica 476
10.2.1 Teorema de interpolación de la TDF 476
10.2.2 Evaluación eficiente de funciones trigonométricas 479
10.3 FFT y el procesamiento de señales 483
10.3.1 Ortogonalidad e interpolación 483
10.3.2 Ajuste por mínimos cuadrados con funciones trigonométricas 485
10.3.3 Sonido, ruido y filtrado 489

Comprobación en la realidad 10: El filtro de Wiener 492
Software y lecturas adicionales 494

CAPÍTULO 11 Compresión 495

11.1 La transformada discreta del coseno 496
11.1.1 TDC unidimensional 496
11.1.2 La TDC y la aproximación por mínimos cuadrados 498
11.2 TDC bidimensional y compresión de imágenes 501
11.2.1 TDC bidimensional 501
11.2.2 Compresión de imágenes 505
11.2.3 Cuantificación 508
11.3 Codificación de Huffman 514
11.3.1 Teoría de la información y codificación 514
11.3.2 Codificación de Huffman para el formato JPEG 517

11.4 TDC modificada y compresión de audio 519
11.4.1 Transformada discreta del coseno modificada 520
11.4.2 Cuantificación de bits 525
Comprobación en la realidad 11: Un codec de audio simple 527
Software y lecturas adicionales 530

CAPÍTULO 12 Valores y vectores característicos y valores singulares 531

12.1 Métodos de iteración de potencia 531
12.1.1 Iteración de potencia 532
12.1.2 Convergencia de la iteración de potencia 534
12.1.3 Iteración de potencia inversa 535
12.1.4 Iteración del cociente de Rayleigh 537
12.2 Algoritmo QR 539
12.2.1 Iteración simultánea 539
12.2.2 Forma real de Schur y el algoritmo QR 542
12.2.3 Forma superior de Hessenberg 544
Comprobación en la realidad 12: Cómo clasifican los motores de búsqueda la calidad de la página 549
12.3 Descomposición de valor singular 552
12.3.1 Localización de la DVS en general 554
12.3.2 Caso especial: matrices simétricas 555
12.4 Aplicaciones de la DVS 557
12.4.1 Propiedades de la DVS 557
12.4.2 Reducción de dimensión 559
12.4.3 Compresión 560
12.4.4 Cálculo de la DVS 561
Software y lecturas adicionales 563

CAPÍTULO 13 Optimización 565

13.1 Optimización no restringida sin derivadas 566
13.1.1 Búsqueda de la sección dorada 566
13.1.2 Interpolación parabólica sucesiva 569
13.1.3 Búsqueda de Nelder-Mead 571
13.2 Optimización no restringida con derivadas 575
13.2.1 Método de Newton 576
13.2.2 Gradiente descendiente 577
13.2.3 Búsqueda del gradiente conjugado 578
Comprobación en la realidad 13: Conformación molecular y optimización numérica 580
Software y lecturas adicionales 582

Apéndice A 583

A.1 Fundamentos de las matrices 583
A.2 Multiplicación en bloque 585
A.3 Valores y vectores propios 586
A.4 Matrices simétricas 587
A.5 Cálculo vectorial 588

Apéndice B **590**

B.1 Inicio de MATLAB 590
B.2 Gráficas 591
B.3 Programación en MATLAB 593
B.4 Control de flujo 594
B.5 Funciones 595
B.6 Operaciones con matrices 597
B.7 Animación y películas 597

Respuestas a los ejercicios seleccionados **599**

Bibliografía **626**

Índice **637**

Prefacio

Análisis numérico es un libro para estudiantes de ingeniería, ciencias, matemáticas e informática que hayan cursado cálculo elemental y álgebra matricial. El objetivo principal de este texto es construir y explorar algoritmos para resolver problemas científicos y de ingeniería. Otra de sus misiones es ayudar al lector a localizar estos algoritmos en un escenario de principios poderosos y de gran alcance. En conjunto, estos principios unificadores constituyen un campo dinámico de la investigación y el desarrollo actuales en la ciencia moderna numérica y computacional.

La disciplina del análisis numérico está repleta de ideas útiles. En los libros de texto se corre el riesgo de presentar los temas como un paquete de trucos brillantes, pero sin relación entre sí, algo que hemos evitado aquí. Para obtener una comprensión profunda, los lectores deben aprender más allá de cómo se codifica el método de Newton, el método de Runge-Kutta y la transformada rápida de Fourier. Deben absorber los grandes principios, aquellos que permean el análisis numérico y se integran a sus intereses de competencia en precisión y eficiencia.

Entre las ideas generales más importantes se encuentran las nociones de convergencia, la complejidad, el condicionamiento, la compresión y la ortogonalidad. Cualquier buen método de aproximación debe converger hacia la respuesta correcta a medida que se le dedican más recursos de cómputo, y la complejidad de un método es una medida del uso de estos recursos. El condicionamiento de un problema, o la susceptibilidad al aumento del error, es fundamental para saber cómo puede ser atacado. Muchas de las aplicaciones más recientes de análisis numérico se esfuerzan en generar datos de una manera más corta o comprimida. Por último, la ortogonalidad es crucial para la eficacia en muchos algoritmos y es insustituible cuando el condicionamiento constituye un problema o la compresión es un objetivo.

En este libro los papeles de los cinco conceptos en el análisis numérico moderno se destacan en elementos temáticos cortos llamados "anotaciones". En estas anotaciones se comenta el tema en estudio y se hacen conexiones informales hacia expresiones del mismo concepto en otras partes del libro. Esperamos que el hecho de destacar los cinco conceptos de esta manera tan explícita funcione acentuando lo que es realmente crucial sobre la teoría en la página.

Aunque es bien sabido que las ideas del análisis numérico son vitales para la práctica de la ciencia y la ingeniería modernas, nunca está de más ser evidente. Las comprobaciones en la realidad ofrecen ejemplos concretos de la forma en que los métodos numéricos conducen a la solución de problemas científicos y tecnológicos importantes. Estas aplicaciones extendidas se escogieron de modo que fueran oportunas y cercanas a la experiencia cotidiana. Aunque es imposible (y probablemente indeseable) presentar todos los detalles de los problemas, las comprobaciones en la realidad intentan profundizar lo suficiente para mostrar cómo puede aprovecharse la aplicación de un poco de matemáticas, a través de una técnica o algoritmo, y obtener una gran recompensa en el diseño y el funcionamiento tecnológicos. En la primera edición, la sección Comprobación en la realidad ha demostrado ser muy popular como fuente de proyectos para los estudiantes, y en esta segunda edición se han ampliado.

NUEVO EN ESTA EDICIÓN. En esta segunda edición se presenta una importante expansión de los métodos para resolver sistemas de ecuaciones. En el capítulo 2 se agregó la factorización de Cholesky para solucionar ecuaciones matriciales simétricas definidas y positivas. En el capítulo 4 se ha añadido un análisis del enfoque de Krylov para los grandes sistemas lineales, incluyendo el método GMRES, junto con material nuevo sobre el uso de precondicionadores para problemas simétricos y no simétricos. También se presentan, como novedades, la ortogonalización modificada de Gram-Schmidt y el método Levenberg-Marquardt. En el capítulo 8 el estudio de las EDP se ha extendido a las EDP no lineales, incluyendo las ecuaciones de reacción-difusión y de formación de patrones. El material presentado se ha modificado con el fin de hacer más fácil su lectura, con base en la retroalimentación de los estudiantes; además, se han añadido ejercicios y problemas de computadora nuevos a lo largo de todo el libro.

TECNOLOGÍA. El paquete de software MATLAB se utiliza para la exposición de los algoritmos y como plataforma sugerida en la realización de tareas y proyectos de los estudiantes. La cantidad de código de MATLAB que se proporciona en el texto está modulada cuidadosamente,

porque sabemos que el exceso tiende a ser contraproducente. En los primeros capítulos se incluye una mayor cantidad de código de MATLAB, lo que permite al lector obtener un dominio sobre este programa de una manera gradual. Cuando se proporciona un código más elaborado (por ejemplo, en el estudio de la interpolación y las ecuaciones diferenciales ordinarias y parciales), se espera que el lector lo utilice como un punto de partida para explotarlo y aumentarlo o mejorarlo.

El uso de una plataforma computacional específica con este libro no es indispensable, pero la presencia creciente de MATLAB en los departamentos de ingeniería y ciencias demuestra que un lenguaje común puede ayudar a salvar muchos obstáculos. Con MATLAB, todos los problemas de interfaz (datos de entrada/salida, graficado, etcétera) se resuelven de un solo golpe. Las dificultades de la estructura de datos (por ejemplo, las que surgen cuando se estudian los métodos de matriz dispersa) están normalizadas con base en comandos apropiados. MATLAB dispone de capacidades para la entrada y salida de archivos de audio e imagen. La simulación de ecuaciones diferenciales puede realizarse con facilidad debido a los comandos de animación incorporados en MATLAB. Estos objetivos pueden alcanzarse de otras maneras; sin embargo, siempre resulta útil contar con un paquete que pueda ejecutarse en casi todos los sistemas operativos y ayude a simplificar los detalles para que los estudiantes puedan concentrarse en los problemas matemáticos reales. El apéndice B es un tutorial sobre MATLAB que puede utilizarse como una primera introducción a los estudiantes o como una referencia para quienes ya estén familiarizados con este paquete.

En el sitio web de este libro (`www.pearsonenespañol.com/sauer`) encontrará todos los programas de MATLAB utilizados en el texto. Además, constantemente se publica material nuevo y actualizaciones que los usuarios de este texto pueden descargar.

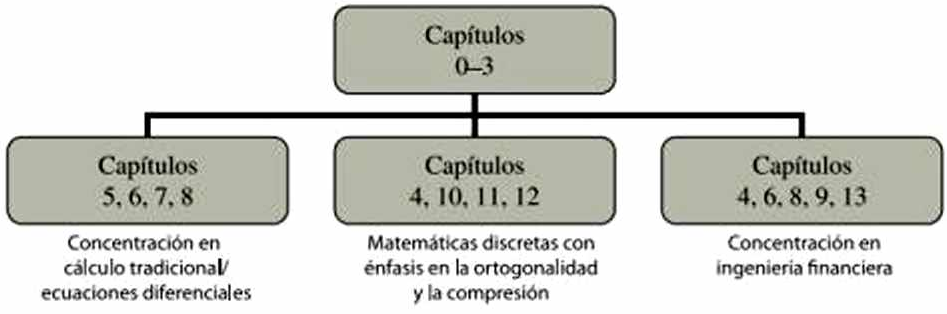
COMPLEMENTOS. El **Instructor's Solutions Manual** (Manual de soluciones para el profesor, ISM: 0-321-783689) contiene soluciones detalladas de los ejercicios impares y las respuestas a los ejercicios pares. Este manual, en inglés, también muestra la manera de utilizar el software MATLAB como una ayuda para resolver los tipos de problemas que se presentan en los ejercicios y en los problemas de computadora.

DISEÑO DEL CURSO. Este libro está estructurado para pasar de las ideas fundamentales y elementales a los conceptos más complejos y sofisticados. En el capítulo 0 se proporcionan las piezas fundamentales, que se usarán más adelante. A algunos profesores les gusta comenzar por el principio, mientras que otros (entre ellos el autor) prefieren empezar por el capítulo 1 y "regresar" a los temas del capítulo 0 cuando sea necesario. En los capítulos 1 y 2 se estudia la resolución de ecuaciones en sus diversas formas. Los capítulos 3 y 4 tratan principalmente del ajuste de datos, la interpolación y los métodos de mínimos cuadrados. En los capítulos 5 a 8 se revisan dos áreas clásicas del análisis numérico de las matemáticas continuas: la diferenciación e integración numérica, y la solución de ecuaciones diferenciales ordinarias y parciales con condiciones iniciales y de frontera.

En el capítulo 9 se desarrollan números aleatorios con el fin de proporcionar métodos complementarios a los presentados en los capítulos 5 a 8: la alternativa Monte Carlo para los esquemas de integración numérica estándar y el contrapunto de las ecuaciones diferenciales estocásticas son necesarios cuando el modelo presenta incertidumbre.

La compresión es un tema central del análisis numérico, aunque a menudo se oculta tras la interpolación por mínimos cuadrados y el análisis de Fourier. En los capítulos 10 y 11 se presentan las técnicas modernas de compresión. En el primero, se estudia la transformada rápida de Fourier como un dispositivo para llevar a cabo la interpolación trigonométrica, en los sentidos exacto y por mínimos cuadrados. También se destaca la relación de las técnicas mencionadas con la compresión de audio, y ésta se aborda por completo en el capítulo 11 con la transformada discreta del coseno, que es el caballo de batalla más común para la compresión moderna de audio e imágenes. En el capítulo 12, que trata sobre los valores propios y singulares, se describen las conexiones de estos valores con la compresión de datos, la cual crece en importancia en las aplicaciones contemporáneas. Finalmente, el capítulo 13 proporciona una introducción breve a las técnicas de optimización.

Análisis numérico puede utilizarse igualmente para un curso de un semestre, mediante una elección precisa de los temas. Los capítulos 0 a 3 son fundamentales para cualquier curso del área y los demás pueden separarse de la siguiente manera:



AGRADECIMIENTOS

Esta segunda edición está en deuda con una gran cantidad de personas, incluyendo a los estudiantes de muchas clases que han leído y comentado las versiones anteriores. Además, Paul Lorczak, Maurino Bautista y Tom Wegleitner fueron de gran ayuda para evitar que cometiera errores graves. Aprecio mucho las sugerencias de Nicholas Allgaier, Regan Beckham, Paul Calamai, Mark Friedman, David Hiebeler, Ashwani Kapila, Andrew Knyazev, Bo Li, Yijang Li, Jeff Parker, Robert Sachs, Evelyn Sander, Gantumur Tsogtgerel y Thomas Wanner. El personal de apoyo en Pearson, que incluye a William Hoffman, Caroline Celano, Beth Houston, Jeff Weidenaar y Brandon Rawnsley; así como Shiny Rajesh de Integra-PDY, ha hecho que la producción de esta segunda edición haya sido casi agradable. Por último, agradezco a los lectores de otras universidades por su impulso y asesoría en este proyecto, y por su consejo, indispensable para mejorar las versiones anteriores:

Eugene Allgower	Colorado State University
Constantin Bacuta	University of Delaware
Michele Benzi	Emory University
Jerry Bona	University of Illinois at Chicago
George Davis	Georgia State University
Chris Danforth	University of Vermont
Alberto Delgado	Bradley University
Robert Dillon	Washington State University
Qiang Du	Pennsylvania State University
Ahmet Duran	University of Michigan, Ann Arbor
Gregory Goeckel	Presbyterian College
Herman Gollwitzer	Drexel University
Don Hardcastle	Baylor University
David R. Hill	Temple University
Hideaki Kaneko	Old Dominion University
Daniel Kaplan	Macalester College
Fritz Keinert	Iowa State University
Akhtar A. Khan	Rochester Institute of Technology
Lucia M. Kimball	Bentley College
Colleen M. Kirk	California Polytechnic State University
Seppo Korpela	Ohio State University
William Layton	University of Pittsburgh
Brenton LeMesurier	College of Charleston
Melvin Leok	University of California, San Diego

Doron Levy	Stanford University
Shankar Mahalingam	University of California, Riverside
Amnon Meir	Auburn University
Peter Monk	University of Delaware
Joseph E. Pasciak	Texas A&M University
Jeff Parker	Harvard University
Steven Pav	University of California, San Diego
Jacek Polewczak	California State University
Jorge Rebaza	Southwest Missouri State University
Jeffrey Scroggs	North Carolina State University
Sergei Suslov	Arizona State University
Daniel Szyld	Temple University
Ahlam Tannouri	Morgan State University
Jin Wang	Old Dominion University
Bruno Welfert	Arizona State University
Nathaniel Whitaker	University of Massachusetts

Análisis numérico

CAPÍTULO

0

Fundamentos

Este capítulo introductorio proporciona los elementos esenciales para la elaboración y la comprensión de los algoritmos presentados en el libro. Éstos incluyen las ideas fundamentales del cálculo introductorio y de la evaluación de funciones, los detalles de la aritmética de máquina* tal como se lleva a cabo en las computadoras modernas, y el análisis de la pérdida de cifras significativas debido a cálculos mal diseñados.

Después de analizar los métodos eficientes para la evaluación de polinomios, se estudia el sistema numérico binario, la representación de números de punto flotante y los protocolos comunes que se utilizan para el redondeo. Los efectos de los pequeños errores de redondeo en los cálculos se magnifican en los problemas mal condicionados. La batalla para limitar estos efectos dañinos es un tema que se repetirá en el resto de los capítulos.

*Uno de los componentes de las computadoras modernas es la Unidad Lógica Aritmética (ALU, por sus siglas en inglés) la cual es encargada de realizar las operaciones aritméticas.

El objetivo de este libro es presentar y analizar los métodos de resolución de problemas matemáticos con computadoras. Las operaciones más fundamentales de la aritmética son la suma y la multiplicación. Éstas también son las operaciones necesarias para evaluar un polinomio $P(x)$ en un valor particular x. No es casualidad que los polinomios sean los elementos básicos para muchas técnicas de computación que se desarrollarán aquí.

Debido a lo anterior, es importante saber cómo se evalúa un polinomio. Es probable que el lector ya sepa cómo hacerlo y pueda pensar que invertir tiempo en un problema tan fácil ¡es un poco ridículo! Pero entre más básica sea una operación, mayor beneficio puede obtenerse al realizarla correctamente. Por lo tanto, ahora es importante considerar cómo implementar la evaluación de polinomios de una manera tan eficiente como sea posible.

0.1 EVALUACIÓN DE UN POLINOMIO

¿Cuál es la mejor manera de evaluar

$$P(x) = 2x^4 + 3x^3 - 3x^2 + 5x - 1,$$

por ejemplo, en $x = 1/2$? Suponga que los coeficientes del polinomio y el número 1/2 se almacenan en la memoria y trate de disminuir al mínimo el número de sumas y multiplicaciones requeridas

para obtener $P(1/2)$. Para simplificar las cosas, no se tomará en cuenta el tiempo que se invierte en el almacenamiento y obtención de números desde y hacia la memoria.

MÉTODO 1 El primer enfoque, y el más sencillo, es

$$P\left(\frac{1}{2}\right) = 2 * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} + 3 * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} - 3 * \frac{1}{2} * \frac{1}{2} + 5 * \frac{1}{2} - 1 = \frac{5}{4}. \quad (0.1)$$

El número de multiplicaciones requeridas es 10, junto con 4 adiciones. Dos de las adiciones son en realidad sustracciones, pero como la sustracción puede ser vista como la adición de un número negativo almacenado, no es necesario preocuparse por la diferencia.

Lo más seguro es que hay una mejor manera de realizar esta tarea que la mostrada por la ecuación (0.1). El esfuerzo se está duplicando (es posible ahorrarse operaciones al eliminar la multiplicación repetida de la entrada 1/2). Una mejor estrategia consiste en calcular primero $(1/2)^4$, almacenando los productos parciales a medida que se avanza. Esto conduce al siguiente método:

MÉTODO 2 Encontrar primero las potencias del número de entrada $x = 1/2$, y almacenarlas para su uso futuro:

$$\frac{1}{2} * \frac{1}{2} = \left(\frac{1}{2}\right)^2$$
$$\left(\frac{1}{2}\right)^2 * \frac{1}{2} = \left(\frac{1}{2}\right)^3$$
$$\left(\frac{1}{2}\right)^3 * \frac{1}{2} = \left(\frac{1}{2}\right)^4.$$

Ahora es posible sumar los términos:

$$P\left(\frac{1}{2}\right) = 2 * \left(\frac{1}{2}\right)^4 + 3 * \left(\frac{1}{2}\right)^3 - 3 * \left(\frac{1}{2}\right)^2 + 5 * \frac{1}{2} - 1 = \frac{5}{4}.$$

Ahora se tienen 3 multiplicaciones de 1/2, junto con otras 4 multiplicaciones. Al hacer el conteo, se ha logrado una reducción a 7 multiplicaciones, con las mismas 4 adiciones. ¿Es una mejora significativa la reducción de 14 a 11 operaciones? Si sólo debe hacerse una evaluación, probablemente no lo sea. Si se usa el método 1 o el método 2, la respuesta estará disponible antes de poder quitar los dedos del teclado de la computadora. Sin embargo, imagine que es necesario evaluar el polinomio con diferentes entradas x varias veces por segundo. Entonces la diferencia puede ser crucial para obtener la información cuando se necesita.

¿Es lo mejor que se puede hacer con un polinomio de grado 4? Resulta difícil imaginar que sea posible eliminar otras tres operaciones, pero sí se puede. El mejor método elemental es el siguiente:

MÉTODO 3 (Multiplicación anidada) Volver a escribir el polinomio de modo que pueda evaluarse de adentro hacia afuera:

$$\begin{aligned} P(x) &= -1 + x(5 - 3x + 3x^2 + 2x^3) \\ &= -1 + x(5 + x(-3 + 3x + 2x^2)) \\ &= -1 + x(5 + x(-3 + x(3 + 2x))) \\ &= -1 + x * (5 + x * (-3 + x * (3 + x * 2))). \end{aligned} \quad (0.2)$$

Aquí el polinomio está escrito en sentido inverso, y las potencias de x se factorizan para el resto del polinomio. Una vez que pueda ver cómo se escribe de esta manera (no se requiere hacer ningún cálculo para realizar la reescritura) los coeficientes no cambian. Ahora evalúe de adentro hacia afuera:

$$\begin{aligned}
&\text{multiplique } \frac{1}{2} * 2, && \text{sume } + 3 \rightarrow 4\\
&\text{multiplique } \frac{1}{2} * 4, && \text{sume } - 3 \rightarrow - 1\\
&\text{multiplique } \frac{1}{2} * -1, && \text{sume } + 5 \rightarrow \frac{9}{2}\\
&\text{multiplique } \frac{1}{2} * \frac{9}{2}, && \text{sume } - 1 \rightarrow \frac{5}{4}.
\end{aligned} \tag{0.3}$$

Este método, llamado **multiplicación anidada** o **método de Horner**, evalúa el polinomio en 4 multiplicaciones y 4 sumas. Un polinomio de grado general d puede evaluarse en d multiplicaciones y d adiciones. La multiplicación anidada se relaciona estrechamente con la división sintética de la aritmética polinomial.

El ejemplo de la evaluación de un polinomio es característico en todo el tema de métodos de cómputo para el cálculo científico. En primer lugar, las computadoras son muy rápidas al realizar cosas muy simples. En segundo lugar, es importante ejecutar incluso las tareas más sencillas tan eficientemente como sea posible, puesto que pueden llevarse a cabo muchas veces. En tercer lugar, la mejor manera de hacerlo puede no ser la más obvia. Durante el último medio siglo, los campos del análisis numérico y el cálculo científico, de la mano con la tecnología del hardware de computación, han desarrollado técnicas de solución eficientes para abordar los problemas comunes.

Aunque la forma estándar de un polinomio $c_1 + c_2x + c_3x^2 + c_4x^3 + c_5x^4$ puede escribirse en forma anidada como

$$c_1 + x(c_2 + x(c_3 + x(c_4 + x(c_5)))), \tag{0.4}$$

algunas aplicaciones requieren una forma más general. En particular, los cálculos de interpolación del capítulo 3 requerirán la forma

$$c_1 + (x - r_1)(c_2 + (x - r_2)(c_3 + (x - r_3)(c_4 + (x - r_4)(c_5)))), \tag{0.5}$$

donde r_1, r_2, r_3 y r_4 se denominan los **puntos base**. Observe que al establecer $r_1 = r_2 = r_3 = r_4 = 0$ en (0.5) se recupera la forma original anidada (0.4).

El siguiente código* de MATLAB implementa la forma general de multiplicación anidada (compárelo con (0.3)):

```
%Programa 0.1 Multiplicación anidada
%Evalúa un polinomio de forma anidada usando el método de Horner
%Entrada: grado d del polinomio,
%         arreglo de d+1 coeficientes c (primero el término constante),
%         coordenada x donde se va a evaluar y
%         arreglo de d puntos base b, si es necesario
%Salida:  valor y del polinomio en x
function y=nest (d, c, x, b)
if nargin<4, b=zeros(d, 1); end
y=c (d+1);
for i=d:-1:1
  y = y.*(x-b(i))+c(i);
end
```

La ejecución de esta función de MATLAB consiste en sustituir los datos de entrada, que son el grado, los coeficientes, los puntos de evaluación y los puntos base. Por ejemplo, el polinomio (0.2) puede evaluarse en $x = 1/2$ mediante el comando de MATLAB

* Si su equipo no acepta acentos u otros símbolos en los comentarios, omítalos.

```
>> nest(4,[-1 5 -3 3 2],1/2,[0 0 0 0])

ans =

    1.2500
```

como ya se determinó previamente en forma manual. El archivo `nest.m`, como el resto del código MATLAB que se muestra en este libro, debe estar accesible desde la ruta de MATLAB (o en el directorio actual) al ejecutar el comando.

Si se va a utilizar el comando `nest` con todos los puntos base en 0 como en (0.2), puede usarse la forma abreviada

```
>> nest(4,[-1 5 -3 3 2],1/2)
```

con el mismo resultado. Lo anterior se debe a la declaración `nargin` en `nest.m`. Si el número de argumentos de entrada es menor que 4, los puntos base se establecen en cero de manera automática.

Debido al tratamiento transparente que MATLAB da a la notación vectorial, el comando `nest` puede evaluar arreglo de valores de x a la vez. Esto se ilustra con el código siguiente:

```
>> nest(4,[-1 5 -3 3 2],[-2 -1 0 1 2])

ans =

   -15   -10    -1     6    53
```

Por último, el polinomio de interpolación de grado 3

$$P(x) = 1 + x\left(\frac{1}{2} + (x-2)\left(\frac{1}{2} + (x-3)\left(-\frac{1}{2}\right)\right)\right)$$

del capítulo 3 con puntos base $r_1 = 0$, $r_2 = 2$, $r_3 = 3$, puede evaluarse en $x = 1$ mediante

```
>> nest(3,[1 1/2 1/2 -1/2],1,[0 2 3])

ans =

     0
```

► **EJEMPLO 0.1** Encuentre un método eficaz para evaluar el polinomio $P(x) = 4x^5 + 7x^8 - 3x^{11} + 2x^{14}$.

Si se realiza una reescritura parcial del polinomio es posible reducir el esfuerzo de cálculo requerido para la evaluación. La idea es factorizar x^5 en cada término y escribir la expresión en términos de x^3:

$$\begin{aligned} P(x) &= x^5(4 + 7x^3 - 3x^6 + 2x^9) \\ &= x^5 * (4 + x^3 * (7 + x^3 * (-3 + x^3 * (2)))). \end{aligned}$$

Para cada entrada x, primero debe calcularse $x * x = x^2$, $x * x^2 = x^3$, y $x^2 * x^3 = x^5$. Estas tres multiplicaciones, combinadas con la multiplicación de x^5, y las tres multiplicaciones y tres adiciones del grado 3 en el polinomio x^3 dan un conteo total de operaciones de 7 multiplicaciones y 3 sumas por evaluación. ◄

0.1 Ejercicios

1. Reescriba los siguientes polinomios de manera anidada. Evalúe en forma anidada y sin anidar para $x = 1/3$.
   (a) $P(x) = 6x^4 + x^3 + 5x^2 + x + 1$
   (b) $P(x) = -3x^4 + 4x^3 + 5x^2 - 5x + 1$
   (c) $P(x) = 2x^4 + x^3 - x^2 + 1$

2. Reescriba los siguientes polinomios en forma anidada y evalúe en $x = -1/2$:
   (a) $P(x) = 6x^3 - 2x^2 - 3x + 7$
   (b) $P(x) = 8x^5 - x^4 - 3x^3 + x^2 - 3x + 1$
   (c) $P(x) = 4x^6 - 2x^4 - 2x + 4$

3. Evalúe $P(x) = x^6 - 4x^4 + 2x^2 + 1$ en $x = 1/2$, considerando $P(x)$ como un polinomio en x^2 y utilice la multiplicación anidada.

4. Evalúe el polinomio anidado con los puntos base $P(x) = 1 + x(1/2 + (x - 2)(1/2 + (x - 3)(-1/2)))$ en (a) $x = 5$ y (b) $x = -1$.

5. Evalúe el polinomio anidado con los puntos base $P(x) = 4 + x(4 + (x - 1)(1 + (x - 2)(3 + (x - 3)(2))))$ en (a) $x = 1/2$ y (b) $x = -1/2$.

6. Explique cómo evaluar el polinomio para una entrada x dada, usando el menor número posible de operaciones. ¿Cuántas multiplicaciones y cuántas adiciones son necesarias?
   (a) $P(x) = a_0 + a_5x^5 + a_{10}x^{10} + a_{15}x^{15}$
   (b) $P(x) = a_7x^7 + a_{12}x^{12} + a_{17}x^{17} + a_{22}x^{22} + a_{27}x^{27}$.

7. ¿Cuántas adiciones y multiplicaciones son necesarias para evaluar un polinomio de grado n con puntos base, utilizando el algoritmo general de multiplicación anidada?

0.1 Problemas de computadora

1. Utilice la función `nest` para evaluar $P(x) = 1 + x + \ldots + x^{50}$ en $x = 1.00001$. (Utilice el comando `ones` de MATLAB para ahorrar tiempo de escritura). Encuentre el error del cálculo al compararlo con la expresión equivalente $Q(x) = (x^{51} - 1)/(x - 1)$.

2. Utilice `nest.m` para evaluar $P(x) = 1 - x + x^2 - x^3 + \cdots + x^{98} - x^{99}$ en $x = 1.00001$. Encuentre una expresión equivalente más sencilla y utilícela para estimar el error de la multiplicación anidada.

0.2 NÚMEROS BINARIOS

Como preparación para el estudio detallado de la aritmética de computadora de la próxima sección, es necesario comprender el sistema numérico binario. Los números decimales se convierten de base 10 a base 2 con el fin de almacenar números en una computadora y para simplificar las operaciones hechas por la computadora, como la suma y la multiplicación. Para dar la salida en notación decimal, el proceso se invierte. En esta sección se analizan las formas para convertir entre números decimales y binarios.

Los números binarios se expresan como

$$\ldots b_2b_1b_0.b_{-1}b_{-2}\ldots,$$

donde cada dígito binario, o **bit**, es 0 o 1. El equivalente en base 10 de un número es

$$\ldots b_2 2^2 + b_1 2^1 + b_0 2^0 + b_{-1} 2^{-1} + b_{-2} 2^{-2} \ldots.$$

Por ejemplo, el número decimal 4 se expresa como $(100.)_2$ en base 2, y 3/4 se representa como $(0.11)_2$.

0.2.1 Decimal a binario

El número decimal 53 se representará como $(53)_{10}$ para destacar que se debe interpretar como base 10. Para convertirlo a binario, lo más simple es dividir el número en partes enteras y fraccionales, y convertir cada parte por separado. Para obtener el número $(53.7)_{10} = (53)_{10} + (0.7)_{10}$, se convertirá cada parte a binario y se combinarán los resultados.

Parte entera. Los enteros decimales se convierten a binario al dividir sucesivamente entre 2 y registrar los residuos. Los residuos, 0 o 1, se registran comenzando en el punto decimal (o más exactamente, la **base**) y después alejándose de éste (hacia la izquierda). Para $(53)_{10}$, se tendría

$$\begin{aligned} 53 \div 2 &= 26 \text{ R } 1 \\ 26 \div 2 &= 13 \text{ R } 0 \\ 13 \div 2 &= 6 \text{ R } 1 \\ 6 \div 2 &= 3 \text{ R } 0 \\ 3 \div 2 &= 1 \text{ R } 1 \\ 1 \div 2 &= 0 \text{ R } 1. \end{aligned}$$

Por lo tanto, el número 53 con base 10 puede escribirse en bits como 110101, indicado como $(53)_{10} = (110101)_2$. Al comprobar el resultado se tiene $110101 = 2^5 + 2^4 + 2^2 + 2^0 = 32 + 16 + 4 + 1 = 53$.

Parte fraccional. $(0.7)_{10}$ se convierte a binario invirtiendo los pasos anteriores. Se multiplica por 2 sucesivamente y se registran las partes enteras, alejándose del punto decimal hacia la derecha.

$$\begin{aligned} .7 \times 2 &= .4 + 1 \\ .4 \times 2 &= .8 + 0 \\ .8 \times 2 &= .6 + 1 \\ .6 \times 2 &= .2 + 1 \\ .2 \times 2 &= .4 + 0 \\ .4 \times 2 &= .8 + 0 \\ &\vdots \end{aligned}$$

Tenga en cuenta que el proceso se repite después de cuatro pasos y se repetirá en forma indefinida exactamente del mismo modo. Por lo tanto,

$$(0.7)_{10} = (.1011001100110\ldots)_2 = (.\overline{1011}0)_2,$$

donde la notación de una barra superior se utiliza para denotar los bits que se repiten en forma infinita. Al unir las dos partes, se concluye que

$$(53.7)_{10} = (110101.\overline{1011}0)_2.$$

0.2.2 De binario a decimal

Para convertir un número binario a decimal, de nuevo lo mejor es separarlo en partes enteras y fraccionales.

Parte entera. Tan sólo se suman las potencias de 2 como se hizo anteriormente. El número binario $(10101)_2$: es $1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = (21)_{10}$.

Parte fraccional. Si la parte fraccional es finita (una expansión terminada en base 2), debe procederse de la misma manera. Por ejemplo,

$$(.1011)_2 = \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = \left(\frac{11}{16}\right)_{10}.$$

La única complicación surge cuando la parte fraccional no es una expansión finita en base 2. La conversión de una expansión binaria infinitamente repetitiva a una fracción decimal puede hacerse de varias maneras. Tal vez la forma más sencilla es utilizar la propiedad de cambio de la multiplicación por 2.

Por ejemplo, suponga que $x = (0.\overline{1011})_2$ debe convertirse a decimal. Multiplique x por 2^4, lo cual desplaza 4 posiciones a la izquierda en sistema binario. Luego reste la x original:

$$\begin{aligned} 2^4 x &= 1011.\overline{1011} \\ x &= 0000.\overline{1011}. \end{aligned}$$

Al restar se obtiene

$$(2^4 - 1)x = (1011)_2 = (11)_{10}.$$

Después se despeja x para obtener $x = (.\overline{1011})_2 = 11/15$ en base 10.

Como ejemplo adicional, suponga que la parte fraccional no se repite inmediatamente, como en $x = .10\overline{101}$. Al multiplicar por 22 cambia a $y = 2^2 x = 10.\overline{101}$. La parte fraccional de y, es decir $z = .\overline{101}$, se calcula como antes:

$$\begin{aligned} 2^3 z &= 101.\overline{101} \\ z &= 000.\overline{101}. \end{aligned}$$

Por lo tanto, $7z = 5$, y $y = 2 + 5/7$, $x = 2^{-2} y = 19/28$ en base 10. Un buen ejercicio consiste en comprobar este resultado al convertir 19/28 a binario y compararlo con la x original.

Los números binarios son los elementos básicos de los cálculos en máquina, pero resultan ser largos y difíciles de manejar para que los seres humanos los interpreten. En ocasiones resulta útil emplear la base 16, sólo para presentar los números de una manera más sencilla. Los **números hexadecimales** se representan mediante los 16 símbolos 0, 1, 2, ..., 9, *A*, *B*, *C*, *D*, *E*, *F*. Cada número hexadecimal puede representarse mediante 4 bits. Así, $(1)_{16} = (0001)_2$, $(8)_{16} = (1000)_2$ y $(F)_{16} = (1111)_2 = (15)_{10}$. En la siguiente sección se describirá el `format hex` de MATLAB para representar los números de máquina.

0.2 Ejercicios

1. Encuentre la representación binaria de los números enteros en base 10. (a) 64 (b) 17 (c) 79 (d) 227
2. Encuentre la representación binaria de los números en base 10. (a) 1/8 (b) 7/8 (c) 35/16 (d) 31/64
3. Convierta los siguientes números en base 10 a binario. Utilice la notación de la barra superior para los números binarios infinitos. (a) 10.5 (b) 1/3 (c) 5/7 (d) 12.8 (e) 55.4 (f) 0.1
4. Convierta los siguientes números en base 10 a binario. (a) 11.25 (b) 2/3 (c) 3/5 (d) 3.2 (e) 30.6 (f) 99.9

5. Encuentre los primeros 15 bits en la representación binaria de π.
6. Encuentre los primeros 15 bits en la representación binaria de e.
7. Convierta los siguientes números binarios a base de 10: (a) 1010101 (b) 1011.101 (c) $10111.\overline{01}$ (d) $110.\overline{10}$ (c) $10.\overline{110}$ (f) $110.1\overline{101}$ (g) $10.010\overline{1101}$ (b) $111.\overline{1}$
8. Convierta los siguientes números binarios a base 10: (a) 11011 (b) 110111.001 (c) $111.\overline{001}$ (d) $1010.\overline{01}$ (e) $10111.\overline{10101}$ (f) $1111.010\overline{0001}$

0.3 REPRESENTACIÓN DEL PUNTO FLOTANTE DE LOS NÚMEROS REALES

En esta sección se presenta un modelo para la aritmética de computadora de números en punto flotante. Existen varios modelos, pero para simplificar las cosas se elegirá un modelo en particular y se describirá con detalle. El modelo elegido se denomina estándar IEEE 754 de punto flotante. El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés) tiene un interés activo en el establecimiento de estándares para la industria. Su formato de aritmética de punto flotante se ha convertido en el estándar común para la precisión aritmética sencilla y doble en toda la industria de la computación.

Los errores de redondeo son inevitables cuando se usan localidades de memoria de precisión finita para representar números reales de precisión infinita. Aunque se espera que los pequeños errores cometidos durante un cálculo largo sólo tengan un efecto menor en la respuesta, en muchos casos esto resulta ser sólo una ilusión. **Los algoritmos simples, como la eliminación de Gauss o los métodos para resolver ecuaciones diferenciales, pueden aumentar los errores microscópicos a un tamaño macroscópico.** De hecho, el tema principal de este libro es ayudar al lector a reconocer cuándo un cálculo se encuentra en riesgo de ser poco fiable debido a la amplificación de los pequeños errores cometidos por las computadoras digitales y saber cómo evitar o minimizar el riesgo.

0.3.1 Formatos de punto flotante

El estándar IEEE consiste en una serie de representaciones binarias de los números reales. Un **número de punto flotante** consta de tres partes: el **signo** (+ o −), una **mantisa**, que contiene la cadena de bits significativos, y un **exponente**. Las tres partes se almacenan juntas en una sola **palabra** de computadora.

Existen tres niveles de uso general para la precisión de los números de punto flotante: precisión simple, precisión doble y precisión extendida, también conocida como precisión doble larga. El número de bits asignados a cada número de punto flotante en los tres formatos es 32, 64, y 80, respectivamente. Los bits se dividen entre las partes como se muestra a continuación:

precisión	signo	exponente	mantisa
sencilla	1	8	23
doble	1	11	52
doble larga	1	15	64

Los tres tipos de precisión funcionan en esencia de la misma manera. La forma de un número de punto flotante IEEE **normalizado** es

$$\pm 1.bbb\ldots b \times 2^p, \tag{0.6}$$

donde cada uno de los N valores de b es 0 o 1, y p es un número binario de M bits que representa el exponente. La normalización significa que, como se muestra en (0.6), el bit inicial (que se encuentra más a la izquierda) debe ser 1.

Cuando un número binario se almacena como un número de punto flotante normalizado, está "alineado a la izquierda", lo que significa que el número 1 a la extrema izquierda se desplaza justo

a la izquierda del punto base. El cambio se compensa por un cambio en el exponente. Por ejemplo, el número decimal 9, que es el 1001 en binario, se almacenaría como

$$+1.001 \times 2^3,$$

debido a un cambio de 3 bits, o multiplicación por 2^3, es necesario mover el uno que se encuentra en el extremo izquierdo a la posición correcta.

Para ser más específicos, se utilizará sólo el formato de precisión doble para la mayor parte del libro. Las precisiones sencilla y doble larga se manejan de la misma manera, con la excepción de las diferentes longitudes M y N del exponente y la mantisa. En la precisión doble, utilizada por muchos compiladores de C y MATLAB, $M = 11$ y $N = 52$.

El número 1 de precisión doble es

$$+1.\boxed{0000000000000000000000000000000000000000000000000000} \times 2^0,$$

donde el cuadro encierra los 52 bits de la mantisa. El siguiente número de punto flotante mayor que 1 es

$$+1.\boxed{0000000000000000000000000000000000000000000000000001} \times 2^0,$$

o bien $1 + 2^{-52}$.

DEFINICIÓN 0.1 El número **épsilon de máquina**, que se indica como ϵ_{maq}, es la distancia entre 1 y el menor número de punto flotante mayor que 1. Para el punto flotante estándar IEEE de precisión doble,

$$\epsilon_{maq} = 2^{-52}$$ ❐

El número decimal $9.4 = (1001.\overline{0110})_2$ se alinea a la izquierda como

$$+1.\boxed{0010110011001100110011001100110011001100110011001100} 110\ldots \times 2^3,$$

donde se han encerrado en un cuadro los primeros 52 bits de la mantisa. De aquí surge una nueva pregunta: ¿Cómo se ajusta el número binario infinito que representa 9.4 en un número finito de bits?

Es necesario truncar el número de alguna manera y, al hacerlo, necesariamente cometer un pequeño error. Un método, llamado **recorte**, consiste tan sólo en eliminar los bits que están más allá de cierto extremo final (es decir, aquellos que están más allá del 52vo bit a la derecha del punto decimal). Este protocolo es simple, pero está sesgado ya que siempre acerca el resultado a cero.

El método alternativo es el **redondeo**. En base 10, los números se redondean por lo regular si el siguiente dígito es 5 o superior, y se redondea hacia abajo en caso contrario. En binario, esto corresponde al redondeo si el bit es 1. En específico, el bit importante en el formato de precisión doble es el 53vo bit a la derecha del punto de raíz, el primero ubicado fuera del cuadro. La técnica de redondeo predeterminada, implementada por el estándar IEEE, consiste en sumar 1 al bit 52 (redondeo hacia arriba) si el bit 53 es 1, y no hacer nada con el bit 52 (redondeo hacia abajo) si el bit 53 es 0, con una excepción: si los siguientes bits después del bit 52 son 10000..., exactamente a la mitad entre arriba y abajo, se redondea hacia arriba o hacia abajo de acuerdo con la elección que haga el bit 52 igual a 0. (Aquí sólo se trata con la mantisa, ya que el signo no juega algún papel).

¿Por qué existe un caso excepcional? A excepción de este caso, la regla de redondeo significa que se busca el número de punto flotante normalizado más próximo al número original, de ahí su nombre, la regla de redondeo al número más cercano. Es igual de probable cometer un error en redondeo hacia arriba o hacia abajo. Por lo tanto, en el caso excepcional, aquél donde haya dos números flotantes igualmente distantes del punto a redondear, debe tomarse una decisión de manera que no se prefiera ir hacia arriba o hacia abajo de forma sistemática. Lo anterior se hace para tratar de evitar la posibilidad de una desviación lenta no deseada en los cálculos largos debido tan sólo a un redondeo sesgado. La elección de hacer el bit 52 final igual a 0 en el caso de un empate es algo arbitraria, pero al menos no muestra una preferencia hacia arriba o hacia abajo. El problema 8 muestra un poco por qué se hace una elección arbitraria de 0 en el caso de un empate.

Regla de redondeo IEEE al número más cercano

Para precisión doble, si el 53vo bit a la derecha del punto binario es 0, entonces redondee hacia abajo (truncar después de 52vo bit). Si el 53vo bit es 1, entonces redondee hacia arriba (añadir 1 en el bit 52), a menos que todos los bits conocidos a la derecha del 1 sean 0, en cuyo caso se añade 1 al bit 52 si, y sólo si, el bit 52 es 1.

Para obtener el número 9.4 analizado previamente, el 53vo. bit a la derecha del punto binario es un 1 y es seguido por otros bits distintos de cero. La regla de redondeo al número más cercano dice redondear hacia arriba, o añadir 1 al 52vo bit. Por lo tanto, el número de punto flotante que representa a 9.4 es

$$+1.\boxed{0010110011001100110011001100110011001100110011001101} \times 2^3. \tag{0.7}$$

DEFINICIÓN 0.2 Indique el número de punto flotante IEEE de precisión doble asociado a x, utilizando la regla de redondeo al número más cercano, por **fl(x)**. ❐

En la aritmética de computadora, el número real x se sustituye con la cadena de bits $\text{fl}(x)$. Según esta definición, fl(9.4) es el número en representación binaria (0.7). Se llega a la representación del punto flotante al descartar la cola infinita $.\overline{1100} \times 2^{-52} \times 2^3 = .\overline{0110} \times 2^{-51} \times 2^3 = .4 \times 2^{-48}$ desde el extremo derecho del número y después añadiendo $2^{-52} \times 2^3 = 2^{-49}$ en el paso de redondeo. Por lo tanto,

$$\begin{aligned} \text{fl}(9.4) &= 9.4 + 2^{-49} - 0.4 \times 2^{-48} \\ &= 9.4 + (1 - 0.8)2^{-49} \\ &= 9.4 + 0.2 \times 2^{-49}. \end{aligned} \tag{0.8}$$

En otras palabras, una computadora que utiliza la representación de precisión doble y la regla de redondeo al número más cercano comete un error de 0.2×2^{-49} al almacenar 9.4. A este valor de 0.2×2^{-49} se le denomina **error de redondeo.**

El mensaje importante es que el número de punto flotante que representa a 9.4 no es igual a 9.4, aunque está muy cerca. Para cuantificar esa cercanía, se usa la definición estándar de error.

DEFINICIÓN 0.3 Sea x_c una versión calculada de la cantidad exacta x. Entonces

$$\textbf{error absoluto} = |x_c - x|,$$

y

$$\text{error relativo} = \frac{|x_c - x|}{|x|},$$

si esta última cantidad existe. ❐

Error de redondeo relativo

En el modelo de aritmética de computadora IEEE, el error de redondeo relativo de $\text{fl}(x)$ no es más de la mitad de la épsilon de máquina:

$$\frac{|\text{fl}(x) - x|}{|x|} \leq \frac{1}{2}\epsilon_{\text{maq}} \tag{0.9}$$

En el caso del número $x = 9.4$, se trabajó con el error de redondeo en (0.8), el cual debe satisfacer (0.9):

$$\frac{|\text{fl}(9.4) - 9.4|}{9.4} = \frac{0.2 \times 2^{-49}}{9.4} = \frac{8}{47} \times 2^{-52} < \frac{1}{2}\epsilon_{\text{maq}}$$

► EJEMPLO 0.2 Encuentre la representación fl(x) de precisión doble y el error de redondeo para $x = 0.4$.

Como $(0.4)_{10} = (.\overline{0110})_2$, al alinear a la izquierda el número binario resulta en

$$\begin{aligned} 0.4 &= 1.10\overline{0110} \times 2^{-2} \\ &= +1.\boxed{1001100110011001100110011001100110011001100110011001} \\ &\quad 100110\ldots \times 2^{-2}. \end{aligned}$$

Por lo tanto, de acuerdo con la regla de redondeo, fl(0.4) es

$$+1.\boxed{1001100110011001100110011001100110011001100110011010} \times 2^{-2}.$$

Aquí, se ha añadido 1 al 52vo bit, lo que ocasionó que el 51vo bit también cambiara, debido a la adición binaria.

Analizando cuidadosamente, se descartó $2^{-53} \times 2^{-2} + .\overline{0110} \times 2^{-54} \times 2^{-2}$ en el truncamiento y se añadió $2^{-52} \times 2^{-2}$ mediante el redondeo. Por lo tanto,

$$\begin{aligned} \text{fl}(0.4) &= 0.4 - 2^{-55} - 0.4 \times 2^{-56} + 2^{-54} \\ &= 0.4 + 2^{-54}(-1/2 - 0.1 + 1) \\ &= 0.4 + 2^{-54}(.4) \\ &= 0.4 + 0.1 \times 2^{-52}. \end{aligned}$$

Observe que el error de redondeo relativo para 0.4 es $0.1/0.4 \times \epsilon_{\text{máq}} = 1/4 \times \epsilon_{\text{máq}}$, de acuerdo con (0.9). ◄

0.3.2 Representación en máquina

Hasta ahora, se ha descrito una representación de punto flotante en abstracto. A continuación se presentan más detalles sobre cómo llevar a cabo esta representación en una computadora. Una vez más, en esta sección se empleará el formato de precisión doble, los otros formatos son muy similares.

A cada número de punto flotante de precisión doble se le asigna una palabra de 8 bytes o 64 bits, para almacenar sus tres partes. Cada una de estas palabras tiene una forma

$$\boxed{se_1e_2\ldots e_{11}b_1b_2\ldots b_{52}}, \tag{0.10}$$

donde se almacena el signo, seguida de 11 bits que representan el exponente y los 52 bits después del punto decimal, que representa la mantisa. La bit de signo s es 0 para un número positivo y 1 para un número negativo. Los 11 bits que representan el exponente provienen del entero binario positivo resultante de la adición de $2^{10} - 1 = 1023$ para el exponente, al menos para los exponentes entre -1022 y 1023. Esto cubre valores de $e_1 \ldots e_{11}$ desde 1 hasta 2046, dejando el 0 y el 2047 para fines especiales, que se estudiarán más adelante.

El número 1023 se denomina el **sesgo del exponente** del formato de precisión doble. Se utiliza para convertir exponentes positivos y negativos a números binarios positivos para su almacenamiento en los bits de exponente. Para las precisiones sencilla y doble largo, los valores del sesgo exponente son 127 y 16383, respectivamente.

El comando `format hex` de MATLAB consiste simplemente en expresar los 64 bits del número de máquina (0.10) como 16 números hexadecimales, o de base 16, sucesivos. Así, los primeros tres números hexadecimales representan el signo y el exponente combinados, mientras que los últimos 13 contienen la mantisa.

Por ejemplo, el número 1, o

$$1 = +1.\boxed{0000000000000000000000000000000000000000000000000000} \times 2^0,$$

tiene la forma de un número de máquina de precisión doble

0	01111111111	0000000000000000000000000000000000000000000000000000

una vez que se añade el usual 1023 al exponente. Los tres primeros dígitos hexadecimales corresponden a

$$001111111111 = 3FF.$$

por lo que la representación en formato hexadecimal del número de punto flotante 1 será $3FF0000000000000$. Esto puede comprobarse al escribir `format hex` en MATLAB e introducir el número 1.

► EJEMPLO 0.3 Encuentre la representación en número de máquina hexadecimal del número real 9.4.

A partir de (0.7) se tiene que el signo es $s = 0$, el exponente es 3 y los 52 bits de la mantisa después del punto decimal son

0010	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	1101

$\rightarrow (2CCCCCCCCCCCD)_{16}$.

Al agregar 1023 para el exponente se obtiene $1026 = 2^{10} + 2$, o $(10000000010)_2$. La combinación de signo y exponente es $(010000000010)_2 = (402)_{16}$, por lo que el formato hexadecimal es $4022CCCCCCCCCCCD$. ◄

Ahora se analizarán los valores especiales del exponente 0 y 2047. Este último, 2047, se utiliza para representar ∞ si la cadena de bits de mantisa es todo ceros y NaN, que significa Not a Number (No es un número), en el caso contrario. Como 2047 se representa mediante once bits 1, o $e_1e_2 \ldots e_{11} = (111\ 1111\ 1111)_2$, los primeros doce bits de `Inf` y `-Inf` son | 0111 | 1111 | 1111 | y | 1111 | 1111 | 1111 |, respectivamente; y los restantes 52 bits (la mantisa) son cero. El número de máquina NaN también comienza | 1111 | 1111 | 1111 |, pero tiene una mantisa distinta de cero. En resumen,

número de máquina	ejemplo	formato hexadecimal
`+Inf`	1/0	7FF0000000000000
`-Inf`	−1/0	FFF0000000000000
`Nan`	0/0	FFFxxxxxxxxxxxx

donde las x indican bits que no son todos cero.

El exponente especial 0, que significa $e_1e_2 \ldots e_{11} = (000\ 0000\ 0000)_2$, también indica una desviación en la forma de punto flotante estándar. En este caso, el número de máquina se interpreta como el número de punto flotante no normalizado

$$\pm 0.\boxed{b_1b_2\ldots b_{52}} \times 2^{-1022}. \tag{0.11}$$

Es decir, *en este caso, ya no se supone que el bit ubicado más a la izquierda es* 1. Estos números no normalizados se llaman números de punto flotante subnormales. Éstos extienden el rango de números muy pequeños a unos pocos órdenes de magnitud más. Por lo tanto, $2^{-52} \times 2^{-1022} = 2^{-1074}$ es el menor número distinto de cero representable en precisión doble. Su palabra máquina es

0	00000000000	0000000000000000000000000000000000000000000000000001

Asegúrese de entender la diferencia entre el número más pequeño representable 2^{-1074} y $\epsilon_{\text{máq}} = 2^{-52}$. Es posible representar en computadora muchos números inferiores a $\epsilon_{\text{máq}}$, a pesar de que si se suman a 1 el efecto puede ser casi nulo. Por otra parte, los números de precisión doble por debajo de 2^{-1074} no pueden representarse en absoluto.

Los números subnormales incluyen el número más importante 0. De hecho, la representación subnormal incluye dos números de punto flotante diferentes, $+0$ y -0, que se tratan en los cálculos como el mismo número real. La representación en computadora de $+0$ tiene un bit de signo $s = 0$, bits de exponente $e_1 \ldots e_{11} = 00000000000$, y mantisa de 52 ceros, en pocas palabras, los 64 bits son cero. El formato hexadecimal de 0 es 0000000000000000. Para obtener el número -0 todo es exactamente igual, excepto que el bit de signo $s = 1$. En formato hexadecimal -0 es 8000000000000000.

0.3.3 Suma de números de punto flotante

La adición en computadora consiste en alinear los puntos decimales de los dos números que se van a sumar, sumarlos, y después almacenar el resultado de nuevo como un número de punto flotante. La misma suma puede hacerse con una precisión más alta (con más de 52 bits) puesto que se lleva a cabo en un registro dedicado sólo a este propósito. Luego de la suma, el resultado debe redondearse de nuevo a 52 bits después del punto binario para almacenarlo como un número de máquina.

Por ejemplo, la suma de 1 más 2^{-53} aparecería de la siguiente manera:

$$
\begin{aligned}
& 1.\boxed{00\ldots0} \times 2^0 + 1.\boxed{00\ldots0} \times 2^{-53} \\
= \; & 1.\boxed{00000000000000000000000000000000000000000000000000} \times 2^0 \\
+ \; & 0.\boxed{00000000000000000000000000000000000000000000000001} \times 2^0 \\
\hline
= \; & 1.\boxed{00000000000000000000000000000000000000000000000001} \times 2^0
\end{aligned}
$$

Lo anterior se guarda como $1. \times 2^0 = 1$, de acuerdo con la regla de redondeo. Por lo tanto, $1 + 2^{-53}$ es igual a 1 con aritmética IEEE de precisión doble. Tenga en cuenta que 2^{-53} es el mayor número de punto flotante con esta propiedad; cualquier cosa más grande que se sume a 1 se traduciría en una suma superior a 1 en la aritmética de computadora.

El hecho de que $\epsilon_{maq} = 2^{-52}$ no significa que los números menores a ϵ_{maq} sean insignificantes en el modelo IEEE. Mientras sean representables en el modelo, los cálculos con números de este tamaño son exactos, suponiendo que no se sumen o resten a números que tengan un tamaño de una unidad.

Es importante darse cuenta de que la aritmética de computadora, debido al truncamiento y redondeo que realiza, en ocasiones puede dar resultados sorprendentes. Por ejemplo, si a una computadora de precisión doble con redondeo IEEE al número más cercano se le pide almacenar 9.4, después restar 9, y luego restar 0.4, ¡el resultado será algo diferente de cero! Lo que ocurre es lo siguiente: Primero, 9.4 se almacena como $9.4 + 0.2 \times 2^{-49}$, como se muestra anteriormente. Cuando se resta 9 (tenga en cuenta que 9 puede representarse sin error), el resultado es $0.4 + 0.2 \times 2^{-49}$. Ahora, cuando se le pide a la computadora que reste 0.4, realiza la resta (como se vio en el ejemplo 0.2) del número de máquina $\text{fl}(0.4) = 0.4 + 0.1 \times 2^{-52}$, de donde resulta

$$0.2 \times 2^{-49} - 0.1 \times 2^{-52} = .1 \times 2^{-52}(2^4 - 1) = 3 \times 2^{-53}$$

en vez de cero. Éste es un número pequeño, del orden de ϵ_{maq}, pero no es cero. Dado que el tipo de datos básico de MATLAB es el número de precisión doble IEEE, es posible ilustrar este descubrimiento en una sesión de MATLAB:

```
>> format long
>> x=9.4

x =

   9.40000000000000

>> y=x-9
```

```
y =

   0.40000000000000

>> z=y-0.4

z =

     3.330669073875470e-16

>> 3*2^(-53)

ans =

     3.330669073875470e-16
```

► EJEMPLO 0.4 Encuentre la suma de punto flotante de precisión doble $(1 + 3 \times 2^{-53}) - 1$.

Por supuesto, en la aritmética real la respuesta es 3×2^{-53}. Sin embargo, la aritmética de punto flotante puede ser diferente. Tenga en cuenta que $3 \times 2^{-53} = 2^{-52} + 2^{-53}$. La primera suma es

$$\begin{aligned} & 1.\boxed{00\ldots0} \times 2^0 + 1.\boxed{10\ldots0} \times 2^{-52} \\ = \ & 1.\boxed{00000000000000000000000000000000000000000000000000000} \times 2^0 \\ + \ & 0.\boxed{00000000000000000000000000000000000000000000000000001}1 \times 2^0 \\ \hline = \ & 1.\boxed{00000000000000000000000000000000000000000000000000001}1 \times 2^0. \end{aligned}$$

Éste es de nuevo el caso excepcional de la regla de redondeo. Como el bit 52 en la suma es 1, es necesario redondear hacia arriba, lo que significa añadir 1 en el bit 52. Después de hacer esto, se obtiene

$$+ \ 1.\boxed{00000000000000000000000000000000000000000000000000010} \times 2^0,$$

que es la representación de $1 + 2^{-51}$. Por lo tanto, después de restar 1, el resultado será 2^{-51}, que es igual a $2\epsilon_{\text{máq}} = 4 \times 2^{-53}$. Una vez más, observe la diferencia entre la aritmética de computadora y la aritmética exacta. Compruebe este resultado utilizando MATLAB.

Los cálculos en MATLAB, o en cualquier compilador que realice el cálculo en punto flotante según la norma IEEE, siguen las reglas precisas descritas en esta sección. A pesar de que el cálculo de punto flotante puede dar resultados sorprendentes debido a que difiere de la aritmética exacta, siempre es predecible. El redondeo al número más cercano es la regla de redondeo predeterminada típica, aunque, si se desea, es posible cambiar a otras normas de redondeo mediante el uso de banderas en el compilador. En ocasiones, es útil comparar los resultados entre diferentes protocolos de redondeo como una manera informal para evaluar la estabilidad de un cálculo.

Puede resultar sorprendente que los pequeños errores de redondeo, de tamaño relativo ϵ_{maq}, por sí solos sean capaces de afectar cálculos significativos. En la siguiente sección se presenta un mecanismo para abordar este asunto. De manera más general, el estudio de magnificación y condicionamiento del error es un tema que se repite en los capítulos 1, 2 y posteriores.

0.3 Ejercicios

1. Convierta los siguientes números de base 10 a binario y exprese cada uno como un número de punto flotante $\text{fl}(x)$ utilizando la regla de redondeo al número más cercano: (a) 1/4 (b) 1/3 (c) 2/3 (d) 0.9

2. Convierta los siguiente números de base 10 a binario y exprese cada uno como un número de punto flotante $\text{fl}(x)$ utilizando la regla de redondeo al número más cercano: (a) 9.5 (b) 9.6 (c) 100.2 (d) 44/7

3. ¿Para cuáles números enteros positivos k es posible representar exactamente (sin error de redondeo) el número $5 + 2^{-k}$ en la aritmética de punto flotante de precisión doble?

4. Encuentre el mayor entero k para el cual $\text{fl}(19 + 2^{-k}) > \text{fl}(19)$ en la aritmética de punto flotante de precisión doble.

5. Realice manualmente las siguientes sumas con la aritmética de computadora IEEE de precisión doble, utilizando la regla de redondeo al número más cercano (compruebe sus respuestas utilizando MATLAB).

   (a) $(1 + (2^{-51} + 2^{-53})) - 1$
   (b) $(1 + (2^{-51} + 2^{-52} + 2^{-53})) - 1$

6. Realice manualmente las siguientes sumas con la aritmética de computadora IEEE de precisión doble utilizando la regla de redondeo al número más cercano:

   (a) $(1 + (2^{-51} + 2^{-52} + 2^{-54})) - 1$
   (b) $(1 + (2^{-51} + 2^{-52} + 2^{-60})) - 1$

7. Escriba cada uno de los números utilizando el comando `format hex` de MATLAB. Desarrolle sus cálculos. Después, compruebe sus respuestas con MATLAB. (a) 8 (b) 21 (c) 1/8 (d) fl(1/3) (e) fl(2/3) (f) fl(0.1) (g) fl(−0,1) (h) fl(−0.2)

8. ¿Es 1/3 + 2/3 exactamente igual a 1 si se utiliza la aritmética de punto flotante de precisión doble y la regla de redondeo al número más cercano de IEEE? Deberá utilizar fl(1/3) y fl(2/3) del ejercicio 1. ¿Esto ayuda a explicar por qué la regla se expresa tal como es?, ¿la suma sería igual si se usara el recorte después del bit 52 en vez del redondeo IEEE?

9. (a) Explique por qué puede determinar épsilon de máquina en una computadora que usa la precisión doble IEEE y la regla de redondeo IEEE al número más cercano mediante el cálculo de (7/3 − 4/3) − 1. (b) ¿(4/3 − 1/3) − 1 también resulta en $\epsilon_{\text{máq}}$? Explique esto convirtiendo a números de punto flotante y llevando a cabo la aritmética de máquina.

10. Decida si $1 + x > 1$ en la aritmética de punto flotante de precisión doble, con redondeo al número más cercano. (a) $x = 2^{-53}$ (b) $x = 2^{-53} + 2^{-60}$

11. ¿Se cumple la ley asociativa para la suma IEEE en computadora?

12. Encuentre la representación $\text{fl}(x)$ de precisión doble IEEE y determine la diferencia exacta $\text{fl}(x) - x$ para los números reales dados. Compruebe que el error de redondeo relativo no es mayor que $\epsilon_{\text{maq}}/2$. (a) $x = 1/3$ (b) $x = 3.3$ (c) $x = 9/7$

13. Existen 64 números en punto flotante de precisión doble cuyas representaciones en máquina 64 bits tienen exactamente un bit distinto de cero. Encuentre el (a) mayor, (b) el segundo mayor y (c) el menor de estos números.

14. Realice manualmente las siguientes operaciones en la aritmética de computadora IEEE de precisión doble, utilizando la regla de redondeo al número más cercano (compruebe sus respuestas, utilizando MATLAB).

    (a) (4.3 − 3.3) − 1 (b) (4.4 − 3.4) − 1 (c) (4.9 − 3.9) − 1

15. Realice manualmente las siguientes operaciones en la aritmética de computadora IEEE de precisión doble, utilizando la regla de redondeo al número más cercano.

    (a) (8.3 − 7.3) − 1 (b) (8.4 − 7.4) − 1 (c) (8.8 − 7.8) − 1

16. Encuentre la representación $fl(x)$ IEEE de precisión doble y determine la diferencia exacta $fl(x) - x$ para los números reales dados. Compruebe que el error de redondeo relativo no es mayor que $\epsilon_{máq}/2$.

(a) $x = 2.75$ (b) $x = 2.7$ (c) $x = 10/3$

0.4 PÉRDIDA DE SIGNIFICANCIA

Una de las ventajas de conocer los detalles de la aritmética de computadora es que tal conocimiento proporciona una mejor posición para entender los obstáculos potenciales en los cálculos en computadora. Un problema importante que se plantea de muchas maneras es la pérdida de dígitos significativos que resulta de la resta de números casi iguales. En su forma más simple, se trata de una afirmación evidente. Suponga que a través de un esfuerzo considerable, como parte de un cálculo largo, se han determinado dos números correctos de siete dígitos significativos, y ahora es necesario restarlos:

$$\begin{array}{r} 123.4567 \\ -\ 123.4566 \\ \hline 000.0001 \end{array}$$

El problema de la sustracción comienza con dos números de entrada que se conocen con una precisión de siete dígitos, y termina con un resultado que sólo tiene un dígito de precisión. Aunque este ejemplo es bastante sencillo, existen otros problemas relacionados con la pérdida de significancia que son más sutiles y, en muchos casos, esto puede evitarse mediante la reestructuración del cálculo.

► **EJEMPLO 0.5** Calcule $\sqrt{9.01} - 3$ en una computadora con tres dígitos decimales.

Este ejemplo aún es bastante sencillo y se presenta sólo para fines ilustrativos. En lugar de utilizar una computadora con una mantisa de 52 bits, como en el formato estándar IEEE de precisión doble, se supondrá la utilización de una computadora de tres dígitos decimales. El uso de una computadora de tres dígitos implica que el almacenamiento de cada cálculo intermedio en el proceso involucra el almacenamiento en de un número de punto flotante con una mantisa de tres dígitos. Los datos del problema (9.01 y 3.00) se dan con una precisión de tres dígitos. Como se va a emplear una computadora de tres dígitos, siendo optimistas, podría esperarse la obtención de una respuesta que será exacta hasta los tres dígitos. (Desde luego, no puede esperarse más que eso porque durante el cálculo sólo se emplean tres dígitos). Al comprobar la operación en una calculadora de mano, se observa que la respuesta correcta es aproximadamente $0.0016662 = 1.6662 \times 10^{-3}$ ¿Cuántos dígitos correctos se obtienen con la computadora de tres dígitos?

Resulta que ninguno. Como $\sqrt{9.01} \approx 3.0016662$, cuando se almacena este resultado intermedio con tres dígitos significativos se obtiene 3.00. Si se resta 3.00, se obtiene una respuesta final de 0.00. No hay dígitos significativos correctos en esta respuesta.

Sorprendentemente, existe una manera de ahorrarse este cálculo, incluso en una computadora de tres dígitos. Lo que ocasiona la pérdida de significancia es el hecho de que explícitamente se estén restando números casi iguales, $\sqrt{9.01}$ y 3. Este problema puede evitarse usando el álgebra para reescribir la expresión:

$$\begin{aligned} \sqrt{9.01} - 3 &= \frac{(\sqrt{9.01} - 3)(\sqrt{9.01} + 3)}{\sqrt{9.01} + 3} \\ &= \frac{9.01 - 3^2}{\sqrt{9.01} + 3} \\ &= \frac{0.01}{3.00 + 3} = \frac{.01}{6} = 0.00167 \approx 1.67 \times 10^{-3}. \end{aligned}$$

Aquí, se ha redondeado el último dígito de la mantisa a 7 puesto que el siguiente dígito es 6. Observe que de esta manera se obtienen los tres dígitos correctos, al menos los tres dígitos a los

que se redondea la respuesta correcta. La lección obtenida es que, siempre que sea posible, es importante evitar la resta de números casi iguales en los cálculos. ◄

El método que funcionó en el ejemplo anterior fue en esencia un truco. La multiplicación por "la expresión conjugada" es un truco que puede ayudar a reestructurar el cálculo. Con frecuencia pueden utilizarse identidades específicas, como ocurre con las expresiones trigonométricas. Por ejemplo, el cálculo de $1 - \cos x$ cuando x es cercana a cero está sujeto a pérdida de significancia. A continuación se comparará el cálculo de las expresiones

$$E_1 = \frac{1-\cos x}{\operatorname{sen}^2 x} \quad \text{y} \quad E_2 = \frac{1}{1+\cos x}$$

para un rango de números de entrada x. Se llega a E_2 multiplicando el numerador y el denominador de E_1 por $1 + \cos x$, y usando la identidad trigonométrica $\operatorname{sen}^2 x + \cos^2 x = 1$. En precisión infinita, las dos expresiones son iguales. Mediante cálculos de MATLAB con precisión doble, se obtiene la siguiente tabla:

x	E_1	E_2
1.00000000000000	0.64922320520476	0.64922320520476
0.10000000000000	0.50125208628858	0.50125208628857
0.01000000000000	0.50001250020848	0.50001250020834
0.00100000000000	0.50000012499219	0.50000012500002
0.00010000000000	0.49999999862793	0.50000000125000
0.00001000000000	0.50000004138685	0.50000000001250
0.00000100000000	0.50004445029134	0.50000000000013
0.00000010000000	0.49960036108132	0.50000000000000
0.00000001000000	0.00000000000000	0.50000000000000
0.00000000100000	0.00000000000000	0.50000000000000
0.00000000010000	0.00000000000000	0.50000000000000
0.00000000001000	0.00000000000000	0.50000000000000
0.00000000000100	0.00000000000000	0.50000000000000

La columna derecha E_2 es correcta hasta los dígitos mostrados. El cálculo de E_1, debido a la resta de números casi iguales, está teniendo problemas importantes por debajo de $x = 10^{-5}$ y no tiene dígitos significativos correctos para las entradas menores o iguales a $x = 10^{-8}$.

La expresión de E_1 ya tiene varios dígitos incorrectos para $x = 10^{-4}$ y empeora a medida que x disminuye. La expresión equivalente de E_2 no resta números casi iguales y no tiene el tipo de problemas descritos.

A menudo, la fórmula cuadrática está sujeta a pérdida de significancia. De nuevo, es fácil evitarla, siempre y cuando se sepa de su existencia y se conozca la manera de reestructurar la expresión.

► EJEMPLO 0.6 Encuentre las dos raíces de la ecuación cuadrática $x^2 + 9^{12}x = 3$.

Intente resolver este problema usando la aritmética de precisión doble, por ejemplo, con MATLAB. Nadie podrá encontrar la respuesta correcta a menos que esté consciente de la pérdida de significancia y sepa cómo contrarrestarla. El problema consiste en encontrar las dos raíces, por ejemplo, con cuatro dígitos de precisión. Hasta ahora se ve como un problema fácil. Las raíces de una ecuación cuadrática de la forma $ax^2 + bx + c = 0$ se obtienen mediante la fórmula cuadrática

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{0.12}$$

Para este problema, lo anterior se traduce como

$$x = \frac{-9^{12} \pm \sqrt{9^{24} + 4(3)}}{2}.$$

Utilizando el signo de menos se obtiene la raíz

$$x_1 = -2.824 \times 10^{11},$$

correcta hasta cuatro dígitos significativos. Para la raíz de signo más

$$x_2 = \frac{-9^{12} + \sqrt{9^{24} + 4(3)}}{2},$$

MATLAB calcula 0. Aunque la respuesta correcta es cercana a 0, esta solución no tiene cifras significativas correctas, aun cuando los números que definen el problema se especificaran exactamente (esencialmente con una cantidad infinita de dígitos correctos) y a pesar de que MATLAB calcula con aproximadamente 16 dígitos significativos (una interpretación del hecho de que el épsilon máquina de MATLAB es $2^{-52} \approx 2.2 \times 10^{-16}$). ¿Cómo puede explicarse la falla total para obtener dígitos exactos para x_2?

La respuesta es la pérdida de significancia. Es evidente que, en términos relativos, 912 y $\sqrt{9^{24} + 4(3)}$ son casi iguales. Con mayor precisión, como números de punto flotante almacenados, sus mantisas no sólo empiezan de manera similar sino que en realidad son idénticas. Cuando se restan según las indicaciones de la fórmula cuadrática, por supuesto, el resultado es cero.

¿Puede salvarse este cálculo? Es necesario solucionar el problema de la pérdida de significancia. La forma correcta de calcular x_2 es mediante la reestructuración de la fórmula cuadrática:

$$\begin{aligned} x_2 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ &= \frac{(-b + \sqrt{b^2 - 4ac})(b + \sqrt{b^2 - 4ac})}{2a(b + \sqrt{b^2 - 4ac})} \\ &= \frac{-4ac}{2a(b + \sqrt{b^2 - 4ac})} \\ &= \frac{-2c}{(b + \sqrt{b^2 - 4ac})}. \end{aligned}$$

Al sustituir a, b y c para este ejemplo se obtiene, de acuerdo con MATLAB, $x_2 = 1.062 \times 10^{-11}$, que es correcta hasta cuatro dígitos significativos de precisión, tal como lo requiere el problema. ◀

En este ejemplo se muestra que la fórmula cuadrática (0.12) debe utilizarse con precaución en los casos en que a y/o c son pequeñas comparadas con b. De manera más precisa, si $4|ac| \ll b^2$, entonces b y $\sqrt{b^2 - 4ac}$ son casi iguales en magnitud, y una de las raíces está sujeta a pérdida de significancia. Si b es positiva en esta situación, entonces las dos raíces deben calcularse como

$$x_1 = -\frac{b + \sqrt{b^2 - 4ac}}{2a} \quad \text{y} \quad x_2 = -\frac{2c}{(b + \sqrt{b^2 - 4ac})}. \tag{0.13}$$

Observe que ninguna de las fórmulas presenta una problemática por restar números casi iguales. Por otro lado, si b es negativa y $4|ac| \ll b^2$, entonces las dos raíces se calculan mejor como

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{y} \quad x_2 = \frac{2c}{(-b + \sqrt{b^2 - 4ac})}. \tag{0.14}$$

0.4 Ejercicios

1. Identifique los valores de x para los cuáles se tiene una resta de números casi iguales, y encuentre una forma alternativa para evitar el problema.

$$\text{(a)}\ \frac{1-\sec x}{\tan^2 x} \quad \text{(b)}\ \frac{1-(1-x)^3}{x} \quad \text{(c)}\ \frac{1}{1+x}-\frac{1}{1-x}$$

2. Encuentre las raíces de la ecuación $x^2 + 3x - 8^{-14} = 0$ con tres dígitos de precisión.
3. Explique cómo se calculan con mayor exactitud las dos raíces de la ecuación $x^2 + bx - 10^{-12} = 0$, donde b es un número mayor que 100.
4. Demuestre la fórmula 0.14.

0.4 Problemas de computadora

1. Calcule las siguientes expresiones en la aritmética de precisión doble (por ejemplo, utilizando MATLAB) para $x = 10^{-1},\ldots, 10^{-14}$. Después, use una forma alternativa de la expresión que no presente la problemática de resta de números casi iguales, repita el cálculo y cree una tabla de resultados. Informe el número de dígitos correctos en la expresión original para cada x.

$$\text{(a)}\ \frac{1-\sec x}{\tan^2 x} \quad \text{(b)}\ \frac{1-(1-x)^3}{x}$$

2. Encuentre el menor valor de p para el cual la expresión calculada con la aritmética de precisión doble en $x = 10^{-p}$ no tenga dígitos significativos correctos. (Sugerencia: Primero encuentre el límite de la expresión cuando $x \to 0$).

$$\text{(a)}\ \frac{\tan x - x}{x^3} \quad \text{(b)}\ \frac{e^x + \cos x - \operatorname{sen} x - 2}{x^3}$$

3. Evalúe la cantidad $a + \sqrt{a^2 + b^2}$ hasta cuatro cifras significativas correctas, donde $a = -\ 12345\ 678987654321$ y $b = 123$.
4. Evalúe la cantidad $\sqrt{c^2 + d} - c$ hasta cuatro dígitos significativos correctos, donde $c = 2468\ 86422468$ y $d = 13579$.
5. Considere un triángulo rectángulo cuyos catetos tengan una longitud 3344556600 y 1.2222222. ¿Cuánto más larga es la hipotenusa que el cateto más largo? Dé su respuesta con al menos cuatro dígitos correctos.

0.5 REPASO DE CÁLCULO

Más adelante serán necesarios algunos datos básicos importantes del cálculo. El teorema del valor intermedio y el teorema del valor medio son necesarios para la resolución de ecuaciones en el capítulo 1. El teorema de Taylor es importante para la comprensión de la interpolación en el capítulo 3, y se vuelve de vital importancia para la resolución de ecuaciones diferenciales en los capítulos 6, 7 y 8.

La gráfica de una función continua no tiene espacios. Por ejemplo, si la función es positiva para un valor x y negativa para otro, debe pasar a través de cero en algún punto. Este hecho es fundamental para poder utilizar los solucionadores de ecuaciones en el próximo capítulo. El primer teorema, ilustrado en la figura 0.1(a), generaliza este concepto.

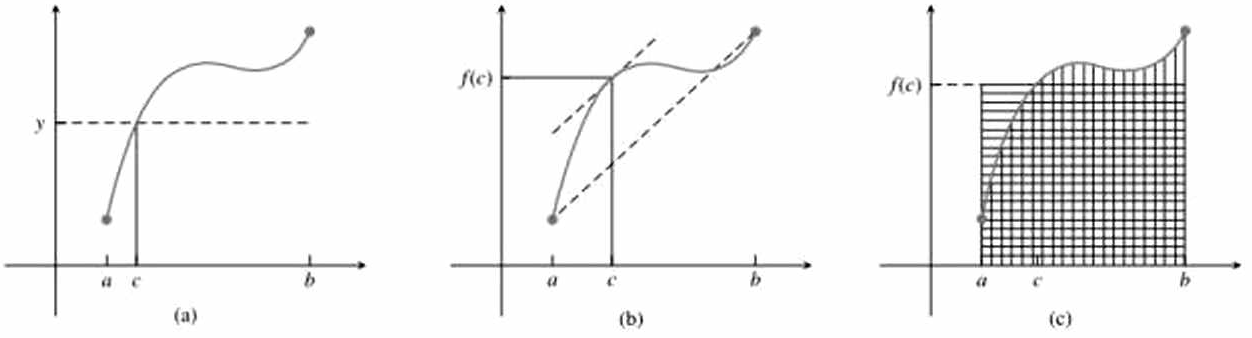


Figura 0.1 Tres teoremas importantes del cálculo. Existen números c entre a y b tales que: (a) $f(c) = y$, para cualquier y dada entre $f(a)$ y $f(b)$, por el teorema 0.4, el teorema del valor intermedio (b) la pendiente instantánea de f en c es igual a $(f(b) - f(a))/(b - a)$ por el teorema 0.6, el teorema del valor medio (c) el área de la región sombreada verticalmente es igual al área de la región sombreada horizontalmente, por el teorema 0.9, el teorema del valor medio para integrales, que se muestra en el caso especial $g(x) = 1$.

TEOREMA 0.4 (Teorema del valor intermedio) Sea f una función continua en el intervalo $[a, b]$. Entonces f reconoce cada valor entre $f(a)$ y $f(b)$. De manera más precisa, si y es un número entre $f(a)$ y $f(b)$, entonces existe un número c donde $a \leq c \leq b$ de tal modo que $f(c) = y$. ■

► **EJEMPLO 0.7** Demuestre que $f(x) = x^2 - 3$ en el intervalo $[1, 3]$ debe incluir los valores 0 y 1.

Como $f(1) = -2$ y $f(3) = 6$, todos los valores entre -2 y 6, incluyendo el 0 y el 1, debe estar incluidos en f. Por ejemplo, si $c = \sqrt{3}$, tenga en cuenta que $f(c) = f(\sqrt{3}) = 0$, y que $f(2) = 1$. ◄

TEOREMA 0.5 (Límites continuos) Sea f una función continua en la cercanía de x_0, y suponga que $\lim_{n\to\infty} x_n = x_0$. Entonces

$$\lim_{n\to\infty} f(x_n) = f\left(\lim_{n\to\infty} x_n\right) = f(x_0).$$

■

En otras palabras, los límites pueden trasladarse al interior de las funciones continuas.

TEOREMA 0.6 (Teorema del valor medio) Sea f una función continuamente diferenciable en el intervalo $[a, b]$. Entonces existe un número c entre a y b de tal forma que $f'(c) = (f(b) - f(a)) / (b - a)$. ■

► **EJEMPLO 0.8** Aplique el teorema del valor medio a $f(x) = x^2 - 3$ en el intervalo $[1, 3]$.

El contenido del teorema indica que como $f(1) = -2$ y $f(3) = 6$, debe existir un número c en el intervalo $(1, 3)$ que satisfaga $f'(c) = (6 - (-2)) / (3 - 1) = 4$. Es fácil encontrar esa c. Puesto que $f'(x) = 2x$, la c correcta es $c = 2$. ◄

El siguiente enunciado es un caso especial del teorema del valor medio.

TEOREMA 0.7 (Teorema de Rolle) Sea f una función continuamente diferenciable en el intervalo $[a, b]$, y suponga que $f(a) = (f(b)$. Entonces existe un número c entre a y b de tal forma que $f'(c) = 0$. ■

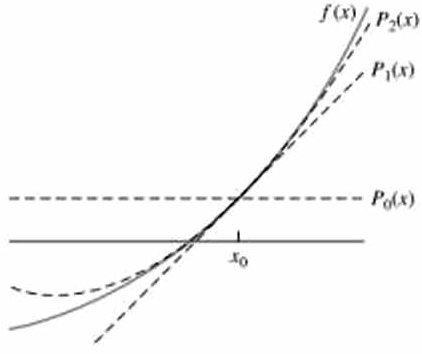


Figura 0.2 Teorema de Taylor con residuo. La función $f(x)$, indicada por la curva continua, es aproximada sucesivamente de mejor manera en la cercanía de x_0, mediante el polinomio de Taylor de grado 0 (recta discontinua horizontal), el polinomio de Taylor de grado 1 (recta discontinua inclinada), y el polinomio de Taylor de grado 2 (parábola discontinua). La diferencia entre $f(x)$ y su aproximación a x es el residuo de Taylor.

La aproximación de Taylor sustenta muchas técnicas de computación simples que se estudiarán en este libro. Si una función f se conoce bien en un punto x_0, entonces es posible conocer mucha información de f en los puntos cercanos. Si la función es continua, entonces para los puntos x cercanos a x_0, el valor de la función $f(x)$ puede aproximarse bastante bien mediante $f(x_0)$. Por otro lado, si $f'(x_0) > 0$, entonces f tiene valores mayores para los puntos cercanos a la derecha, y valores menores para los puntos cercanos a la izquierda, puesto que la pendiente cerca de x_0 está dada aproximadamente por la derivada. La línea a través de $(x_0, f(x_0))$ con pendiente $f'(x_0)$, que se muestra en la figura 0.2, es la aproximación de Taylor de grado 1. Es posible extraer otras pequeñas correcciones de las derivadas con mayor orden y obtener aproximaciones de Taylor de grados más altos. El Teorema de Taylor usa todo el conjunto de derivadas en x_0 para determinar valores de la función en una pequeña cercanía de x_0.

TEOREMA 0.8 (Teorema de Taylor con residuo) Sean x y x_0 números reales, y sea f continuamente diferenciable $k + 1$ veces en el intervalo entre x y x_0. Entonces existe un número c entre x y x_0 de tal forma que

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \cdots + \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k + \frac{f^{(k+1)}(c)}{(k+1)!}(x - x_0)^{k+1}.$$

■

La parte del resultado que es un polinomio de grado k en $x - x_0$, se denomina **polinomio de Taylor de grado k** para una f centrada en x_0. El término final se llama residuo de Taylor. En la medida que el término del residuo de Taylor sea pequeño, el Teorema de Taylor proporcionará una manera de aproximar una función general y suave con un polinomio. Lo anterior es muy conveniente en la resolución de problemas con computadoras, que, como se mencionó anteriormente, pueden evaluar polinomios de una manera muy eficiente.

► EJEMPLO 0.9 Encuentre el polinomio de Taylor $P_4(x)$ de grado 4 para $f(x) = \operatorname{sen} x$ centrada en el punto $x_0 = 0$. Estime el error máximo posible si se utiliza $P_4(x)$ para calcular sen x con $|x| \leq 0.0001$.

El polinomio se calcula fácilmente como $P_4(x) = x - x^3/6$. Observe que el término de grado 4 está ausente, puesto que su coeficiente es cero. El término del residuo es

$$\frac{x^5}{120}\cos c,$$

que en valor absoluto no puede ser mayor a $|x|^5/120$. Para $|x| \le 0.0001$, el residuo es como máximo $10^{-20}/120$ y será invisible cuando, por ejemplo, se utilice $x - x^3/6$ en precisión doble para aproximar sen 0.0001. Compruebe esto calculando ambas expresiones en MATLAB. ◀

Por último, la versión integral del teorema del valor medio se ilustra en la figura 0.1(c).

TEOREMA 0.9 (Teorema del valor medio para integrales) Sea f una función continua en el intervalo $[a, b]$, y sea g una función integrable que no cambia de signo en $[a, b]$. Entonces existe un número c entre a y b de tal forma que

$$\int_a^b f(x)g(x)\,dx = f(c)\int_a^b g(x)\,dx.$$

■

0.5 Ejercicios

1. Use el teorema del valor intermedio para demostrar que $f(c) = 0$ para alguna $0 < c < 1$.
   (a) $f(x) = x^3 - 4x + 1$ (b) $f(x) = 5\cos \pi x - 4$ (c) $f(x) = 8x^4 - 8x^2 + 1$
2. Encuentre la c que satisfaga el teorema del valor medio para $f(x)$ en el intervalo [0, 1].
   (a) $f(x) = e^x$ (b) $f(x) = x^2$ (c) $f(x) = 1/(x + 1)$
3. Encuentre la c que satisfaga el teorema del valor medio para integrales con $f(x)$, $g(x)$ en el intervalo [0, 1].
   (a) $f(x) = x$, $g(x) = x$ (b) $f(x) = x^2$, $g(x) = x$ (c) $f(x) = x$, $g(x) = e^x$
4. Encuentre el polinomio de Taylor de grado 2 en torno al punto $x = 0$ para las siguientes funciones:
   (a) $f(x) = e^{x^2}$ (b) $f(x) = \cos 5x$ (c) $f(x) = 1/(x + 1)$
5. Encuentre el polinomio de Taylor de grado 5 en torno al punto $x = 0$ para las siguientes funciones:
   (a) $f(x) = e^{x^2}$ (b) $f(x) = \cos 2x$ (c) $f(x) = \ln(1 + x)$ (d) $f(x) = \text{sen}^2 x$
6. (a) Encuentre el polinomio de Taylor de grado 4 para $f(x) = x^{-2}$ en torno al punto $x = 1$.
   (b) Utilice el resultado de (a) para aproximar $f(0.9)$ y $f(1.1)$.
   (c) Use el residuo de Taylor para encontrar una fórmula del error del polinomio de Taylor. Proporcione límites del error para cada una de las dos aproximaciones realizadas en el inciso (b). ¿Cuál de las dos aproximaciones del inciso (b) se espera que esté más cercana al valor correcto?
   (d) Utilice una calculadora para comparar el error real en cada caso con su límite de error obtenido en el inciso (c).
7. Realice los ejercicios 6(a) al 6(d) para $f(x) = \ln x$.
8. (a) Determine el polinomio de Taylor $P(x)$ de grado 5, centrado en $x = 0$ para $f(x) = \cos x$.
   (b) Encuentre un límite superior del error en la aproximación de $f(x) = \cos x$, mediante $P(x)$, para las x en $[-\pi/4, \pi/4]$.
9. Una aproximación común para $\sqrt{1+x}$ es $1 + \frac{1}{2}x$, cuando x es pequeña. Utilice el polinomio de Taylor de grado 1 de $f(x) = \sqrt{1+x}$ con residuo para determinar una fórmula de la forma $\sqrt{1+x} = 1 + \frac{1}{2}x \pm E$. Evalúe E para el caso de la aproximación de $\sqrt{1.02}$. Use una calculadora para comparar el error real con su límite de error E.

Software y lecturas adicionales

El estándar de IEEE para el cálculo del punto flotante se publicó en el estándar IEEE 754 [1985]. Goldberg [1991] y Stallings [2003] analizan la aritmética de punto flotante en gran detalle, y Overton [2001] destaca la estándar IEEE 754. Los textos de Wilkinson [1994] y Knuth [1981] tuvieron gran influencia en el desarrollo de hardware y software.

Existen varios paquetes de software que se especializan en el cálculo científico de uso general, la mayor parte de estos en relación con la aritmética de punto flotante. Netlib (http://www.netlib.org) es una colección de software gratuito mantenido por AT&T Bell Laboratories, University of Tennessee y Oak Ridge National Laboratory. La colección consta de programas de alta calidad disponibles en Fortran, C y Java, pero cuenta con poco soporte. Los comentarios en el código están pensados para ser suficientemente instructivos cuando el usuario opere el programa.

El Numerical Algorithms Group (NAG) (http://www.nag.co.uk) comercializa una biblioteca que contiene más de 1400 subrutinas accesibles para el usuario, con el propósito de resolver problemas generales de matemáticas aplicadas. Los programas están disponibles en Fortran y C, y pueden emplearse desde programas en Java. NAG incluye bibliotecas para memoria compartida y el cálculo en memoria distribuida.

La Biblioteca Internacional de Matemáticas y Estadística (IMSL, por sus siglas en inglés) es un producto de software de Rogue Wave (www.roguewave.com), y cubre áreas similares a las incluidas en la biblioteca NAG. Los programas están disponibles en Fortran, C y Java. También proporciona PV-WAVE, que es un poderoso lenguaje de programación para el análisis y la visualización de datos.

Los ambientes de cálculo Mathematica, Maple y MATLAB han crecido hasta abarcar muchos de los métodos de cálculo descritos anteriormente y tienen interfaces integradas de edición y graficación. Mathematica (http://www.wolframresearch.com) y Maple (www.maplesoft.com) se dieron a conocer gracias a sus novedosos motores de cálculo simbólico. MATLAB se ha desarrollado para dar servicio a muchas aplicaciones científicas y de ingeniería a través de sus "cajas de herramientas" que aprovechan el software básico de alta calidad en diversas direcciones.

En este libro se ilustran con frecuencia los algoritmos básicos mediante implementaciones de MATLAB. El código MATLAB que se proporciona tiene sólo propósitos de enseñanza. Muy a menudo, la velocidad y la confiabilidad se sacrifican para obtener una mayor claridad y legibilidad. Los lectores principiantes en la utilización de MATLAB deben empezar con el tutorial del apéndice B, y pronto podrán realizar sus propias implementaciones.

CAPÍTULO

1

Resolución de ecuaciones

Una placa de piedra con escritura cuneiforme recién encontrada en una excavación muestra que los babilonios calculaban la raíz cuadrada de 2 correctamente hasta una precisión de cinco decimales. Su técnica se desconoce, pero en este capítulo se presentan los métodos iterativos que podrían haber empleado y que todavía utilizan las calculadoras modernas para encontrar las raíces cuadradas.

La plataforma de Stewart, un robot con seis grados de libertad que puede localizarse con precisión extrema, fue desarrollada en un principio por Eric Gough de Dunlop Tire Corporation en la década de 1950 para poner a prueba los neumáticos de avión. Hoy sus aplicaciones van desde simuladores de vuelo, que a menudo tienen una masa considerable, hasta aplicaciones médicas y quirúrgicas, donde la precisión es muy importante. La resolución del problema directo de cinemática requiere la determinación de la posición y orientación de la plataforma, dadas las longitudes de sus puntales.

Comprobación en la realidad En la página 67 se utilizan los métodos desarrollados en este capítulo para resolver el problema directo de cinemática en una versión plana de la plataforma de Stewart.

La solución de ecuaciones es uno de los problemas más básicos en el cálculo científico. Este capítulo presenta una serie de métodos iterativos para la localización de soluciones x de la ecuación $f(x) = 0$. Estos métodos tienen una gran importancia práctica. Además, ilustran los papeles centrales de la convergencia y la complejidad en el cálculo científico.

¿Por qué es necesario conocer más de un método en la resolución de ecuaciones? Con frecuencia, la elección del método dependerá del costo de la evaluación de la función f y quizá de su derivada. Si $f(x) = e^x - \text{sen } x$, puede requerirse menos de un millonésimo de segundo para determinar $f(x)$ y, si así se requiere, su derivada también estará disponible. Si $f(x)$ indica la temperatura de congelación de una solución de glicol etileno bajo x atmósferas de presión, cada evaluación de la función puede requerir un tiempo considerable en un laboratorio bien equipado, por otro lado la determinación de la derivada puede ser irrealizable.

Además de presentar métodos como el método de bisección, la iteración de punto fijo y el método de Newton, se calcularán sus razones de convergencia y se analizará su complejidad de cálculo. Posteriormente, se estudiarán los solucionadores de ecuaciones más complejos, incluyendo el método de Brent, que combina las mejores propiedades de varios solucionadores.

1.1 EL MÉTODO DE BISECCIÓN

¿Cómo puede localizar un nombre en un directorio telefónico desconocido? Para buscar "Smith", puede comenzar por abrir el directorio en su mejor presentimiento, por ejemplo, la letra Q. A continuación, puede seguir buscando en más hojas y terminar en la letra U. Ahora usted tiene "confinado" el apellido Smith y debe centrar su búsqueda mediante límites cada vez más pequeños que finalmente lleven al apellido. El método de bisección representa este tipo de razonamiento, realizado de la manera más eficiente posible.

1.1.1 Confinamiento de una raíz

DEFINICIÓN 1.1 La función $f(x)$ tiene una **raíz** en $x = r$ si $f(r) = 0$. ❐

El primer paso para resolver una ecuación es verificar que existe una raíz. Una forma de asegurar esto consiste en confinar la raíz: encontrar un intervalo $[a, b]$ en la recta real para el cual uno de los elementos del par $\{f(a), f(b)\}$ es positivo y el otro negativo. Lo anterior puede expresarse como $f(a)\,f(b) < 0$. Si f es una función continua, entonces habrá una raíz: una r entre a y b para la cual $f(r) = 0$. Este hecho se resume en el siguiente corolario del teorema 0.4 del valor intermedio:

TEOREMA 1.2 Sea f una función continua en $[a, b]$, que satisface $f(a)\,f(b) < 0$. Entonces f tiene una raíz entre a y b; es decir, existe un número r que satisface $a < r < b$ y $f(r) = 0$. ■

En la figura 1.1, $f(0)\,f(1) = (-1)(1) < 0$. Existe una raíz justo a la izquierda de 0.7. ¿Cómo puede mejorarse esta primera aproximación de la ubicación de la raíz a más posiciones decimales?

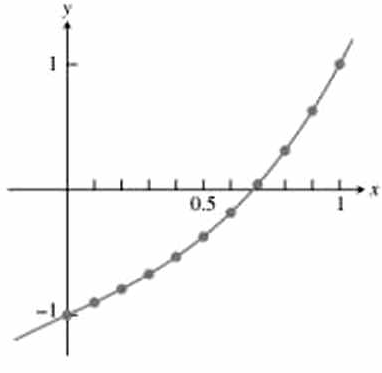


Figura 1.1 Gráfica de $f(x) = x^3 + x - 1$. La función tiene una raíz entre 0.6 y 0.7.

Se tomará como base la forma en la que el ojo humano encuentra una solución cuando se le proporciona una representación gráfica de una función. Es poco probable que comience en el extremo izquierdo del intervalo y que se mueva a la derecha, deteniéndose en la raíz. Tal vez un mejor modelo de lo que en verdad pasa es que el ojo primero decide la ubicación general, es decir, si la raíz es hacia la izquierda o la derecha del intervalo. De esto sigue la decisión más precisa de qué tan lejos a la derecha o a la izquierda se encuentra la raíz y mejorar poco a poco su precisión, de la misma manera que cuando se busca un nombre en el directorio telefónico. Este enfoque general se vuelve muy específico en el método de bisección, que se muestra en la figura 1.2.

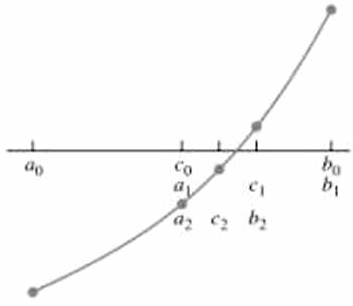


Figura 1.2 El método de bisección. En el primer paso, se verifica el signo de $f(c_0)$. Como $f(c_0)\,f(b_0) < 0$, se establece $a_1 = c_0, b_1 = b_0$, y el intervalo se sustituye por la mitad derecha $[a_1, b_1]$. En el segundo paso, el subintervalo se sustituye por su mitad izquierda $[a_2, b_2]$.

Método de bisección

Dado el intervalo inicial $[a, b]$ tal que $f(a)\,f(b) < 0$
while $(b - a)/2 >$ TOL
    $c = (a + b)/2$
    if $f(c) = 0$, **stop**, **end**
    if $f(a)\,f(c) < 0$
        $b = c$
    else
        $a = c$
    end
end
El intervalo final $[a, b]$ contiene una raíz.
La raíz aproximada es $(a + b)/2$.

Se verifica el valor de la función en el punto medio $c = (a + b)/2$ del intervalo. Como $f(a)$ y $f(b)$ tienen signos opuestos, ya sea $f(c) = 0$ (en cuyo caso se habrá encontrado una raíz y concluye el ejercicio), o el signo de $f(c)$ es opuesto al signo ya sea de $f(a)$ o $f(b)$. Si $f(c)\,f(a) < 0$, por ejemplo, se asegura una solución en el intervalo $[a, c]$, cuya longitud es la mitad del intervalo original $[a, b]$. Si en vez de esto, $f(c)\,f(b) < 0$, puede decirse lo mismo del intervalo $[c, b]$. En cualquier caso, un paso reduce el problema de encontrar una raíz en un intervalo de la mitad del tamaño original. Este paso puede repetirse para localizar la función cada vez con mayor precisión.

En cada paso, una solución estará confinada en el nuevo intervalo, reduciendo la incertidumbre en la ubicación de la solución a medida que el intervalo se vuelve más pequeño. No se requiere una gráfica completa de la función f. Se ha reducido el trabajo de evaluación de la función a lo estrictamente necesario.

► EJEMPLO 1.1 Encuentre una raíz de la función $f(x) = x^3 + x - 1$ utilizando el método de bisección en el intervalo $[0, 1]$.

Como se ha señalado, $f(a_0)\,f(b_0) = (-1)(1) < 0$, por lo que existe una raíz en el intervalo. El punto medio del intervalo es $c_0 = 1/2$. El primer paso consiste en la evaluación $f(1/2) = -3/8 < 0$ y en elegir el nuevo intervalo $[a_1, b_2] = [1/2, 1]$, puesto que $f(1/2)/(1) < 0$. El segundo paso

consiste en la evaluación $f(c_1) = f(3/4) = 11/64 > 0$, lo que conduce al nuevo intervalo $[a_2, b_2] = [1/2, 3/4]$. Al continuar de la misma manera se obtienen los siguientes intervalos:

i	a_i	$f(a_i)$	c_i	$f(c_i)$	b_i	$f(b_i)$
0	0.0000	−	0.5000	−	1.0000	+
1	0.5000	−	0.7500	+	1.0000	+
2	0.5000	−	0.6250	−	0.7500	+
3	0.6250	−	0.6875	+	0.7500	+
4	0.6250	−	0.6562	−	0.6875	+
5	0.6562	−	0.6719	−	0.6875	+
6	0.6719	−	0.6797	−	0.6875	+
7	0.6797	−	0.6836	+	0.6875	+
8	0.6797	−	0.6816	−	0.6836	+
9	0.6816	−	0.6826	+	0.6836	+

Se concluye a partir de la tabla que la solución se encuentra confinada entre $a_9 = 0.6816$ y $c_9 = 0.6826$. El punto medio de ese intervalo $c_{10} = 0.6821$ es la mejor estimación de la raíz.

Aunque el problema consiste en determinar una raíz, lo que se ha encontrado en realidad es un intervalo [0.6816, 0.6826] que contiene una raíz; en otras palabras, la raíz es $r = 0.6821 \pm 0.0005$. Habrá que conformarse con una aproximación. Por supuesto, la aproximación puede mejorarse, si es necesario, completando más pasos del método de bisección. ◄

En cada paso del método de bisección se calcula el punto medio $c_i = (a_i + b_i)/2$ del intervalo actual $[a_i, b_i]$, se calcula $f(c_i)$ y se comparan los signos. Si $f(c_i)\, f(a_i) < 0$, se establece $a_{i+1} = a_i$ y $b_{i+1} = c_i$. En cambio, si $f(c_i) f(a_i) > 0$, se establece $a_{i+1} = c_i$ y $b_{i+1} = b_i$. Cada paso requiere una nueva evaluación de la función f y una bisección del intervalo que contiene una raíz, reduciendo su longitud en la mitad. Después de n etapas de cálculo de c y $f(c)$, se habrán realizado $n + 2$ evaluaciones de la función, y la mejor estimación de la solución es el punto medio del último intervalo. El algoritmo puede escribirse en el siguiente código de MATLAB:

```
%Programa 1.1 Método de bisección
%Calcula una solución aproximada de f(x) = 0
%Entrada: Expresión de la función f; a, b de tal forma que f(a)*f(b) < 0,
%         y la tolerancia tol
%Salida: Solución aproximada xc
function: xc=bisect (f, a, b, tol)
it sign (f(a)) *sign (f(b)) >= 0
  error('¡ no se satisface f(a)f(b) < 0!') %termina la ejecución
end
fa=f(a);
fb=f(b)
while (b-a)/2>tol
  c= (a+b) /2;
  fc=f(c);
  if fc == 0                %c es una solución, fin
```

```
    break
  end
  if sign (fc) *sign (fa)<0  %a y c forman el nuevo intervalo
    b=c; fb=fc;
  else                       %c y b forman el nuevo intervalo
    a=c;fa=fc
  end
end
xc=(a+b)/2                   %el nuevo punto medio es la mejor estimación
```

Para utilizar `bisect.m`, primero debe definirse una función de MATLAB mediante:

```
>> f =@(x) x^3+x-1;
```

En realidad, este comando define un "apuntador de la función" `f`, que puede usarse como entrada para otras funciones de MATLAB. Consulte el apéndice B para obtener más detalles sobre las funciones de MATLAB y la expresión de funciones. Después el comando

```
>> xc=bisect (f, 0, 1, 0.00005)
```

regresa una solución correcta con una tolerancia de 0.00005.

1.1.2 ¿Qué tan exacto y a qué velocidad?

Si $[a, b]$ es el intervalo de inicio, entonces después de n pasos de bisección, el intervalo $[a_n, b_n]$ tiene longitud $(b - a)/2^n$. Si se elige el punto medio $x_c = (a_n + b_n)/2$, se obtiene una mejor estimación de la solución r, que está dentro de la mitad de la longitud del intervalo de la solución verdadera. En resumen, después de n pasos del método de bisección, se tiene que

$$\text{Error de la solución} = |x_c - r| < \frac{b-a}{2^{n+1}} \tag{1.1}$$

y

$$\text{Evaluaciones de la función} = n + 2. \tag{1.2}$$

Una buena manera de evaluar la eficiencia del método de bisección es preguntar cuánta precisión puede obtenerse por cada evaluación de la función. Cada paso, o cada evaluación de la función, reduce la incertidumbre en la raíz por un factor de dos.

DEFINICIÓN 1.3 Una solución es **correcta en p posiciones decimales** si el error es menor que 0.5×10^{-p}. ❐

► EJEMPLO 1.2 Use el método de bisección para encontrar una raíz de $f(x) = \cos x - x$ en el intervalo $[0, 1]$ con hasta seis posiciones correctas.

En primer lugar, se decide el número de pasos de bisección requeridos. De acuerdo con (1.1), el error después de n pasos es $(b - a)/2^{n+1} = 1/2^{n+1}$. Según la definición de p posiciones decimales, es necesario que

$$\frac{1}{2^{n+1}} < 0.5 \times 10^{-6}$$

$$n > \frac{6}{\log_{10} 2} \approx \frac{6}{0.301} = 19.9.$$

Por lo tanto, se requieren $n = 20$ pasos. Continuando con el método de bisección, se produce la tabla siguiente:

k	a_k	$f(a_k)$	c_k	$f(c_k)$	b_k	$f(b_k)$
0	0.000000	+	0.500000	+	1.000000	−
1	0.500000	+	0.750000	−	1.000000	−
2	0.500000	+	0.625000	+	0.750000	−
3	0.625000	+	0.687500	+	0.750000	−
4	0.687500	+	0.718750	+	0.750000	−
5	0.718750	+	0.734375	+	0.750000	−
6	0.734375	+	0.742188	−	0.750000	−
7	0.734375	+	0.738281	+	0.742188	−
8	0.738281	+	0.740234	−	0.742188	−
9	0.738281	+	0.739258	−	0.740234	−
10	0.738281	+	0.738770	+	0.739258	−
11	0.738769	+	0.739014	+	0.739258	−
12	0.739013	+	0.739136	−	0.739258	−
13	0.739013	+	0.739075	+	0.739136	−
14	0.739074	+	0.739105	−	0.739136	−
15	0.739074	+	0.739090	−	0.739105	−
16	0.739074	+	0.739082	+	0.739090	−
17	0.739082	+	0.739086	−	0.739090	−
18	0.739082	+	0.739084	+	0.739086	−
19	0.739084	+	0.739085	−	0.739086	−
20	0.739084	+	0.739085	−	0.739085	−

La raíz aproximada hasta seis posiciones correctas es 0.739085. ◀

Para el método de bisección, la pregunta de cuántos pasos deben ejecutarse es simple, sólo elija la precisión deseada y calcule el número de pasos necesarios, como en (1.1). Se verá que algunos algoritmos más poderosos suelen ser menos predecibles y no tienen una expresión análoga a (1.1). En esos casos, tendrán que establecerse "criterios de detención" definidos que determinen las circunstancias en las cuales el algoritmo terminará. Incluso para el método de bisección, la precisión finita de la aritmética de computadora pondrá un límite al número de posibles dígitos correctos. Este aspecto se examinará con mayor detalle en la sección 1.3.

1.1 Ejercicios

1. Use el teorema del valor intermedio para encontrar un intervalo de longitud uno que contenga una raíz de la ecuación. (a) $x^3 = 9$ (b) $3x^3 + x^2 = x + 5$ (c) $\cos^2 x + 6 = x$

2. Use el teorema del valor intermedio para encontrar un intervalo de longitud uno que contenga una raíz de la ecuación. (a) $x^5 + x = 1$ (b) $\operatorname{sen} x = 6x + 5$ (c) $\ln x + x^2 = 3$

3. Considere las ecuaciones del ejercicio 1. Aplique dos pasos del método de bisección para encontrar una raíz aproximada a 1/8 de la raíz verdadera.

4. Considere las ecuaciones del ejercicio 2. Aplique dos pasos del método de bisección para encontrar una raíz aproximada a 1/8 de la raíz verdadera.

5. Considere la ecuación $x^4 = x^3 + 10$.

   (a) Encuentre un intervalo $[a, b]$ de longitud uno, dentro del cual la ecuación tenga una solución.

   (b) Iniciando con $[a, b]$, ¿cuántos pasos del método de bisección son necesarios para calcular la solución con una precisión de 10^{-10}? Responda con un número entero.

6. Suponga que se usa el método de bisección con intervalo de inicio $[-2, 1]$ para encontrar una raíz de la función $f(x) = 1/x$. ¿El método converge a un número real?, ¿es la raíz?

1.1 Problemas de computadora

1. Utilice el método de bisección para encontrar una raíz con hasta seis posiciones decimales correctas, (a) $x^3 = 9$ (b) $3x^3 + x^2 = x + 5$ (c) $\cos^2 x + 6 = x$
2. Utilice el método de bisección para encontrar una raíz con hasta ocho posiciones decimales correctas, (a) $x^5 + x = 1$ (b) $\operatorname{sen} x = 6x + 5$ (c) $\ln x + x^2 = 3$
3. Utilice el método de bisección para localizar todas las soluciones de las ecuaciones siguientes. Grafique la función mediante el uso del comando `plot` de MATLAB e identifique tres intervalos de longitud uno que contengan una raíz. Después, encuentre las raíces hasta seis decimales correctos, (a) $2x^3 - 6x - 1 = 0$ (b) $e^{x-2} + x^3 - x = 0$ (c) $1 + 5x - 6x^3 - e^{2x} = 0$
4. Calcule las raíces cuadradas de los números siguientes hasta ocho cifras decimales correctas utilizando el método de bisección para resolver $x^2 - A = 0$, donde A es (a) 2 (b) 3 (c) 5. Establezca su intervalo de inicio y el número de pasos necesarios.
5. Calcule las raíces cúbicas de los siguientes números hasta ocho cifras decimales correctas utilizando el método de bisección para resolver $x^3 - A = 0$, donde A es (a) 2 (b) 3 (c) 5. Establezca su intervalo de inicio y el número de pasos necesarios.
6. Utilice el método de bisección para calcular la solución de $\cos x = \operatorname{sen} x$ en el intervalo [0, 1] con seis posiciones decimales correctas.
7. Utilice el método de bisección para encontrar los dos números reales x, con seis posiciones decimales correctas, que hacen que el determinante de la matriz

$$A = \begin{bmatrix} 1 & 2 & 3 & x \\ 4 & 5 & x & 6 \\ 7 & x & 8 & 9 \\ x & 10 & 11 & 12 \end{bmatrix}$$

   sea igual a 1000. Pruebe cada solución encontrada calculando el determinante correspondiente e informe el número de cifras decimales correctas (después del punto decimal) que tiene el determinante al utilizar la solución x. (En la sección 1.2, esto se denominará "el error hacia atrás" asociado a la solución aproximada). Para calcular los determinantes, puede utilizar el comando det de MATLAB.
8. La **matriz de Hilbert** es la matriz $n \times n$ cuya ij-ésima entrada es $1/(i + j - 1)$. Indique con A la matriz de Hilbert de 5×5. Su mayor valor propio es de aproximadamente 1.567. Utilice el método de bisección para decidir cómo cambiar la entrada superior izquierda A_{11} de modo que el mayor valor propio de A sea igual a π. Determine A_{11} hasta seis posiciones decimales correctas. Para simplificar su tarea, puede utilizar los comandos `hilb`, `pi`, `eig` y `max` de MATLAB.
9. Encuentre la altura alcanzada por 1 metro cúbico de agua almacenada en un tanque esférico con un radio de 1 metro. Dé su respuesta con una precisión de ± 1 mm. (Sugerencia: tenga en cuenta que la esfera estará llena hasta menos de la mitad. El volumen de los H metros inferiores de una semiesfera de radio R es $\pi H^2(R - 1/3H)$).

1.2 ITERACIÓN DE PUNTO FIJO

Use una calculadora o una computadora para aplicar la función `cos` varias veces a un número de inicio arbitrario. Es decir, aplique la función `cos` al número de inicio, después aplique `cos` al resultado, luego al nuevo resultado y así sucesivamente. (Si utiliza una calculadora, asegúrese de que se encuentre en modo de radianes). Continúe hasta que los dígitos no cambien más. La secuencia resultante de números converge a 0.7390851332, al menos para los primeros 10 decimales. El ob-

jetivo de esta sección es explicar por qué converge este cálculo, lo cual es un ejemplo de iteración de punto fijo (IPF). Al hacer esto, se analizará la mayoría de los grandes temas de convergencia de algoritmos.

1.2.1 Puntos fijos de una función

La secuencia de números producidos mediante la iteración de la función coseno parece converger a un número r. Las aplicaciones subsecuentes del coseno no cambian el número. Para esta entrada, la salida de la función coseno es igual a la entrada, o $\cos r = r$.

DEFINICIÓN 1.4 El número real r es un **punto fijo** de la función g si $g(r) = r$. ❐

El número $r = 0.7390851332$ es una aproximación del punto fijo de la función $g(x) = \cos x$. La función $g(x) = x^3$ tiene tres puntos fijos, $r = -1, 0$ y 1.

En el ejemplo 1.2 se utilizó el método de bisección para resolver la ecuación $\cos x - x = 0$. La ecuación de punto fijo $\cos x = x$ es el mismo problema desde un punto de vista diferente. Si la salida es igual a la entrada, ese número es un punto fijo de $\cos x$ y al mismo tiempo una solución de la ecuación $\cos x - x = 0$.

Una vez que la ecuación se escribe como $g(x) = x$, procede la iteración de punto fijo, empezando con una estimación inicial x_0 para después iterar la función g.

Iteración de punto fijo

$$x_0 = \text{valor inicial}$$
$$x_{i+1} = g(x_i), \textbf{ para } i = 0, 1, 2, \ldots$$

Por lo tanto,

$$\begin{aligned} x_1 &= g(x_0) \\ x_2 &= g(x_1) \\ x_3 &= g(x_2) \\ &\vdots \end{aligned}$$

y así sucesivamente. La secuencia x_i puede converger o no a medida que el número de pasos tiende a infinito. Sin embargo, si g es continua y las x_i convergen, por ejemplo, a un número r, entonces r es un punto fijo. De hecho, el teorema 0.5 implica que

$$g(r) = g\left(\lim_{i\to\infty} x_i\right) = \lim_{i\to\infty} g(x_i) = \lim_{i\to\infty} x_{i+1} = r. \tag{1.3}$$

El algoritmo de iteración de punto fijo aplicado a una función g puede escribirse de manera sencilla en código de MATLAB:

```
%Programa 1.2 Iteración de Punto Fijo
%Calcula la solución aproximada de g(x) = x
%Entrada: expresión de la función g, iniciando en el supuesto x0,
%    número k de pasos de la iteración
%Salida: solución aproximada xc
function xc=fpi(g, x0,k)
x(1)=x0;
```

```
for i=1:k
  x(i+1)=g(x(i));
end
xc=x (k+1);
```

Después de definir una función de MATLAB mediante

```
>> g=@(x) cos(x)
```

el código del programa 1.2 puede llamarse con

```
>> xc=fpi (g, 0,10)
```

para ejecutar 10 pasos de la iteración de punto fijo con valor inicial 0.

La iteración de punto fijo resuelve el problema de punto fijo $g(x) = x$, pero sobre todo se tiene interés en la solución de ecuaciones. ¿Puede cada ecuación $f(x) = 0$ convertirse en un problema de punto fijo $g(x) = x$? Sí, y de muchas maneras diferentes. Por ejemplo, la ecuación del ejemplo 1.1, de la cual deben determinarse las raíces

$$x^3 + x - 1 = 0, \tag{1.4}$$

puede reescribirse como

$$x = 1 - x^3, \tag{1.5}$$

y es posible definir $g(x) = 1 - x^3$. De manera alternativa, el término x^3 en (1.4) puede aislarse para obtener

$$x = \sqrt[3]{1 - x}, \tag{1.6}$$

donde $g(x) = \sqrt[3]{1 - x}$. Como un tercer método, no muy evidente, se podría añadir $2x^3$ a ambos lados de (1.4) y tener

$$\begin{aligned} 3x^3 + x &= 1 + 2x^3 \\ (3x^2 + 1)x &= 1 + 2x^3 \\ x &= \frac{1 + 2x^3}{1 + 3x^2} \end{aligned} \tag{1.7}$$

luego, puede definirse $g(x) = (1 + 2x^3)/(1 + 3x^2)$.

A continuación, se muestra la iteración de punto fijo para las tres opciones de $g(x)$ anteriores. La ecuación fundamental a resolver es $x^3 + x - 1 = 0$. Primero se considera la forma $x = g(x) = 1 - x^3$. El punto de inicio, $x_0 = 0.5$, se elige de forma un tanto arbitraria. La aplicación de la IPF da el siguiente resultado:

i	x_i
0	0.50000000
1	0.87500000
2	0.33007813
3	0.96403747
4	0.10405419
5	0.99887338
6	0.00337606
7	0.99999996
8	0.00000012
9	1.00000000
10	0.00000000
11	1.00000000
12	0.00000000

En vez de converger, la iteración tiende a alternar entre los números 0 y 1. No es un punto fijo, puesto que $g(0) = 1$ y $g(1) = 0$. La iteración de punto fijo falla. Con el método de bisección, se sabe que si f es continua y $f(a)\,f(b) < 0$ en el intervalo original, debe verse una convergencia hacia la raíz. Esto no es así para la IPF.

La segunda opción es $g(x) = \sqrt[3]{1 - x}$. Se mantendrá la misma estimación inicial, $x_0 = 0.5$.

i	x_i
0	0.50000000
1	0.79370053
2	0.59088011
3	0.74236393
4	0.63631020
5	0.71380081
6	0.65900615
7	0.69863261
8	0.67044850
9	0.69072912
10	0.67625892
11	0.68664554
12	0.67922234

i	x_i
13	0.68454401
14	0.68073737
15	0.68346460
16	0.68151292
17	0.68291073
18	0.68191019
19	0.68262667
20	0.68211376
21	0.68248102
22	0.68221809
23	0.68240635
24	0.68227157
25	0.68236807

Esta vez la IPF tiene éxito. Las iteraciones aparentemente convergen a una cifra cercana a 0.6823.

Por último, se usará la reordenación $x = g(x) = (1 + 2x^3)/(1 + 3x^2)$. Como en el caso anterior, hay una convergencia pero de un modo mucho más notorio.

i	x_i
0	0.50000000
1	0.71428571
2	0.68317972
3	0.68232842
4	0.68232780
5	0.68232780
6	0.68232780
7	0.68232780

Aquí se tienen cuatro dígitos correctos después de cuatro iteraciones de la iteración de punto fijo y muchos más dígitos correctos un poco después. En comparación con los intentos anteriores, éste es un resultado sorprendente. El próximo objetivo es tratar de explicar las diferencias entre los tres resultados.

1.2.2 Geometría de la iteración de punto fijo

En la sección anterior se encontraron tres formas diferentes de reescribir la ecuación $x^3 + x - 1 = 0$ como un problema de punto fijo, con resultados variables. Para saber por qué el método de IPF converge en algunos casos y en otros no, resulta útil observar la geometría del método.

En la figura 1.3 se muestran las tres diferentes $g(x)$ presentadas previamente, junto con una ilustración de los primeros pasos de la IPF en cada caso. El punto fijo r es el mismo para todas las $g(x)$. Se representa por medio del punto en el que las gráficas de $y = g(x)$ y $y = x$ se intersecan. Cada paso de la IPF puede graficarse dibujando segmentos de recta (1) **verticalmente hacia la función** y después (2) **horizontalmente hacia la línea diagonal** $y = x$. Las flechas verticales y horizontales de la figura 1.3 siguen los pasos realizados por la IPF. La flecha vertical que se desplaza desde el valor x hasta la función g representa $x_i \rightarrow g(x_i)$. La flecha horizontal representa la salida $g(x_i)$ en el eje x y su transformación en el mismo número x_{i+1} en el eje x, listo para introducirse

en g en el siguiente paso. Esto se logra al trazar el segmento de recta horizontal desde la altura de salida $g(x_i)$ a través de la recta diagonal $y = x$. Esta ilustración geométrica de una iteración de punto fijo se llama **diagrama de telaraña**.

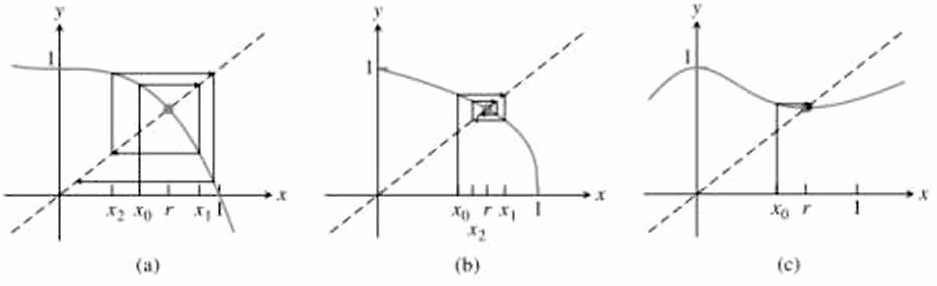


Figura 1.3 Vista geométrica de la IPF. El punto fijo es la intersección de $g(x)$ con la recta diagonal. Se muestran tres ejemplos de $g(x)$ junto con los primeros pasos de la IPF. (a) $g(x) = 1 - x^3$ (b) $g(x) = (1 - x)^{1/3}$ (c) $g(x) = (1 + 2x^3)/(1 + 3x^2)$

En la figura 1.3(a), la trayectoria inicia en $x_0 = 0.5$, y se mueve hacia arriba hacia la función y en forma horizontal hasta el punto (0.875, 0.875) sobre la diagonal, que es (x_1, x_1). A continuación, x_1 debe sustituirse en $g(x)$. Esto se hace de la misma manera que para x_0, desplazándose en forma vertical hacia la función. De lo anterior se obtiene $x_2 \approx 0.3300$ y después de mover en forma horizontal el valor de y hasta un valor de x, se continúa de la misma manera para obtener x_3, x_4, ... Como se vio anteriormente, el resultado de la IPF para esta $g(x)$ no tiene éxito (las iteraciones tienden finalmente a alternarse entre los valores de 0 y 1, ninguno de los cuales son puntos fijos).

La iteración de punto fijo es más exitosa en la figura 1.3(b). Aunque esta $g(x)$ se ve más o menos similar a la $g(x)$ del inciso (a), hay una diferencia significativa que se aclarará en la siguiente sección. Por el momento usted puede especular sobre cuál es la diferencia. ¿Qué hace que la IPF forme una espiral hacia el punto fijo en (b), y una espiral que se aleja del punto fijo en (a)? En la figura 1.3(c) se muestra un ejemplo de convergencia muy rápida. ¿Esta gráfica le ayuda con su especulación? Si usted infirió que algo tiene que ver con la pendiente de $g(x)$ cerca del punto fijo, está en lo correcto.

1.2.3 Convergencia lineal de la iteración de punto fijo

Las propiedades de convergencia de la IPF pueden explicarse con facilidad mediante una observa-ción cuidadosa del algoritmo en la situación más simple posible. En la figura 1.4 se muestra la iteración de punto fijo para dos funciones lineales $g_1(x) = -\frac{3}{2}x + \frac{5}{2}$ y $g_2(x) = -\frac{1}{2}x + \frac{3}{2}$. En cada caso, el punto fijo es $x = 1$, $|g_1'(1)| = \left|-\frac{3}{2}\right| > 1$ mientras que $|g_2'(1)| = \left|-\frac{1}{2}\right| < 1$. Si se siguen las flechas verticales y horizontales que describen la IPF, se observa la razón de la diferencia. Como la pendiente de g_1 en el punto fijo es mayor que uno, los segmentos verticales, que representan el cambio de x_n a x_{n+1}, aumentan en longitud a medida que avanza la IPF. En consecuencia, la iteración "forma espirales hacia afuera" desde el punto fijo $x = 1$, incluso si la estimación inicial x_0 es bastante cercana. Para g_2, la situación se invierte: la pendiente de g_2 es menor que uno, con lo que la longitud de los segmentos verticales disminuye, y la IPF forma "espirales hacia dentro" en torno a la solución. De este modo, $|g'(r)|$ marca la diferencia crucial entre divergencia y convergencia.

Esa es la visión geométrica. En términos de ecuaciones, resulta útil escribir $g_1(x)$ y $g_2(x)$ en función de $x - r$, donde $r = 1$ es el punto fijo:

$$\begin{aligned} g_1(x) &= -\tfrac{3}{2}(x - 1) + 1 \\ g_1(x) - 1 &= -\tfrac{3}{2}(x - 1) \\ x_{i+1} - 1 &= -\tfrac{3}{2}(x_i - 1). \end{aligned} \tag{1.8}$$

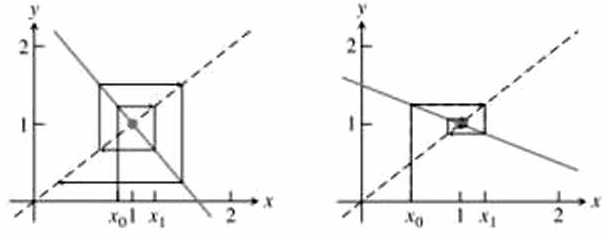


Figura 1.4 Diagrama de telaraña para funciones lineales. (a) Si la función lineal tiene una pendiente mayor que uno en valor absoluto, las estimaciones cercanas se alejarán del punto fijo a medida que progrese la IPF, lo que conduce al fracaso del método. (b) Para pendientes menores que uno en valor absoluto, ocurre lo contrario y es posible encontrar el punto fijo.

Si se considera que $e_i = |r - x_i|$ es el error en el paso i (es decir, la distancia desde la mejor estimación en el paso n hasta el punto fijo), se observa en (1.8) que $e_{i+1} = 3e_i/2$, lo que implica que los errores aumentan en cada paso por un factor de aproximadamente 3/2. Ésta es la divergencia.

Si se repite el procedimiento algebraico anterior para g_2, se tiene

$$\begin{aligned} g_2(x) &= -\tfrac{1}{2}(x-1)+1 \\ g_2(x)-1 &= -\tfrac{1}{2}(x-1) \\ x_{i+1}-1 &= -\tfrac{1}{2}(x_i-1). \end{aligned}$$

El resultado es $e_{i+1} = e_i/2$, lo que implica que el error, la distancia al punto fijo, se multiplica por 1/2 en cada paso. El error disminuye a cero cuando aumenta el número de pasos. Ésta es una convergencia de un tipo particular.

DEFINICIÓN 1.5 Sea e_i el error en el paso i de un método iterativo. Si

$$\lim_{i\to\infty} \frac{e_{i+1}}{e_i} = S < 1,$$

se dice que el método obedece a una **convergencia lineal** con razón S.

La iteración de punto fijo para g_2 es linealmente convergente a la raíz $r = 1$ con una razón de $S = 1/2$. Aunque el análisis anterior se simplificó, porque g_1 y g_2 son lineales, puede aplicarse el mismo razonamiento de manera general a una función $g(x)$ diferenciable continuamente con punto fijo $g(r) = r$, como se muestra en el siguiente teorema. ■

TEOREMA 1.6 Suponga que g es diferenciable continuamente, que $g(r) = r$ y que $S = |g'(r)| < 1$. Entonces la iteración de punto fijo converge linealmente con una razón S hacia el punto fijo r para estimaciones iniciales lo suficientemente cerca de r.

Comprobación. Sea x_i la iteración en el paso i. De acuerdo con el teorema del valor medio, existe un número c_i entre x_i y r de tal modo que

$$x_{i+1} - r = g'(c_i)(x_i - r), \tag{1.9}$$

donde se ha sustituido $x_{i+1} = g(x_i)$ y $r = g(r)$. Si se define $e_i = |x_i - r|$, (1.9) puede escribirse como

$$e_{i+1} = |g'(c_i)|e_i. \tag{1.10}$$

Si $S = |g'(r)|$ es menor que 1, entonces debido a la continuidad de g', hay una pequeña vecindad alrededor de r para la cual $|g(x)| < (S + 1)/2$, un poco mayor que S, pero todavía menor que uno. Si sucede que x_i se encuentra en esta vecindad, entonces sucede lo mismo para c_i (está confinada entre x_i y r), y así

$$e_{i+1} \leq \frac{S+1}{2} e_i.$$

Por lo tanto, el error se reduce como mínimo en un factor de $(S + 1)/2$ en este paso y en los pasos posteriores. Lo anterior significa que $\lim_{i\to\infty} x_i = r$ y si se toma el límite de (1.10), se obtiene

$$\lim_{i\to\infty} \frac{e_{i+1}}{e_i} = \lim_{i\to\infty} |g'(c_i)| = |g'(r)| = S.$$

❐

De acuerdo con el teorema 1.6, la relación del error aproximado

$$e_{i+1} \approx S e_i \tag{1.11}$$

se mantiene en el límite a medida que se aproxima la convergencia, donde $S = |g'(r)|$. En el ejercicio 25 puede verse una variante de este teorema.

DEFINICIÓN 1.7 Un método iterativo se denomina **localmente convergente** a r si converge a r para aproximaciones iniciales suficientemente cercanas a r. ❐

En otras palabras, el método converge localmente a la raíz r si existe una vecindad $(r - \epsilon, r + \epsilon)$, donde $\epsilon > 0$, de tal manera que se dé la convergencia a r a partir de todas las estimaciones iniciales en la vecindad. La conclusión del teorema 1.6 es que la iteración de punto fijo converge localmente si $|g'(r)| < 1$.

El teorema 1.6 explica lo que sucedió en las corridas anteriores de la iteración de punto fijo para $f(x) = x^3 + x - 1 = 0$. Se sabe que la raíz $r \approx 0.6823$. Para $g(x) = 1 - x^3$, la derivada es $g'(x) = -3x^2$. Cerca de la raíz r, la IPF se comporta como $e_{i+1} \approx S e_i$, donde $S = |g'(r)| = |-3(0.6823)^2| \approx 1.3966 > 1$, de modo que los errores aumentan y no puede haber convergencia. Esta relación de error entre e_{i+1} y e_i sólo se garantiza en la cercanía de r, pero sí implica que no es posible la convergencia hacia r.

Para la segunda opción, $g(x) = \sqrt[3]{1 - x}$, la derivada es $g'(x) = 1/3\,(1 - x)^{-2/3}(-1)$, y $S = |(1 - 0.6823)^{-2/3}/3| \approx 0.716 < 1$. El teorema 1.6 implica convergencia, de acuerdo con el cálculo realizado antes.

Para la tercera opción, $g(x) = (1 + 2x^3)/(1 + 3x^2)$,

$$\begin{aligned} g'(x) &= \frac{6x^2(1 + 3x^2) - (1 + 2x^3)6x}{(1 + 3x^2)^2} \\ &= \frac{6x(x^3 + x - 1)}{(1 + 3x^2)^2}, \end{aligned}$$

y $S = |g'(r)| = 0$. Esto es lo más pequeño que puede ser S, lo que conduce a una convergencia muy rápida como se ve en la figura 1.3(c).

► EJEMPLO 1.3 Explique por qué converge la iteración de punto fijo $g(x) = \cos x$.

Ésta es la explicación que se prometió al principio del capítulo. La aplicación repetida de la función coseno corresponde a la IPF con $g(x) = \cos x$. De acuerdo con el teorema 1.6, la solución $r \approx 0.74$ atrae estimaciones más cercanas porque $g'(r) = -\operatorname{sen} r \approx -\operatorname{sen} 0.74 \approx -0.67$ es menor que 1 en valor absoluto. ◄

► EJEMPLO 1.4 Use la iteración de punto fijo para encontrar una raíz de $\cos x = \operatorname{sen} x$.

La manera más sencilla de convertir la ecuación en un problema de punto fijo es sumar x a cada lado de la ecuación. El problema puede reescribirse como

$$x + \cos x - \operatorname{sen} x = x$$

y definir

$$g(x) = x + \cos x - \operatorname{sen} x. \tag{1.12}$$

El resultado de la aplicación del método de la iteración de punto fijo para esta $g(x)$ se muestra en la tabla.

i	x_i	$g(x_i)$	$e_i = \lvert x_i - r\rvert$	e_i/e_{i-1}
0	0.0000000	1.0000000	0.7853982	
1	1.0000000	0.6988313	0.2146018	0.273
2	0.6988313	0.8211025	0.0865669	0.403
3	0.8211025	0.7706197	0.0357043	0.412
4	0.7706197	0.7915189	0.0147785	0.414
5	0.7915189	0.7828629	0.0061207	0.414
6	0.7828629	0.7864483	0.0025353	0.414
7	0.7864483	0.7849632	0.0010501	0.414
8	0.7849632	0.7855783	0.0004350	0.414
9	0.7855783	0.7853235	0.0001801	0.414
10	0.7853235	0.7854291	0.0000747	0.415
11	0.7854291	0.7853854	0.0000309	0.414
12	0.7853854	0.7854035	0.0000128	0.414
13	0.7854035	0.7853960	0.0000053	0.414
14	0.7853960	0.7853991	0.0000022	0.415
15	0.7853991	0.7853978	0.0000009	0.409
16	0.7853978	0.7853983	0.0000004	0.444
17	0.7853983	0.7853981	0.0000001	0.250
18	0.7853981	0.7853982	0.0000001	1.000
19	0.7853982	0.7853982	0.0000000	

Existen varias cosas interesantes que pueden observarse en la tabla. En primer lugar, la iteración parece converger a 0.7853982. Como $\cos \pi/4 = \sqrt{2}/2 = \operatorname{sen} \pi/4$, la solución verdadera a la ecuación $\cos x - \operatorname{sen} x = 0$ es $r = \pi/4 \approx 0.7853982$. La cuarta columna es la "columna de error", que muestra el valor absoluto de la diferencia entre la mejor estimación x_i en el paso i y el punto fijo real r. Esta diferencia se vuelve pequeña cerca de la parte inferior de la tabla, lo que indica convergencia hacia un punto fijo.

Observe el patrón en la columna de error. Los errores parecen disminuir con un factor constante, cada error parece ser algo menos de la mitad del error anterior. Para una mayor precisión, la razón entre errores sucesivos se muestra en la columna final. En la mayor parte de la tabla se observa que la razón e_{k+1}/e_k de los errores sucesivos se acerca a un número constante, alrededor de 0.414. En otras palabras, se ve la relación de convergencia lineal

$$e_i \approx 0.414 e_{i-1}. \tag{1.13}$$

Esto es exactamente lo que se esperaba, puesto que el teorema 1.6 implica que

$$S = |g'(r)| = |1 - \operatorname{sen} r - \cos r| = \left|1 - \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}\right| = |1 - \sqrt{2}| \approx 0.414. \quad \blacktriangleleft$$

El lector atento se dará cuenta de una discrepancia hacia el final de la tabla. Se han utilizado sólo siete dígitos correctos para el punto fijo r correcto en el cálculo de los errores e_i. Como re-

sultado, la precisión relativa de los e_i es deficiente a medida que e_i se acerca a 10^{-8}, y las razones de e_i/e_{i-1} se vuelven incorrectas. Este problema desaparecería si se utilizara un valor mucho más preciso para r.

► **EJEMPLO 1.5** Encuentre los puntos fijos de $g(x) = 2.8x - x^2$.

La función $g(x) = 2.8x - x^2$ tiene dos puntos fijos, 0 y 1.8, que pueden determinarse al resolver $g(x) = x$ en forma manual, o de manera alternativa, al observar los puntos donde se intersecan las gráficas de $y = g(x)$ y $y = x$. En la figura 1.5 se muestra un diagrama de telaraña para la IPF con el valor inicial $x = 0.1$. Para este ejemplo, las iteraciones

$$\begin{aligned} x_0 &= 0.1000 \\ x_1 &= 0.2700 \\ x_2 &= 0.6831 \\ x_3 &= 1.4461 \\ x_4 &= 1.9579, \end{aligned}$$

y las subsecuentes, pueden leerse como las intersecciones a lo largo de la diagonal.

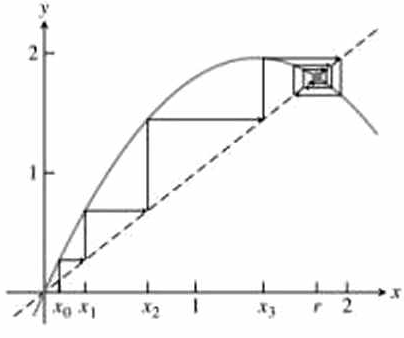


Figura 1.5 Diagrama de telaraña para la iteración de punto fijo. El ejemplo 1.5 tiene dos puntos fijos, 0 y 1.8. Se muestra una iteración con la estimación inicial 0.1. La IPF sólo convergerá a 1.8.

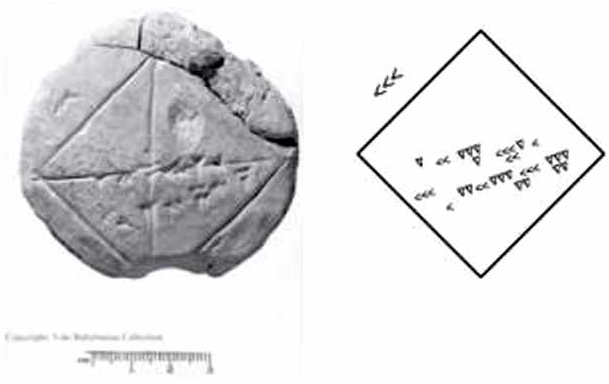
Aunque el punto inicial $x_0 = 0.1$ se encuentra cerca del punto fijo 0, la IPF se desplaza hacia el otro punto fijo, $x = 1.8$, y converge allí. La diferencia entre los dos puntos fijos es que el valor absoluto de la pendiente de g en $x = 1.8$, dada por $g'(1.8) = -0.8$, es menor que uno. Por otra parte, la pendiente de g en el otro punto fijo $x = 0$, del cual se alejan los puntos, es $g'(0) = 2.8$, cuyo valor absoluto es mayor que uno. ◄

El teorema de 1.6 es útil *a posteriori* (al final del cálculo de la IPF, se conoce la raíz y es posible calcular los errores paso por paso). El teorema ayuda a explicar por qué la razón de convergencia S resultó como lo hizo. Sería mucho más útil tener esa información antes de iniciarse el cálculo. En algunos casos, esto es posible como lo muestra el siguiente ejemplo.

► **EJEMPLO 1.6** Calcule $\sqrt{2}$ usando la IPF.

Un método antiguo para determinar raíces cuadradas puede expresarse como una IPF. Suponga que desean encontrarse los primeros 10 dígitos de $\sqrt{2}$. Comience con el valor inicial $x_0 = 1$. Obviamente, esta estimación es demasiado baja; por lo tanto, $2/1 = 2$ es demasiado alto. De hecho, cualquier estimación inicial $0 < x_0 < 2$ junto con $2/x_0$, forman un intervalo donde se encuentra $\sqrt{2}$. Por lo anterior, resulta razonable calcular el promedio de los dos valores para obtener una mejor estimación:

$$x_1 = \frac{1 + \frac{2}{1}}{2} = \frac{3}{2}.$$



(a) (b)

Figura 1.6 Cálculo antiguo de $\sqrt{2}$. (a) Placa YBC7289 (b) Esquema de la placa. Los babilonios calculaban en base 60, pero en ocasiones utilizaban la notación de base 10. El símbolo $<$ denota 10, y ∇ denota 1. En la parte superior izquierda se tiene 30, la longitud del lado. A lo largo de la línea media está 1, 24, 51 y 10, que representan la raíz cuadrada de 2 hasta cinco posiciones decimales correctas (vea la anotación al final de esta página). En la parte inferior, los números 42, 25 y 35 representan $30\sqrt{2}$ en base 60.

Ahora repita. Aunque 3/2 está más cerca, es demasiado grande para ser $\sqrt{2}$ y $2/(3/2) = 4/3$ es demasiado pequeño. Al igual que antes, se promedia para obtener

$$x_2 = \frac{\frac{3}{2} + \frac{4}{3}}{2} = \frac{17}{12} = 1.41\overline{6},$$

que está incluso más cerca de $\sqrt{2}$. Una vez más, x_2 y $2/x_2$ confinan a $\sqrt{2}$.

En el siguiente paso se obtiene

$$x_3 = \frac{\frac{17}{12} + \frac{24}{17}}{2} = \frac{577}{408} \approx 1.414215686.$$

Compruebe con una calculadora para ver que esta estimación concuerde con $\sqrt{2}$ con una precisión de 3×10^{-6}. La IPF que se está ejecutando es

$$x_{i+1} = \frac{x_i + \frac{2}{x_i}}{2}. \tag{1.14}$$

Observe que $\sqrt{2}$ es un punto fijo de la iteración.

ANOTACIÓN

Convergencia El método ingenioso del ejemplo 1.6 converge a $\sqrt{2}$ dentro de cinco posiciones decimales después de sólo tres pasos. Este método sencillo es uno de los más antiguos en la historia de las matemáticas. La placa cuneiforme YBC7289 que se muestra en la figura 1.6(a) se descubrió cerca de Bagdad en 1962 y data de alrededor de 1750 a.C. Contiene la aproximación (1) (24) (51) (10) en base 60 para la longitud del lado de un cuadrado de área 2. En base 10, esto es

$$1 + \frac{24}{60} + \frac{51}{60^2} + \frac{10}{60^3} = 1.41421296.$$

El método de cálculo de los babilonios no se conoce, pero se especula que usaron la misma técnica del ejemplo 1.6, en su base habitual 60. En cualquier caso, este método aparece en el libro 1 de *Métrica*, escrito por Herón de Alejandría en el siglo I d.C., para calcular $\sqrt{720}$.

Antes de terminar el cálculo, se decidirá si éste va a converger. De acuerdo con el teorema 1.6, se tiene que $S < 1$. Para esta iteración, $g(x) = 1/2(x + 2/x)$ y $g'(x) = 1/2(1 - 2/x^2)$. Al evaluar en el punto fijo resulta

$$g'(\sqrt{2}) = \frac{1}{2}\left(1 - \frac{2}{(\sqrt{2})^2}\right) = 0, \tag{1.15}$$

de manera que $S = 0$. Se concluye que la IPF convergerá, y muy rápido.

En el ejercicio 18 se pregunta si este método logrará encontrar la raíz cuadrada de un número positivo arbitrario. ◀

1.2.4 Criterios de detención

A diferencia del caso de la bisección, el número de pasos necesarios para que la IPF converja dentro de una tolerancia dada casi nunca es predecible de antemano. En la ausencia de una fórmula de error como (1.1) para el método de bisección, debe tomarse una decisión sobre la terminación del algoritmo, llamado **criterio de detención.**

Para una tolerancia establecida, TOL, puede pedirse un criterio de detención basado en el error absoluto

$$|x_{i+1} - x_i| < \text{TOL} \tag{1.16}$$

o, en caso de que la solución no esté muy cerca de cero, un criterio de detención basado en el error relativo

$$\frac{|x_{i+1} - x_i|}{|x_{i+1}|} < \text{TOL}. \tag{1.17}$$

Con frecuencia resulta útil un criterio de detención híbrido absoluto y relativo, como

$$\frac{|x_{i+1} - x_i|}{\text{máx}(|x_{i+1}|, \theta)} < \text{TOL} \tag{1.18}$$

para algún $\theta > 0$ es útil en casos donde la solución es cercana a 0. Además, un buen código para la IPF establece un límite en el número máximo de pasos, en caso de que la convergencia falle. La cuestión de los criterios de detención es importante y se repasará en una forma más compleja cuando se analicen los errores hacia adelante y hacia atrás en la sección 1.3.

El método de bisección garantiza la convergencia lineal. La iteración de punto fijo sólo es convergente a nivel local, y cuando converge lo hace linealmente. Ambos métodos requieren una evaluación de la función en cada paso. La bisección reduce la incertidumbre en 1/2 en cada paso, mientras que la IPF lo hace aproximadamente en $S = |g'(r)|$. Por lo tanto, la iteración de punto fijo puede ser más rápida o más lenta que la bisección, dependiendo de si S es menor o mayor que 1/2. En la sección 1.4 se estudiará el método de Newton, una versión bastante mejorada de la IPF, donde S está diseñada para ser cero.

1.2 Ejercicios

1. Encuentre todos los puntos fijos de la siguientes $g(x)$.
   (a) $\frac{3}{x}$ (b) $x^2 - 2x + 2$ (c) $x^2 - 4x + 2$
2. Encuentre todos los puntos fijos de la siguiente $g(x)$.
   (a) $\frac{x+6}{3x-2}$ (b) $\frac{8+2x}{2+x^2}$ (c) x^5
3. Muestre que 1, 2 y 3 son puntos fijos de la siguiente $g(x)$.
   (a) $\frac{x^3 + x - 6}{6x - 10}$ (b) $\frac{6 + 6x^2 - x^3}{11}$

4. Muestre que -1, 0 y 1 son puntos fijos de la siguiente $g(x)$.

   (a) $\dfrac{4x}{x^2+3}$ (b) $\dfrac{x^2-5x}{x^2+x-6}$

5. ¿Para cuál de las siguientes $g(x)$ $r=\sqrt{3}$ es un punto fijo?

   (a) $g(x)=\dfrac{x}{\sqrt{3}}$ (b) $g(x)=\dfrac{2x}{3}+\dfrac{1}{x}$ (c) $g(x)=x^2-x$ (d) $g(x)=1+\dfrac{2}{x+1}$

6. ¿Para cuál de las siguientes $g(x)$ $r=\sqrt{5}$ es un punto fijo?

   (a) $g(x)=\dfrac{5+7x}{x+7}$ (b) $g(x)=\dfrac{10}{3x}+\dfrac{x}{3}$ (c) $g(x)=x^2-5$ (d) $g(x)=1+\dfrac{4}{x+1}$

7. Use el teorema 1.6 para determinar si la iteración de punto fijo de $g(x)$ converge localmente al punto fijo dado r. (a) $g(x)=(2x-1)^{1/3}$, $r=1$ (b) $g(x)=(x^3+1)/2$, $r=1$ (c) $g(x)=\operatorname{sen} x+x$, $r=0$

8. Use el teorema 1.6 para determinar si la iteración de punto fijo de $g(x)$ converge localmente al punto fijo dado r. (a) $g(x)=(2x-1)/x^2$, $r=1$ (b) $g(x)=\cos x+\pi+1$, $r=\pi$ (c) $g(x)=e^{2x}-1$, $r=0$

9. Encuentre cada punto fijo y decida si la iteración de punto fijo converge localmente a éste.

   (a) $g(x)=\frac{1}{2}x^2+\frac{1}{2}x$ (b) $g(x)=x^2-\frac{1}{4}x+\frac{3}{8}$

10. Encuentre cada punto fijo y decida si la iteración de punto fijo converge localmente a éste.

    (a) $g(x)=x^2-\frac{3}{2}x+\frac{3}{2}$ (b) $g(x)=x^2+\frac{1}{2}x-\frac{1}{2}$

11. Exprese cada ecuación como un problema de punto fijo $x=g(x)$ en tres formas diferentes.

    (a) $x^3-x+e^x=0$ (b) $3x^{-2}+9x^3=x^2$

12. Considere la iteración de punto fijo $x\to g(x)=x^2-0.24$. (a) ¿Espera que la iteración de punto fijo calcule la raíz -0.2, digamos, hasta 10 decimales correctos, más rápido o más lento que el método de bisección? (b) Encuentre el punto fijo. ¿La IPF convergerá a éste?

13. (a) Encuentre todos los puntos fijos de $g(x)=0.39-x^2$. (b) ¿A cuál de los puntos fijos converge la iteración de punto fijo a nivel local? (c) ¿La IPF converge en este punto fijo más rápido o más lento que el método de bisección?

14. ¿Cuál de las siguientes tres iteraciones de punto fijo convergen a $\sqrt{2}$? Clasifique las iteraciones convergentes de la más rápida a la más lenta.

    (a) $x\longrightarrow \dfrac{1}{2}x+\dfrac{1}{x}$ (b) $x\longrightarrow \dfrac{2}{3}x+\dfrac{2}{3x}$ (c) $x\longrightarrow \dfrac{3}{4}x+\dfrac{1}{2x}$

15. ¿Cuál de las siguientes tres iteraciones de punto fijo convergen a $\sqrt{5}$? Clasifique las iteraciones convergentes de la más rápida a la más lenta.

    (a) $x\longrightarrow \dfrac{4}{5}x+\dfrac{1}{x}$ (b) $x\longrightarrow \dfrac{x}{2}+\dfrac{5}{2x}$ (c) $x\longrightarrow \dfrac{x+5}{x+1}$

16. ¿Cuál de las siguientes tres iteraciones de punto fijo convergen a la raíz cúbica de 4? Clasifique las iteraciones convergentes de la más rápida a la más lenta.

    (a) $g(x)=\dfrac{2}{\sqrt{x}}$ (b) $g(x)=\dfrac{3x}{4}+\dfrac{1}{x^2}$ (c) $g(x)=\dfrac{2}{3}x+\dfrac{4}{3x^2}$

17. Verifique si $1/2$ y -1 son raíces de $f(x)=2x^2+x-1=0$. Aísle el término x^2 y despeje x a fin de encontrar dos candidatos para $g(x)$. ¿Cuál de las raíces se encontrará mediante las dos iteraciones de punto fijo?

18. Demuestre que el método del ejemplo 1.6 calculará la raíz cuadrada de cualquier número positivo.

19. Explore la idea del ejemplo 1.6 para las raíces cúbicas. Si x es una estimación menor que $A^{1/3}$, entonces A/x^2 será más grande que $A^{1/3}$, de modo que el promedio de los dos será una mejor aproximación que x. Sugiera una iteración de punto fijo sobre la base de este hecho y utilice el teorema 1.6 para decidir si convergerá a la raíz cúbica de A.

20. Mejore el algoritmo de la raíz cúbica presentado en el ejercicio 19 al cambiar las constantes del promedio. Si se establece $g(x) = \omega x + (1 - \omega) A/x^2$ para algún número fijo $0 < \omega < 1$, ¿cuál es la mejor opción para ω?

21. Considere la iteración de punto fijo aplicado a $g(x) = 1 - 5x + \frac{15}{2}x^2 - \frac{5}{2}x^3$. (a) Muestre que $1 - \sqrt{3/5}$, 1, y $1 + \sqrt{3/5}$ son puntos fijos. (b) Muestre que ninguno de los tres puntos fijos es localmente convergente. (El problema de computadora 7 aborda este ejemplo más a fondo).

22. Demuestre que las estimaciones iniciales 0, 1 y 2 conducen a un punto fijo en el ejercicio 21. ¿Qué pasa con otras estimaciones iniciales cercanas a esos números?

23. Suponga que $g(x)$ es continuamente diferenciable y que la iteración de punto fijo $g(x)$ tiene con exactitud tres puntos fijos, $r_1 < r_2 < r_3$. Suponga también que $|g'(r_1)| = 0.5$ y $|g'(r_3)| = 0.5$. Presente todos los valores de $|g'(r_2)|$ que son posibles en estas condiciones.

24. Suponga que g es una función continuamente diferenciable y que la iteración de punto fijo $g(x)$ tiene con exactitud tres puntos fijos, -3, 1 y 2. También asuma que $g'(-3) = 2.4$ y que si la IPF inicia suficientemente cerca del punto fijo 2 converge a este valor. Encuentre $g'(1)$.

25. Demuestre la variante del teorema 1.6: Si g es continuamente diferenciable y $|g'(x)| \leq B < 1$ en un intervalo $[a, b]$ que contiene el punto fijo r, entonces la IPF converge a r a partir de cualquier estimación inicial en $[a, b]$.

26. Demuestre que una función continuamente diferenciable $g(x)$ que satisface $|g'(x)| < 1$ en un intervalo cerrado no puede tener dos puntos fijos en ese intervalo.

27. Considere la iteración de punto fijo con $g(x) = x - x^3$. (a) Muestre que $x = 0$ es el único punto fijo. (b) Muestre que si $0 < x_0 < 1$, entonces $x_0 > x_1 > x_2 \ldots > 0$. (c) Muestre que la IPF converge a $r = 0$, mientras que $g'(0) = 1$. (Sugerencia: utilice el hecho de que toda sucesión monótona acotada converge a un límite).

28. Considere la iteración de punto fijo con $g(x) = x + x^3$. (a) Muestre que $x = 0$ es el único punto fijo. (b) Muestre que si $0 < x_0 < 1$, entonces $x_0 < x_1 < x_2 < \ldots$ (C) Muestre que la IPF no converge a un punto fijo, mientras que $g'(0) = 1$. Junto con el ejercicio 27, esto demuestra que la IPF puede converger a un punto fijo r o divergir de r cuando $|g'(r)| = 1$.

29. Considere la ecuación $x^3 + x - 2 = 0$, con raíz $r = 1$. Sume el término cx a ambos lados y divida entre c para obtener $g(x)$. (a) ¿Para cuál c la IPF es localmente convergente a $r = 1$? (b) ¿Para cuál c la IPF converge más rápido?

30. Suponga que la iteración de punto fijo se aplica a una función $g(x)$ que es dos veces continuamente diferenciable y que $g'(r) = 0$ para un punto fijo r. Muestre que si la IPF converge a r, entonces el error obedece $\lim_{i\to\infty}(e_{i+1})/e_i^2 = M$, donde $M = |g''(r)|/2$.

31. Defina la iteración de punto fijo sobre la ecuación $x^2 + x = 5/16$ al aislar el término x. Encuentre los dos puntos fijos y determine cuáles estimaciones iniciales conducen a cada punto fijo bajo iteración. (Sugerencia: grafique $g(x)$ y dibuje diagramas de telaraña).

32. Encuentre el conjunto de todas las estimaciones iniciales para las cuales la iteración de punto fijo $x \to 4/9 - x^2$ converge a un punto fijo.

33. Sea $g(x) = a + bx + cx^2$ para las constantes a, b y c. (a) Especifique un conjunto de constantes a, b y c para los cuales $x = 0$ es un punto fijo de $x = g(x)$, y la iteración de punto fijo converge localmente a 0. (b) Especifique un conjunto de constantes a, b y c para los cuales $x = 0$ es un punto fijo de $x = g(x)$, pero la iteración de punto fijo no converge localmente a 0.

1.2 Problemas de computadora

1. Aplique la iteración de punto fijo para encontrar la solución de cada ecuación hasta ocho posiciones decimales correctas. (a) $x^3 = 2x + 2$ (b) $e^x + x = 7$ (c) $e^x + \operatorname{sen} x = 4$

2. Aplique la iteración de punto fijo para encontrar la solución de cada ecuación hasta ocho posiciones decimales correctas. (a) $x^5 + x = 1$ (b) $\operatorname{sen} x = 6x + 5$ (c) $\ln x + x^2 = 3$

3. Calcule las raíces cuadradas de los números siguientes hasta ocho cifras decimales correctas usando la iteración de punto fijo como en el ejemplo 1.6: (a) 3 (b) 5. Establezca su estimación inicial y el número de pasos necesarios.

4. Calcule las raíces cúbicas de los números siguientes hasta ocho posiciones decimales correctas usando la iteración de punto fijo con $g(x) = (2x + A/x^2)/3$, donde A es (a) 2 (b) 3 (c) 5. Establezca su estimación inicial y el número de pasos necesarios.

5. En el ejemplo 1.3 se muestra que $g(x) = \cos x$ es una IPF convergente, ¿Pasa lo mismo para $g(x) = \cos^2 x$? Encuentre el punto fijo hasta seis decimales correctos e informe el número de pasos necesarios de la IPF. Analice la convergencia local utilizando el teorema 1.6.

6. Deduzca tres diferentes $g(x)$ para encontrar las raíces hasta seis cifras decimales correctas de las siguientes $f(x) = 0$ mediante la iteración de punto fijo. Ejecute la IPF para cada $g(x)$, reporte los resultados y si hay convergencia o divergencia. Cada ecuación $f(x) = 0$ tiene tres raíces. Si es necesario deduzca más $g(x)$ hasta encontrar todas las raíces mediante la IPF. Para cada serie convergente, determine el valor de S a partir de los errores e_{i+1}/e_i y compárelo con la S determinada a partir de cálculos como los de (1.11). (a) $f(x) = 2x^3 - 6x - 1$ (b) $f(x) = e^{x-2} + x^3 - x$ (c) $f(x) = 1 + 5x - 6x^3 - e^{2x}$

7. En el ejercicio 21 se consideró la iteración de punto fijo aplicada a $g(x) = 1 - 5x + \frac{15}{2}x^2 - \frac{5}{2}x^3 = x$. Encuentre estimaciones iniciales para las cuales la IPF (a) entre en un ciclo sin fin en números dentro del intervalo (0, 1), (b) haga lo mismo que en (a), pero dentro del intervalo (1, 2) y (c) diverja al infinito. Los casos (a) y (b) son ejemplos de una dinámica caótica. En los tres casos, la IPF no tiene éxito.

1.3 LÍMITES DE EXACTITUD

Uno de los objetivos del análisis numérico es calcular las respuestas dentro de un nivel específico de precisión. El trabajo en precisión doble significa que se almacenan y operan números que se mantienen a 52-bits de precisión, aproximadamente 16 dígitos decimales.

¿Es posible calcular siempre las respuestas hasta 16 dígitos significativos correctos? En el capítulo 0 se demostró que, con un algoritmo sencillo para calcular las raíces de una ecuación cuadrática, es posible perder algunos o todos los dígitos significativos. Un algoritmo mejorado elimina el problema. En esta sección, se estudiará algo nuevo: un cálculo que una computadora de precisión doble no puede realizar siquiera cerca de 16 dígitos correctos, incluso usando el mejor algoritmo.

1.3.1 Error hacia adelante y hacia atrás

En el primer ejemplo se muestra que, en algunos casos, el lápiz y el papel aún pueden superar a una computadora.

► EJEMPLO 1.7 Use el método de bisección para encontrar la raíz de $f(x) = x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ hasta seis cifras significativas correctas.

Tenga en cuenta que $f(0)\,f(1) = (-8/27)(1/27) < 0$, por lo que el teorema del valor intermedio garantiza una solución en [0, 1]. De acuerdo con el ejemplo 1.2, bastarían 20 pasos de bisección para obtener seis posiciones correctas.

De hecho, resulta fácil comprobar sin una computadora que $r = 2/3 = 0.666666666\ldots$ es una raíz:

$$f(2/3) = \frac{8}{27} - 2\left(\frac{4}{9}\right) + \left(\frac{4}{3}\right)\left(\frac{2}{3}\right) - \frac{8}{27} = 0.$$

¿Cuántas de estas cifras pueden obtenerse mediante el método de bisección?

i	a_i	$f(a_i)$	c_i	$f(c_i)$	b_i	$f(b_i)$
0	0.0000000	−	0.5000000	−	1.0000000	+
1	0.5000000	−	0.7500000	+	1.0000000	+
2	0.5000000	−	0.6250000	−	0.7500000	+
3	0.6250000	−	0.6875000	+	0.7500000	+
4	0.6250000	−	0.6562500	−	0.6875000	+
5	0.6562500	−	0.6718750	+	0.6875000	+
6	0.6562500	−	0.6640625	−	0.6718750	+
7	0.6640625	−	0.6679688	+	0.6718750	+
8	0.6640625	−	0.6660156	−	0.6679688	+
9	0.6660156	−	0.6669922	+	0.6679688	+
10	0.6660156	−	0.6665039	−	0.6669922	+
11	0.6665039	−	0.6667480	+	0.6669922	+
12	0.6665039	−	0.6666260	−	0.6667480	+
13	0.6666260	−	0.6666870	+	0.6667480	+
14	0.6666260	−	0.6666565	−	0.666687	+
15	0.6666565	−	0.6666718	+	0.6666870	+
16	0.6666565	−	0.6666641	0	0.6666718	+

Es sorprendente que el método de bisección se detenga después de 16 pasos, cuando se calcula $f(0.6666641) = 0$. Esto es una falla grave si se requieren seis o más dígitos de precisión. En la figura 1.7 se muestra la dificultad. En cuanto a la precisión doble del IEEE, hay muchos números de punto flotante a 10^{-5} de la raíz correcta $r = 2/3$ que se evalúan como el cero de máquina, y por lo tanto ¡tienen el mismo derecho a ser llamados raíz! Por si fuera poco, a pesar de que la función f es monótona creciente, el inciso (b) de la figura muestra que incluso el signo del valor de precisión doble de f con frecuencia es erróneo.

En la figura 1.7 se muestra que el problema no reside en el método de bisección, sino en la incapacidad de la aritmética de precisión doble para calcular la función f con precisión suficiente cerca de la raíz. Cualquier otro método de solución que se base en esta aritmética de máquina está destinado al fracaso. Para este ejemplo, los 16 dígitos de precisión ni siquiera pueden comprobar si una solución candidata tiene seis posiciones correctas. ◄

Para convencerle de que esto no es culpa del método de bisección, se aplica el localizador de raíces de propósito múltiple más poderoso de MATLAB, `fzero.m`. Los detalles se analizarán más

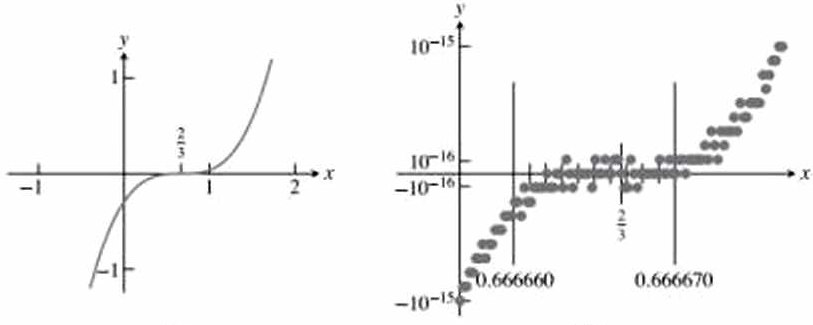


Figura 1.7 La forma de una función cerca de una raíz múltiple. (a) Gráfica de $f(x) = x^3 - 2x^2 + 4/3x - 8/27$. (b) Magnificación de (a), cerca de la raíz $r = 2/3$. En lo que a la computadora se refiere, hay muchos números de punto flotante a 10^{-5} de 2/3 que son raíces. Con base en el cálculo se sabe que 2/3 es la única raíz.

adelante en este capítulo; por ahora, sólo es necesario alimentarlo con la función y una estimación inicial. No tiene mayor suerte:

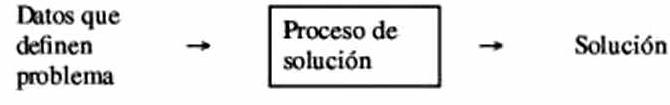
```
>> fzero('x.^3-2*x.^2+4*x/3-8/27',1)

ans =

   0.66666250845989
```

La razón por la que todos los métodos fallan en la búsqueda de más de cinco dígitos correctos para este ejemplo se ve claramente en la figura 1.7. La única información que tiene cualquier método es la función, calculada en precisión doble. Si la aritmética de computadora está mostrando que la función es igual a cero en un valor que no es una raíz, no hay manera de que el método puede recuperarse. Otra forma de expresar esta dificultad consiste en decir que una solución aproximada puede estar tan cerca como sea posible a una solución en lo que al eje y se refiere, pero no tan cerca en el eje x.

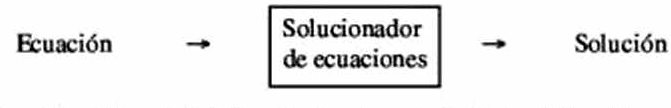
Estas observaciones motivan algunas definiciones clave.

DEFINICIÓN 1.8 Suponga que f es una función y que r es una raíz, lo que significa que satisface $f(r) = 0$. También asuma que x_a es una aproximación a r. Para el problema de localización de una raíz, el **error hacia atrás** de la aproximación x_a es $|f(x_a)|$, y el **error hacia adelante** es $|r - x_a|$. ❐

El uso de "hacia atrás" y "hacia adelante" puede necesitar cierta explicación. Se considera que el proceso de encontrar una solución es esencial. El problema es la entrada y la solución es la salida:



En este capítulo, el "problema" es una ecuación de una variable y el "proceso de solución" es un algoritmo que resuelve ecuaciones:



El error hacia atrás está en el lado izquierdo o la entrada (datos del problema). Es la cantidad que tendría que cambiar el problema (la función f) para hacer que la ecuación se equilibre con la

aproximación x_a de salida. Esta cantidad es $|f(x_a)|$. El error hacia delante se encuentra en el lado derecho o la salida (solución del problema). Es la cantidad que tendría que cambiar la solución aproximada para que sea correcta, esto es $|r - x_a|$.

La dificultad con el ejemplo 1.7 es que, según la figura 1.7, el error hacia atrás es cercano a $\epsilon_{máq} \approx 2.2 \times 10^{-16}$, mientras que el error hacia adelante es de aproximadamente 10^{-5}. Los números de precisión doble no pueden calcularse de manera confiable por debajo de un error relativo del orden del épsilon máquina. Como el error hacia atrás no puede disminuirse de manera confiable, tampoco es posible hacerlo con el error hacia adelante.

El ejemplo 1.7 es bastante especial puesto que la función tiene una raíz triple en $r = 2/3$. Observe que

$$f(x) = x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27} = \left(x - \frac{2}{3}\right)^3.$$

Éste es un ejemplo de una raíz múltiple.

DEFINICIÓN 1.9 Suponga que r es una raíz de la función diferenciable f; es decir, asuma que $f(r) = 0$. Entonces Si $0 = f(r) = f'(r) = f''(r) = \ldots = f^{(m-1)}(r)$, pero $f^{(m)}(r) \neq 0$, se dice que f que tiene una **raíz** de **multiplicidad** m en r. Se dice que f tiene una **raíz múltiple** en r si la multiplicidad es mayor que uno. La raíz se denomina **simple** si la multiplicidad es igual a uno. ❐

Por ejemplo, $f(x) = x^2$ tiene una multiplicidad de dos, o una raíz doble, en $r = 0$, puesto que $f(0) = 0, f'(0) = 2(0) = 0$, pero $f''(0) = 2 \neq 0$. Asimismo, $f(x) = x^3$ tiene una multiplicidad de tres, o una raíz triple, en $r = 0$ y $f(x) = x^m$ tiene una multiplicidad m de la raíz en ese punto. El ejemplo 1.7 tiene una multiplicidad de tres, o una raíz triple, en $r = 2/3$.

Debido a que la gráfica de la función es relativamente plana cerca de una raíz múltiple, existe una gran disparidad entre los errores hacia atrás y hacia adelante para las soluciones aproximadas cercanas. El error hacia atrás, medido en la dirección vertical, suele ser mucho menor que el error hacia delante, medido en la dirección horizontal.

► EJEMPLO 1.8 La función $f(x) = \operatorname{sen} x - x$ tiene una raíz triple en $r = 0$. Encuentre el error hacia adelante y hacia atrás de la raíz aproximada $x_c = 0.001$.

La raíz en 0 tiene una multiplicidad de tres, porque

$$\begin{aligned} f(0) &= \operatorname{sen} 0 - 0 = 0 \\ f'(0) &= \cos 0 - 1 = 0 \\ f''(0) &= -\operatorname{sen} 0 - 0 = 0 \\ f'''(0) &= -\cos 0 = -1. \end{aligned}$$

El error hacia adelante es $ED = |r - x_a| = 10^{-3}$. El error hacia atrás es la constante que tendría que añadirse a $f(x)$ para hacer de x_a una raíz, es decir, $EA = |f(x_a)| = |\operatorname{sen}(0.001) - 0.001| \approx 1.6667 \times 10^{-10}$. ◄

El tema del error hacia atrás y hacia adelante es importante para los criterios de detención en los solucionadores de ecuaciones. El objetivo es encontrar la raíz r que satisface $f(r) = 0$. Suponga que el algoritmo usado produce una solución aproximada x_a. ¿Cómo puede decidirse si es suficientemente buena?

Existen dos posibilidades que vienen a la mente: (1) hacer pequeña a $|x_a - r|$ y (2) hacer pequeña a $|f(x_a)|$. En caso de que $x_a = r$, no hay ninguna decisión que tomar (en ambos sentidos las opciones son iguales). Sin embargo, esta situación se presenta en muy pocas ocasiones. En el caso más típico, los enfoques (1) y (2) son diferentes y corresponden al error hacia adelante y hacia atrás.

La conveniencia del error hacia adelante o hacia atrás depende de las circunstancias que rodean al problema. Si se usa el método de bisección, ambos errores se aprecian con facilidad. Para una raíz aproximada x_a, puede encontrarse el error hacia atrás mediante la evaluación $f(x_a)$ y el

error hacia adelante no puede ser más de la mitad de la longitud del intervalo actual. Para la IPF, las opciones son más limitadas, puesto que no se tiene intervalo de confinamiento. Como antes, el error hacia atrás se conoce con $f(x_a)$, pero para saber cuál es el error hacia adelante sería necesario conocer la raíz verdadera, que es lo que se está tratando de encontrar.

Los criterios de detención para los métodos de resolución de ecuaciones pueden basarse en el error hacia adelante o en el error hacia atrás. Existen otros criterios de detención que pueden ser relevantes, como un límite en el tiempo de cálculo. La elección del criterio debe estar guiada por el contexto del problema.

Las funciones son planas en la vecindad de una raíz múltiple, puesto que en ese punto la derivada f' es igual a cero. Debido a lo anterior, pueden esperarse algunos problemas al tratar de aislar una raíz múltiple, como ya se ha mostrado. Pero la multiplicidad sólo es la punta del iceberg; existen dificultades similares que pueden surgir incluso cuando no hay raíces múltiples a la vista, tal como se muestra en la siguiente sección.

1.3.2 El polinomio de Wilkinson

En Wilkinson [1994] se analiza un ejemplo famoso con raíces simples, que son difíciles de determinar de manera numérica. El **polinomio de Wilkinson** es

$$W(x) = (x-1)(x-2)\ldots(x-20) \tag{1.19}$$

que, cuando se multiplica resulta en

$$\begin{aligned} W(x) = x^{20} &- 210x^{19} + 20615x^{18} - 1256850x^{17} + 53327946x^{16} - 1672280820x^{15} \\ &+ 40171771630x^{14} - 756111184500x^{13} + 11310276995381x^{12} \\ &- 135585182899530x^{11} + 1307535010540395x^{10} - 10142299865511450x^{9} \\ &+ 63030812099294896x^{8} - 311333643161390640x^{7} \\ &+ 1206647803780373360x^{6} - 3599979517947607200x^{5} \\ &+ 8037811822645051776x^{4} - 12870931245150988800x^{3} \\ &+ 13803759753640704000x^{2} - 8752948036761600000x \\ &+ 2432902008176640000. \end{aligned} \tag{1.20}$$

Las raíces son los números enteros del 1 al 20. Sin embargo, cuando se define $W(x)$ de acuerdo con su forma no factorizada (1.20), su evaluación experimenta el problema de la cancelación de números casi iguales y muy grandes. Para ver el efecto en la localización de raíces, defina el archivo m, `wilkpoly.m` en MATLAB escribiendo el polinomio en forma no factorizada (1.20), u obteniéndolo de la página web del libro de texto.

De nuevo se probará con `fzero` de MATLAB. Para que sea lo más fácil posible, se alimentará una raíz real $x = 16$ como estimación inicial:

```
>> fzero(@wilkpoly,16)

ans =

  16.01468030580458
```

El sorprendente resultado es que la aritmética de precisión doble de MATLAB no pudo conseguir el segundo decimal correcto, ni siquiera para la raíz simple $r = 16$. Esto no se debe a una deficiencia del algoritmo, tanto `fzero` como el método de bisección tienen el mismo problema, así como la iteración de punto fijo y cualquier otro método de punto flotante. Al referirse a su trabajo con este polinomio, Wilkinson escribió en 1984: "Por mi parte considero que es la experiencia más traumática de mi carrera como analista numérico". Las raíces de $W(x)$ son claras: los enteros $x = 1,\ldots, 20$. Para Wilkinson la sorpresa reside en la enorme magnificación del error en las raíces, causada por pequeños errores relativos al almacenar los coeficientes, lo cual acaba de verse en acción.

La dificultad de obtener raíces exactas del polinomio de Wilkinson desaparece cuando se usa su forma factorizada (1.19) en vez de (1.20). Por supuesto, si el polinomio se factoriza antes de empezar, no hay necesidad de calcular raíces.

1.3.3 Sensibilidad de la localización de raíces

El polinomio de Wilkinson y el ejemplo 1.7 con raíz triple ocasionan dificultades por razones similares (los pequeños errores de punto flotante en la ecuación se traducen en grandes errores en la raíz). Un problema se denomina **sensible** si los pequeños errores en la entrada, en este caso la ecuación que debe resolverse, dan lugar a grandes errores en la salida, o solución. En esta sección, se cuantificará la sensibilidad y se presentarán los conceptos del factor de magnificación del error y número de condición.

Para entender qué ocasiona esta magnificación de error, se establecerá una fórmula para predecir cuán lejos se mueve una raíz cuando la ecuación cambia. Suponga que el problema es encontrar una raíz r de $f(x) = 0$, pero que se hace un ligero cambio a la entrada $\epsilon\, g(x)$, donde ϵ es pequeño. Sea Δr el cambio correspondiente en la raíz, de modo que

$$f(r + \Delta r) + \epsilon g(r + \Delta r) = 0.$$

La expansión de f y g en polinomios de Taylor de grado 1 implica que

$$f(r) + (\Delta r) f'(r) + \epsilon g(r) + \epsilon(\Delta r) g'(r) + O((\Delta r)^2) = 0,$$

donde se utiliza la notación de la "gran O" $O((\Delta r)^2)$ para representar los términos que incluyen $(\Delta r)^2$ y las potencias superiores de Δr. Para Δr pequeños, los términos de $O((\Delta r)^2)$ pueden despreciarse para obtener

$$(\Delta r)(f'(r) + \epsilon g'(r)) \approx -f(r) - \epsilon g(r) = -\epsilon g(r)$$

o

$$\Delta r \approx \frac{-\epsilon g(r)}{f'(r) + \epsilon g'(r)} \approx -\epsilon \frac{g(r)}{f'(r)},$$

suponiendo que ϵ es pequeña en comparación con $f'(r)$ y, en particular, que, $f'(r) \neq 0$.

Fórmula de sensibilidad para las raíces

Suponga que r es una raíz de $f(x)$ y $r + \Delta r$ es una raíz de $f(x) + \epsilon\, g(x)$. Entonces

$$\Delta r \approx -\frac{\epsilon g(r)}{f'(r)} \tag{1.21}$$

si $\epsilon \ll f'(r)$.

► EJEMPLO 1.9 Estime la mayor raíz de $P(x) = (x-1)(x-2)(x-3)(x-4)(x-5)(x-6) - 10^{-6}x^7$.

Sea $f(x) = (x-1)(x-2)(x-3)(x-4)(x-5)(x-6)$, $\epsilon = -10^{-6}$ y $g(x) = x^7$. Sin el término $\epsilon\, g(x)$, la mayor raíz es $r = 6$. La pregunta es, ¿hasta dónde se traslada la raíz cuando se agrega el término extra?

Con la fórmula de sensibilidad se obtiene

$$\Delta r \approx -\frac{\epsilon 6^7}{5!} = -2332.8\epsilon,$$

lo que significa que los errores de entrada de tamaño relativo ϵ en $f(x)$ se magnifican por un factor de más de 2000 en la raíz de salida. Se estima que la mayor raíz de $P(x)$ es $r + \Delta r = 6 - 2332.8\epsilon = 6.0023328$. Si se utiliza `fzero` en $P(x)$, se obtiene el valor correcto 6.0023268. ◄

La estimación en el ejemplo 1.9 es suficientemente buena para conocer cómo se propagan los errores en el problema de localización de raíces. Un error en el sexto dígito de los datos del problema provoca un error en el tercer dígito de la respuesta, lo que significa que se pierden tres dígitos decimales debido al factor de 2332.8. Resulta útil disponer de un nombre para este factor. Para un algoritmo general que produce una aproximación x_c, se define su

$$\textbf{factor de magnificación del error} = \frac{\text{error relativo hacia delante}}{\text{error relativo hacia atrás}}$$

El error hacia adelante es el cambio en la solución que haría que x_a fuera correcta, lo que en los problemas de localización de raíces es $|x_a - r|$. El error hacia atrás es un cambio en la entrada que hace que x_c sea la solución correcta. Existe una variedad más amplia de opciones, dependiendo de la sensibilidad que se desee investigar. La selección que se utilizó antes en esta sección fue la de cambiar el término constante por $|f(x_a)|$, correspondiente a $g(x) = 1$ en la fórmula de sensibilidad (1.21). De manera más general, cualquier cambio en los datos de entrada puede utilizarse como el error hacia atrás, como la elección de $g(x) = x^7$ en el ejemplo 1.9. El factor de magnificación del error en la determinación de raíces es

$$\text{factor de magnificación del error} = \left|\frac{\Delta r/r}{\epsilon g(r)/g(r)}\right| = \left|\frac{-\epsilon g(r)/(rf'(r))}{\epsilon}\right| = \frac{|g(r)|}{|rf'(r)|}, \quad (1.22)$$

que en el ejemplo 1.9 es $6^7/(5!6) = 388.8$.

► **EJEMPLO 1.10** Use la fórmula de sensibilidad para las raíces e investigue el efecto de los cambios en el término x^{15} del polinomio de Wilkinson sobre la raíz $r = 16$. Encuentre el factor de magnificación del error para este problema.

Defina la función perturbada $W_\epsilon(x) = W(x) + \epsilon g(x)$, donde $g(x) = -1.672{,}280{,}820x^{15}$. Observe que $W'(16) = 15!\,4!$ (vea el ejercicio 7). Si se usa (1.21), el cambio en la raíz puede aproximarse por

$$\Delta r \approx \frac{16^{15}1{,}672{,}280{,}820\epsilon}{15!4!} \approx 6.1432 \times 10^{13}\epsilon. \quad (1.23)$$

Hablando de manera práctica, en el capítulo 0 se vio que es necesario asumir un error relativo del orden de épsilon máquina para cada número almacenado. Un cambio relativo en el término x^{15} de épsilon máquina ϵ_{maq} hará que la raíz $r = 16$ se desplace en

$$\Delta r \approx (6.1432 \times 10^{13})(\pm 2.22 \times 10^{-16}) \approx \pm 0.0136$$

hasta $r + \Delta r \approx 16.0136$, no muy lejos de lo que se observó en la página 47. Por supuesto, muchas otras potencias de x en el polinomio Wilkinson están haciendo sus propias contribuciones, por lo que la imagen completa es complicada. Sin embargo, la fórmula de sensibilidad permite ver el mecanismo para la enorme magnificación del error.

Por último, el factor de magnificación del error se calcula a partir de (1.22) como

$$\frac{|g(r)|}{|rf'(r)|} = \frac{16^{15}1{,}672{,}280{,}820}{15!4!16} \approx 3.8 \times 10^{12}. \quad \blacktriangleleft$$

La importancia del factor de magnificación del error es que indica cuántos de los 16 dígitos de precisión operativa se pierden desde la entrada hasta la salida. Para un problema con un factor de magnificación del error de 10^{12}, se espera perder 12 de los 16 y conservar alrededor de cuatro dígitos significativos correctos en la raíz, que es el caso para la aproximación de Wilkinson $x_c = 16.014...$.

ANOTACIÓN

Condicionamiento Ésta es la primera aparición del concepto de número de condición, una medida de la magnificación del error. El análisis numérico es el estudio de algoritmos, los cuales toman como entrada datos que definen al problema y entregan una respuesta como salida. El número de condición se refiere a la parte de esta magnificación que es inherente al problema teórico en sí, independientemente del algoritmo particular usado para resolverlo.

Es importante observar que el factor de magnificación del error mide sólo el aumento debido al problema. Junto con el condicionamiento hay un concepto paralelo, la estabilidad, que se refiere a la magnificación de los pequeños errores de entrada debida al algoritmo, no al problema en sí. Un algoritmo se denomina estable si siempre proporciona una solución aproximada con un pequeño error hacia atrás. Si el problema está bien condicionado y el algoritmo es estable, pueden esperarse errores pequeños tanto hacia atrás como hacia adelante.

Los ejemplos anteriores de magnificación del error muestran la sensibilidad de la localización de raíces a una entrada en particular. El problema puede ser más o menos sensible, dependiendo de cómo se diseñe el cambio de entrada. El **número de condición** de un problema se define como la máxima magnificación del error debida a todos los posibles cambios en la entrada, o por lo menos a todos los cambios de un tipo preestablecido. Un problema con número de condición alto se llama **mal condicionado** y un problema con un número de condición cercano a 1 se llama **bien condicionado**. Este concepto se abordará de nuevo cuando se estudien los problemas matriciales en el capítulo 2.

1.3 Ejercicios

1. Encuentre el error hacia adelante y hacia atrás para las siguientes funciones, donde la raíz es 3/4 y la raíz aproximada es $x_a = 0.74$: (a) $f(x) = 4x - 3$ (b) $f(x) = (4x - 3)^2$ (c) $f(x) = (4x - 3)^3$ (d) $f(x) = (4x - 3)^{1/3}$

2. Encuentre el error hacia adelante y hacia atrás para las siguientes funciones, donde la raíz es 1/3 y la raíz aproximada es $x_a = 0.3333$: (a) $f(x) = 3x - 1$ (b) $f(x) = (3x - 1)^2$ (c) $f(x) = (3x - 1)^3$ (d) $f(x) = (3x - 1)^{1/3}$

3. (a) Encuentre la multiplicidad de la raíz $r = 0$ de $f(x) = 1 - \cos x$. (b) Encuentre los errores hacia adelante y hacia atrás de la raíz aproximada $x_a = 0.0001$.

4. (a) Encuentre la multiplicidad de la raíz $r = 0$ de $f(x) = x^2 \operatorname{sen} x^2$. (b) Encuentre los errores hacia adelante y hacia atrás de la raíz aproximada $x_a = 0.01$.

5. Encuentre la relación entre los errores hacia adelante y hacia atrás al localizar la raíz de la función lineal $f(x) = ax - b$.

6. Sea n un entero positivo. La ecuación que define la raíz n-ésima de un número positivo A es $x^n - A = 0$. (a) Encuentre la multiplicidad de la raíz. (b) Muestre que, para una raíz aproximada n-ésima con un pequeño error hacia adelante, el error hacia atrás es aproximadamente $nA^{(n-1)/n}$ veces el error hacia adelante.

7. Sea $W(x)$ el polinomio de Wilkinson. (a) Demuestre que $W'(16) = 15!4!$ (b) Encuentre una fórmula análoga para $W'(j)$, donde j es un número entero entre 1 y 20.

8. Sean $f(x) = x^n - ax^{n-1}$ y $g(x) = x^n$. (a) Utilice la fórmula de sensibilidad para hacer una predicción de la raíz distinta de cero de $f_\epsilon(x) = x^n - ax^{n-1} + \epsilon x^n$ para una ϵ pequeña. (b) Encuentre la raíz diferente de cero y compárela con la predicción.

1.3 Problemas de computadora

1. Sea $f(x) = \operatorname{sen} x - x$. (a) Encuentre la multiplicidad de la raíz $r = 0$. (b) Utilice el comando `fzero` de MATLAB con la estimación inicial $x = 0.1$ para localizar una raíz. ¿Cuáles son los errores hacia adelante y hacia atrás de la respuesta de `fzero`?

2. Resuelva el problema de computadora 1 para $f(x) = \operatorname{sen} x^3 - x^3$.

3. (a) Utilice `fzero` para encontrar la raíz de $f(x) = 2x \cos x - 2x + \operatorname{sen} x^3$ en $[-0.1, 0.2]$. Reporte los errores hacia adelante y hacia atrás. (b) Ejecute el método de bisección con el intervalo inicial $[-0.1, 0.2]$ para encontrar el mayor número posible de dígitos correctos, y registre su conclusión.

4. (a) Use (1.21) para aproximar la raíz cercana a 3 de $f_\epsilon(x) = (1 + \epsilon)x^3 - 3x^2 + x - 3$ para una ϵ constante. (b) Establezca $\epsilon = 10^{-3}$, encuentre la raíz real y compárela con la del inciso (a).

5. Use (1.21) para aproximar la raíz de $f(x) = (x - 1)(x - 2)(x - 3)(x - 4) - 10^{-6}x^6$ cerca de $r = 4$. Encuentre el factor de magnificación del error. Utilice `fzero` a fin de comprobar su aproximación.

6. Utilice el comando `fzero` de MATLAB para encontrar la raíz del polinomio de Wilkinson cerca de $x = 15$ con un cambio relativo de $\epsilon = 2 \times 10^{-15}$ en el coeficiente de x^{15}, haciendo que el coeficiente sea un poco más negativo. Compare el resultado con la predicción hecha por (1.21).

1.4 MÉTODO DE NEWTON

El método de Newton, también llamado método de Newton-Raphson, por lo general converge mucho más rápido que los métodos linealmente convergentes que se han visto hasta ahora. La imagen geométrica del método de Newton se muestra en la figura 1.8. Para encontrar una raíz de $f(x) = 0$, se da una estimación inicial x_0 y se traza la recta tangente a la función f en x_0. La recta tangente seguirá en forma aproximada a la función hasta el eje x hacia la raíz. El punto de intersección de la línea con el eje x es una raíz aproximada, pero probablemente no es exacta si la f es curva. Por lo tanto, este paso se itera.

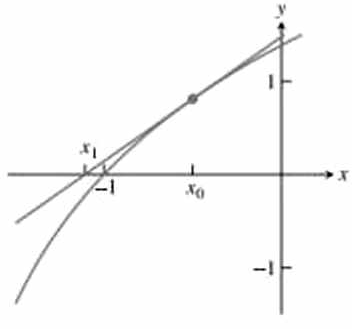


Figura 1.8 Un paso del método de Newton. A partir de x_0, se traza la recta tangente a la curva $y = f(x)$. El punto de intersección con el eje x es x_1, la siguiente aproximación a la raíz.

Con base en la imagen geométrica, es posible desarrollar una fórmula algebraica para el método de Newton. La recta tangente en x_0 tiene una pendiente dada por la derivada $f'(x_0)$. Un punto sobre la recta tangente es $(x_0, f(x_0))$. La fórmula de la pendiente de un punto para la ecuación de una

recta es $y - f(x_0) = f'(x_0)(x - x_0)$, de modo que para buscar el punto de intersección de la tangente con el eje x basta con sustituir $y = 0$ en la recta:

$$f'(x_0)(x - x_0) = 0 - f(x_0)$$
$$x - x_0 = -\frac{f(x_0)}{f'(x_0)}$$
$$x = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Al despejar x se obtiene una aproximación de la raíz, que se denomina x_1. Después, todo el proceso se repite, empezando con x_1, para generar x_2, y así sucesivamente. Con esto se obtiene la siguiente fórmula iterativa:

Método de Newton

$$x_0 = \text{estimación inicial}$$
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad \textbf{para } i = 0, 1, 2, \ldots$$

► **EJEMPLO 1.11** Encuentre la fórmula del método de Newton para la ecuación $x^3 + x - 1 = 0$.

Como $f'(x) = 3x^2 + 1$, la fórmula está dada por

$$x_{i+1} = x_i - \frac{x_i^3 + x_i - 1}{3x_i^2 + 1}$$
$$= \frac{2x_i^3 + 1}{3x_i^2 + 1}.$$

Al iterar esta fórmula desde la estimación inicial $x_0 = -0.7$, se obtiene

$$x_1 = \frac{2x_0^3 + 1}{3x_0^2 + 1} = \frac{2(-0.7)^3 + 1}{3(-0.7)^2 + 1} \approx 0.1271$$
$$x_2 = \frac{2x_1^3 + 1}{3x_1^2 + 1} \approx 0.9577.$$

Estos pasos se muestran geométricamente en la figura 1.9. Los pasos subsecuentes se dan en la siguiente tabla:

i	x_i	$e_i = \lvert x_i - r \rvert$	e_i / e_{i-1}^2
0	−0.70000000	1.38232780	
1	0.12712551	0.55520230	0.2906
2	0.95767812	0.27535032	0.8933
3	0.73482779	0.05249999	0.6924
4	0.68459177	0.00226397	0.8214
5	0.68233217	0.00000437	0.8527
6	0.68232780	0.00000000	0.8541
7	0.68232780	0.00000000	

Después de sólo seis pasos, se conoce la raíz hasta ocho dígitos correctos. Hay más cosas que pueden decirse acerca del error y de la rapidez con que éste se reduce. Observe en la tabla que, una vez que la convergencia empieza a dominar, el número de posiciones correctas en x_i se duplica para cada iteración. Esto es característico de los métodos "cuadráticamente convergentes", como se verá a continuación.

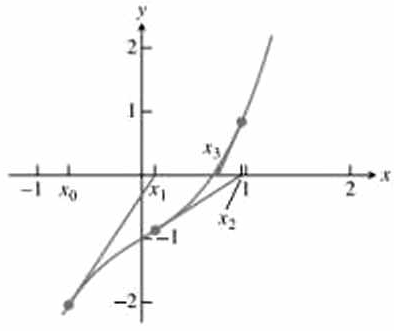


Figura 1.9 Tres pasos del método de Newton. Ilustración del ejemplo 1.11. A partir de $x_0 = -0.7$, se trazan las iteraciones del método de Newton junto con las rectas tangentes. El método parece que converge a la raíz. ◀

1.4.1 Convergencia cuadrática del método de Newton

La convergencia en el ejemplo 1.11 es cualitativamente más rápida que la convergencia lineal observada para el método de bisección y la iteración de punto fijo. Se requiere una nueva definición.

DEFINICIÓN 1.10 Sea e_i el error después del paso i de un método iterativo. La iteración es **cuadráticamente convergente** si

$$M = \lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} < \infty.$$

❐

TEOREMA 1.11 Sea f dos veces continuamente diferenciable y $f(r) = 0$. Si $f'(r) \neq 0$, entonces el método de Newton es local y cuadráticamente convergente a r. El error e_i en el paso i satisface

$$\lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} = M,$$

donde

$$M = \frac{f''(r)}{2f'(r)}.$$

■

Comprobación. Para comprobar la convergencia local, tenga en cuenta que el método de Newton es una forma particular de la iteración de punto fijo, donde

$$g(x) = x - \frac{f(x)}{f'(x)},$$

con una derivada

$$g'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}.$$

Como $g'(r) = 0$. El método de Newton converge localmente de acuerdo con el teorema 1.6.

Para comprobar la convergencia cuadrática, se deduce el método de Newton de una segunda manera, esta vez manteniendo una estrecha vigilancia sobre el error en cada paso. Por error, se entiende la diferencia entre la raíz correcta y la mejor estimación actual.

La fórmula de Taylor en el teorema 0.8 indica la diferencia entre los valores de una función en un punto dado y otro punto cercano. Para los dos puntos, se utilizará la raíz r y la estimación actual x_i después de i pasos, analizamos en esta iteración el resultado de la aproximación al describir el residuo con dos términos:

$$f(r) = f(x_i) + (r - x_i) f'(x_i) + \frac{(r - x_i)^2}{2} f''(c_i).$$

Aquí, c_i está entre x_i y r. Como r es la raíz, se tiene

$$0 = f(x_i) + (r - x_i) f'(x_i) + \frac{(r - x_i)^2}{2} f''(c_i)$$

$$-\frac{f(x_i)}{f'(x_i)} = r - x_i + \frac{(r - x_i)^2}{2} \frac{f''(c_i)}{f'(x_i)},$$

suponiendo que $f'(x_i) \neq 0$. Con algunos arreglos, es posible comparar la próxima iteración de Newton con la raíz:

$$x_i - \frac{f(x_i)}{f'(x_i)} - r = \frac{(r - x_i)^2}{2} \frac{f''(c_i)}{f'(x_i)}$$

$$x_{i+1} - r = e_i^2 \frac{f''(c_i)}{2f'(x_i)}$$

$$e_{i+1} = e_i^2 \left| \frac{f''(c_i)}{2f'(x_i)} \right|. \tag{1.24}$$

En esta ecuación, se ha definido que el error en el paso i es $e_i = |x_i - r|$. Como c_i se encuentra entre r y x_i, converge a r igual que lo hace x_i, y

$$\lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} = \left| \frac{f''(r)}{2f'(r)} \right|,$$

la definición de convergencia cuadrática. ❐

La fórmula de error (1.24) que se ha desarrollado puede verse como

$$e_{i+1} \approx M e_i^2, \tag{1.25}$$

donde $M = |f''(r)/2f'(r)|$, bajo el supuesto de que $f'(r) \neq 0$. La aproximación mejora a medida que el método de Newton converge, puesto que las estimaciones x_i se desplazan hacia r, y porque c_i está capturada entre x_i y r. Esta fórmula de error debe compararse con $e_{i+1} \approx S e_i$ para los métodos linealmente convergentes, donde $S = |g'(r)|$ para la IPF y $S = 1/2$ para la bisección.

Aunque el valor de S es crítico para los métodos linealmente convergentes, el valor de M resulta menos crítico, debido a que la fórmula incluye el cuadrado del error anterior. Una vez que el error se coloca significativamente por debajo de 1, la elevación al cuadrado causará una disminución aún mayor; y, siempre y cuando M no sea muy grande, el error de acuerdo con (1.25) también disminuirá.

De regreso al ejemplo 1.11, es posible analizar la tabla de resultados para demostrar esta razón del error. La columna de la derecha muestra la relación e_i/e_{i-1}^2, que, de acuerdo con la fórmula de error del método de Newton (1.25), debe tender a M a medida que se presente la convergencia hacia la raíz. Para $f(x) = x^3 + x - 1$, las derivadas son $f'(x) = 3x^2 + 1$ y $f''(x) = 6x$; al evaluar en $x_c \approx 0{,}6823$ se obtiene $M \approx 0{,}85$, que coincide con la relación de error en la columna derecha de la tabla.

Con este nuevo entendimiento del método de Newton, es posible explicar con más detalle la calculadora de raíces cuadradas del ejemplo 1.6. Sea a un número positivo y considere la localización de las raíces de $f(x) = x^2 - a$ por el método de Newton. La iteración es

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{x_i^2 - a}{2x_i}$$

$$= \frac{x_i^2 + a}{2x_i} = \frac{x_i + \frac{a}{x_i}}{2}, \tag{1.26}$$

que es el método del ejemplo 1.6, para una a arbitraria.

Para estudiar su convergencia, evalúe las derivadas en la raíz $\sqrt{a}$:

$$f'(\sqrt{a}) = 2\sqrt{a}$$
$$f''(\sqrt{a}) = 2. \tag{1.27}$$

Newton es cuadráticamente convergente, puesto que $f'(\sqrt{a}) = 2\sqrt{a} \neq 0$, y la razón de convergencia es

$$e_{i+1} \approx Me_i^2, \tag{1.28}$$

donde $M = 2/(2 \cdot 2\sqrt{a}) = 1/(2\sqrt{a})$.

1.4.2 Convergencia lineal del método de Newton

El teorema 1.11 no dice que el método de Newton siempre converge cuadráticamente. Recuerde que es necesario dividir entre $f'(r)$ para que el argumento de convergencia cuadrática tenga sentido. Este supuesto resulta ser crucial. En el siguiente ejemplo se muestra un caso donde el método de Newton no converge cuadráticamente:

► EJEMPLO 1.12 Utilice el método de Newton para encontrar una raíz de $f(x) = x^2$.

Éste puede parecer un problema trivial, puesto que ya se sabe que existe una raíz: $r = 0$. Pero con frecuencia resulta instructivo aplicar un nuevo método con un ejemplo que se entienda por completo. La fórmula del método de Newton es

$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)}{f'(x_i)} \\ &= x_i - \frac{x_i^2}{2x_i} \\ &= \frac{x_i}{2}. \end{aligned}$$

El resultado sorprendente es que el método de Newton se simplifica a una división entre dos. Como la raíz es $r = 0$, se tiene la siguiente tabla de iteraciones de Newton para la estimación inicial $x_0 = 1$:

i	x_i	$e_i = \lvert x_i - r \rvert$	e_i/e_{i-1}
0	1.000	1.000	
1	0.500	0.500	0.500
2	0.250	0.250	0.500
3	0.125	0.125	0.500
⋮	⋮	⋮	⋮

El método de Newton converge a la raíz $r = 0$. La fórmula del error es $e_{i+1} = e_i/2$, de modo que la convergencia es lineal con una constante de proporcionalidad de convergencia $S = 1/2$. ◄

Para x^m, donde m es cualquier entero positivo, existe un resultado similar, como lo muestra el siguiente ejemplo.

► EJEMPLO 1.13 Utilice el método de Newton para encontrar una raíz de $f(x) = x^m$.

La fórmula de Newton es

$$\begin{aligned} x_{i+1} &= x_i - \frac{x_i^m}{mx_i^{m-1}} \\ &= \frac{m-1}{m} x_i. \end{aligned}$$

ANOTACIÓN

Convergencia Las ecuaciones de convergencia (1.28) y (1.29) expresan las dos razones de convergencia diferentes a la raíz r que son posibles en el método de Newton. En una raíz simple, $f'(r) \neq 0$, la convergencia es cuadrática, o rápida, lo que obedece a (1.28). En una raíz múltiple, $f'(r) = 0$, la convergencia es lineal, lo que obedece a (1.29). En este último caso de convergencia lineal, la razón más lenta pone al método de Newton en la misma categoría que la bisección y la IPF.

Una vez más, la única raíz es $r = 0$, por lo que al definir $e_i = |x_i - r| = x_i$ resulta

$$e_{i+1} = Se_i,$$

donde $S = (m - 1)/m$. ◄

Éste es un ejemplo del comportamiento general del método de Newton con las raíces múltiples. Tenga en cuenta que la definición 1.9 de raíz múltiple es equivalente a $f(r) = f'(r) = 0$, exactamente el caso en el que no se ha podido realizar el trabajo de obtención de la fórmula de error en el método de Newton. Existe una fórmula de error distinta en raíces múltiples. El patrón que se vio en las raíces múltiples de monomios es representativo del caso general, como se resume en el teorema 1.12.

TEOREMA 1.12 Suponga que la función f, continuamente diferenciable $(m + 1)$ veces en $[a, b]$, tiene una raíz r con multiplicidad m. Entonces, el método de Newton converge localmente a r, y el error e_i en el paso i satisface

$$\lim_{i \to \infty} \frac{e_{i+1}}{e_i} = S, \tag{1.29}$$

donde $S = (m - 1)/m$. ■

► EJEMPLO 1.14 Encuentre la multiplicidad de la raíz $r = 0$ de $f(x) = \operatorname{sen} x + x^2 \cos x - x^2 - x$, y estime el número de pasos necesarios del método de Newton para converger a seis posiciones correctas (use $x_0 = 1$).

Es fácil comprobar que

$$\begin{aligned} f(x) &= \operatorname{sen} x + x^2 \cos x - x^2 - x \\ f'(x) &= \cos x + 2x \cos x - x^2 \operatorname{sen} x - 2x - 1 \\ f''(x) &= -\operatorname{sen} x + 2\cos x - 4x \operatorname{sen} x - x^2 \cos x - 2 \end{aligned}$$

y que cada una se evalúa como 0 en $r = 0$. La tercera derivada,

$$f'''(x) = -\cos x - 6 \operatorname{sen} x - 6x \cos x + x^2 \operatorname{sen} x, \tag{1.30}$$

satisface $f'''(x) = -1$, por lo que la raíz $r = 0$ es una raíz triple, lo que significa que la multiplicidad es $m = 3$. Por el teorema 1.12, Newton debería converger linealmente con $e_{i+1} \approx 2e_i/3$.

Si se usa la estimación inicial $x_0 = 1$, se tiene $e_0 = 1$. Cerca de la convergencia, el error se reducirá en 2/3 a cada paso. Por lo tanto, una aproximación al número de pasos necesarios para obtener el error hasta seis posiciones decimales, o menor que 0.5×10^{-6}, puede encontrarse al resolver

$$\begin{aligned} \left(\frac{2}{3}\right)^n &< 0.5 \times 10^{-6} \\ n &> \frac{\log_{10}(.5) - 6}{\log_{10}(2/3)} \approx 35.78. \end{aligned} \tag{1.31}$$

Se necesitarán aproximadamente 36 pasos. En la siguiente tabla se muestran los primeros 20 pasos.

i	x_i	$e_i = \lvert x_i - r \rvert$	e_i/e_{i-1}
1	1.00000000000000	1.00000000000000	
2	0.72159023986075	0.72159023986075	0.72159023986075
3	0.52137095182040	0.52137095182040	0.72253049309677
4	0.37530830859076	0.37530830859076	0.71984890466250
5	0.26836349052713	0.26836349052713	0.71504809348561
6	0.19026161369924	0.19026161369924	0.70896981301561
7	0.13361250532619	0.13361250532619	0.70225676492686
8	0.09292528672517	0.09292528672517	0.69548345417455
9	0.06403926677734	0.06403926677734	0.68914790617474
10	0.04377806216009	0.04377806216009	0.68361279513559
11	0.02972805552423	0.02972805552423	0.67906284694649
12	0.02008168373777	0.02008168373777	0.67551285759009
13	0.01351212730417	0.01351212730417	0.67285828621786
14	0.00906579564330	0.00906579564330	0.67093770205249
15	0.00607029292263	0.00607029292263	0.66958192766231
16	0.00405885109627	0.00405885109627	0.66864171927113
17	0.00271130367793	0.00271130367793	0.66799781850081
18	0.00180995966250	0.00180995966250	0.66756065624029
19	0.00120772384467	0.00120772384467	0.66726561353325
20	0.00080563307149	0.00080563307149	0.66706728946460

Observe en la columna de la derecha la convergencia de la razón de error hacia la predicción de 2/3. ◀

Si la multiplicidad de una raíz se conoce de antemano, la convergencia del método de Newton puede mejorarse con una pequeña modificación.

TEOREMA 1.13 Si f es continuamente diferenciable $(m + 1)$ veces en $[a, b]$, que contiene una raíz r de multiplicidad $m > 1$, entonces el **método de Newton modificado**

$$x_{i+1} = x_i - \frac{mf(x_i)}{f'(x_i)} \tag{1.32}$$

converge local y cuadráticamente a r. ■

De regreso al ejemplo 1.14, es posible aplicar el método de Newton modificado para lograr la convergencia cuadrática. Después de cinco pasos, se presenta la convergencia a la raíz $r = 0$ aproximadamente hasta ocho dígitos de precisión:

i	x_i
0	1.00000000000000
1	0.16477071958224
2	0.01620733771144
3	0.00024654143774
4	0.00000006072272
5	−0.00000000633250

Existen varios puntos a tener en cuenta en la tabla. En primer lugar, es observable la convergencia cuadrática a la raíz aproximada, puesto que el número de posiciones decimales correctas en la aproximación más o menos se duplica a cada paso, hasta el paso 4. Los pasos 6, 7, ... son idénticos al paso 5. La razón por la que el método de Newton carece de convergencia a la precisión de máquina resulta familiar si se recuerda la sección 1.3.

Se sabe que 0 es una raíz múltiple. Mientras el método de Newton conduce al error hacia atrás cerca de ϵ_{maq}, el error hacia adelante, igual a x_i, es varias veces más grande.

El método de Newton, al igual que la IPF, puede no converger a una raíz. El siguiente ejemplo muestra sólo uno de sus comportamientos no convergentes posibles.

► **EJEMPLO 1.15** Aplique el método de Newton para $f(x) = 4x^4 - 6x^2 - 11/4$ con estimación inicial $x_0 = 1/2$.

Esta función tiene raíces, puesto que es continua, negativa en $x = 0$ y tiende al infinito positivo para las x positivas y negativas de gran tamaño. Sin embargo, con la estimación inicial $x_0 = 1/2$, no se encuentra ninguna raíz como lo muestra la figura 1.10. La fórmula de Newton es

$$x_{i+1} = x_i - \frac{4x_i^4 - 6x_i^2 - \frac{11}{4}}{16x_i^3 - 12x_i}. \tag{1.33}$$

La sustitución da $x_1 = -1/2$ y después $x_2 = 1/2$ de nuevo. El método de Newton fluctúa en este ejemplo entre los valores 1/2 y $-1/2$, que no son raíces, por lo tanto no se encuentra ninguna raíz.

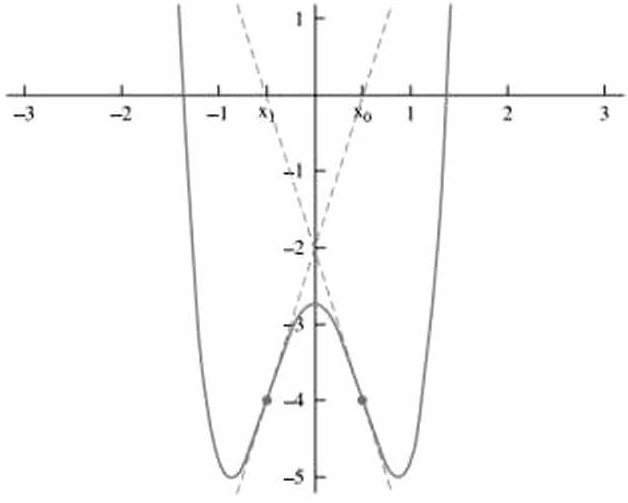


Figura 1.10 Falla del método de Newton en el ejemplo 1.15. Las iteraciones fluctúan entre 1/2 y − 1/2, y no convergen a ninguna raíz. ◄

El método de Newton puede fallar en otras formas. Obviamente, si $f'(x_i) = 0$ en cualquier paso de la iteración, el método no puede continuar. Existen otros ejemplos en los que la iteración diverge al infinito (vea el ejercicio 6) o imita a un generador de números aleatorios (vea el problema de computadora 13). Aunque no todas las estimaciones iniciales conducen a la convergencia hacia una raíz, los teoremas 1.11 y 1.12 garantizan una vecindad de estimaciones iniciales que rodean cada raíz para que la convergencia a esa raíz esté asegurada.

1.4 Ejercicios

1. Aplique dos pasos del método de Newton con una estimación inicial $x_0 = 0$. (a) $x^3 + x - 2 = 0$ (b) $x^4 - x^2 + x - 1 = 0$ (c) $x^2 - x - 1 = 0$
2. Aplique dos pasos del método de Newton con una estimación inicial $x_0 = 1$. (a) $x^3 + x^2 - 1 = 0$ (b) $x^2 + 1/(x + 1) - 3x = 0$ (c) $5x - 10 = 0$
3. Use los teoremas 1.11 o 1.12 para estimar el error e_{i+1} en términos del error previo e_i cuando el método de Newton converge a las raíces dadas. ¿La convergencia es lineal o cuadrática?

(a) $x^5 - 2x^4 + 2x^2 - x = 0$; $r = -1, r = 0, r = 1$ (b) $2x^4 - 5x^3 + 3x^2 + x - 1 = 0$; $r = -1/2, r = 1$

4. Estime e_{i+1} como en el ejercicio 3. (a) $32x^3 - 32x^2 - 6x + 9 = 0$; $r = -1/2$, $r = 3/4$ (b) $x^3 - x^2 - 5x - 3 = 0$; $r = -1$, $r = 3$

5. Considere la ecuación $8x^4 - 12x^3 + 6x^2 - x = 0$. Para cada una de las dos soluciones $x = 0$ y $x = 1/2$, decida cuál método convergirá más rápido (por ejemplo, hasta una precisión de ocho posiciones), el método de bisección o el método de Newton, sin ejecutar el cálculo.

6. Trace una función f y una estimación inicial para la cual el método de Newton diverja.

7. Sea $f(x) = x^4 - 7x^3 + 18x^2 - 20x + 8$. ¿El método de Newton converge cuadráticamente a la raíz $r = 2$? Encuentre $\lim_{i \to \infty} e_{i+1}/e_i$, donde e_i indica el error en el paso i.

8. Demuestre que el método de Newton aplicado a $f(x) = ax + b$ converge en un solo paso.

9. Muestre que al aplicar el método de Newton a $f'(x) = x^2 - A$ se produce la iteración del ejemplo 1.6.

10. Encuentre la iteración de punto fijo producida al aplicar el método de Newton a $f(x) = x^3 - A$. Vea el ejercicio 1.2.10.

11. Utilice el método de Newton para producir un método cuadráticamente convergente en el cálculo de la raíz n-ésima de un número positivo A, donde n es un entero positivo. Demuestre la convergencia cuadrática.

12. Suponga que el método de Newton se aplica a la función $f(x) = 1/x$. Si la estimación inicial es $x_0 = 1$, encuentre x_{50}.

13. (a) La función $f(x) = x^3 - 4x$ tiene una raíz en $r = 2$. Si el error $e_i = x_i - r$ después de cuatro pasos del método de Newton es $e_4 = 10^{-6}$, estime e_5. (b) Resuelva el problema planteado en el inciso (a) para la raíz $r = 0$. (Precaución: La fórmula habitual no resulta útil).

14. Si $g(x) = x - f(x)/f'(x)$, indique la iteración del método de Newton para la función f. Defina $h(x) = g(g(x))$ que es el resultado de dos pasos sucesivos del método de Newton. Entonces $h'(x) = g'(g(x))g'(x)$ de acuerdo con la regla de la cadena de cálculo. (a) Suponga que c es un punto fijo de h, pero no de g, como en el ejemplo 1.15. Muestre que si c es un punto de inflexión de $f(x)$, es decir $f''(x) = 0$, entonces la iteración de punto fijo h converge localmente a c. De lo anterior se deduce que, para aproximaciones iniciales cercanas a c, el mismo método de Newton no converge a una raíz de f, sino que tiende a la secuencia oscilante $\{c, g(c)\}$. (b) Verifique que la oscilación estable descrita en (a) ocurra de verdad en el ejemplo 1.15. El problema de computadora 14 profundiza en esto.

1.4 Problemas de computadora

1. Cada ecuación tiene una raíz. Utilice el método de Newton para aproximar la raíz hasta ocho decimales correctos. (a) $x^3 = 2x + 2$ (b) $e^x + x = 7$ (c) $e^x + \operatorname{sen} x = 4$

2. Cada ecuación tiene una raíz real. Utilice el método de Newton para aproximar la raíz a ocho decimales correctos, (a) $x^5 + x = 1$ (b) $\operatorname{sen} x = 6x + 5$ (c) $\ln x + x^2 = 3$

3. Aplique el método de Newton para encontrar la única raíz hasta la mayor exactitud posible y determine la multiplicidad de la raíz. Después, utilice el método de Newton modificado para converger a la raíz cuadrática. Registre los errores hacia adelante y hacia atrás de la mejor aproximación obtenida con cada método. (a) $f(x) = 27x^3 - 54x^2 - 36x + 8$ (b) $f(x) = = 36x^4 - 12x^3 + 37x^2 - 12x + 1$

4. Realice los pasos del problema de computadora 3 para: (a) $f(x) = 2e^{x-1} - x^2 - 1$ (b) $f(x) = \ln(3 - x) + x - 2$.

5. Un silo compuesto por un cilindro circular recto de 10 m altura y con tapa formada por una cúpula hemisférica, contiene 400 m^3 de volumen. Encuentre el radio de la base del silo con cuatro decimales correctos.

6. Un cono de 10 cm de alto contiene 60 cm^3 de helado, incluyendo una bola semiesférica en la parte superior. Encuentre el radio de la bola con cuatro decimales correctos.

7. Considere la función $f(x) = e^{\text{sen}^3 x} + x^6 - 2x^4 - x^3 - 1$ en el intervalo $[-2, 2]$. Grafique la función en el intervalo y encuentre las tres raíces con seis cifras decimales correctas. Determine qué raíces convergen cuadráticamente y encuentre la multiplicidad de las raíces que convergen linealmente.

8. Realice los pasos del problema de computadora 7 para la función $f(x) = 94\cos^3 x - 24\cos x + 177\,\text{sen}^2 x - 108\,\text{sen}^4 x - 72\cos^3 x\,\text{sen}^2 x - 65$ en el intervalo $[0, 3]$.

9. Aplique el método de Newton para encontrar las dos raíces de la función $f(x) = 14xe^{x-2} - 12e^{x-2} - 7x^3 + 20x^2 - 26x + 12$ en el intervalo $[0, 3]$. Para cada raíz, imprima la secuencia de iteraciones, los errores e_i y la razón de error correspondiente e_{i+1}/e_i^2 o e_{i+1}/e_i que converja a un límite distinto de cero. Busque la coincidencia del límite con el valor esperado M del teorema 1.11 o S del teorema 1.12.

10. Sea $f(x) = 54x^6 + 45x^5 - 102x^4 - 69x^3 + 35x^2 + 16x - 4$. Grafique la función en el intervalo $[-2, 2]$ y utilice el método de Newton para encontrar las cinco raíces en el intervalo. Determine las raíces para las que Newton converge linealmente y para las cuales la convergencia es cuadrática.

11. La ley de los gases ideales para un gas a baja temperatura y presión es $PV = nRT$, donde P es la presión (en atm), V es el volumen (en L), T es la temperatura (en K), n es el número de moles del gas y $R = 0.0820578$ es la constante molar del gas. La ecuación de van der Waals

$$\left(P + \frac{n^2 a}{V^2}\right)(V - nb) = nRT$$

se refiere al caso no ideal donde estos supuestos no se cumplen. Utilice la ley de los gases ideales para calcular una estimación inicial, seguida por la aplicación del método de Newton a la ecuación de van der Waals a fin de encontrar el volumen de un mol de oxígeno a 320 K y una presión de 15 atm. Para el oxígeno, $a = 1.36$ L^2-atm/mol^2 y $b = 0.003183$ L/mol. Indique su estimación inicial y la solución con tres cifras significativas.

12. Use los datos del problema de computadora 11 para encontrar el volumen de 1 mol de vapor de benceno a 700 K bajo una presión de 20 atm. Para el benceno, $a = 18.0$ L^2-atm/mol^2 y $b = 0.1154$ L/mol.

13. (a) Encuentre la raíz de la función $f(x) = (1 - 3/(4x))^{1/3}$. (b) Aplique el método de Newton con una estimación inicial cercana a la raíz y grafique las primeras 50 iteraciones. Ésta es otra manera en la que el método de Newton puede fallar, produciendo una trayectoria caótica. (c) ¿Por qué los teoremas 1.11 y 1.12 no son aplicables?

14. (a) Fije los números reales a, $b > 0$ y trace la gráfica de $f(x) = a^2x^4 - 6abx^2 - 11b^2$ para sus valores elegidos. No utilice $a = 2$, $b = 1/2$, puesto que estos valores ya aparecen en el ejemplo 1.15. (b) Aplique el método de Newton para encontrar tanto la raíz negativa como la raíz positiva de $f(x)$. Después, encuentre los intervalos de las estimaciones iniciales positivas $[d_1, d_2]$, donde $d_2 > d_1$, para los cuales el método de Newton: (c) converge a la raíz positiva, (d) converge a la raíz negativa, (e) está definido, pero no converge a ninguna raíz. Sus intervalos no deben contener ningún valor inicial donde $f'(x) = 0$, en el que el método de Newton no está definido.

1.5 LOCALIZACIÓN DE RAÍCES SIN DERIVADAS

Además de las raíces múltiples, el método de Newton converge a una velocidad más rápida que los métodos de bisección y de IPF. Esto se logra porque utiliza más información (en particular, información acerca de la recta tangente a la función, que se obtiene de la derivada de la función). En algunas circunstancias, la derivada puede no estar disponible.

El método de la secante es un buen sustituto del método de Newton en este caso. Sustituye la recta tangente con una aproximación llamada recta secante, y converge casi con la misma rapidez. Las variantes del método de la secante reemplazan la recta con una parábola de aproximación, cuyo eje es vertical (método de Muller) u horizontal (interpolación cuadrática inversa). La sección termina con la descripción del método de Brent, un método híbrido que combina las mejores características de los métodos iterativos y de confinamiento.

1.5.1 Método de la secante y sus variantes

El método de la secante es similar al método de Newton, pero sustituye la derivada con un cociente de diferencias. Geométricamente, la recta tangente se sustituye por una línea que pasa por las dos últimas estimaciones conocidas. El punto de intersección de la "recta secante" es el nuevo valor estimado.

Una aproximación de la derivada en la estimación actual x_i es el cociente de diferencias

$$\frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}.$$

Con un reemplazo directo de esta aproximación por $f'(x_i)$ en el método de Newton se obtiene el método de la secante.

Método de la secante

$$x_0, x_1 = \text{estimaciones iniciales}$$
$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})} \quad \textbf{para } i = 1, 2, 3, \ldots$$

A diferencia de la iteración de punto fijo y del método de Newton, para comenzar a aplicar el método de la secante se requieren dos estimaciones iniciales.

Es posible demostrar que, bajo el supuesto de que el método de la secante converge a r y $f'(r) \neq 0$, se cumple la relación aproximada del error

$$e_{i+1} \approx \left| \frac{f''(r)}{2f'(r)} \right| e_i e_{i-1}$$

y esto implica que

$$e_{i+1} \approx \left| \frac{f''(r)}{2f'(r)} \right|^{\alpha-1} e_i^{\alpha},$$

donde $\alpha = (1 + \sqrt{5})/2 \approx 1.62$. (Vea el ejercicio 6). La convergencia del método de la secante hacia las raíces sencillas se llama **superlineal**, lo que significa que se encuentra entre los métodos lineal y cuadráticamente convergentes.

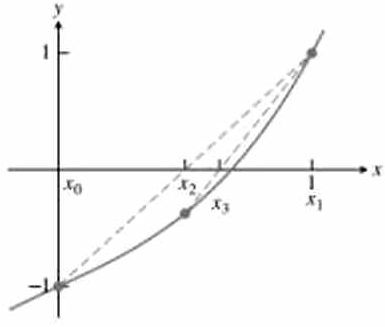


Figura 1.11 Dos pasos del método de la secante. Ilustración del ejemplo 1.16. Inicia con $x_0 = 0$ y $x_1 = 1$, las iteraciones del método de la secante se representan junto con las rectas secantes.

► **EJEMPLO 1.16** Aplique el método de la secante con estimaciones iniciales $x_0 = 0$ y $x_1 = 1$, para encontrar la raíz de $f(x) = x^3 + x - 1$.

La fórmula da

$$x_{i+1} = x_i - \frac{(x_i^3 + x_i - 1)(x_i - x_{i-1})}{x_i^3 + x_i - (x_{i-1}^3 + x_{i-1})}. \tag{1.34}$$

A partir de $x_0 = 0$ y $x_1 = 1$, se calcula

$$x_2 = 1 - \frac{(1)(1-0)}{1+1-0} = \frac{1}{2}$$

$$x_3 = \frac{1}{2} - \frac{-\frac{3}{8}(1/2 - 1)}{-\frac{3}{8} - 1} = \frac{7}{11},$$

como se muestra en la figura 1.11. Las iteraciones posteriores forman la siguiente tabla:

i	x_i
0	0.00000000000000
1	1.00000000000000
2	0.50000000000000
3	0.63636363636364
4	0.69005235602094
5	0.68202041964819
6	0.68232578140989
7	0.68232780435903
8	0.68232780382802
9	0.68232780382802

◄

Existen tres generalizaciones del método de la secante que también son importantes. El **método de la posición falsa**, o **regula falsi**, es similar al método de bisección, excepto que el punto medio se sustituye por un método secante, como aproximación. Dado un intervalo $[a, b]$ que contiene a una raíz (se asume que $f(a)\,f(b) < 0$), se define el siguiente punto

$$c = a - \frac{f(a)(a-b)}{f(a) - f(b)} = \frac{bf(a) - af(b)}{f(a) - f(b)}$$

como en el método de la secante, pero a diferencia de éste, se garantiza que el nuevo punto se encuentre en $[a, b]$, puesto que los puntos $(a, f(a))$ y $(b, f(b))$ se encuentran en lados diferentes del eje x.

El nuevo intervalo, $[a, c]$ o bien $[c, b]$, se elige dependiendo de si $f(a)\,f(c) < 0$ o $f(c)\,f(b) < 0$, respectivamente, y todavía contiene una raíz.

Método de la posición falsa

Dado el intervalo $[a, b]$ tal que $f(a)\,f(b) < 0$

for $i = 1, 2, 3, \ldots$

$$c = \frac{bf(a) - af(b)}{f(a) - f(b)}$$

if $f(c) = 0$, **stop**, **end**

if $f(a)\,f(c) < 0$

$b = c$

else

$a = c$

end

end

El método de la posición falsa a primera vista parece ser una mejora tanto del método de bisección como del método de la secante, con las mejores propiedades de cada uno. Sin embargo, mientras el método de bisección garantiza reducir la incertidumbre en 1/2 a cada paso, la posición falsa no hace tal promesa y en algunos casos puede converger muy lentamente.

► EJEMPLO 1.17 Aplique el método de la posición falsa sobre el intervalo inicial $[-1, 1]$ para encontrar la raíz $r = 0$ de $f(x) = x^3 - 2x^2 + \frac{3}{2}x$.

Dado $x_0 = -1, x_1 = 1$ como el intervalo de confinamiento inicial, se calcula el nuevo punto

$$x_2 = \frac{x_1 f(x_0) - x_0 f(x_1)}{f(x_0) - f(x_1)} = \frac{1(-9/2) - (-1)1/2}{-9/2 - 1/2} = \frac{4}{5}.$$

Como $f(-1)\,f(4/5) < 0$, el nuevo intervalo de confinamiento es $[x_0, x_2] = [-1, 0.8]$. Esto completa el primer paso. Observe que la incertidumbre en la solución se ha reducido en mucho menos que un factor de 1/2. Como se muestra en la figura 1.12(b), los pasos subsecuentes continúan avanzando con lentitud hacia la raíz en $x = 0$.

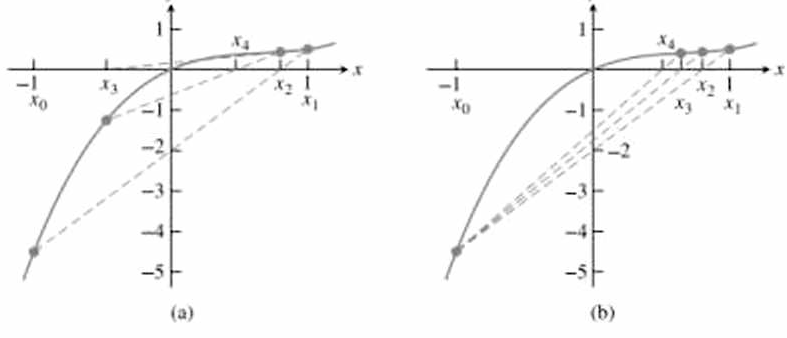


Figura 1.12 Convergencia lenta en el ejemplo 1.17. Tanto (a) el método de la secante como (b) el método de la posición falsa convergen con lentitud a la raíz $r = 0$. ◄

El **método de Muller** es una generalización del método de la secante en una dirección diferente. En lugar de intersecar la recta que pasa por dos puntos anteriores con el eje x, se utilizan tres puntos previos x_0, x_1, x_2, se dibuja la parábola $y = p(x)$ a través de ellos, y se interseca la parábola

con el eje x. La parábola por lo general intersecará en 0 o en 2 puntos. Si hay dos puntos de intersección, se elige el más cercano al último punto x_2 para ser x_3. Para determinar las dos posibilidades, tan sólo se aplica la fórmula cuadrática. Si la parábola no toca al eje x, las soluciones son números complejos. Lo anterior permite que el software capaz de manejar la aritmética compleja pueda localizar raíces complejas. Esta idea no se tratará más, aunque existen varias fuentes en los libros que siguen esta dirección.

La **interpolación cuadrática inversa** (ICI) es una generalización similar del método de la secante hacia las parábolas. Sin embargo, la parábola tiene la forma $x = p(y)$ en vez de $y = p(x)$, como en el método de Muller. Un problema se resuelve de inmediato: esta parábola intersecará al eje x en un solo punto, por lo que no hay ambigüedad en la localización de x_{i+3} a partir de las tres estimaciones anteriores, x_i, x_{i+1} y x_{i+2}.

El polinomio de segundo grado $x = P(y)$ que pasa a través de los tres puntos (a, A), (b, B), (c, C) es

$$P(y) = a\frac{(y-B)(y-C)}{(A-B)(A-C)} + b\frac{(y-A)(y-C)}{(B-A)(B-C)} + c\frac{(y-A)(y-B)}{(C-A)(C-B)}. \tag{1.35}$$

Éste es un ejemplo de interpolación de Lagrange, uno de los temas del capítulo 3. Por ahora, basta con notar que $P(A) = a$, $P(B) = b$ y $P(C) = c$. Al sustituir $y = 0$ se obtiene una fórmula para el punto de intersección de la parábola con el eje x. Después de algunos reordenamientos y sustituciones, se tiene

$$P(0) = c - \frac{r(r-q)(c-b) + (1-r)s(c-a)}{(q-1)(r-1)(s-1)}, \tag{1.36}$$

donde $q = f(a)/f(b)$, $r = f(c)/f(b)$ y $s = f(c)/f(a)$.

Para ICI, después de establecer $a = x_i$, $b = x_{i+1}$, $c = x_{i+2}$ y $A = f(x_i))$, $B = f(x_{i+1})$, $C = f(x_{i+2})$, la siguiente estimación $x_{i+3} = P(0)$ es

$$x_{i+3} = x_{i+2} - \frac{r(r-q)(x_{i+2}-x_{i+1}) + (1-r)s(x_{i+2}-x_i)}{(q-1)(r-1)(s-1)}, \tag{1.37}$$

donde $q = f(x_i)/f(x_{i+1})$, $r = f(x_{i+2})/f(x_{i+1})$ y $s = f(x_{i+2})/f(x_i)$. Dadas las tres estimaciones iniciales, el método ICI continúa iterando (1.37), utilizando la nueva estimación x_{i+3} para sustituir la antigua aproximación x_i. Una implementación alternativa de ICI utiliza la nueva estimación para reemplazar a una de las previas tres aproximaciones con el mayor error hacia atrás.

En la figura 1.13 se compara la geometría del método de Muller con la interpolación cuadrática inversa. Ambos métodos convergen más rápido que el método de la secante debido a la interpolación de orden superior. La interpolación se estudiará con mayor detalle en el capítulo 3. Los conceptos del método de la secante y sus generalizaciones, junto con el método de bisección, son los ingredientes claves del método de Brent, el tema de la siguiente sección.

1.5.2 Método de Brent

El método de Brent [Brent, 1973] es un método híbrido; utiliza partes de otras técnicas de resolución presentadas con anterioridad para desarrollar un nuevo enfoque que conserva las propiedades más útiles de cada una. Lo ideal es combinar la característica de convergencia garantizada, del método de bisección, con la característica de convergencia rápida de los métodos más sofisticados. Fue propuesto en un principio por Dekker y Van Wijngaarden en la década de 1960.

El método se aplica a una función f continua en un intervalo delimitado por a y b, donde $f(a)$ $f(b) < 0$. El método de Brent sigue la pista de un punto x_i actual que es el mejor en el sentido del error hacia atrás, y un intervalo $[a_i, b_i]$ para la raíz. En términos generales, se intenta el método de

interpolación cuadrática inversa y el resultado se utiliza para reemplazar x_i, a_i o b_i si (1) el error hacia atrás mejora y (2) el intervalo de confinamiento se reduce por lo menos a la mitad. Si no es así, se intenta el método de la secante con el mismo objetivo. Si éste también falla, se realiza un paso del método de bisección, lo que garantiza que la incertidumbre se reduzca por lo menos a la mitad.

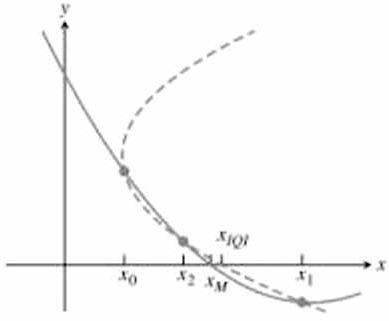


Figura 1.13 Comparación de un paso del método de Muller con un paso de la iteración inversa cuadrática. El primero se determina mediante una parábola de interpolación $y = p(x)$, el segundo por medio de una parábola de interpolación $x = p(y)$.

El comando `fzero` de MATLAB implementa una versión del método de Brent, junto con un paso de preprocesamiento para descubrir un buen intervalo de confinamiento inicial si éste no ha sido proporcionado por el usuario. El criterio de paro es de un tipo de error mixto hacia adelante y hacia atrás. El algoritmo termina cuando el cambio de x_i al nuevo punto x_{i+1} es menor que $2\epsilon_{\text{máq}}$ máx$(1, x_i)$, o cuando el error hacia atrás $|f(x_i)|$ alcanza el cero de máquina.

El paso de preprocesamiento no se activa si el usuario proporciona un intervalo de confinamiento inicial. El siguiente uso del comando introduce la función $f(x) = x^3 + x - 1$ y el intervalo de confinamiento inicial [0, 1], y le pide a MATLAB mostrar los resultados parciales en cada iteración:

```
>> f=@(x) x^3+x-1;
>> fzero(f,[0 1],optimset('Display','iter'))

 Func-count       x                  f(x)          Procedure
    1               0                    -1        initial
    2               1                     1        initial
    3             0.5                -0.375        bisection
    4        0.636364             -0.105935        interpolation
    5        0.684910            0.00620153        interpolation
    6        0.682225           -0.000246683        interpolation
    7        0.682328          -5.43508e-007        interpolation
    8        0.682328           1.50102e-013        interpolation
    9        0.682328                      0        interpolation
Zero found in the interval: [0, 1].

ans=

   0.68232780382802
```

De manera alternativa, el comando

```
>> fzero(f,1)
```

busca una raíz de $f(x)$ cerca de $x = 1$ al localizar primero un intervalo de confinamiento para después aplicar el método de Brent.

1.5 Ejercicios

1. Aplique dos pasos del método de la secante a las siguientes ecuaciones con estimaciones iniciales $x_0 = 1$ y $x_1 = 2$. (a) $x^3 = 2x + 2$ (b) $e^x + x = 7$ (c) $e^x + \text{sen } x = 4$
2. Aplique dos pasos del método de la posición falsa con un intervalo de confinamiento inicial [1, 2] para las ecuaciones de ejercicio 1.
3. Aplique dos pasos de la interpolación cuadrática inversa a las ecuaciones del ejercicio 1. Use estimaciones iniciales $x_0 = 1$, $x_1 = 2$ y $x_2 = 0$, y actualice reteniendo las tres iteraciones más recientes.
4. Una pescadora comercial quiere poner la red en el agua a una profundidad donde la temperatura sea de 10 grados C. Se hace decender una línea de pesca con un termómetro pegado y encuentra que la temperatura es de 8 grados a una profundidad de 9 metros, y 15 grados a una profundidad de 5 metros. Use el método de la secante para determinar una mejor estimación de la profundidad a la que la temperatura es de 10 grados.
5. Deduzca la ecuación (1.36) al sustituir $y = 0$ en (1.35).
6. Si el método de la secante converge a r, $f'(r) \neq 0$ y $f''(r) \neq 0$, entonces puede mostrarse que se cumple la relación del error aproximado $e_{i+1} \approx |f''(r)/(2f'(r))|e_i e_{i-1}$. Demuestre que si además $\lim_{i\to\infty} e_{i+1}/e_i^\alpha$ existe y es distinto de cero para algunas $\alpha > 0$, entonces $\alpha = (1 + \sqrt{5})/2$ y $e_{i+1} \approx |(f''(r)/2f'(r))|^{\alpha-1} e_i^\alpha$.
7. Considere los siguientes cuatro métodos para calcular $2^{1/4}$, la raíz cuarta de 2. (a) Clasifique los métodos de acuerdo con su velocidad de convergencia, del más rápido al más lento. Asegúrese de dar razones para su clasificación.

   (A) Método de bisección aplicado a $f(x) = x^4 - 2$

   (B) Método de la secante aplicado a $f(x) = x^4 - 2$

   (C) Iteración de punto fijo aplicado a $g(x) = \dfrac{x}{2} + \dfrac{1}{x^3}$

   (D) Iteración de punto fijo aplicado a $g(x) = \dfrac{x}{3} + \dfrac{1}{3x^3}$

   (b) ¿Existe algún método que converja con mayor rapidez que todos los sugeridos aquí?

1.5 Problemas de computadora

1. Utilice el método de la secante para encontrar la solución (individual) de cada ecuación del ejercicio 1.
2. Use el método de la posición falsa para encontrar la solución de cada ecuación en el ejercicio 1.
3. Mediante la interpolación cuadrática inversa encuentre la solución de cada ecuación del ejercicio 1.
4. Sea $f(x) = 54x^6 + 45x^5 - 102x^4 - 69x^3 + 35x^2 + 16x - 4$. Grafique la función en el intervalo $[-2, 2]$ y utilice el método de la secante para encontrar las cinco raíces en el intervalo. ¿Para cuáles de las raíces la convergencia es lineal y para cuáles es superlineal?
5. En el ejercicio 1.1.6 se preguntó cuál es el resultado del método de bisección para $f(x) = 1/x$ en el intervalo $[-2, 1]$. Ahora compare ese resultado con la aplicación de `fzero` al problema.
6. ¿Qué sucede si se le pide a `fzero` encontrar la raíz de $f(x) = x^2$ cerca de 1?, (no use un intervalo de confinamiento). Explique el resultado. (b) Aplique la misma pregunta para $f(x) = 1 + \cos x$ cerca de -1.

Comprobación en la realidad **1** ***Cinemática de la plataforma Stewart***

Una plataforma Stewart consta de seis puntales, o juntas prismáticas, de longitud variable que soportan una carga. Las juntas prismáticas operan cambiando la longitud del puntal, por lo general en forma neumática o hidráulica. Como un robot de seis grados de libertad, la plataforma Stewart puede colocarse en cualquier punto y a la inclinación en el espacio tridimensional que esté dentro de su alcance.

Para simplificar las cosas, el proyecto se refiere a una versión en dos dimensiones de la plataforma de Stewart. Podría diseñarse un manipulador compuesto por una plataforma triangular en un plano fijo controlado por tres puntales, como se muestra en la figura 1.14. El triángulo interior representa la plataforma Stewart plana, cuyas dimensiones están definidas por las tres longitudes L_1, L_2 y L_3. Sea γ el ángulo a través del lado L_1. La posición de la plataforma se controla mediante los tres números p_1, p_2 y p_3, las longitudes variables de los tres puntales.

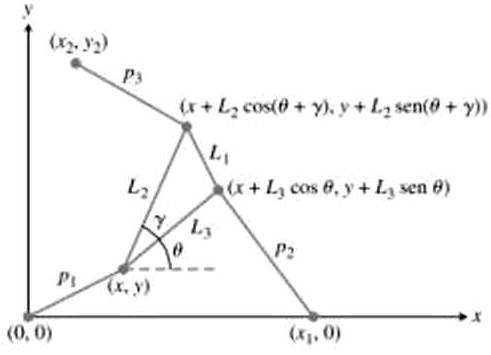


Figura 1.14 Esquema de plataforma Stewart plana. El problema directo de cinemática consiste en utilizar las longitudes p_1, p_2 y p_3 para determinar las incógnitas x, y, θ.

La determinación de la posición de la plataforma, dadas las tres longitudes de los puntales, se conoce como el problema directo de cinemática para este manipulador. Es decir, el problema consiste en calcular (x, y) y θ para cada p_1, p_2, p_3 dadas. Como hay tres grados de libertad, es natural esperar tres números para especificar la posición. A fin de planear el movimiento, es importante resolver este problema tan rápido como sea posible, a menudo en tiempo real. Desafortunadamente, no se conoce una solución cerrada para el problema directo de cinemática de la plataforma Stewart plana.

Los mejores métodos actuales implican la reducción de la geometría de la figura 1.14 a una sola ecuación y resolverla mediante el uso de uno de los solucionadores analizados en este capítulo. El trabajo consiste en completar la deducción de esta ecuación y escribir el código para encontrar su solución.

Si se aplica la trigonometría sencilla a la figura 1.14, se obtienen las siguientes tres ecuaciones:

$$\begin{aligned} p_1^2 &= x^2 + y^2 \\ p_2^2 &= (x + A_2)^2 + (y + B_2)^2 \\ p_3^2 &= (x + A_3)^2 + (y + B_3)^2. \end{aligned} \tag{1.38}$$

En estas ecuaciones,

$$\begin{aligned} A_2 &= L_3\cos\theta - x_1 \\ B_2 &= L_3\,\text{sen}\,\theta \\ A_3 &= L_2\cos(\theta+\gamma) - x_2 = L_2[\cos\theta\cos\gamma - \text{sen}\,\theta\,\text{sen}\,\gamma] - x_2 \\ B_3 &= L_2\,\text{sen}(\theta+\gamma) - y_2 = L_2[\cos\theta\,\text{sen}\,\gamma + \text{sen}\,\theta\cos\gamma] - y_2. \end{aligned}$$

Observe que (1.38) resuelve el problema inverso de cinemática para la plataforma Stewart plana, que consiste en encontrar p_1, p_2 y p_3, dados x, y, θ. Su objetivo es resolver el problema directo, es decir, encontrar x, y y θ dadas p_1, p_2 y p_3.

Al multiplicar las dos últimas ecuaciones de (1.38) y usar la primera se obtiene

$$p_2^2 = x^2 + y^2 + 2A_2x + 2B_2y + A_2^2 + B_2^2 = p_1^2 + 2A_2x + 2B_2y + A_2^2 + B_2^2$$
$$p_3^2 = x^2 + y^2 + 2A_3x + 2B_3y + A_3^2 + B_3^2 = p_1^2 + 2A_3x + 2B_3y + A_3^2 + B_3^2,$$

de donde pueden despejarse x y y como

$$x = \frac{N_1}{D} = \frac{B_3(p_2^2 - p_1^2 - A_2^2 - B_2^2) - B_2(p_3^2 - p_1^2 - A_3^2 - B_3^2)}{2(A_2B_3 - B_2A_3)}$$
$$y = \frac{N_2}{D} = \frac{-A_3(p_2^2 - p_1^2 - A_2^2 - B_2^2) + A_2(p_3^2 - p_1^2 - A_3^2 - B_3^2)}{2(A_2B_3 - B_2A_3)}, \tag{1.39}$$

siempre y cuando $D = 2(A_2B_3 - B_2A_3) \neq 0$.

Sustituyendo estas expresiones para x y y en la primera ecuación de (1.38) y multiplicando por D^2, resulta la siguiente ecuación,

$$f = N_1^2 + N_2^2 - p_1^2D^2 = 0 \tag{1.40}$$

con una sola incógnita, θ. (Recuerde que $p_1, p_2, p_3, L_1, L_2, L_3, \gamma, x_1, x_2$ y y_2 se conocen). Si pueden encontrarse las raíces de $f(\theta)$, los valores correspondientes de x y y, resultan inmediatamente de (1.39).

Tenga en cuenta que $f(\theta)$ es un polinomio de sen θ y cos θ, por lo que, dada cualquier raíz θ, existen otras raíces $\theta + 2\pi k$ que son equivalentes para la plataforma. Por esa razón, puede restringirse la atención a θ en $[-\pi, \pi]$. Puede demostrarse que $f(\theta)$ tiene como máximo seis raíces en ese intervalo.

Actividades sugeridas:

1. Escriba un archivo de función en MATLAB para $f(\theta)$. Los parámetros $L_1, L_2, L_3, \gamma, x_1, x_2$ y y_2 son constantes fijas y las longitudes de puntal p_1, p_2 y p_3 se conocerán para una posición dada. Vea el apéndice B.5 si es principiante en la creación de archivos de función en MATLAB. A continuación se presentan la primera y última líneas:

```
function out=f(theta)
      :
      :
out=N1^2+N2^2-p1^2*D^2;
```

   Para probar su código, establezca los parámetros $L_1 = 2, L_2 = L_3 = \sqrt{2}, \gamma = \pi/2, p_1 = p_2 = p_3 = \sqrt{5}$ a partir de la figura 1.15. Después, al sustituir $\theta = -\pi/4$ o $\theta = \pi/4$, correspondientes a las figuras 1.15 (a y b), respectivamente, debe obtenerse $f(\theta) = 0$.

2. Grafique $f(\theta)$ en $[-\pi, \pi]$. Puede utilizar el símbolo @ como se describe en el apéndice B.5 para asignar un apuntador de función a su archivo de función para el comando de graficación. También quizá tenga que preceder las operaciones aritméticas con el carácter "." a fin de vectorizar las operaciones, tal como se explica en el apéndice B.2. Como una comprobación de su trabajo, verifique que haya raíces en $\pm\pi/4$.

3. Reproduzca la figura 1.15. Los comandos de MATLAB

```
>> plot([u1 u2 u3 u1],[v1 v2 v3 v1],'r'); hold on
>> plot([0 x1 x2],[0 0 y2],'bo')
```

   trazarán un triángulo rojo con vértices `(u1, v1)`, `(u2, v2)`, `(u3, v3)` y colocarán pequeños círculos en los puntos de anclaje de los puntales `(0,0)`, `(0, x1)`, `(x2, y2)`. Además, dibuje los puntales.

4. Resuelva el problema directo de cinemática para la plataforma Stewart plana especificada por $x_1 = 5, (x_2, y_2) = (0, 6), L_1 = L_3 = 3, L_2 = 3\sqrt{2}, \gamma = \pi/4, p_1 = p_2 = 5, p_3 = 3$. Comience con la graficación de $f(\theta)$. Utilice un solucionador de ecuaciones para encontrar las cuatro posiciones

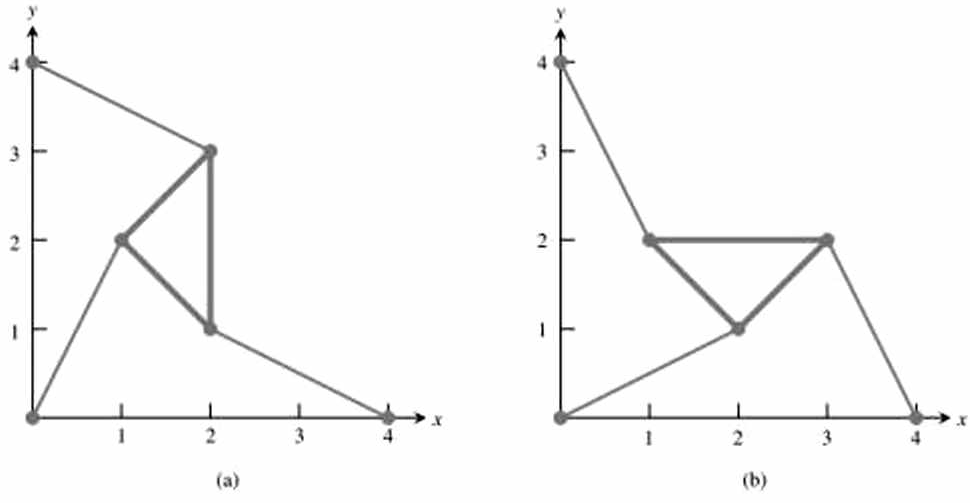


Figura 1.15 Dos posiciones de la plataforma Stewart plana con longitudes de brazo idénticas. Cada posición corresponde a una solución de (1,38) con longitudes de puntal $p_1 = p_2 = p_3 = \sqrt{5}$. La forma del triángulo está definida por $L_1 = 2$, $L_2 = L_3 = \sqrt{2}$, $\gamma = \pi/2$.

y grafíquelas. Compruebe sus respuestas verificando que p_1, p_2 y p_3 son las longitudes de los puntales en su gráfica.

5. Cambie la longitud de puntal a $p_2 = 7$ y vuelva a resolver el problema. Para estos parámetros existen seis posiciones.

6. Encuentre una longitud de puntal p_2, con el resto de los parámetros como en el paso 4, de modo que sólo haya dos posiciones.

7. Calcule los intervalos en p_2, con el resto de los parámetros como en el paso 4, de modo que haya 0, 2, 4 y 6 posiciones, respectivamente.

8. Deduzca o busque las ecuaciones que representan la cinemática directa de la plataforma Stewart en tres dimensiones, con seis grados de libertad. Escriba un programa en MATLAB y demuestre su utilidad para resolver la cinemática directa. Consulte en Merlet [2000] una interesante introducción a los brazos prismáticos de robots y plataformas. ✔

Software y lecturas adicionales

Existen muchos algoritmos para localizar las soluciones de ecuaciones no lineales. Los algoritmos lentos, pero siempre convergentes, como el método de bisección contrastan con las rutinas de una convergencia más rápida, pero sin garantías de convergencia, incluido el método de Newton y sus variantes. Los solucionadores de ecuaciones también pueden dividirse en dos grupos, dependiendo de si requieren información derivada de la ecuación o no. El método de bisección, el método de la secante y la interpolación cuadrática inversa son ejemplos de métodos que requieren sólo una caja negra que proporcione un valor de la función para una entrada dada, mientras que el método de Newton requiere derivadas. El método de Brent es un híbrido que combina los mejores aspectos de los algoritmos lentos y rápidos, y no requiere cálculos derivados. Por esta razón, se usa con frecuencia como solucionador de ecuaciones con propósito general y se incluye en muchos paquetes de software grandes.

El comando `fzero` de MATLAB implementa el método de Brent y sólo requiere un intervalo inicial o una estimación inicial como entrada. El programa ZBREN de IMSL, la rutina c05adc de NAG, y el programa fzero.f de `netlib` FORTRAN se basan en este enfoque básico.

El comando `roots` de MATLAB busca todas las raíces de un polinomio con un enfoque totalmente diferente, calcula los valores propios de la matriz adjunta, construida para tener valores propios iguales a todas las raíces del polinomio.

Otros algoritmos citados con frecuencia se basan en el método de Muller y el método de Laguerre, que, bajo las condiciones adecuadas, es cúbicamente convergente. Para mayores detalles, consulte los textos clásicos sobre la resolución de ecuaciones: Traub [1964], Ostrowski [1966], y Householder [1970].

CAPÍTULO

2

Sistemas de ecuaciones

Las leyes físicas rigen todas las estructuras ingenieriles, desde los rascacielos y los puentes hasta los trampolines y los dispositivos médicos. Las cargas estáticas y dinámicas hacen que los materiales se deformen, o flexionen. Los modelos matemáticos de flexión son herramientas básicas en el entorno de trabajo de un ingeniero estructural. El grado en que una estructura se flexiona bajo una carga depende de la rigidez del material, medida mediante su módulo de Young. La competencia entre el esfuerzo y la rigidez se modela por medio de una ecuación diferencial que, después de una discretización, se reduce a un sistema de ecuaciones lineales que debe resolverse.

Para aumentar la exactitud, se utiliza una discretización fina, la cual ocasiona que el sistema de ecuaciones lineales sea grande y, por lo regular, disperso. Los métodos de eliminación gaussiana (o eliminación gaussiana) son eficientes para las matrices de tamaño moderado, pero para los sistemas grandes y dispersos se requieren algoritmos iterativos especiales.

Comprobación en la realidad En la página 102 se estudian los métodos de solución aplicables al modelo de Euler-Bernoulli para vigas ancladas y en voladizo.

En el capítulo anterior se estudiaron los métodos para resolver una sola ecuación de una sola variable. En este capítulo se considera el problema de resolver varias ecuaciones simultáneas con varias variables. Se prestará mucha atención al caso en el que el número de ecuaciones es igual al número de variables desconocidas.

La eliminación gaussiana es una herramienta eficaz para los sistemas de ecuaciones lineales con un tamaño razonable. El capítulo comienza con el desarrollo de versiones eficientes y estables de esta famosa técnica. Más adelante en el capítulo se hablará sobre los métodos iterativos, requeridos para los sistemas muy grandes. Por último, se desarrollan métodos para los sistemas de ecuaciones no lineales.

2.1 ELIMINACIÓN GAUSSIANA

Considere el sistema

$$\begin{aligned} x + y &= 3 \\ 3x - 4y &= 2. \end{aligned} \tag{2.1}$$

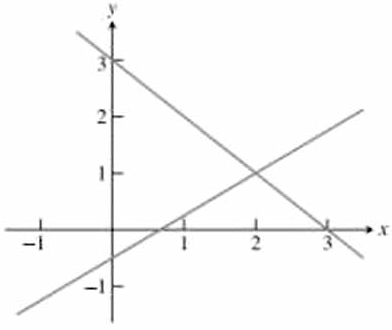


Figura 2.1 Solución geométrica de un sistema de ecuaciones. Cada ecuación de (2.1) corresponde a una recta en el plano. El punto de intersección es la solución.

Un sistema de dos ecuaciones con dos incógnitas puede considerarse en términos del álgebra o bien de la geometría. Desde el punto de vista geométrico, cada ecuación lineal representa una recta en el plano xy, como se muestra en la figura 2.1. El punto $x = 2$, $y = 1$ en el que las rectas se cruzan satisface ambas ecuaciones y es la solución que se está buscando.

El punto de vista geométrico es muy útil para visualizar las soluciones de los sistemas, pero para calcular la solución con una gran precisión es necesario regresar al álgebra. El método conocido como la eliminación gaussiana es una manera eficaz de resolver n ecuaciones con n incógnitas. En las siguientes secciones, se explorarán las implementaciones de la eliminación gaussiana que mejor funcionan para los problemas típicos.

2.1.1 Eliminación gaussiana simple

Se iniciará con la descripción de la forma más sencilla de la eliminación gaussiana. De hecho, es tan sencilla que no se garantiza llegar hasta su terminación, y mucho menos encontrar una solución precisa. Las modificaciones necesarias para mejorar el método "simple" se presentarán a partir de la siguiente sección.

Existen tres operaciones útiles que pueden aplicarse a un sistema de ecuaciones lineales para generar un sistema equivalente, es decir, un sistema que tenga las mismas soluciones. Estas operaciones son las siguientes:

(1) Intercambiar una ecuación por otra.
(2) Sumar o restar un múltiplo de una ecuación de otra.
(3) Multiplicar una ecuación por una constante diferente de cero.

Para la ecuación (2.1), es posible restar 3 veces la primera ecuación de la segunda a fin de eliminar la variable x de la segunda ecuación. Si se resta $3 \cdot [x + y = 3]$ de la segunda ecuación, queda el sistema

$$\begin{aligned} x + y &= 3 \\ -7y &= -7. \end{aligned} \tag{2.2}$$

A partir de la ecuación inferior, puede "resolverse hacia atrás" hasta encontrar una solución completa, como en

$$-7y = -7 \longrightarrow y = 1$$

y

$$x + y = 3 \longrightarrow x + (1) = 3 \longrightarrow x = 2.$$

Por lo tanto, la solución de (2.1) es $(x, y) = (2, 1)$.

El mismo trabajo de eliminación puede hacerse en ausencia de variables al escribir el sistema en forma de tabla:

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 3 & -4 & 2 \end{array}\right] \longrightarrow \begin{array}{c}\text{reste } 3 \times \text{renglón } 1 \\ \text{del renglón } 2\end{array} \longrightarrow \left[\begin{array}{cc|c} 1 & 1 & 3 \\ 0 & -7 & -7 \end{array}\right]. \tag{2.3}$$

La ventaja de la forma de tabla es que, durante la eliminación, las variables se ocultan. Cuando el arreglo cuadrado del lado izquierdo de la tabla es "triangular", es posible resolver hacia atrás para obtener la solución, comenzando en la parte inferior.

► EJEMPLO 2.1 Aplique la eliminación gaussiana en forma de tabla para el sistema de tres ecuaciones con tres incógnitas:

$$\begin{aligned} x + 2y - z &= 3 \\ 2x + y - 2z &= 3 \\ -3x + y + z &= -6. \end{aligned} \tag{2.4}$$

Lo anterior se escribe en forma de tabla como

$$\left[\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 2 & 1 & -2 & 3 \\ -3 & 1 & 1 & -6 \end{array}\right]. \tag{2.5}$$

Se requieren dos pasos para eliminar la columna 1:

$$\left[\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 2 & 1 & -2 & 3 \\ -3 & 1 & 1 & -6 \end{array}\right] \longrightarrow \begin{array}{c}\text{resta } 2 \times \text{renglón } 1 \\ \text{del renglón } 2\end{array} \longrightarrow \left[\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 0 & -3 & 0 & -3 \\ -3 & 1 & 1 & -6 \end{array}\right]$$

$$\longrightarrow \begin{array}{c}\text{resta } -3 \times \text{renglón } 1 \\ \text{del renglón } 3\end{array} \longrightarrow \left[\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 0 & -3 & 0 & -3 \\ 0 & 7 & -2 & 3 \end{array}\right]$$

y un paso más para eliminar la columna 2:

$$\left[\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 0 & -3 & 0 & -3 \\ 0 & 7 & -2 & 3 \end{array}\right] \longrightarrow \begin{array}{c}\text{resta } -\frac{7}{3} \times \text{renglón } 2 \\ \text{del renglón } 3\end{array} \longrightarrow \left[\begin{array}{ccc|c} 1 & 2 & -1 & 3 \\ 0 & -3 & 0 & -3 \\ 0 & 0 & -2 & -4 \end{array}\right]$$

De regreso a las ecuaciones

$$\begin{aligned} x + 2y - z &= 3 \\ -3y &= -3 \\ -2z &= -4, \end{aligned} \tag{2.6}$$

pueden despejarse las variables

$$\begin{aligned} x &= 3 - 2y + z \\ -3y &= -3 \\ -2z &= -4 \end{aligned} \tag{2.7}$$

y resolver para z, y y x, en ese orden. La parte final se denomina **sustitución hacia atrás** o **resolviendo hacia atrás** porque, después de la eliminación, las ecuaciones se resuelven con facilidad de abajo hacia arriba. La solución es $x = 3, y = 1, z = 2$. ◄

2.1.2 Conteo de operaciones

En esta sección se hace un conteo aproximado de operaciones para las dos partes de la eliminación gaussiana: el paso de eliminación y el paso de sustitución hacia atrás. Con el fin de lograr esto, será útil extender al caso general las operaciones que se llevaron a cabo en los dos ejemplos anteriores. Para empezar, recuerde dos hechos acerca de la suma de números enteros.

LEMA 2.1 Para cualquier entero positivo n, (a) $1 + 2 + 3 + 4 + \cdots + n = n(n + 1)/2$ y (b) $1^2 + 2^2 + 3^2 + 4^2 + \cdots + n^2 = n(n + 1)(2n + 1)/6$. ■

La forma general de la tabla para n ecuaciones con n incógnitas es

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right].$$

Para llevar a cabo la etapa de eliminación, es necesario poner ceros en el triángulo inferior, utilizando las operaciones por renglón permitidas.

El paso de eliminación puede escribirse como el ciclo

```
for j = 1 : n-1
  eliminate column j
end
```

donde, "eliminate column j (eliminar la columna j)", significa "usar operaciones por renglón para poner un cero en cada lugar por debajo de la diagonal principal, que son los lugares a_{j+1}, a_{j+2}, j, ..., a_{nj}". Por ejemplo, para realizar la eliminación en la columna 1, deben ponerse ceros en $a_{21}, \ldots, a_{n1}$. Esto puede escribirse como el siguiente ciclo dentro del ciclo anterior:

```
for j = 1 : n-1
  for i = j+1 : n
    eliminate entry a(i,j)
  end
end
```

Lo que resta es llenar el paso interior del doble ciclo, para aplicar una operación por renglón que convierta la entrada a_{ij} en cero. Por ejemplo, la primera entrada a eliminar es la entrada a_{21}. Para lograr esto, se resta a_{21}/a_{11} veces el renglón 1 del renglón 2, suponiendo que $a_{11} \neq 0$. Es decir, los dos primeros renglones cambian de

$$\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \end{array}$$

a

$$\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ 0 & a_{22} - \dfrac{a_{21}}{a_{11}}a_{12} & \cdots & a_{2n} - \dfrac{a_{21}}{a_{11}}a_{1n} & b_2 - \dfrac{a_{21}}{a_{11}}b_1 \end{array}.$$

Si se cuentan las operaciones, esto requiere una división (para encontrar el multiplicador a_{21}/a_{11}), además de n multiplicaciones y n sumas. La operación por renglón usada para eliminar la entrada a_{i1} de la primera columna, es decir,

$$\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & a_{i2} - \dfrac{a_{i1}}{a_{11}}a_{12} & \cdots & a_{in} - \dfrac{a_{i1}}{a_{11}}a_{1n} & b_i - \dfrac{a_{i1}}{a_{11}}b_1 \end{array}$$

requiere un número similar de operaciones.

El procedimiento antes descrito se realiza mientras el número a_{11} es diferente de cero. Este número y los demás a_{ii}, que a la larga se convierten en divisores en la eliminación gaussiana, se denominan **pivotes**. Un pivote igual a cero hará que el algoritmo se detenga, como se ha explicado hasta ahora. Este problema se ignorará por el momento y se abordará con más cuidado en la sección 2.4.

De regreso al conteo de operaciones, observe que la eliminación de cada entrada a_{i1} en la primera columna utiliza una división, n multiplicaciones y n adiciones/sustracciones, o $2n + 1$ operaciones en total. La colocación de ceros en la primera columna requiere una repetición de estas $2n + 1$ operaciones un total de $n - 1$ veces.

Después de eliminar la primera columna, se utiliza el pivote a_{22} para eliminar la segunda columna de la misma manera y después de eso las columnas restantes. Por ejemplo, la operación por renglón usada para eliminar la entrada a_{ij} es

$$\begin{array}{cccccccc} 0 & 0 & a_{jj} & a_{j,j+1} & \cdots & a_{jn} & | & b_j \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & | & \vdots \\ 0 & 0 & 0 & a_{i,j+1} - \dfrac{a_{ij}}{a_{jj}} a_{j,j+1} & \cdots & a_{in} - \dfrac{a_{ij}}{a_{jj}} a_{jn} & | & b_i - \dfrac{a_{ij}}{a_{jj}} b_j. \end{array}$$

En la notación empleada aquí, a_{22}, por ejemplo, se refiere al número modificado en esa posición después de la eliminación de la columna 1, que no es el a_{22} original. La operación por renglón para eliminar a_{ij} requiere una división, $n - j + 1$ multiplicaciones, y $n - j + 1$ sumas/restas.

Al insertar este paso en el mismo ciclo doble, resulta

```
for j = 1 : n-1
  if abs(a(j,j))<eps; error('pivote cero encontrado'); end
  for i = j+1 : n
    mult = a(i,j)/a(j,j);
    for k = j+1:n
      a(i,k) = a(i,k) - mult*a(j,k);
    end
    b(i) = b(i) - mult*b(j);
  end
end
```

Ahora son necesarios dos comentarios sobre este fragmento de código: en primer lugar, cuando se pide al índice k que pase de j a n se coloca un cero en la ubicación a_{ij}; sin embargo, al pasar de $j + 1$ a n es una codificación más eficiente. Ésta no pondrá un cero en la entrada a_{ij}, ¡que era la entrada que se trataba de eliminar! Aunque esto parece ser un error, observe que nunca se regresa a esta entrada en el resto de la eliminación gaussiana o en el proceso de sustitución hacia atrás. Así que en realidad la colocación de un cero en esa posición representa un paso perdido desde el punto de vista de la eficiencia. En segundo lugar, se pide que el código se detenga, con el comando `error` de MATLAB, si se encuentra un pivote cero. Como se mencionó, esta posibilidad se considerará con mayor seriedad cuando se analicen los intercambios de renglón en la sección 2.4.

Puede hacerse un conteo total de operaciones para el paso de eliminación de la eliminación gaussiana. La eliminación de cada a_{ij} requiere el siguiente número de operaciones, incluyendo las divisiones, las multiplicaciones y las sumas/restas:

$$\begin{bmatrix} 0 & & & & & & \\ 2n+1 & 0 & & & & & \\ 2n+1 & 2(n-1)+1 & 0 & & & & \\ 2n+1 & 2(n-1)+1 & 2(n-2)+1 & 0 & & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & & \\ \vdots & \vdots & \vdots & & & & \\ 2n+1 & 2(n-1)+1 & 2(n-2)+1 & \cdots & 2(3)+1 & 0 & \\ 2n+1 & 2(n-1)+1 & 2(n-2)+1 & \cdots & 2(3)+1 & 2(2)+1 & 0 \end{bmatrix}.$$

Resulta conveniente sumar las operaciones en orden inverso a cómo se aplican. Empezando por la derecha, la sumatoria total de las operaciones es

$$\begin{aligned}\sum_{j=1}^{n-1}\sum_{i=1}^{j} 2(j+1)+1 &= \sum_{j=1}^{n-1} 2j(j+1)+j \\ &= 2\sum_{j=1}^{n-1} j^2 + 3\sum_{j=1}^{n-1} j = 2\frac{(n-1)n(2n-1)}{6} + 3\frac{(n-1)n}{2} \\ &= (n-1)n\left[\frac{2n-1}{3}+\frac{3}{2}\right] = \frac{n(n-1)(4n+7)}{6} \\ &= \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n,\end{aligned}$$

donde se ha aplicado el lema 2.1.

Conteo de operaciones para el paso de eliminación en la eliminación gaussiana

El paso de eliminación para un sistema de n ecuaciones con n variables puede completarse en $\frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n$ operaciones.

En general, el conteo exacto de operaciones es menos importante que las estimaciones del orden de magnitud, puesto que los detalles de la implementación en los diferentes procesadores de computadora difieren. El punto principal es que el número de operaciones es proporcional al tiempo de ejecución del algoritmo. Con frecuencia se hace la aproximación de $\frac{2}{3}n^3$ operaciones de eliminación, que es una aproximación razonablemente exacta cuando n es grande.

Después de completar la eliminación, la tabla es triangular superior:

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ 0 & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a_{nn} & b_n \end{array}\right].$$

En forma de ecuación,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{nn}x_n &= b_n, \end{aligned} \tag{2.8}$$

donde, de nuevo, las a_{ij} se refieren a las entradas modificadas, no a las originales. Para completar el cálculo de la solución x debe llevarse a cabo el paso de sustitución hacia atrás, que es tan sólo una reescritura de (2.8):

$$\begin{aligned} x_1 &= \frac{b_1 - a_{12}x_2 - \cdots - a_{1n}x_n}{a_{11}} \\ x_2 &= \frac{b_2 - a_{23}x_3 - \cdots - a_{2n}x_n}{a_{22}} \\ &\vdots \\ x_n &= \frac{b_n}{a_{nn}}. \end{aligned} \tag{2.9}$$

ANOTACIÓN

Resumen El conteo de operaciones muestra que la solución directa de n ecuaciones con n incógnitas mediante la eliminación gaussiana es un proceso $O(n^3)$. Éste es un factor útil en la estimación del tiempo necesario para la resolución de grandes sistemas. Por ejemplo, si se desea estimar el tiempo necesario para resolver un sistema de ecuaciones con $n = 500$ en una computadora determinada, podría obtenerse una estimación razonable para resolver un sistema de ecuaciones con $n = 50$ y luego escalar el tiempo transcurrido por $10^3 = 1000$.

Debido a la forma triangular de los coeficientes diferentes de cero en las ecuaciones, se comienza desde abajo y se avanza hacia la ecuación superior. De esta manera, cada x_i requerida se conoce cuando es necesaria para calcular la siguiente x_i. Al contar las operaciones se tiene

$$1 + 3 + 5 + \cdots + (2n-1) = \sum_{i=1}^{n} 2i - 1 = 2\sum_{i=1}^{n} i - \sum_{i=1}^{n} 1 = 2\frac{n(n+1)}{2} - n = n^2.$$

En la sintaxis de MATLAB, el paso de sustitución hacia atrás es

```
for i = n : -1 : 1
  for j = i+1 : n
    b(i) = b(i) - a(i,j)*x(j);
  end
  x(i) = b(i)/a(i,i);
end
```

Conteo de operaciones para el paso de sustitución hacia atrás de la eliminación gaussiana

El paso de sustitución hacia atrás para un sistema triangular de n ecuaciones con n variables puede completarse en n^2 operaciones.

Los dos conteos de operación, tomados en conjunto, muestran que la eliminación gaussiana se compone de dos partes desiguales: el paso de eliminación relativamente complicado y el paso de sustitución hacia atrás relativamente sencillo. Si no se toman en cuenta los términos de orden inferior en las expresiones para el número de multiplicaciones/divisiones, se observa que la eliminación necesita el orden de $2n^3/3$ operaciones y que la sustitución hacia atrás requiere el orden de n^2.

Con frecuencia se utilizará la terminología abreviada de "la gran O" que significa "del orden de", para decir que la eliminación es un algoritmo $O(n^3)$ y que la sustitución hacia atrás es $O(n^2)$.

Este uso implica que el énfasis está en las n grandes, donde las potencias menores de n se vuelven en comparación insignificantes. Por ejemplo, si $n = 100$, sólo alrededor del 1 por ciento o menos del tiempo de cálculo de la eliminación gaussiana se ocupa en el paso de sustitución hacia atrás. En general, la eliminación gaussiana requiere $2n^3/3 + n^2 \approx 2n^3/3$ operaciones. En otras palabras, para una n grande, los términos de orden inferior en el conteo de complejidad no tendrán un efecto importante en la estimación del tiempo de ejecución del algoritmo y pueden omitirse si sólo se requiere un tiempo estimado.

► EJEMPLO 2.2 Estime el tiempo requerido para llevar a cabo la sustitución hacia atrás en un sistema de 500 ecuaciones con 500 incógnitas, en una computadora donde la eliminación tarda 1 segundo.

Como se acaba de establecer que la eliminación consume mucho más tiempo que la sustitución hacia atrás, la respuesta será una fracción de un segundo. Si se usa el número aproximado $2(500)^3/3$ para el número de operaciones de multiplicación/división durante el paso de eliminación, y $(500)^2$ para el paso de sustitución hacia atrás, se estima que el tiempo para la sustitución hacia atrás será

$$\frac{(500)^2}{2(500)^3/3} = \frac{3}{2(500)} = .003 \text{ seg.}$$ ◄

El ejemplo muestra dos puntos: (1) A menudo en los conteos de operaciones, las potencias más pequeñas de n pueden omitirse de manera segura y (2) las dos partes de la eliminación gaussiana

pueden ser muy desiguales en el tiempo de funcionamiento requerido (el tiempo de cálculo total es de 1.003 segundos, la mayor parte del cual sería utilizado para el paso de eliminación). El siguiente ejemplo muestra un tercer punto. Aunque en ocasiones el tiempo de sustitución hacia atrás es insignificante, puede ser un factor a tomar en cuenta en un cálculo importante.

► **EJEMPLO 2.3** En una computadora en particular, la sustitución hacia atrás de una matriz triangular de 5000 × 5000 tarda 0.1 segundos. Calcule el tiempo necesario para resolver un sistema general de 3000 ecuaciones con 3000 incógnitas mediante la eliminación gaussiana.

La computadora puede llevar a cabo $(5000)^2$ operaciones en 0.1 segundos, o $(5000)^2(10) = 2.5 \times 10^8$ operaciones/segundo. Para resolver un sistema general (no triangular) requiere aproximadamente $2(3000)^3/3$ operaciones, las cuales pueden realizarse en

$$\frac{2(3000)^3/3}{(5000)^2(10)} \approx 72 \text{ seg.}$$ ◄

2.1 Ejercicios

1. Utilice la eliminación gaussiana para resolver los sistemas:

$$\text{(a)} \quad \begin{aligned} 2x - 3y &= 2 \\ 5x - 6y &= 8 \end{aligned} \qquad \text{(b)} \quad \begin{aligned} x + 2y &= -1 \\ 2x + 3y &= 1 \end{aligned} \qquad \text{(c)} \quad \begin{aligned} -x + y &= 2 \\ 3x + 4y &= 15 \end{aligned}$$

2. Utilice la eliminación gaussiana para resolver los sistemas:

$$\text{(a)} \quad \begin{aligned} 2x - 2y - z &= -2 \\ 4x + y - 2z &= 1 \\ -2x + y - z &= -3 \end{aligned} \qquad \text{(b)} \quad \begin{aligned} x + 2y - z &= 2 \\ 3y + z &= 4 \\ 2x - y + z &= 2 \end{aligned} \qquad \text{(c)} \quad \begin{aligned} 2x + y - 4z &= -7 \\ x - y + z &= -2 \\ -x + 3y - 2z &= 6 \end{aligned}$$

3. Resuelva mediante la sustitución hacia atrás:

$$\text{(a)} \quad \begin{aligned} 3x - 4y + 5z &= 2 \\ 3y - 4z &= -1 \\ 5z &= 5 \end{aligned} \qquad \text{(b)} \quad \begin{aligned} x - 2y + z &= 2 \\ 4y - 3z &= 1 \\ -3z &= 3 \end{aligned}$$

4. Resuelva los problemas en forma de tabla

$$\text{(a)} \quad \left[\begin{array}{ccc|c} 3 & -4 & -2 & 3 \\ 6 & -6 & 1 & 2 \\ -3 & 8 & 2 & -1 \end{array}\right] \qquad \text{(b)} \quad \left[\begin{array}{ccc|c} 2 & 1 & -1 & 2 \\ 6 & 2 & -2 & 8 \\ 4 & 6 & -3 & 5 \end{array}\right]$$

5. Utilice el conteo aproximado de operaciones $2n^3/3$ para la eliminación gaussiana a fin de estimar cuánto tiempo se requiere para resolver n ecuaciones con n incógnitas si n se triplica.

6. Suponga que su computadora completa una sustitución hacia atrás de 5000 ecuaciones en 0.005 segundos. Utilice los conteos aproximados de operaciones n^2 para la sustitución hacia atrás y $2n^3/3$ para la eliminación, a fin de estimar cuánto tiempo tomará hacer una eliminación gaussiana completa de este tamaño. Redondee su respuesta al segundo más cercano.

7. Suponga que una determinada computadora requiere 0.002 segundos para completar la sustitución hacia atrás en una ecuación matricial triangular superior de 4000 × 4000. Calcule el tiempo necesario para resolver un sistema general de 9000 ecuaciones con 9000 incógnitas. Redondee su respuesta al segundo más cercano.

8. Si un sistema de 3000 ecuaciones con 3000 incógnitas puede resolverse mediante la eliminación gaussiana en 5 segundos usando una computadora determinada, ¿cuántas sustituciones hacia atrás del mismo tamaño pueden hacerse por segundo?

2.1 Problemas de computadora

1. Reúna los fragmentos de código presentados en esta sección para crear un programa en MATLAB que realice la eliminación gaussiana "simple" (es decir, sin permitir el intercambio de renglones). Úselo para resolver los sistemas del ejercicio 2.
2. Sea H la matriz de Hilbert de $n \times n$, cuya entrada (i, j) es $1/(i + j - 1)$. Utilice el programa en MATLAB del problema de computadora 1 para resolver $Hx = b$, donde b es un vector que contiene sólo unos, para (a) $n = 2$ (b) $n = 5$ (c) $n = 10$.

2.2 LA FACTORIZACIÓN LU

Si se lleva la idea de la forma tabular un paso adelante, se llega a la forma matricial de un sistema de ecuaciones. A largo plazo, la forma matricial ahorrará tiempo por la simplificación de los algoritmos y su análisis.

2.2.1 Forma matricial de la eliminación gaussiana

El sistema (2.1) puede escribirse como $Ax = b$ en forma de matriz, o

$$\begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}. \tag{2.10}$$

Por lo general, se indicará la **matriz de coeficientes** con A y el **vector del lado derecho** con b. En la forma matricial de los sistemas de ecuaciones, se interpreta x como un vector columna y Ax como la multiplicación matriz-vector. Se desea encontrar una x de tal forma que el vector Ax sea igual al vector b. Por supuesto, esto es equivalente a que Ax y b coincidan en todos sus componentes, que es exactamente lo que requiere el sistema original (2.1).

La ventaja de escribir los sistemas de ecuaciones en forma matricial es que pueden utilizarse operaciones con matrices, como la multiplicación de matrices, a fin de dar seguimiento de los pasos de la eliminación gaussiana. La factorización LU es una representación matricial de esta eliminación. Consiste en escribir la matriz de coeficientes A como un producto de una matriz triangular inferior L y una matriz triangular superior U. La factorización LU es una versión de la eliminación gaussiana surgida de una larga tradición en ciencia e ingeniería: fragmentar un objeto complicado en partes más simples.

DEFINICIÓN 2.2 Una matriz L de $m \times n$ es **triangular inferior** si sus entradas satisfacen $l_{ij} = 0$ para $i < j$. Una matriz U de $m \times n$ es **triangular superior** si sus entradas satisfacen $u_{ij} = 0$ para $i > j$. ❐

► EJEMPLO 2.4 Encuentre la factorización LU para la matriz A (2.10).

Los pasos de la eliminación son los mismos que para la forma tabular estudiada anteriormente:

$$\begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} \xrightarrow[\text{del renglón 2}]{\text{resta 3} \times \text{renglón 1}} \begin{bmatrix} 1 & 1 \\ 0 & -7 \end{bmatrix} = U. \tag{2.11}$$

La diferencia es que ahora se almacena el multiplicador 3 utilizado en el paso de eliminación. Tenga en cuenta que se ha definido U como la matriz triangular superior que muestra el resultado de la eliminación gaussiana. Defina L como la matriz triangular inferior de 2×2 con unos en la diagonal principal y el multiplicador 3 en la ubicación (2,1):

$$\begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}.$$

A continuación, compruebe que

$$LU = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & -7 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} = A. \tag{2.12}$$

◄

En breve se analizará la razón por la que esto funciona, pero primero deben mostrarse los pasos con un ejemplo de 3×3.

► EJEMPLO 2.5 Encuentre la factorización LU de

$$A = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix}. \tag{2.13}$$

Ésta es la matriz de coeficientes del sistema (2.4). Los pasos de la eliminación proceden de la misma forma que antes:

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix} \xrightarrow[\text{del renglón 2}]{\text{resta 2} \times \text{renglón 1}} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ -3 & 1 & 1 \end{bmatrix}$$

$$\xrightarrow[\text{del renglón 3}]{\text{resta } -3 \times \text{renglón 1}} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 7 & -2 \end{bmatrix}$$

$$\xrightarrow[\text{del renglón 3}]{\text{resta } -\frac{7}{3} \times \text{renglón 2}} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} = U.$$

La matriz triangular inferior L está formada, como en el ejemplo anterior, al poner unos en la diagonal principal y los multiplicadores en el triángulo inferior, en las posiciones específicas que se utilizaron para la eliminación. Esto es,

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -\frac{7}{3} & 1 \end{bmatrix}. \tag{2.14}$$

Por ejemplo, observe que 2 es la entrada (2, 1) de L, porque fue el multiplicador utilizado para eliminar la entrada (2, 1) de A. Ahora verifique que

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -\frac{7}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix} = A. \tag{2.15}$$

◄

La razón de que este procedimiento proporcione la factorización LU se deriva de tres hechos sobre las matrices triangulares inferiores.

HECHO 1 Sea $L_{ij}(-c)$ la matriz triangular inferior, cuyas únicas entradas diferentes de cero son unos en la diagonal principal y $-c$ en la posición (i, j). Entonces $A \rightarrow L_{ij}(-c)A$ representa la operación por renglón "restar c veces el renglón j del renglón i".

Por ejemplo, la multiplicación por $L_{21}(-c)$ resulta en

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 0 & 0 \\ -c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$
$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} - ca_{11} & a_{22} - ca_{12} & a_{23} - ca_{13} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}. \qquad \square$$

HECHO 2 $L_{ij}(-c)^{-1} = L_{ij}(c)$.

Por ejemplo,

$$\begin{bmatrix} 1 & 0 & 0 \\ -c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Usando los hechos 1 y 2, es posible entender la factorización LU del ejemplo 2.4. Puesto que el paso de eliminación puede representarse mediante

$$L_{21}(-3)A = \begin{bmatrix} 1 & 0 \\ -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & -7 \end{bmatrix},$$

se pueden multiplicar ambos lados de la izquierda por $L_{21}(-3)^{-1}$ para obtener

$$A = \begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & -7 \end{bmatrix}, \qquad \square$$

que es la factorización LU de A.

Para manejar matrices de $n \times n$ para $n > 2$, se necesita un hecho más.

HECHO 3 Se cumple la siguiente ecuación de producto matricial.

$$\begin{bmatrix} 1 & & \\ c_1 & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ c_2 & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & c_3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ c_1 & 1 & \\ c_2 & c_3 & 1 \end{bmatrix}.$$

Este hecho permite reunir las L_{ij} inversas en una matriz, que se convierte en la L de la factorización LU. Para el ejemplo 2.5, esto equivale a

$$\begin{bmatrix} 1 & & \\ & 1 & \\ \frac{7}{3} & 1 & \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ 3 & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ -2 & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} = U$$
$$A = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ -3 & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & -\frac{7}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & & \\ 2 & 1 & \\ -3 & -\frac{7}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} = LU. \qquad (2.16)$$

$\square$

2.2.2 Sustitución hacia atrás con la factorización LU

Ahora que se ha expresado el paso de eliminación de la eliminación gaussiana como un producto matricial LU, ¿cómo puede traducirse el paso de sustitución hacia atrás? Más importante aún, ¿cómo puede obtenerse la solución x?

Una vez que se conocen L y U, el problema $Ax = b$ puede escribirse como $LUx = b$. Defina un nuevo vector "auxiliar" $c = Ux$. Entonces, la sustitución hacia atrás es un procedimiento de dos pasos:

(a) Resolver $Lc = b$ para c.
(b) Resolver $Ux = c$ para x.

Ambos pasos son directos puesto que L y U son matrices triangulares. Esto se demuestra con los dos ejemplos utilizados previamente.

► EJEMPLO 2.6 Resuelva el sistema (2.10), utilizando la factorización LU (2.12).

El sistema tiene factorización LU

$$\begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} = LU = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & -7 \end{bmatrix}$$

a partir de (2.12), y el lado derecho es $b = [3, 2]$. El paso (a) es

$$\begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix},$$

que corresponde al sistema

$$\begin{aligned} c_1 + 0c_2 &= 3 \\ 3c_1 + c_2 &= 2. \end{aligned}$$

Comenzando en la parte superior, las soluciones son $c_1 = 3, c_2 = -7$.

El paso (b) es

$$\begin{bmatrix} 1 & 1 \\ 0 & -7 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -7 \end{bmatrix},$$

que corresponde al sistema

$$\begin{aligned} x_1 + x_2 &= 3 \\ -7x_2 &= -7. \end{aligned}$$

Comenzando en la parte inferior, las soluciones son $x_2 = 1, x_1 = 2$. Lo anterior concuerda con el cálculo "clásico" de la eliminación gaussiana realizado antes. ◄

► EJEMPLO 2.7 Resuelva el sistema (2.4), utilizando la factorización LU (2.15).

El sistema tiene factorización LU

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & 1 & -2 \\ -3 & 1 & 1 \end{bmatrix} = LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -\frac{7}{3} & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

a partir de (2.15), y $b = (3, 3, -6)$. El paso de $Lc = b$ es

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -\frac{7}{3} & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ -6 \end{bmatrix},$$

que corresponde al sistema

$$\begin{aligned} c_1 &= 3 \\ 2c_1 + c_2 &= 3 \\ -3c_1 - \frac{7}{3}c_2 + c_3 &= -6. \end{aligned}$$

Comenzando en la parte superior, las soluciones son $c_1 = 3, c_2 = -3, c_3 = -4$.

El paso $Ux = c$ es

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \\ -4 \end{bmatrix},$$

que se corresponde al sistema

$$\begin{aligned} x_1 + 2x_2 - x_3 &= 3 \\ -3x_2 &= -3 \\ -2x_3 &= -4, \end{aligned}$$

y se resuelve de abajo hacia arriba para obtener $x = [3, 1, 2]$. ◀

2.2.3 Complejidad de la factorización LU

Ahora que se ha analizado el "cómo" de la factorización LU, a continuación se presentan algunos conceptos acerca del "porqué". La eliminación gaussiana clásica involucra tanto a A como a b en el paso de eliminación dentro del cálculo. Por mucho, ésta es la parte más compleja del proceso, como ya se ha visto. Ahora, suponga que es necesario resolver una serie de problemas distintos con la misma A y diferente b. Es decir, se presenta la serie de problemas

$$\begin{aligned} Ax &= b_1 \\ Ax &= b_2 \\ &\vdots \\ Ax &= b_k \end{aligned}$$

con varios vectores b_i del lado derecho. La eliminación gaussiana clásica requiere aproximadamente $2kn^3/3$ operaciones, donde A es una matriz de $n \times n$, puesto que debe comenzarse desde el principio para cada problema. Por otro lado, con el enfoque LU las b_i $(i = 1,\ldots, k)$ del lado derecho no entran en los cálculos hasta que se termina la eliminación (la factorización $A = LU$). Al aislar los cálculos que involucran a A de b_i, puede resolverse la serie anterior de ecuaciones con una sola eliminación, seguida de dos sustituciones hacia atrás ($Lc = b_i$, $Ux = c$) para cada nueva b_i. El número aproximado de operaciones con el enfoque LU es, por lo tanto, $2n^3/3 + 2kn^2$. Cuando n^2 es pequeña en comparación con n^3 (es decir, cuando n es grande), ésta es una diferencia significativa.

Incluso cuando $k = 1$, no hay algún trabajo de cálculo adicional que realice el enfoque $A = LU$, en comparación con la eliminación gaussiana clásica. Aunque parece haber una sustitución

ANOTACIÓN

Resumen La razón principal de la existencia del enfoque de factorización LU para la eliminación gaussiana es la ubicuidad de problemas con la forma $Ax = b_1, Ax = b_2 \ldots$ Con frecuencia, A es lo que se denomina una matriz estructural, dependiendo sólo del diseño de un sistema mecánico o dinámico, y b corresponde a un "vector de carga". En ingeniería estructural, el vector de carga proporciona las fuerzas aplicadas en diferentes puntos de la estructura. Entonces, la solución x corresponde a los esfuerzos sobre la estructura inducidos por esa combinación particular de cargas. La solución repetida de $Ax = b$ para las diversas b se utiliza para probar los posibles diseños estructurales. En la comprobación en la realidad 2 se presenta este análisis para la carga de una viga.

hacia atrás extra que no forma parte de la eliminación gaussiana clásica, estos cálculos "extras" remplazan justo los cálculos ahorrados durante la eliminación debido a que la b del lado derecho estaba ausente.

Si todas las b_i estuvieran disponibles al principio, podrían resolverse los k problemas de manera simultánea en el mismo número de operaciones. Sin embargo, en las aplicaciones típicas, se solicita resolver algunos de los problemas $Ax = b_i$, antes de que otras b_i estén disponibles. El enfoque LU permite un manejo eficiente de todos los problemas presentes y futuros que involucren la misma matriz de coeficientes A.

► **EJEMPLO 2.8** Suponga que se necesita un segundo para factorizar la matriz A de 300×300 como $A = LU$. ¿Cuántos problemas $Ax = b_1, \ldots, Ax = b_k$ pueden resolverse en el siguiente segundo?

Las dos sustituciones hacia atrás para cada b_i requieren un total de $2n^2$ operaciones. Por lo tanto, el número aproximado de b_i que puede manejarse por segundo es

$$\frac{\frac{2n^3}{3}}{2n^2} = \frac{n}{3} = 100.$$

◄

La factorización LU es un paso importante hacia delante en la búsqueda de una ejecución eficaz de la eliminación gaussiana. Lamentablemente, no toda matriz permite tal factorización.

► **EJEMPLO 2.9** Demuestre que $A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$ no tiene una factorización LU.

La factorización debe tener la forma

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix} \begin{bmatrix} b & c \\ 0 & d \end{bmatrix} = \begin{bmatrix} b & c \\ ab & ac + d \end{bmatrix}.$$

Al igualar los coeficientes se obtiene $b = 0$ y $ab = 1$, que es una contradicción. ◄

El hecho de que no todas las matrices tengan una factorización LU significa que se requiere más trabajo antes de poder declarar que la factorización LU es un algoritmo general para la eliminación gaussiana. El problema relacionado de dominancia se describe en la siguiente sección. En la sección 2.4 se presenta la factorización PA = LU, que da solución a ambos problemas.

2.2 Ejercicios

1. Encuentre la factorización LU de las matrices dadas. Compruebe mediante la multiplicación de matrices.

$$\text{(a)} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \quad \text{(c)} \begin{bmatrix} 3 & -4 \\ -5 & 2 \end{bmatrix}$$

2. Encuentre la factorización LU de las matrices dadas. Compruebe mediante la multiplicación de matrices.

$$\text{(a)} \begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 4 & 2 & 0 \\ 4 & 4 & 2 \\ 2 & 2 & 3 \end{bmatrix} \quad \text{(c)} \begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 3 & 4 & 4 \\ 0 & 2 & 1 & -1 \end{bmatrix}$$

3. Resuelva el sistema encontrando la factorización LU y después realice la sustitución hacia atrás en dos pasos.

$$\text{(a)} \begin{bmatrix} 3 & 7 \\ 6 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -11 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 2 & 3 \\ 4 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

4. Resuelva el sistema encontrando la factorización LU y después realice la sustitución hacia atrás en dos pasos.

$$\text{(a)} \begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 4 & 2 & 0 \\ 4 & 4 & 2 \\ 2 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

5. Resuelva la ecuación $Ax = b$, donde

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 \\ 4 & 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{y} \quad b = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \end{bmatrix}.$$

6. Dada la matriz A de 1000×1000, su computadora puede resolver los 500 problemas $Ax = b_1, \ldots, Ax = b_{500}$ en exactamente un minuto, utilizando los métodos de factorización $A = LU$. ¿Cuánto de ese minuto su computadora estará trabajando en la factorización $A = LU$? Redondee su respuesta al segundo más cercano.

7. Suponga que su computadora puede resolver 1000 problemas del tipo $Ux = c$, donde U es una matriz triangular superior de 500×500, cada segundo. Calcule cuánto tiempo se tardará en resolver el problema completo de la matriz $Ax = b$ de 5000×5000. Dé su respuesta en minutos y segundos.

8. Suponga que su computadora puede resolver un sistema lineal $Ax = b$ de 2000×2000 en 0.1 segundos. Estime el tiempo requerido para resolver 100 sistemas de 8000 ecuaciones con 8000 incógnitas, con la misma matriz de coeficientes y utilizando el método de factorización LU.

9. Sea A una matriz de $n \times n$. Suponga que su computadora puede resolver 100 problemas $Ax = b_1, \ldots, Ax = b_{100}$ mediante el método LU en la misma cantidad de tiempo requerida para resolver el primer problema $Ax = b_0$. Estime n.

2.2 Problemas de computadora

1. Utilice los fragmentos de código para la eliminación gaussiana que se han presentado en la sección previa, con el fin de escribir un script de MATLAB que tome una matriz A como entrada y a L y U como salida. No se permiten intercambios renglones, el programa debe diseñarse para detenerse si encuentra un pivote cero. Verifique su programa al factorizar las matrices del ejercicio 2.

2. Añada la sustitución hacia atrás en dos pasos a su script del problema de computadora 1, y utilícelo para resolver los sistemas del ejercicio 4.

2.3 FUENTES DE ERROR

Como se ha descrito hasta ahora, en la eliminación gaussiana existen dos fuentes potenciales de error principales. El concepto de mal condicionamiento se refiere a la sensibilidad de la solución a los datos de entrada. Se analizará el número de condición, utilizando los conceptos de error hacia atrás y hacia delante del capítulo 1. Puede hacerse muy poco para evitar errores de cálculo en la solución de ecuaciones matriciales mal condicionadas, por lo que es importante tratar de reconocer y evitar las matrices mal condicionadas siempre que sea posible. La segunda fuente de error es la dominancia, que puede evitarse en la gran mayoría de los problemas mediante una simple corrección llamada pivoteo parcial, éste es el objeto de estudio de la sección 2.4.

A continuación se presenta el concepto de normas vectoriales y matriciales para medir el tamaño de los errores, que ahora son vectores. Se dará un énfasis especial a la norma denominada infinito.

2.3.1 Error de magnificación y número de condición

En el capítulo 1 se encontró que algunos problemas de resolución de ecuaciones muestran una gran diferencia entre el error hacia atrás y hacia delante. Lo mismo se cumple para los sistemas de ecuaciones lineales. Con el fin de cuantificar los errores, se iniciará con una definición de la norma infinito de un vector.

DEFINICIÓN 2.3 La **norma infinito**, o **norma máximo** del vector $x = (x_1, \ldots, x_n)$ es $\|x\|_\infty = \max |x_i|$, $i = 1, \ldots, n$; es decir, el máximo de los valores absolutos de las componentes de x. ❐

Los errores hacia atrás y hacia delante se especifican en analogía con la definición 1.8. El error hacia atrás representa diferencias en la entrada o el lado de los datos del problema, y el error hacia delante representa las diferencias en la salida, el lado de la solución del algoritmo.

DEFINICIÓN 2.4 Sea x_a una solución aproximada del sistema lineal $Ax = b$. El **residuo** es el vector $r = b - Ax_a$. El **error hacia atrás** es la norma del residuo $\|b - Ax_a\|_\infty$, y el **error hacia delante** es $\|x - x_a\|_\infty$. ❐

► EJEMPLO 2.10 Encuentre los errores hacia atrás y hacia delante para la solución aproximada $x_a = [1, 1]$ del sistema

$$\begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$$

La solución correcta es $x = [2, 1]$. En la norma infinito, el error hacia atrás es

$$\begin{aligned} \|b - Ax_a\|_\infty &= \left\| \begin{bmatrix} 3 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 3 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|_\infty \\ &= \left\| \begin{bmatrix} 1 \\ 3 \end{bmatrix} \right\|_\infty = 3, \end{aligned}$$

y el error hacia delante es

$$\|x - x_a\|_\infty = \left\| \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\|_\infty = \left\| \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\|_\infty = 1. \quad ◄$$

En otros casos, los errores hacia atrás y hacia delante pueden tener diferentes órdenes de magnitud.

► EJEMPLO 2.11 Encuentre los errores hacia delante y hacia atrás para la solución aproximada $[-1, 3{,}0001]$ del sistema

$$\begin{aligned} x_1 + x_2 &= 2 \\ 1.0001x_1 + x_2 &= 2.0001. \end{aligned} \tag{2.17}$$

Primero, encuentre la solución exacta $[x_1, x_2]$. La eliminación gaussiana consta de los pasos

$$\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 1.0001 & 1 & 2.0001 \end{array}\right] \xrightarrow[\text{del renglón 2}]{\text{resta } 1.0001 \times \text{renglón 1}} \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & -0.0001 & -0.0001 \end{array}\right].$$

Al resolver las ecuaciones resultantes

$$\begin{aligned} x_1 + x_2 &= 2 \\ -0.0001x_2 &= -0.0001 \end{aligned}$$

se obtiene la solución $[x_1, x_2] = [1, 1]$.

El error hacia atrás es la norma infinito del vector

$$\begin{aligned} b - Ax_a &= \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1.0001 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 3.0001 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix} - \begin{bmatrix} 2.0001 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.0001 \\ 0.0001 \end{bmatrix}, \end{aligned}$$

que es 0.0001. El error hacia delante es la norma infinito de la diferencia

$$x - x_a = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ 3.0001 \end{bmatrix} = \begin{bmatrix} 2 \\ -2.0001 \end{bmatrix},$$

que es 2.0001. ◀

La figura 2.2 ayuda a aclarar cómo puede haber un error pequeño hacia atrás y un error grande hacia delante al mismo tiempo. Aunque la "raíz aproximada" $(-1, 3.0001)$ está relativamente lejos de la raíz exacta $(1, 1)$, se encuentra muy cercana a ambas rectas. Esto es posible porque las dos líneas son casi paralelas. Si las rectas están lejos de ser paralelas, las magnitudes de los errores hacia delante y hacia atrás serán más parecidas.

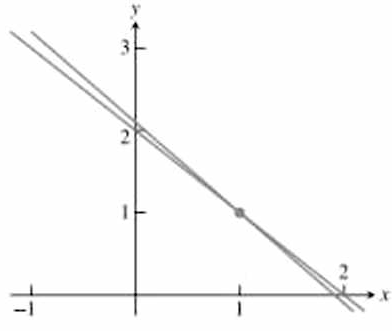


Figura 2.2 La geometría detrás del ejemplo 2.11. El sistema (2.17) está representado por las rectas $x_2 = 2 - x_1$ y $x_2 = 2.0001 - 1.0001x_1$, que se intersecan en (1, 1). El punto (−1, 3.0001) casi se encuentra en ambas rectas y está cerca de ser una solución. En la figura, las diferencias entre las rectas se han exagerado, en realidad están mucho más cercanas.

Si se indica el residuo mediante $r = b - Ax_a$. El **error relativo hacia atrás** del sistema $Ax = b$ se define como

$$\frac{||r||_\infty}{||b||_\infty},$$

y el **error relativo hacia delante** es

$$\frac{||x - x_a||_\infty}{||x||_\infty}.$$

ANOTACIÓN

Condicionamiento El número de condición es un tema siempre presente en el análisis numérico. En el análisis del polinomio de Wilkinson del capítulo 1 se encontró la forma de calcular el factor de magnificación del error en la determinación de raíces, dadas las pequeñas perturbaciones de una ecuación $f(x) = 0$. Para las ecuaciones matriciales $Ax = b$, existe un factor de magnificación del error similar, y el factor máximo posible está dado por $\text{cond}(A) = \|A\|\,\|A^{-1}\|$.

El **factor de magnificación del error** para $Ax = b$ es la razón de los dos, o

$$\text{factor de magnificación del error} = \frac{\text{error relativo hacia delante}}{\text{error relativo hacia atrás}} = \frac{\dfrac{\|x - x_a\|_\infty}{\|x\|_\infty}}{\dfrac{\|r\|_\infty}{\|b\|_\infty}}. \tag{2.18}$$

Para el sistema (2.17), el error relativo hacia atrás es

$$\frac{0.0001}{2.0001} \approx 0.00005 = 0.005\%,$$

y el error relativo hacia delante es

$$\frac{2.0001}{1} = 2.0001 \approx 200\%.$$

El factor de magnificación del error es $2.0001/(0.0001/2.0001) = 40004.0001$.

En el capítulo 1 se definió el concepto del número de condición como la magnificación máxima del error en un rango determinado de errores de entrada. El "rango prescrito" depende del contexto. Ahora se tendrá mayor precisión al respecto en el contexto actual de los sistemas de ecuaciones lineales. Para una matriz fija A, considere la solución de $Ax = b$ para diversos vectores b. En este contexto, b es la entrada y la solución de x es la salida. Un pequeño cambio en la entrada es un pequeño cambio en b, que tiene un factor de magnificación del error. Por lo tanto, se hace la siguiente definición:

DEFINICIÓN 2.5 El **número de condición** de una matriz cuadrada A, **cond**(A), es el factor de magnificación del error máximo posible para resolver $Ax = b$, sobre todos los lados derechos b. ❐

Sorprendentemente, existe una fórmula compacta para el número de condición de una matriz cuadrada. Análoga a la norma de un vector, defina la **norma matricial** de una matriz A de $n \times n$ como

$$\|A\|_\infty = \text{máximo valor absoluto de la suma por renglones}, \tag{2.19}$$

es decir, sume los valores absolutos de cada renglón y asigne el máximo de estos n números a la norma de A.

TEOREMA 2.6 El número de condición de la matriz A de $n \times n$ es

$$\text{cond}(A) = \|A\| \cdot \|A^{-1}\|. \qquad \blacksquare$$

El teorema 2.6, que se demuestra más adelante, permite calcular el número de condición de la matriz de coeficientes del ejemplo 2.11. La norma de

$$A = \begin{bmatrix} 1 & 1 \\ 1.0001 & 1 \end{bmatrix}$$

es $\|A\| = 2.0001$, de acuerdo con (2.19). La inversa de A es

$$A^{-1} = \begin{bmatrix} -10000 & 10000 \\ 10001 & -10000 \end{bmatrix},$$

que tiene norma $\|A^{-1}\| = 20001$. El número de condición de A es

$$\text{cond}(A) = (2.0001)(20001) = 40004.0001.$$

Esto es exactamente la magnificación del error que se encontró en el ejemplo 2.11, con lo que evidentemente se logra el peor de los casos, que define al número de condición. El factor de magnificación del error para cualquier otra b en este sistema será menor que o igual a 40004.0001. En el ejercicio 3 se pide el cálculo de algunos de los otros factores de magnificación del error.

La significancia del número de condición es la misma que en el capítulo 1. Es posible que se presenten factores de magnificación del error con magnitud cond(A). En la aritmética de punto flotante no puede esperarse que el error relativo hacia atrás sea menor que ϵ_{maq}, puesto que el almacenamiento de las entradas de b ya causa errores de ese tamaño. De acuerdo con (2.18), es posible que se presenten errores relativos hacia delante de tamaño $\epsilon_{maq} \cdot \text{cond}(A)$ en la solución de $Ax = b$. En otras palabras, si $\text{cond}(A) \approx 10^k$, se debe estar preparado para perder k dígitos de precisión en el cálculo de x.

En el ejemplo 2.11, $\text{cond}(A) \approx 4 \times 10^4$, por lo que en la precisión doble deben esperarse aproximadamente $16 - 4 = 12$ dígitos correctos en la solución x. Lo anterior puede probarse presentando el mejor solucionador de ecuaciones lineales de MATLAB de propósito general: \.

En MATLAB, el comando de barra inversa `x = A\b` resuelve el sistema lineal usando una versión avanzada de la factorización LU que se estudiará en la sección 2.4. Por ahora, se utilizará como ejemplo de lo que puede esperarse del mejor algoritmo posible funcionando en la aritmética de punto flotante. Los siguientes comandos de MATLAB entregan la solución de computadora x_a para el ejemplo 2.10:

```
>> A = [1 1;1.0001 1]; b=[2;2.0001];
>> xa = A\b
xa =
   1.00000000000222
   0.99999999999778
```

Comparada con la solución correcta, $x = [1, 1]$, la solución calculada tiene alrededor de 11 dígitos correctos, cerca de la predicción hecha a partir del número de condición.

La matriz de Hilbert H, con entradas $H_{ij} = 1/(i + j - 1)$, es notable por su número de condición grande.

► EJEMPLO 2.12 Sea H la matriz de Hilbert de $n \times n$. Use el comando \ de MATLAB para calcular la solución de $Hx = b$, donde $b = H \cdot [1, \ldots, 1]^T$, para $n = 6$ y 10.

El lado derecho b se elige de modo que la solución correcta sea el vector de n unos, a fin de facilitar la comprobación del error hacia delante. MATLAB encuentra el número de condición (en la norma infinito) y calcula la solución:

```
>> n=6;H=hilb(n);
>> cond(H,inf)
ans =
    2.907027900294064e+007
>> b=H*ones(n,1);
>> xa=H\b
xa =
   0.99999999999923
   1.00000000002184
   0.99999999985267
```

```
   1.00000000038240
   0.99999999957855
   1.00000000016588
```

El número de condición de alrededor de 10^7 predice $16 - 7 = 9$ dígitos correctos en el peor de los casos; hay aproximadamente 9 dígitos correctos en la solución calculada. Ahora repita con $n = 10$:

```
>> n=10;H=hilb(n);
>> cond(H,inf)
ans =
    3.535371683074594e+013
>> b=H*ones(n,1);
>> xa=H\b
xa =
   0.99999999875463
   1.00000010746631
   0.99999771299818
   1.00002077769598
   0.99990094548472
   1.00027218303745
   0.99955359665722
   1.00043125589482
   0.99977366058043
   1.00004976229297
```

Como el número de condición es 10^{13}, sólo aparecen $16 - 13 = 3$ dígitos correctos en la solución.

Para una n algo mayor que 10, el número de condición de la matriz de Hilbert es mayor que 10^{16} y no hay dígitos correctos que puedan garantizarse en la x_a calculada. ◄

Incluso un software excelente puede no tener defensa contra un problema mal condicionado. Una mayor precisión ayuda; en la precisión extendida, $\epsilon_{maq} = 2^{-64} \approx 5.42 \times 10^{-20}$, y se comienza con 20 dígitos en lugar de 16. Sin embargo, el número de condición de la matriz de Hilbert crece lo suficientemente rápido con n, como para desarmar, a la larga, cualquier precisión finita razonable.

Por fortuna, los números de condición grandes de la matriz de Hilbert son inusuales. Los sistemas bien condicionados de n ecuaciones lineales con n incógnitas se resuelven por lo general en una precisión doble para $n = 10^4$ y más grandes. Sin embargo, es importante saber que los problemas mal condicionados existen y que el número de condición es útil para el diagnóstico de dicha posibilidad. Consulte los problemas de computadora del 1 al 4 y obtenga más ejemplos de magnificación del error y números de condición.

La norma vectorial del infinito se usa en esta sección como un modo sencillo de asignar una longitud a un vector. Es un ejemplo de una **norma vectorial** $\|x\|$ que satisface tres propiedades:

(i) $\|x\| \geq 0$ con igualdad si y sólo si $x = [0, \ldots, 0]$
(ii) para cada escalar α y vector x, $\|\alpha x\| = |\alpha| \cdot \|x\|$
(iii) para los vectores x, y, $\|x + y\| \leq \|x\| + \|y\|$.

Además, $\|A\|_\infty$ es un ejemplo de una **norma matricial**, que satisface tres propiedades similares:

(i) $\|A\| \geq 0$ con igualdad si y sólo si $A = 0$
(ii) para cada escalar α y matriz A, $\|\alpha A\| = |\alpha| \cdot \|A\|$
(iii) para las matrices A, B, $\|A + B\| \leq \|A\| + \|B\|$.

Como otro ejemplo, la **norma 1** del vector $x = [x_1, \ldots, x_n]$ es $\|x\|_1 = |x_1| + \ldots + |x_n|$. La norma 1 de la matriz A de $n \times n$ es $\|A\|_1 =$ máximo valor absoluto de suma de columnas, es decir, el máximo de las normas 1 de los vectores columna. Vea en los ejercicios 9 y 10 la verificación de estas definiciones de normas.

El factor de magnificación del error, el número de condición y la norma de la matriz que acaban de analizarse pueden definirse para cualquier norma vectorial y matricial. En este libro se restringirá la atención a las normas matriciales que son **normas del operador**, lo que significa que pueden definirse en términos de una norma vectorial particular como

$$||A|| = \text{máx}\, \frac{||Ax||}{||x||},$$

donde el máximo se toma sobre todos los vectores x distintos de cero. Entonces, por definición, la norma matricial es consistente con la norma vectorial asociada, en el sentido de que

$$||Ax|| \leq ||A|| \cdot ||x|| \tag{2.20}$$

para cualquier matriz A y vector x. Vea en los ejercicios 10 y 11 la verificación de que la norma $\|A\|_\infty$ definida por (2.20) no sólo es una norma matricial, sino también la norma del operador para la norma vectorial infinito.

Este hecho permite demostrar la expresión simple mencionada con anterioridad para cond(A). La prueba funciona para la norma infinito y para cualquier otra norma de operador.

Demostración del teorema 2.6. Se usan las igualdades $A(x - x_a) = r$ y $Ax = b$. Por la propiedad de consistencia (2.20),

$$||x - x_a|| \leq ||A^{-1}|| \cdot ||r||$$

y

$$\frac{1}{||b||} \geq \frac{1}{||A||\,||x||}.$$

Al colocar las dos desigualdades juntas se obtiene

$$\frac{||x - x_a||}{||x||} \leq \frac{||A||}{||b||} ||A^{-1}|| \cdot ||r||,$$

lo que muestra que $\|A\|\,\|A^{-1}\|$ es una cota superior para todos los factores de magnificación del error. En segundo lugar, puede demostrarse que la cantidad siempre es alcanzable. Escoja una x de tal forma que $\|A\| = \|Ax\|/\|x\|$ y r de manera que $\|A^{-1}\| = \|A^{-1}r\|/\|r\|$, ambas posibles por la definición de la norma matricial del operador. Sea $x_a = x - A^{-1}r$ de modo que $x - x_a = \mathrm{A}^{-1}r$. Entonces queda por comprobar la igualdad

$$\frac{||x - x_a||}{||x||} = \frac{||A^{-1}r||}{||x||} = \frac{||A^{-1}||\,||r||\,||A||}{||Ax||}$$

para esta elección particular de x y r.

2.3.2 Dominancia

Una segunda fuente importante de error en la eliminación gaussiana clásica es mucho más fácil de arreglar. La dominancia se muestra en el siguiente ejemplo.

► EJEMPLO 2.13 Considere el sistema de ecuaciones

$$\begin{aligned} 10^{-20}x_1 + x_2 &= 1 \\ x_1 + 2x_2 &= 4. \end{aligned}$$

Se resolverá el sistema tres veces: una vez con precisión completa, una segunda vez imitando a una computadora con aritmética de precisión doble IEEE, y una vez más cambiando en primer lugar el orden de las ecuaciones.

1. **Solución exacta.** En forma de tabla. La eliminación gaussiana procede de la siguiente manera

$$\left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 2 & 4 \end{array}\right] \xrightarrow[\text{del renglón 2}]{\text{resta } 10^{20} \times \text{renglón 1}} \left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & 2-10^{20} & 4-10^{20} \end{array}\right].$$

La ecuación inferior es

$$(2-10^{20})x_2 = 4-10^{20} \longrightarrow x_2 = \frac{4-10^{20}}{2-10^{20}},$$

y de la ecuación superior se obtiene

$$10^{-20}x_1 + \frac{4-10^{20}}{2-10^{20}} = 1$$

$$x_1 = 10^{20}\left(1 - \frac{4-10^{20}}{2-10^{20}}\right)$$

$$x_1 = \frac{-2 \times 10^{20}}{2-10^{20}}.$$

La solución exacta es

$$[x_1, x_2] = \left[\frac{2 \times 10^{20}}{10^{20}-2}, \frac{4-10^{20}}{2-10^{20}}\right] \approx [2, 1].$$

2. **Precisión doble IEEE.** La versión en computadora de la eliminación gaussiana avanza en forma un poco diferente:

$$\left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 2 & 4 \end{array}\right] \xrightarrow[\text{del renglón 2}]{\text{resta } 10^{20} \times \text{renglón 1}} \left[\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & 2-10^{20} & 4-10^{20} \end{array}\right].$$

En la precisión doble IEEE, $2 - 10^{20}$ es igual que -10^{20}, debido al redondeo. De manera similar, $4 - 10^{20}$ se almacena como -10^{20}. Ahora la ecuación inferior es

$$-10^{20}x_2 = -10^{20} \longrightarrow x_2 = 1.$$

La versión en aritmética de máquina de la ecuación superior se convierte en

$$10^{-20}x_1 + 1 = 1,$$

así que $x_1 = 0$. La solución calculada es exactamente

$$[x_1, x_2] = [0, 1].$$

Esta solución tiene un error relativo grande en comparación con la solución exacta.

3. **Precisión doble IEEE, después del intercambio de renglones.** Se repite la versión en computadora de la eliminación gaussiana, después de cambiar el orden de las dos ecuaciones:

$$\left[\begin{array}{cc|c} 1 & 2 & 4 \\ 10^{-20} & 1 & 1 \end{array}\right] \xrightarrow[\text{del renglón 2}]{\text{reste } 10^{-20} \times \text{renglón 1}} \left[\begin{array}{cc|c} 1 & 2 & 4 \\ 0 & 1-2 \times 10^{-20} & 1-4 \times 10^{-20} \end{array}\right].$$

En la precisión doble IEEE, $1 - 2 \times 10^{-20}$ se almacena como 1 y $1 - 4 \times 10^{-20}$ se almacena como 1. Las ecuaciones son ahora

$$x_1 + 2x_2 = 4$$

$$x_2 = 1,$$

de donde se obtiene la solución calculada $x_1 = 2$ y $x_2 = 1$. Por supuesto, ésta no es la respuesta exacta, pero es correcta hasta aproximadamente 16 dígitos, que es lo más que se puede hacer a partir de un cálculo que usa números de punto flotante con 52 bits.

La diferencia entre los dos últimos cálculos es significativa. La versión 3 dio una solución aceptable, mientras que la versión 2 no lo hizo. Un análisis de lo que salió mal con la versión 2 conduce a considerar el multiplicador 10^{20} que se utilizó para el paso de eliminación. El efecto de restar 10^{20} veces la ecuación superior de la ecuación inferior fue atenuar, o "dominar", la ecuación final. Aun cuando en un principio había dos ecuaciones independientes, o fuentes de información, después del paso de eliminación en la versión 2, existen en esencia dos copias de la ecuación superior. Como la ecuación inferior ha desaparecido, para todos los efectos prácticos, no puede esperarse que la solución calculada satisfaga la ecuación inferior; y no lo hace.

Por otro lado, la versión 3 completa la eliminación sin dominar ecuaciones, porque el multiplicador es 10^{-20}. Después de la eliminación, en gran medida las dos ecuaciones originales todavía existen, un poco cambiadas en la forma triangular. El resultado es una solución aproximada que es mucho más precisa. ◀

La moraleja del ejemplo 2.13 es que los multiplicadores en la eliminación gaussiana deben ser lo más pequeños posible para evitar problemas de dominación. Por fortuna, existe una sencilla modificación de la eliminación gaussiana simple que hace que el valor absoluto de los multiplicadores no sea mayor que 1. Este nuevo protocolo, que implica intercambios juiciosos de renglones en la tabla, se llama pivoteo parcial, el tema de la siguiente sección.

2.3 Ejercicios

1. Encuentre la norma $\|A\|_\infty$ de cada uno de las siguientes matrices:

$$\text{(a)}\quad A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{(b)}\quad A = \begin{bmatrix} 1 & 5 & 1 \\ -1 & 2 & -3 \\ 1 & -7 & 0 \end{bmatrix}.$$

2. Encuentre el número de condición (norma infinito) de

$$\text{(a)}\quad A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{(b)}\quad A = \begin{bmatrix} 1 & 2.01 \\ 3 & 6 \end{bmatrix} \quad \text{(c)}\quad A = \begin{bmatrix} 6 & 3 \\ 4 & 2 \end{bmatrix}.$$

3. Encuentre los errores hacia delante y hacia atrás, y el factor de magnificación del error (en la norma infinito) para las siguientes soluciones aproximadas x_a del sistema del ejemplo 2.11: (a) $[-1, 3]$ (b) $[0, 2]$ (c) $[2, 2]$ (d) $[-2, 4]$ (e) $[-2, 4.0001]$.

4. Encuentre los errores hacia delante y hacia atrás, y el factor de magnificación del error para las siguientes soluciones aproximadas del sistema $x_1 + 2x_2 = 1$, $2x_1 + 4.01x_2 = 2$: (a) $[-1, 1]$ (b) $[3, -1]$ (c) $[2, -1/2]$.

5. Encuentre los errores relativos hacia delante y hacia atrás, y el factor de magnificación del error para las siguientes soluciones aproximadas del sistema $x_1 - 2x_2 = 3$, $3x_1 - 4x_2 = 7$: (a) $[-2, -4]$ (b) $[-2, -3]$ (c) $[0, -2]$ (d) $[-1, -1]$ (e) ¿Cuál es el número de condición de la matriz de coeficientes?

6. Encuentre los errores relativos hacia delante y hacia atrás, y el factor de magnificación del error para las siguientes soluciones aproximadas del sistema $x_1 + 2x_2 = 3$, $2x_1 + 4.01x_2 = 6.01$: (a) $[-10, 6]$ (b) $[-100, 52]$ (c) $[-600, 301]$ (d) $[-599, 301]$ (e) ¿Cuál es el número de condición de la matriz de coeficientes?

7. Encuentre la norma $\|H\|_\infty$ de la matriz de Hilbert de 5×5.

8. (a) Encuentre el número de condición de la matriz de coeficientes del sistema $\begin{bmatrix} 1 & 1 \\ 1+\delta & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2+\delta \end{bmatrix}$ como una función de $\delta > 0$. (b) Determine el factor de magnificación del error para la raíz aproximada $x_a = [-1, 3 + \delta]$.

9. (a) Demuestre que la norma infinito $\|x\|_\infty$ es una norma vectorial. (b) Demuestre que la norma 1 $\|x\|_1$ es una norma vectorial.

10. (a) Demuestre que la norma infinito $\|A\|_\infty$ es una norma matricial. (b) Demuestre que la norma 1 $\|A\|_1$ es una norma matricial.

11. Demuestre que la norma infinito matricial es la norma del operador de la norma infinito vectorial.

12. Demuestre que la norma 1 matricial es la norma del operador de la norma 1 vectorial.

13. Para las matrices del ejercicio 1, encuentre un vector x que satisfaga $\|A\|_\infty = \|Ax\|_\infty / \|x\|_\infty$.

14. Para las matrices del ejercicio 1, encuentre un vector x que satisfaga $\|A\|_1 = \|Ax\|_1 / \|x\|_1$.

15. Encuentre la factorización LU de

$$A = \begin{bmatrix} 10 & 20 & 1 \\ 1 & 1.99 & 6 \\ 0 & 50 & 1 \end{bmatrix}.$$

¿Cuál es el multiplicador l_{ij} de mayor magnitud requerido?

2.3 Problemas de computadora

1. Para la matriz $n \times n$ con entradas $A_{ij} = 5/(i + 2j - 1)$, establezca $x = [1, \ldots, 1]^T$ y $b = Ax$. Utilice el programa de MATLAB del problema de computadora 2.1.1 o el comando de diagonal invertida (\) de MATLAB para calcular x_c, la solución en computadora de precisión doble. Encuentre la norma infinito del error hacia delante y el factor de magnificación del error del problema $Ax = b$, y compárelo con el número de condición de A: (a) $n = 6$ (b) $n = 10$.

2. Resuelva el problema de computadora 1 para la matriz con entradas $A_{ij} = 1/(|i - j| + 1)$.

3. Sea A la matriz de $n \times n$ con entradas $A_{ij} = |i - j| + 1$. Defina $x = [1, \ldots, 1]^T$ y $b = Ax$. Para $n = 100, 200, 300, 400$ y 500, utilice el programa de MATLAB para el problema de computadora 2.1.1 o el comando de barra invertida de MATLAB para calcular x_c, la solución en computadora de precisión doble. Calcule la norma infinito del error hacia delante para cada solución. Encuentre los cinco factores de magnificación del error para los problemas $Ax = b$ y compárelos con los números de condición correspondientes.

4. Realice los pasos del problema de computadora 3 para la matriz con entradas $A_{ij} = \sqrt{(i - j)^2 + n/10}$.

5. ¿Para qué valores de n la solución del problema de computadora 1 no tiene cifras significativas correctas?

6. Utilice el programa de MATLAB del problema de computadora 2.1.1 para llevar a cabo implementaciones de precisión doble con las versiones 2 y 3 del ejemplo 2.13, además compare sus soluciones con los resultados teóricos que se encuentran en el libro.

2.4 LA FACTORIZACIÓN PA = LU

La forma de eliminación gaussiana considerada hasta ahora se conoce como "simple", y se enfrenta a dos graves dificultades: encontrar un pivote cero y la dominancia. Para una matriz no singular, ambas pueden evitarse con un algoritmo mejorado. La clave de esta mejora es un protocolo eficiente para el intercambio de renglones en la matriz de coeficientes, llamado pivoteo parcial.

2.4.1 Pivoteo parcial

Al inicio de la eliminación gaussiana clásica de n ecuaciones con n incógnitas, el primer paso es usar el elemento de la diagonal a_{11} como pivote para eliminar la primera columna. El protocolo del **pivoteo parcial** consiste en comparar los números antes de la ejecución de cada etapa de eliminación. Se localiza la entrada más grande de la primera columna y su renglón se intercambia con el renglón pivote, en este caso el renglón superior.

En otras palabras, al inicio de la eliminación gaussiana, el pivoteo parcial pide que se seleccione el p-ésimo renglón, donde

$$|a_{p1}| \geq |a_{i1}| \tag{2.21}$$

para todo $1 \leq i \leq n$, e intercambiar los renglones 1 y p. A continuación, procede la eliminación de la columna 1 como de costumbre, con la "nueva" versión de a_{11} como pivote. El multiplicador utilizado para eliminar a_{i1} será

$$m_{i1} = \frac{a_{i1}}{a_{11}}$$

y $|m_{i1}| \leq 1$.

La misma comprobación se aplica a cada elección de pivote durante el algoritmo. Para decidir sobre el segundo pivote, se inicia con la a_{22} actual y se revisan todas las entradas que están justo abajo. Se selecciona el renglón p de tal forma que

$$|a_{p2}| \geq |a_{i2}|$$

para toda $2 \leq i \leq n$, y si $p \neq 2$ los renglones 2 y p se intercambian. El renglón 1 nunca se involucra en este paso. Si $|a_{22}|$ ya es la más grande, no se hace intercambio de renglón.

El protocolo se aplica a cada columna durante la eliminación. Antes de eliminar la columna k, se localiza la p con $k \leq p \leq n$ y el $|a_{pk}|$ más grande, y los renglones p y k se intercambian si es necesario antes de continuar con la eliminación. Observe que el uso del pivoteo parcial asegura que todos los multiplicadores, o entradas de L, no sean mayores que 1 en valor absoluto. Con este cambio de menor importancia en la aplicación de la eliminación gaussiana, el problema de dominancia ilustrado en el ejemplo 2.13 se evita por completo.

► EJEMPLO 2.14 Aplique la eliminación gaussiana con un pivoteo parcial para resolver el sistema (2.1).

Las ecuaciones pueden escribirse en la forma de tabla como

$$\left[\begin{array}{cc|c} 1 & 1 & 3 \\ 3 & -4 & 2 \end{array}\right].$$

De acuerdo con el pivoteo parcial, se compara $|a_{11}| = 1$ con todas las entradas por debajo de ella, en este caso la única entrada $a_{21} = 3$. Como $|a_{21}| > |a_{11}|$, deben intercambiarse los renglones 1 y 2. La nueva tabla es

$$\left[\begin{array}{cc|c} 3 & -4 & 2 \\ 1 & 1 & 3 \end{array}\right] \longrightarrow \begin{array}{c}\text{resta } \frac{1}{3} \times \text{renglón 1} \\ \text{del renglón 2}\end{array} \longrightarrow \left[\begin{array}{cc|c} 3 & -4 & 2 \\ 0 & \frac{7}{3} & \frac{7}{3} \end{array}\right].$$

Después de la sustitución hacia atrás, la solución es $x_2 = 1$ y entonces $x_1 = 2$, como se determinó antes. Cuando este sistema se resolvió por primera vez, el multiplicador fue 3, pero bajo el pivoteo parcial esto nunca podría ocurrir. ◄

► **EJEMPLO 2.15** Aplique la eliminación gaussiana con pivoteo parcial para resolver el sistema

$$\begin{aligned} x_1 - x_2 + 3x_3 &= -3 \\ -x_1 - 2x_3 &= 1 \\ 2x_1 + 2x_2 + 4x_3 &= 0. \end{aligned}$$

Este ejemplo se escribe en forma de tabla como

$$\left[\begin{array}{rrr|r} 1 & -1 & 3 & -3 \\ -1 & 0 & -2 & 1 \\ 2 & 2 & 4 & 0 \end{array}\right].$$

Bajo el pivoteo parcial se compara $|a_{11}| = 1$ con $|a_{21}| = 1$ y $|a_{31}| = 2$, y se elige a_{31} como nuevo pivote. Esto se logra a través del intercambio de los renglones 1 y 3:

$$\left[\begin{array}{rrr|r} 1 & -1 & 3 & -3 \\ -1 & 0 & -2 & 1 \\ 2 & 2 & 4 & 0 \end{array}\right] \xrightarrow{\text{intercambia el renglón 1 y el renglón 3}} \left[\begin{array}{rrr|r} 2 & 2 & 4 & 0 \\ -1 & 0 & -2 & 1 \\ 1 & -1 & 3 & -3 \end{array}\right]$$

$$\xrightarrow{\text{resta } -\frac{1}{2} \times \text{renglón 1 del renglón 2}} \left[\begin{array}{rrr|r} 2 & 2 & 4 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & -1 & 3 & -3 \end{array}\right]$$

$$\xrightarrow{\text{resta } \frac{1}{2} \times \text{renglón 1 del renglón 3}} \left[\begin{array}{rrr|r} 2 & 2 & 4 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -2 & 1 & -3 \end{array}\right].$$

Antes de eliminar la columna 2 debe compararse el actual $|a_{22}|$ con el actual $|a_{32}|$. Debido a que este último es más grande, se vuelven a intercambiar renglones:

$$\left[\begin{array}{rrr|r} 2 & 2 & 4 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -2 & 1 & -3 \end{array}\right] \xrightarrow{\text{intercambia el renglón 2 y el renglón 3}} \left[\begin{array}{rrr|r} 2 & 2 & 4 & 0 \\ 0 & -2 & 1 & -3 \\ 0 & 1 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{\text{resta } -\frac{1}{2} \times \text{renglón 2 del renglón 3}} \left[\begin{array}{rrr|r} 2 & 2 & 4 & 0 \\ 0 & -2 & 1 & -3 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{array}\right].$$

Observe que los tres multiplicadores son menores que 1 en valor absoluto.

Ahora, las ecuaciones pueden resolverse con facilidad. A partir de

$$\begin{aligned} \frac{1}{2}x_3 &= -\frac{1}{2} \\ -2x_2 + x_3 &= -3 \\ 2x_1 + 2x_2 + 4x_3 &= 0, \end{aligned}$$

se encuentra que $x = [1, 1, -1]$. ◄

Observe que el pivoteo parcial también resuelve el problema de pivotes cero. Cuando se encuentra un potencial pivote cero; por ejemplo, si $a_{11} = 0$, se cambia de inmediato por un pivote distinto de cero en algún lugar de su columna. Si no hay tal entrada distinta de cero en o por debajo de la entrada diagonal, entonces la matriz es singular y de cualquier modo la eliminación gaussiana no podrá proporcionar una solución.

2.4.2 Matrices de permutación

Antes de mostrar cómo pueden utilizarse los intercambios de renglones con el método de factorización LU para la eliminación gaussiana, se presentarán las propiedades fundamentales de las matrices de permutación.

DEFINICIÓN 2.7 Una **matriz de permutación** es una matriz de $n \times n$ que consta sólo de ceros, a excepción de un 1 en cada renglón y columna. ❐

De manera equivalente, una matriz de permutación P se crea al aplicar arbitrariamente intercambios de renglones (o de columnas) a la matriz identidad de $n \times n$. Por ejemplo,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

son las únicas matrices de permutación de 2×2, y

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$
$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

son las seis matrices de permutación de 3×3.

El siguiente teorema indica de un vistazo qué acción causa una matriz de permutación cuando se multiplica en el lado izquierdo de otra matriz.

TEOREMA 2.8 **Teorema fundamental de las matrices de permutación.** Sea P la matriz de permutación de $n \times n$ formada por un conjunto particular de intercambios de renglones aplicado a la matriz identidad. Entonces, para cualquier matriz A de $n \times n$, PA es la matriz obtenida al aplicar exactamente la misma serie de intercambios de fila a A. ■

Por ejemplo, la matriz de permutación

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

se forma mediante el intercambio de los renglones 2 y 3 de la matriz identidad. Al multiplicar una matriz arbitraria por la izquierda con P tiene el efecto de intercambiar los renglones 2 y 3:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} a & b & c \\ g & h & i \\ d & e & f \end{bmatrix}.$$

Una buena manera de recordar el teorema 2.8 es imaginar la multiplicación P veces de la matriz identidad I:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Existen dos formas distintas de ver esta igualdad: primero, como la multiplicación por la matriz identidad (de modo que se obtenga la matriz de permutación a la derecha); segundo, como la matriz de permutación que actúa sobre los renglones de la matriz identidad. El contenido del teorema 2.8 es que los intercambios de renglones causados por la multiplicación por P son exactamente los que están involucrados en la construcción de P.

2.4.3 Factorización PA = LU

En esta sección, todos los conocimientos adquiridos hasta ahora acerca de la eliminación gaussiana se reúnen en la factorización PA = LU. Ésta consiste en la formulación matricial de la eliminación con pivoteo parcial. La factorización PA = LU es la herramienta eficaz más utilizada para la resolución de sistemas de ecuaciones lineales.

Como su nombre lo indica, la factorización PA = LU es tan sólo la factorización *LU* de una versión de *A* con renglones intercambiados. Bajo el pivoteo parcial, los renglones que deben intercambiarse no se conocen desde el principio, por lo que es necesario tener cuidado de colocar la información del intercambio de renglones en la factorización. En particular, debe darse seguimiento a los multiplicadores previos cuando se hace un intercambio de renglones. A continuación se presenta un ejemplo inicial.

► **EJEMPLO 2.16** Encuentre la factorización PA = LU de la matriz

$$A = \begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix}.$$

En primer lugar, los renglones 1 y 2 deben intercambiarse de acuerdo con el pivoteo parcial:

$$\begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix} \longrightarrow \begin{matrix} P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \text{intercambia los} \\ \text{renglones 1 y 2} \end{matrix} \longrightarrow \begin{bmatrix} 4 & 4 & -4 \\ 2 & 1 & 5 \\ 1 & 3 & 1 \end{bmatrix}.$$

Se usará la matriz de permutación *P* para mantener un seguimiento de la permutación acumulada de renglones que se ha realizado en el camino. Ahora se realizan dos operaciones por renglón, es decir,

$$\begin{matrix} \longrightarrow \\ \text{resta } \frac{1}{2} \times \text{renglón 1} \\ \text{del renglón 2} \end{matrix} \begin{bmatrix} 4 & 4 & -4 \\ \textcircled{\frac{1}{2}} & -1 & 7 \\ 1 & 3 & 1 \end{bmatrix} \begin{matrix} \longrightarrow \\ \text{resta } \frac{1}{4} \times \text{renglón 1} \\ \text{del renglón 3} \end{matrix} \begin{bmatrix} 4 & 4 & -4 \\ \textcircled{\frac{1}{2}} & -1 & 7 \\ \textcircled{\frac{1}{4}} & 2 & 2 \end{bmatrix},$$

para eliminar la primera columna. Se ha hecho algo nuevo: en lugar de poner sólo un cero en la posición eliminada, se ha hecho del cero una ubicación de almacenamiento. Dentro del cero en la posición (i, j), se almacenó el multiplicador m_{ij} que se utilizó para eliminar esa posición. Esto se hace por una razón. Éste es el mecanismo mediante el cual los multiplicadores se quedarán con su renglón, en caso de que se hagan intercambios de renglones en el futuro.

Enseguida debe hacerse una comparación para elegir el segundo pivote. Como $|a_{22}| = 1 < 2 = |a_{32}|$, se requiere un intercambio de renglones antes de eliminar la segunda columna. Observe que los multiplicadores previos se mueven junto con el cambio de renglón:

$$\begin{matrix} \longrightarrow \\ \\ \\ \end{matrix} \begin{matrix} P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \\ \text{intercambia los} \\ \text{renglones 2 y 3} \end{matrix} \longrightarrow \begin{bmatrix} 4 & 4 & -4 \\ \textcircled{\frac{1}{4}} & 2 & 2 \\ \textcircled{\frac{1}{2}} & -1 & 7 \end{bmatrix}$$

Por último, la eliminación termina con una operación por renglón más:

$$\begin{matrix} \longrightarrow \\ \text{resta } -\frac{1}{2} \times \text{renglón 2} \\ \text{del renglón 3} \end{matrix} \begin{bmatrix} 4 & 4 & -4 \\ \textcircled{\frac{1}{4}} & 2 & 2 \\ \textcircled{\frac{1}{2}} & \textcircled{-\frac{1}{2}} & 8 \end{bmatrix}.$$

Ésta es la eliminación terminada. Ahora es posible leer la factorización PA = LU:

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}}_{P}\underbrace{\begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix}}_{A}=\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}}_{L}\underbrace{\begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & 0 & 8 \end{bmatrix}}_{U} \tag{2.22}$$

Las entradas de L se ubican dentro de los ceros en el triángulo inferior de la matriz (debajo de la diagonal principal), y U se encuentra en la parte superior del triángulo. La matriz de permutación final (acumulativa) sirve como P. ◄

El uso de la factorización PA = LU para resolver un sistema de ecuaciones $Ax = b$ es una pequeña variante de la versión $A = LU$. Multiplique por la izquierda la ecuación $Ax = b$ por P, y después proceda como antes:

$$\begin{aligned} PAx &= Pb \\ LUx &= Pb. \end{aligned} \tag{2.23}$$

Resuelva

$$\begin{aligned} &1.\ Lc = Pb \text{ para } c. \\ &2.\ Ux = c \text{ para } x. \end{aligned} \tag{2.24}$$

El punto importante, como se mencionó con anterioridad, es que la parte complicada del cálculo, la determinación de PA = LU, puede hacerse sin conocer b. Como la factorización LU resultante es de PA, una versión con renglones permutados de los coeficientes de la ecuación, es necesario permutar el vector derecho b exactamente del mismo modo antes de proceder con el paso de sustitución hacia atrás. Esto se consigue utilizando Pb en el primer paso de la sustitución hacia atrás. El valor de la formulación de una matriz para la eliminación gaussiana es evidente: todos los detalles del seguimiento de la eliminación y el pivoteo son automáticos y están contenidos en las ecuaciones matriciales.

► EJEMPLO 2.17 Use la factorización PA = LU para resolver el sistema $Ax = b$, donde

$$A=\begin{bmatrix} 2 & 1 & 5 \\ 4 & 4 & -4 \\ 1 & 3 & 1 \end{bmatrix}, \quad b=\begin{bmatrix} 5 \\ 0 \\ 6 \end{bmatrix}.$$

La factorización PA = LU se conoce a partir de (2.22). Queda por completar las dos sustituciones hacia atrás.

1. $Lc = Pb$:

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 5 \\ 0 \\ 6 \end{bmatrix}=\begin{bmatrix} 0 \\ 6 \\ 5 \end{bmatrix}.$$

Comenzando en la parte superior, se tiene

$$\begin{aligned} c_1 &= 0 \\ \frac{1}{4}(0) + c_2 &= 6 \Rightarrow c_2 = 6 \\ \frac{1}{2}(0) - \frac{1}{2}(6) + c_3 &= 5 \Rightarrow c_3 = 8. \end{aligned}$$

2. $Ux = c$:

$$\begin{bmatrix} 4 & 4 & -4 \\ 0 & 2 & 2 \\ 0 & 0 & 8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 0 \\ 6 \\ 8 \end{bmatrix}$$

Comenzando en la parte inferior,

$$\begin{aligned} 8x_3 = 8 &\Rightarrow x_3 = 1 \\ 2x_2 + 2(1) = 6 &\Rightarrow x_2 = 2 \\ 4x_1 + 4(2) - 4(1) = 0 &\Rightarrow x_1 = -1. \end{aligned} \tag{2.25}$$

Por lo tanto, la solución es $x = [-1, 2, 1]$. ◄

► **EJEMPLO 2.18** Resuelva el sistema $2x_1 + 3x_2 = 4$, $3x_1 + 2x_2 = 1$ usando la factorización PA = LU con pivoteo parcial.

En forma matricial, se escribe la ecuación

$$\begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}.$$

Se inicia haciendo caso omiso del lado derecho b. De acuerdo con el pivoteo parcial, los renglones 1 y 2 deben intercambiarse (porque $a_{21} > a_{11}$). El paso de eliminación es

$$A = \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix} \xrightarrow[\text{renglones 1 y 2}]{P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ intercambia los}} \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$$

$$\xrightarrow[\text{del renglón 2}]{\text{resta } \frac{2}{3} \times \text{ renglón 1}} \begin{bmatrix} 3 & 2 \\ \textcircled{\frac{2}{3}} & \frac{5}{3} \end{bmatrix}.$$

Por lo tanto, la factorización PA = LU es

$$\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 \\ \frac{2}{3} & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 3 & 2 \\ 0 & \frac{5}{3} \end{bmatrix}}_{U}.$$

La primera sustitución hacia atrás $Lc = Pb$ es

$$\begin{bmatrix} 1 & 0 \\ \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}.$$

Comenzando en la parte superior, se tiene

$$\begin{aligned} c_1 &= 1 \\ \frac{2}{3}(1) + c_2 = 4 &\Rightarrow c_2 = \frac{10}{3}. \end{aligned}$$

La segunda sustitución hacia atrás $Ux = c$ es

$$\begin{bmatrix} 3 & 2 \\ 0 & \frac{5}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{10}{3} \end{bmatrix}.$$

Empezando en la parte inferior, se tiene

$$\begin{aligned} \frac{5}{3}x_2 = \frac{10}{3} &\Rightarrow x_2 = 2 \\ 3x_1 + 2(2) = 1 &\Rightarrow x_1 = -1. \end{aligned} \tag{2.26}$$

Por lo tanto, la solución es $x = [-1, 2]$. ◄

Cada matriz de $n \times n$ tiene una factorización PA = LU. Sólo se sigue la regla del pivoteo parcial, y si el pivote resultante es cero, significa que todas las entradas que tienen que ser eliminadas ya son iguales a cero, por lo que la columna está terminada.

Todas las técnicas descritas hasta ahora se implementan en MATLAB. La forma más sofisticada de eliminación gaussiana que se ha analizado es la factorización PA = LU. El comando de MATLAB `lu` acepta una matriz cuadrada de coeficientes A y regresa P, L y U. La siguiente secuencia de comandos en MATLAB define la matriz del ejemplo 2.16 y calcula su factorización:

```
>> A=[2 1 5; 4 4 -4; 1 3 1];
>> [L,U,P]=lu(A)

L=

    1.0000         0         0
    0.2500    1.0000         0
    0.5000   -0.5000    1.0000

U=

     4     4    -4
     0     2     2
     0     0     8

P=

     0     1     0
     0     0     1
     1     0     0
```

2.4 Ejercicios

1. Encuentre la factorización PA = LU (usando el pivoteo parcial) de las matrices siguientes:

$$\text{(a)} \begin{bmatrix} 1 & 3 \\ 2 & 3 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} \quad \text{(c)} \begin{bmatrix} 1 & 5 \\ 5 & 12 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

2. Encuentre la factorización PA = LU (usando el pivoteo parcial) de las matrices siguientes:

$$\text{(a)} \begin{bmatrix} 1 & 1 & 0 \\ 2 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 0 & 1 & 3 \\ 2 & 1 & 1 \\ -1 & -1 & 2 \end{bmatrix} \quad \text{(c)} \begin{bmatrix} 1 & 2 & -3 \\ 2 & 4 & 2 \\ -1 & 0 & 3 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ -2 & 1 & 0 \end{bmatrix}$$

3. Resuelva el sistema al encontrar la factorización PA = LU, para después realizar la sustitución hacia atrás en dos etapas.

$$\text{(a)} \begin{bmatrix} 3 & 7 \\ 6 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -11 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

4. Resuelva el sistema al encontrar la factorización PA = LU, para después realizar la sustitución hacia atrás en dos etapas.

$$\text{(a)} \begin{bmatrix} 4 & 2 & 0 \\ 4 & 4 & 2 \\ 2 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} -1 & 0 & 1 \\ 2 & 1 & 1 \\ -1 & 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 17 \\ 3 \end{bmatrix}$$

5. Escriba una matriz P de 5×5 de tal modo que al multiplicar por la izquierda de otra matriz por P cause que los renglones 2 y 5 se intercambien.

6. (a) Escriba una matriz P de 4×4 de tal modo que al multiplicar por la izquierda otra matriz por P cause que los renglones segundo y cuarto de la matriz se intercambien. (b) ¿Cuál es el efecto de multiplicar por la derecha por P? Pruebe su respuesta con un ejemplo.

7. Cambie cuatro entradas de la matriz de la izquierda para hacer la ecuación matricial correcta:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 5 & 6 \\ 5 & 6 & 7 & 8 \\ 7 & 8 & 9 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 0 \\ 1 & 2 & 3 & 4 \end{bmatrix}.$$

8. Encuentre la factorización PA = LU de la matriz A del ejercicio 2.3.15. ¿Cuál es el mayor multiplicador l_{ij} requerido?

9. (a) Encuentre la factorización PA = LU de $A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{bmatrix}$. (b) Sea A la matriz de $n \times n$ con la misma forma que en (a). Describa las entradas de cada matriz de su factorización PA = LU.

10. (a) Suponga que A es una matriz de $n \times n$ con entradas $|a_{ij}| \leq 1$ para $1 \leq i, j \leq n$. Demuestre que la matriz U en su factorización PA = LU satisface $|u_{ij}| \leq 2^{n-1}$ para toda $1 \leq i, j \leq n$. Vea el ejercicio 9(b). (b) Formule y pruebe un hecho análogo para una matriz arbitraria A de $n \times n$.

Comprobación en la realidad ✔ **2** ***La viga de Euler-Bernoulli***

La viga de Euler-Bernoulli es un modelo fundamental para un material de flexión bajo esfuerzo. La discretización convierte al modelo de ecuaciones diferenciales en un sistema de ecuaciones lineales. Cuanto menor sea el tamaño de la discretización, mayor será el sistema resultante de ecuaciones. Este ejemplo proporcionará un interesante caso de estudio de los papeles del tamaño del sistema y del mal condicionamiento en el cálculo científico.

El desplazamiento vertical de la viga se representa mediante una función $y(x)$, donde $0 \leq x \leq L$ a lo largo de la viga con longitud L. En el cálculo se usarán unidades MKS: metros, kilogramos, segundos. El desplazamiento $y(x)$ satisface la ecuación de Euler-Bernoulli

$$EIy'''' = f(x) \tag{2.27}$$

donde E, el módulo de Young del material, e I, el momento de inercia del área, son constantes a lo largo de la viga. La $f(x)$ en el lado derecho es la carga aplicada, incluyendo el peso de la viga, en unidades de fuerza por unidad de longitud.

Las técnicas para las derivadas de discretización se encuentran en el capítulo 5, donde se muestra que una aproximación razonable para la cuarta derivada es

$$y''''(x) \approx \frac{y(x-2h) - 4y(x-h) + 6y(x) - 4y(x+h) + y(x+2h)}{h^4} \tag{2.28}$$

para un pequeño incremento h. El error de discretización de esta aproximación es proporcional a h^2 (vea el ejercicio 5.1.21.). La estrategia a emplear será la consideración de la viga como la unión de muchos segmentos de longitud h, y la aplicación de la versión discretizada de la ecuación diferencial en cada segmento.

Para un entero positivo n, establezca $h = L/n$. Considere la cuadrícula uniformemente espaciada $0 = x_0 < x_1 < \ldots < x_n = L$, donde $h = x_i - x_{i-1}$ para $i = 1, \ldots, n$. Al remplazar la ecuación diferencial (2.27) con la aproximación por diferencias (2.28) a fin de obtener el sistema de ecuaciones lineales para los desplazamientos $y_i = y(x_i)$, se obtiene

$$y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2} = \frac{h^4}{EI} f(x_i). \tag{2.29}$$

Se desarrollarán n ecuaciones con las n incógnitas $y_1, \ldots, y_n$. La matriz de coeficientes, o matriz de estructura, tendrá los coeficientes de la parte izquierda de esta ecuación. Sin embargo, observe que es necesario alterar las ecuaciones cerca de los extremos de la viga para tomar en cuenta las condiciones de frontera.

Un trampolín es una viga con un extremo sujeto en el soporte y con el otro extremo libre. Esto se llama viga **anclada-libre** o viga en voladizo. Las condiciones de frontera para el extremo anclado (izquierda) y el extremo libre (derecha) son

$$y(0) = y'(0) = y''(L) = y'''(L) = 0.$$

En particular, $y_0 = 0$. Sin embargo, observe que la determinación de y_1 presenta un problema, puesto que la aplicación de la aproximación (2.29) a la ecuación diferencial (2.27) en x_1 resulta en

$$y_{-1} - 4y_0 + 6y_1 - 4y_2 + y_3 = \frac{h^4}{EI} f(x_1), \tag{2.30}$$

y y_{-1} no está definida. En su lugar, debe utilizarse una aproximación alternativa a la derivada en un punto x_1 cercano al extremo fijo. En el ejercicio 5.1.22(a) se obtiene la aproximación

$$y'''(x_1) \approx \frac{16y(x_1) - 9y(x_1 + h) + \frac{8}{3}y(x_1 + 2h) - \frac{1}{4}y(x_1 + 3h)}{h^4} \tag{2.31}$$

que es válida cuando $y(x_0) = y'(x_0) = 0$.

Por ahora, si la aproximación se considera "válida", entonces el error de discretización de la aproximación es proporcional a h^2, lo mismo que para la ecuación (2.28). En teoría, esto significa que el error al aproximar la derivada de esta manera se reducirá a cero en el límite de una h pequeña. Este concepto será el punto focal del análisis de la diferenciación numérica en el capítulo 5. El resultado aquí es que la aproximación (2.31) puede utilizarse para tomar en cuenta la condición del extremo para $i = 1$, de donde resulta

$$16y_1 - 9y_2 + \frac{8}{3}y_3 - \frac{1}{4}y_4 = \frac{h^4}{EI} f(x_1).$$

El extremo libre a la derecha de la viga requiere un poco más de trabajo porque deben calcularse todas las y_i hasta el extremo de la viga. Una vez más, se requieren aproximaciones alternativas de las derivadas en los dos últimos puntos x_{n-1} y x_n. En el ejercicio 5.1.22 se proporcionan las aproximaciones

$$y''''(x_{n-1}) \approx \frac{-28y_n + 72y_{n-1} - 60y_{n-2} + 16y_{n-3}}{17h^4} \tag{2.32}$$

$$y'''(x_n) \approx \frac{72y_n - 156y_{n-1} + 96y_{n-2} - 12y_{n-3}}{17h^4} \tag{2.33}$$

que son válidas bajo el supuesto $y''(x_n) = y'''(x_n) = 0$.

Ahora es posible escribir el sistema de n ecuaciones con n incógnitas para el trampolín. Esta ecuación matricial resume las versiones aproximadas de la ecuación diferencial original (2.27) en cada punto $x_1, \ldots, x_n$, exactas en términos de orden h^2.

$$\begin{bmatrix} 16 & -9 & \frac{8}{3} & -\frac{1}{4} & & & & & \\ -4 & 6 & -4 & 1 & & & & & \\ 1 & -4 & 6 & -4 & 1 & & & & \\ & 1 & -4 & 6 & -4 & 1 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & 1 & -4 & 6 & -4 & 1 & \\ & & & & 1 & -4 & 6 & -4 & 1 \\ & & & & & \frac{16}{17} & -\frac{60}{17} & \frac{72}{17} & -\frac{28}{17} \\ & & & & & -\frac{12}{17} & \frac{96}{17} & -\frac{156}{17} & \frac{72}{17} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \frac{h^4}{EI} \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ \vdots \\ \vdots \\ f(x_{n-1}) \\ f(x_n) \end{bmatrix}. \tag{2.34}$$

La matriz de estructura A en (2.34) es una **matriz en bandas**, lo que significa que todas las entradas suficientemente lejos de la diagonal principal son iguales a cero. En concreto, las entradas a_{ij} de la matriz son iguales a 0, excepto para $|i-j| \leq 3$. El **ancho de banda** de la matriz en bandas es 7, puesto que $i - j$ toma 7 valores para las a_{ij} distintas de cero.

Por último, ya es posible modelar la viga anclada-libre. Considere un trampolín de madera sólida hecho de abeto Douglas. Suponga que el trampolín tiene una longitud de $L = 2$ metros, 30 cm de ancho y 3 cm de espesor. La densidad del abeto de Douglas es de aproximadamente 480 kg/m^3. Un Newton de fuerza es 1 kg-m/seg^2, y el módulo de Young para esta madera es de aproximadamente $E = 1{,}3 \times 10^{10}$ Pascales, o Newton/m^2. El momento de inercia de área I alrededor del centro de masa de una viga es $wd^3/12$, donde w es el ancho y d el espesor de la viga.

Se empezará calculando el desplazamiento de la viga sin carga, de modo que $f(x)$ representa sólo el peso propio de la viga, en unidades de fuerza por metro. Por lo tanto, $f(x)$ es la masa por metro $480wd$ veces la aceleración de la gravedad $-g = -9{,}81$ m/s^2, o la constante $f(x) = f = -480wdg$. El lector debe verificar que las unidades coincidan a ambos lados de (2.27). Existe una solución de forma cerrada para (2.27) en el caso cuando f es constante, de modo que es posible comprobar la exactitud del resultado de su cálculo.

Tras la comprobación de su código para la viga sin carga, modelará otros dos casos. En el primero, se añadirá a la viga una carga sinusoidal (o "pila"). En este caso, existe de nuevo una solución conocida de forma cerrada, pero las aproximaciones a las derivadas no son exactas, así que usted será capaz de controlar el error de su modelo como una función del tamaño de la cuadrícula h, y ver el efecto de los problemas de condicionamiento para una n grande. Después, colocará un clavadista sobre la viga.

Actividades sugeridas:

1. Escriba un programa en MATLAB para definir la matriz de estructura A en (2.34). Luego, utilizando el comando de MATLAB \ o un código de su propio diseño, resuelva el sistema para los desplazamientos y_i usando $n = 10$ pasos de la cuadrícula.

2. Grafique la solución del paso 1 contra la solución correcta $y(x) = (f/24EI)x^2(x^2 - 4Lx + 6L^2)$, donde $f = f(x)$ es la constante definida anteriormente. Verifique el error en el extremo de la viga, $x = L$ metros. En este caso simple, las aproximaciones a las derivadas son exactas, por lo que su error debe estar cerca del redondeo de máquina.

3. Realice de nuevo el cálculo en el paso 1 para $n = 10 \cdot 2^k$, donde $k = 1, \ldots, 11$. Haga una tabla de los errores en $x = L$ para cada n. ¿Para qué n el error es el más pequeño?, ¿por qué el error empieza a aumentar con n después de cierto punto? Es posible que desee hacer una tabla que acompañe al número de condición de A como una función de n para ayudar a responder la última pregunta. A fin de llevar a cabo este paso para una k grande, puede ser necesario solicitar a MATLAB que

almacene la matriz A como una matriz dispersa para evitar quedarse sin memoria. Para ello, basta con inicializar A con el comando `A = sparse(n, n)`, y proceda como antes. En la siguiente sección se analizarán las matrices dispersas con mayor detalle.

4. Añada una pila sinusoidal a la viga. Esto significa añadir una función de la forma $s(x) = -pg\,\text{sen}\frac{\pi}{L}x$ al término de fuerza $f(x)$. Pruebe que la solución

$$y(x) = \frac{f}{24EI}x^2(x^2 - 4Lx + 6L^2) - \frac{pgL}{EI\pi}\left(\frac{L^3}{\pi^3}\,\text{sen}\frac{\pi}{L}x - \frac{x^3}{6} + \frac{L}{2}x^2 - \frac{L^2}{\pi^2}x\right)$$

satisface la ecuación de Euler-Bernoulli y las condiciones de frontera anclada-libre.

5. Vuelva a realizar el cálculo como en el paso 3 para la carga sinusoidal. (Asegúrese de incluir el peso propio de la viga). Establezca $p = 100$ kg/m y grafique sus soluciones calculadas contra la solución correcta. Responda a las preguntas del paso 3, además de la siguiente: ¿el error en $x = L$ es proporcional a h^2 como se indicó anteriormente? Es posible que desee graficar el error contra h en una gráfica log-log a fin de investigar este asunto. ¿El número de condición entra en juego?

6. Ahora retire la carga sinusoidal y añada un clavadista de 70 kg a la viga, balanceándose en los últimos 20 cm de ésta. Debe agregar una fuerza por unidad de longitud $-g$ veces 70/0.2 kg/m a $f(x_i)$ para toda $1.8 \leq x_i \leq 2$ y resolver el problema de nuevo con el valor óptimo de n encontrado en el paso 5. Grafique la solución y encuentre la deflexión del trampolín en el extremo libre.

7. Si también se ancla el extremo libre del trampolín, se tiene una viga "anclada-anclada", que obedece a las mismas condiciones de frontera en cada extremo: $y(0) = y'(0) = y(L) = y'(L) = 0$. Esta versión se usa para modelar el pandeo de una estructura, como un puente. Comience con una cuadrícula uniformemente espaciada un poco diferente $0 = x_0 < x_1 < \ldots < x_n < x_{n+1} = L$, donde $h = x_i - x_{i-1}$ para $i = 1, \ldots, n$ y encuentre el sistema de n ecuaciones con n incógnitas que determina $y_1, \ldots, y_n$. (Debe ser similar a la versión anclada-libre, salvo que los dos últimos renglones de la matriz de coeficientes A deben ser los dos primeros renglones en orden inverso). Resuelva para una carga sinusoidal y responda a las preguntas del paso 5 para el centro $x = L/2$ de la viga. La solución exacta para la viga fija-fija bajo una carga sinusoidal es

$$y(x) = \frac{f}{24EI}x^2(L - x)^2 - \frac{pgL^2}{\pi^4 EI}\left(L^2\,\text{sen}\frac{\pi}{L}x + \pi x(x - L)\right).$$

8. Ideas para una exploración más profunda: si el ancho del trampolín se duplica, ¿cómo cambia el desplazamiento del clavadista?, ¿cambia más o menos que si el espesor se duplica? (ambas vigas tienen la misma masa). ¿Cómo cambia el desplazamiento máximo si la sección transversal es circular o anular con la misma área que la sección rectangular? (el momento de inercia del área para una sección transversal circular de radio r es $I = \pi r^4/4$, y para una sección transversal anular con radio interior r_1 y radio exterior r_2 es $I = \pi(r_2^4 - r_1^4)/4$). Averigüe el momento de inercia del área para las vigas I. Los módulos de Young para diferentes materiales también están tabulados y se encuentran disponibles. Por ejemplo, la densidad del acero es de unos 7850 kg/m^3 y su módulo de Young es de aproximadamente 2×10^{11} pascales.

La viga de Euler-Bernoulli es un modelo clásico, relativamente simple. Los modelos más recientes, como la viga de Timoshenko, toman en cuenta una flexión más inusual, donde la sección transversal de la viga no puede ser perpendicular al eje principal de la viga. ✔

2.5 MÉTODOS ITERATIVOS

La eliminación gaussiana es una secuencia finita de $O(n^3)$ operaciones de punto flotante que resultan en una solución. Por esa razón, esta eliminación se considera un método **directo** para resolver los sistemas de ecuaciones lineales. Los métodos directos, en teoría, proporcionan la solución exacta en un número finito de pasos. (Por supuesto, cuando se lleva a cabo mediante una computadora que utiliza una precisión limitada, la solución resultante será sólo aproximada. Como se vio anteriormente, la pérdida de exactitud se cuantifica mediante el número de condición). Los métodos directos contrastan con los métodos para la búsqueda de raíces descritos en el capítulo 1, que tienen una forma iterativa.

Los métodos denominados **iterativos** también pueden aplicarse a la resolución de los sistemas de ecuaciones lineales. De manera similar a la iteración de punto fijo, los métodos comienzan con un valor inicial y mejoran la estimación en cada paso, convergiendo hacia el vector solución.

2.5.1 Método de Jacobi

El método de Jacobi es una forma de iteración de punto fijo para un sistema de ecuaciones. En la IPF el primer paso consiste en reescribir las ecuaciones, despejando las incógnitas. El primer paso del método de Jacobi es hacer esto de la forma estandar que se indica a continuación: Resolver la i-ésima ecuación para la i-ésima incógnita. A continuación, iterar como en la iteración de punto fijo, a partir de una estimación inicial.

► EJEMPLO 2.19 Aplique el método de Jacobi al sistema $3u + v = 5, u + 2v = 5$.

Comience por despejar u de la primera ecuación y v de la segunda ecuación. Se usará el valor inicial $(u_0, v_0) = (0, 0)$. Se tiene

$$\begin{aligned} u &= \frac{5 - v}{3} \\ v &= \frac{5 - u}{2}. \end{aligned} \tag{2.35}$$

Se iteran las dos ecuaciones:

$$\begin{aligned} \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix} \\ \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-5/2}{3} \\ \frac{5-5/3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix} \\ \begin{bmatrix} u_3 \\ v_3 \end{bmatrix} &= \begin{bmatrix} \frac{5-5/3}{3} \\ \frac{5-5/6}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix}. \end{aligned} \tag{2.36}$$

Los pasos subsiguientes de Jacobi muestran la convergencia hacia la solución, que es $[1, 2]$. ◄

Ahora suponga que las ecuaciones se dan en el orden inverso.

► EJEMPLO 2.20 Aplique el método de Jacobi al sistema $u + 2v = 5, 3u + v = 5$.

Resuelva la primera ecuación para la primera variable u y la segunda ecuación para v. Inicie con

$$\begin{aligned} u &= 5 - 2v \\ v &= 5 - 3u. \end{aligned} \tag{2.37}$$

Las dos ecuaciones se iteran como antes, pero los resultados son bastante diferentes:

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 5 - 2v_0 \\ 5 - 3u_0 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$
$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 5 - 2v_1 \\ 5 - 3u_1 \end{bmatrix} = \begin{bmatrix} -5 \\ -10 \end{bmatrix}$$
$$\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 5 - 2(-10) \\ 5 - 3(-5) \end{bmatrix} = \begin{bmatrix} 25 \\ 20 \end{bmatrix}. \tag{2.38}$$

En este caso, el método de Jacobi falla, puesto que la iteración diverge. ◄

Como el método de Jacobi no siempre tiene éxito, resulta útil conocer las condiciones bajo las cuales funciona. Una condición importante se da en la siguiente definición:

DEFINICIÓN 2.9 La matriz $A = (a_{ij})$ de $n \times n$ es de **diagonal estrictamente dominante** si, para cada $1 \leq i \leq n$, $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$. En otras palabras, cada entrada diagonal principal domina su renglón en el sentido de que es mayor en magnitud que la suma de las magnitudes del resto de las entradas en ese renglón. ❐

TEOREMA 2.10 Si la matriz A de $n \times n$ es de diagonal estrictamente dominante, entonces (1) A es una matriz no singular y (2) para todo vector b y toda estimación inicial, el método de Jacobi aplicado a $Ax = b$ converge a la solución (única). ■

El teorema 2.10 dice que, si A es de diagonal estrictamente dominante, entonces el método de Jacobi aplicado a la ecuación $Ax = b$ converge a una solución para cada estimación inicial. La prueba de este hecho se da en la sección 2.5.3. En el ejemplo 2.19, la matriz de coeficientes es en un inicio

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix},$$

que es de diagonal estrictamente dominante porque $3 > 1$ y $2 > 1$. La convergencia está garantizada en este caso. Por otra parte, en el ejemplo 2.20, Jacobi se aplica a la matriz

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix},$$

que no es de diagonal dominante, y no existe tal garantía. Tenga en cuenta que la dominancia diagonal estricta es sólo una condición suficiente. El método de Jacobi puede converger aún en su ausencia.

► EJEMPLO 2.21 Determine si las matrices

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & -5 & 2 \\ 1 & 6 & 8 \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} 3 & 2 & 6 \\ 1 & 8 & 1 \\ 9 & 2 & -2 \end{bmatrix}$$

son de diagonal estrictamente dominante.

La matriz A es diagonalmente dominante porque $|3| > + |1| + |-1|$, $|-5| > |2| + |2|$, y $|8| > |1| + |6|$. B no lo es porque, por ejemplo, $|3| > 2 + |6|$ no es verdad. Sin embargo, si la primera y tercera filas de B se intercambian, entonces B es de diagonal estrictamente dominante y se garantiza la convergencia de Jacobi. ◄

El método de Jacobi es una forma de iteración de punto fijo. Si D indica la diagonal principal de A, L indica la matriz triangular inferior de A (entradas debajo de la diagonal principal), y U indica la matriz triangular superior (entradas arriba de la diagonal principal). Entonces $A = L + D + U$, y la ecuación que debe resolverse es $Lx + Dx + Ux = b$. Observe que este uso de L y U difiere del uso en la factorización LU, puesto que todos los elementos de las diagonales de estas L y U son cero. El sistema de ecuaciones $Ax = b$ puede reordenarse en una iteración de punto fijo de la forma:

$$\begin{aligned} Ax &= b \\ (D + L + U)x &= b \\ Dx &= b - (L + U)x \\ x &= D^{-1}(b - (L + U)x). \end{aligned} \tag{2.39}$$

Como D es una matriz diagonal, su inversa es la matriz recíproca de las entradas diagonales de A. El método de Jacobi es justo la iteración de punto fijo de (2.39):

Método de Jacobi

$$\begin{aligned} x_0 &= \text{vector inicial} \\ x_{k+1} &= D^{-1}(b - (L + U)x_k) \textbf{ para } k = 0, 1, 2, \ldots \end{aligned} \tag{2.40}$$

Para el ejemplo 2.19,

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix},$$

la iteración de punto fijo (2.40) con $x_k = \begin{bmatrix} u_k \\ v_k \end{bmatrix}$ es

$$\begin{aligned} \begin{bmatrix} u_{k+1} \\ v_{k+1} \end{bmatrix} &= D^{-1}(b - (L + U)x_k) \\ &= \begin{bmatrix} 1/3 & 0 \\ 0 & 1/2 \end{bmatrix} \left(\begin{bmatrix} 5 \\ 5 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_k \\ v_k \end{bmatrix} \right) \\ &= \begin{bmatrix} (5 - v_k)/3 \\ (5 - u_k)/2 \end{bmatrix}, \end{aligned}$$

lo que concuerda con la versión original.

2.5.2 Método de Gauss-Seidel y SRS

En estrecha relación con el método de Jacobi existe una iteración llamada método de **Gauss-Seidel**. La única diferencia entre Gauss-Seidel y Jacobi es que en el primero, los valores más recientemente actualizados de las incógnitas se utilizan en cada paso, incluso si la actualización se produce en el paso actual. De regreso al ejemplo 2.19, Gauss-Seidel se ve como sigue:

$$\begin{aligned} \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-5/3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{3} \end{bmatrix} \\ \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_2}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-5/3}{3} \\ \frac{5-10/9}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{35}{18} \end{bmatrix} \\ \begin{bmatrix} u_3 \\ v_3 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_2}{3} \\ \frac{5-u_3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-35/18}{3} \\ \frac{5-55/54}{2} \end{bmatrix} = \begin{bmatrix} \frac{55}{54} \\ \frac{215}{108} \end{bmatrix}. \end{aligned} \tag{2.41}$$

Observe la diferencia entre Gauss-Seidel y Jacobi: la definición de v_1 utiliza u_1, no u_0. La aproximación a la solución [1, 2] se aprecia como con el método de Jacobi, pero con mayor exactitud en el mismo número de pasos. Con frecuencia, Gauss-Seidel converge más rápido que el método de Jacobi cuando éste es convergente. Con el teorema 2.11 se verifica que el método de Gauss-Seidel, como el de Jacobi, converge a la solución siempre y cuando la matriz de coeficientes sea de diagonal estrictamente dominante.

Gauss-Seidel puede escribirse en forma matricial y se identifica como una iteración de punto fijo en la que se aísla la ecuación $(L + \mathrm{D} + U)x = b$ como

$$(L + D)x_{k+1} = -Ux_k + b.$$

Tenga en cuenta que el uso de entradas recién determinadas de x_{k+1} se acomoda al incluir el triángulo inferior de A en el lado izquierdo. Reordenando la ecuación, se obtiene el método de Gauss-Seidel.

Método de Gauss-Seidel

$$x_0 = \text{vector inicial}$$
$$x_{k+1} = D^{-1}(b - Ux_k - Lx_{k+1}) \textbf{ para } k = 0, 1, 2, \ldots.$$

► EJEMPLO 2.22 Aplique el método de Gauss-Seidel al sistema

$$\begin{bmatrix} 3 & 1 & -1 \\ 2 & 4 & 1 \\ -1 & 2 & 5 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}.$$

La iteración de Gauss-Seidel es

$$u_{k+1} = \frac{4 - v_k + w_k}{3}$$
$$v_{k+1} = \frac{1 - 2u_{k+1} - w_k}{4}$$
$$w_{k+1} = \frac{1 + u_{k+1} - 2v_{k+1}}{5}.$$

A partir de $x_0 = [u_0, v_0, w_0] = [0, 0, 0]$, se calcula

$$\begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} \frac{4-0-0}{3} = \frac{4}{3} \\ \frac{1-8/3-0}{4} = -\frac{5}{12} \\ \frac{1+4/3+5/6}{5} = \frac{19}{30} \end{bmatrix} \approx \begin{bmatrix} 1.3333 \\ -0.4167 \\ 0.6333 \end{bmatrix}$$

y

$$\begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{101}{60} \\ -\frac{3}{4} \\ \frac{251}{300} \end{bmatrix} \approx \begin{bmatrix} 1.6833 \\ -0.7500 \\ 0.8367 \end{bmatrix}.$$

El sistema es de diagonal estrictamente dominante y, por lo tanto, la iteración converge a la solución [2, −1, 1]. ◄

El método llamado **Sobre-Relajación Sucesiva (SRS)** toma la dirección de Gauss-Seidel hacia la solución y lo "rebasa" para tratar de acelerar la convergencia. Sea w un número real y defina

cada componente de la nueva estimación x_{k+1} como una media ponderada de w veces la fórmula de Gauss-Seidel y $1 - w$ veces la estimación actual x_k. El número w se conoce como el **parámetro de relajación**, y $w > 1$ se conoce como **sobre-relajación**.

► EJEMPLO 2.23 Aplique la SRS con $w = 1.25$ para el sistema del ejemplo 2.22.

La sobre-relajación sucesiva produce

$$u_{k+1} = (1 - \omega)u_k + \omega\frac{4 - v_k + w_k}{3}$$
$$v_{k+1} = (1 - \omega)v_k + \omega\frac{1 - 2u_{k+1} - w_k}{4}$$
$$w_{k+1} = (1 - \omega)w_k + \omega\frac{1 + u_{k+1} - 2v_{k+1}}{5}.$$

A partir de $[u_0, v_0, w_0] = [0, 0, 0]$, se calcula

$$\begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} \approx \begin{bmatrix} 1.6667 \\ -0.7292 \\ 1.0312 \end{bmatrix}$$

y

$$\begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} \approx \begin{bmatrix} 1.9835 \\ -1.0672 \\ 1.0216 \end{bmatrix}.$$

En este ejemplo, la iteración SRS converge más rápido que Jacobi y Gauss-Seidel a la solución $[2, -1, 1]$. ◄

Al igual que con Jacobi y Gauss-Seidel, es posible una derivación alternativa de la SRS si se trata al sistema como un problema de punto fijo. El problema $Ax = b$ puede escribirse como $(L + D + U)x = b$ y, tras la multiplicación por w y el reordenamiento,

$$(\omega L + \omega D + \omega U)x = \omega b$$
$$(\omega L + D)x = \omega b - \omega U x + (1 - \omega)Dx$$
$$x = (\omega L + D)^{-1}[(1 - \omega)Dx - \omega U x] + \omega(D + \omega L)^{-1}b.$$

Sobre-Relajación Sucesiva (SRS)

x_0 = vector inicial

$x_{k+1} = (\omega L + D)^{-1}[(1 - \omega)Dx_k - \omega U x_k] + \omega(D + \omega L)^{-1}b$ **para** $k = 0, 1, 2, \ldots$.

La SRS con $w = 1$ es exactamente igual que Gauss-Seidel. También está permitido que el parámetro w sea menor que 1, en un método llamado Sub-Relajación Sucesiva.

► EJEMPLO 2.24 Compare Jacobi, Gauss-Seidel y SRS aplicados sobre el sistema de seis ecuaciones con seis incógnitas:

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & \frac{1}{2} \\ -1 & 3 & -1 & 0 & \frac{1}{2} & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & \frac{1}{2} & 0 & -1 & 3 & -1 \\ \frac{1}{2} & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} \frac{5}{2} \\ \frac{3}{2} \\ 1 \\ 1 \\ \frac{3}{2} \\ \frac{5}{2} \end{bmatrix}. \tag{2.42}$$

La solución es $x = [1, 1, 1, 1, 1, 1]$. En la siguiente tabla se muestran los vectores x_6 de la solución aproximada, después de ejecutar seis pasos de cada uno de los tres métodos:

Jacobi	Gauss–Seidel	SOR
0.9879	0.9950	0.9989
0.9846	0.9946	0.9993
0.9674	0.9969	1.0004
0.9674	0.9996	1.0009
0.9846	1.0016	1.0009
0.9879	1.0013	1.0004

El parámetro w para la sobre-relajación sucesiva se fijó en 1.1. La SRS parece ser superior para este problema. ◀

En la figura 2.3 se compara el error de la norma infinito para el ejemplo 2.24 después de seis iteraciones con diferentes w. Aunque no existe una teoría general que describa la mejor opción de w, resulta claro que existe una mejor opción en este caso. Consulte en Ortega [1972] un análisis de la w óptima en algunos casos especiales comunes.

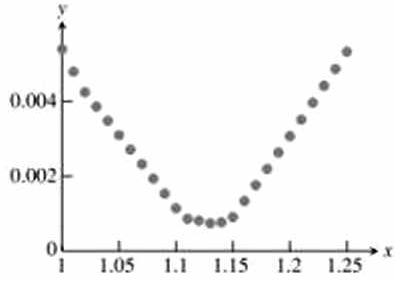


Figura 2.3 Error de la norma infinito después de seis pasos de la SRS en el ejemplo 2,24, como una función del parámetro de sobre-relajación ω. Gauss-Seidel corresponde a $\omega = 1$. El error mínimo se produce para $\omega \approx 1.13$

2.5.3 Convergencia de los métodos iterativos

En esta sección se demuestra que los métodos de Jacobi y Gauss-Seidel convergen para matrices de diagonal estrictamente dominante. Éste es el contenido de los teoremas 2.10 y 2.11.

El método de Jacobi se escribe como

$$x_{k+1} = -D^{-1}(L + U)x_k + D^{-1}b. \tag{2.43}$$

El teorema A.7 del apéndice A abarca la convergencia de este tipo de iteración. De acuerdo con dicho teorema, es necesario saber que el radio espectral $\rho(D^{-1}(L + U)) < 1$ con el fin de garantizar la convergencia del método de Jacobi. Esto es exactamente lo que implica la dominancia diagonal estricta, como se muestra a continuación.

Demostración del teorema 2.10. Sea $R = L + U$ la parte no diagonal de la matriz. Para comprobar que $\rho(D^{-1}R) < 1$, sea λ un valor característico de $D^{-1}R$ un vector característico correspondiente v. Elija este v de modo que $\|v\|_\infty = 1$, por lo que para alguna $1 \leq m \leq n$, el componente $v_m = 1$ y todos los demás componentes no sean mayores que 1. (Esto puede lograrse comenzando con cualquier vector característico y dividiéndolo entre el componente más grande. Cualquier múltiplo constante de un vector característico es de nuevo un vector característico con el mismo valor característico). La definición de valor característico significa que $D^{-1}Rv = \lambda v$, o $Rv = \lambda Dv$.

Como $r_{mm} = 0$, si se toman los valores absolutos de la m-ésima componente de esta ecuación vectorial se tiene

$$|r_{m1}v_1 + r_{m2}v_2 + \cdots + r_{m,m-1}v_{m-1} + r_{m,m+1}v_{m+1} + \cdots + r_{mn}v_n| = |\lambda d_{mm}v_m| = |\lambda||d_{mm}|.$$

Dado que todos los $|v_i| \leq 1$, el lado izquierdo es como máximo $\sum_{j \neq m} |r_{mj}|$, que, de acuerdo con la hipótesis de dominancia diagonal estricta, es menor que $|d_{mm}|$. Esto implica que $|\lambda||d_{mm}| < |d_{mm}|$, lo que a su vez obliga a que $|\lambda| < 1$. Puesto que λ es un valor característico arbitrario, se ha demostrado que $\rho(D^{-1}R) < 1$, tal como se deseaba. Ahora el teorema A.7 del apéndice A implica que Jacobi converge a una solución de $Ax = b$. Por último, como $Ax = b$ tiene una solución para una b arbitraria, A es una matriz no singular.

Si se escribe el método de Gauss-Seidel en la forma de (2.43), resulta

$$x_{k+1} = -(L + D)^{-1}Ux_k + (L + D)^{-1}b.$$

Entonces es evidente que la convergencia de Gauss-Seidel se presenta si el radio espectral de la matriz

$$(L + D)^{-1}U \tag{2.44}$$

es menor que uno. El siguiente teorema muestra que la dominancia diagonal estricta implica que este requisito se impone sobre los valores característicos.

TEOREMA 2.11 Si la matriz A de $n \times n$ es de diagonal estrictamente dominante, entonces (1) A es una matriz no singular y (2) para cada vector b y cada estimación inicial, el método de Gauss-Seidel aplicado a $Ax = b$ converge a una solución. ■

Demostración. Sea λ un valor característico de (2.44), con el correspondiente vector característico v. Elija el vector característico de modo que $v_m = 1$ y que todos los demás componentes sean de menor magnitud, como en la demostración anterior. Tenga en cuenta que las entradas de L son las a_{ij} para $i > j$, y las entradas de U son los a_{ij} para $i < j$. Entonces, viendo el renglón m de la ecuación de valor propio de (2.44),

$$\lambda(D + L)v = Uv,$$

se obtiene una cadena de desigualdades similar a la de la demostración anterior:

$$\begin{aligned}
|\lambda|\left(\sum_{i>m}|a_{mi}|\right) &< |\lambda|\left(|a_{mm}| - \sum_{i<m}|a_{mi}|\right) \\
&\leq |\lambda|\left(|a_{mm}| - \left|\sum_{i<m}a_{mi}v_i\right|\right) \\
&\leq |\lambda|\left|a_{mm} + \sum_{i<m}a_{mi}v_i\right| \\
&= \left|\sum_{i>m}a_{mi}v_i\right| \\
&\leq \sum_{i>m}|a_{mi}|.
\end{aligned}$$

De ello se deduce que $|\lambda| < 1$, con lo que concluye la demostración. ❐

2.5.4 Cálculos de matrices dispersas

Los métodos directos basados en la eliminación gaussiana proporcionan al usuario un número finito de pasos que terminan en la solución. ¿Cuál es la razón de emplear los métodos iterativos, que sólo son aproximados y pueden requerir varios pasos para la convergencia?

Existen dos razones principales para el uso de métodos iterativos como Gauss-Seidel. Ambos motivos se derivan del hecho de que un paso de un método iterativo requiere sólo una fracción de las operaciones de punto flotante de una factorización LU completa. Como se ha establecido antes en este capítulo, la eliminación gaussiana de una matriz de $n \times n$ implica aproximadamente n^3 operaciones. Por ejemplo, un solo paso del método de Jacobi requiere alrededor de n^2 multiplicaciones (una para cada entrada de la matriz) y aproximadamente el mismo número de sumas. La pregunta es: ¿cuántos pasos serán necesarios para la convergencia dentro de la tolerancia del usuario?

Una circunstancia particular que argumenta a favor de las técnicas iterativas es cuando ya se conoce una buena aproximación a la solución. Por ejemplo, suponga que se conoce una solución de $Ax = b$, después de lo cual A y/o b cambian en una pequeña cantidad. Es posible imaginar un problema dinámico donde A y b vuelven a medirse constantemente a medida que cambian, y constantemente se requiere una solución exacta x actualizada. Si la solución al problema anterior se utiliza como una estimación inicial para el nuevo, pero similar, problema, puede esperarse una convergencia rápida de Jacobi o Gauss-Seidel.

Suponga que la b del problema (2.42) se cambia ligeramente de la original $b = [2.5, 1.5, 1, 1, 1.5, 2.5]$ a una nueva $b = [2.2, 1.6, 0.9, 1.3, 1.4, 2.45]$. Se puede comprobar que la verdadera solución del sistema cambia de $[1, 1, 1, 1, 1, 1]$ a $[0.9, 1, 1, 1.1, 1, 1]$. Suponga que se tiene en memoria el sexto paso de la iteración de Gauss-Seidel x_6 de la tabla anterior, el cual se utilizará como una estimación inicial. Si Gauss-Seidel continúa con la nueva b y con la útil estimación inicial x_6, se obtiene una buena aproximación en un solo paso más. Los siguientes dos pasos son como lo muestra la tabla:

x_7	x_8
0.8980	0.8994
0.9980	0.9889
0.9659	0.9927
1.0892	1.0966
0.9971	1.0005
0.9993	1.0003

Esta técnica suele llamarse **pulido**, debido a que el método inicia con una solución aproximada, que puede ser la solución de un problema anterior relacionado; después, sólo se mejora la solución aproximada para que sea más exacta. El pulido es común en aplicaciones en tiempo real donde el mismo problema debe resolverse en varias ocasiones con datos actualizados a medida que pasa el tiempo. Si el sistema es grande y el tiempo es corto, puede ser imposible ejecutar toda una eliminación gaussiana o incluso una sustitución hacia atrás en el tiempo asignado. Si la solución no cambia demasiado, unos pocos pasos de un método iterativo relativamente sencillo podrían mantener la suficiente precisión mientras la solución se mueve a través del tiempo.

La segunda razón importante para utilizar los métodos iterativos es la resolución de los sistemas de ecuaciones dispersas. Una matriz de coeficientes se llama **dispersa** si se sabe que muchas de las entradas de la matriz son iguales a cero. Con frecuencia, de las n^2 entradas elegibles de una matriz dispersa, sólo $O(n)$ de ellas son distintas de cero. Una matriz llena es lo opuesto, donde algunas entradas pueden ser iguales a cero. La eliminación gaussiana aplicada a una matriz dispersa por lo general causa el efecto **de relleno**, donde la matriz de coeficientes cambia de dispersa a llena debido a las operaciones de filas requeridas. Por esta razón, la eficiencia de la eliminación gaussiana y su aplicación $PA = LU$ puede llegar a ser cuestionable para las matrices dispersas, con lo que los métodos iterativos se convierten en una alternativa viable.

El ejemplo 2.24 puede extenderse a una matriz dispersa de la siguiente manera:

► EJEMPLO 2.25 Use el método de Jacobi para resolver la versión de 100,000 ecuaciones del ejemplo 2.24.

Sea n un entero par y considere la matriz A de $n \times n$ con 3 en la diagonal principal, -1 la super y subdiagonal, y 1/2 en la posición $(i, n + 1 - i)$ para toda $i = 1, \ldots, n$, excepto para $i = n/2$ y $n/2 + 1$. Para $n = 12$,

$$A = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & -1 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 \end{bmatrix}. \quad (2.45)$$

Defina el vector $b = (2.5, 1.5, \ldots, 1.5, 1.0, 1.0, 1.5, \ldots, 1.5, 2.5)$, donde hay $n - 4$ repeticiones de 1.5 y 2 repeticiones de 1.0. Observe que si $n = 6$, A y b definen el sistema del ejemplo 2.24. La solución del sistema para n en general es $[1, \ldots, 1]$. Ningún renglón de A tiene más de 4 entradas diferentes de cero. Dado que menos de $4n$ de las n^2 entradas posibles son distintas de cero, la matriz A puede llamarse dispersa.

Si se desea resolver este sistema de ecuaciones para $n = 100.000$ o más, ¿cuáles son las opciones? Si se trata la matriz de coeficientes A como una matriz llena, significa almacenar $n^2 = 10^{10}$ entradas, cada una como un número de punto flotante de precisión doble que requiere 8 bytes de almacenamiento. Tenga en cuenta que 8×10^{10} bytes son aproximadamente 80 gigabytes. Dependiendo de su configuración computacional, puede ser imposible acomodar las n^2 entradas enteras en la memoria RAM.

No sólo el tamaño es un enemigo, también lo es el tiempo. El número de operaciones requeridas por la eliminación gaussiana será del orden de $n^3 \approx 10^{15}$. Si su computadora funciona en el orden de unos pocos GHz (109 ciclos por segundo), un límite superior en el número de operaciones en punto flotante por segundo es de alrededor de 10^8. Por lo tanto, $10^{15}/10^8 = 10^7$ es una suposición razonable del número de segundos que requiere la eliminación gaussiana. En un año hay 3×10^7 segundos. Aunque este conteo es aproximado, resulta evidente que la eliminación gaussiana para este problema no es un cálculo que pueda hacerse de la noche a la mañana.

Por otra parte, un paso de un método iterativo requerirá aproximadamente $2 \times 4n = 800,000$ operaciones, dos para cada entrada diferente de cero en la matriz. Podrían hacerse 100 pasos de la iteración de Jacobi y aun así terminar con menos de 10^8 operaciones, que pueden tomar alrededor de un segundo o menos en una PC moderna. Para el sistema que se acaba de definir, con $n = 100,000$, el siguiente código `jacobi.m` necesita sólo 50 pasos para converger desde una estimación inicial de $(0, \ldots, 0)$ hasta la solución $(1, \ldots, 1)$ con seis decimales correctos. Los 50 pasos requieren menos de 1 segundo en una PC típica.

```
% Programa 2.1 Configuración de matriz dispersa
% Entrada: n = tamaño del sistema
% Salidas: matriz dispersa a, lado derecho b
function [a,b] = sparsesetup(n)
e = ones(n,1); n2=n/2;
a = spdiags([-e 3*e -e],-1:1,n,n);    % Entradas de a
c=spdiags([e/2],0,n,n);c=fliplr(c);a=a+c;
a(n2+1,n2) = -1; a(n2,n2+1) = -1;     % Definición de 2 entradas
b=zeros(n,1);                         % Entradas del lado derecho b
b(1)=2.5;b(n)=2.5;b(2:n-1)=1.5;b(n2:n2+1)=1;
```

```
% Programa 2.2 Método de Jacobi
% Entradas: matrix llena o dispersa a,lado derecho b,
%         número de iteraciones de Jacobi, k
% Salida: solución x
function x = jacobi(a,b,k)
n=length(b);        % encuentra n
d=diag(a);          % extrae diagonal de a
r=a-diag(d);        % r es el residuo
x=zeros(n,1);       % inicializa el vector x
for j=1:k           % ciclo para la iteración de Jacobi
  x = (b-r*x)./d;
end                 % Fin del ciclo de la iteración de Jacobi
```

Observe algunos aspectos interesantes del código anterior. El programa `scarsesetup.m` utiliza el comando de MATLAB `spdiags`, que define la matriz A como una estructura de datos dispersos. En esencia, esto significa que la matriz está representada por un conjunto de triples (i, j, d), donde d es la entrada de un número real en la posición (i, j) de la matriz. No se reserva memoria para las n^2 entradas enteras posibles, sino sólo para aquellas que sean necesarias. El comando `spdiags` toma las columnas de una matriz y las coloca a lo largo de la diagonal principal, o en una determinada sub o super diagonal por debajo o por encima de la diagonal principal.

Los comandos de MATLAB que realizan la manipulación de matrices están diseñados para funcionar perfectamente con la estructura de datos de una matriz dispersa. Por ejemplo, una alternativa para el código anterior sería el comando de MATLAB `lu` para resolver el sistema de manera directa. Sin embargo, para este ejemplo, a pesar de que A es dispersa, la matriz triangular superior U que se deriva de la eliminación gaussiana experimenta el fenómeno de relleno durante el proceso. Por ejemplo, la U triangular superior de la eliminación gaussiana para un tamaño $n = 12$ de la matriz A anterior es

$$\begin{bmatrix}
3 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.500 \\
0 & 2.7 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.500 & 0.165 \\
0 & 0 & 2.6 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0.500 & 0.187 & 0.062 \\
0 & 0 & 0 & 2.6 & -1.000 & 0 & 0 & 0 & 0.500 & 0.191 & 0.071 & 0.024 \\
0 & 0 & 0 & 0 & 2.618 & -1.000 & 0 & 0.500 & 0.191 & 0.073 & 0.027 & 0.009 \\
0 & 0 & 0 & 0 & 0 & 2.618 & -1.000 & 0.191 & 0.073 & 0.028 & 0.010 & 0.004 \\
0 & 0 & 0 & 0 & 0 & 0 & 2.618 & -0.927 & 0.028 & 0.011 & 0.004 & 0.001 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.562 & -1.032 & -0.012 & -0.005 & -0.001 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.473 & -1.047 & -0.018 & -0.006 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.445 & -1.049 & -0.016 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.440 & -1.044 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.458
\end{bmatrix}.$$

Como U se convierte en una matriz relativamente llena, las restricciones de memoria mencionadas antes vuelven a ser una limitación. Se requerirá una fracción significativa de las n^2 posiciones de memoria para almacenar U en el camino de completar el proceso de solución. Resulta más eficiente, por varios órdenes de magnitud del tiempo de ejecución y del almacenamiento, resolver este gran sistema disperso mediante un método iterativo. ◄

2.5 Ejercicios

1. Calcule los dos primeros pasos de los métodos de Jacobi y Gauss-Seidel con el vector inicial $[0, \ldots, 0]$.

(a) $\begin{bmatrix} 3 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$ (b) $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$

(c) $\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \\ 5 \end{bmatrix}$

2. Reordene las ecuaciones para formar un sistema de diagonal estrictamente dominante. Aplique dos pasos de los métodos de Jacobi y Gauss-Seidel con el vector inicial [0, ... , 0].

$$\text{(a)}\quad \begin{aligned} u + 3v &= -1 \\ 5u + 4v &= 6 \end{aligned} \qquad \text{(b)}\quad \begin{aligned} u - 8v - 2w &= 1 \\ u + v + 5w &= 4 \\ 3u - v + w &= -2 \end{aligned} \qquad \text{(c)}\quad \begin{aligned} u + 4v &= 5 \\ v + 2w &= 2 \\ 4u + 3w &= 0 \end{aligned}$$

3. Aplique dos pasos de la SRS a los sistemas del ejercicio 1. Utilice el vector inicial [0, ... , 0] y $w = 1.5$.

4. Aplique dos pasos de la SRS a los sistemas del ejercicio 2 después de reordenarlo. Utilice el vector inicial [0, ... , 0] y $w = 1$ y 1.2.

5. Sea λ un valor característico de una matriz A de $n \times n$. (a) Demuestre el teorema del círculo de Gershgorin: Existe una entrada diagonal A_{mm} de tal forma que $|A_{mm} - \lambda| \leq \sum_{j \neq m} |A_{mj}|$. (Sugerencia: comience con un vector propio v de tal forma que $\|v\|_\infty = 1$, como en la demostración del teorema 2.10). (b) Demuestre que una matriz de diagonal estrictamente dominante no puede tener un valor característico igual a cero. Ésta es una demostración alternativa de la parte (1) del teorema 2.10.

2.5 Problemas de computadora

1. Utilice el método de Jacobi para resolver el sistema disperso con seis posiciones decimales correctas (error hacia adelante en la norma infinito) para $n = 100$ y $n = 100000$. La solución correcta es [1, ... , 1]. Indique el número de pasos necesarios y el error hacia atrás. El sistema es

$$\begin{bmatrix} 3 & -1 & & & & \\ -1 & 3 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 3 & -1 & \\ & & & -1 & 3 & \end{bmatrix} \begin{bmatrix} x_1 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ \vdots \\ 1 \\ 2 \end{bmatrix}.$$

2. Utilice el método de Jacobi para resolver el sistema disperso con tres posiciones decimales correctas (error hacia adelante en la norma infinito) para $n = 100$. La solución correcta es [1, −1, 1, −1, ..., 1, −1]. Indique el número de pasos necesarios y el error hacia atrás. El sistema es

$$\begin{bmatrix} 2 & 1 & & & & \\ 1 & 2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & 2 & 1 & \\ & & & 1 & 2 & \end{bmatrix} \begin{bmatrix} x_1 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}.$$

3. Reescriba el programa 2.2 para realizar la iteración de Gauss-Seidel. Resuelva el problema del ejemplo 2.24 para verificar su trabajo.

4. Reescriba el programa 2.2 para llevar a cabo la SRS. Utilice $w = 1.1$ y vuelva a comprobar el ejemplo 2.24.

5. Realice los pasos del problema de computadora 1 con $n = 100$ para (a) el método de Gauss-Seidel y (b) la SRS con $w = 1.2$.

6. Realice los pasos del problema de computadora 2 para (a) el método de Gauss-Seidel y (b) la SRS con $w = 1.5$.

7. Use el programa del problema de computadora 3 y decida qué tan grande puede ser un sistema del tipo (2.38), si éste debe resolverse con precisión mediante el método de Gauss-Seidel en el primer segundo de cálculo. Indique el tiempo requerido y el error hacia adelante para diversos valores de n.

2.6 MÉTODOS PARA MATRICES SIMÉTRICAS DEFINIDAS POSITIVAS

Las matrices simétricas ocupan una posición favorable en el análisis de sistemas lineales, debido a su estructura especial y porque tienen sólo la mitad de las entradas independientes que poseen las matrices generales. Esto plantea la pregunta de si una factorización como la LU puede realizarse para un cálculo con la mitad de complejidad, y utilizando sólo la mitad de las posiciones de memoria. Para las matrices simétricas definidas positivas, este objetivo puede lograrse mediante la factorización de Cholesky.

Las matrices simétricas definidas positivas también permiten un enfoque bastante diferente para resolver $Ax = b$, el cual no depende de una factorización de matrices. Este nuevo enfoque, llamado el método del gradiente conjugado, es muy útil para las matrices grandes y dispersas, puesto que pertenece a la familia de los métodos iterativos.

Para comenzar la sección, se define el concepto de matrices simétricas definidas positivas. Después, se muestra que cualquier matriz simétrica definida positiva A puede factorizarse como $A = R^T R$ con R una matriz triangular superior, la factorización de Cholesky. Como resultado, el problema $Ax = b$ puede resolverse empleando dos sustituciones hacia atrás, igual que con la factorización LU en el caso no simétrico. La sección se cierra con el algoritmo de gradiente conjugado y una introducción al precondicionamiento.

2.6.1 Matrices simétricas definidas positivas

DEFINICIÓN 2.12 La matriz A de $n \times n$ es **simétrica** si $A^T = A$. La matriz A es **definida positiva** si $x^T Ax > 0$ para todos los vectores $x \neq 0$. ❐

► EJEMPLO 2.26 Demuestre que la matriz $A = \begin{bmatrix} 2 & 2 \\ 2 & 5 \end{bmatrix}$ es simétrica definida positiva.

Es evidente que A es simétrica. Para demostrar que es definida positiva, se aplica la siguiente definición:

$$\begin{aligned} x^T Ax &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= 2x_1^2 + 4x_1x_2 + 5x_2^2 \\ &= 2(x_1 + x_2)^2 + 3x_2^2 \end{aligned}$$

Esta expresión siempre es no negativa y no puede ser cero a menos que tanto $x_2 = 0$ y $x_1 + x_2 = 0$, lo que en conjunto implica que $x = 0$.

► EJEMPLO 2.27 Demuestre que la matriz simétrica $A = \begin{bmatrix} 2 & 4 \\ 4 & 5 \end{bmatrix}$ no es definida positiva.

Calcule $x^T Ax$ completando el cuadro:

$$\begin{aligned} x^T Ax &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= 2x_1^2 + 8x_1x_2 + 5x_2^2 \\ &= 2(x_1^2 + 4x_1x_2) + 5x_2^2 \\ &= 2(x_1 + 2x_2)^2 - 8x_2^2 + 5x_2^2 \\ &= 2(x_1 + 2x_2)^2 - 3x_2^2 \end{aligned}$$

Por ejemplo, si se establece $x_1 = -2$ y $x_2 = 1$, el resultado será menor que cero, lo que contradice la definición de definida positiva. ◄

Tenga en cuenta que una matriz simétrica definida positiva debe ser no singular, puesto que es imposible para un vector x distinto de cero satisfacer $Ax = 0$. Existen tres hechos adicionales importantes acerca de esta clase de matrices.

Propiedad 1 Si la matriz A de $n \times n$ es simétrica, entonces A es definida positiva si y sólo si todos sus valores característicos son positivos.

Demostración. El teorema A.5 dice que: el conjunto de vectores característicos es ortonormal y abarca R^n. Si A es definida positiva y $Av = \lambda v$ para un vector v distinto de cero, entonces $0 < v^T Av = v^T(\lambda v) = \lambda ||v||_2^2$, de modo que $\lambda > 0$. Por otra parte, si todos los valores característicos de A son positivos, entonces escriba cualquier $x = c_1 v_1 + \ldots + c_n v_n$ distinta de cero donde las v_i son los vectores ortonormales unitarios y no todas las c_i son cero. Entonces, $x^T Ax = (c_1 v_1 + \ldots + c_n v_n)^T (\lambda_1 c_1 v_1 + \ldots + \lambda_n c_n v_n) = \lambda_1 c_1^2 + \ldots + \lambda_n c_n^2 > 0$, por lo que A es definida positiva. ❐

Los valores característicos de A en el ejemplo 2.26 son 6 y 1. Los valores característicos de A en el ejemplo 2.27 son aproximadamente 7.77 y -0.77.

Propiedad 2 Si A es una matriz de $n \times n$, simétrica definida positiva y X es una matriz de $n \times m$ de rango completo con $n \geq m$, entonces $X^T AX$ es una matriz de $m \times m$ simétrica definida positiva.

Demostración. La matriz es simétrica puesto que $[X^T AX)^T = X^T AX$. Para probar que es definida positiva, considere un vector v no nulo de tamaño m. Note que $v^T (X^T AX)v = (Xv)^T A(Xv) \geq 0$, con igualdad sólo si $Xv = 0$, debido a que A es definida positiva. Como X es de rango completo, sus columnas son linealmente independientes, por lo que $Xv = 0$ implica que $v = 0$. ❐

DEFINICIÓN 2.13 Una submatriz **principal** de una matriz cuadrada A es una submatriz cuadrada cuyas entradas diagonales son las entradas diagonales de A. ❐

Propiedad 3 Cualquier submatriz principal de una matriz simétrica definida positiva es simétrica definida positiva.

Demostración. Ejercicio 12.

Por ejemplo, si

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

es simétrica definida positiva, entonces también lo es

$$\begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}.$$

2.6.2 Factorización de Cholesky

Para demostrar la idea principal, se inicia con un caso de 2×2. Todas las cuestiones importantes surgen allí; la extensión al tamaño general sólo requiere algunos cálculos adicionales.

Considere la matriz simétrica definida positiva

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix}.$$

Por la propiedad 3 de las matrices simétricas definidas positivas, se sabe que $a > 0$. Además, se sabe que el determinante $ac - b^2$ de A es positivo, puesto que el determinante es el producto de los valores propios, todos positivos por la propiedad 1. Si se escribe $A = R^T R$ con una R triangular superior, entonces se tiene la forma

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} \sqrt{a} & 0 \\ u & v \end{bmatrix} \begin{bmatrix} \sqrt{a} & u \\ 0 & v \end{bmatrix} = \begin{bmatrix} a & u\sqrt{a} \\ u\sqrt{a} & u^2 + v^2 \end{bmatrix},$$

y debe verificarse si esto es posible. Al comparar los lados izquierdo y derecho se obtienen las identidades $u = b/\sqrt{a}$ y $v^2 = c - u^2$. Observe que $v^2 = c - (b/\sqrt{a})^2 = c - b^2/a > 0$ a partir del conocimiento que se tiene del determinante. Esto verifica que v puede definirse como un número real por lo que la factorización de Cholesky

$$A = \begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} \sqrt{a} & 0 \\ \frac{b}{\sqrt{a}} & \sqrt{c - b^2/a} \end{bmatrix} \begin{bmatrix} \sqrt{a} & \frac{b}{\sqrt{a}} \\ 0 & \sqrt{c - b^2/a} \end{bmatrix} = R^T R$$

existe para matrices simétricas definidas positivas de 2×2. La factorización de Cholesky no es única; resulta claro que podría haberse elegido v como la raíz cuadrada negativa de $c - b^2/a$.

El siguiente resultado garantiza que esta misma idea funciona para el caso de $n \times n$.

TEOREMA 2.14 (Teorema de la factorización de Cholesky) Si A es una matriz simétrica definida positiva de $n \times n$, entonces existe una matriz R triangular superior de $n \times n$ de tal forma que $A = R^T R$. ■

Demostración. R se construye por inducción con un tamaño n. Anteriormente se abordó el caso $n = 2$. Considere la partición de A como

$$A = \left[\begin{array}{c|c} a & b^T \\ \hline b & C \end{array}\right]$$

donde b es un vector de tamaño $(n-1)$ y C es una submatriz de $(n-1) \times (n-1)$. Se usará la multiplicación de bloque (vea la sección A.2 del apéndice) para simplificar el argumento. Establezca $u = b/\sqrt{a}$ como en el caso de 2×2. Si $A_1 = C - uu^T$ y se define la matriz invertible

$$S = \left[\begin{array}{c|c} \sqrt{a} & u^T \\ \hline 0 & \\ \vdots & I \\ 0 & \end{array}\right]$$

se obtiene

$$S^T\left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A_1 & \\ 0 & & & \end{array}\right]S = \left[\begin{array}{c|ccc} \sqrt{a} & 0 & \cdots & 0 \\ \hline u & & I & \end{array}\right]\left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A_1 & \\ 0 & & & \end{array}\right]\left[\begin{array}{c|ccc} \sqrt{a} & & u^T & \\ \hline 0 & & & \\ \vdots & & I & \\ 0 & & & \end{array}\right]$$

$$= \left[\begin{array}{c|ccc} a & & b^T & \\ \hline b & & uu^T + A_1 & \end{array}\right] = A$$

Observe que A_1 es simétrica definida positiva. Esto se deduce de los hechos que

$$\left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A_1 & \\ 0 & & & \end{array}\right] = (S^T)^{-1}AS^{-1}$$

es simétrica definida positiva por la propiedad 2 y, por lo tanto, también lo es la submatriz principal A_1 de $(n-1)\times(n-1)$ por la propiedad 3. Debido a la hipótesis de inducción, $A_1 = V^TV$ donde V es triangular superior. Por último, defina la matriz triangular superior

$$R = \left[\begin{array}{c|ccc} \sqrt{a} & & u^T & \\ \hline 0 & & & \\ \vdots & & V & \\ 0 & & & \end{array}\right]$$

y verifique que

$$R^TR = \left[\begin{array}{c|ccc} \sqrt{a} & 0 & \cdots & 0 \\ \hline u & & V^T & \end{array}\right]\left[\begin{array}{c|ccc} \sqrt{a} & & u^T & \\ \hline 0 & & & \\ \vdots & & V & \\ 0 & & & \end{array}\right] = \left[\begin{array}{c|ccc} a & & b^T & \\ \hline b & & uu^T + V^TV & \end{array}\right] = A,$$

con lo que se completa la demostración.

La construcción de la demostración puede llevarse a cabo en forma explícita, en lo que se ha convertido en el algoritmo estándar para la factorización de Cholesky. La matriz R se construye desde afuera hacia adentro. Primero se encuentra $r_{11} = \sqrt{a_{11}}$ y se establece el resto del renglón superior de R hasta $u^T = b^T/r_{11}$. Después se resta uu^T de la submatriz principal inferior de $(n-1)\times(n-1)$, y se repiten los mismos pasos para llenar la segunda fila de R. Se continúa con estos pasos hasta que se determinan todos los renglones de R. De acuerdo con el teorema, la nueva submatriz principal es definida positiva en todas las etapas de la construcción; asimismo, por la propiedad 3, la entrada de la esquina superior izquierda es positiva, y la operación de raíz cuadrada tiene éxito. Este enfoque puede colocarse directamente dentro del siguiente algoritmo. Para indicar submatrices se utiliza la "notación de dos puntos" donde se considera conveniente.

Factorización de Cholesky

for $k = 1, 2, \ldots, n$
  if $A_{kk} < 0$, **stop, end**
  $R_{kk} = \sqrt{A_{kk}}$
  $u^T = \frac{1}{R_{kk}} A_{k,k+1:n}$
  $R_{k,k+1:n} = u^T$
  $A_{k+1:n,k+1:n} = A_{k+1:n,k+1:n} - uu^T$
end

La R resultante es triangular superior y satisface $A = R^T R$.

► EJEMPLO 2.28 Encuentre la factorización de Cholesky de $\begin{bmatrix} 4 & -2 & 2 \\ -2 & 2 & -4 \\ 2 & -4 & 11 \end{bmatrix}$.

El renglón superior de R es $R_{11} = \sqrt{a_{11}} = 2$, seguida por $R_{1,2:3} = [-2, 2]/R_{11} = [-1, 1]$:

$$R = \left[\begin{array}{c|cc} 2 & -1 & 1 \\ \hline & & \\ & & \end{array}\right].$$

Si se resta el producto externo $uu^T = \begin{bmatrix} -1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \end{bmatrix}$ de la submatriz principal inferior $A_{2:3,2:3}$ de 2×2 perteneciente a A, se obtiene

$$\left[\begin{array}{c|cc} & & \\ \hline & 2 & -4 \\ & -4 & 11 \end{array}\right] - \left[\begin{array}{c|cc} & & \\ \hline & 1 & -1 \\ & -1 & 1 \end{array}\right] = \left[\begin{array}{c|cc} & & \\ \hline & 1 & -3 \\ & -3 & 10 \end{array}\right].$$

Ahora se repiten los mismos pasos en la submatriz de 2×2 para encontrar $R_{22} = 1$ y $R_{23} = -3/1 = -3$

$$R = \left[\begin{array}{c|c|c} 2 & -1 & 1 \\ \hline & 1 & -3 \\ \hline & & \end{array}\right].$$

La submatriz principal inferior de 1×1 de A es $10 - (-3)(-3) = 1$, por lo que $R_{33} = \sqrt{1}$. El factor de Cholesky de A es

$$R = \begin{bmatrix} 2 & -1 & 1 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}.$$

Al resolver $Ax = b$ para una A simétrica definida positiva, se sigue la misma idea que la factorización LU. Ahora que $A = R^T R$ es un producto de dos matrices triangulares, es necesario resolver el sistema triangular inferior $R^T c = b$ y el sistema triangular superior $Rx = c$ para determinar la solución de x.

2.6.3 Método del gradiente conjugado

La introducción del método del gradiente conjugado (Hestenes y Steifel, 1952) marcó el comienzo de una nueva era en los métodos iterativos para resolver los problemas de matrices dispersas. Aunque el método se adoptó poco a poco, una vez que se desarrollaron los precondicionadores eficaces, los enormes problemas que no podían abordarse de otra manera se volvieron posibles. Este logro condujo en poco tiempo a enormes avances y a una nueva generación de solucionadores iterativos.

ANOTACIÓN

Ortogonalidad La primera aplicación real de la ortogonalidad en este libro se utiliza de una manera indirecta, para resolver un problema que no tiene relación evidente con la ortogonalidad. El método del gradiente conjugado sigue la solución de un sistema lineal definido positivo de $n \times n$ al localizar y eliminar sucesivamente los n componentes ortogonales del error, uno por uno. La complejidad del algoritmo se reduce al mínimo si se usan las indicaciones establecidas para los pares de vectores residuales ortogonales. Este punto de vista se desarrollará a mayor profundidad en el capítulo 4, terminando con el método GMRES, una contraparte no simétrica para gradientes conjugados.

Las ideas detrás de los gradientes conjugados recaen en la generalización de la idea habitual del producto interno. El producto interno euclidiano $(v, w) = v^T w$ es simétrico y lineal en las entradas de v y w, puesto que $(v, w) = (w, v)$ y $(\alpha v + \beta w, u) = \alpha(v, u) + \beta(w, u)$ para los escalares α y β. El producto interno euclidiano también es definido positivo, en que $(v, v) > 0$ si $v \neq 0$.

DEFINICIÓN 2.15 Sea A una matriz simétrica definida positiva de $n \times n$. Para dos vectores v y w de tamaño n, defina el producto interno de A

$$(v, w)_A = v^T A w.$$ ❐

La vectores v y w son un **conjugado** de A si $(v, w)_A = 0$.

Observe que el nuevo producto interior hereda las propiedades de simetría, linealidad y positividad de la matriz A. Como A es simétrica, también lo es el producto interior de A: $(v, w)_A = v^T A w = (v^T A w)^T = w^T A v = (w, v)_A$. El producto interior de A también es lineal y la positividad se deduce del hecho de que si A es definida positiva, entonces,

$$(v, v)_A = v^T A v > 0$$

si $v \neq 0$.

Hablando en sentido estricto, el método del gradiente conjugado es un método directo y llega a la solución x del sistema simétrico definido positivo $Ax = b$ con el siguiente ciclo finito:

Método del gradiente conjugado

$x_0 =$ valor inicial
$d_0 = r_0 = b - Ax_0$
for $k = 0, 1, 2, \ldots, n-1$
  if $r_k = 0$, **stop**, **end**
  $\alpha_k = \frac{r_k^T r_k}{d_k^T A d_k}$
  $x_{k+1} = x_k + \alpha_k d_k$
  $r_{k+1} = r_k - \alpha_k A d_k$
  $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
  $d_{k+1} = r_{k+1} + \beta_k d_k$
end

A continuación se presenta una descripción informal de la iteración, seguida de la demostración de los hechos necesarios en el teorema 2.16. La iteración del gradiente conjugado actualiza tres vectores diferentes en cada paso. El vector x_k es la solución aproximada en el paso k. El vector

rk representa el residuo de la solución aproximada x_k. Por definición, esto es claro para r_0; además, durante la iteración observe que

$$\begin{aligned} Ax_{k+1} + r_{k+1} &= A(x_k + \alpha_k d_k) + r_k - \alpha_k Ad_k \\ &= Ax_k + r_k, \end{aligned}$$

y así por inducción $r_k = b - Ax_k$ para todo k. Por último, el vector d_k representa la nueva dirección de búsqueda que se utiliza para actualizar la aproximación x_k con la versión mejorada x_{k+1}.

El método tiene éxito porque cada residuo está dispuesto para ser ortogonal a todos los residuos previos. Si esto puede hacerse, el método se ejecuta de las direcciones ortogonales en las cuales buscar, y debe llegarse a un residuo cero y a una solución correcta cuando mucho en n pasos. La clave para lograr la ortogonalidad entre los residuos resulta ser la elección de las direcciones de búsqueda d_k por pares conjugados. El concepto de conjugación generaliza la ortogonalidad y da nombre al algoritmo.

A continuación se explican las elecciones de α_k y β_k. Las direcciones d_k se eligen del intervalo de espacio vectorial de los residuos anteriores, como se ve de manera inductiva a partir de la última línea de pseudocódigo. Con el fin de asegurar que el siguiente residuo sea ortogonal a todos los residuos pasados, α_k se elige precisamente para que el nuevo residuo r_{k+1} sea ortogonal a la dirección d_k:

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k d_k \\ b - Ax_{k+1} &= b - Ax_k - \alpha_k Ad_k \\ r_{k+1} &= r_k - \alpha_k Ad_k \\ 0 = d_k^T r_{k+1} &= d_k^T r_k - \alpha_k d_k^T Ad_k \\ \alpha_k &= \frac{d_k^T r_k}{d_k^T Ad_k}. \end{aligned}$$

α_k no se escribe exactamente así en el algoritmo, pero observe que como d_{k-1} es ortogonal a r_k, se tiene

$$\begin{aligned} d_k - r_k &= \beta_{k-1} d_{k-1} \\ r_k^T d_k - r_k^T r_k &= 0, \end{aligned}$$

lo que justifica la reescritura $r_k^T d_k = r_k^T r_k$. En segundo lugar, el coeficiente β_k se elige para asegurar la conjugación por pares de la d_k de A:

$$\begin{aligned} d_{k+1} &= r_{k+1} + \beta_k d_k \\ 0 = d_k^T Ad_{k+1} &= d_k^T Ar_{k+1} + \beta_k d_k^T Ad_k \\ \beta_k &= -\frac{d_k^T Ar_{k+1}}{d_k^T Ad_k}. \end{aligned}$$

La expresión para β_k puede escribirse en la forma más sencilla vista en el algoritmo, como se muestra en (2.47) más adelante.

El teorema de 2.16 que se enuncia a continuación verifica que todas las r_k producidas por la iteración gradiente conjugada sean ortogonales entre sí. Puesto que son vectores de n dimensiones, a lo sumo n de las r_k pueden ser pares ortogonales, así que r_n o una r_k anterior debe ser igual a cero, al resolver $Ax = b$. Por lo tanto, después de realizar a lo sumo n pasos, el gradiente conjugado llega a una solución. En teoría, el método es directo, no iterativo.

Antes de pasar al teorema que garantiza el éxito del método del gradiente conjugado, es instructivo resolver un ejemplo en aritmética exacta.

► EJEMPLO 2.29 Resuelva $\begin{bmatrix} 2 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$ usando el método del gradiente conjugado.

Si se sigue el algoritmo anterior resulta

$$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, r_0 = d_0 = \begin{bmatrix} 6 \\ 3 \end{bmatrix}$$

$$\alpha_0 = \frac{\begin{bmatrix} 6 \\ 3 \end{bmatrix}^T \begin{bmatrix} 6 \\ 3 \end{bmatrix}}{\begin{bmatrix} 6 \\ 3 \end{bmatrix}^T \begin{bmatrix} 2 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 6 \\ 3 \end{bmatrix}} = \frac{45}{6 \cdot 18 + 3 \cdot 27} = \frac{5}{21}$$

$$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{5}{21} \begin{bmatrix} 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 10/7 \\ 5/7 \end{bmatrix}$$

$$r_1 = \begin{bmatrix} 6 \\ 3 \end{bmatrix} - \frac{5}{21} \begin{bmatrix} 18 \\ 27 \end{bmatrix} = 12 \begin{bmatrix} 1/7 \\ -2/7 \end{bmatrix}$$

$$\beta_0 = \frac{r_1^T r_1}{r_0^T r_0} = \frac{144 \cdot 5/49}{36 + 9} = \frac{16}{49}$$

$$d_1 = 12 \begin{bmatrix} 1/7 \\ -2/7 \end{bmatrix} + \frac{16}{49} \begin{bmatrix} 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 180/49 \\ -120/49 \end{bmatrix}$$

$$\alpha_1 = \frac{\begin{bmatrix} 12/7 \\ -24/7 \end{bmatrix}^T \begin{bmatrix} 12/7 \\ -24/7 \end{bmatrix}}{\begin{bmatrix} 180/49 \\ -120/49 \end{bmatrix}^T \begin{bmatrix} 2 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 180/49 \\ -120/49 \end{bmatrix}} = \frac{7}{10}$$

$$x_2 = \begin{bmatrix} 10/7 \\ 5/7 \end{bmatrix} + \frac{7}{10} \begin{bmatrix} 180/49 \\ -120/49 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

$$r_2 = 12 \begin{bmatrix} 1/7 \\ -2/7 \end{bmatrix} - \frac{7}{10} \begin{bmatrix} 2 & 2 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} 180/49 \\ -120/49 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Como $r_2 = b - Ax_2 = 0$, la solución es $x_2 = [4, -1]$. ◀

TEOREMA 2.16 Sea A una matriz simétrica definida positiva de $n \times n$ y sea b un vector distinto de cero. En el método del gradiente conjugado, suponga que $r_k \neq 0$ para $k < n$ (si $r_k = 0$, la ecuación está resuelta). Entonces, para cada $1 \leq k \leq n$,

(a) Los siguientes tres subespacios de R^n son iguales:

$$\langle x_1, \ldots, x_k \rangle = \langle r_0, \ldots, r_{k-1} \rangle = \langle d_0, \ldots, d_{k-1} \rangle,$$

(b) los residuos r_k son ortogonales por pares: $r_k^T r_j = 0$ para $j < k$.

(c) las direcciones d_k son conjugadas de A por pares: $d_k^T A d_j = 0$ para $j < k$. ■

Demostración. (a) Para $k = 1$, observe que $\langle x_1 \rangle = \langle d_0 \rangle = \langle r_0 \rangle$, puesto que $x_0 = 0$. Por definición $x_k = x_{k-1} + \alpha_{k-1} d_{k-1}$. Esto implica por inducción que $\langle x_1, \ldots, x_k \rangle = \langle d_0, \ldots, d_{k-1} \rangle$. Un argumento similar usando $d_k = r_k + \beta_{k-1} d_{k-1}$ muestra que $\langle r_0, \ldots, r_{k-1} \rangle$ es igual a $\langle d_0, \ldots, d_{k-1} \rangle$.

Para (b) y (c), se procede por inducción. Cuando $k = 0$ no hay nada que demostrar. Suponga que (b) y (c) se cumplen para k, y se demostrará (b) y (c) para $k + 1$. Multiplique la definición de r_{k+1} por r_j^T por la izquierda:

$$r_j^T r_{k+1} = r_j^T r_k - \frac{r_k^T r_k}{d_k^T A d_k} r_j^T A d_k. \tag{2.46}$$

Si $j \leq k - 1$, entonces $r_j^T r_k = 0$ por la hipótesis de inducción (b). Dado que r_j puede expresarse como una combinación de $d_0, \ldots, d_j$, el término $r_j^T A d_k = 0$ a partir de la hipótesis de inducción (c),

y se cumple (b). Por otro lado, si $j = k$, entonces $r_k^T r_{k+1} = 0$ de nuevo se deduce de (2.46), porque $d_k^T A d_k = r_k^T A d_k + \beta_{k-1} d_{k-1}^T A d_k = r_k^T A d_k$, usando la hipótesis de inducción (c). Lo anterior demuestra (b).

Ahora que $r_k^T r_{k+1} = 0$, (2.46) con $j = k + 1$ dice

$$\frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} = -\frac{r_{k+1}^T A d_k}{d_k^T A d_k}. \tag{2.47}$$

Esto, junto con la multiplicación por la izquierda de la definición de d_{k+1} por $d_j^T A$ resulta en

$$d_j^T A d_{k+1} = d_j^T A r_{k+1} - \frac{r_{k+1}^T A d_k}{d_k^T A d_k} d_j^T A d_k. \tag{2.48}$$

Si $j = k$, entonces $d_k^T A d_{k+1} = 0$ a partir de (2.48), utilizando la simetría de A. Si $j \leq k - 1$, entonces $Adj = (r_j - r_{j+1})/\alpha_j$ (a partir de la definición de r_{k+1}) es ortogonal a r_{k+1}, lo que demuestra que el primer término del lado derecho de (2.48) es cero, y el segundo término es cero por la hipótesis de inducción, lo que completa el argumento para (c). ❐

En el ejemplo 2.29, observe que r_1 es ortogonal a r_0, como lo garantiza el teorema 2.16. Este hecho es la clave del éxito para el método del gradiente conjugado: cada nuevo residuo r_i es ortogonal a todos los r_i anteriores. Si uno de los r_i resulta ser cero, entonces $Ax_i = b$ y x_i es la solución. Si no es así, después de n pasos a través del ciclo, r^n es ortogonal a un espacio abarcado por los n vectores ortogonales por pares $r_0, \ldots, r_{n-1}$, que debe ser en total de R^n. Así r_n debe ser el vector cero, y $Ax_n = b$.

En algunos aspectos, el método del gradiente conjugado es más simple que la eliminación gaussiana. Por ejemplo, la escritura de código parece más fácil de manejar (no hay operaciones de fila de las cuales preocuparse y no hay un triple ciclo como en la eliminación gaussiana). Ambos son métodos directos y ambos llegan a la solución teóricamente correcta en un número finito de pasos. Por lo tanto, quedan dos preguntas: ¿por qué no debería preferirse el gradiente conjugado sobre la eliminación gaussiana, y por qué el gradiente conjugado se trata con frecuencia como un método iterativo?

La respuesta a ambas preguntas inicia con un conteo de operaciones. El desplazamiento a través del ciclo requiere un producto matriz-vector Ad_{n-1} y varios productos punto adicionales. El producto matriz-vector solo requiere n^2 multiplicaciones en cada paso (junto con aproximadamente el mismo número de adiciones), para un total de n^3 multiplicaciones después de n pasos. En comparación con el conteo de $n^3/3$ operaciones para la eliminación gaussiana, el gradiente conjugado es tres veces más caro.

El panorama cambia si A es dispersa. Suponga que n es demasiado grande para que las $n^3/3$ operaciones de la eliminación gaussiana sean viables. A diferencia de la eliminación gaussiana debe ejecutarse hasta el final para dar una solución x, el gradiente conjugado da una aproximación x_i en cada paso.

El error hacia atrás, la longitud euclidiana del residuo, disminuye en cada paso, y así por lo menos en esa medida Ax_i cada vez está más cerca de b en cada paso. Por lo tanto, al monitorear r_i, puede encontrarse una solución x_i suficientemente buena para evitar la ejecución de todos los n pasos. En este contexto, el gradiente conjugado se vuelve indistinguible de un método iterativo.

El método cayó en desuso poco después de su descubrimiento, debido a su susceptibilidad a la acumulación de errores de redondeo cuando A es una matriz mal condicionada. De hecho, su desempeño en matrices mal condicionadas es inferior a la eliminación gaussiana con pivoteo parcial. En tiempos modernos, esta obstrucción se mejora con el **precondicionamiento**, que en esencia cambia el problema para tener un sistema matricial mejor condicionado, después de lo cual se aplica el gradiente conjugado. En la siguiente sección se analizará el método del gradiente conjugado precondicionado.

El título del método proviene de lo que el método del gradiente conjugado realmente hace: se desliza por las pendientes de un paraboloide cuadrático en n dimensiones. El "gradiente" que

forma parte del título significa que se trata de encontrar la dirección con la declinación más rápida usando el cálculo, y "conjugado" significa que no todos sus pasos individuales son ortogonales entre sí, pero al menos los residuos r_i sí lo son. Los detalles geométricos del método y su motivación son interesantes. El artículo original de Hestenes y Steifel [1952] da una descripción completa.

► EJEMPLO 2.30 Aplique el método del gradiente conjugado al sistema (2.45) con $n = 100\,000$.

Después de 20 pasos del método del gradiente conjugado, la diferencia entre la solución x calculada y la solución verdadera $(1, \ldots, 1)$ es inferior a 10^{-9} en la norma infinito del vector. El tiempo total de ejecución es de menos de un segundo en una PC. ◄

2.6.4 Precondicionamiento

La convergencia de los métodos iterativos como el método del gradiente conjugado puede acelerarse mediante el uso de una técnica llamada precondicionamiento. Con frecuencia, las razones de convergencia de los métodos iterativos dependen, directa o indirectamente, del número de condición de la matriz de coeficientes A. La idea del precondicionamiento es reducir el número de condición efectivo del problema.

La forma precondicionada del sistema lineal $Ax = b$ de $n \times n$ es

$$M^{-1}Ax = M^{-1}b,$$

donde M es una matriz invertible de $n \times n$ llamada el **precondicionador**. Lo único que se ha hecho es multiplicar por la izquierda la ecuación por la matriz. Un precondicionador eficaz reduce el número de condición del problema al tratar de invertir A. Conceptualmente, se trata de hacer dos cosas a la vez: la matriz M debe (1) estar lo más cerca posible de A y (2) ser fácil de invertir. Por lo general, estos dos objetivos se oponen entre sí.

La matriz más cercana a A es la misma A. Si se usa $M = A$, el número de condición del problema se eleva a 1, pero se supone que A no es fácil para invertir o no se estaría utilizando un método de solución sofisticado. La matriz más sencilla de invertir es la matriz identidad $M = \mathrm{I}$, pero esto no reduce el número de condición. El precondicionador perfecto sería una matriz en el punto medio de los dos extremos, que combina las mejores propiedades de ambos.

Una elección en particular sencilla es el **precondicionador de Jacobi** $M = D$, donde D es la diagonal de A. La inversa de D es la matriz diagonal de recíprocos de las entradas de D. Por ejemplo, en una matriz de diagonal estrictamente dominante el precondicionador de Jacobi tiene un gran parecido con A y además es simple de invertir. Observe que cada entrada diagonal de una matriz simétrica definida positiva es estrictamente positiva por la propiedad 3 de la sección 2.6.1, así que encontrar recíprocos no representa un problema.

Cuando A sea un matriz simétrica definida positiva de $n \times n$, se elegirá una matriz M simétrica definida positiva para usarla como un precondicionador. Recuerde que el producto interno de M, $(v, w)_M = v^T Mw$, como se define en la sección 2.6.3. El método del gradiente conjugado precondicionado ahora es fácil de describir: reemplace $Ax = b$ por la ecuación precondicionada $M^{-1}Ax = M^{-1}b$, y sustituya el producto escalar euclidiano con $(v, w)_M$. El razonamiento utilizado para el método del gradiente conjugado original aún se aplica porque la matriz $M^{-1}A$ permanece simétrica definida positiva en el nuevo producto interno.

Por ejemplo,

$$(M^{-1}Av, w)_M = v^T AM^{-1}Mw = v^T Aw = v^T MM^{-1}Aw = (v, M^{-1}Aw)_M.$$

Para convertir el algoritmo de la sección 2.6.3 en la versión precondicionada, sea $z_k = M^{-1}b - M^{-1}Ax_k = M^{-1}r_k$ el residuo del sistema precondicionado. Entonces

$$\begin{aligned}
\alpha_k &= \frac{(z_k, z_k)_M}{(d_k, M^{-1}Ad_k)_M} \\
x_{k+1} &= x_k + \alpha d_k \\
z_{k+1} &= z_k - \alpha M^{-1}Ad_k \\
\beta_k &= \frac{(z_{k+1}, z_{k+1})_M}{(z_k, z_k)_M} \\
d_{k+1} &= z_{k+1} + \beta_k d_k.
\end{aligned}$$

Las multiplicaciones por M puede reducirse si se observa que

$$\begin{aligned}
(z_k, z_k)_M &= z_k^T M z_k = z_k^T r_k \\
(d_k, M^{-1}Ad_k)_M &= d_k^T Ad_k \\
(z_{k+1}, z_{k+1})_M &= z_{k+1}^T M z_{k+1} = z_{k+1}^T r_{k+1}.
\end{aligned}$$

Con estas simplificaciones, el pseudocódigo para la versión precondicionada es el siguiente.

Método del gradiente conjugado precondicionado

x_0 = estimación inicial
$r_0 = b - Ax_0$
$d_0 = z_0 = M^{-1}r_0$
for $k = 0, 1, 2, \ldots, n-1$
  if $r_k = 0$, **stop**, **end**
  $\alpha_k = r_k^T z_k / d_k^T Ad_k$
  $x_{k+1} = x_k + \alpha_k d_k$
  $r_{k+1} = r_k - \alpha_k Ad_k$
  $z_{k+1} = M^{-1}r_{k+1}$
  $\beta_k = r_{k+1}^T z_{k+1} / r_k^T z_k$
  $d_{k+1} = z_{k+1} + \beta_k d_k$
end

La aproximación a la solución de $Ax = b$ después de k pasos es x_k. Observe que no es necesario realizar multiplicaciones explícitas por M^{-1}. Deben reemplazarse con sustituciones apropiadas hacia atrás debido a la simplicidad relativa de M.

El precondicionador de Jacobi es el más sencillo de una serie amplia y creciente de opciones posibles. Se describirá una familia adicional de ejemplos, y se guiará al lector hacia la literatura que describe alternativas más sofisticadas.

El **precondicionador de la sobre-relajación sucesiva simétrica (SRSS)** está definido por

$$M = (D + \omega L)D^{-1}(D + \omega U)$$

donde $A = L + D + U$ se divide en su parte triangular inferior, su diagonal y su parte triangular superior. Como en el método SRS, w es una constante entre 0 y 2. El caso especial de $w = 1$ se denomina el **precondicionador de Gauss-Seidel**.

Un precondicionador resulta de poca utilidad si es difícil de invertir. Observe que el precondicionador SRSS se define como el producto $M = (I + wLD^{-1})(D + wU)$ de una matriz triangular inferior y una matriz triangular superior, de modo que la ecuación $z = M^{-1}v$ puede resolverse mediante dos sustituciones hacia atrás:

$$\begin{aligned}
(I + \omega L D^{-1})c &= v \\
(D + \omega U)z &= c
\end{aligned}$$

Para una matriz dispersa, las dos sustituciones hacia atrás pueden realizarse en un tiempo proporcional al número de entradas distintas de cero. En otras palabras, la multiplicación por M^{-1} no es significativamente más compleja que la multiplicación por M.

► EJEMPLO 2.31 Sea A la matriz con entradas diagonales $A_{ii} = \sqrt{i}$ para $i = 1, \ldots, n$ y $A_{i,i+10} = A_{i+10,i} = \cos i$ para $i = 1, \ldots, n - 10$, con todos las demás entradas iguales a cero. Establezca como el vector de n unos, y defina $b = Ax$. Para $n = 500$, resuelva $Ax = b$ mediante el método del gradiente conjugado de tres maneras: sin utilizar un precondicionador, usando el precondicionador de Jacobi y empleando el precondicionador de Gauss-Seidel.

La matriz puede definirse en MATLAB mediante

```
A=diag(sqrt(1:n))+ diag(cos(1:(n-10)),10)
                 + diag(cos(1:(n-10)),-10).
```

En la figura 2.4 se muestran los tres distintos resultados. Incluso con esta matriz definida en forma sencilla, el método del gradiente conjugado es bastante lento en su convergencia si no se usa precondicionamiento. El precondicionador de Jacobi, que es bastante fácil de aplicar, permite una mejora significativa, mientras que el precondicionador de Gauss-Seidel sólo requiere aproximadamente 10 pasos para alcanzar la precisión de máquina. ◄

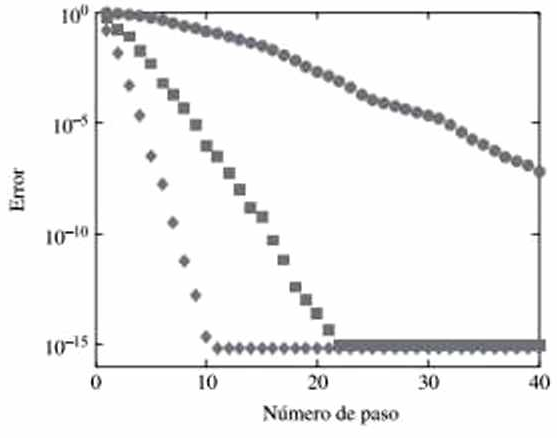


Figura 2.4 Eficiencia del método del gradiente conjugado precondicionado para la solución del ejemplo 2.31. El error se grafica de acuerdo con el número de pasos. Círculos: sin precondicionador. Cuadros: precondicionador de Jacobi. Rombos: precondicionador de Gauss-Seidel.

2.6 Ejercicios

1. Muestre que las siguientes matrices son simétricas definidas positivas, expresando $x^T Ax$ como una suma de cuadrados.

$$(a)\begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} (b)\begin{bmatrix} 1 & 3 \\ 3 & 10 \end{bmatrix} (c)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

2. Muestre que las siguientes matrices simétricas no son definidas positivas, mediante la búsqueda de un vector $x \neq 0$ de tal forma que $x^T Ax < 0$.

$$(a)\begin{bmatrix} 1 & 0 \\ 0 & -3 \end{bmatrix} (b)\begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix} (c)\begin{bmatrix} 1 & -1 \\ -1 & 0 \end{bmatrix} (d)\begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

3. Utilice el procedimiento de factorización de Cholesky para expresar las matrices del ejercicio 1 en la forma $A = R^T R$.

4. Muestre que el procedimiento de factorización de Cholesky falla para las matrices del ejercicio 2.

5. Encuentre la factorización de Cholesky $A = R^T R$ para cada matriz.

$$(a)\begin{bmatrix} 1 & 2 \\ 2 & 8 \end{bmatrix} (b)\begin{bmatrix} 4 & -2 \\ -2 & 5/4 \end{bmatrix} (c)\begin{bmatrix} 25 & 5 \\ 5 & 26 \end{bmatrix} (d)\begin{bmatrix} 1 & -2 \\ -2 & 5 \end{bmatrix}$$

6. Encuentre la factorización de Cholesky $A = R^T R$ para cada matriz.

$$(a)\begin{bmatrix} 4 & -2 & 0 \\ -2 & 2 & -3 \\ 0 & -3 & 10 \end{bmatrix} (b)\begin{bmatrix} 1 & 2 & 0 \\ 2 & 5 & 2 \\ 0 & 2 & 5 \end{bmatrix} (c)\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} (d)\begin{bmatrix} 1 & -1 & -1 \\ -1 & 2 & 1 \\ -1 & 1 & 2 \end{bmatrix}$$

7. Resuelva el sistema de ecuaciones encontrando la factorización de Cholesky para A, seguida por dos sustituciones hacia atrás.

$$(a)\begin{bmatrix} 1 & -1 \\ -1 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -7 \end{bmatrix} \quad (b)\begin{bmatrix} 4 & -2 \\ -2 & 10 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \end{bmatrix}$$

8. Resuelva el sistema de ecuaciones encontrando la factorización de Cholesky para A, seguida por dos sustituciones hacia atrás.

$$(a)\begin{bmatrix} 4 & 0 & -2 \\ 0 & 1 & 1 \\ -2 & 1 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix} \quad (b)\begin{bmatrix} 4 & -2 & 0 \\ -2 & 2 & -1 \\ 0 & -1 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ -7 \end{bmatrix}$$

9. Demuestre que si $d > 4$, la matriz $A = \begin{bmatrix} 1 & 2 \\ 2 & d \end{bmatrix}$ es definida positiva.

10. Encuentre todos los números d tales que $A = \begin{bmatrix} 1 & -2 \\ -2 & d \end{bmatrix}$ sea definida positiva.

11. Encuentre todos los números d tales que $A = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & d \end{bmatrix}$ sea definida positiva.

12. Demuestre que una submatriz principal de una matriz simétrica definida positiva es simétrica definida positiva. (Sugerencia: considere una X adecuada y utilice la propiedad 2).

13. Resuelva los problemas ejecutando a mano el método del gradiente conjugado.

$$(a)\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} (b)\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

14. Resuelva los problemas ejecutando a mano el método del gradiente conjugado.

$$\text{(a)}\begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{(b)}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -3 \\ 3 \end{bmatrix}$$

15. Ejecute la iteración del gradiente conjugado en el caso general escalar $Ax = b$, donde A es una matriz de 1×1. Encuentre α_1, x_1 y confirme que $r_1 = 0$ y $Ax_1 = b$.

2.6 Problemas de computadora

1. Escriba una versión para MATLAB del método del gradiente conjugado y utilícelo para resolver los sistemas

$$\text{(a)}\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \text{(b)}\begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

2. Utilice una versión en MATLAB del gradiente conjugado para resolver los siguientes problemas:

$$\text{(a)}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 3 \end{bmatrix} \text{(b)}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 5 \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \\ 4 \end{bmatrix}$$

3. Resuelva el sistema $Hx = b$ mediante el método del gradiente conjugado, donde H es la matriz de Hilbert de $n \times n$ y b es el vector de unos, para (a) $n = 4$ (b) $n = 8$.

4. Resuelva el problema disperso (2.45) mediante el método del gradiente conjugado para (a) $n = 6$ (b) $n = 12$.

5. Utilice el método del gradiente conjugado para resolver (2.45) con $n = 100$, 1000, y 10 000. Informe el tamaño del residuo final y el número de pasos necesarios.

6. Sea A la matriz de $n \times n$ con $n = 1000$ y las entradas

   $A(i,i) = i, A(i,i+1) = A(i+1,i) = 1/2, A(i,i+2) = A(i+2,i) = 1/2$ para todo i que se ajuste a la matriz. (a) Imprima la estructura diferente de cero `spy(A)`. (b) Sea x_e el vector de n unos. Establezca $b = Ax_e$ y aplique el método del gradiente conjugado sin precondicionador, con el precondicionador de Jacobi y con el precondicionador de Gauss-Seidel. Compare los errores de las tres corridas en una gráfica contra el número de pasos.

7. Sea $n = 1000$. Comience con la matriz A de $n \times n$ del problema de computadora 6 y añada los elementos distintos de cero $A(i, 2i) = A(2i, i) = 1/2$ para $1 \leq i \leq n/2$. Lleve a cabo los pasos (a) y (b) como en ese problema.

8. Sean $n = 500$ y A la matriz de $n \times n$ con entradas

   $A(i,i) = 2, A(i,i+2) = A(i+2,i) = 1/2, A(i,i+4) = A(i+4,i) = 1/2$ para toda i y $A(i, 500) = A(i, 500) = -0.1$ para $1 \leq i \leq 495$. Realice los pasos (a) y (b) como en el problema de computadora 6.

9. Sea A la matriz del problema de computadora 8, pero con los elementos de la diagonal cambiados por $A(i,i) = \sqrt[3]{i}$. Realice los incisos (a) y (b) como en ese problema.

10. Sea C el bloque de matrices de 195×195 con $C(i,i) = 2, C(i,i+3) = C(i+3,i) = 0.1$, $C(i,i+39) = C(i+39,i) = 1/2, C(i,i+42) = C(i+42,i) = 1/2$ para toda i. Defina A como la matriz de $n \times n$ con $n = 780$ formada por cuatro bloques C dispuestos en diagonal, y con bloques $\frac{1}{2}C$ en la super y sub-diagonal. Lleve a cabo los pasos (a) y (b) como en el problema de computadora 6 para resolver $Ax = b$.

2.7 SISTEMAS DE ECUACIONES NO LINEALES

El capítulo 1 contiene métodos para resolver una ecuación con una sola incógnita, por lo general no lineal. En este capítulo, se han estudiado los métodos para resolver sistemas de ecuaciones, pero se requiere que las ecuaciones sean lineales. La combinación de no lineal y "más de una ecuación" eleva el grado de dificultad de manera considerable. En esta sección se describe el método de Newton y sus variantes para la resolución de sistemas de ecuaciones no lineales.

2.7.1 Método de Newton multivariado

El método de Newton de una variable

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

proporciona el esquema principal para el método de Newton multivariado. Ambos se derivan de la aproximación lineal proporcionada por la expansión de Taylor. Por ejemplo, sean

$$\begin{aligned} f_1(u, v, w) &= 0 \\ f_2(u, v, w) &= 0 \\ f_3(u, v, w) &= 0 \end{aligned} \tag{2.49}$$

tres ecuaciones no lineales con tres incógnitas u, v, w. Defina el vector de valores de función $F(u, v, w) = (f_1, f_2, f_3)$ y denote el problema (2.49) mediante $F(x) = 0$, donde $x = (u, v, w)$.

El análogo de la derivada f' en el caso de una variable es la **matriz Jacobiana** definida por

$$DF(x) = \begin{bmatrix} \frac{\partial f_1}{\partial u} & \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial w} \\ \frac{\partial f_2}{\partial u} & \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial w} \\ \frac{\partial f_3}{\partial u} & \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial w} \end{bmatrix}.$$

La expansión de Taylor para funciones vectoriales de variable vectorial alrededor de x_0 es

$$F(x) = F(x_0) + DF(x_0) \cdot (x - x_0) + O(x - x_0)^2.$$

Por ejemplo, la expansión lineal de $F(u, v) = (e^{u+v}, \operatorname{sen} u)$ alrededor de $x_0 = (0, 0)$ es

$$\begin{aligned} F(x) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} e^0 & e^0 \\ \cos 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + O(x^2) \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} u + v \\ u \end{bmatrix} + O(x^2). \end{aligned}$$

El método de Newton se basa en una aproximación lineal, sin tomar en cuenta los términos $O(x^2)$. Como en el caso unidimensional, sea $x = r$ la raíz y sea x_0 la estimación actual. Entonces

$$0 = F(r) \approx F(x_0) + DF(x_0) \cdot (r - x_0),$$

o

$$-DF(x_0)^{-1} F(x_0) \approx r - x_0. \tag{2.50}$$

Por lo tanto, se puede obtener una mejor aproximación de la raíz al resolver (2.50) para r.

Método de Newton multivariado

$$\begin{aligned} x_0 &= \text{vector inicial} \\ x_{k+1} &= x_k - (DF(x_k))^{-1} F(x_k) \quad \textbf{para } k = 0, 1, 2, \ldots. \end{aligned}$$

Dado que el cálculo de inversas es computacionalmente pesado, se utiliza un truco para evitarlo. En cada paso, en vez de seguir la definición anterior, establezca $x_{k+1} = x_k - s$, donde s es la solución de $DF(x_k)s = F(x_k)$. Ahora, sólo se requiere la eliminación gaussiana ($n^3/3$ multiplicaciones) para realizar un paso, en vez de calcular una inversa (aproximadamente tres veces más multiplicaciones). Por lo tanto, el paso de la iteración para el método de Newton multivariado es

$$\begin{cases} DF(x_k)s = -F(x_k) \\ x_{k+1} = x_k + s. \end{cases} \tag{2.51}$$

► EJEMPLO 2.32 Utilice el método de Newton con una estimación inicial (1, 2) para encontrar una solución del sistema

$$\begin{aligned} v - u^3 &= 0 \\ u^2 + v^2 - 1 &= 0. \end{aligned}$$

En la figura 2.5 se muestran los conjuntos en los que $f_1(u, v) = v - u^3$ y $f_2(u, v) = u^2 + v^2 - 1$ son cero y los dos puntos de intersección, que son las soluciones del sistema de ecuaciones.

La matriz jacobiana es

$$DF(u, v) = \begin{bmatrix} -3u^2 & 1 \\ 2u & 2v \end{bmatrix}.$$

Si se usa el punto de inicio $x_0 = (1, 2)$, en el primer paso debe resolverse la ecuación matricial (2.51):

$$\begin{bmatrix} -3 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -\begin{bmatrix} 1 \\ 4 \end{bmatrix}.$$

La solución es $s = (0, -1)$, por lo que la primera iteración produce $x_1 = x_0 + s = (1, 1)$. El segundo paso requiere resolver

$$\begin{bmatrix} -3 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = -\begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

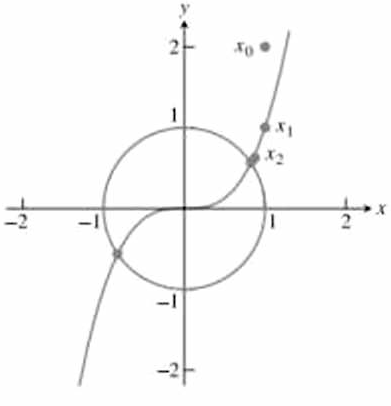


Figura 2.5 Método de Newton para el ejemplo 2.32. Las dos raíces son los puntos sobre el círculo. El método de Newton produce los puntos que convergen a la solución en aproximadamente (0.8260, 0.5636).

La solución es $s = (-1/8, -3/8)$ y $x_2 = x_1 + s = (7/8, 5/8)$. Las dos iteraciones se muestran en la figura 2.5. Los pasos subsiguientes producen la siguiente tabla:

paso	u	v
0	1.00000000000000	2.00000000000000
1	1.00000000000000	1.00000000000000
2	0.87500000000000	0.62500000000000
3	0.82903634826712	0.56434911242604
4	0.82604010817065	0.56361977350284
5	0.82603135773241	0.56362416213163
6	0.82603135765419	0.56362416216126
7	0.82603135765419	0.56362416216126

La duplicación de posiciones decimales correctas característica de la convergencia cuadrática es evidente en la secuencia de salida. La simetría de las ecuaciones muestra que si (u, v) es una solución, entonces también lo es $(-u, -v)$, como es evidente en la figura 2.5. La segunda solución también puede encontrarse al aplicar el método de Newton con una estimación inicial cercana. ◄

► EJEMPLO 2.33 Use el método de Newton para encontrar las soluciones del sistema

$$f_1(u, v) = 6u^3 + uv - 3v^3 - 4 = 0$$
$$f_2(u, v) = u^2 - 18uv^2 + 16v^3 + 1 = 0.$$

Observe que $(u, v) = (1, 1)$ es una solución. Resulta que hay otras dos. La matriz jacobiana es

$$DF(u, v) = \begin{bmatrix} 18u^2 + v & u - 9v^2 \\ 2u - 18v^2 & -36uv + 48v^2 \end{bmatrix}.$$

La solución encontrada por el método de Newton depende de la estimación inicial, al igual que en el caso unidimensional. Si se usa el punto inicial $(u_0, v_0) = (2, 2)$ y se itera la fórmula anterior, se obtiene la tabla siguiente:

paso	u	v
0	2.00000000000000	2.00000000000000
1	1.37258064516129	1.34032258064516
2	1.07838681200443	1.05380123264984
3	1.00534968896520	1.00269261871539
4	1.00003367866506	1.00002243772010
5	1.00000000111957	1.00000000057894
6	1.00000000000000	1.00000000000000
7	1.00000000000000	1.00000000000000

Con otros vectores iniciales se llega a las otras dos raíces, que son aproximadamente (0.865939, 0.462168) y (0.886809, −0.294007). Vea el problema de computadora 2. ◄

El método de Newton es una buena opción si es posible calcular la jacobiana. Si no es así, la mejor alternativa es el método de Broyden, el tema de la siguiente sección.

2.7.2 Método de Broyden

El método de Newton para resolver una ecuación con una sola incógnita requiere saber cuál es la derivada. El desarrollo de este método en el capítulo 1 estuvo seguido por el análisis del método de la secante, que se utiliza cuando la derivada no existe o es demasiado complicada de evaluar.

Ahora que ya se tiene una versión del método de Newton para los sistemas de ecuaciones no lineales $F(x) = 0$, surge la misma pregunta: ¿qué pasa si la matriz jacobiana DF no está disponi-

ble? Aunque no existe una extensión simple del método de Newton a un método de la secante para sistemas, Broyden [1965] propuso un método que suele considerarse la mejor opción.

Suponga que A_i es la mejor aproximación disponible en el paso i hacia la matriz jacobiana y que se ha utilizado para crear

$$x_{i+1} = x_i - A_i^{-1} F(x_i). \tag{2.52}$$

Para actualizar A_i a A_{i+1} para el siguiente paso, sería ideal respetar el carácter derivado de la jacobiana DF y satisfacer

$$A_{i+1}\delta_{i+1} = \Delta_{i+1}, \tag{2.53}$$

donde $\delta_{i+1} = x_{i+1}$ y $\Delta_{i+1} = F(x_{i+1}) - F(x_i)$. Por otro lado, para el complemento ortogonal de δ_{i+1}, no se tiene información nueva. Por lo tanto, se pide que

$$A_{i+1}w = A_i w \tag{2.54}$$

para cada w que satisface $\delta_{i+1}^T w = 0$. Se puede demostrar que una matriz que satisface tanto (2.53) como (2.54) es

$$A_{i+1} = A_i + \frac{(\Delta_{i+1} - A_i\delta_i)\delta_{i+1}^T}{\delta_{i+1}^T\delta_{i+1}}. \tag{2.55}$$

El método Broyden utiliza el paso (2.52) del método de Newton para avanzar desde la estimación actual, mientras actualiza la jacobiana aproximada mediante (2.55). En resumen, el algoritmo comienza con un valor inicial x_0 y una jacobiana aproximada inicial A_0, para la cual puede elegirse la matriz identidad si no hay otra opción mejor.

Método de Broyden I

x_0 = vector inicial
A_0 = matriz inicial
for $i = 0, 1, 2, \ldots$
$$x_{i+1} = x_i - A_i^{-1} F(x_i)$$
$$A_{i+1} = A_i + \frac{(\Delta_{i+1} - A_i\delta_{i+1})\delta_{i+1}^T}{\delta_{i+1}^T\delta_{i+1}}$$
end

donde $\delta_{i+1} = x_{i+1} - x_i$ y $\Delta_{i+1} = F(x_{i+1}) - F(x_i)$.

Observe que el paso tipo Newton se lleva a cabo al resolver $A_i\delta_{i+1} = F(x_i)$, como en el método de Newton. Al igual que el método de Newton, el método de Broyden no garantiza su convergencia a una solución.

Un segundo enfoque para el método de Broyden evita el paso relativamente complicado de resolver la matriz $A_i\delta_{i+1} = F(x_i)$. Como el mejor resultado se obtiene sólo al aproximar la derivada DF durante la iteración, también es posible lograrlo aproximando la inversa de DF, que es lo requerido por el paso de Newton.

Se realiza de nuevo la deducción de Broyden desde el punto de vista de $B_i = A_i^{-1}$. Sería deseable tener

$$\delta_{i+1} = B_{i+1}\Delta_{i+1}, \tag{2.56}$$

donde $\delta_{i+1} = x_{i+1} - x_i$ y $\Delta_{i+1} = F(x_{i+1}) - F(x_i)$, y para cada w que satisface $\delta_{i+1}^T w = 0$, aún satisface $A_{i+1}w = A_i w$, o

$$B_{i+1}A_i w = w. \tag{2.57}$$

Una matriz que satisface tanto (2.56) y (2.57) es

$$B_{i+1} = B_i + \frac{(\delta_{i+1} - B_i \Delta_{i+1})\delta_{i+1}^T B_i}{\delta_{i+1}^T B_i \Delta_{i+1}}. \tag{2.58}$$

La nueva versión de la iteración, que no necesita la resolución de ninguna matriz, es

$$x_{i+1} = x_i - B_i F(x_i). \tag{2.59}$$

El algoritmo resultante se denomina método de Broyden II.

Método de Broyden II

x_0 = vector inicial
B_0 = matriz inicial
for $i = 0, 1, 2, \ldots$
$x_{i+1} = x_i - B_i F(x_i)$
$B_{i+1} = B_i + \dfrac{(\delta_{i+1} - B_i \Delta_{i+1})\delta_{i+1}^T B_i}{\delta_{i+1}^T B_i \Delta_{i+1}}$
end

donde $\delta_i = x_i - x_{i-1}$ y $\Delta i = F(x) - F(x_{i-1})$.

Para empezar, se necesitan un vector inicial x_0 y una estimación inicial para B_0. Si es imposible calcular derivadas, puede usarse la elección $B_0 = I$.

Una desventaja percibida en Broyden II es que las estimaciones de la jacobiana, necesarias para algunas aplicaciones, no se encuentran con facilidad. La matriz B_i es una estimación de la matriz inversa de la jacobiana. Por otra parte, Broyden I sigue la pista de A_i, que estima la jacobiana. Por esta razón, en algunos círculos Broyden I y II se conocen como "Broyden bueno" y "Broyden malo", respectivamente.

Ambas versiones del método Broyden convergen superlinealmente (a las raíces simples), en forma un poco más lenta que la convergencia cuadrática del método de Newton. Si se dispone de una fórmula para la jacobiana, la convergencia suele acelerarse al utilizar la inversa de $DF(x_0)$ como la matriz inicial B_0.

El código de MATLAB para el método de Broyden II es como sigue:

```
% Programa 2.3  Método de Broyden II
% Entrada: vector inicial x0, pasos max k
% Salida: solución x
% Uso como ejemplo: broyden2(f, [1; 1], 10)
function x=broyden2(f,x0,k)
[n,m]=size(x0);
b=eye(n,n);              % b inicial
for i=1:k
  x=x0-b*f(x0);
  del=x-x0;delta=f(x)-f(x0);
  b=b+(del-b*delta)*del'*b/(del'*b*delta);
  x0=x;
end
```

Como muestra, una solución para el sistema del ejemplo 2.32 se encuentra al definir una función

```
>> f=@(x)  [x(2)-x(1)^3;x(1)^2+x(2)^2-1];
```

y al llamar el método de Broyden II

```
>> x=broyden2(f,[1;1],10)
```

El método de Broyden, en cualquiera de sus implementaciones, es muy útil en los casos donde la jacobiana no está disponible. Un ejemplo típico de esta situación se muestra en el modelo de pandeo de tuberías de la comprobación en la realidad 7.

2.7 Ejercicios

1. Encuentre la jacobiana de las funciones (a) $F(u, v) = (u^3, uv^3)$.
   (b) $F(u, v) = (\text{sen}\, uv, e^{uv})$ (c) $F(u, v) = (u^2 + v^2 - 1, (u - 1)^2 + v^2 - 1)$
   (d) $F(u, v, w) = (u^2 + v - w^2, \text{sen}\, uvw, uvw^4)$.

2. Use el desarrollo de Taylor para encontrar la aproximación lineal $L(x)$ a $F(x)$ cerca de x_0.
   (a) $F(u, v) = (1 + e^{u+2v}, \text{sen}(u + v)), x_0 = (0, 0)$
   (b) $F(u, v) = (u + e^{u-v}, 2u + v), x_0 = (1, 1)$

3. Trace las dos curvas en el plano uv y encuentre todas las soluciones exactas empleando álgebra simple.

   (a) $\begin{cases} u^2 + v^2 = 1 \\ (u - 1)^2 + v^2 = 1 \end{cases}$ (b) $\begin{cases} u^2 + 4v^2 = 4 \\ 4u^2 + v^2 = 4 \end{cases}$ (c) $\begin{cases} u^2 - 4v^2 = 4 \\ (u - 1)^2 + v^2 = 4 \end{cases}$

4. Aplique dos pasos del método de Newton a los sistemas del ejercicio 3, con el punto inicial (1, 1).

5. Aplique dos pasos de Broyden I a los sistemas del ejercicio 3, con el punto inicial (1, 1) y usando $A_0 = \mathbf{I}$.

6. Aplique dos pasos de Broyden II a los sistemas del ejercicio 3, con el punto inicial (1, 1) y usando $B_0 = \mathbf{I}$.

7. Demuestre que (2.55) satisface (2.53) y (2.54).

8. Demuestre que (2.58) satisface (2.56) y (2.57).

2.7 Problemas de computadora

1. Implemente el método de Newton con los puntos iniciales adecuados para encontrar todas las soluciones. Consulte el ejercicio 3 para asegurarse de que sus respuestas sean correctas.

   (a) $\begin{cases} u^2 + v^2 = 1 \\ (u - 1)^2 + v^2 = 1 \end{cases}$ (b) $\begin{cases} u^2 + 4v^2 = 4 \\ 4u^2 + v^2 = 4 \end{cases}$ (c) $\begin{cases} u^2 - 4v^2 = 4 \\ (u - 1)^2 + v^2 = 4 \end{cases}$

2. Utilice el método de Newton para encontrar las tres soluciones del ejemplo 2.31.

3. Utilice el método de Newton para encontrar las dos soluciones del sistema $u^3 - v^3 + u = 0$ y $u^2 + v^2 = 1$.

4. (a) Aplique el método de Newton para encontrar las dos soluciones del sistema de tres ecuaciones.

$$\begin{aligned} 2u^2 - 4u + v^2 + 3w^2 + 6w + 2 &= 0 \\ u^2 + v^2 - 2v + 2w^2 - 5 &= 0 \\ 3u^2 - 12u + v^2 + 3w^2 + 8 &= 0 \end{aligned}$$

5. Utilice el método de Newton multivariado para encontrar los dos puntos en común de las tres esferas dadas en el espacio tridimensional. (a) Cada esfera tiene radio 1, con los centros (1, 1, 0), (1, 0, 1) y (0, 1, 1). [Resp. (1, 1, 1) y (1/3, 1/3, 1/3)] (b) Cada esfera tiene un radio de 5, con los centros (1, −2, 0), (−2, 2, −1) y (4, −2, 3).

6. Aunque una intersección genérica de tres esferas en el espacio tridimensional tiene dos puntos, puede tener un solo punto. Aplique el método de Newton multivariado para encontrar el punto de intersección de las esferas con centro (1, 0, 1) y radio $\sqrt{8}$, centro (0, 2, 2) y radio $\sqrt{2}$, y centro (0, 3, 3) y radio $\sqrt{2}$. ¿La iteración aún converge cuadráticamente? Explique.

7. Aplique Broyden I con estimaciones iniciales $x_0 = (1, 1)$ y $A_0 = \mathbf{I}$ a los sistemas del ejercicio 3. Informe las soluciones con la mayor precisión posible y el número de pasos necesarios.

8. Aplique Broyden II con estimaciones iniciales $x_0 = (1, 1)$ y $B_0 = \mathbf{I}$ a los sistemas del ejercicio 3. Informe las soluciones con la mayor precisión posible y el número de pasos necesarios.

9. Aplique Broyden I para encontrar los conjuntos de dos puntos de intersección en el problema de computadora 5.

10. Aplique Broyden I para encontrar el punto de intersección en el problema de computadora 6. ¿Qué puede observarse acerca de la velocidad de convergencia?

11. Aplique Broyden II para encontrar los conjuntos de dos puntos de intersección en el problema de computadora 5.

12. Aplique Broyden II para encontrar el punto de intersección en el problema de computadora 6. ¿Qué puede observarse acerca de la velocidad de convergencia?

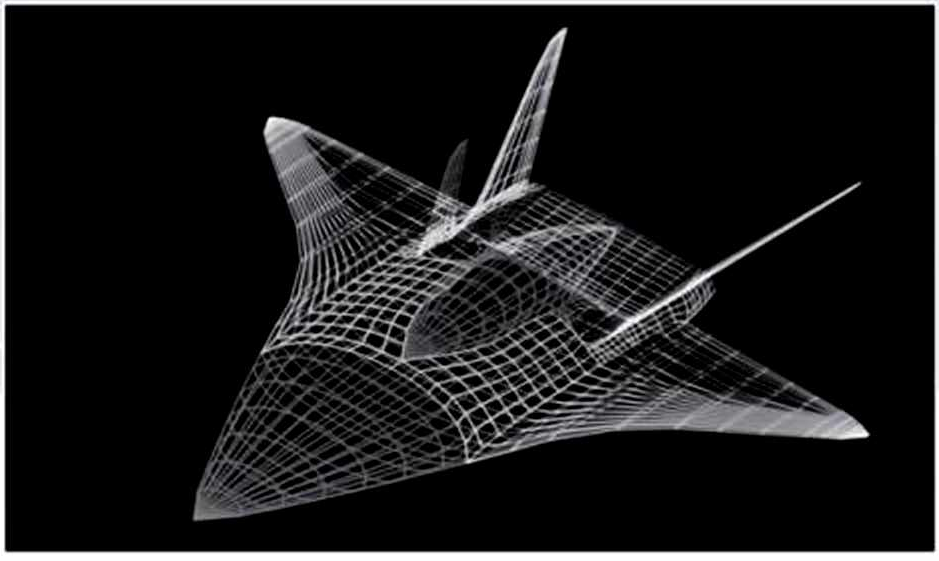
Software y lecturas adicionales

Han aparecido muchos libros excelentes sobre álgebra lineal numérica, incluyendo Stewart [1973] y la referencia completa de Golub y Van Loan [1996]. Dos libros excelentes con un enfoque moderno del álgebra lineal numérica son Demmel [1997] y Trefethen y Bau [1997]. Algunos libros en los que puede consultar los métodos iterativos son Axelsson [1994], Hackbush [1994], Kelley [1995], Saad [1996], Traub [1964], Varga [2000], Young [1971], y Dennis y Schnabel [1983].

LAPACK es un paquete de software completo, de dominio público que contiene rutinas de alta calidad para realizar cálculos con álgebra matricial, incluyendo los métodos para la solución de $Ax = b$, las factorizaciones matriciales y la estimación del número de condición. Está escrito con sumo cuidado de modo que pueda trasladarse a las arquitecturas de computación modernas, como el vector de memoria compartida y los procesadores en paralelo. Vea Anderson et al. [1990].

La portabilidad de LAPACK reside en el hecho de que sus algoritmos están escritos de tal manera que incrementan al máximo el uso de subprogramas de álgebra lineal básica (BLAS), un conjunto de cálculos primitivos de matrices y vectores que pueden ajustarse para optimizar el rendimiento en máquinas y arquitecturas particulares. BLAS se divide en tres partes: Nivel 1, que requiere $O(n)$ operaciones como productos punto; Nivel 2, operaciones como la multiplicación de matrices y vectores, que son $O(n^2)$; y Nivel 3, incluyendo la multiplicación completa matriz y matriz que tiene una complejidad $O(n^3)$.

La rutina general de matriz densa en LAPACK para resolver $Ax = b$ con precisión doble, usando la factorización $PA = LU$, se llama DGESV, y hay otras versiones para matrices dispersas y de bandas. Para conocer más detalles consulte el sitio `www.netlib.org/lapack`. Las implementaciones de las rutinas de LAPACK también forman la base para los cálculos con álgebra matricial en MATLAB y en los paquetes IMSL y NAG.



CAPÍTULO

3

Interpolación

La interpolación polinomial es una práctica antigua, pero su uso industrial pesado comenzó con la aplicación de las splines (curvas de trazo suave sobre un conjunto de puntos) cúbicas en el siglo XX. Sobre todo en las industrias constructoras de barcos y aviones, los ingenieros Paul de Casteljau y Pierre Bézier que trabajaban para los fabricantes de automóviles europeos rivales Citroen y Renault, respectivamente, seguidos por otros como General Motors en los Estados Unidos, estimularon el desarrollo de lo que ahora se conocen como splines cúbicas y curvas de Bézier.

Aunque se desarrollaron para estudios aerodinámicos de automóviles, estas curvas se han utilizado en muchas aplicaciones, incluyendo la composición tipográfica por computadora. Dos ingenieros de Xerox causaron una revolución en la impresión, al formar una empresa llamada Adobe y lanzar el lenguaje PostScript™ en 1984. Llamaron la atención de Steve Jobs de Apple Corporation, quien estaba buscando una manera de controlar la impresora láser recién inventada. Las curvas de Bézier representaron una manera sencilla de adaptar las mismas curvas matemáticas a fuentes con múltiples resoluciones de impresora. Más tarde, Adobe utilizó muchas de las ideas fundamentales de PostScript como la base de un formato más flexible llamado PDF (Portable Document Format), que se convirtió en un tipo de documento presente en el siglo XXI.

Comprobación en la realidad En la página 183 se muestra un ejemplo de archivos PDF que utilizan curvas de Bézier para representar caracteres impresos en fuentes arbitrarias.

Una manera eficiente de representar y entender los datos en los problemas científicos es la llamada aproximación polinomial. Suponga que se toman los puntos (x, y) de una función dada $y = f(x)$, o quizá de un experimento donde x indica la temperatura y y la velocidad de reacción. Cualesquiera de estas dos formas de información representa una cantidad infinita de información. Encontrar un polinomio que pasa a través de la serie de datos significa sustituir la información con una regla que puede evaluarse en un número finito de pasos. Aunque es poco realista esperar que el polinomio represente con exactitud la función para cualquier entrada nueva x, puede ser suficientemente cercano para resolver problemas prácticos.

En este capítulo se presenta la interpolación polinomial y la interpolación de curvas como herramientas convenientes para la búsqueda de funciones que pasan a través de ciertos puntos dados como información.

3.1 DATOS Y FUNCIONES DE INTERPOLACIÓN

Se dice que una función interpola a un conjunto de puntos si pasa a través de esos puntos. Suponga que se tiene un conjunto de puntos (x, y), como $(0, 1)$, $(2, 2)$ y $(3, 4)$. Existe una parábola que pasa por los tres puntos, como se muestra en la figura 3.1. Esta parábola se denomina polinomio de interpolación de segundo grado que pasa por los tres puntos.

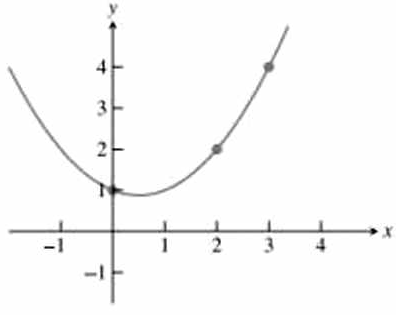


Figura 3.1 Interpolación mediante una parábola. Los puntos (0, 1), (2, 2) y (3, 4) son interpolados por la función $P(x) = \frac{1}{2}x^2 - \frac{1}{2}x + 1$.

DEFINICIÓN 3.1 La función $y = P(x)$ **interpola** los puntos $(x_1, y_1), \ldots, (x_n, y_n)$ si $P(x_i) = y_i$ para cada $1 \leq i \leq n$. ❐

Tenga en cuenta que es necesario que P sea una función, es decir, cada valor x corresponde a una y única. Esto pone una restricción sobre el conjunto de puntos de datos $\{(x_i, y_i)\}$ que pueden interpolarse: todas las x_i deben ser distintas para que una función pase a través de ellas. No hay ninguna restricción de este tipo para las y_i.

Para empezar, se buscará un polinomio de interpolación. ¿Existe siempre dicho polinomio? Asumiendo que las coordenadas x de los puntos son distintas, la respuesta es sí. No importa cuántos puntos se den, hay algún polinomio $y = P(x)$ que pasa por todos los puntos. En esta sección se demuestran éste y otros hechos sobre los polinomios de interpolación.

La interpolación es el inverso de la evaluación. En la evaluación de polinomios (como la multiplicación anidada del capítulo 0), se proporciona un polinomio y se pide evaluar un valor de y para un valor dado de x, es decir, calcular los puntos situados sobre la curva. La interpolación polinómica pide el proceso contrario: dados estos puntos, debe calcularse un polinomio que pueda generarlos.

ANOTACIÓN

Resumen ¿Por qué se utilizan polinomios? Los polinomios se utilizan con mucha frecuencia para la interpolación debido a sus propiedades matemáticas sencillas. Una teoría simple dice que siempre existirá un polinomio de interpolación de cierto grado para un conjunto dado de puntos. Más importante aún, en un sentido real, los polinomios son fácilmente programables en las computadoras digitales. Las unidades centrales de procesamiento suelen tener métodos rápidos en hardware para sumar y multiplicar los números (representados en punto flotante), que son las únicas operaciones necesarias para evaluar un polinomio. Las funciones complicadas puede aproximarse mediante la interpolación de polinomios, siendo más fácil su cálculo con estas dos operaciones de hardware.

3.1.1 Interpolación de Lagrange

Supongamos que se tienen n puntos de datos $(x_1, y_1), \ldots, (x_n, y_n)$, y que se desea encontrar un polinomio de interpolación. Existe una fórmula explícita, llamada fórmula de interpolación de Lagrange, para escribir un polinomio de grado $d = n - 1$ que interpola los puntos. Por ejemplo, suponga que se proporcionan tres puntos (x_1, y_1), (x_2, y_2), (x_3, y_3). Entonces, el polinomio

$$P_2(x) = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} \tag{3.1}$$

es el **polinomio de interpolación de Lagrange** que pasa por estos puntos. En primer lugar, observe por qué cada uno de lo puntos pertenece al polinomio. Cuando x_1 sustituye a x, los términos se evalúan como $y_1 + 0 + 0 = y_1$. Los términos segundo y tercero desaparecen cuando se sustituye x_1, y el primer término representa sólo y_1. Algo similar ocurre cuando se sustituyen x_2 y x_3. Al sustituir cualquier otro número por x, se tiene poco control sobre el resultado. Pero entonces, el trabajo consiste sólo en interpolar en los tres puntos (hasta ahí es donde interesa). En segundo lugar, observe que el polinomio (3.1) es de segundo grado con una sola variable x.

► EJEMPLO 3.1 Encuentre el polinomio de interpolación para los puntos (0, 1), (2, 2) y (3, 4) que se muestran en la figura 3.1.

Al sustituir en la fórmula de Lagrange (3.1) se obtiene

$$\begin{aligned} P_2(x) &= 1\frac{(x-2)(x-3)}{(0-2)(0-3)} + 2\frac{(x-0)(x-3)}{(2-0)(2-3)} + 4\frac{(x-0)(x-2)}{(3-0)(3-2)} \\ &= \frac{1}{6}(x^2 - 5x + 6) + 2\left(-\frac{1}{2}\right)(x^2 - 3x) + 4\left(\frac{1}{3}\right)(x^2 - 2x) \\ &= \frac{1}{2}x^2 - \frac{1}{2}x + 1. \end{aligned}$$

Se cumple que $P_2(0) = 1$, $P_2(2) = 2$ y $P_2(3) = 4$. ◄

En general, suponga que se tienen n puntos $(x_1, y_1), \ldots, (x_n, y_n)$. Para cada k entre 1 y n, se define el polinomio de grado $n - 1$

$$L_k(x) = \frac{(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}.$$

La propiedad interesante de L_k es que $L_k(x_k) = 1$, mientras que $L_k(x_j) = 0$, donde x_j es cualquiera de los otros puntos. Entonces se define el polinomio de grado $n - 1$

$$P_{n-1}(x) = y_1 L_1(x) + \cdots + y_n L_n(x).$$

Ésta es una generalización directa del polinomio en (3.1) y funciona de la misma manera. Al sustituir x_k en lugar de x se obtiene

$$P_{n-1}(x_k) = y_1 L_1(x_k) + \cdots + y_n L_n(x_k) = 0 + \cdots + 0 + y_k L_k(x_k) + 0 + \cdots + 0 = y_k,$$

por lo que funciona como fue diseñado.

Se ha construido un polinomio cuyo grado máximo es $n - 1$ y que pasa a través de cualquier conjunto de n puntos con x_i distintas. Curiosamente, es el único.

TEOREMA 3.2 **Teorema principal de la interpolación polinomial.** Sean $(x_1, y_1), \ldots, (x_n, y_n)$ n parejas de puntos con distintas x_i. Entonces existe uno y sólo un polinomio P de grado $n-1$ o menor que satisface $P(x_i) = y_i$ para $i = 1, \ldots, n$. ■

Demostración. La existencia se demuestra mediante la fórmula explícita para la interpolación de Lagrange. Para demostrar que sólo hay un polinomio de estas características, suponga, por el bien del argumento, que hay dos polinomios $P(x)$ y $Q(x)$ que son de grado $n-1$ y que ambos interpolan todos los n puntos. Es decir, se supone que $P(x_1) = Q(x_1) = y_1$, $P(x_2) = Q(x_2) = y_2, \ldots, P(x_n) = Q(x_n) = y_n$. Ahora se define el nuevo polinomio $H(x) = P(x) - Q(x)$. Claramente, el grado de H también es $n-1$ y se observa que $0 = H(x_1) = H(x_2) = \ldots = H(x_n)$; es decir, H tiene n ceros distintos. De acuerdo con el teorema fundamental del álgebra, un polinomio de grado d tiene d ceros, a menos que sea el polinomio idénticamente igual a cero. Por lo tanto, H es el polinomio idénticamente igual a cero y $P(x) = Q(x)$. Se llega a la conclusión de que hay un único $P(x)$ de grado $\leq n-1$ que interpola los n puntos (x_i, y_i). ❐

► EJEMPLO 3.2 Encuentre el polinomio de tercer grado o menor que interpole los puntos (0, 2), (1, 1), (2, 0) y (3, −1).

La forma de Lagrange es la siguiente:

$$\begin{aligned} P(x) &= 2\frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} + 1\frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} \\ &\quad + 0\frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} - 1\frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)} \\ &= -\frac{1}{3}(x^3 - 6x^2 + 11x - 6) + \frac{1}{2}(x^3 - 5x^2 + 6x) - \frac{1}{6}(x^3 - 3x^2 + 2x) \\ &= -x + 2. \end{aligned}$$

El teorema 3.2 dice que existe exactamente un polinomio de interpolación de tercer grado o menor, pero puede ser o no exactamente de tercer grado. En el ejemplo 3.2, los puntos de datos están alineados sobre una línea recta, por lo que el polinomio de interpolación es de primer grado. El teorema 3.2 implica que no hay polinomios de interpolación de segundo o tercer grado. Quizá para usted ya sea evidente que ninguna parábola o curva cúbica puede pasar a través de cuatro puntos alineados, ésta es la razón. ◄

3.1.2 Diferencias divididas de Newton

El método de interpolación de Lagrange, como se describió en la sección anterior, es una manera constructiva de escribir el polinomio único prometido por el teorema 3.2. También es intuitivo: un vistazo rápido explica por qué funciona. Sin embargo, rara vez se utiliza en cálculos porque existen métodos alternativos que tienen formas más manejables y cálculos menos complejos.

Las diferencias divididas de Newton proporcionan una manera muy sencilla de escribir el polinomio de interpolación. Dados n puntos, el resultado será un polinomio con un grado máximo de $n-1$, igual que en el caso de Lagrange. El teorema 3.2 dice que no puede ser otro polinomio que el mismo polinomio de interpolación de Lagrange, escrito en una forma diferente.

La idea de las diferencias divididas es bastante simple, pero requiere el dominio de cierta notación antes de enunciarla. Suponga que las parejas de puntos provienen de una función $f(x)$, por lo que el objetivo es interpolar $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$.

DEFINICIÓN 3.3 Indique mediante $f[x_1 \ldots x_n]$ el coeficiente del término x^{n-1} en el polinomio (único) que interpola $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$. ❐

En el ejemplo 3.1 se muestra que $f[0\ 2\ 3] = 1/2$, donde sabemos que $f(0) = 1$, $f(2) = 2$ y $f(3) = 4$. Por supuesto, por la singularidad, todas las permutaciones de 0, 2, 3 dan el mismo valor: $1/2 = f[0\ 3\ 2] = f[3\ 0\ 2]$, etcétera. Usando esta definición, se obtiene la siguiente fórmula alternativa bastante notable para el polinomio de interpolación, llamada fórmula de las diferencias divididas de Newton.

$$\begin{aligned} P(x) = f[x_1] &+ f[x_1\ x_2](x - x_1) \\ &+ f[x_1\ x_2\ x_3](x - x_1)(x - x_2) \\ &+ f[x_1\ x_2\ x_3\ x_4](x - x_1)(x - x_2)(x - x_3) \\ &+ \cdots \\ &+ f[x_1 \cdots x_n](x - x_1)\cdots(x - x_{n-1}). \end{aligned} \tag{3.2}$$

Además, los coeficientes $f[x_1 \ldots x_k]$ de la definición anterior pueden calcularse recursivamente como se describe a continuación. Construya con los puntos una tabla:

$$\begin{array}{c|c} x_1 & f(x_1) \\ x_2 & f(x_2) \\ \vdots & \vdots \\ x_n & f(x_n). \end{array}$$

Ahora se definen las diferencias divididas, que son:

$$\begin{aligned} f[x_k] &= f(x_k) \\ f[x_k\ x_{k+1}] &= \frac{f[x_{k+1}] - f[x_k]}{x_{k+1} - x_k} \\ f[x_k\ x_{k+1}\ x_{k+2}] &= \frac{f[x_{k+1}\ x_{k+2}] - f[x_k\ x_{k+1}]}{x_{k+2} - x_k} \\ f[x_k\ x_{k+1}\ x_{k+2}\ x_{k+3}] &= \frac{f[x_{k+1}\ x_{k+2}\ x_{k+3}] - f[x_k\ x_{k+1}\ x_{k+2}]}{x_{k+3} - x_k}, \end{aligned} \tag{3.3}$$

y así sucesivamente. Dos hechos importantes se presentan, (1) la interpolación polinomial única $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$ está dada por (3.2) y (2) los coeficientes pueden calcularse con (3.3), no son evidentes de inmediato y las demostraciones se darán en la sección 3.2.2. Observe que la fórmula de la diferencia dividida da el polinomio de interpolación como un polinomio anidado, listo para evaluarse de una manera eficiente.

Diferencias divididas de Newton

Dados $x = [x_1, \ldots, x_n]$, $y = [y_1, \ldots, y_n]$

for $j = 1, \ldots, n$
  $f[x_j] = y_j$
end

for $i = 2, \ldots, n$
  for $j = 1, \ldots, n + 1 - i$
    $f[x_j \ldots x_{j+i-1}] = (f[x_{j+1} \ldots x_{j+i-1}] - f[x_j \ldots x_{j+i-2}])/(x_{j+i-1} - x_j)$
  end
end

El polinomio de interpolación es

$$P(x) = \sum_{i=1}^{n} f[x_1 \ldots x_i](x - x_1)\cdots(x - x_{i-1})$$

Las diferencias divididas de Newton se presentan en forma de tabla. Para los tres puntos, la tabla tiene la forma

$$\begin{array}{c|c|c|c} x_1 & f[x_1] & & \\ & & f[x_1\ x_2] & \\ x_2 & f[x_2] & & f[x_1\ x_2\ x_3] \\ & & f[x_2\ x_3] & \\ x_3 & f[x_3] & & \end{array}$$

Los coeficientes del polinomio (3.2) pueden leerse desde el borde superior del triángulo formado en la obtención de las diferentes diferencias.

► EJEMPLO 3.3 Use las diferencias divididas para encontrar el polinomio de interpolación que pasa a través de los puntos (0, 1), (2, 2), (3,4).

Al aplicar las definiciones de las diferencias divididas se obtiene la siguiente tabla:

$$\begin{array}{c|c|c|c} 0 & 1 & & \\ & & \frac{1}{2} & \\ 2 & 2 & & \frac{1}{2} \\ & & 2 & \\ 3 & 4 & & \end{array}$$

Esta tabla se calcula de la manera siguiente: después de representar la tabulación de los puntos, calcule las siguientes columnas como diferencias divididas de izquierda a derecha, como en (3.3). Por ejemplo,

$$\frac{2-1}{2-0} = \frac{1}{2}$$

$$\frac{2-\frac{1}{2}}{3-0} = \frac{1}{2}$$

$$\frac{4-2}{3-2} = 2.$$

Después de completar el triángulo de diferencias divididas, los coeficientes del polinomio 1, 1/2, 1/2 pueden leerse desde el borde superior del triángulo de la tabla. El polinomio de interpolación puede escribirse como

$$P(x) = 1 + \frac{1}{2}(x-0) + \frac{1}{2}(x-0)(x-2),$$

o, en forma anidada,

$$P(x) = 1 + (x-0)\left(\frac{1}{2} + (x-2)\cdot\frac{1}{2}\right).$$

Los puntos básicos para la forma anidada (vea el capítulo 0) son $r_1 = 0$ y $r_2 = 2$. De manera alternativa, usando el álgebra, el polinomio de interpolación se puede escribir como

$$P(x) = 1 + \frac{1}{2}x + \frac{1}{2}x(x-2) = \frac{1}{2}x^2 - \frac{1}{2}x + 1,$$

que coincide con la versión de interpolación de Lagrange mostrada con anterioridad. ◄

Utilizando el concepto enfoque de las diferencias divididas, los nuevos puntos que se agreguen después de calcular el polinomio de interpolación original pueden incorporarse con facilidad.

► EJEMPLO 3.4 Agregue el cuarto punto de datos (1, 0) a la tabla del ejemplo 3.3.

La tabla de las diferencias divididas queda de la siguiente manera:

$$\begin{array}{c|cccc} 0 & 1 & & & \\ & & \frac{1}{2} & & \\ 2 & 2 & & \frac{1}{2} & \\ & & 2 & & -\frac{1}{2} \\ 3 & 4 & & 0 & \\ & & 2 & & \\ 1 & 0 & & & \end{array}$$

El resultado es un término nuevo que debe añadirse al polinomio original $P_2(x)$. Si el triángulo se lee desde su borde superior, se observa que el nuevo polinomio de interpolación de tercer grado es

$$P_3(x) = 1 + \frac{1}{2}(x-0) + \frac{1}{2}(x-0)(x-2) - \frac{1}{2}(x-0)(x-2)(x-3).$$

Tenga en cuenta que $P_3(x) = P_2(x) - \frac{1}{2}(x-0)(x-2)(x-3)$, que el polinomio anterior puede agregarse como parte del nuevo. ◄

Resulta interesante comparar el trabajo extra necesario para añadir un nuevo punto a la fórmula de Lagrange comparado con la fórmula de las diferencias divididas. Cuando se añade un punto nuevo, el polinomio de Lagrange debe reiniciarse desde el principio; ningún cálculo anterior puede utilizarse. Por otra parte, en la forma de las diferencias divididas, se conserva el trabajo anterior y se añade un nuevo término al polinomio. Por lo tanto, el enfoque de las diferencias divididas tiene una gran ventaja sobre Lagrange.

► EJEMPLO 3.5 Utilice las diferencias divididas de Newton para encontrar el polinomio de interpolación que pasa por $(0, 2), (1, 1), (2, 0), (3, -1)$.

La forma triangular de las diferencias divididas es

$$\begin{array}{c|cccc} 0 & 2 & & & \\ & & -1 & & \\ 1 & 1 & & 0 & \\ & & -1 & & 0 \\ 2 & 0 & & 0 & \\ & & -1 & & \\ 3 & -1 & & & \end{array}$$

Al leer los coeficientes, se encuentra que el polinomio de interpolación de tercer grado o menor es

$$P(x) = 2 + (-1)(x-0) = 2 - x,$$

lo que concuerda con el ejemplo 3.2, pero con mucho menos trabajo realizado. ◄

3.1.3 ¿Cuántos polinomios de grado *d* pasan por *n* puntos?

El teorema 3.2, el teorema principal de la interpolación polinomial, responde a esta pregunta si $0 \leq d \leq n-1$. Dado $n = 3$ puntos $(0, 1), (2, 2), (3, 4)$, hay un polinomio de interpolación de segundo grado o menor. En el ejemplo 3.1 se muestra que es de segundo grado, por lo que no hay polinomios de interpolación de grado 0 o 1 para los tres puntos dados.

¿Cuántos polinomios de tercer grado interpolan los mismos tres puntos? Una forma de construir un polinomio se desprende del análisis anterior: agregar un cuarto punto. Al extender el triángulo de las diferencias divididas de Newton se obtiene un nuevo coeficiente superior. En el ejemplo 3.4 se añadió el punto $(1, 0)$. El polinomio resultante,

$$P_3(x) = P_2(x) - \frac{1}{2}(x-0)(x-2)(x-3), \tag{3.4}$$

que pasa por los tres puntos en cuestión, si agregamos el punto (1, 0). Habrá por lo menos un polinomio de tercer grado que pasa a través de los tres puntos originales (0, 1), (2, 2), (3,4).

Por supuesto, existen muchas maneras en las que se podría haber elegido el cuarto punto. Por ejemplo, si se mantiene el mismo $x_4 = 1$ y simplemente se cambia $y_4 = 0$, se *debe* obtener un polinomio de interpolación de tercer grado diferente, puesto que una función sólo puede pasar por un valor de y en x_4. Ahora se sabe que hay un número infinito de polinomios que interpolan los tres puntos $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, dado que para cualquier x_4 fijo hay infinitas maneras en que se puede elegir y_4, donde cada uno proporciona un polinomio diferente. De esta manera se muestra que dados n puntos datos (x_i, y_i) con distintas x_i, hay un número infinito de polinomios de grado n que pasan a través de ellos.

Una segunda mirada a (3.4) sugiere una forma más directa de producir polinomios de interpolación de tercer grado a través de tres puntos. En lugar de añadir un cuarto punto para generar un término más, ¿por qué no sólo se escribe ese término?, ¿el resultado interpola los tres puntos originales? Sí, porque $P_2(x)$ lo hace, y el término nuevo se evalúa como cero en x_1, x_2 y x_3. Así que en realidad no hay necesidad de construir diferencias divididas de Newton adicionales para este fin. Cualquier polinomio de tercer grado con la forma

$$P_3(x) = P_2(x) + cx(x - 2)(x - 3)$$

con $c \neq 0$ pasará a través de (0, 1), (2, 2) y (3, 4). De esta forma se construirán con facilidad (muchos de manera infinita) polinomios de grado $\geq n$ para n puntos de datos dados, como se ilustra en el siguiente ejemplo.

► **EJEMPLO 3.6** ¿Cuántos polinomios de cada grado $0 \leq d \leq 5$ pasan a través de los puntos $(-1, -5)$, $(0, -1)$, $(2, 1)$ y $(3, 11)$?

El triángulo de diferencias divididos de Newton es

$$\begin{array}{c|cccc} -1 & -5 & & & \\ & & 4 & & \\ 0 & -1 & & -1 & \\ & & 1 & & 1 \\ 2 & 1 & & 3 & \\ & & 10 & & \\ 3 & 11 & & & \end{array}$$

Así que no hay polinomios de interpolación de grado 0, 1 o 2 y el único de tercer grado es

$$P_3(x) = -5 + 4(x + 1) - (x + 1)x + (x + 1)x(x - 2).$$

Hay un número infinito de polinomios de interpolación de cuarto grado

$$P_4(x) = P_3(x) + c_1(x + 1)x(x - 2)(x - 3)$$

para las $c_1 \neq 0$ arbitrarias, y una cantidad infinita de polinomios de interpolación de quinto grado

$$P_5(x) = P_3(x) + c_2(x + 1)x^2(x - 2)(x - 3)$$

para las $c_2 \neq 0$ arbitrarias. ◄

3.1.4 Código para la interpolación

A continuación se presenta el programa `newtdd.m` de MATLAB para calcular los coeficientes:

```
%Programa 3.1 Método de interpolación de diferencia dividida de Newton
%Calcula los coeficientes del polinomio de interpolación
%Entrada: x y y son vectores que contienen las coordenadas x y y
%          de los n puntos de datos
%Salida: coeficientes c del polinomio de interpolación en forma anidada
%Uso con nest.m para evaluar el polinomio de interpolación
function c=newtdd(x, y, n)
for j=1:n
  v(j,1)=y(j);          % Llene la columna y del triángulo de Newton
end
for i=2:n               % Para la columna i,
  for j=1:n+1-i         % llene la columna de arriba a abajo
    v(j,i)=(v(j+1,i-1)-v(j,i-1))/(x(j+i-1)-x(j));
  end
end
for i=1:n
  c(i)=v(1,i);          % Lea a lo largo de la parte superior del triángulo
end                     % para los coeficientes de salida
```

Este programa puede aplicarse a los puntos del ejemplo 3.3 para obtener los coeficientes 1, 1/2, 1/2 que se encontraron anteriormente. Estos coeficientes pueden usarse en el programa de multiplicación anidada a fin de evaluar el polinomio de interpolación en varios valores de x.

Por ejemplo, el segmento de código en MATLAB

```
x0=[0 2 3];
y0=[1 2 4];
c=newtdd(x0,y0,3);
x=0:.01:4;
y=nest(2,c,x,x0);
plot(x0,y0,'o',x,y)
```

Proporciona la gráfica del polinomio que se muestra en la figura 3.1.

ANOTACIÓN

Resumen Éste es el primer encuentro con el concepto de análisis numérico. En principio, la interpolación puede no parecer fácil de comprender. Después de todo, se tienen n puntos como entrada y se entregan n términos (del polinomio de interpolación) como salida. ¿Qué se ha entendido?

Piense en cada uno de los puntos como provenientes de alguna parte representativa, elegida de entre la gran cantidad de puntos sobre una curva $y = f(x)$. El polinomio de grado $n - 1$, caracterizado por n términos, es una representación de $f(x)$, y en algunos casos puede usarse como un representante bastante simple de $f(x)$ para fines de cálculo.

Por ejemplo, ¿qué sucede cuando se oprime la tecla SIN en una calculadora? La calculadora tiene hardware para sumar y multiplicar, pero ¿cómo calcular el seno de un número? De alguna manera, la operación debe reducirse a la evaluación de un polinomio, que requiere exactamente esas operaciones. Al elegir los puntos situados en la curva sinusoidal, es posible calcular y almacenar un polinomio de interpolación en la calculadora como una versión aproximada de la función seno.

Este tipo de representación se denomina "aproximación polinomial con error", lo que significa que habrá un error involucrado puesto que la función seno no es en realidad un polinomio. El tema de la siguiente sección es estimar el error que se produce cuando una función $f(x)$ se sustituye por un polinomio de interpolación.

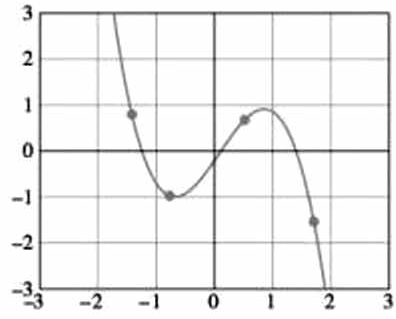


Figura 3.2 Programa de interpolación 3.2 usando como entrada el mouse. Captura de pantalla del código `clickinterp.m` de MATLAB con cuatro puntos de entrada.

Ahora que ya se tiene el código de MATLAB para encontrar los términos del polinomio de interpolación (`newtdd.m`) y para evaluar el polinomio (`nest.m`), es posible reunirlos a fin de crear una rutina de interpolación polinómica. El programa `clickinterp.m` utiliza la capacidad gráfica de MATLAB para graficar el polinomio de interpolación, mientras se está creando. Vea la figura 3.2. El comando de entrada `ginput` de MATLAB se utiliza para facilitar la entrada de datos con el mouse.

```
%Programa 3.2. Programa de interpolación polinomial
%Haga clic en la ventana de figuras en MATLAB para localizar los puntos de datos.
%     Continúe añadiendo más puntos.
%     Presione regresar para terminar el programa.
function clickinterp
xl=-3;xr=3;yb=-3;yt=3;
plot([xl xr],[0 0],'k',[0 0],[yb yt],'k');grid on;
xlist=[];ylist=[];
k=0;       % inicializa el contador k
while(0==0)
  [xnew,ynew] = ginput(1);   % obtiene clic del ratón
  if length (xnew) <1
    break                    % si presiona regresar, terminar
  end
  k=k+1;          %  k cuenta los clics
  xlist(k)=xnew; ylist(k)=ynew; %  añade un nuevo punto a la lista
  c=newtdd(xlist,ylist,k); %  obtiene coeficientes de interpolación
  x=xl:.01:xr;    %  define las coordenadas x de la curva
  y=nest(k-1,c,x,xlist); %  obtiene las coordenadas y de la curva
  plot(xlist,ylist,'o',x,y,[xl xr],[0,0],'k',[0 0],[yb yt],'k');
  axis([xl xr yb yt]);grid on;
end
```

3.1.5 Representación de funciones mediante polinomios de aproximación

Un uso importante de la interpolación polinomial es reemplazar una función complicada por un polinomio, que implica sólo las operaciones elementales en computadora como suma, resta y multiplicación. Piense en esto como una forma de entender: algo complejo se sustituye por algo más simple y programable, tal vez con cierta pérdida de exactitud que será necesario analizar. Se iniciará con un ejemplo de trigonometría.

► **EJEMPLO 3.7** Interpole la función $f(x) = \operatorname{sen} x$ en 4 puntos equidistantes en $[0, \pi/2]$.

Se tabula la función seno en el intervalo $[0, \pi/2]$. Se toman cuatro puntos igualmente espaciados y se forma el triángulo de diferencias divididas. Se presentan los valores hasta cuatro posiciones correctas:

$$
\begin{array}{r|llll}
0 & 0.0000 & & & \\
 & & 0.9549 & & \\
\pi/6 & 0.5000 & & -0.2443 & \\
 & & 0.6990 & & -0.1139 \\
2\pi/6 & 0.8660 & & -0.4232 & \\
 & & 0.2559 & & \\
3\pi/6 & 1.0000 & & &
\end{array}
$$

Por lo tanto, el polinomio de interpolación de tercer grado es

$$
\begin{aligned}
P_3(x) &= 0 + 0.9549x - 0.2443x(x - \pi/6) - 0.1139x(x - \pi/6)(x - \pi/3) \\
&= 0 + x(0.9549 + (x - \pi/6)(-0.2443 + (x - \pi/3)(-0.1139))). \qquad (3.5)
\end{aligned}
$$

Este polinomio se representa de manera gráfica junto con la función seno en la figura 3.3. En este nivel de resolución, $P_3(x)$ y sen x son virtualmente indistinguibles en el intervalo $[0, \pi/2]$. En (3.5) se ha reducido la cantidad infinita de información contenida por la curva sinusoidal en unos cuantos términos almacenados realizando 3 sumas y 3 multiplicaciones. ◄

¿Qué tan cerca se está de diseñar la tecla `sin` en una calculadora? Ciertamente es necesario conocer los datos de entrada sobre el eje x. Sin embargo, debido a las simetrías de la función seno, ya se ha hecho la parte difícil. El intervalo $[0, \pi/2]$ es el **dominio de la función** seno, lo que significa que una entrada en cualquier otro intervalo puede reflejarse en dicho dominio. Por ejemplo, dada una entrada x de $[\pi/2, \pi]$, es posible calcular sen x como sen$(\pi - x)$, puesto que el seno es simétrico respecto de $x = \pi/2$. Dada una entrada x de $[\pi, 2\pi]$, sen $x = -\text{sen}(2\pi - x)$ debido a la antisimetría alrededor de $x = \pi$. Por último, debido a que el seno repite su comportamiento en el intervalo $[0, 2\pi]$ en todo el eje x, éste se puede calcular para cualquier entrada al reducir primero el módulo 2π. Lo anterior conduce a un diseño sencillo para la tecla `sin`:

```
%Programa 3.3 programación de la tecla sin en calculadora, intento #1
%Se aproxima la curva sen con un polinomio de grado 3
%    (Precaución: no utilizar para construir puentes,
%    al menos hasta que se haya analizado su precisión).
%Entrada: x
%Salida: aproximación por sen(x)
function y=sin1(x)
%Primero calcula el polinomio de interpolación y
%    almacena coeficientes
b=pi*(0:3)/6;yb=sin(b);    % b conserva puntos básicos
c=newtdd(b,yb,4);
%Para cada entrada x, mueve x al dominio de la función y evalúa
%      el polinomio de interpolación
s=1;                       %Corrige el signo del seno
x1=mod(x,2*pi);
if x1>pi
  x1 = 2*pi-x1;
  s = -1;
end
if x1 > pi/2
  x1 = pi-x1;
end
y = s*nest(3,c,x1,b);
```

La mayor parte del trabajo en el programa 3.3 es colocar a x en el dominio de la función. Después, se evalúa el polinomio de tercer grado mediante la multiplicación anidada. A continuación se presentan algunas salidas típicas del programa 3.3:

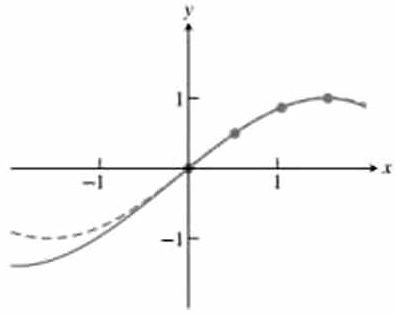


Figura 3.3 Interpolación de tercer grado de sen *x*. El polinomio de interpolación (curva continua) se representa a lo largo de $y = \text{sen } x$. Los nodos de interpolación igualmente espaciados son 0, $\pi/6$, $2\pi/6$ y $3\pi/6$. La aproximación es muy cercana entre 0 y $\pi/2$.

x	sen x	`sinl(x)`	error
1	0.8415	0.8411	0.0004
2	0.9093	0.9102	0.0009
3	0.1411	0.1428	0.0017
4	−0.7568	−0.7557	0.0011
14	0.9906	0.9928	0.0022
1000	0.8269	0.8263	0.0006

Esto no está mal para ser un primer intento. Por lo general, el error es menor al uno por ciento. Con el fin de obtener cifras correctas suficientes para llenar la mantisa de la calculadora, es necesario saber un poco más sobre el error de interpolación, que es el tema de la siguiente sección.

3.1 Ejercicios

1. Use la interpolación de Lagrange para encontrar el polinomio que pasa por los puntos.
   (a) (0, 1), (2, 3), (3, 0)
   (b) (−1, 0), (2, 1), (3, 1), (5, 2)
   (c) (0, −2), (2, 1), (4, 4)

2. Utilice diferencias divididas de Newton para encontrar los polinomios de interpolación de los puntos del ejercicio 1 y compruebe su concordancia con el polinomio de interpolación de Lagrange.

3. ¿Cuántos polinomios de grado d pasan a través de los cuatro puntos (−1, 3), (1, 1), (2, 3) (3, 7)? Escríbalos si es posible. (a) $d = 2$ (b) $d = 3$ (c) $d = 6$.

4. (a) Encuentre un polinomio $P(x)$ de tercer grado o menor cuya gráfica pase por los puntos (0, 0), (1, 1), (2, 2), (3, 7). (b) Encuentre otros dos polinomios (de cualquier grado) que pasen a través de estos cuatro puntos. (c) Decida si existe un polinomio $P(x)$ de tercer grado o menor cuya gráfica pase por los puntos (0, 0), (1, 1), (2, 2), (3, 7) y (4, 2).

5. (a) Encuentre un polinomio $P(x)$ de tercer grado o menor cuya gráfica pase por los cuatro puntos de datos (−2, 8), (0, 4), (1, 2), (3, −2). (b) Describa otros polinomios cualesquiera de cuarto grado o menor que pasen a través de los cuatro puntos indicados en el inciso (a).

6. Escriba un polinomio de quinto grado exactamente que interpole los cuatro puntos (1, 1), (2, 3), (3, 3), (4, 4).

7. Encuentre $P(0)$, donde $P(x)$ es el polinomio de décimo grado que es igual a cero en $x = 1, \ldots, 10$ y satisface $P(12) = 44$.

8. Sea $P(x)$ el polinomio de noveno grado que toma el valor 112 en $x = 1$, el valor 2 en $x = 10$, y es igual a cero para $x = 2, \ldots, 9$. Calcule $P(0)$.

9. Dé un ejemplo de lo siguiente, o explique por qué no existe tal ejemplo. (a) Un polinomio $L(x)$ de sexto grado que sea igual a cero en $x = 1, 2, 3, 4, 5, 6$ e igual a 10 en $x = 7$. (b) Un polinomio $L(x)$ de sexto grado que sea cero en $x = 1, 2, 3, 4, 5, 6$, igual a 10 en $x = 7$ e igual a 70 en $x = 8$.

10. Sea $P(x)$ el polinomio de quinto grado que toma el valor 10 en $x = 1, 2, 3, 4, 5$ y el valor 15 en $x = 6$. Encuentre $P(7)$.

11. Sean P_1, P_2, P_3 y P_4 cuatro diferentes puntos que se encuentran sobre una parábola $y = ax^2 + bx + c$. ¿Cuántos polinomios cúbicos (de tercer grado) pasan a través de esos cuatro puntos? Explique su respuesta.

12. ¿Puede un polinomio de tercer grado intersectar a un polinomio de cuarto grado en exactamente cinco puntos? Explique.

13. Sea $P(x)$ el polinomio de décimo grado que pasa a través de los 11 puntos
$(-5, 5), (-4, 5), (-3, 5), (-2, 5), (-1, 5), (0, 5), (1, 5), (2, 5), (3, 5), (4, 5), (5, 42)$.
Calcule $P(6)$.

14. Escriba 4 puntos no colineales $(1, y_1), (2, y_2), (3, y_3), (4, y_4)$ que no se encuentren sobre ningún polinomio $y = P_3(x)$ de tercer grado exactamente.

15. Escriba el polinomio de vigesimoquinto grado que pasa por los puntos $(1, -1), (2, -2), \ldots, (25, -25)$ y tiene un término constante igual a 25.

16. Muestre todos los polinomios de grado 42 que pasan a través de los once puntos $(-5, 5), (-4, 4), \ldots, (4, -4), (5, -5)$ y tiene un término constante igual a 42.

17. En la tabla siguiente se muestra la estimación de la concentración atmosférica media de dióxido de carbono en la atmósfera terrestre, en partes por millón por volumen. Encuentre el polinomio de tercer grado que interpole los datos y úselo para estimar la concentración de CO_2 en (a) 1950 y (b) 2050. (La concentración real en 1950 fue de 310 ppm).

año	CO_2 (ppm)
1800	280
1850	283
1900	291
2000	370

18. En la tabla siguiente se muestra la vida útil prevista de un ventilador industrial cuando se opera a la temperatura indicada. Estime la vida útil a 70 °C usando (a) la parábola de los tres últimos puntos de los datos (b) la curva de tercer grado correspondiente a los cuatro puntos.

temp (°C)	hrs (×1000)
25	95
40	75
50	63
60	54

3.1 Problemas de computadora

1. Aplique los siguientes datos de población del mundo para estimar la población de 1980 usando (a) la línea recta que pasa por las estimaciones de 1970 y 1990; (b) la parábola a través de las estimaciones de 1960, 1970 y 1990; y (c) la curva cúbica a través de los cuatro puntos de datos. Compare los resultados con la estimación de 1980: 4452584592.

año	población
1960	3039585530
1970	3707475887
1990	5281653820
2000	6079603571

2. Escriba una versión del programa 3.2 que sea una función de MATLAB, cuyas entradas `x` y `y` sean vectores con el mismo número de puntos y cuya salida sea una gráfica del polinomio de interpolación. De esta manera, los puntos pueden introducirse con mayor precisión usando como entrada el mouse. Verifique su programa con el de la figura 3.2.

3. Escriba una función `polyinterp.m` de MATLAB que tome como entrada un conjunto de puntos de interpolación (x, y) y otro punto x_0, que tenga como salida y_0, el valor del polinomio de interpolación en x_0. La primera línea del programa debe ser `function y0 = polyinterp(x, y, x0)`, donde `x` y `y` son los vectores de entrada de los puntos de datos. Su función puede llamar a `newtdd` del programa 3.1 y `nest` del capítulo 0, y puede estar estructurado de manera similar al programa 3.2, pero sin las gráficas. Pruebe que el programa funciona.

4. Remodele la tecla de calculadora `sin x` del intento 1 en el programa 3.3 para construir `cos1`, una tecla de coseno que sigue los mismos principios. Primero decida el dominio de la función para el coseno.

5. (a) Utilice las propiedades de adición para sen y cos a fin de demostrar que $\tan(\pi/2 - x) = 1/\tan x$. (b) Demuestre que $[0, \pi/4]$ puede usarse como dominio para tan x. (c) Diseñe una tecla para la tangente bajo los principios del programa 3.3, usando la interpolación polinomial de tercer grado en su dominio de la función. (d) Calcule empíricamente el error máximo de la tecla de tangente en $[0, \pi/4]$.

3.2 ERROR DE INTERPOLACIÓN

La precisión de la tecla `sin` de la calculadora depende de la aproximación a la figura 3.3. ¿Qué tan cerca está? Se presenta un cuadro que indica que, para algunos ejemplos, los dos primeros dígitos son bastante confiables, pero después las cifras no siempre son correctas. En esta sección se estudian las formas de medir este error y determinar la manera de reducirlo.

3.2.1 Fórmula del error en la interpolación

Suponga que se inicia con una función $y = f(x)$ y se toman los puntos de ésta para construir un polinomio de interpolación $P(x)$, como se hizo con $f(x) = \text{sen } x$ en el ejemplo 3.7. El error en la **interpolación** en x es $f(x) - P(x)$, la diferencia entre la función original que proporciona los puntos y el polinomio de interpolación, evaluado en x. El error de interpolación es la distancia vertical entre las curvas de la figura 3.3. El siguiente teorema proporciona una fórmula para el error de interpolación que por lo regular es imposible de evaluar con exactitud, pero a menudo puede conducir por lo menos a un límite del error.

TEOREMA 3.4 Suponga que $P(x)$ es el polinomio de interpolación (de grado $n-1$ o menor) que ajusta los n puntos $(x_1, y_1), \ldots, (x_n, y_n)$. El error de interpolación es

$$f(x) - P(x) = \frac{(x - x_1)(x - x_2)\cdots(x - x_n)}{n!} f^{(n)}(c), \tag{3.6}$$

donde c se encuentra entre el más pequeño y el más grande de los datos en $x, x_1, \ldots, x_n$. ■

Vea en la sección 3.2.2 una demostración del teorema 3.3. Este teorema puede usarse para evaluar la exactitud de la tecla `sin` que se construyó en el ejemplo 3.7. A partir de la ecuación (3.6) se obtiene

$$\operatorname{sen} x - P(x) = \frac{(x - 0)\left(x - \frac{\pi}{6}\right)\left(x - \frac{\pi}{3}\right)\left(x - \frac{\pi}{2}\right)}{4!} f'''(c),$$

donde $0 < c < \pi/2$. La cuarta derivada $f''''(c) = \sin c$ varía de 0 a 1 en este rango. En el peor de los casos, $|\operatorname{sen} c|$ no es mayor que 1, por lo que se puede estar seguro de un límite superior en el error de interpolación:

$$|\operatorname{sen} x - P(x)| \leq \frac{\left|(x - 0)\left(x - \frac{\pi}{6}\right)\left(x - \frac{\pi}{3}\right)\left(x - \frac{\pi}{2}\right)\right|}{24} |1|.$$

En $x = 1$, el error en el peor caso es

$$|\operatorname{sen} 1 - P(1)| \leq \frac{\left|(1 - 0)\left(1 - \frac{\pi}{6}\right)\left(1 - \frac{\pi}{3}\right)\left(1 - \frac{\pi}{2}\right)\right|}{24} |1| \approx 0.0005348. \tag{3.7}$$

Ésta es una cota superior para el error, puesto que se usó un límite en el "peor de los casos" para la cuarta derivada. Observe que el error real en $x = 1$ fue .0004, que está dentro del límite de error dado por (3.7). Es posible obtener algunas conclusiones sobre la fórmula del error de interpolación. Se esperan errores más pequeños cuando x está más cerca del centro del intervalo de las x_i que cuando está cerca de uno de los extremos, porque habrá términos más pequeños en el producto. Por ejemplo, se compara el límite del error anterior con el caso $x = 0.2$, que está cerca del extremo izquierdo del rango de los puntos dados. En este caso, la fórmula del error es

$$|\operatorname{sen} 0.2 - P(0.2)| \leq \frac{\left|(.2 - 0)\left(.2 - \frac{\pi}{6}\right)\left(.2 - \frac{\pi}{3}\right)\left(.2 - \frac{\pi}{2}\right)\right|}{24} |1| \approx 0.00313,$$

alrededor de seis veces más grande. Donde se observa que, el error real es mayor.

$$|\operatorname{sen} 0.2 - P(0.2)| = |0.19867 - 0.20056| = 0.00189.$$

► EJEMPLO 3.8 Encuentre una cota para el error $x = 0.25$ y $x = 0.75$ entre $f(x) = e^x$ y el polinomio que interpola los puntos $-1, -0.5, 0, 0.5, 1$.

No es necesario construir el polinomio de interpolación que se muestra en la figura 3.4 para encontrar el error. La fórmula del error de interpolación (3.6) da

$$f(x) - P_4(x) = \frac{(x + 1)\left(x + \frac{1}{2}\right)x\left(x - \frac{1}{2}\right)(x - 1)}{5!} f^{(5)}(c),$$

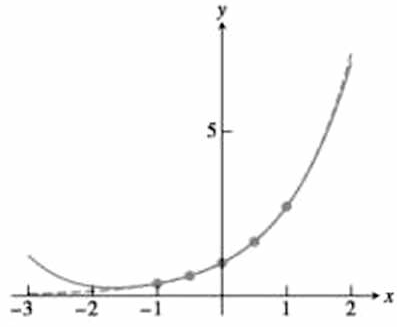


Figura 3.4 Polinomio de interpolación para aproximar $f(x) = e^x$. Los puntos igualmente espaciados -1, $-0.5, 0, 0.5, 1$. En donde la curva continua es el polinomio de interpolación.

donde $-1 < c < 1$. La quinta derivada es $f^{(5)}(c) = e^c$. Puesto que e^x crece con x, su máximo está a la derecha del intervalo, de modo que $|f^{(5)}| \leq e^1$ en $[-1, 1]$. Para $-1 \leq x \leq 1$, la fórmula del error resulta

$$|e^x - P_4(x)| \leq \frac{(x+1)\left(x+\frac{1}{2}\right)x\left(x-\frac{1}{2}\right)(x-1)}{5!}e.$$

En $x = 0.25$, el error de interpolación tiene el límite superior

$$\begin{aligned}|e^{0.25} - P_4(0.25)| &\leq \frac{(1.25)(0.75)(0.25)(-0.25)(-0.75)}{120}e \\ &\approx .000995.\end{aligned}$$

En $x = 0.75$, el error de interpolación es mayor:

$$\begin{aligned}|e^{0.75} - P_4(0.75)| &\leq \frac{(1.75)(1.25)(0.75)(0.25)(0.25)}{120}e \\ &\approx .002323.\end{aligned}$$

Observe de nuevo que el error de interpolación tenderá a ser más pequeño cerca del centro del intervalo de interpolación. ◀

3.2.2 Demostración de la forma de Newton y la fórmula del error

En esta sección se explica el razonamiento bajo dos hechos importantes que se han utilizado hasta ahora. En primer lugar, se establece la forma de las diferencias divididas de Newton del polinomio de interpolación, y después se demuestra el error en la fórmula de interpolación.

Recuerde lo que sabe hasta ahora. Si $x_1, \ldots, x_n$ son n puntos distintos en el eje x y $y_1, \ldots, y_n$ son arbitrarias, se sabe por el teorema 3.2 que hay exactamente un polinomio de interpolación (con grado máximo $n-1$) $P_{n-1}(x)$ para estos puntos. También se sabe que la fórmula de interpolación de Lagrange da un polinomio de ese tipo.

Aún falta demostrar que la fórmula de las diferencias divididas de Newton también da un polinomio de interpolación. Una vez que se compruebe que lo hace en el teorema 3.5, se sabrá que debe concordar con la versión de Lagrange.

Sea $P(x)$ el polinomio (único) que interpola $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$, y como en la definición 3.3, se denota con $f[x_1 \ldots x_n]$ a los términos del polinomio de grado $n-1$ de $P(x)$. Por lo tanto, $P(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{n-1}x^{n-1}$, donde $a_{n-1} = f[x_1 \ldots x_n]$, y dos hechos son fácilmente evidentes.

HECHO 1 $f[x_1 \ldots x_n] = f[\sigma(x_1) \ldots \sigma(x_n)]$ para cualquier permutación σ de las x_i. ❐

Demostración. Resulta evidente por la unicidad del polinomio de interpolación, demostrada en el teorema 3.2. ❐

HECHO 2 $P(x)$ puede escribirse en la forma

$$P(x) = c_0 + c_1(x - x_1) + c_2(x - x_1)(x - x_2) + \ldots + c_{n-1}(x - x_1)\cdots(x - x_{n-1}).$$

❐

Demostración. Es claro que debe elegirse $c_{n-1} = a_{n-1}$. Las restantes $c_{n-2}, c_{n-3}, \ldots, c_0$ se definen en forma recursiva al establecer c_k como los términos de grado k del polinomio (con grado máximo k).

$$P(x) - c_{n-1}(x - x_1)\cdots(x - x_{n-1}) - c_{n-2}(x - x_1)\cdots(x - x_{n-2})$$
$$- \ldots - c_{k+1}(x - x_1)\cdots(x - x_{k+1}).$$

(Éste es un polinomio con grado máximo k debido a la elección de c_{k+1}). ❐

TEOREMA 3.5 Sea $P(x)$ el polinomio de interpolación de $(x_1, f(x_1)), \ldots, (x_n, f(x_n))$ donde las x_i son distintas. Entonces,

(a) $P(x) = f[x_1] + f[x_1x_2](x - x_1) + f[x_1x_2x_3](x - x_1)(x - x_2) + \ldots$
$+ f[x_1x_2 \ldots x_n](x - x_1)(x - x_2)\cdots(x - x_{n-1})$, y

(b) para $k > 1$, $f[x_1 \ldots x_k] = \dfrac{f[x_2 \ldots x_k] - f[x_1 \ldots x_{k-1}]}{x_k - x_1}$. ■

Demostración. (a) Debe demostrarse que $c_{k-1} = f[x_1 \ldots x_k]$ para $k = 1, \ldots, n$. Por definición, ya está claro para $k = n$. En general, se sustituye sucesivamente $x_1, \ldots, x_k$ en la forma de $P(x)$ del hecho 2. Sólo los primeros k términos son distintos de cero. Se llega a la conclusión de que el polinomio que consta de los primeros k términos de $P(x)$ es suficiente para interpolar $x_1, \ldots, x_k$; entonces, por la definición 3.2 y la unicidad del polinomio de interpolación, $c_{k-1} = f[x_1 \ldots x_k]$. (b) De acuerdo con (a), el polinomio de interpolación de $x_2, x_3, \ldots, x_{k-1}, x_1, x_k$ es

$$P_1(x) = f[x_2] + f[x_2\, x_3](x - x_2) + \ldots + f[x_2\, x_3 \ldots x_{k-1}x_1](x - x_2)\cdots(x - x_{k-1})$$
$$+ f[x_2\, x_3 \ldots x_{k-1}x_1x_k](x - x_2)\cdots(x - x_{k-1})(x - x_1)$$

y el polinomio de interpolación de $x_2, x_3, \ldots, x_{k-1}, x_k, x_1$ es

$$P_2(x) = f[x_2] + f[x_2\, x_3](x - x_2) + \ldots + f[x_2\, x_3 \ldots x_{k-1}x_k](x - x_2)\cdots(x - x_{k-1})$$
$$+ f[x_2\, x_3 \ldots x_{k-1}x_kx_1](x - x_2)\cdots(x - x_{k-1})(x - x_k).$$

Por igualdad de polinomios, $P_1 = P_2$. Al establecer $P_1(x_k) = P_2(x_k)$ y cancelar términos se obtiene

$$f[x_2 \ldots x_{k-1}x_1](x_k - x_2)\cdots(x_k - x_{k-1}) + f[x_2 \ldots x_{k-1}x_1x_k](x_k - x_2)$$
$$\cdots(x_k - x_{k-1})(x_k - x_1) = f[x_2 \ldots x_k](x_k - x_2)\cdots(x_k - x_{k-1})$$

o bien

$$f[x_2 \ldots x_{k-1}x_1] + f[x_2 \ldots x_{k-1}x_1x_k](x_k - x_1) = f[x_2 \ldots x_k].$$

Usando el hecho 1, esto puede reordenarse como

$$f[x_1 \dots x_k] = \frac{f[x_2 \dots x_k] - f[x_1 \dots x_{k-1}]}{x_k - x_1}.$$

❐

A continuación se demuestra el teorema del error en la interpolación 3.4. Consideremos sumar un punto x al conjunto de puntos de interpolación. El nuevo polinomio de interpolación sería

$$P_n(t) = P_{n-1}(t) + f[x_1 \dots x_n x](t - x_1) \cdots (t - x_n).$$

Evaluado en el punto x adicional, $P_n(x) = f(x)$, por lo que

$$f(x) = P_{n-1}(x) + f[x_1 \dots x_n x](x - x_1) \cdots (x - x_n). \tag{3.8}$$

Esta fórmula es verdadera para todo x. Ahora se define

$$h(t) = f(t) - P_{n-1}(t) - f[x_1 \dots x_n x](t - x_1) \cdots (t - x_n).$$

Observe que $h(x) = 0$ por (3.8) y $0 = h(x_1) = \dots = h(x_n)$ porque P_{n-1} interpola a la función f en estos puntos. Entre cada par vecino de los $n + 1$ puntos $x, x_1, \dots, x_n$, debe haber un nuevo punto donde $h' = 0$, por el teorema de Rolle (vea el capítulo 0). Hay n de estos puntos. Entre cada par de ellos, debe haber un nuevo punto donde $h'' = 0$; hay $n - 1$ de éstos. Si se continúa de esta manera, debe haber un punto c para el que $h^{(n)}(c) = 0$, donde c se encuentra entre la más pequeña y la más grande de $x, x_1, \dots, x_n$. Observe que

$$h^{(n)}(t) = f^{(n)}(t) - n! f[x_1 \dots x_n x],$$

porque la n-ésima derivada del polinomio $P_{n-1}(t)$ es cero. Al sustituir c resulta

$$f[x_1 \dots x_n x] = \frac{f^{(n)}(c)}{n!},$$

lo que conduce a

$$f(x) = P_{n-1}(x) + \frac{f^{(n)}(c)}{n!}(x - x_1) \cdots (x - x_n),$$

utilizando (3.8).

3.2.3 Fenómeno de Runge

Los polinomios pueden ajustarse a cualquier conjunto de puntos de datos, como lo muestra el teorema 3.2. Sin embargo, hay otras formas para representar los polinomios sobre las demás. Para entender mejor este punto nos ayudamos del programa 3.2. Grafique los puntos que permitan que la función sea igual a cero: en los puntos igualmente espaciados $x = -3, -2.5, -2, -1.5, \dots, 2.5, 3$, con excepción de $x = 0$, donde se obtiene un valor de 1. Los puntos están sobre el eje x, a excepción de un desplazamiento "protuberancia" triangular en $x = 0$, como se muestra en la figura 3.5.

El polinomio que pasa a través de los puntos graficados excepto entre 0 y 1, están con valor cero. Ésta es una ilustración del efecto llamado fenómeno de Runge. Por lo general, se utiliza para describir un "balanceo extremo del polinomio" que se asocia con la interpolación polinomial de grados altos en puntos uniformemente espaciados.

► EJEMPLO 3.9 Interpole $f(x) = 1/(1 + 12x^2)$ en puntos uniformemente espaciados en el intervalo $[-1, 1]$.

Éste se conoce como el **ejemplo de Runge**. La función tiene la misma forma general que el desplazamiento triangular de la figura 3.5. La figura 3.6 muestra el resultado de la interpolación,

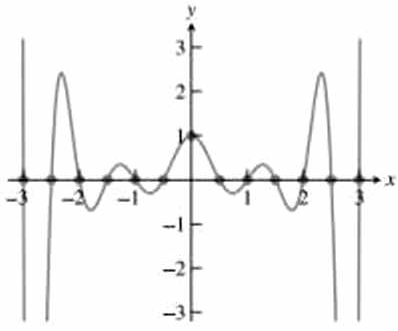


Figura 3.5 Interpolación de la función desplazada en forma triangular. El polinomio de interpolación oscila mucho más que los puntos introducidos.

el comportamiento característico del fenómeno de Runge: balanceo del polinomio cerca de los extremos del intervalo de interpolación. ◀

Como se ha visto, los ejemplos con el fenómeno de Runge tienen por característica un error mayor cerca de la parte exterior del intervalo de los datos. La solución para este problema es intuitiva: mover algunos de los puntos de interpolación hacia el exterior del intervalo, donde la función que produce los datos puede tener un mejor ajuste. En la siguiente sección se verá cómo lograr esto mediante la interpolación de Chebyshev.

3.2 Ejercicios

1. (a) Encuentre el polinomio de interpolación $P_2(x)$ de segundo grado que pasa por los puntos $(0, 0)$, $(\pi/2, 1)$ y $(\pi, 0)$. (b) Calcule $P_2(\pi/4)$, una aproximación para $\text{sen}(\pi/4)$. (c) Use el teorema 3.3 y proporcione una cota de error para la aproximación del inciso (b). (d) Use una calculadora o MATLAB y compare el error real con su cota de error.

2. (a) Dados los puntos $(1, 0)$, $(2, \ln 2)$, $(4, \ln 4)$, encuentre el polinomio de interpolación de segundo grado. (b) Use el resultado de (a) para aproximar $\ln 3$. (c) Use el teorema 3.3 y proporcione una cota de error para la aproximación del inciso (b). (d) Compare el error real con su cota de error calculado.

3. Suponga que el polinomio $P_9(x)$ interpola la función $f(x) = e^{-2x}$ en los 10 puntos igualmente espaciados $x = 0, 1/9, 2/9, 3/9, \ldots, 8/9, 1$. (a) Encuentre una cota superior para el error $|f(1/2) - P_9(1/2)|$. (b) ¿Cuántos decimales pueden garantizarse como correctos si se utiliza $P_9(1/2)$ para aproximar e?

4. Considere el polinomio de interpolación para $f(x) = 1/(x + 5)$ con nodos de interpolación $x = 0, 2, 4, 6, 8, 10$. Encuentre una cota superior para el error de interpolación en (a) $x = 1$ y (b) $x = 5$.

5. Suponga que una función $f(x)$ se ha aproximado mediante el polinomio de interpolación $P(x)$ de grado de 5, utilizando las parejas de puntos $(x_i, f(x_i))$, donde $x_1 = 0.1, x_2 = 0.2, x_3 = 0.3, x_4 = 0.4$, $x_5 = 0.5, x_6 = 0.6$. ¿Espera que el error de interpolación $|f(x) - P(x)|$ sea más pequeño para $x = 0.35$ o para $x = 0.55$? Cuantifique su respuesta.

6. Suponga que el polinomio $P_5(x)$ interpola una función $f(x)$ en seis puntos $(x_i, f(x_i))$ con las coordenadas x siguientes: $x_1 = 0, x_2 = 0.2, x_3 = 0.4, x_4 = 0.6, x_5 = 0.8$ y $x_6 = 1$. Suponga que el error de interpolación para $x = 0.3$ es $|f(0.3) - P_5(0.3)| = 0.01$. Estime el nuevo error de interpolación

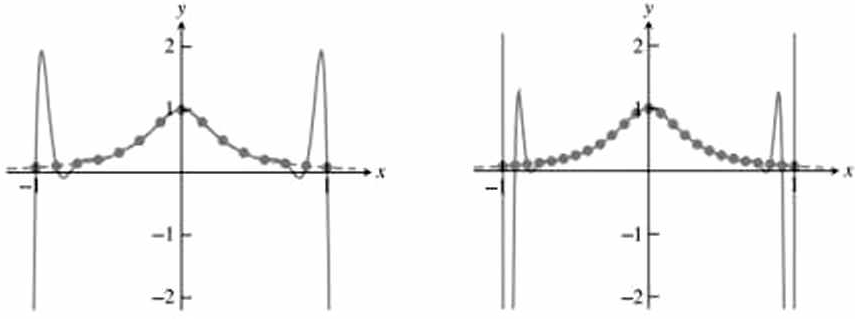


Figura 3.6 Ejemplo de Runge. El polinomio de interpolación de la función de Runge del ejemplo 3.9 ocasiona una gran variación en los puntos básicos uniformemente espaciados cercanos a los extremos del intervalo, en forma similar a la figura 3.5 (a) calculada con 15 puntos básicos (b) calculada con 25 puntos básicos.

$|f(0.3) - P_7(0.3)|$ que resultaría si se añadieran dos puntos de interpolación adicionales $(x_6, y_6) = (0.1, f(0.1))$ y $(x_7, y_7) = (0.5, f(0.5))$. ¿Qué supuestos se han hecho para generar esta estimación?

3.2 Problemas de computadora

1. (a) Utilice el método de las diferencias divididas para encontrar el polinomio de interpolación $P_4(x)$ de cuarto grado para los puntos (0.6, 1.433329), (0.7, 1.632316), (0.8, 1.896481), (0.9, 2.247908), y (1.0, 2.718282). (b) Calcule $P_4(0.82)$ y $P_4(0.98)$. (c) Los datos anteriores provienen de la función $f(x) = e^{x^2}$. Utilice la fórmula del error de interpolación para encontrar los límites superiores del error en $x = 0.82$ y $x = 0.98$ y compare con el error real, (d) Grafique el error real de interpolación $P(x) - e^{x^2}$ en los intervalos [.5, 1] y [0, 2].

2. Grafique el error de interpolación de la tecla `sen1` del programa 3.3 en el intervalo $[-2\pi, 2\pi]$.

3. En la tabla siguiente se muestra la producción mundial total de petróleo en millones de barriles diarios. Determine y grafique el polinomio de noveno grado que pasa por los datos. Úselo para estimar la producción de petróleo en 2010. ¿En este ejemplo se presenta el fenómeno de Runge? En su opinión, ¿el polinomio de interpolación es un buen modelo de los datos? Explique.

año	brls/día ($\times 10^6$)
1994	67.052
1995	68.008
1996	69.803
1997	72.024
1998	73.400
1999	72.063
2000	74.669
2001	74.487
2002	74.065
2003	76.777

4. Utilice el polinomio de tercer grado en los primeros cuatro puntos del problema de computadora 3 para estimar la producción mundial de petróleo en 1998. ¿Se presenta el fenómeno de Runge?

3.3 INTERPOLACIÓN DE CHEBYSHEV

Es común elegir los puntos básicos x_i para la interpolación uniformemente espaciados. En muchos casos, los datos que deben interpolarse sólo están disponibles en esa forma (por ejemplo, cuando los datos consisten en lecturas de instrumentos separados por un intervalo de tiempo constante). En otros casos (por ejemplo, la tecla seno) se tiene la libertad de elegir los puntos básicos que sean convenientes. La elección de la separación entre los puntos básicos puede tener un efecto significativo en el error de interpolación. La interpolación de Chebyshev se refiere a una forma particular de espaciado óptimo de los puntos.

3.3.1 Teorema de Chebyshev

La motivación para la interpolación de Chebyshev es mejorar el control del valor máximo del error de interpolación

$$\frac{(x - x_1)(x - x_2)\cdots(x - x_n)}{n!} f^{(n)}(c)$$

en el intervalo de interpolación. Por ahora, se fijará en $[-1, 1]$.

Los términos

$$(x - x_1)(x - x_2)\cdots(x - x_n) \tag{3.9}$$

de la fórmula del error de interpolación es en sí mismo un polinomio de grado n en x y tiene un valor máximo en $[-1, 1]$. ¿Es posible encontrar $x_1, \ldots, x_n$ particulares en $[-1, 1]$ que permiten que el valor máximo en (3.9) sea lo más pequeño posible? Esto se conoce como problema minimax de interpolación.

Por ejemplo, en la figura 3.7(a) se muestra una gráfica del polinomio de noveno grado (3.9) cuando $x_1, \ldots, x_9$ están uniformemente espaciados. La tendencia de este polinomio a ser grande cerca de los extremos del intervalo $[-1, 1]$ es una manifestación del fenómeno de Runge. En la figura 3.7(b) se muestra el mismo polinomio de (3.9), pero donde los puntos $x_1, \ldots, x_9$ se han elegido pasando por las raíces del polinomio en $[-1, 1]$. Los puntos han sido elegidos de acuerdo con el teorema 3.8, que se presentará en breve.

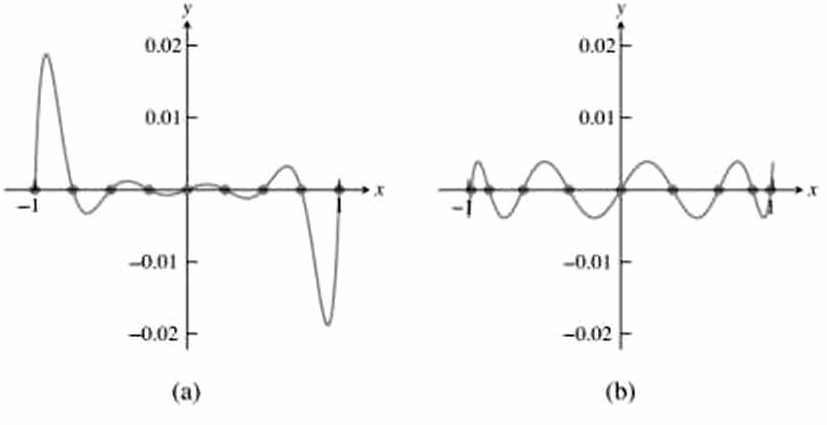


Figura 3.7 Parte de la fórmula del error de interpolación. Gráficas de $(x - x_1) \ldots (x - x_9)$ para (a) nueve puntos básicos x_i espaciados uniformemente (b) nueve raíces de Chebyshev x_i.

De hecho, este posicionamiento preciso, en el que los puntos básicos x_i se eligen como $\cos\frac{\pi}{18}, \cos\frac{3\pi}{18}, \ldots, \cos\frac{17\pi}{18}$, hace que el valor máximo absoluto de (3.9) sea igual a 1/256, el mínimo posible para nueve puntos en el intervalo $[-1, 1]$. Tal posicionamiento, debido a Chebyshev, se resume en el siguiente teorema:

TEOREMA 3.6 La selección de los valores $-1 \le x_1, \ldots, x_n \le 1$ que hace que el valor de

$$\max_{-1 \le x \le 1} |(x - x_1) \cdots (x - x_n)|$$

sea lo más pequeño posible es

$$x_i = \cos \frac{(2i - 1)\pi}{2n} \quad \text{para } i = 1, \ldots, n,$$

y el valor mínimo es $1/2^{n-1}$. De hecho, el mínimo es alcanzado por

$$(x - x_1) \cdots (x - x_n) = \frac{1}{2^{n-1}} T_n(x),$$

donde $T_n(x)$ denota el polinomio de Chebyshev de grado n. ■

La demostración de este teorema se da más adelante, después de establecer algunas propiedades de los polinomios de Chebyshev. A partir del teorema, se llega a la conclusión de que el error de interpolación puede minimizarse si los n puntos básicos de interpolación en $[-1, 1]$ se eligen como las raíces del polinomio de interpolación de Chebyshev $T_n(x)$ de grado n. Estas raíces son

$$x_i = \cos \frac{\text{impar } \pi}{2n} \tag{3.10}$$

donde "impar" indica los números impares de 1 a $2n - 1$. Entonces se garantiza que el valor absoluto de (3.9) sea menor que $1/2^{n-1}$ para toda x en $[-1, 1]$.

La elección de las raíces de Chebyshev como puntos básicos para la interpolación distribuye el error de interpolación tan uniformemente como sea posible en todo el intervalo $[-1, 1]$. Se llamará al polinomio de interpolación que utiliza las raíces de Chebyshev como puntos básicos el **polinomio de interpolación de Chebyshev**.

► **EJEMPLO 3.10** Encuentre una cota de error en el peor de los casos en el intervalo $[-1, 1]$ entre la función $f(x) = e^x$ y el polinomio de interpolación de Chebyshev de cuarto grado.

La fórmula del error de interpolación (3.6) establece

$$f(x) - P_4(x) = \frac{(x - x_1)(x - x_2)(x - x_3)(x - x_4)(x - x_5)}{5!} f^{(5)}(c),$$

donde

$$x_1 = \cos \frac{\pi}{10}, \quad x_2 = \cos \frac{3\pi}{10}, \quad x_3 = \cos \frac{5\pi}{10}, \quad x_4 = \cos \frac{7\pi}{10}, \quad x_5 = \cos \frac{9\pi}{10}$$

son las raíces del polinomio de Chebyshev en donde $-1 < c < 1$. De acuerdo con el teorema de Chebyshev 3.6, para $-1 \le x \le 1$,

$$|(x - x_1) \cdots (x - x_5)| \le \frac{1}{2^4}.$$

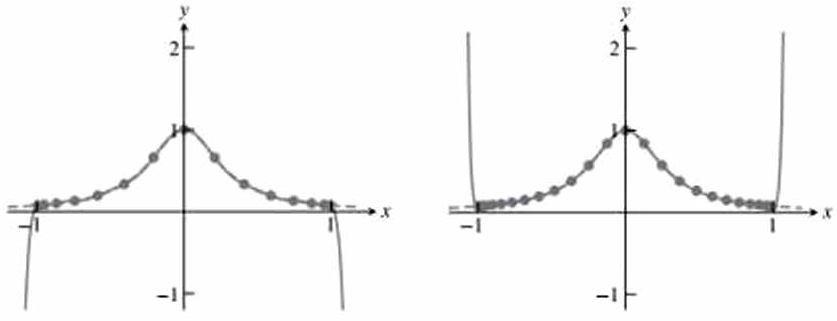
Además $|f^{(5)}| \le e^1$ en $[-1, 1]$. El error de interpolación está dado por

$$|e^x - P_4(x)| \le \frac{e}{2^4 5!} \approx 0.00142$$

para toda x en el intervalo $[-1, 1]$.

Compare este resultado con el del ejemplo 3.8. La cota del error de la interpolación de Chebyshev para todo el intervalo sólo es ligeramente mayor que la cota para un punto cercano al centro del intervalo, cuando se utiliza la interpolación uniformemente espaciada. Cerca de los extremos del intervalo, el error de Chebyshev es mucho menor. ◄

De regreso al ejemplo de Runge 3.9, es posible eliminar el fenómeno de Runge al elegir los puntos de interpolación de acuerdo con la idea de Chebyshev. En la figura 3.8 se muestra que el error de interpolación se reduce en todo el intervalo $[-1, 1]$.



(a) (b)

Figura 3.8 Ejemplo de interpolación de Runge con nodos de Chebyshev. La función de Runge $f(x) = 1/(1 + 12x^2)$ se grafica junto con su polinomio de interpolación de Chevyshev para (a) 15 puntos (b) 25 puntos. El error en $[-1, 1]$ es insignificante para la presente resolución. El deplazamieto del polinomio de la figura 3.6 se ha minimizado, al menos entre -1 y 1.

3.3.2 Polinomios de Chebyshev

Defina el **polinomio enésimo de Chebyshev** como $T_n(x) = \cos(n \arccos x)$. A pesar de su apariencia, es un polinomio en la variable x para cada n. Por ejemplo, para $n = 0$ da el polinomio 1 de grado 0, y para $n = 1$ se obtiene $T_1(x) = \cos(\arccos x) = x$. Para $n = 2$, recuerde la fórmula de adición del coseno $\cos(a + b) = \cos a \cos b - \operatorname{sen} a \operatorname{sen} b$. Establezca $y = \arccos x$, de modo que $\cos y = x$. Entonces $T_2(x) = \cos 2y = \cos^2 y - \operatorname{sen}^2 y = 2\cos^2 y - 1 = 2x^2 - 1$, un polinomio de segundo grado. En general, tenga en cuenta que

$$\begin{aligned} T_{n+1}(x) &= \cos(n+1)y = \cos(ny + y) = \cos ny \cos y - \operatorname{sen} ny \operatorname{sen} y \\ T_{n-1}(x) &= \cos(n-1)y = \cos(ny - y) = \cos ny \cos y - \operatorname{sen} ny \operatorname{sen}(-y). \end{aligned} \tag{3.11}$$

Como $\operatorname{sen}(-y) = -\operatorname{sen} y$, pueden sumarse las ecuaciones anteriores para obtener

$$T_{n+1}(x) + T_{n-1}(x) = 2\cos ny \cos y = 2x T_n(x). \tag{3.12}$$

Resultando,

$$T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x), \tag{3.13}$$

se llama la **relación recursiva** para los polinomios de Chebyshev. Existen varios hechos derivados de (3.13):

HECHO 1 Los T_n son polinomios. Esto se demostró explícitamente para T_0, T_1 y T_2. Como T_3 es una combinación polinomial de T_1 y T_2, T_3 también es un polinomio. El mismo argumento vale para todos los T_n. Los primeros polinomios de Chebyshev (vea la figura 3.9) son

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x. \end{aligned}$$

❐

HECHO 2 $\operatorname{grad}(T_n) = n$, y el coeficiente principal es 2^{n-1}. Esto resulta claro para $n = 1$ y 2, y la relación recursiva extiende el hecho para toda n. ❐

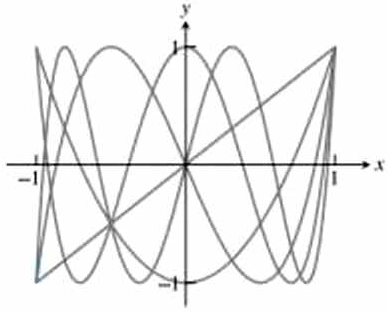


Figura 3.9 Gráficas de los polinomios de Chevishev de grado 1 a 5. Observe que $T_n(1) = 1$ y el máximo valor absoluto asumido por $T_n(x)$ dentro de $[-1, 1]$ es 1.

HECHO 3 $T_n(1) = 1$ y $T_n(-1) = (-1)^n$. Las dos igualdades son claras para $n = 1$ y 2. En general,

$$T_{n+1}(1) = 2(1)T_n(1) - T_{n-1}(1) = 2(1) - 1 = 1$$

y

$$\begin{aligned} T_{n+1}(-1) &= 2(-1)T_n(-1) - T_{n-1}(-1) \\ &= -2(-1)^n - (-1)^{n-1} \\ &= (-1)^{n-1}(2 - 1) = (-1)^{n-1} = (-1)^{n+1}. \end{aligned}$$

❐

HECHO 4 El valor máximo absoluto de $T_n(x)$ para $-1 \le x \le 1$ es 1. Esto se deriva inmediatamente del hecho de que $T_n(x) = \cos y$ para toda y.

HECHO 5 Todos los ceros de $T_n(x)$ se encuentran entre -1 y 1. Vea la figura 3.10. De hecho, los ceros son la solución de $0 = \cos(n \arccos x)$. Como $\cos y = 0$ si y sólo si $y =$ entero impar $\cdot (\pi/2)$, se encuentra que

$$\begin{aligned} n \arccos x &= \text{impar} \cdot \pi/2 \\ x &= \cos \frac{\text{impar} \cdot \pi}{2n}. \end{aligned}$$

❐

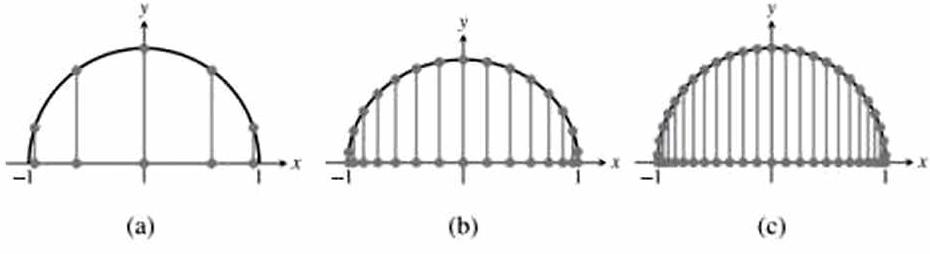


Figura 3.10 Ubicación de los ceros del polinomio de Chevyshev. Las raíces son las coordenadas x de los puntos uniformemente espaciados alrededor del círculo (a) de quinto grado, (b) de décimo quinto grado y (c) de vigésimo quinto grado.

HECHO 6 $T_n(x)$ alterna entre -1 y 1 un total de $n + 1$ veces. De hecho, esto ocurre en $\cos 0$, $\cos \pi/n$, ... , $\cos(n - 1)\pi/n$, $\cos \pi$.

Se deduce del hecho 2 que el polinomio $T_n(x)/2^{n-1}$ es mónico (tiene coeficiente principal 1). Dado que, según el hecho 5, todas las raíces de $T_n(x)$ son reales, puede escribirse que $T_n(x)/2^{n-1}$

en forma factorizada como $(x - x_1) \cdots (x - x_n)$ donde las x_i son los nodos de Chebyshev como se describen en el teorema 3.8.

El teorema de Chebyshev se deduce directamente de estos hechos.

Demostración del teorema 3.6. Sea $P_n(x)$ un polinomio mónico con un máximo absoluto aún más pequeño en $[-1, 1]$, en otras palabras, $|P_n(x)| < 1/2^{n-1}$ para $-1 \le x \le 1$. Este supuesto conduce a una contradicción. Como $T_n(x)$ alterna entre -1 y 1 un total de $n + 1$ veces (hecho 6), en estos $n + 1$ puntos la diferencia $P_n - T_n/2^{n-1}$ es alternativamente positiva y negativa. Por lo tanto, $P_n - T_n/2^{n-1}$ debe cruzar por cero al menos n veces, es decir, debe tener al menos n raíces. Esto contradice el hecho de que, debido a que P_n y $T_n/2^{n-1}$ son mónico, su diferencia es de grado $\le n - 1$.

3.3.3 Cambio de intervalo

Hasta ahora, el análisis de la interpolación de Chebyshev se ha restringido al intervalo $[-1, 1]$, porque el teorema 3.6 es más fácil de definir para este intervalo. A continuación, se cambiará considerando un intervalo general $[a, b]$.

Los puntos básicos se mueven de modo que tengan las mismas posiciones relativas en $[a, b]$ que tenían en $[-1, 1]$. Se hace en dos pasos: (1) *Expandir* los puntos por el factor $(b - a)/2$ (la relación de las dos longitudes de intervalo) y (2) *Trasladar* los puntos por $(b + a)/2$ para mover el centro de masa desde 0 hasta el punto medio de $[a, b]$. En otras palabras, desplazar los puntos originales

$$\cos \frac{\text{impar } \pi}{2n}$$

a

$$\frac{b - a}{2} \cos \frac{\text{impar } \pi}{2n} + \frac{b + a}{2}.$$

Con los nuevos puntos básicos de Chebyshev $x_1, \ldots, x_n$ en $[a, b]$, la cota superior del término de la fórmula del error de interpolación cambia debido a la ampliación por $(b - a)/2$ en cada factor $x - x_i$. En consecuencia, el valor minimax $1/2^{n-1}$ debe reemplazarse por $[(b - a)/2]^n/2^{n-1}$.

Nodos de interpolación de Chebyshev

En el intervalo $[a, b]$,

$$x_i = \frac{b + a}{2} + \frac{b - a}{2} \cos \frac{(2i - 1)\pi}{2n}$$

para $i = 1, \ldots, n$. La desigualdad

$$|(x - x_1) \cdots (x - x_n)| \le \frac{\left(\frac{b-a}{2}\right)^n}{2^{n-1}} \tag{3.14}$$

se cumple en $[a, b]$.

El siguiente ejemplo ilustra el uso de la interpolación de Chebyshev en un intervalo general.

► EJEMPLO 3.11 Encuentre los cuatro puntos básicos de Chebyshev para la interpolación en el intervalo $[0, \pi/2]$ y determine una cota superior del error de interpolación de Chebyshev para $f(x) = \operatorname{sen} x$ en el intervalo.

ANOTACIÓN

Resumen Como se muestra en esta sección, la interpolación de Chebyshev es una buena manera de convertir las funciones generales en polinomios para mayor facilidad de cálculo. La cota superior para el error cometido se determina con facilidad y usualmente es menor que para las interpolaciones uniformemente espaciadas, además puede hacerse tan pequeña como se desee.

Aunque para demostrar este proceso se ha utilizado la función seno, en la mayoría de las calculadoras y software de paquete se toma un enfoque diferente para construir la "tecla de seno" real. Las propiedades especiales de la función seno permiten que pueda aproximarse mediante una simple expansión de la serie de Taylor, ligeramente modificada para tomar en cuenta los efectos de redondeo. Debido a que el seno es una función impar, no existen términos con número par en su serie de Taylor alrededor de cero, por lo que su cálculo es muy eficiente.

Éste es un segundo intento. En el ejemplo 3.7 se utilizaron puntos básicos uniformemente espaciados. Los puntos básicos de Chebyshev son

$$\frac{\frac{\pi}{2}-0}{2}\cos\left(\frac{\text{impar }\pi}{2(4)}\right)+\frac{\frac{\pi}{2}+0}{2},$$

o bien

$$x_1=\frac{\pi}{4}+\frac{\pi}{4}\cos\frac{\pi}{8}, x_2=\frac{\pi}{4}+\frac{\pi}{4}\cos\frac{3\pi}{8}, x_3=\frac{\pi}{4}+\frac{\pi}{4}\cos\frac{5\pi}{8}, x_4=\frac{\pi}{4}+\frac{\pi}{4}\cos\frac{7\pi}{8}.$$

De acuerdo con (3.14), el error de interpolación en el peor de los casos para $0 \le x \le \pi/2$ es

$$|\operatorname{sen} x - P_3(x)| = \frac{|(x-x_1)(x-x_2)(x-x_3)(x-x_4)|}{4!}|f^{iv}(c)|$$
$$\le \frac{\left(\frac{\frac{\pi}{2}-0}{2}\right)^4}{4!2^3}1 \approx 0.00198.$$

El polinomio de interpolación de Chebyshev para este ejemplo se evalúa en varios puntos de la tabla siguiente:

x	sen x	$P_3(x)$	error
1	0.8415	0.8408	0.0007
2	0.9093	0.9097	0.0004
3	0.1411	0.1420	0.0009
4	−0.7568	−0.7555	0.0013
14	0.9906	0.9917	0.0011
1000	0.8269	0.8261	0.0008

Los errores de interpolación están muy por debajo de la estimación más pesimista. En la figura 3.11 se grafica el error de interpolación como una función de x en el intervalo $[0, \pi/2]$ comparado con el mismo error para una interpolación espaciada uniformemente. El error de Chebyshev (curva punteada) es un poco más pequeño y se distribuye de manera más uniforme a través de todo el intervalo de interpolación. ◄

► EJEMPLO 3.12 Diseñe la tecla del seno que dé una salida correcta hasta 10 decimales.

Gracias a nuestro trabajo realizado con anterioridad para la creación de un dominio fundamental para la función seno, es posible seguir concentrándose en el intervalo $[0, \pi/2]$. Repita el cálculo anterior, pero deje que n, el número de puntos básicos, sea una incógnita que debe determinarse. El error de interpolación máximo para el polinomio $P_{n-1}(x)$ en el intervalo $[0, \pi/2]$ es

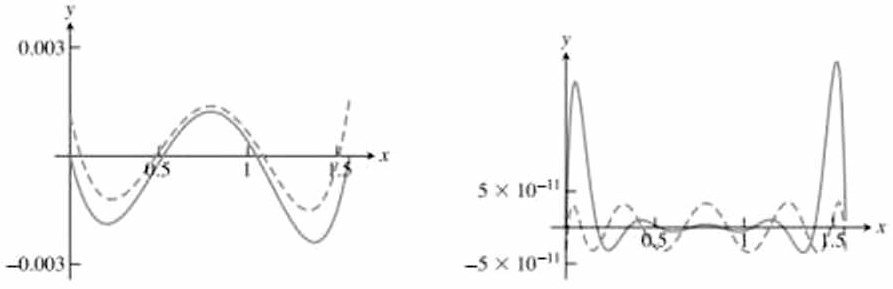


Figura 3.11 Error de interpolación al aproximar $f(x) = \text{sen } x$. (a) Error de interpolación para el polinomio de interpolación de tercer grado con puntos básicos uniformemente espaciados (curva sólida) y puntos básicos de Chevyshev (curva de punteada). (b) Igual que (a), pero de noveno grado.

$$|\text{sen } x - P_{n-1}(x)| = \frac{|(x - x_1)\cdots(x - x_n)|}{n!}|f^{(n)}(c)|$$
$$\leq \frac{\left(\frac{\pi}{2}-0\right)^n}{n!2^{n-1}}1.$$

Esta ecuación no es fácil de resolver para n, pero al aplicar un poco de ensayo y error se encuentra que para $n = 9$ la cota de error es $\approx 0.1224 \times 10^{-8}$ y para $n = 10$ es $\approx 0.4807 \times 10^{-10}$. Este último valor cumple con el criterio de 10 cifras decimales correctas. En la figura 3.11(b) se compara el error real del polinomio de interpolación de Chebyshev con el error del polinomio de interpolación uniformemente espaciado.

Los 10 puntos básicos de Chebyshev en $[0, \pi/2]$ son $\pi/4 + (\pi/4)$ cos (impar $\pi/20$). La tecla puede diseñarse para almacenar los 10 valores de y calculados para el seno en los puntos básicos y hacer una evaluación de multiplicación anidada para cada pulsación de la tecla. ◄

El siguiente programa `sen2.m` de MATLAB realiza la tarea anterior. El programa es un poco ineficiente tal como está escrito: deben hacerse 10 evaluaciones del seno, en los 10 nodos de Chebyshev, con el fin de establecer el polinomio de interpolación para aproximar el seno en un punto. Por supuesto, en una implementación real, estos números se calculan y almacenan una sola vez.

```
%Programa 3.4 Creación de una tecla de la calculadora para el seno, intento #2
%Aproximación a la curva de seno con un polinomio de grado 9
%Entrada: x
%Salida: aproximación para el sen(x), correcta hasta 10 posiciones decimales
function  y=sin2(x)
%Primero calcula el polinomio de interpolación y
%   almacena los coeficientes
n=10;
b=pi/4+(pi/4)*cos((1:2:2*n-1)*pi/(2*n));
yb=sin(b);                          % b contiene los puntos básicos de Chebyshev
c=newtdd(b,yb,n);
%Para cada entrada x, mueve x al dominio fundamental y evalúa
%   el polinomio de interpolación
s=1;                                % Corrige el signo del seno
x1=mod(x,2*pi);
if x1>pi
  x1 = 2*pi-x1;
  s = -1;
```

```
end
if x1 > pi/2
  x1 = pi-x1;
end
y = s*nest(n-1,c,x1,b);
```

En este capítulo se ha ilustrado la interpolación polinomial, ya sea espaciada uniformemente o usando nodos de Chebyshev, con el fin de aproximar las funciones trigonométricas. Aunque la interpolación polinómica puede utilizarse para aproximar el seno y el coseno hasta una precisión arbitrariamente alta, la mayoría de las calculadoras utilizan un enfoque un poco más eficiente llamado el algoritmo de rotación digital de coordenadas en computadora (CORDIC, por sus siglas en inglés) (Volder [1959]). CORDIC es un método iterativo elegante, basado en la aritmética compleja, que puede aplicarse a varias funciones especiales. La interpolación polinomial sigue siendo una técnica sencilla y útil para aproximar funciones generales, así como representarlas y hacer fácil el manejo de sus datos.

3.3 Ejercicios

1. Encuentre los nodos de interpolación de Chebyshev $x_1, \ldots, x_n$ en el intervalo dado, (a) $[-1, 1], n = 6$ (b) $[-2, 2], n = 4$ (c) $[4, 12], n = 6$ (d) $[-0.3, 0.7], n = 5$

2. Encuentre la cota superior de $|(x - x_1) \ldots (x - x_n)|$ en los intervalos y los nodos de Chebyshev del ejercicio 1.

3. Suponga que se utiliza la interpolación de Chebyshev para encontrar un polinomio de interpolación $Q_5(x)$ de quinto grado en el intervalo $[-1, 1]$ para la función $f(x) = e^x$. Use la fórmula del error de interpolación para encontrar una estimación del error en el peor de los casos $|e^x - Q_5(x)|$ que sea válida para toda x en el intervalo $[-1, 1]$. ¿Cuántos dígitos después del punto decimal están correctos cuando se utiliza $Q_5(x)$ para aproximar e^x?

4. Responda las mismas preguntas que en el ejercicio 3, pero para el intervalo [0.6, 1.0].

5. Encuentre una cota superior para el error en [0, 2] si se utiliza el polinomio de interpolación de Chebyshev de tercer grado para aproximar $f(x) = \text{sen } x$.

6. Suponga que usted debe utilizar la interpolación de Chebyshev para encontrar un polinomio de interpolación $Q_3(x)$ de tercer grado para aproximar la función $f(x) = x^{-3}$ en el intervalo [3, 4]. (a) Escriba los puntos (x, y) que servirán como nodos de interpolación para Q_3. (b) Encuentre una estimación del error $|x^{-3} - Q_3(x)|$ en el peor de los casos, la cual sea válida para toda x en el intervalo [3, 4]. ¿Cuántos dígitos después del punto decimal están correctos cuando se utiliza $Q_3(x)$ para aproximar x^{-3}?

7. Suponga que usted está diseñando la tecla **ln** para una calculadora cuya pantalla muestra seis dígitos a la derecha del punto decimal. Encuentre el polinomio con menor grado d para el cual la interpolación de Chebyshev en el intervalo $[1, e]$ se aproximará cumpliendo con esta precisión.

8. Sea $T_n(x)$ el polinomio de Chebyshev de grado n. Encuentre una fórmula para $T_n(0)$.

9. Determine los valores siguientes: (a) $T_{999}(-1)$ (b) $T_{1000}(-1)$ (c) $T_{999}(0)$ (d) $T_{1000}(0)$ (e) $T_{999}(-1/2)$ (f) $T_{1000}(-1/2)$.

3.3 Problemas para computadora

1. Codifique el programa 3.3 para implementar el polinomio de interpolación de Chebyshev con cuatro nodos en el intervalo $[0, \pi/2]$. (Sólo debe cambiar una línea del programa). Después grafique el polinomio y la función seno en el intervalo $[-2, 2]$.

2. Programe en MATLAB para evaluar la función del coseno correcta hasta 10 lugares decimales, mediante la interpolación de Chebyshev. Comience por la interpolación en un dominio $[0, \pi/2]$ y amplíe su respuesta a las entradas entre -10^4 y 10^4. Es posible que desee utilizar una parte del código en MATLAB escrito para este capítulo.

3. Utilice el programa del ejercicio 2 para $\ln x$, con entradas x entre 10^{-4} y 10^4. Utilice $[1, e]$ como el dominio de la función. ¿Cuál es el grado del polinomio de interpolación que garantiza 10 dígitos correctos? Su programa debe empezar por encontrar un entero k de tal forma que $e^k \leq x \leq e^{k+1}$. Entonces, xe^{-k} se encontrará en el dominio. Demuestre la precisión de su programa comparándolo con el comando `log` de MATLAB.

4. Sea $f(x) = e^{|x|}$. Compare la interpolación uniformemente espaciada con la interpolación de Chebyshev al graficar polinomios de grado n de ambos tipos en el intervalo $[-1, 1]$, para $n = 10$ y 20. En el caso de la interpolación uniformemente espaciada, los puntos básicos de interpolación izquierdo y derecho deben ser -1 y 1. Mediante muestras con tamaño de paso de 0.01, cree los errores de interpolación empíricos para cada tipo y grafique una comparación. ¿Se presenta el fenómeno de Runge en este problema?

5. Realice los pasos del problema de computadora 4 para $f(x) = e^{-x^2}$.

3.4 SPLINES CÚBICAS

Las curvas o splines representan un enfoque alternativo a la interpolación de datos. En la interpolación polinomial se utiliza una fórmula única, dada por un polinomio, para cumplir con todos los puntos de datos. La idea de las splines es usar varias fórmulas, cada una correspondiente a un polinomio de grado bajo, para pasar a través de los puntos.

El ejemplo más simple de una spline es una spline lineal, en la que se "conectan los puntos" con segmentos de recta. Suponga que se dan una serie de puntos $(x_1, y_1) \ldots, (x_n, y_n)$ con $x_1 < \ldots < x_n$. Una spline lineal se compone de $n - 1$ segmentos de recta que se dibujan entre pares de puntos vecinos. En la figura 3.12(a) se muestra una spline lineal donde, entre cada par de puntos vecinos $(x_i, y_i), (x_{i+1}, y_{i+1})$, se traza la función lineal $y = a_i + bi\,x$ a través de los dos puntos. Los puntos de datos dados en la figura son (1, 2), (2, 1), (4, 4) y (5, 3), y la spline lineal está dada por

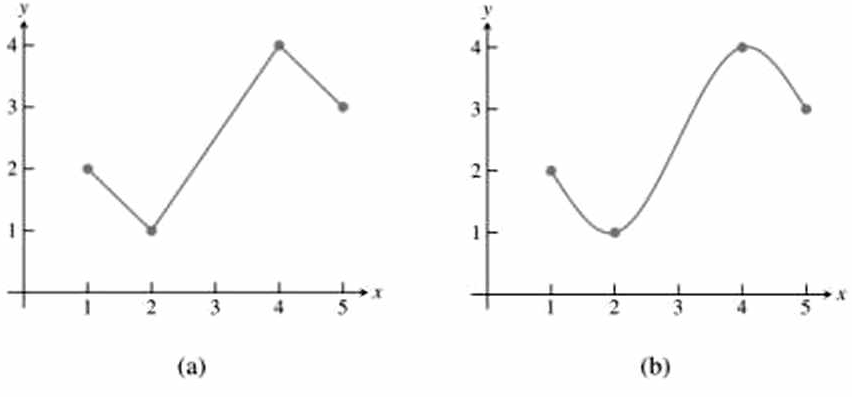


Figura 3.12 Splines a través de cuatro puntos de datos. (a) La spline lineal que pasa por (1, 2), (2, 1), (4, 4) y (5, 3) se compone de tres polinomios lineales dados por (3.15). (b) La spline cúbica a través de los mismos puntos está dada por (3.16).

$$S_1(x) = 2 - (x - 1) \text{ en } [1, 2]$$
$$S_2(x) = 1 + \frac{3}{2}(x - 2) \text{ en } [2, 4]$$
$$S_3(x) = 4 - (x - 4) \text{ en } [4, 5]. \tag{3.15}$$

La spline lineal interpola con éxito un conjunto arbitrario de n puntos. Sin embargo, las splines lineales carecen de suavidad. Las splines cúbicas están destinadas a corregir esta deficiencia de las splines lineales. Una spline cúbica reemplaza las funciones lineales entre los puntos de datos con polinomios de tercer grado (cúbicos).

En la figura 3.12(b) se muestra un ejemplo de una spline cúbica que interpola los mismos puntos (1, 2), (2, 1), (4, 4) y (5, 3). Las ecuaciones que definen la spline son

$$S_1(x) = 2 - \frac{13}{8}(x - 1) + 0(x - 1)^2 + \frac{5}{8}(x - 1)^3 \text{ en } [1, 2]$$
$$S_2(x) = 1 + \frac{1}{4}(x - 2) + \frac{15}{8}(x - 2)^2 - \frac{5}{8}(x - 2)^3 \text{ en } [2, 4]$$
$$S_3(x) = 4 + \frac{1}{4}(x - 4) - \frac{15}{8}(x - 4)^2 + \frac{5}{8}(x - 4)^3 \text{ en } [4, 5]. \tag{3.16}$$

Observe en particular la transición suave de una S_i a la siguiente en los puntos básicos, o "nudos", $x = 2$ y $x = 4$. Esto se consigue al disponer las piezas vecinas de la spline, S_i y S_{i+1}, de modo que tengan el valor de sus derivadas, primera y segunda igual a cero cuando son evaluadas en los nudos. La manera de hacer esto es el tema de la siguiente sección.

Dadas las n parejas de puntos $(x_1, y_1), ..., (x_n, y_n)$, es evidente que existe una y sólo una spline lineal a través de cada dos puntos. Esto no será cierto para las splines cúbicas. Se verá que hay muchas de manera infinita a través de cualquier conjunto de puntos. Cuando sea necesario, se añadirán condiciones adicionales para determinar con precisión una spline particular de interés.

3.4.1 Propiedades de las splines

Para tener un poco más de precisión acerca de las propiedades de una spline cúbica, se hace la siguiente definición: suponga que se dan las n parejas de puntos $(x_1, y_1), \ldots, (x_n, y_n)$, donde las x_i son distintas y están en orden creciente. Una **spline cúbica** $S(x)$ a través de los puntos $(x_1, y_1), \ldots, (x_n, y_n)$ es un conjunto de polinomios cúbicos

$$\begin{aligned} S_1(x) &= y_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3 \text{ en } [x_1, x_2] \\ S_2(x) &= y_2 + b_2(x - x_2) + c_2(x - x_2)^2 + d_2(x - x_2)^3 \text{ en } [x_2, x_3] \\ &\vdots \\ S_{n-1}(x) &= y_{n-1} + b_{n-1}(x - x_{n-1}) + c_{n-1}(x - x_{n-1})^2 + d_{n-1}(x - x_{n-1})^3 \text{ en } [x_{n-1}, x_n] \end{aligned} \tag{3.17}$$

con las siguientes propiedades:

Propiedad 1 $S_i(x_i) = y_i$ y $S_i(x_{i+1}) = y_{i+1}$ para $i = 1, \ldots, n - 1$.

Propiedad 2 $S'_{i-1}(x_i) = S'_i(x_i)$ para $i = 2, \ldots, n - 1$.

Propiedad 3 $S''_{i-1}(x_i) = S''_i(x_i)$ para $i = 2, \ldots, n - 1$.

La propiedad 1 garantiza que la spline $S(x)$ interpole los puntos de datos. La propiedad 2 obliga a las pendientes de las partes adyacentes de la spline a concordar en los puntos donde se encuentran, y la propiedad 3 hace lo mismo con la curvatura, representada por la segunda derivada.

► EJEMPLO 3.13 Verifique que $\{S_1, S_2, S_3\}$ en (3.16) satisfaga todas las propiedades de las splines cúbicas para los puntos (1, 2), (2, 1), (4, 4) y (5, 3).

Se verificarán las tres propiedades.

Propiedad 1. Hay $n = 4$ puntos. Se debe comprobar que

$$\begin{aligned} S_1(1) &= 2 \quad \text{y} \quad S_1(2) = 1 \\ S_2(2) &= 1 \quad \text{y} \quad S_2(4) = 4 \\ S_3(4) &= 4 \quad \text{y} \quad S_3(5) = 3. \end{aligned}$$

Esto se deduce con facilidad de las ecuaciones definidas en (3.16).

Propiedad 2. Las primeras derivadas de las funciones splines son

$$\begin{aligned} S_1'(x) &= -\frac{13}{8} + \frac{15}{8}(x-1)^2 \\ S_2'(x) &= \frac{1}{4} + \frac{15}{4}(x-2) - \frac{15}{8}(x-2)^2 \\ S_3'(x) &= \frac{1}{4} - \frac{15}{4}(x-4) + \frac{15}{8}(x-4)^2. \end{aligned}$$

Se debe verificar que $S_1'(2) = S_2'(2)$ y $S_2'(4) = S_3'(4)$. La primera igualdad es

$$-\frac{13}{8} + \frac{15}{8} = \frac{1}{4},$$

y la segunda es

$$\frac{1}{4} + \frac{15}{4}(4-2) - \frac{15}{8}(4-2)^2 = \frac{1}{4},$$

por lo que ambas cumplen.

Propiedad 3. Las segundas derivadas son

$$\begin{aligned} S_1''(x) &= \frac{15}{4}(x-1) \\ S_2''(x) &= \frac{15}{4} - \frac{15}{4}(x-2) \\ S_3''(x) &= -\frac{15}{4} + \frac{15}{4}(x-4). \end{aligned} \tag{3.18}$$

Se debe verificar que $S_1''(2) = S_2''(2)$ y $S_2''(4) = S_3''(4)$, donde ambas igualdades son ciertas. Por lo tanto, (3.16) es una spline cúbica. ◄

La construcción de una spline a partir de un conjunto de puntos de datos significa encontrar los coeficientes b_i, c_i, d_i que hacen que se cumplan las propiedades 1, 2 y 3. Antes de analizar cómo se determinan los coeficientes desconocidos b_i, c_i, d_i de la spline, se contará el número de condiciones impuestas por la definición. La primera parte de la propiedad 1 ya está reflejada en la forma (3.17); dice que el término constante de la S_i cúbica debe ser y_i. La segunda parte de la propiedad 1 consiste de $n - 1$ ecuaciones independientes que deben ser satisfechas por los coeficientes, que se consideran como incógnitas. Cada una de las propiedades 2 y 3 añade $n - 2$ ecuaciones, para un total de $n - 1 + 2(n - 2) = 3n - 5$ ecuaciones independientes que deben satisfacerse.

¿Cuántos coeficientes desconocidos hay? Para cada parte S_i de la spline, se requieren tres coeficientes b_i, c_i, d_i, lo que resulta en un total de $3(n-1) = 3n-3$. Por lo tanto, la determinación de los coeficientes consiste en resolver un sistema de $3n-5$ ecuaciones lineales con $3n-3$ incógnitas. A menos que haya ecuaciones inconsistentes en el sistema (y no las hay), el sistema de ecuaciones es indeterminado y por lo tanto tiene un número infinito de soluciones. En otras palabras, existe una infinidad de splines cúbicas que pasan por el conjunto arbitrario de puntos de datos $(x_1, y_1), \ldots, (x_n, y_n)$.

Por lo general, los usuarios de splines para volver el sistema determinado agregan dos ecuaciones al sistema para tener $3n-5$ ecuaciones, para llegar a un sistema de m ecuaciones con m incógnitas, donde $m = 3n-3$. Además de permitir al usuario restringir la spline con las especificaciones dadas, la reducción del campo a una solución única simplifica el cálculo y la descripción del resultado.

La forma más sencilla de añadir dos restricciones más consiste en permitir que la spline $S(x)$ tenga un punto de inflexión en cada extremo del intervalo de definición $[x_1, x_n]$, además de las $3n-5$ anteriores. Las restricciones añadidas a las propiedades 1, 2 y 3 son

Propiedad 4a **Spline natural.** $S_1''(x_1) = 0$ y $S_{n-1}''(x_n) = 0$.

Una spline cúbica que satisface estas dos condiciones adicionales se llama spline cúbica natural. Observe que (3.16) es una spline cúbica natural, puesto que puede verificarse con facilidad a partir de (3.18) que $S_1''(1) = 0$ y $S_3''(5) = 0$.

Hay varias maneras distintas de agregar dos condiciones adicionales. Por lo general, como en el caso de la spline natural, determinan las propiedades adicionales en los extremos izquierdo y derecho de la spline, por lo que se denominan condiciones de extremo. Este tema se abordará en la siguiente sección, pero por ahora el estudio se concentrará en las splines cúbicas naturales.

Ahora que se tiene el sistema de ecuaciones, $3n-3$, con $3n-3$ incógnitas, es posible escribir una función de MATLAB para resolver los coeficientes de la spline. En primer lugar, escriba las ecuaciones de las incógnitas b_i, c_i, d_i. Entonces, la parte 2 de la propiedad 1 implica las $n-1$ ecuaciones:

$$\begin{aligned} y_2 &= S_1(x_2) = y_1 + b_1(x_2 - x_1) + c_1(x_2 - x_1)^2 + d_1(x_2 - x_1)^3 \\ &\vdots \\ y_n &= S_{n-1}(x_n) = y_{n-1} + b_{n-1}(x_n - x_{n-1}) + c_{n-1}(x_n - x_{n-1})^2 \\ &\qquad + d_{n-1}(x_n - x_{n-1})^3. \end{aligned} \tag{3.19}$$

La propiedad 2 genera las $n-2$ ecuaciones,

$$\begin{aligned} 0 &= S_1'(x_2) - S_2'(x_2) = b_1 + 2c_1(x_2 - x_1) + 3d_1(x_2 - x_1)^2 - b_2 \\ &\vdots \\ 0 &= S_{n-2}'(x_{n-1}) - S_{n-1}'(x_{n-1}) = b_{n-2} + 2c_{n-2}(x_{n-1} - x_{n-2}) \\ &\quad + 3d_{n-2}(x_{n-1} - x_{n-2})^2 - b_{n-1}, \end{aligned} \tag{3.20}$$

y la propiedad 3 implica las $n-2$ ecuaciones:

$$\begin{aligned} 0 &= S_1''(x_2) - S_2''(x_2) = 2c_1 + 6d_1(x_2 - x_1) - 2c_2 \\ &\vdots \\ 0 &= S_{n-2}''(x_{n-1}) - S_{n-1}''(x_{n-1}) = 2c_{n-2} + 6d_{n-2}(x_{n-1} - x_{n-2}) - 2c_{n-1}. \end{aligned} \tag{3.21}$$

En lugar de resolver las ecuaciones de esta forma, el sistema puede simplificarse. Con un poco de álgebra, puede resolverse primero un sistema de ecuaciones mucho más pequeño en c_i, seguido por fórmulas explícitas para b_i y d_i en términos de la c_i conocida.

Lo anterior resulta conceptualmente más simple si se introduce una variable adicional desconocida $c_n = S''_{n-1}(x_n)/2$. Además, se añade la notación abreviada $\delta_i = x_{i+1} - x_i$ y $\Delta_i = y_{i+1} - y_i$. Entonces (3.21) puede resolverse para los coeficientes

$$d_i = \frac{c_{i+1} - c_i}{3\delta_i} \quad \text{para } i = 1, \ldots, n-1. \tag{3.22}$$

Si se despeja b_i de (3.19) resulta

$$\begin{aligned} b_i &= \frac{\Delta_i}{\delta_i} - c_i\delta_i - d_i\delta_i^2 \\ &= \frac{\Delta_i}{\delta_i} - c_i\delta_i - \frac{\delta_i}{3}(c_{i+1} - c_i) \\ &= \frac{\Delta_i}{\delta_i} - \frac{\delta_i}{3}(2c_i + c_{i+1}) \end{aligned} \tag{3.23}$$

para $i = 1, \ldots, n-1$.

Al sustituir (3.22) y (3.23) en (3.20) se obtienen las siguientes $n-2$ ecuaciones en $c_1, \ldots, c_n$:

$$\delta_1 c_1 + 2(\delta_1 + \delta_2)c_2 + \delta_2 c_3 = 3\left(\frac{\Delta_2}{\delta_2} - \frac{\Delta_1}{\delta_1}\right)$$

$$\vdots$$

$$\delta_{n-2}c_{n-2} + 2(\delta_{n-2} + \delta_{n-1})c_{n-1} + \delta_{n-1}c_n = 3\left(\frac{\Delta_{n-1}}{\delta_{n-1}} - \frac{\Delta_{n-2}}{\delta_{n-2}}\right).$$

Dos ecuaciones más están dadas por las condiciones de las splines naturales (propiedad 4a):

$$S''_1(x_1) = 0 \rightarrow 2c_1 = 0$$
$$S''_{n-1}(x_n) = 0 \rightarrow 2c_n = 0.$$

Esto da un total de n ecuaciones con n incógnitas c_i, lo cual puede escribirse en la forma matricial

$$\begin{bmatrix} 1 & 0 & 0 & & & \\ \delta_1 & 2\delta_1 + 2\delta_2 & \delta_2 & \ddots & & \\ 0 & \delta_2 & 2\delta_2 + 2\delta_3 & \delta_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & \delta_{n-2} & 2\delta_{n-2} + 2\delta_{n-1} & \delta_{n-1} \\ & & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ \\ \\ \vdots \\ \\ \\ c_n \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 3\left(\frac{\Delta_2}{\delta_2} - \frac{\Delta_1}{\delta_1}\right) \\ \\ \vdots \\ \\ 3\left(\frac{\Delta_{n-1}}{\delta_{n-1}} - \frac{\Delta_{n-2}}{\delta_{n-2}}\right) \\ 0 \end{bmatrix}. \tag{3.24}$$

Después de obtener $c_1, \ldots, c_n$ a partir de (3.24), $b_1, \ldots, b_{n-1}$ y $d_1, \ldots, d_{n-1}$ se determinan con base en (3.22) y (3.23).

Observe que (3.24) siempre es resuelta para c_i. La matriz de coeficientes es de diagonal estrictamente dominante, por lo que debido al teorema 2.10, existe una solución única para las c_i, y por lo tanto también para las b_i y d_i. Así, queda demostrado el siguiente teorema:

TEOREMA 3.7 Sea $n \geq 2$. Para un conjunto de puntos $(x_1, y_1), \ldots, (x_n, y_n)$ con x_i distintas, existe una sola spline cúbica natural que se ajusta a los puntos. ■

Spline cúbica natural

Dada $x = [x_1, \ldots, x_n]$ donde $x_1 < \cdots < x_n$, $y = [y_1, \ldots, y_n]$
for $i = 1, \ldots, n-1$
$a_i = y_i$
$\delta_i = x_{i+1} - x_i$
$\Delta_i = y_{i+1} - y_i$
end
Resuelve (3.24) para $c_1, \ldots, c_n$
for $i = 1, \ldots, n-1$

$$d_i = \frac{c_{i+1} - c_i}{3\delta_i}$$

$$b_i = \frac{\Delta_i}{\delta_i} - \frac{\delta_i}{3}(2c_i + c_{i+1})$$

end
La spline cúbica natural es
$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$ en $[x_i, x_{i+1}]$ para $i = 1, \ldots, n-1$.

► EJEMPLO 3.14 Encuentre la spline cúbica natural que pasa por $(0, 3)$, $(1, -2)$ y $(2, 1)$.

Las coordenadas x son $x_1 = 0, x_2 = 1$, y $x_3 = 2$. Las coordenadas y son $a_1 = y_1 = 3, a_2 = y_2 = -2$, y $a_3 = y_3 = 1$, y las diferencias son $\delta_1 = \delta_2 = 1, \Delta_1 = -5$, y $\Delta_2 = 3$. La ecuación matricial tridiagonal (3.24) es

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 4 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 24 \\ 0 \end{bmatrix}.$$

La solución es $[c_1, c_2, c_3] = [0, 6, 0]$. Ahora, de (3.22) y (3.23) se obtiene

$$d_1 = \frac{c_2 - c_1}{3\delta_1} = \frac{6}{3} = 2$$

$$d_2 = \frac{c_3 - c_2}{3\delta_2} = \frac{-6}{3} = -2$$

$$b_1 = \frac{\Delta_1}{\delta_1} - \frac{\delta_1}{3}(2c_1 + c_2) = -5 - \frac{1}{3}(6) = -7$$

$$b_2 = \frac{\Delta_2}{\delta_2} - \frac{\delta_2}{3}(2c_2 + c_3) = 3 - \frac{1}{3}(12) = -1.$$

Por lo tanto, la spline cúbica es

$$S_1(x) = 3 - 7x + 0x^2 + 2x^3 \text{ en } [0, 1]$$

$$S_2(x) = -2 - 1(x - 1) + 6(x - 1)^2 - 2(x - 1)^3 \text{ en } [1, 2].$$ ◄

A continuación se presenta el programa en MATLAB para este cálculo. Para condiciones diferentes en los extremos (no naturales), que se analizan en la sección siguiente, los renglones superior e inferior de (3.24) se sustituyen por las filas apropiadas.

```
%Programa 3.5 Cálculo de coeficientes de splines
%Calcula los coeficientes de splines cúbicas
%Entrada: vectores x, y de puntos de datos
%    más dos datos extra v1, vn opcionales
%Salida: matriz de coeficientes b1, c1,d1,b2,c2,d2; ...
function coeff=splinecoeff(x,y)
n=length(x);v1=0;vn=0;
A=zeros(n,n);          % la matriz A es de xn
r=zeros(n,1);
for i=1:n-1            % define las deltas
    dx(i)= x(i+1)-x(i); dy(i)=y(i+1)-y(i);
end
for i=2:n-1            % carga la matriz A
    A(i,i-1:i+1)=[dx(i-1) 2*(dx(i-1)+dx(i)) dx(i)];
    r(i)=3*(dy(i)/dx(i)-dy(i-1)/dx(i-1)); % lado derecho
end
% Establece condiciones de extremo
% Use sólo uno de los siguientes 5 pares:
A(1,1) = 1;            % condiciones de spline natural
A(n,n) = 1;
%A(1,1)=2;r(1)=v1;     % condiciones de curvatura ajustada
%A(n,n)=2;r(n)=vn;
%A(1,1:2)=[2*dx(1) dx(1)];r(1)=3*(dy(1)/dx(1)-v1);  %fija
%A(n,n-1:n)=[dx(n-1) 2*dx(n-1)];r(n)=3*(vn-dy(n-1)/dx(n-1));
%A(1,1:2)=[1 -1];      % condiciones de parábola, para n>=3
%A(n,n-1:n)=[1 -1];
%A(1,1:3)=[dx(2) -(dx(1)+dx(2)) dx(1)]; % sin-nudo, para n>=4
%A(n,n-2:n)=[dx(n-1) -(dx(n-2)+dx(n-1)) dx(n-2)];
coeff=zeros(n,3);
coeff(:,2)=A\r;        % resuelve para los coeficientes c
for i=1:n-1            % resuelve para b y d
    coeff(i,3)=(coeff(i+1,2)-coeff(i,2))/(3*dx(i));
    coeff(i,1)=dy(i)/dx(i)-dx(i)*(2*coeff(i,2)+coeff(i+1,2))/3;
end
coeff=coeff(1:n-1,1:3);
```

En este programa se han listado otras opciones para las condiciones de extremo, aunque hasta el momento éstas sólo han sido comentadas. Las condiciones alternativas se analizarán en la sección siguiente. Otra de las funciones de MATLAB, titulada **splineplot.m**, llama a **splinecoeff.m** para obtener los coeficientes y después grafica la spline cúbica:

```
%Programa 3.6 Gráfica de spline cúbica
%Calcula y grafica splines a partir de puntos de datos
%Entrada: vectores x, y de los puntos de datos, el número k de puntos graficados
%    por segmento
%Salida: Valores x1, y1 de la spline en los puntos graficados
function [x1,y1]=splineplot(x,y,k)
n=length(x);
coeff=splinecoeff(x,y);
x1=[]; y1=[];
for i=1:n-1
    xs=linspace(x(i),x(i+1),k+1);
    dx=xs-x(i);
```

```
    ys=coeff(i,3)*dx; %  evalúa usando multiplicación anidada
    ys=(ys+coeff(i,2)).*dx;
    ys=(ys+coeff(i,1)).*dx+y(i);
    xl=[xl; xs(1:k)']; yl=[yl;ys(1:k)'];
end
xl=[xl; x(end)];yl=[yl;y(end)];
plot(x,y,'o',xl,yl)
```

En la figura 3.13(a) se muestra una spline cúbica natural, generada mediante `splineplot.m`.

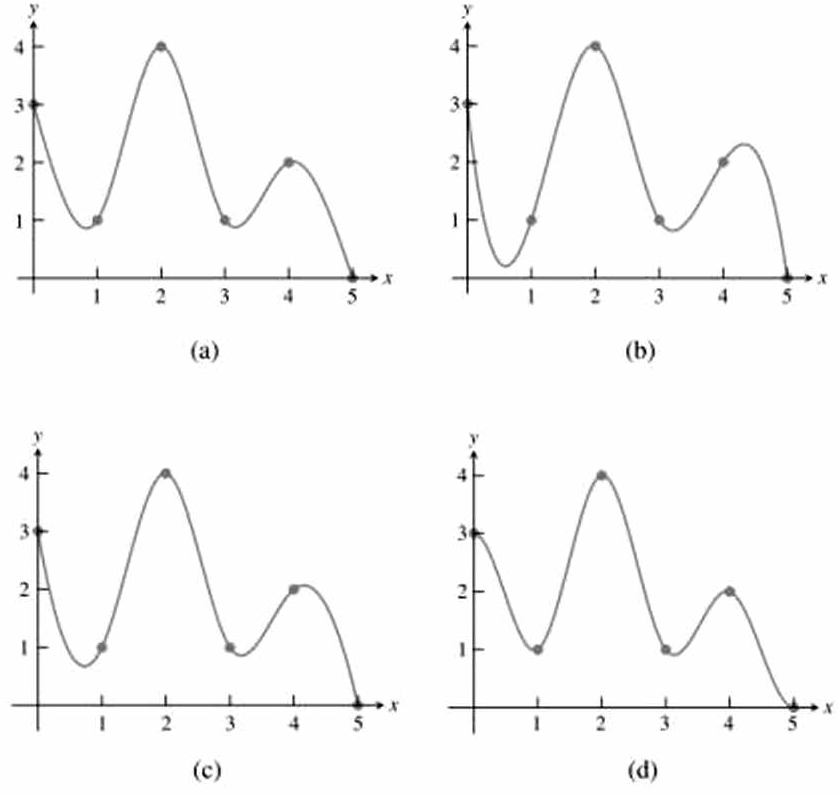


Figura 3.13 Splines cúbicas a través de seis puntos. Las gráficas son generadas por `splineplot(x, y, 10)` con los vectores de entrada `x = [0 1 2 3 4 5]` y `y = [3 1 4 1 2 0]`. (a) Spline cúbica natural (note los puntos de inflexión en los extremos) (b) Spline cúbica sin nudo (ecuación cúbica única en [0, 2] y en [3, 5]) (c) Spline terminada parabólicamente (d) Spline cúbica fija (fija en pendiente 0 en ambos extremos).

3.4.2 Condiciones de extremo

Las dos condiciones adicionales que se especifican en la propiedad 4a se llaman las "condiciones de extremo" para una spline natural. El requisito de que estas condiciones deben satisfacerse junto con las propiedades 1 a 3 reduce el campo justo a una spline cúbica, de acuerdo con el teorema 3.9. Resulta que hay muchas versiones diferentes de la propiedad 4, lo que significa que hay muchos otros pares de condiciones de extremo, para los cuales se cumple un teorema análogo. En esta sección se presentan algunos de los más populares.

Propiedad 4b **Spline cúbica de curvatura ajustada.** La primera alternativa para una spline cúbica natural requiere establecer $S_1''(x_1)$ y $S_{n-1}''(x_n)$ para valores arbitrarios, elegidos por el usuario, en vez de cero.

Esta opción corresponde a establecer las curvaturas deseadas en los extremos de puntos izquierdo y derecho de la spline. En términos de (3.23), esto se traduce como las dos condiciones adicionales

$$2c_1 = v_1$$
$$2c_n = v_n,$$

donde v_1, v_n indica los valores deseados. Las ecuaciones se convierten en los dos renglones de la tabla

$$\left[\begin{array}{cccccccccc|c} 2 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & | & v_1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 2 & | & v_n \end{array}\right]$$

para reemplazar los renglones superior e inferior de (3.24), que se añadieron para la spline natural. Observe que la nueva matriz de coeficientes es otra vez de diagonal estrictamente dominante, de manera que es una forma generalizada del teorema 3.9 que se cumple para las splines de curvatura ajustada (vea el teorema 3.10 que se presenta en breve). En `splinecoeff.m`, las dos líneas

```
A(1,1)=2;r(1)=v1;          %  condiciones de curvatura ajustada
A(n,n)=2;r(n)=vn;
```

deben sustituirse en lugar de los dos renglones existentes para la spline natural.

El siguiente conjunto alternativo de condiciones de extremo son

Propiedad 4c **Spline cúbica fija.** Esta alternativa es similar a la anterior, pero aquí son las primeras derivadas $S_1'(x_1)$ y $S_{n-1}'(x_n)$ las que se establecen a los valores respectivos de v_1 y v_n especificados por el usuario. Por lo tanto, la pendiente al comienzo y al final de la spline está bajo el control de los usuarios.

Usando (3.22) y (3.23), la condición adicional $S_1'(x_1) = v_1$ puede escribirse como

$$2\delta_1 c_1 + \delta_1 c_2 = 3\left(\frac{\Delta_1}{\delta_1} - v_1\right)$$

y $S_{n-1}'(x_n) = v_n$ como

$$\delta_{n-1}c_{n-1} + 2\delta_{n-1}c_n = 3\left(v_n - \frac{\Delta_{n-1}}{\delta_{n-1}}\right).$$

Los dos renglones correspondientes de la tabla son

$$\left[\begin{array}{cccccccccc|c} 2\delta_1 & \delta_1 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & | & 3(\Delta_1/\delta_1 - v_1) \\ 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & \delta_{n-1} & 2\delta_{n-1} & | & 3(v_n - \Delta_{n-1}/\delta_{n-1}) \end{array}\right].$$

Observe que la dominancia de la diagonal se aplica también a la matriz de coeficientes revisada en (3.24), por lo que el teorema 3.9 también se cumple con la spline natural reemplazada por la spline fija. En `splinecoeff.m`, deben sustituirse los dos renglones

```
A(1,1:2)=[2*dx(1)  dx(1)];r(1)=3*(dy(1)/dx(1)-v1);
A(n,n-1:n)=[dx(n-1)  2*dx(n-1)];r(n)=3*(vn-dy(n-1)/dx(n-1));
```

vea en la figura 3.13 una spline fija con $v_1 = v_n = 0$.

Propiedad 4d **Spline cúbica terminada parabólicamente.** La primera y la última partes de la spline, S_1 y S_{n-1}, se obligan a tener un grado máximo de 2, al especificar que $d_1 = 0 = d_{n-1}$. En forma equivalente, de acuerdo con (3.22), se requiere que $c_1 = c_2$ y $c_{n-1} = c_n$. Las ecuaciones forman los dos renglones de la tabla

$$\left[\begin{array}{cccccccccc|c} 1 & -1 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1 & -1 & | & 0 \end{array}\right]$$

que se utilizan como los renglones superior e inferior de (3.24). Suponga que el número n de puntos de datos satisface $n \geq 3$ (vea el ejercicio 19 para el caso $n = 2$). En este caso, al sustituir c_1 por c_2 y c_n por c_{n-1}, sucede que la ecuación se reduce a una ecuación matricial de diagonal estrictamente dominante de $n-2 \times n-2$ en $c_2, \ldots, c_{n-1}$. Por lo tanto, una versión del teorema 3.9 se cumple para las splines terminadas parabólicamente, suponiendo que $n \geq 3$.

En `splinecoeff.m` deben sustituirse los dos renglones

```
A(1,1:2)=[1 -1];                    % condiciones de terminado parabólico
A(n,n-1:n)=[1 -1];
```

debe ser sustituido.

Propiedad 4e **Spline cúbica sin nodo.** Las dos ecuaciones añadidas son $d_1 = d_2$ y $d_{n-2} = d_{n-1}$ o de manera equivalente $S_1'''(x_2) = S_2'''(x_2)$ y $S_{n-2}'''(x_{n-1}) = S_{n-1}'''(x_{n-1})$. Como S_1 y S_2 son polinomios de tercer grado o menor, se requiere que sus terceras derivadas concuerden en x_2, mientras que sus derivadas de orden cero, primero y segundo ya concuerdan ahí, causan que S_1 y S_2 sean polinomios cúbicos idénticos (los polinomios cúbicos están definidos por cuatro puntos y se especifican cuatro condiciones). Así, x_2 no se requiere como punto básico: la spline está dada por la misma fórmula $S_1 = S_2$ en todo el intervalo $[x_1, x_3]$. El mismo razonamiento muestra que $S_{n-2} = S_{n-1}$, por lo que no sólo x_2, sino también x_{n-1}, "ya no es un nudo".

Observe que $d_1 = d_2$ implica que $(c_2 - c_1)/\delta_1 = (c_3 - c_2)/\delta_2$, o

$$\delta_2 c_1 - (\delta_1 + \delta_2)c_2 + \delta_1 c_3 = 0,$$

y de manera similar, $d_{n-2} = d_{n-1}$ implica que

$$\delta_{n-1} c_{n-2} - (\delta_{n-2} + \delta_{n-1})c_{n-1} + \delta_{n-2} c_n = 0.$$

De esto se deduce que los dos renglones de la tabla son

$$\left(\begin{array}{cccccccccc|c} \delta_2 & -(\delta_1+\delta_2) & \delta_1 & 0 & \cdots & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & \delta_{n-1} & -(\delta_{n-2}+\delta_{n-1}) & \delta_{n-2} & 0 \end{array}\right).$$

En `splinecoeff.m` se utilizan los dos renglones

```
A(1,1:3)=[dx(2) -(dx(1)+dx(2)) dx(1)]; % Condiciones sin nudo
A(n,n-2:n)=[dx(n-1) -(dx(n-2)+dx(n-1)) dx(n-2)];
```

En la figura 3.13(b) se muestra un ejemplo de una spline cúbica sin nudo, comparada con la spline natural a través de los mismos puntos de datos del inciso (a) de la figura.

Como se mencionó anteriormente, existe un teorema análogo al teorema 3.7 para cada una de las opciones anteriores de condiciones finales:

TEOREMA 3.8 Suponga que $n \geq 2$. Entonces, para un conjunto de puntos $(x_1, y_1), \ldots, (x_n, y_n)$ y para cualquiera de las condiciones finales dadas por las propiedades 4a a 4c, existe una spline cúbica única que satisface las condiciones finales y se ajusta a los puntos. Lo mismo es cierto si se supone que $n \geq 3$ para la propiedad 4d y $n \geq 4$ para la propiedad 4e. ■

El comando predeterminado `spline` de MATLAB construye una spline sin nudo cuando se le proporcionan cuatro o más puntos. Sean `x` y `y` vectores que contienen los datos x_i y y_i, respectivamente. Entonces, la coordenada y de la spline sin nudo en otra entrada x^0 se calcula mediante el comando de MATLAB

```
>> y0 = spline(x,y,x0);
```

Si `x0` es el vector de coordenadas x, y `y0` es el vector correspondiente de coordenadas y, adecuado para su graficación, etcétera. De manera alternativa, si el vector de entrada `y` tiene exactamente dos puntos más que `x`, se calcula la spline cúbica fija, con restricciones v_1 y v_n iguales a la primera y última entradas de `y`.

3.4 Ejercicios

1. Verifique si las ecuaciones forman un spline cúbica.

$$\text{(a)}\quad S(x)=\begin{cases}x^3+x-1 & \text{en } [0,1]\\ -(x-1)^3+3(x-1)^2+3(x-1)+1 & \text{en } [1,2]\end{cases}$$

$$\text{(b)}\quad S(x)=\begin{cases}2x^3+x^2+4x+5 & \text{en } [0,1]\\ (x-1)^3+7(x-1)^2+12(x-1)+12 & \text{en } [1,2]\end{cases}$$

2. (a) Verifique las condiciones de spline para

$$\begin{cases}S_1(x)=1+2x+3x^2+4x^3 & \text{en } [0,1]\\ S_2(x)=10+20(x-1)+15(x-1)^2+4(x-1)^3 & \text{en } [1,2]\end{cases}.$$

(b) Independientemente de su respuesta en (a), decida si alguna de las siguientes condiciones adicionales se satisfacen para este ejemplo: natural, terminada parabólicamente, sin nudo.

3. Encuentre c en las siguientes splines cúbicas. ¿Cuál de las tres condiciones de extremo (natural, terminada parabólicamente, sin nudo) en su caso, se satisface?

$$\text{(a)}\quad S(x)=\begin{cases}4-\frac{11}{4}x+\frac{3}{4}x^3 & \text{en } [0,1]\\ 2-\frac{1}{2}(x-1)+c(x-1)^2-\frac{3}{4}(x-1)^3 & \text{en } [1,2]\end{cases}$$

$$\text{(b)}\quad S(x)=\begin{cases}3-9x+4x^2 & \text{en } [0,1]\\ -2-(x-1)+c(x-1)^2 & \text{en } [1,2]\end{cases}$$

$$\text{(c)}\quad S(x)=\begin{cases}-2-\frac{3}{2}x+\frac{7}{2}x^2-x^3 & \text{en } [0,1]\\ -1+c(x-1)+\frac{1}{2}(x-1)^2-(x-1)^3 & \text{en } [1,2]\\ 1+\frac{1}{2}(x-2)-\frac{5}{2}(x-2)^2-(x-2)^3 & \text{en } [2,3]\end{cases}$$

4. Encuentre k_1, k_2, k_3 en la siguiente spline cúbica. ¿Cuál de las tres condiciones de extremo (natural, terminada parabólicamente o sin nudo) en su caso, se satisface?

$$S(x)=\begin{cases}4+k_1x+2x^2-\frac{1}{6}x^3 & \text{en } [0,1]\\ 1-\frac{4}{3}(x-1)+k_2(x-1)^2-\frac{1}{6}(x-1)^3 & \text{en } [1,2].\\ 1+k_3(x-2)+(x-2)^2-\frac{1}{6}(x-2)^3 & \text{en } [2,3]\end{cases}$$

5. ¿Cuántas splines cúbicas naturales existen en [0, 2] para los puntos dados (0, 0), (1, 1), (2, 2)? Muestre una spline de este tipo.

6. Encuentre la spline cúbica terminada parabólicamente que pasa por los puntos de datos (0, 1), (1, 1), (2, 1), (3, 1), (4, 1). ¿Esta spline también es sin nudo?, ¿natural?

7. Resuelva las ecuaciones (3.24) para encontrar la spline cúbica natural que pasa por los tres puntos (a) (0, 0), (1, 1), (2, 4) (b) (−1, 1), (1, 1), (2, 4).

8. Resuelva las ecuaciones (3.24) para encontrar la spline cúbica natural que pasa por los tres puntos (a) (0, 1), (2, 3), (3, 2) (b) (0, 0), (1, 1), (2, 6).

9. Encuentre $S'(0)$ y $S'(3)$ para la spline cúbica

$$\begin{cases} S_1(x) = 3 + b_1 x + x^3 & \text{en } [0, 1] \\ S_2(x) = 1 + b_2(x-1) + 3(x-1)^2 - 2(x-1)^3 & \text{en } [1, 3] \end{cases}$$

10. Verdadero o falso: dados $n = 3$ puntos de datos, la spline cúbica terminada parabólicamente que pasa por los puntos debe ser sin nudo.

11. (a) ¿Cuántas splines cúbicas terminadas parabólicamente existen en [0, 2] para los datos dados (0, 2), (1, 0), (2, 2)? Muestre una spline de este tipo. (b) Responda a la misma pregunta para una spline sin nudo.

12. ¿Cuántas splines cúbicas sin nudo existen para los datos dados (1, 3), (3, 3), (4, 2), (5, 0)? Muestre una spline de este tipo.

13. (a) Encuentre b_1 y c_3 en la spline cúbica

$$S(x) = \begin{cases} -1 + b_1 x - \frac{5}{9}x^2 + \frac{5}{9}x^3 & \text{en } [0, 1] \\ \frac{14}{9}(x-1) + \frac{10}{9}(x-1)^2 - \frac{2}{3}(x-1)^3 & \text{en } [1, 2] \\ 2 + \frac{16}{9}(x-2) + c_3(x-2)^2 - \frac{1}{9}(x-2)^3 & \text{en } [2, 3] \end{cases}$$

(b) ¿Esta spline es natural? (c) Esta spline satisface condiciones "fijas" en los extremos. ¿Cuáles son los valores de las dos restricciones?

14. Considere la spline cúbica

$$\begin{cases} S_1(x) = 6 - 2x + \frac{1}{2}x^3 & \text{en } [0, 2] \\ S_2(x) = 6 + 4(x-2) + c(x-2)^2 + d(x-2)^3 & \text{en } [2, 3] \end{cases}$$

(a) Encuentre c. (b) ¿Existe un grado d de tal forma que la spline sea natural? Si es así, encuentre d.

15. ¿Puede una spline ser cúbica natural y a la vez ser terminada parabólicamente? Si es así, ¿qué más se puede decir de una spline de ese tipo?

16. ¿Existe simultáneamente una spline cúbica natural, terminada parabólicamente y sin nudo que pase por cada conjunto de puntos $(x_1, y_1), \ldots, (x_{100}, y_{100})$ con x_i distintas? Si es así, dé una razón. Si no es así, explique qué condiciones deben cumplirse en los 100 puntos para que exista una spline como ésta.

17. Suponga que la pieza del extremo izquierdo de una spline cúbica natural determinada es la función constante $S_1(x) = 1$ en el intervalo $[-1, 0]$. Encuentre tres posibilidades diferentes para la pieza vecina $S_2(x)$ de la spline en [0, 1].

18. Suponga que un automóvil viaja a lo largo de una carretera recta de un punto a otro a partir de un punto inicial en el tiempo $t = 0$, hasta un punto final en el tiempo $t = 1$. La distancia a lo largo de la carretera se muestrea en tiempos determinados entre 0 y 1. ¿Qué spline cúbica (en términos de condiciones de extremo) será más adecuada para describir la distancia en función del tiempo?

19. El caso $n = 2$ para splines cúbicas terminadas parabólicamente no está cubierto por el teorema 3.8. Analice la existencia y la unicidad de la spline cúbica en este caso.

20. Analice la existencia y la unicidad de una spline cúbica sin nudo cuando $n = 2$ y $n = 3$.

21. El teorema 3.8 dice que hay exactamente una spline sin nudo que pasa por cuatro puntos determinados con distintas x_i. (a) ¿Cuántas splines sin nudo pasan por cualesquiera 3 puntos dados con x_i distintas? (b) Encuentre una spline sin nudo que pase por (0, 0), (1, 1), (2, 4), que no sea terminada parabólicamente.

3.4 Problemas de computadora

1. Encuentre las ecuaciones y grafique la spline cúbica natural que interpola los puntos (a) $(0, 3)$, $(1, 5)$, $(2, 4)$, $(3, 1)$ (b) $(-1, 3)$, $(0, 5)$, $(3, 1)$, $(4, 1)$, $(5, 1)$.

2. Encuentre y grafique la spline cúbica sin nudo que interpola los puntos de datos (a) $(0, 3)$, $(1, 5)$, $(2, 4)$, $(3, 1)$ (b) $(-1, 3)$, $(0, 5)$, $(3, 1)$, $(4, 1)$, $(5, 1)$.

3. Encuentre y grafique la spline cúbica S que satisface $S(0) = 1, S(1) = 3, S(2) = 3, S(3) = 4$, $S(4) = 2$ con $S'(0) = S''(4) = 0$.

4. Encuentre y grafique la spline cúbica S que satisface $S(0) = 1, S(1) = 3, S(2) = 3, S(3) = 4$, $S(4) = 2$ con $S''(0) = 3$ y $S''(4) = 2$.

5. Encuentre y grafique la spline cúbica S que satisface $S(0) = 1, S(1) = 3, S(2) = 3, S(3) = 4$, $S(4) = 2$ con $S'(0) = 0$ y $S'(4) = 1$.

6. Encuentre y grafique la spline cúbica S que satisface $S(0) = 1, S(1) = 3, S(2) = 3, S(3) = 4$, $S(4) = 2$ con $S'(0) = -2$ y $S'(4) = 1$.

7. Encuentre la spline cúbica fija que interpola $f(x) = \cos x$ en cinco puntos espaciados uniformemente en $[0, \pi/2]$, incluyendo los puntos extremos. ¿Cuál es la mejor opción para $S'(0)$ y $S'(\pi/2)$ a fin de disminuir al mínimo el error de interpolación? Grafique la spline y el $\cos x$ en $[0, 2]$.

8. Realice los pasos del problema de computadora 7 para la función $f(x) = \operatorname{sen} x$.

9. Encuentre la spline cúbica fija que interpola $f(x) = \ln x$ en cinco puntos espaciados uniformemente dentro de $[1, 3]$, incluyendo los puntos extremos. Encuentre de manera empírica el error de interpolación máximo en $[1, 3]$.

10. Encuentre el número necesario de nodos de interpolación en el problema de computadora 9 para que el máximo error de interpolación sea cuando mucho de 0.5×10^{-7}.

11. (a) Considere la spline cúbica natural que pasa por los puntos de datos de la población mundial en el problema de computadora 3.1.1. Evalúe el año 1980 y compárelo con la población correcta. (b) Use una spline lineal y calcule para 1960 y 2000, después emplee estos valores para determinar la spline cúbica fija que pasa por los datos. Grafique la spline y estime la población de 1980. ¿Cuál opción estima mejor, la natural o la fija?

12. Con los datos del dióxido de carbono del ejercicio 3.1.17. (a) Encuentre y grafique la spline cúbica natural que pase a través de los datos y calcule la estimación de la spline para la concentración de CO_2 en 1950. (b) Realice el mismo análisis para la spline terminada parabólicamente. (c) ¿En qué se diferencia la spline sin nudo de la solución al ejercicio 3.1.17?

13. En una sola gráfica, muestre la forma de las splines cúbicas natural, sin nudo y la terminada parabólicamente que pasan por los datos de la producción mundial de petróleo del problema de computadora 3.2.3.

14. Recabe una lista de 101 precios diarios consecutivos al cierre de una acción negociada en la bolsa de valores, tomada de algún sitio web de información financiera. (a) Grafique el polinomio de interpolación que pase a través de cada cinco puntos. Es decir, sean `x0=0:5:100` y `y0` indique los precios de las acciones en los días 0, 5, 10, ..., 100. Grafique el polinomio de interpolación de grado 20 en los puntos `x=0:1:100` y compárelo con los datos de los precios diarios. ¿Cuál es el máximo error de interpolación?, ¿el fenómeno de Runge se presenta en su gráfica? (b) Grafique la spline cúbica natural con los nodos de interpolación `0:5:100` en vez del polinomio de interpolación, junto con los datos diarios. Responda a las mismas dos preguntas. (c) Compare los dos métodos de representación de los datos.

15. Recabe una lista de 121 temperaturas horarias durante cinco días consecutivos a partir de un sitio web que reporte datos del clima. Sean `x0=0:6:120` las horas y `y0` las temperaturas en las horas 0, 6, 12, ... , 120. Realice los pasos (a)–(c) del problema de computadora 14, adaptado en forma adecuada.

3.5 CURVAS DE BÉZIER

Las curvas de Bézier son splines que permiten al usuario controlar las pendientes en los nudos. A cambio de la libertad adicional, la suavidad de la primera y segunda derivadas a través del nudo, que es la característica automática de las splines cúbicas presentadas en la sección anterior, ya no está garantizada. Las splines de Bézier son adecuadas para los casos donde las esquinas (primeras derivadas discontinuas) y los cambios bruscos de curvatura (segundas derivadas discontinuas).

Pierre Bézier desarrolló la idea durante su trabajo para la empresa de automóviles Renault. La misma idea fue descubierta en forma independiente por Paul de Casteljau, quien trabajaba para Citroen, una empresa de automóviles rival. Se consideró un secreto industrial por ambas compañías, y el hecho de que los dos investigadores habían desarrollado la misma idea surgió a la luz sólo después de que Bézier publicara su trabajo. Hoy en día, la curva de Bézier es una piedra angular en el diseño y la manufactura asistida por computadora.

Cada pieza de una spline de Bézier plana está determinada por cuatro puntos (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4). El primero y el último de los puntos son puntos extremos de la curva spline, y los dos puntos de en medio son **puntos de control,** como se muestra en la figura 3,14. La curva inicia en (x_1, y_1), avanza a lo largo de la dirección de la tangente $(x_2 - x_1, y_2 - y_1)$ y termina en (x_4, y_4) a lo largo de la dirección de la tangente $(x_4 - x_3, y_4 - y_3)$. Las ecuaciones que logran esto se expresan como una curva paramétrica $(x(t), y(t))$ para $0 \le t \le 1$.

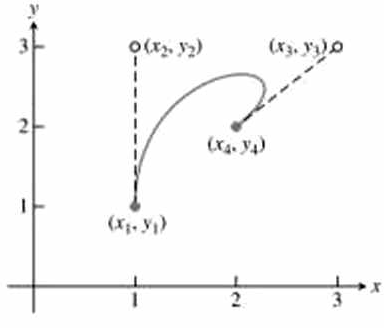


Figura 3.14 Curva de Bézier del ejemplo 3.15. Los puntos (x_1, y_1) y (x_4, y_4) son puntos de inflexión (cambio de pendiente), mientras que (x_2, y_2) y (x_3, y_3) son puntos de control.

Curva de Bézier

Dados los puntos extremos (x_1, y_1), (x_4, y_4)
los puntos de control (x_2, y_2), (x_3, y_3)

Establezca

$$
\begin{aligned}
b_x &= 3(x_2 - x_1) \\
c_x &= 3(x_3 - x_2) - b_x \\
d_x &= x_4 - x_1 - b_x - c_x \\
b_y &= 3(y_2 - y_1) \\
c_y &= 3(y_3 - y_2) - b_y \\
d_y &= y_4 - y_1 - b_y - c_y.
\end{aligned}
$$

La curva de Bézier se define para $0 \le t \le 1$ mediante

$$x(t) = x_1 + b_x t + c_x t^2 + d_x t^3$$
$$y(t) = y_1 + b_y t + c_y t^2 + d_y t^3.$$

Es fácil comprobar las afirmaciones del párrafo anterior a partir de las ecuaciones. De hecho, de acuerdo con el ejercicio 11,

$$\begin{aligned} x(0) &= x_1 \\ x'(0) &= 3(x_2 - x_1) \\ x(1) &= x_4 \\ x'(1) &= 3(x_4 - x_3), \end{aligned} \tag{3.25}$$

y en forma análoga se cumplen para $y(t)$.

► EJEMPLO 3.15 Encuentre la curva de Bézier $(x(t), y(t))$ que pasa a través de los puntos $(x, y) = (1, 1)$ y $(2, 2)$ con los puntos de control $(1, 3)$ y $(3, 3)$.

Los cuatro puntos son $(x_1, y_1) = (1, 1)$, $(x_2, y_2) = (1, 3)$, $(x_3, y_3) = (3, 3)$, y $(x_4, y_4) = (2, 2)$. De las fórmulas de Bézier se obtiene $b_x = 0$, $c_x = 6$, $d_x = -5$ y $b_y = 6$, $c_y = -6$, $d_y = 1$. La spline de Bézier

$$x(t) = 1 + 6t^2 - 5t^3$$
$$y(t) = 1 + 6t - 6t^2 + t^3$$

se muestra en la figura 3.14 junto con los puntos de control. ◄

Las curvas de Bézier son bloques de construcción que pueden almacenarse para adaptarse a valores y pendientes arbitrarios de las funciones. Se trata de una mejora con respecto a las splines cúbicas, en el sentido de que las pendientes en los nodos pueden especificarse según lo desee el usuario. Sin embargo, esta libertad se da a expensas de la suavidad: por lo general, las segundas derivadas de las dos direcciones diferentes no concuerdan en los nodos. En algunas aplicaciones, esta discrepancia es una ventaja.

Como caso especial, cuando los puntos de control son iguales a los puntos extremos, la spline es un segmento de línea sencillo, como se muestra a continuación.

► EJEMPLO 3.16 Demuestre que la spline de Bézier con $(x_1, y_1) = (x_2, y_2)$ y $(x_3, y_3) = (x_4, y_4)$ es un segmento de línea recta.

Las fórmulas de Bézier muestran que las ecuaciones son

$$x(t) = x_1 + 3(x_4 - x_1)t^2 - 2(x_4 - x_1)t^3 = x_1 + (x_4 - x_1)t^2(3 - 2t)$$
$$y(t) = y_1 + 3(y_4 - y_1)t^2 - 2(y_4 - y_1)t^3 = y_1 + (y_4 - y_1)t^2(3 - 2t)$$

para $0 \le t \le 1$. Cada punto en la spline tiene la forma

$$\begin{aligned} (x(t), y(t)) &= (x_1 + r(x_4 - x_1), y_1 + r(y_4 - y_1)) \\ &= ((1 - r)x_1 + rx_4, (1 - r)y_1 + ry_4), \end{aligned}$$

donde $r = t^2(3 - 2t)$. En el intervalo $0 \le r \le 1$, cada punto se encuentra en el segmento de línea que une a (x_1, y_1) con (x_4, y_4).

Las curvas de Bézier son fáciles de programar y se utilizan con frecuencia en la elaboración de software. Una curva a mano alzada en el plano puede verse como una curva paramétrica $(x(t), y(t))$ y representarse mediante una spline de Bézier. Las ecuaciones se aplican en el siguiente programa de dibujo a mano alzada de MATLAB. El usuario hace clic en el ratón una vez para fijar

un punto de inicio (x_0, y_0) en el plano, y tres clics más para marcar el primer punto de control, el segundo punto de control y el punto final. Una spline de Bézier se dibuja entre el inicio y el punto final. Cada tres clics posteriores del ratón extienden aún más la curva, utilizando el punto final anterior como punto de inicio para la siguiente pieza. El comando `ginput` de MATLAB se utiliza para leer la ubicación del ratón. En la figura 3.15 se muestra una captura de pantalla de `bezierdraw.m`.

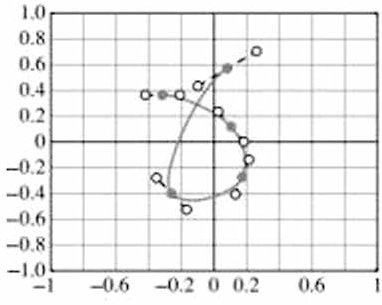


Figura 3.15 Programa 3.7 construido a partir de curvas de Bézier. Captura de pantalla del código `bezierdraw.m`, de MATLAB se incluyen los vectores de dirección dibujados en cada punto de control.

```
%Programa 3.7 Programa de dibujo a mano alzada usando splines de Bézier
%Haga clic en la ventana de figuras de Matlab para localizar el primer punto y haga
%      tres veces más para especificar 2 puntos de control y el siguiente
%      punto de la spline. Continúe con grupos de 3 puntos para añadir más
%      longitud a la curva. Pulse retorno para terminar el programa.
function bezierdraw
plot([-1 1],[0,0],'k',[0 0],[-1 1],'k');hold on
t=0:.02:1;
[x,y]=ginput(1);              % obtiene un clic del ratón
while(0 == 0)
  [xnew,ynew] = ginput(3);    % obtiene tres clics de ratón
  if length(xnew) < 3
    break                     % si se pulsa retorno, termina
  end
  x=[x;xnew];y=[y;ynew];      % grafica puntos de la spline y pts de control
  plot([x(1) x(2)],[y(1) y(2)],'r:',x(2),y(2),'rs');
  plot([x(3) x(4)],[y(3) y(4)],'r:',x(3),y(3),'rs');
  plot(x(1),y(1),'bo',x(4),y(4),'bo');
  bx=3*(x(2)-x(1)); by=3*(y(2)-y(1)); % ecuaciones de la spline ...
  cx=3*(x(3)-x(2))-bx;cy=3*(y(3)-y(2))-by;
  dx=x(4)-x(1)-bx-cx;dy=y(4)-y(1)-by-cy;
  xp=x(1)+t.*(bx+t.*(cx+t*dx));    % método de Horner
  yp=y(1)+t.*(by+t.*(cy+t*dy));
  plot(xp,yp)                 % gráfica de la curva spline
  x=x(4);y=y(4);              % convertir último en primero y repetir
end
hold off
```

Aunque este análisis se ha restringido a las curvas de Bézier de dos dimensiones, las ecuaciones que las definen pueden extenderse con facilidad a tres dimensiones, en lo que se denomina curvas de Bézier en el espacio. Cada pieza de la spline requiere cuatro puntos (x, y, z), dos puntos extremos y dos puntos de control, tal como en el caso de las dos dimensiones. En los ejercicios se exploran ejemplos de las curvas de Bézier en el espacio.

3.5 Ejercicios

1. Encuentre la curva de Bézier de una sola pieza $(x(t), y(t))$ definida por los cuatro puntos dados.
   (a) (0,0), (0,2), (2,0), (1,0) (b) (1,1), (0,0), (−2,0), (−2,1) (c) (1,2), (1,3), (2,3), (2,2)

2. Encuentre el primer punto extremo, dos puntos de control, y el último punto extremo para las siguientes curvas de Bézier de una sola pieza.

   (a) $\begin{cases} x(t) = 1 + 6t^2 + 2t^3 \\ y(t) = 1 - t + t^3 \end{cases}$ (b) $\begin{cases} x(t) = 3 + 4t - t^2 + 2t^3 \\ y(t) = 2 - t + t^2 + 3t^3 \end{cases}$

   (c) $\begin{cases} x(t) = 2 + t^2 - t^3 \\ y(t) = 1 - t + 2t^3 \end{cases}$

3. Encuentre la curva de Bézier de tres piezas que forme el triángulo con vértices (1, 2), (3, 4) y (5, 1).

4. Construya una spline de Bézier de cuatro piezas que forme un cuadrado con lados de longitud 5.

5. Describa el carácter dibujado por la siguiente curva de Bézier de dos piezas:
   (0,2) (1,2) (1,1) (0,1)
   (0,1) (1,1) (1,0) (0,0)

6. Describa el carácter dibujado por la siguiente curva de Bézier de tres piezas:
   (0,1) (0,1) (0,0) (0,0)
   (0,0) (0,1) (1,1) (1,0)
   (1,0) (1,1) (2,1) (2,0)

7. Encuentre una spline de Bézier de una sola pieza que tenga tangentes verticales en sus puntos extremos $(-1, 0)$ y $(1, 0)$ y que pase por $(0, 1)$.

8. Encuentre una spline de Bézier de una sola pieza que tenga una tangente horizontal en el punto extremo (0, 1), una tangente vertical en el punto extremo (1, 0), y que pase por (1/3, 2/3) en $t = 1/3$.

9. Encuentre la curva de Bézier en el espacio de una sola pieza $(x(t), y(t), z(t))$ definida por los cuatro puntos dados.
   (a) (1, 0, 0), (2, 0, 0), (0, 2, 1), (0, 1, 0) (b) (1, 1, 2), (1, 2, 3), (−1, 0, 0), (1, 1, 1)
   (c) (2, 1, 1), (3, 1, 1), (0, 1, 3), (3, 1, 3)

10. Encuentra los nudos y puntos de control de las siguientes curvas de Bézier en el espacio.

    (a) $\begin{cases} x(t) = 1 + 6t^2 + 2t^3 \\ y(t) = 1 - t + t^3 \\ z(t) = 1 + t + 6t^2 \end{cases}$ (b) $\begin{cases} x(t) = 3 + 4t - t^2 + 2t^3 \\ y(t) = 2 - t + t^2 + 3t^3 \\ z(t) = 3 + t + t^2 - t^3 \end{cases}$

    (c) $\begin{cases} x(t) = 2 + t^2 - t^3 \\ y(t) = 1 - t + 2t^3 \\ z(t) = 2t^3 \end{cases}$

11. Pruebe los hechos en (3.25) y explique cómo se justifican las fórmulas de Bézier.

12. Dados (x_1, y_1), (x_2, y_2), (x_3, y_3) y (x_4, y_4), muestre que las ecuaciones

$$x(t) = x_1(1-t)^3 + 3x_2(1-t)^2t + 3x_3(1-t)t^2 + x_4t^3$$
$$y(t) = y_1(1-t)^3 + 3y_2(1-t)^2t + 3y_3(1-t)t^2 + y_4t^3$$

    dan la curva de Bézier con puntos extremos (x_1, y_1), (x_4, y_4) y puntos de control (x_2, y_2), (x_3, y_3).

3.5 Problemas de computadora

1. Grafique la curva del ejercicio 7.
2. Grafique la curva del ejercicio 8.
3. Grafique la letra a partir de curvas de Bézier, (a) W (b) B (c) C (d) D.

Comprobación en la realidad ✔ 3

Fuentes a partir de las curvas de Bézier

En este proyecto se explica cómo dibujar letras y números empleando curvas de Bézier de dos dimensiones. Pueden aplicarse al modificar el código de MATLAB en el programa 3.7 o escribiendo un archivo PDF.

Las fuentes modernas se construyen directamente a partir de las curvas de Bézier, con el fin de ser independientes de la impresora o del dispositivo de formación de imágenes. Las curvas de Bézier fueron una parte fundamental del lenguaje PostScript desde sus inicios en la década de 1980, y los comandos de PostScript curvas de dibujo que han emigrado en forma ligeramente modificada al formato PDF. A continuación se muestra un archivo PDF completo que ilustra la curva analizada en el ejemplo 3.15.

```
%PDF-1.7
1 0 obj
<<
/Length 2 0 R
>>
stream
100 100 m
100 300 300 300 200 200 c
S
endstream
endobj
2 0 obj
1000
endobj
4 0 obj
<<
/Type /Page
/Parent 5 0 R
/Contents 1 0 R
>>
endobj
5 0 obj
<<
/Kids [4 0 R]
/Count 1
/Type /Pages
/MediaBox [0 0 612 792]
>>
endobj
3 0 obj
<<
/Pages 5 0 R
/Type /Catalog
>>
endobj
xref
0 6
0000000000 65535 f
0000000100 00000 n
0000000200 00000 n
0000000500 00000 n
0000000300 00000 n
0000000400 00000 n
trailer
<<
/Size 6
/Root 3 0 R
>>
startxref
1000
%%EOF
```

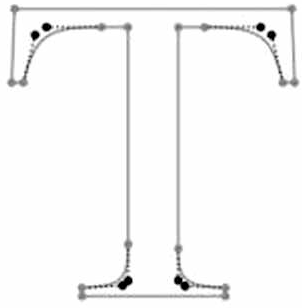


Figura 3.16 T en Times-Roman hecha con splines de Bézier. Los círculos grises son puntos extremos de la spline, y los círculos negros son puntos de control.

La mayoría de los renglones de este archivo hacen varias tareas rutinarias. Por ejemplo, el primer renglón identifica el archivo como un PDF. Es importante enfocarse en las líneas entre `stream` y `endstream` (en la comprobación 3), que son las que identifican a la curva de Bézier. El comando de movimiento (`m`) establece el punto actual de la gráfica como el punto (x, y) especificado por los dos números anteriores, en este caso, el punto (100, 100). El comando de curva (`c`) acepta tres puntos (x, y) y construye la curva de Bézier comenzando en el punto actual de la gráfica, tratando a los tres pares (x, y) como los dos puntos de control y el punto extremo, respectivamente. El comando de trazo (`S`) dibuja la curva.

Este archivo de texto `sample.pdf` puede descargarse desde el sitio web de este libro. Si se abre con un visor de PDF, se mostrará la curva de Bézier de la figura 3.14. Las coordenadas se han multiplicado por 100 para que coincidan con las convenciones predeterminadas de PDF, que son 72 unidades por pulgada. Una hoja de papel tamaño carta tiene 612 unidades de ancho y 792 de alto.

En la actualidad, los caracteres de cientos de fuentes se dibujan en la pantalla de la computadora y en las impresoras empleando curvas de Bézier. Por supuesto, como los archivos PDF suelen contener muchos caracteres, existen atajos hacia las fuentes predefinidas. La información de las curvas de Bézier para las fuentes comunes suele almacenarse en el lector de PDF en vez de en un archivo PDF. Por ahora, se elegirá ignorar este hecho para ver lo que puede hacerse por cuenta propia.

Se iniciará con un ejemplo típico. El carácter de la T mayúscula en la fuente Times Roman se construye a partir de las siguientes 16 curvas de Bézier. Cada línea se compone de los números x_1 y_1 x_2 y_2 x_3 y_3 x_4 y_4 que definen una parte de la spline de Bézier.

```
237 620 237 620 237 120 237 120;
237 120 237  35 226  24 143  19;
143  19 143  19 143   0 143   0;
143   0 143   0 435   0 435   0;
435   0 435   0 435  19 435  19;
435  19 353  23 339  36 339 109;
339 109 339 108 339 620 339 620;
339 620 339 620 393 620 393 620;
393 620 507 620 529 602 552 492;
552 492 552 492 576 492 576 492;
576 492 576 492 570 662 570 662;
570 662 570 662   6 662   6 662;
  6 662   6 662   0 492   0 492;
  0 492   0 492  24 492  24 492;
 24 492  48 602  71 620 183 620;
183 620 183 620 237 620 237 620;
```

Para crear un archivo PDF que escriba la letra T, es necesario añadir comandos en el área stream/endstream del archivo de plantilla anterior. En primer lugar, vaya al punto extremo (237, 620).

```
237 620 m
```

Figura 3.17 5 en Times-Roman hecho con splines Bézier. Los círculos grises son puntos extremos de la spline, y los círculos negros son los puntos de control.

después de lo cual se dibuja la primera curva mediante el comando

```
237 620 237 120 237 120 c
```

seguido por quince comandos `c` adicionales y el comando de trazo (`S`) para terminar la letra T, que se muestra en la figura 3.16. Observe que el comando de movimiento es necesario sólo en el primer paso; después de eso el comando de la siguiente curva toma el punto actual de la gráfica como el primer punto de la curva Bézier siguiente, y sólo se requieren tres puntos más para completar el comando de curva. El comando de la siguiente curva se completa de la misma manera, y así sucesivamente. Como una alternativa al comando de trazo `S`, el comando `f` rellenará el contorno si la figura está cerrada. El comando `b` trazará y también rellenará.

El número 5 se dibuja mediante la siguiente curva de Bézier de 21 piezas y se muestra en la figura 3.17:

```
149 597 149 597 149 597 345 597;
345 597 361 597 365 599 368 606;
368 606 406 695 368 606 406 695;
406 695 397 702 406 695 397 702;
397 702 382 681 372 676 351 676;
351 676 351 676 351 676 142 676;
142 676  33 439 142 676  33 439;
 33 439  32 438  32 436  32 434;
 32 434  32 428  35 426  44 426;
 44 426  74 426 109 420 149 408;
149 408 269 372 324 310 324 208;
324 208 324 112 264  37 185  37;
185  37 165  37 149  44 119  66;
119  66  86  90  65  99  42  99;
 42  99  14  99   0  87   0  62;
  0  62   0  24  46   0 121   0;
121   0 205   0 282  27 333  78;
333  78 378 123 399 180 399 256;
399 256 399 327 381 372 333 422;
333 422 288 468 232 491 112 512;
112 512 112 512 149 597 149 597;
```

Actividades sugeridas

1. Utilice el programa `bezierdraw.m` de la sección 3.5 para trazar la mayúscula inicial de su primer nombre.

2. Modifique el programa de dibujo para que acepte una matriz de números de $n \times 8$, cada fila representa una pieza de una spline de Bézier. Haga que el programa dibuje la letra *f* minúscula en la fuente Times-Roman, utilizando la siguiente curva de Bézier de 21 piezas:

```
289 452 289 452 166 452 166 452;
166 452 166 452 166 568 166 568;
166 568 166 627 185 657 223 657;
223 657 245 657 258 647 276 618;
276 618 292 589 304 580 321 580;
321 580 345 580 363 598 363 621;
363 621 363 657 319 683 259 683;
259 683 196 683 144 656 118 611;
118 611  92 566  84 530  83 450;
 83 450  83 450   1 450   1 450;
  1 450   1 450   1 418   1 418;
  1 418   1 418  83 418  83 418;
 83 418  83 418  83 104  83 104;
 83 104  83  31  72  19   0  15;
  0  15   0  15   0   0   0   0;
  0   0   0   0 260   0 260   0;
260   0 260   0 260  15 260  15;
260  15 178  18 167  29 167 104;
167 104 167 104 167 418 167 418;
167 418 167 418 289 418 289 418;
289 418 289 418 289 452 289 452;
```

3. Use la plantilla anterior y su editor de texto favorito y escriba un archivo PDF que dibuje la letra minúscula `f`. El programa debe comenzar con un comando `m` para trasladarse al primer punto, seguido de 21 comandos `c` y un comando de trazo o relleno. Estos comandos deben estar entre los comandos `stream` y `endstream`. Pruebe el archivo abriéndolo en un visor de PDF.

4. A continuación se presentan algunos otros comandos PDF:

```
1.0 0.0 0.0 RG  % establecer el color de trazo en rojo
0.0 1.0 0.0 rg  % establece el color de relleno en verde
2 w             % establece el ancho de trazo en 2
b               % traza y rellena (S es trazo, f es relleno,
                                                  b es ambos)
```

Los colores se representan de acuerdo con la convención RGB, mediante tres números entre 0 y 1 que incorporan las contribuciones relativas de rojo, verde y azul. Para cambiar el tamaño de las curvas de Bézier, así como para rotar y sesgar los resultados, pueden usarse transformaciones lineales. Tales cambios de coordenadas se realizan con el comando `cm`. Al preceder los comandos de curva con

```
a b c d e f cm
```

para los números reales a, b, c, d, e, f se transforma el sistema de coordenadas mediante

$$x' = ax + by + e$$
$$y' = cx + dy + f.$$

Por ejemplo, si se usa el comando `cm` con $a = d = 0.5, b = c = e = f = 0$, se reduce el tamaño por un factor de 2, y $a = d = -0.5, b = c = 0$, y $e = f = 400$ gira por completo el resultado y traslada en 400 unidades las direcciones x y y. Otras opciones pueden realizar rotaciones, reflexiones o sesgos de las curvas de Bézier originales. Los cambios de coordenadas son acumulativos. En este paso, utilice los comandos del sistema de coordenadas para presentar una versión redimensionada, sesgada y a color de la f minúscula o de otros caracteres.

5. Aunque la información de las fuentes fue un secreto muy bien guardado durante muchos años, gran parte de ésta se encuentra disponible de manera gratuita en la web. Busque otras fuentes y encuentre los datos de las curvas de Bézier que dibujarán las letras de su elección en PDF o empleando `bezierdraw.m`.

6. Diseñe su propia letra o número. Debe comenzar por dibujar la figura en papel cuadriculado, respetando las simetrías que pueden estar presentes. Calcule los puntos de control y esté listo para modificarlos más adelante si así se requiere. ✔

Software y lecturas adicionales

Por lo general, el software de interpolación consta de códigos separados para determinar y evaluar el polinomio de interpolación. MATLAB proporciona los comandos `polyfit` y `polyval` para este propósito. El comando `spline` de MATLAB calcula de manera predeterminada splines sin nudo, pero tiene opciones para otras condiciones de extremo comunes. El comando `interp1` combina otras opciones de interpolación unidimensionales. La biblioteca NAG contiene las subrutinas `e01aef` y `e01baf` para la interpolación de polinomios y splines, y la biblioteca IMSL tiene un número de rutinas para splines basadas en diferentes condiciones de extremo.

Una referencia clásica para los hechos básicos de la interpolación es Davis [1975], y las referencias Rivlin [1981] y Rivlin [1990] cubren la aproximación de funciones y la interpolación de Chebyshev. DeBoor [2001] también es un clásico en lo referente a splines, también vea Schultz [1973] y Schumaker [1981]. En Farin [1990] y Yamaguchi [1988] se estudian las aplicaciones para el modelado y diseño asistidos por computadora. El método CORDIC para la aproximación de funciones especiales se presentó en Volder [1959]. Para obtener más información sobre los archivos PDF, consulte *PDF Reference*, 6a. ed., publicado por Adobe Systems Inc. [2006].

CAPÍTULO

4

Mínimos cuadrados

El sistema de posicionamiento global (GPS) es una tecnología de localización por satélite que proporciona una ubicación precisa en cualquier momento, desde cualquier punto de la Tierra. En tan sólo algunos años, el GPS ha pasado de ser una tecnología de navegación especial utilizada por pilotos, capitanes de barco y excursionistas, a ser utilizada diariamente en automóviles, teléfonos celulares y PDA.

El sistema consta de 24 satélites que siguen órbitas reguladas de manera precisa y que emiten señales sincronizadas. Un receptor con base en la Tierra recoge las señales del satélite, encuentra su distancia desde todos los satélites visibles y utiliza los datos para triangular su posición.

Comprobación en la realidad En la página 238 se muestra el uso de la solución de ecuaciones y el cálculo por mínimos cuadrados para estimar localizaciones a través del GPS.

El concepto de mínimos cuadrados se remonta a la labor pionera de Gauss y Legendre a inicios del siglo XIX. Su uso se extiende a la estadística y a los modelos matemáticos modernos. Las principales técnicas de estimación y parámetros de regresión se han convertido en herramientas fundamentales en las ciencias y la ingeniería.

En este capítulo se presentan las ecuaciones normales y se aplican a una variedad de problemas de ajuste de datos. Posteriormente se explora un enfoque más sofisticado que usa la factorización QR, seguido de un análisis de problemas por mínimos cuadrados no lineales.

4.1 MÍNIMOS CUADRADOS Y ECUACIONES NORMALES

La necesidad de métodos de mínimos cuadrados proviene de dos direcciones, cada una surgió de los estudios realizados en los capítulos 2 y 3. En el capítulo 2 se aprendió a encontrar la solución de $Ax = b$ cuando existe una solución. En este capítulo se analiza qué hacer cuando no hay una solución. Cuando las ecuaciones son inconsistentes, lo cual se presenta si el número de ecuaciones es mayor que el número de incógnitas, la respuesta consiste en encontrar una mejor opción: la aproximación por mínimos cuadrados.

El capítulo 3 estuvo orientado a encontrar polinomios que se ajustan exactamente a conjuntos de puntos. Sin embargo, si los puntos son muchos o se recaban con un cierto margen de error, el ajuste exacto a un polinomio de grado alto rara vez es el mejor enfoque. En tales casos, resulta más razonable adaptarse a un modelo más simple que quizá sólo se aproxime a los datos. Ambos problemas, la resolución de sistemas de ecuaciones inconsistentes y el ajuste aproximado a los datos, son las fuerzas impulsoras detrás de los mínimos cuadrados.

4.1.1 Sistemas de ecuaciones inconsistentes

No es difícil escribir un sistema de ecuaciones que no tenga solución. Considere las siguientes tres ecuaciones con dos incógnitas:

$$\begin{aligned} x_1 + x_2 &= 2 \\ x_1 - x_2 &= 1 \\ x_1 + x_2 &= 3. \end{aligned} \tag{4.1}$$

Cualquier solución debe satisfacer las ecuaciones primera y tercera, lo cual no puede ser verdadero. Un sistema de ecuaciones que no tiene solución se llama **inconsistente**.

¿Cuál es el significado de un sistema sin solución? Tal vez los coeficientes son algo inexactos. En muchos casos, el número de ecuaciones es mayor que el número de incógnitas, por lo que es poco probable que una solución pueda satisfacer todas las ecuaciones. De hecho, las m ecuaciones con n incógnitas por lo regular no tienen solución cuando $m > n$. A pesar de que la eliminación de Gauss no proporcionará una solución para un sistema $Ax = b$ inconsistente, no debe renunciarse a éste por completo. Una alternativa en esta situación es encontrar un vector x que sea lo más cercano a una solución.

Si se elige esta "cercanía" en el sentido de una proximidad en distancia euclidiana, existe un algoritmo sencillo para encontrar las x más cercanas. Estas x especiales se denominan **solución por mínimos cuadrados**.

Es posible tener una mejor idea de la inexistencia de una solución para el sistema (4.1) al escribirlo de una manera diferente. La forma matricial del sistema es $Ax = b$, o

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}. \tag{4.2}$$

La visión alternativa de la multiplicación de matrices/vectores consiste en escribir la ecuación equivalente

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}. \tag{4.3}$$

De hecho, cualquier sistema $Ax = b$ de $m \times n$ puede verse como una ecuación vectorial

$$x_1 v_1 + x_2 v_2 + \cdots + x_n v_n = b, \tag{4.4}$$

que expresa a b como una combinación lineal de las columnas v_i de A, con coeficientes $x_1, \ldots, x_n$. En este caso, se trata de encontrar el vector solución b como una combinación lineal de otros dos vectores tridimensionales. Como las combinaciones de dos vectores tridimensionales forman un plano dentro de R^3, la ecuación (4.3) tiene una solución sólo si el vector b se encuentra en ese plano. Ésta será la situación siempre que se pretendan resolver m ecuaciones con n incógnitas, donde $m > n$. el mayor número de ecuaciones hacen que el problema esté sobredimensionado y que el sistema se vuelva indeterminado.

En la figura 4.1(b) se muestra una dirección hacia la cual ir cuando no existe una solución. No existe un par x_1, x_2 que resuelva (4.1), pero hay un punto en el plano Ax de todas las posibles

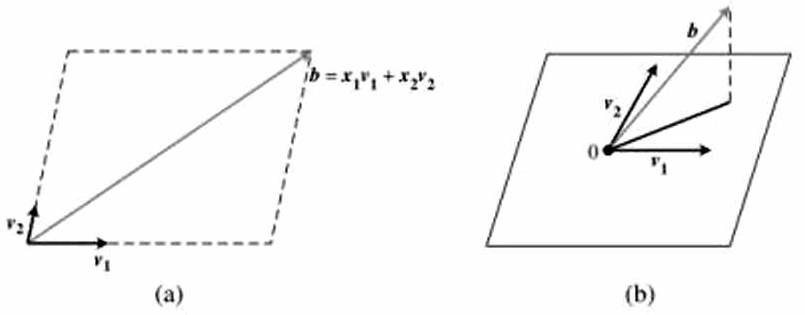


Figura 4.1 solución geométrica de un sistema de tres ecuaciones con dos incógnitas. (a) La ecuación (4.3) requiere que el vector b, el lado derecho de la ecuación, sea una combinación lineal de los vectores columna v_1 y v_2. (b) Si b se encuentra fuera del plano definido por v_1 y v_2, no habrá solución. La solución por mínimos cuadrados $\overline{x}$ hace que el vector de combinación $A\overline{x}$ sea aquel perteneciente al plano Ax que esté más cercano a b.

soluciones que están más cerca de b. Este vector especial $A\overline{x}$ se distingue por el siguiente hecho: El vector residuo $b - A\overline{x}$ es perpendicular al plano $\{Ax | x \in R^n\}$. Este hecho se aprovechará a fin de encontrar una fórmula para $\overline{x}$, la "solución" por mínimos cuadrados.

Primero se establecerá una notación. Recuerde el concepto de la **transpuesta** A^T de la matriz A de $m \times n$, que es la matriz de $n \times m$ cuyos renglones son las columnas de A, y cuyas columnas son los renglones de A, en el mismo orden. La transpuesta de la suma de dos matrices es la suma de las transpuestas, $(A + B)^T = A^T + B^T$. La transpuesta de un producto de dos matrices es el producto de las transpuestas en el orden inverso, es decir, $(AB)^T = B^T A^T$.

Para trabajar con perpendicularidad, recuerde que dos vectores forman un ángulo recto entre sí cuando su producto escalar es cero. Para dos vectores columna de m dimensiones u y v, es posible escribir el producto escalar sólo en términos de multiplicación de matrices

$$u^T v = [u_1, \ldots, u_m] \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}. \tag{4.5}$$

Los vectores u y v son perpendiculares u **ortogonales**, si $u^T \cdot v = 0$, utilizando la multiplicación de matrices.

Ahora, de regreso a la búsqueda de una fórmula para $\overline{x}$. Se ha establecido que

$$(b - A\overline{x}) \perp \{Ax | x \in R^n\}.$$

Si se expresa la perpendicularidad en términos de la multiplicación de matrices, se encuentra que

$$(Ax)^T (b - A\overline{x}) = 0 \text{ para todas las } x \text{ en } R^n.$$

Usando el hecho anterior sobre las transpuestas, es posible reescribir esta expresión como

$$x^T A^T (b - A\overline{x}) = 0 \text{ para todas las } x \text{ en } R^n,$$

ANOTACIÓN

Ortogonalidad Los mínimos cuadrados se basan en la ortogonalidad. La distancia más corta de un punto a un plano se obtiene mediante un segmento de línea ortogonal al plano. Las ecuaciones normales son una forma de cálculo mediante la cual se localiza el segmento de línea que representa el error de mínimos cuadrados.

lo que significa que el vector de n dimensiones $A^T(b - A\overline{x})$ es perpendicular a todo vector x en R^n, incluido él mismo. Sólo hay una forma de que esto suceda:

$$A^T(b - A\overline{x}) = 0.$$

Lo anterior da un sistema de ecuaciones que define la solución por mínimos cuadrados,

$$A^T A\overline{x} = A^T b. \tag{4.6}$$

El sistema de ecuaciones (4.6) se conoce como las **ecuaciones normales**. Su solución $\overline{x}$ se denomina la solución por mínimos cuadrados del sistema $Ax = b$.

Ecuaciones normales para mínimos cuadrados

Dado el sistema inconsistente

$$Ax = b,$$

resuelva

$$A^T A\overline{x} = A^T b$$

para obtener la solución por mínimos cuadrados $\overline{x}$ que disminuye al mínimo la longitud del residuo $r = b - Ax$.

► EJEMPLO 4.1 Use ecuaciones normales para encontrar la solución por mínimos cuadrados del sistema inconsistente (4.1).

El problema en su forma matricial $Ax = b$ tiene

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}.$$

Los componentes de las ecuaciones normales son

$$A^T A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

y

$$A^T b = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}.$$

Las ecuaciones normales

$$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

resolviendo mediante la eliminación de Gauss. La forma de la tabla es

$$\left[\begin{array}{cc|c} 3 & 1 & 6 \\ 1 & 3 & 4 \end{array}\right] \longrightarrow \left[\begin{array}{cc|c} 3 & 1 & 6 \\ 0 & 8/3 & 2 \end{array}\right],$$

que puede resolverse para obtener $\overline{x} = (\overline{x}_1, \overline{x}_2) = (7/4, 3/4)$.

Al sustituir la solución por mínimos cuadrados en el problema original se obtiene

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{7}{4} \\ \frac{3}{4} \end{bmatrix} = \begin{bmatrix} 2.5 \\ 1 \\ 2.5 \end{bmatrix} \neq \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}.$$

Para conocer el ajuste de los datos, se calcula el residuo en la solución $\bar{x}$ por mínimos cuadrados como

$$r = b - A\bar{x} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix} - \begin{bmatrix} 2.5 \\ 1 \\ 2.5 \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.0 \\ 0.5 \end{bmatrix}.$$

Si el residuo es el vector cero, entonces se ha resuelto el sistema original $Ax = b$ con exactitud. Si no es así, la longitud euclidiana del vector residuo es una medida del error de qué tan lejos está $\bar{x}$ de ser solución.

Existen al menos tres formas de expresar el tamaño del residuo. La longitud euclidiana de un vector,

$$||r||_2 = \sqrt{r_1^2 + \cdots + r_m^2}, \tag{4.7}$$

es una norma en el sentido del capítulo 2, llamada la **norma 2**. El **error cuadrático**

$$\text{EC} = r_1^2 + \cdots + r_m^2,$$

y la **raíz media del error cuadrático** (la raíz de la media del error cuadrático)

$$\text{RMEC} = \sqrt{\text{EC}/m} = \sqrt{\left(r_1^2 + \cdots + r_m^2\right)/m}, \tag{4.8}$$

también se usan para medir el error en la solución por mínimos cuadrados. Las tres expresiones están estrechamente relacionadas, a saber:

$$\text{RMEC} = \frac{\sqrt{\text{EC}}}{\sqrt{m}} = \frac{||r||_2}{\sqrt{m}},$$

por lo que al encontrar la $\bar{x}$ que al minimizarse minimiza todo. Para el ejemplo 4.1, el $\text{EC} = (0.5)^2 + 0^2 + (-0.5)^2 = 0.5$, la norma 2 del error es $||r||_2 = \sqrt{0.5} \approx 0.707$, y la $\text{RMEC} = \sqrt{0.5/3} = 1/\sqrt{6} \approx 0.408$.

► EJEMPLO 4.2 Resuelva el problema de mínimos cuadrados $\begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 15 \\ 9 \end{bmatrix}$.

Las ecuaciones normales $A^T Ax = A^T b$ son

$$\begin{bmatrix} 9 & 6 \\ 6 & 29 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 45 \\ 75 \end{bmatrix}.$$

Las soluciones de las ecuaciones normales son $\bar{x}_1 = 3.8$ y $\bar{x}_2 = 1.8$. El vector residual es

$$r = b - A\bar{x} = \begin{bmatrix} -3 \\ 15 \\ 9 \end{bmatrix} - \begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 3.8 \\ 1.8 \end{bmatrix}$$

$$= \begin{bmatrix} -3 \\ 15 \\ 9 \end{bmatrix} - \begin{bmatrix} -3.4 \\ 13 \\ 11.2 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 2 \\ -2.2 \end{bmatrix},$$

que tiene una norma euclidiana $||e||_2 = \sqrt{(0.4)^2 + 2^2 + (-2.2)^2} = 3$. Este problema se resuelve de una manera alternativa en el ejemplo 4.14. ◄

4.1.2 Modelos de ajuste a los datos

Sea $(t_1, y_1), \ldots, (t_m, y_m)$ un conjunto de puntos en el plano, que suelen llamarse "los datos". Existen diferentes formas, como las rectas $y = c_1 + c_2 t$, que permitan ser el modelo que mejor se ajusta a los datos de acuerdo con la norma 2. El concepto principal de los mínimos cuadrados consiste en medir los errores del ajuste que se hace al conjunto de datos y encontrar los parámetros del modelo que disminuyan al mínimo esta cantidad. Este criterio se muestra en la figura 4.2.

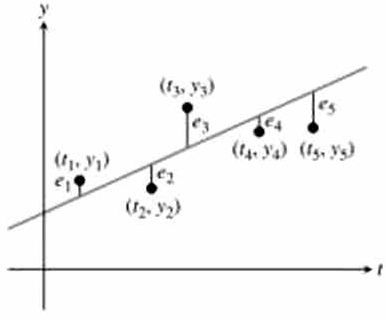


Figura 4.2 Ajuste por mínimos cuadrados de una recta a los datos. La mejor recta es aquella para la cual el error cuadrático $e_1^2 + e_2^2 + \cdots + e_5^2$ es tan pequeño como sea posible, de entre todas las rectas $y = c_1 + c_2 t$.

► **EJEMPLO 4.3** Encuentre la recta que mejor se ajusta a los tres puntos $(t, y) = (1, 2), (-1, 1)$ y $(1,3)$ en la figura 4.3.

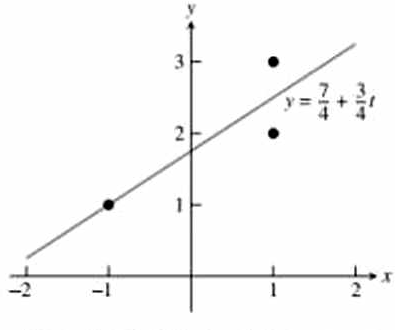


Figura 4.3 La mejor recta en el ejemplo 4.3. Cada uno de los puntos se encuentra por arriba, sobre y por abajo de la mejor recta.

El modelo es $y = c_1 + c_2 t$ y la idea es encontrar los mejores c_1 y c_2. Que al sustituirse en el modelo se tenga

$$\begin{aligned} c_1 + c_2(1) &= 2 \\ c_1 + c_2(-1) &= 1 \\ c_1 + c_2(1) &= 3, \end{aligned}$$

o, en forma matricial,

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}.$$

Se sabe que este sistema no tiene solución (c_1, c_2) por dos razones. En primer lugar, si hay una solución, entonces $y = c_1 + c_2 t$ sería una línea que contenga los tres puntos dados. Sin embargo, es fácil ver que los puntos no están alineados. En segundo lugar, éste es el sistema de la ecuación (4.2) que se comentó al principio de este capítulo. En ese momento se observó que la primera y tercera ecuaciones son inconsistentes, y se encontró que la mejor solución en términos de mínimos cuadrados es $(c_1, c_2) = (7/4, 3/4)$. Por lo tanto, la mejor recta es $y = 7/4 + 3/4t$. ◄

El ajuste puede evaluarse usando las fórmulas de medición de los errores indicadas con anterioridad. Los residuos en los puntos de datos son

t	y	línea	error
1	2	2.5	−0.5
−1	1	1.0	0.0
1	3	2.5	0.5

y la RMEC es $1/\sqrt{6}$, como se vio con anterioridad.

El ejemplo anterior sugiere un programa de tres pasos para resolver los problemas de ajustar datos por mínimos cuadrados.

Ajuste de datos por mínimos cuadrados

Dado un conjunto de m puntos $(t_1, y_1), \ldots, (t_m, y_m)$.

PASO 1. Elija un modelo. Identifique un modelo tal como $y = c_1 + c_2 t$, que se utilizará para ajustar los datos.

PASO 2. Haga que el modelo se ajuste a los datos. Sustituya los datos en el modelo. Cada punto crea una ecuación cuyas incógnitas son los parámetros, c_1 y c_2 en el modelo de recta. Esto resulta en un sistema $Ax = b$, donde la incógnita x representa los parámetros desconocidos.

PASO 3. Resuelva las ecuaciones normales. La solución por mínimos cuadrados se encontrará como la solución al sistema de ecuaciones normales $A^T Ax = A^T b$.

Estos pasos se muestran en el siguiente ejemplo:

► EJEMPLO 4.4 Encuentre la mejor recta y la mejor parábola para los cuatro puntos $(-1, 1), (0, 0), (1, 0), (2, -2)$ de la figura 4.4.

De acuerdo con lo anterior, se seguirán los tres pasos:
(1) Se elige el modelo $y = c_1 + c_2 t$. (2) Al hacer que el modelo se ajuste a los datos, se obtiene

ANOTACIÓN

Resumen La técnica de mínimos cuadrados es un ejemplo clásico de manejo de datos. La entrada consiste en un conjunto de puntos y la salida es un modelo que, con un número relativamente reducido de parámetros, se ajusta a los datos tanto como sea posible. Por lo general, la razón para utilizar los mínimos cuadrados es reemplazar los datos problemáticos con un modelo convincente. Después, el modelo suele utilizarse para predecir señales o con fines de clasificación.

En la sección 4.2 se usan diferentes modelos para ajustar datos, incluyendo funciones polinómicas, exponenciales y trigonométricas. El enfoque trigonométrico se estudiará en los capítulos 10 y 11, donde se discute el análisis elemental de Fourier como una introducción al procesamiento de señales.

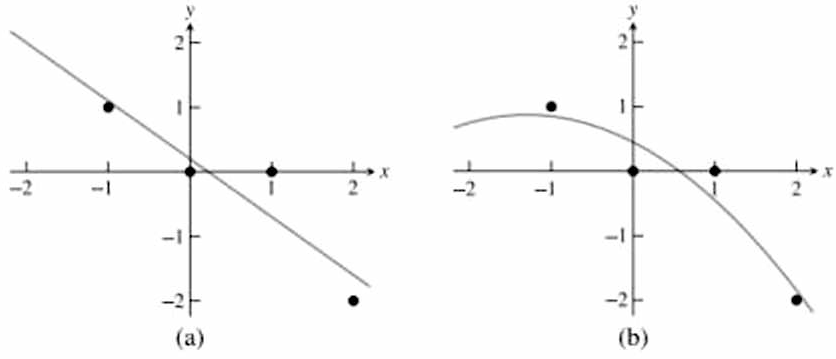


Figura 4.4 Ajuste por mínimos cuadrados a los puntos del ejemplo 4.4. (a) La mejor recta es $y = 0.2 - 0.9t$. La RMEC es 0.418. (b) La mejor parábola es $y = 0.45 - 0.65t - 0.25t^2$. La RMEC es 0.335.

$$\begin{aligned} c_1 + c_2(-1) &= 1 \\ c_1 + c_2(0) &= 0 \\ c_1 + c_2(1) &= 0 \\ c_1 + c_2(2) &= -2, \end{aligned}$$

o, en forma matricial,

$$\begin{bmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -2 \end{bmatrix}.$$

(3) Las ecuaciones normales son

$$\begin{bmatrix} 4 & 2 \\ 2 & 6 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -5 \end{bmatrix}.$$

Resolviendo para c_1 y c_2 la recta que mejor se ajusta es $y = c_1 + c_2 t = 0.2 - 0.9t$.

Los residuos son

t	y	línea	error
−1	1	1.1	−0.1
0	0	0.2	−0.2
1	0	−0.7	0.7
2	−2	−1.6	−0.4

Las estadísticas de error son el error cuadrático $EC = (-.1)^2 + (-.2)^2 + (.7)^2 + (-.4)^2 = 0.7$ y la $RMEC = \sqrt{.7}/\sqrt{4} = 0.418$.

Enseguida, manteniendo los mismos cuatro puntos, pero con un cambio en el modelo. Establezcamos $y = c_1 + c_2 t + c_3 t^2$ y sustituyamos los puntos para obtener

$$\begin{aligned} c_1 + c_2(-1) + c_3(-1)^2 &= 1 \\ c_1 + c_2(0) + c_3(0)^2 &= 0 \\ c_1 + c_2(1) + c_3(1)^2 &= 0 \\ c_1 + c_2(2) + c_3(2)^2 &= -2, \end{aligned}$$

ANOTACIÓN

Condicionamiento Debido a que en los problemas de mínimos cuadrados se supone que los datos de entrada están sujetos a error, es muy importante minimizar el error. Se han presentado las ecuaciones normales como el método más sencillo para resolver el problema de mínimos cuadrados, y está muy bien para los problemas pequeños. Sin embargo, el número de condición cond(A^TA) es aproximadamente el cuadrado del cond(A) original, lo que aumentará en gran medida la posibilidad de que el problema esté mal condicionado. Los métodos más sofisticados permiten calcular la solución por mínimos cuadrados directamente a partir de A sin formar A^TA. Estos métodos se basan en la factorización QR, que se presenta en la sección 4.3, y en la descomposición singular de valor del capítulo 12.

o, en forma matricial,

$$\begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -2 \end{bmatrix}.$$

Esta vez, las ecuaciones normales son tres ecuaciones con tres incógnitas:

$$\begin{bmatrix} 4 & 2 & 6 \\ 2 & 6 & 8 \\ 6 & 8 & 18 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -5 \\ -7 \end{bmatrix}.$$

Al resolver se obtiene que la mejor parábola es $y = c_1 + c_2t + c_3t^2 = 0.45 - 0.65t - 0.25t^2$. Los errores residuales se dan en la siguiente tabla:

t	y	parábola	error
−1	1	0.85	0.15
0	0	0.45	−0.45
1	0	−0.45	0.45
2	−2	−1.85	−0.15

Los errores son: el error cuadrático EC = $(.15)^2 + (-.45)^2 + (.45)^2 + (-.15)^2 = 0.45$ y RMEC = $\sqrt{.45}/\sqrt{4} \approx 0.335$.

Los comandos `polyfit` y `polyval` de MATLAB están diseñados no sólo para interpolar datos, sino también para ajustar los datos con modelos polinomiales. Para n puntos de datos de entrada, `polyfit` usado con un grado $n - 1$ devuelve los coeficientes del polinomio de interpolación de grado $n - 1$. Por otro lado, si el grado de entrada es menor que $n - 1$, `polyfit` encuentra el mejor polinomio de ajuste por mínimos cuadrados de ese grado. Por ejemplo, los comandos

```
>> x0=[-1 0 1 2];
>> y0=[1 0 0 -2];
>> c=polyfit(x0,y0,2);
>> x=-1:.01:2;
>> y=polyval(c,x);
>> plot(x0,y0,'o',x,y)
```

encuentra los coeficientes del polinomio de mínimos cuadrados de grado dos y lo grafica junto con los datos dados en el ejemplo 4.4.

El ejemplo 4.4 muestra que el modelado por mínimos cuadrados no necesita limitarse a la búsqueda de las mejores rectas. Al ampliar la definición del modelo, es posible ajustar los coeficientes para cualquier modelo, siempre y cuando los coeficientes se introduzcan al modelo de una manera lineal.

4.1.3 Condicionamiento de mínimos cuadrados

Se ha visto que el problema de mínimos cuadrados se reduce a resolver las ecuaciones normales $A^T A\overline{x} = A^T b$. ¿Con qué precisión puede determinarse la solución por mínimos cuadrados $\overline{x}$? Ésta es una pregunta acerca del error de las ecuaciones normales. Para contestar esta pregunta se llevará a cabo un experimento con el cálculo en doble precisión numérica, resolviendo las ecuaciones normales en un caso donde se conoce la respuesta correcta.

► **EJEMPLO 4.5** Sean $x_1 = 2.0, x_2 = 2.2, x_3 = 2.4, \ldots, x_{11} = 4.0$ puntos igualmente espaciados en el intervalo $[2, 4]$ y establezca $y_i = 1 + x_i + x_i^2 + x_i^3 + x_i^4 + x_i^5 + x_i^6 + x_i^7$ para $1 \le i \le 11$. Use las ecuaciones normales para encontrar el polinomio de mínimos cuadrados $P(x) = c_1 + c_2 x + \cdots + c_8 x^7$ que ajusta al conjunto de puntos (x_i, y_i).

Un polinomio de séptimo grado ajusta los 11 puntos que se encuentran ese polinomio es $P(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 + x^7$. Obviamente, la solución correcta por mínimos cuadrados es $c_1 = c_2 = \cdots = c_8 = 1$. Al sustituir los puntos en el modelo $P(x)$ resulta el sistema $Ac = b$:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^7 \\ 1 & x_2 & x_2^2 & \cdots & x_2^7 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{11} & x_{11}^2 & \cdots & x_{11}^7 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_8 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{11} \end{bmatrix}.$$

La matriz de coeficientes A es una **matriz de Van der Monde**, cuya j-ésima columna se compone de los elementos de la segunda columna elevados a la potencia $(j - 1)$. Si se usa MATLAB para resolver las ecuaciones normales, se tiene:

```
>> x = (2+(0:10)/5)';
>> y = 1+x+x.^2+x.^3+x.^4+x.^5+x.^6+x.^7;
>> A = [x.^0 x x.^2 x.^3 x.^4 x.^5 x.^6 x.^7];
>> c = (A'*A)\(A'*y)

c=
    1.5134
   -0.2644
    2.3211
    0.2408
    1.2592
    0.9474
    1.0059
    0.9997

>> cond(A'*A)

ans=
  1.4359e+019
```

Al resolver las ecuaciones normales en doble precisión no puede obtenerse un valor preciso para la solución por mínimos cuadrados. El número de condiciones al estructurar $A^T A$ es grande para resolver con aritmética de doble precisión, y las ecuaciones normales están mal condicionadas, a pesar de que el problema original de mínimos cuadrados está moderadamente condicionado. Es evidente que se puede mejorar el enfoque de las ecuaciones normales para los mínimos cuadrados. En el ejemplo 4.15, este problema se retoma después de desarrollar una alternativa que evita la formación de $A^T A$. ◄

4.1 Ejercicios

1. Resuelva las ecuaciones normales y encuentre la solución por mínimos cuadrados y el error de la norma 2 para los siguientes sistemas indeterminados:

(a) $\begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix}$ (c) $\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 2 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 2 \end{bmatrix}$

2. Encuentre las soluciones por mínimos cuadrados y las RMEC de los siguientes sistemas:

(a) $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 2 \\ 1 & 1 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 2 \end{bmatrix}$

3. Encuentre la solución por mínimos cuadrados del sistema indeterminado

$$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 6 \end{bmatrix}.$$

4. Sea $m \geq n$, A la matriz identidad de $m \times n$ (la submatriz principal de la matriz identidad de $m \times m$) y sea $b = [b_1, \ldots, b_m]$ un vector. Encuentre la solución por mínimos cuadrados de $Ax = b$ y el error de la norma 2.

5. Demuestre que la norma 2 es una norma vectorial. Deberá utilizar la desigualdad de Cauchy-Schwarz $|u \cdot v| \leq \|u\|_2 \|v\|_2$.

6. Sea A una matriz no singular de $n \times n$. (a) Demuestre que $(A^T)^{-1} = (A^{-1})^T$. (b) Sea b un vector de n datos, entonces $Ax = b$ tiene exactamente una solución. Demuestre que esta solución satisface las ecuaciones normales.

7. Encuentre la mejor recta que pasa a través del siguiente conjunto de puntos y determine la RMEC:

(a) $(-3, 3), (-1, 2), (0, 1), (1, -1), (3, -4)$ (b) $(1, 1), (1, 2), (2, 2), (2, 3), (4, 3)$.

8. Encuentre la mejor recta que pasa por cada conjunto de puntos y determine la RMEC:

(a) $(0, 0), (1, 3), (2, 3), (5, 6)$ (b) $(1, 2), (3, 2), (4, 1), (6, 3)$ (c) $(0, 5), (1, 3), (2, 3), (3, 1)$.

9. Encuentre la mejor parábola que pasa por cada conjunto de puntos del ejercicio 8, y compare la RMEC con el mejor ajuste lineal.

10. Encuentre el mejor polinomio de tercer grado para cada conjunto de datos del ejercicio 8. Además, encuentre el polinomio de interpolación de tercer grado y compárelos.

11. Suponga que la altura registrada de un cohete se mide en cuatro instantes distintos, y que los tiempos y las alturas medidas son $(t, h) = (1, 135), (2, 265), (3, 385), (4, 485)$, en segundos y metros respectivamente. Ajuste el modelo $h = a + bt - 4.905t^2$ para estimar la altura máxima posible del cohete y cuándo regresará a tierra.

12. Dados los puntos $(x, y, z) = (0, 0, 3), (0, 1, 2), (1, 0, 3), (1, 1, 5), (1, 2, 6)$, encuentre el plano tridimensional (modelo $z = c_0 + c_1 x + c_2 y$) que mejor se ajuste a los datos.

4.1 Problemas de computadora

1. Forme las ecuaciones normales y calcule la solución por mínimos cuadrados y el error de la norma 2 para los siguientes sistemas indeterminados:

(a) $$\begin{bmatrix} 3 & -1 & 2 \\ 4 & 1 & 0 \\ -3 & 2 & 1 \\ 1 & 1 & 5 \\ -2 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ -5 \\ 15 \\ 0 \end{bmatrix}$$

(b) $$\begin{bmatrix} 4 & 2 & 3 & 0 \\ -2 & 3 & -1 & 1 \\ 1 & 3 & -4 & 2 \\ 1 & 0 & 1 & -1 \\ 3 & 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 2 \\ 0 \\ 5 \end{bmatrix}$$

2. Considere los datos de la producción mundial de petróleo del problema de computadora 3.2.3. Encuentre por mínimos cuadrados (a) la mejor recta, (b) la mejor parábola y (c) la mejor curva cúbica que pasa por los 10 puntos de datos y la RMEC de los ajustes. Use cada ajuste para estimar el nivel de producción en 2010. ¿Cuál ajuste representa mejor los datos en términos de la RMEC?

3. Considere los datos de la población mundial del problema de computadora 3.1.1. Encuentre por mínimos cuadrados (a) la mejor recta y (b) la mejor parábola que pasa por los puntos de datos, y la RMEC del ajuste. En cada caso, estime la población de 1980. ¿Cuál ajuste ofrece la mejor estimación?

4. Considere los datos de la concentración de dióxido de carbono del ejercicio 3.1.13. Encuentre por mínimos cuadrados (a) la mejor recta, (b) la mejor parábola y (c) la mejor curva cúbica que pasa por los puntos, calcule las RMEC y estime la concentración de CO_2 de 1950, en cada caso.

5. Una empresa está probando en el mercado un nuevo refresco en 22 ciudades de población aproximadamente igual. El precio de venta (en dólares) y el número de refrescos vendidos por semana en las ciudades son los siguientes:

ciudad	precio	ventas/semana
1	0.59	3980
2	0.80	2200
3	0.95	1850
4	0.45	6100
5	0.79	2100
6	0.99	1700
7	0.90	2000
8	0.65	4200
9	0.79	2440
10	0.69	3300
11	0.79	2300

ciudad	precio	ventas/semana
12	0.49	6000
13	1.09	1190
14	0.95	1960
15	0.79	2760
16	0.65	4330
17	0.45	6960
18	0.60	4160
19	0.89	1990
20	0.79	2860
21	0.99	1920
22	0.85	2160

(a) En primer lugar, la empresa quiere encontrar la "curva de demanda": ¿cuántos refrescos venderá a cada precio ? Sean P y S el precio y las ventas por semana, respectivamente. Encuentre la recta $S = c_1 + c_2P$ que mejor se ajuste a los datos de la tabla en el sentido de los mínimos cuadrados. Encuentre las ecuaciones normales y los coeficientes c_1 y c_2 de la recta de mínimos cuadrados. Grafique la recta de mínimos cuadrados junto con los datos y calcule el error cuadrático medio.

(b) Después de estudiar los resultados de la prueba de mercado, la compañía fijará un precio de venta P único en todo el país. Si el costo de fabricación es de 0.23 dólares por unidad, la utilidad total (por ciudad, por semana) es $S(P - 0.23)$ dólares. Utilice los resultados de la anterior aproximación por mínimos cuadrados para encontrar el precio de venta que incremente al máximo el beneficio de la empresa.

6. ¿Cuál es la "pendiente" de la parábola $y = x^2$ en el intervalo [0, 1]? Encuentre la mejor recta de mínimos cuadrados que se ajuste a la parábola en n puntos igualmente espaciados en el intervalo

para (a) $n = 10$ y (b) $n = 20$. Grafique la parábola y las rectas. ¿Cuál espera que sea el resultado cuando $n \to \infty$? (c) Encuentre el mínimo de la función $F(c_1, c_2) = \int_0^1 (x^2 - c_1 - c_2 x)^2 \, dx$, y explique su relación con el problema.

7. Encuentre por mínimos cuadrados (a) la recta, (b) la parábola que pasa por los 13 puntos de la figura 3.5 y la RMEC para cada ajuste.

8. Sea A la matriz de $10 \times n$ formada por las primeras n columnas de la matriz de Hilbert de 10×10. Sea c el vector de n entradas $[1, \ldots, 1]$, y establezca $b = Ac$. Utilice las ecuaciones normales y resuelva el problema de mínimos cuadrados $Ax = b$ para (a) $n = 6$ (b) $n = 8$ y comparare su respuesta con la solución correcta por mínimos cuadrados $\overline{x} = c$. ¿Cuántas posiciones decimales correctas pueden calcularse? Utilice las condiciones para explicar los resultados. (Este problema se resuelve con el problema de computadora 7 de la sección 4.3).

9. Sean $x_1, \ldots, x_{11}$ 11 puntos espaciados uniformemente en el intervalo $[2, 4]$ y $y_i = 1 + x_i + x_i^2 + \cdots + x_i^d$. Utilice las ecuaciones normales para calcular el mejor polinomio de grado d, donde (a) $d = 5$ (b) $d = 6$ (c) $d = 8$. Compare su respuesta con la del ejemplo 4.5. ¿Cuántos posiciones decimales correctas pueden calcularse? Utilice la condición para explicar los resultados. (Este problema se resuelve con el problema de computadora 8 de la sección 4.3).

10. Los siguientes datos, presentados por el Departamento de Comercio de Estados Unidos, representan el cambio porcentual anual en el promedio del ingreso personal disponible en Estados Unidos durante los últimos 15 años. Además, se presenta la proporción del electorado que votó a favor del candidato presidencial del partido en el poder. La primera línea de la tabla dice que el ingreso aumentó en un 1.49% de 1951 a 1952 y que el 44.6% de los electores votaron a favor de Adlai Stevenson, el candidato presidencial del partido demócrata en el poder. Encuentre el mejor modelo lineal de mínimos cuadrados para la votación por el partido en el poder como una función del cambio en el ingreso. Grafique esta recta a lo largo de los 15 puntos presentados. ¿Cuántos puntos porcentuales del voto puede esperar el partido en el poder por cada punto porcentual adicional de cambio en el ingreso personal?

año	porcentaje de cambio en el ingreso	porcentaje de voto por el partido en el poder
1952	1.49	44.6
1956	3.03	57.8
1960	0.57	49.9
1964	5.74	61.3
1968	3.51	49.6
1972	3.73	61.8
1976	2.98	49.0
1980	−0.18	44.7
1984	6.23	59.2
1988	3.38	53.9
1992	2.15	46.5
1996	2.10	54.7
2000	3.93	50.3
2004	2.47	51.2
2008	−0.41	45.7

4.2 EXPLORACIÓN DE MODELOS

Los modelos lineales y polinómicos anteriores ilustran el uso de los mínimos cuadrados para ajustar los datos. La técnica de modelado de datos incluye una amplia variedad de modelos, algunos derivados de los principios físicos detrás de la fuente de los datos y otros basados en factores empíricos.

4.2.1 Datos periódicos

Los datos periódicos requieren modelos periódicos. Por ejemplo, las temperaturas al aire libre obedecen a ciclos en diferentes escalas de tiempo, que incluyen ciclos diarios y anuales regidos por la rotación de la Tierra y su traslación alrededor del sol. Como un primer ejemplo, los datos horarios de la temperatura se ajustan a funciones periódicas como el seno y el coseno.

► **EJEMPLO 4.6** Ajuste a un modelo periódico las temperaturas registradas en Washington, D.C. el 1 de enero de 2001, que se indican en la siguiente tabla:

hora del día	t	temp (C)
12 de la noche	0	−2.2
3 am	$\frac{1}{8}$	−2.8
6 am	$\frac{1}{4}$	−6.1
9 am	$\frac{3}{8}$	−3.9
12 del mediodía	$\frac{1}{2}$	0.0
3 pm	$\frac{5}{8}$	1.1
6 pm	$\frac{3}{4}$	−0.6
9 pm	$\frac{7}{8}$	−1.1

Se elige el modelo $y = c_1 + c_2\cos 2\pi t + c_3 \operatorname{sen} 2\pi t$ para que coincida con el hecho de que la temperatura es periódica en 24 horas. El modelo utiliza esta información al fijar el periodo como exactamente de un día, donde se usan días como las unidades de t. En la tabla, la variable t se presenta en estas unidades.

Al sustituir los datos en el modelo resulta el siguiente sistema de ecuaciones lineales:

$$
\begin{aligned}
c_1 + c_2\cos 2\pi(0) + c_3 \operatorname{sen} 2\pi(0) &= -2.2\\
c_1 + c_2\cos 2\pi\left(\frac{1}{8}\right) + c_3 \operatorname{sen} 2\pi\left(\frac{1}{8}\right) &= -2.8\\
c_1 + c_2\cos 2\pi\left(\frac{1}{4}\right) + c_3 \operatorname{sen} 2\pi\left(\frac{1}{4}\right) &= -6.1\\
c_1 + c_2\cos 2\pi\left(\frac{3}{8}\right) + c_3 \operatorname{sen} 2\pi\left(\frac{3}{8}\right) &= -3.9\\
c_1 + c_2\cos 2\pi\left(\frac{1}{2}\right) + c_3 \operatorname{sen} 2\pi\left(\frac{1}{2}\right) &= 0.0\\
c_1 + c_2\cos 2\pi\left(\frac{5}{8}\right) + c_3 \operatorname{sen} 2\pi\left(\frac{5}{8}\right) &= 1.1\\
c_1 + c_2\cos 2\pi\left(\frac{3}{4}\right) + c_3 \operatorname{sen} 2\pi\left(\frac{3}{4}\right) &= -0.6\\
c_1 + c_2\cos 2\pi\left(\frac{7}{8}\right) + c_3 \operatorname{sen} 2\pi\left(\frac{7}{8}\right) &= -1.1
\end{aligned}
$$

ANOTACIÓN **Ortogonalidad** El problema de mínimos cuadrados puede simplificarse considerablemente mediante la elección de funciones adecuadas. Por ejemplo, las opciones de los ejemplos 4.6 y 4.7 producen ecuaciones normales que ya están en forma diagonal. Esta propiedad de funciones básicas ortogonales se analiza a detalle en el capítulo 10. El modelo (4.9) es una expansión de Fourier.

La correspondiente ecuación matricial inconsistente es $Ax = b$, donde

$$A = \begin{bmatrix} 1 & \cos 0 & \operatorname{sen} 0 \\ 1 & \cos\frac{\pi}{4} & \operatorname{sen}\frac{\pi}{4} \\ 1 & \cos\frac{\pi}{2} & \operatorname{sen}\frac{\pi}{2} \\ 1 & \cos\frac{3\pi}{4} & \operatorname{sen}\frac{3\pi}{4} \\ 1 & \cos\pi & \operatorname{sen}\pi \\ 1 & \cos\frac{5\pi}{4} & \operatorname{sen}\frac{5\pi}{4} \\ 1 & \cos\frac{3\pi}{2} & \operatorname{sen}\frac{3\pi}{2} \\ 1 & \cos\frac{7\pi}{4} & \operatorname{sen}\frac{7\pi}{4} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & \sqrt{2}/2 & \sqrt{2}/2 \\ 1 & 0 & 1 \\ 1 & -\sqrt{2}/2 & \sqrt{2}/2 \\ 1 & -1 & 0 \\ 1 & -\sqrt{2}/2 & -\sqrt{2}/2 \\ 1 & 0 & -1 \\ 1 & \sqrt{2}/2 & -\sqrt{2}/2 \end{bmatrix} \quad \text{y} \quad b = \begin{bmatrix} -2.2 \\ -2.8 \\ -6.1 \\ -3.9 \\ 0.0 \\ 1.1 \\ -0.6 \\ -1.1 \end{bmatrix}.$$

Las ecuaciones normales $A^T Ac = A^T b$ son

$$\begin{bmatrix} 8 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} -15.6 \\ -2.9778 \\ -10.2376 \end{bmatrix},$$

que puede resolverse fácilmente como $c_1 = -1.95$, $c_2 = -0.7445$ y $c_3 = -2.5594$. La mejor versión del modelo, en el sentido de los mínimos cuadrados, es $y = -1.9500 - 0.7445 \cos 2\pi t - 2.5594 \operatorname{sen} 2\pi t$, con RMEC ≈ 1.063. En la figura 4.5(a) se compara el ajuste de los mínimos cuadrados con las temperaturas reales registradas cada hora. ◄

► EJEMPLO 4.7 Ajuste los datos de temperatura al modelo mejorado

$$y = c_1 + c_2 \cos 2\pi t + c_3 \operatorname{sen} 2\pi t + c_4 \cos 4\pi t. \tag{4.9}$$

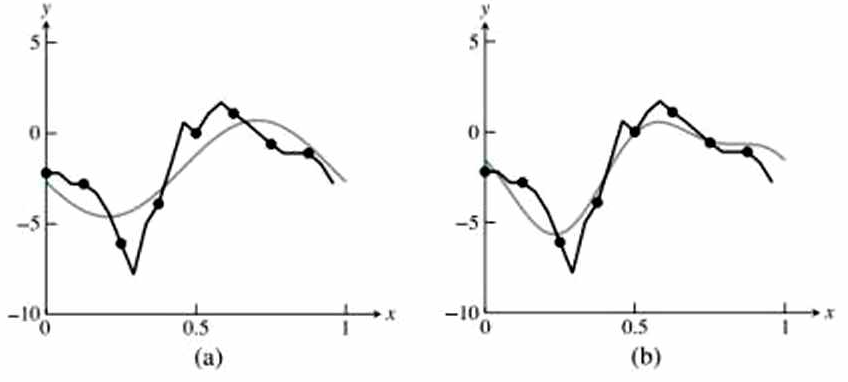


Figura 4.5 Ajustes por mínimos cuadrados a los datos periódicos del ejemplo 4.6. (a) El modelo sinusoidal $y = -1.95 - 0.7445\cos 2\pi t - 2.5594\operatorname{sen} 2\pi t$ se presenta en color gris, junto con el trazo de las temperaturas registradas el 1 de enero de 2001. (b) El sinusoidal mejorado $y = -1.95 - 0.7445\cos 2\pi t - 2.5594\operatorname{sen} 2\pi t + 1.125\cos 4\pi t$ se ajusta mejor a los datos.

Ahora, el sistema de ecuaciones es

$$
\begin{aligned}
c_1 + c_2\cos 2\pi(0) + c_3\,\text{sen}\,2\pi(0) + c_4\cos 4\pi(0) &= -2.2\\
c_1 + c_2\cos 2\pi\left(\frac{1}{8}\right) + c_3\,\text{sen}\,2\pi\left(\frac{1}{8}\right) + c_4\cos 4\pi\left(\frac{1}{8}\right) &= -2.8\\
c_1 + c_2\cos 2\pi\left(\frac{1}{4}\right) + c_3\,\text{sen}\,2\pi\left(\frac{1}{4}\right) + c_4\cos 4\pi\left(\frac{1}{4}\right) &= -6.1\\
c_1 + c_2\cos 2\pi\left(\frac{3}{8}\right) + c_3\,\text{sen}\,2\pi\left(\frac{3}{8}\right) + c_4\cos 4\pi\left(\frac{3}{8}\right) &= -3.9\\
c_1 + c_2\cos 2\pi\left(\frac{1}{2}\right) + c_3\,\text{sen}\,2\pi\left(\frac{1}{2}\right) + c_4\cos 4\pi\left(\frac{1}{2}\right) &= 0.0\\
c_1 + c_2\cos 2\pi\left(\frac{5}{8}\right) + c_3\,\text{sen}\,2\pi\left(\frac{5}{8}\right) + c_4\cos 4\pi\left(\frac{5}{8}\right) &= 1.1\\
c_1 + c_2\cos 2\pi\left(\frac{3}{4}\right) + c_3\,\text{sen}\,2\pi\left(\frac{3}{4}\right) + c_4\cos 4\pi\left(\frac{3}{4}\right) &= -0.6\\
c_1 + c_2\cos 2\pi\left(\frac{7}{8}\right) + c_3\,\text{sen}\,2\pi\left(\frac{7}{8}\right) + c_4\cos 4\pi\left(\frac{7}{8}\right) &= -1.1,
\end{aligned}
$$

lo que conduce a las siguientes ecuaciones normales:

$$
\begin{bmatrix} 8 & 0 & 0 & 0\\ 0 & 4 & 0 & 0\\ 0 & 0 & 4 & 0\\ 0 & 0 & 0 & 4 \end{bmatrix}
\begin{bmatrix} c_1\\ c_2\\ c_3\\ c_4 \end{bmatrix} =
\begin{bmatrix} -15.6\\ -2.9778\\ -10.2376\\ 4.5 \end{bmatrix}.
$$

Las soluciones son $c_1 = -1.95$, $c_2 = -0.7445$, $c_3 = -2.5594$, y $c_4 = 1.125$, con RMEC ≈ 0.705. En la figura 4.5(b) se muestra que el modelo $y = -1.95 - 0.7445\cos 2\pi t - 2.5594\,\text{sen}\,2\pi t + 1.125\cos 4\pi t$ mejora el ajuste de manera sustancial. ◀

4.2.2 Linealización de datos

El crecimiento exponencial de una población está implícito cuando su tasa de cambio es proporcional a su tamaño. En condiciones perfectas, cuando el crecimiento es estable y la población no crece, el modelo es una buena representación.

El **modelo exponencial**

$$y = c_1 e^{c_2 t} \tag{4.10}$$

no puede ajustarse en forma directa mediante mínimos cuadrados debido a que c_2 no aparece linealmente en la ecuación del modelo. Una vez que se sustituyen los datos en el modelo, la dificultad es clara: el conjunto de ecuaciones para resolver el sistema son no lineales y no pueden expresarse como un sistema lineal $Ax = b$. Por lo tanto, la deducción de las ecuaciones normales es irrelevante.

Existen dos maneras de enfrentar el problema de las ecuaciones no lineales. La manera más difícil consiste en disminuir al mínimo directamente el error por mínimos cuadrados, es decir, resolver el problema no lineal de mínimos cuadrados. Este problema se abordará de nuevo en la sección 4.5. La forma más sencilla es cambiar el problema. En vez de resolver el problema original de mínimos cuadrados, puede resolverse uno diferente, que está relacionado con el original, cambiando a un modelo "lineal".

En el caso del modelo exponencial (4.10), el modelo se vuelve lineal mediante la aplicación del logaritmo natural:

$$\ln y = \ln(c_1 e^{c_2 t}) = \ln c_1 + c_2 t. \tag{4.11}$$

Observe que para un modelo exponencial, la gráfica de $\ln y$ es una gráfica lineal en t. A primera vista, parece que sólo se ha intercambiado un problema por otro. Ahora, el coeficiente c_2 es lineal en el modelo, pero c_1 ya no lo es. Sin embargo, al renombrar $k = \ln c_1$, es posible escribir

$$\ln y = k + c_2 t. \tag{4.12}$$

Ahora ambos coeficientes k y c_2 son lineales en el modelo. Después de resolver las ecuaciones normales para obtener los mejores k y c_2, puede encontrarse la correspondiente $c_1 = e^k$ si así se desea.

Cabe señalar que la forma de evitar la dificultad de los coeficientes no lineales consistió en cambiar el problema. El problema original de mínimos cuadrados consistía en ajustar los datos a (4.10), es decir, en encontrar los c_1, c_2 que minimicen

$$(c_1 e^{c_2 t_1} - y_1)^2 + \cdots + (c_1 e^{c_2 t_m} - y_m)^2, \tag{4.13}$$

la suma de los cuadrados de los residuos de las ecuaciones $c_1 e^{c_2 t_i} = y_i$ para $i = 1, \ldots, m$. Por el momento, puede resolverse el problema modificado minimizando el error de mínimos cuadrados en el "espacio log", es decir, al encontrar c_1, c_2 que minimicen

$$(\ln c_1 + c_2 t_1 - \ln y_1)^2 + \cdots + (\ln c_1 + c_2 t_m - \ln y_m)^2, \tag{4.14}$$

la suma de los cuadrados de los residuos de las ecuaciones $\ln c_1 + c_2 t_i = \ln y_i$ para $i = 1, \ldots, m$. Éstas son dos minimizaciones diferentes que tienen soluciones distintas, lo que significa que suelen dar como resultado diferentes valores para los coeficientes c_1 y c_2.

¿Cuál método es el correcto para este problema, los mínimos cuadrados no lineales de (4.13) o la versión del modelo lineal (4.14)? El primero es un problema de mínimos cuadrados, tal como se ha definido. El segundo no lo es. Sin embargo, dependiendo del contexto de los datos, cualquiera de los dos puede ser la elección más natural. Para responder a la pregunta, el usuario debe decidir cuáles errores son más importantes de minimizar, los errores en el sentido original o los errores en el "espacio log". De hecho, el modelo log es lineal y es posible argumentar que, sólo después de transformar logarítmicamente los datos a una relación lineal, resulta natural evaluar la capacidad del modelo.

► EJEMPLO 4.8 Use el modelo lineal para encontrar el mejor ajuste exponencial $y = c_1 e^{c_2 t}$ por mínimos cuadrados a los datos de la oferta mundial de automóviles:

año	automóviles ($\times 10^6$)
1950	53.05
1955	73.04
1960	98.31
1965	139.78
1970	193.48
1975	260.20
1980	320.39

Los datos describen el número de automóviles circulando en todo el mundo en el año dado. Defina la variable t del tiempo en términos de años a partir de 1950. Al resolver el problema lineal de mínimos cuadrados produce $k_1 \approx 3.9896$, $c_2 \approx 0.06152$. Como $c_1 \approx e^{3.9896} \approx 54.03$, el modelo es

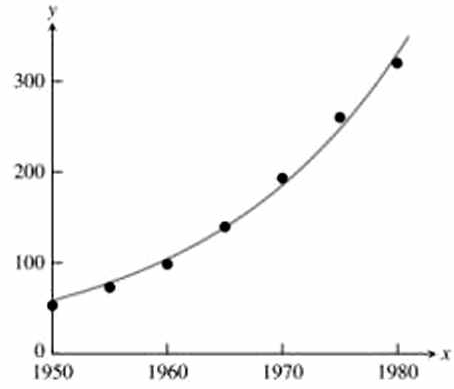


Figura 4.6 Ajuste exponencial de los datos de la oferta mundial de automóviles usando la linealización. El mejor ajuste por mínimos cuadrados es $y = 54.03e^{0.06152t}$. Compare esta figura con la 4.14.

$y = 54.03e^{0.06152t}$. La RMEC para el modelo logarítmico linealizado en el espacio log es ≈ 0.0357, mientras que la RMEC para el modelo exponencial original ≈ 9.56. En la figura 4.6 se grafica el mejor modelo ajustado a los datos. ◄

► EJEMPLO 4.9 En la tabla siguiente se presenta el número de transistores en las unidades centrales de procesamiento Intel desde principios la década de 1970. Ajuste el modelo $y = c_1e^{c_2t}$ a los datos.

CPU	año	transistores
4004	1971	2,250
8008	1972	2,500
8080	1974	5,000
8086	1978	29,000
286	1982	120,000
386	1985	275,000
486	1989	1,180,000
Pentium	1993	3,100,000
Pentium II	1997	7,500,000
Pentium III	1999	24,000,000
Pentium 4	2000	42,000,000
Itanium	2002	220,000,000
Itanium 2	2003	410,000,000

Los parámetros se ajustarán mediante el modelo lineal de (4.11). Al linealizar el modelo se obtiene

$$\ln y = k + c_2t.$$

Se considerará que $t = 0$ correspondiente al año 1970. Al sustituir los datos en el modelo lineal resulta

$$\begin{aligned} k + c_2(1) &= \ln 2250 \\ k + c_2(2) &= \ln 2500 \\ k + c_2(4) &= \ln 5000 \\ k + c_2(8) &= \ln 29000, \end{aligned} \tag{4.15}$$

y así sucesivamente. La ecuación matricial es $Ax = b$, donde $x = (k, c_2)$,

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 4 \\ 1 & 8 \\ \vdots & \vdots \\ 1 & 33 \end{bmatrix}, \quad \text{y} \quad b = \begin{bmatrix} \ln 2250 \\ \ln 2500 \\ \ln 5000 \\ \ln 29000 \\ \vdots \\ \ln 410000000 \end{bmatrix}. \tag{4.16}$$

Las ecuaciones normales $A^T Ax = A^T b$ son

$$\begin{bmatrix} 13 & 235 \\ 235 & 5927 \end{bmatrix} \begin{bmatrix} k \\ c_2 \end{bmatrix} = \begin{bmatrix} 176.90 \\ 3793.23 \end{bmatrix},$$

que tienen la solución $k \approx 7.197$ y $c_2 \approx 0.3546$, lo que conduce a $c_1 = e^k \approx 1335.3$. La curva exponencial $y = 1335.3e^{0.3546t}$ se muestra en la figura 4.7. El tiempo para duplicar la producción está dada por la relación $\ln 2/c_2 \approx 1.95$ años. Gordon C. Moore, cofundador de Intel, predijo en 1965 que en la década siguiente, la potencia de cómputo se duplicaría cada 2 años. Sorprendentemente, esa tasa exponencial ha seguido durante 40 años. Existe alguna evidencia en la figura 4.7 de que esta tasa se ha acelerado desde 2000.

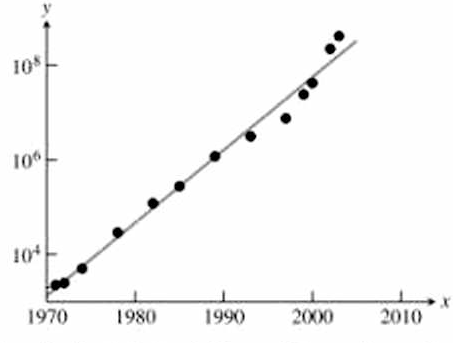


Figura 4.7 Gráfica semilogarítmica de la ley de Moore. Número de transistores en los chips del CPU contra el tiempo (años). ◀

Otro ejemplo importante con coeficientes no lineales es el modelo de la **ley de potencia** $y = c_1 t^{c_2}$. Este modelo también puede simplificarse mediante la linealización al aplicar logaritmos en ambos lados:

$$\begin{aligned} \ln y &= \ln c_1 + c_2 \ln t \\ &= k + c_2 \ln t. \end{aligned} \tag{4.17}$$

Al sustituir los datos en el modelo se obtendrá

$$k + c_2 \ln t_1 = \ln y_1 \tag{4.18}$$

$$\vdots$$

$$k + c_2 \ln t_n = \ln y_n, \tag{4.19}$$

dando como resultado la forma matricial

$$A = \begin{bmatrix} 1 & \ln t_1 \\ \vdots & \vdots \\ 1 & \ln t_n \end{bmatrix} \quad \text{y} \quad b = \begin{bmatrix} \ln y_1 \\ \vdots \\ \ln y_n \end{bmatrix}. \tag{4.20}$$

Las ecuaciones normales permiten determinar k y c_2, así como $c_1 = e^k$.

► EJEMPLO 4.10 Use la linealización para ajustar los datos dados de altura y peso con el modelo de la ley de potencia.

En 2002, mediante la encuesta nacional de salud y nutrición en Estados Unidos, los centros para el control de enfermedades (CCE) recopilaron los datos de la altura y el peso medio de los niños con edades entre 2 y 11 años, lo que resultó en la siguiente tabla:

edad (años)	altura (m)	peso (kg)
2	0.9120	13.7
3	0.9860	15.9
4	1.0600	18.5
5	1.1300	21.3
6	1.1900	23.5
7	1.2600	27.2
8	1.3200	32.7
9	1.3800	36.0
10	1.4100	38.6
11	1.4900	43.7

Siguiendo la estrategia anterior, la ley de potencia resultante para el peso contra la altura es $W = 16.3H^{2.42}$. La relación se grafica en la figura 4.8. Dado que el peso es un indicador de volumen, el coeficiente $c_2 \approx 2.42$ puede verse como la "dimensión efectiva" del cuerpo humano.

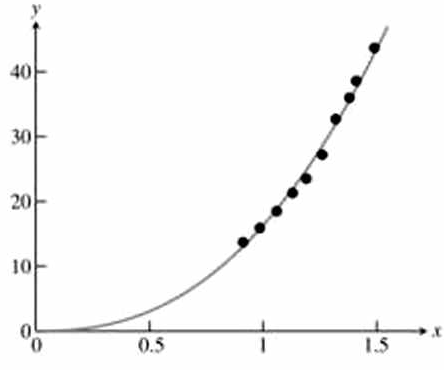


Figura 4.8 Ley de potencia para el peso y la altura en niños de 2 a 11 años de edad. La mejor fórmula de ajuste es $W = 16.3H^{2.42}$. ◄

La concentración de fármacos temporal y en el torrente sanguíneo está descrito por

$$y = c_1 t e^{c_2 t}, \tag{4.21}$$

donde t denota el tiempo transcurrido después de administrar el fármaco. Las características del modelo son un aumento rápido cuando el medicamento entra en el torrente sanguíneo, seguido de un decaimiento exponencial lento. La **vida media** del fármaco es el tiempo desde la concentración máxima hasta el momento cuando se reduce a un medio de ese nivel. El modelo puede linealizarse al aplicar el logaritmo natural en ambos lados, lo que produce

$$\ln y = \ln c_1 + \ln t + c_2 t$$
$$k + c_2 t = \ln y - \ln t,$$

donde se ha establecido $k = \ln c_1$. Esto conduce a la ecuación matricial $Ax = b$, en la que

$$A = \begin{bmatrix} 1 & t_1 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \quad \text{y} \quad b = \begin{bmatrix} \ln y_1 - \ln t_1 \\ \vdots \\ \ln y_m - \ln t_m \end{bmatrix}. \tag{4.22}$$

Las ecuaciones normales se resuelven para k y c_2, así como para $c_1 = e^k$.

► **EJEMPLO 4.11** Ajuste el modelo de la ecuación (4.21) con el nivel medido del fármaco norfluoxetina en el torrente sanguíneo de un paciente, dado en la tabla siguiente:

hora	concentración (ng/ml)
1	8.0
2	12.3
3	15.5
4	16.8
5	17.1
6	15.8
7	15.2
8	14.0

Al resolver las ecuaciones normales se obtiene $k \approx 2.28$, $c_2 \approx -0.215$ y $c_1 \approx e^{2.28} \approx 9.77$. La mejor versión del modelo es $y = 9.77te^{-0.215t}$, que se grafica en la figura 4.9. A partir del modelo, puede estimarse el momento de concentración máxima y la vida media (vea el problema de computadora 5).

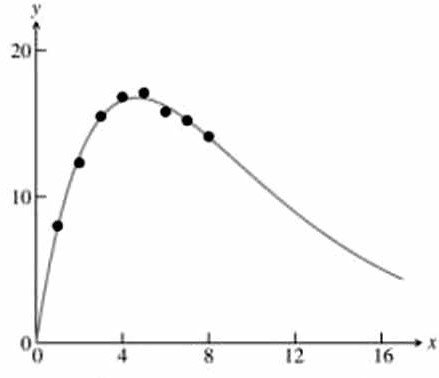


Figura 4.9 Gráfica de la concentración del fármaco en la sangre. El modelo (4.21) muestra un decaimiento exponencial después del pico inicial. ◄

Es importante darse cuenta de que el modelo de linealización cambia el problema de mínimos cuadrados. La solución obtenida disminuirá al mínimo la RMEC con respecto al problema linealizado, no necesariamente al problema original, que en general tiene un conjunto de parámetros óptimos diferente. Si se introducen en el modelo en forma no lineal, no pueden calcularse a partir de las ecuaciones normales, y es necesario emplear técnicas no lineales para resolver el problema de mínimos cuadrados original. Esto se logra mediante el método de Gauss-Newton presentado en la sección 4.5, donde se revisan los datos de la oferta de automóviles y se compara el ajuste del modelo exponencial en las formas linealizada y no linealizada.

4.2 Ejercicios

1. Ajuste los datos al modelo periódico $y = F_3(t) = c_1 + c_2 \cos 2\pi t + c_3 \operatorname{sen} 2\pi t$. Encuentre el error de la norma 2 y la RMEC.

(a)

t	y
0	1
1/4	3
1/2	2
3/4	0

(b)

t	y
0	1
1/4	3
1/2	2
3/4	1

(c)

t	y
0	3
1/2	1
1	3
3/2	2

2. Ajuste los datos a los modelos periódicos $F_3(t) = c_1 + c_2 \cos 2\pi t + c_3 \operatorname{sen} 2\pi t$ y $F_4(t) = c_1 + c_2 \cos 2\pi t + c_3 \operatorname{sen} 2\pi t + c_4 \cos 4\pi t$. Encuentre los errores de la norma 2 $\|e\|_2$ y compárelos con los ajustes de F_3 y F_4.

(a)

t	y
0	0
1/6	2
1/3	0
1/2	−1
2/3	1
5/6	1

(b)

t	y
0	4
1/6	2
1/3	0
1/2	−5
2/3	−1
5/6	3

3. Ajuste los datos al modelo exponencial mediante la linealización. Encuentre la norma 2 de la diferencia entre los puntos de datos y_i y el mejor modelo $c_1 e^{c_2 t_i}$.

(a)

t	y
−2	1
0	2
1	2
2	5

(b)

t	y
0	1
1	1
1	2
2	4

4. Ajuste los datos al modelo exponencial mediante la linealización. Encuentre la norma 2 de la diferencia entre los puntos de datos y_i y el mejor modelo $c_1 e^{c_2 t_i}$.

(a)

t	y
−2	4
−1	2
1	1
2	1/2

(b)

t	y
0	10
1	5
2	2
3	1

5. Ajuste los datos al modelo de la ley de potencia usando la linealización. Encuentre la RMEC del ajuste.

(a)

t	y
1	6
2	2
3	1
4	1

(b)

t	y
1	2
1	4
2	5
3	6
5	10

6. Ajuste los datos al modelo de concentración del fármaco (4.21). Encuentre la RMEC para el ajuste.

(a)

t	y
1	3
2	4
3	5
4	5

(b)

t	y
1	2
2	4
3	3
4	2

4.2 Problemas de computadora

1. Ajuste los datos mensuales de consumo de petróleo en Japón (2003) que se muestran en la siguiente tabla, con el modelo periódico (4.9). Además calcule la RMEC:

mes	uso de petróleo (10^6 brls/día)
Ene	6.224
Feb	6.665
Mar	6.241
Abr	5.302
May	5.073
Jun	5.127
Jul	4.994
Ago	5.012
Sep	5.108
Oct	5.377
Nov	5.510
Dic	6.372

2. Los datos de temperatura del ejemplo 4.6 se tomaron del sitio del Clima Subterráneo `www.wunderground.com`. Encuentre una selección similar de datos de temperatura por hora en un lugar y fecha de su elección, y ajústelos con los dos modelos sinusoidales del ejemplo mencionado.

3. Considere los datos de la población mundial en el problema de computadora de la sección 3.1. Encuentre el mejor ajuste exponencial de los puntos usando la linealización. Estime la población de 1980 y encuentre el error de la estimación.

4. Considere los datos de la concentración de dióxido de carbono en el ejercicio 17 de la sección 3.1. Encuentre el mejor ajuste exponencial de la diferencia entre el nivel de CO_2 y la base (279 ppm) usando la linealización. Estime la concentración de CO_2 en 1950 y encuentre el error de la estimación.

5. (a) Determine el tiempo en el que se alcanzará la concentración máxima en el modelo (4.21). (b) Utilice un solucionador de ecuaciones para estimar la vida media, a partir del modelo del ejemplo 4.11.

6. En la tabla adjunta se proporciona la concentración de un fármaco en el torrente sanguíneo, medida cada hora después de su administración. Ajuste el modelo (4.21). Encuentre la concentración máxima estimada y la vida media del fármaco. Suponga que el rango terapéutico para el fármaco es 4-15 ng/ml. Utilice el método para resolver el sistema de ecuaciones de su preferencia para estimar el tiempo que la concentración del fármaco se mantiene dentro de los niveles terapéuticos.

hora	concentración (ng/ml)
1	6.2
2	9.5
3	12.3
4	13.9
5	14.6
6	13.5
7	13.3
8	12.7
9	12.4
10	11.9

7. El archivo `windmill.txt`, disponible en el sitio web de este libro, es una lista de 60 números que representan los megavatios-hora mensuales generados desde enero de 2005 hasta diciembre de 2009 por un aerogenerador propiedad de la Cooperativa de Energía Minnkota cerca de Valley City, Dakota del Norte. Los datos están disponibles en http://www.minnkota.com. Como referencia, un hogar típico consume alrededor de 1 MWh por mes.

   (a) Encuentre un modelo aproximado de la salida de potencia como una función periódica anual. Ajuste los datos a la ecuación (4.9),

   $$f(t) = c_1 + c_2 \cos 2\pi t + c_3 \operatorname{sen} 2\pi t + c_4 \cos 4\pi t$$

   donde las unidades de t están en años, es decir, $0 \leq t \leq 5$, y escriba la función resultante.

   (b) Grafique los datos y la función del modelo para los años $0 \leq t \leq 5$. ¿Qué características de los datos están capturadas en el modelo?

8. El archivo `scrippsy.txt`, disponible en el sitio web de este libro, muestra una lista de 50 números que representan la concentración de dióxido de carbono en la atmósfera, en partes por millón por volumen (ppv), registradas en Mauna Loa, Hawai, cada día 15 de mayo desde el año 1961 hasta 2010. Los datos son parte de un esfuerzo de recopilación de datos iniciado por Charles Keeling, del Instituto Scripps de Oceanografía (Keeling *et al.* [2001]). Reste el nivel básico de 279 ppm como en el problema de computadora 4, y ajuste los datos a un modelo exponencial. Grafique los datos junto con la mejor función exponencial de ajuste y calcule RMEC.

9. El archivo `scrippsm.txt`, disponible en el sitio web de este libro, muestra una lista de 180 números que representan la concentración de dióxido de carbono en la atmósfera, en partes por millón por volumen (ppv), registrada mensualmente en Mauna Loa desde enero de 1996 hasta diciembre de 2010, y que se tomó del mismo estudio de Scripps que el problema de computadora 8.

   (a) Realice un ajuste por mínimos cuadrados de los datos de CO_2 empleando el modelo

   $$f(t) = c_1 + c_2 t + c_3 \cos 2\pi t + c_4 \operatorname{sen} 2\pi t$$

   donde t se mide en meses. Reporte los mejores coeficientes c_i de ajuste y la RMEC del ajuste. Grafique la curva continua desde enero de 1989 hasta el final de este año, incluyendo los 180 puntos de datos de la gráfica.

   (b) Utilice el modelo para predecir la concentración de CO_2 en mayo de 2004, septiembre de 2004, mayo de 2005 y septiembre de 2005. Estos meses tienden a contener los máximos y mínimos anuales del ciclo de CO_2. Los valores reales registrados son 380.63, 374.06, 382.45 y 376.73 ppv, respectivamente. Reporte el error del modelo en estos cuatro puntos.

   (c) Añada el término extra $c_5 \cos 4\pi t$ y vuelva a realizar los incisos (a) y (b). Compare la nueva RMEC y los cuatro errores del modelo.

(d) Repita el inciso (c) empleando el término c_5t^2 extra. ¿Cuál de los términos conduce a una mayor mejora del modelo, el del inciso (c) o (d)?

(e) Añada ambos términos de los incisos (c) y (d) y vuelva a realizar los incisos (a) y (b). Prepare una tabla que resuma los resultados de todos los incisos del problema y trate de dar una explicación de los resultados.

Consulte el sitio web `http://scrippsco2.ucsd.edu` donde encontrará muchos más datos y análisis del estudio del dióxido de carbono de Scripps.

4.3 FACTORIZACIÓN QR

En el capítulo 2 se usó la factorización LU para resolver ecuaciones matriciales. La factorización es útil debido a que codifica los pasos de la eliminación gaussiana. En esta sección, se desarrolla la factorización QR como una manera de resolver los cálculos de mínimos cuadrados que son superiores a las ecuaciones normales.

Después de introducir la factorización por medio de la ortogonalización de Gram-Schmidt, se retomará el ejemplo 4.5, para el que las ecuaciones normales resultaron ser inadecuadas. Más adelante en esta sección, se presentan las reflexiones de Householder como un método más eficiente para calcular Q y R.

4.3.1 Ortogonalización de Gram-Schmidt y mínimos cuadrados

El método de Gram-Schmidt ortogonaliza un conjunto de vectores. Dado un conjunto de entradas de vectores m-dimensionales, el objetivo es encontrar un sistema de coordenadas ortogonal para el subespacio generado por el conjunto. De manera más precisa, dados n vectores de entrada linealmente independientes, calcula n vectores unitarios mutuamente perpendiculares que abarcan el mismo subespacio que los vectores de entrada. La longitud unitaria es con respecto a la norma 2 o euclidiana (4.7), que se utiliza en todo el capítulo 4.

Sean $A_1, \ldots, A_n$ vectores linealmente independientes de R^m. Para $n \leq m$. El método Gram-Schmidt comienza por dividir A_1 entre su longitud para que sea un vector unitario. Defina

$$y_1 = A_1 \quad \text{y} \quad q_1 = \frac{y_1}{||y_1||_2}. \tag{4.23}$$

Para encontrar el segundo vector unitario, reste la proyección de A_2 en la dirección de q_1 y normalice el resultado:

$$y_2 = A_2 - q_1(q_1^T A_2), \quad \text{y} \quad q_2 = \frac{y_2}{||y_2||_2}. \tag{4.24}$$

Entonces $q_1^T y_2 = q_1^T (A_2 - q_1(q_1^T A_2)) = q_1^T A_2 - q_1^T A_2 = 0$, por lo que q_1 y q_2 son pares ortogonales, como se muestra en la figura 4.10.

En el j-ésimo paso, defina

$$y_j = A_j - q_1(q_1^T A_j) - q_2(q_2^T A_j) - \ldots - q_{j-1}(q_{j-1}^T A_j) \quad \text{y} \quad q_j = \frac{y_j}{||y_j||_2}. \tag{4.25}$$

Es evidente que q_j es ortogonal a cada uno de las q_i producidas anteriormente para $i = 1, \ldots, j-1$, puesto que (4.25) implica

$$\begin{aligned} q_i^T y_j &= q_i^T A_j - q_i^T q_1 q_1^T A_j - \ldots - q_i^T q_{j-1} q_{j-1}^T A_j \\ &= q_i^T A_j - q_i^T q_i q_i^T A_j = 0, \end{aligned}$$

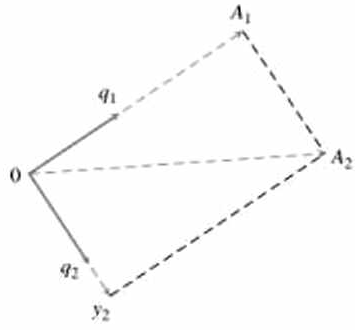


Figura 4.10 Ortogonalización de Gram-Schmidt. Los vectores de entrada son A_1 y A_2, y la salida es el conjunto ortonormal que consta de q_1 y q_2. El segundo vector ortogonal q_2 se forma al restar la proyección de A_2 en la dirección de q_1 desde A_2, seguida por la normalización.

donde por inducción, las q_i, son pares ortogonales para $i < j$. Geométricamente, (4.25) corresponde a restar de A_j las proyecciones de A_j sobre los vectores ortogonales previamente determinados q_i, $i = 1, \ldots, j-1$. Lo que es ortogonal a la q_i y, después de dividir entre su longitud, para convertirlo en un vector unitario, se utiliza como q_j. Por lo tanto, el conjunto $\{q_1, \ldots, q_n\}$ consta de vectores mutuamente ortogonales que atraviesan el mismo subespacio de R^m como $\{A_1, \ldots, A_n\}$.

El resultado de la ortogonalización de Gram-Schmidt puede ponerse en forma matricial al introducir una nueva notación para los productos punto en el cálculo anterior. Defina $r_{jj} = ||y_j||_2$ y $r_{ij} = q_i^T A_j$. Entonces (4.23) y (4.24) pueden escribirse como

$$A_1 = r_{11} q_1$$
$$A_2 = r_{12} q_1 + r_{22} q_2,$$

y el caso general (4.25) se traduce en

$$A_j = r_{1j} q_1 + \cdots + r_{j-1,j} q_{j-1} + r_{jj} q_j.$$

Por lo tanto, el resultado de la ortogonalización de Gram-Schmidt puede escribirse en forma matricial como

$$(A_1 | \cdots | A_n) = (q_1 | \cdots | q_n) \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}, \tag{4.26}$$

o $A = QR$, donde se considera que A es la matriz consistente en las columnas A_j. A esto se le llama **factorización QR reducida**; la versión completa se presenta más adelante. El supuesto de que los vectores A_j son linealmente independientes garantiza que los coeficientes de la diagonal principal r_{jj} sean distintos de cero. A la inversa, si A_j está en el intervalo de $A_1, \ldots, A_{j-1}$, entonces las proyecciones sobre los últimos vectores constituyen el vector entero, y $r_{jj} = ||y_j||_2 = 0$.

► EJEMPLO 4.12 Encuentre la factorización QR reducida aplicando la ortogonalización de Gram-Schmidt a las columnas de $A = \begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix}$.

Establezca $y_1 = A_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$. Entonces $r_{11} = ||y_1||_2 = \sqrt{1^2 + 2^2 + 2^2} = 3$, y el primer vector unitario es

$$q_1 = \frac{y_1}{||y_1||_2} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{bmatrix}.$$

Para encontrar el segundo vector unitario, establezca

$$y_2 = A_2 - q_1 q_1^T A_2 = \begin{bmatrix} -4 \\ 3 \\ 2 \end{bmatrix} - \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{bmatrix} 2 = \begin{bmatrix} -\frac{14}{3} \\ \frac{5}{3} \\ \frac{2}{3} \end{bmatrix}$$

y

$$q_2 = \frac{y_2}{||y_2||_2} = \frac{1}{5} \begin{bmatrix} -\frac{14}{3} \\ \frac{5}{3} \\ \frac{2}{3} \end{bmatrix} = \begin{bmatrix} -\frac{14}{15} \\ \frac{1}{3} \\ \frac{2}{15} \end{bmatrix}.$$

Como $r_{12} = q_1^T A_2 = 2$ y $r_{22} = ||y_2||_2 = 5$, el resultado en la forma matricial (4.26) es

$$A = \begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 1/3 & -14/15 \\ 2/3 & 1/3 \\ 2/3 & 2/15 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 0 & 5 \end{bmatrix} = QR.$$

◀

Para esta versión de Gram-Schmidt se usa el término "clásica", puesto que al final de esta sección se proporcionará una versión actualizada o "modificada".

Ortogonalización clásica de Gram-Schmidt

Sean $A_j, j = 1, \ldots, n$ vectores linealmente independientes.

for $j = 1, 2, \ldots, n$
  $y = A_j$
  for $i = 1, 2, \ldots, j - 1$
    $r_{ij} = q_i^T A_j$
    $y = y - r_{ij} q_i$
  end
  $r_{jj} = ||y||_2$
  $q_j = y / r_{jj}$
end

Cuando el método es correcto, se acostumbra llenar la matriz de vectores unitarios ortogonales para una base completa de R^m, para alcanzar la factorización QR "completa". Esto puede lograrse, por ejemplo, al añadir $m - n$ vectores adicionales a la A_j, de modo que los m vectores abarquen R^m, para después llevar a cabo el método de Gram-Schmidt. En términos de la base de R^m formada por $q_1, \ldots, q_m$, los vectores originales pueden expresarse como

$$(A_1|\cdots|A_n) = (q_1|\cdots|q_m)\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}. \tag{4.27}$$

Esta ecuación matricial es la **factorización QR completa** de la matriz $A = (A_1 \mid \ldots \mid An)$, formada por los vectores de entrada originales. Observe los tamaños de matriz en la factorización QR completa: A es de $m \times n$, Q es una matriz cuadrada de $m \times m$, y la matriz triangular superior R es de $m \times n$, del mismo tamaño que A. La matriz Q en la factorización QR completa tiene un lugar especial en el análisis numérico y se le da una definición especial.

DEFINICIÓN 4.1 Una matriz cuadrada Q es **ortogonal** si $Q^T = Q^{-1}$. ❐

Observe que una matriz cuadrada es ortogonal si y sólo si sus columnas son vectores unitarios en pares ortogonales (ejercicio 9). Por lo tanto, una factorización QR completa es la ecuación $A = QR$, donde Q es una matriz cuadrada ortogonal y R es una matriz triangular superior del mismo tamaño que A.

La propiedad clave de una matriz ortogonal es que conserva la norma euclidiana de un vector.

LEMA 4.1 Si Q es una matriz ortogonal de $m \times m$ y x es un vector m-dimensional, entonces $||Qx||_2 = ||x||_2$. ■

Demostración. $||Qx||_2^2 = (Qx)^T Qx = x^T Q^T Qx = x^T x = ||x||_2^2$. ❐

El producto de dos matrices ortogonales de $m \times m$ es de nuevo ortogonal (ejercicio 10). La factorización QR de una matriz de $m \times m$ mediante el método de Gram-Schmidt requiere aproximadamente m^3 de multiplicaciones/divisiones, tres veces más que la factorización LU, además de aproximadamente el mismo número de sumas (ejercicio 11).

► EJEMPLO 4.13 Encuentre la factorización QR completa de $A = \begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix}$.

ANOTACIÓN **Ortogonalidad** En el capítulo 2 se encontró que la factorización LU es un medio eficiente para representar la información de la eliminación gaussiana. De la misma manera, la factorización QR registra la ortogonalización de una matriz, a saber, la construcción de un conjunto ortogonal que abarca el espacio de los vectores columna de A. Se prefiere la realización de cálculos con matrices ortogonales porque (1) su definición las hace fáciles de invertir y, (2) por el lema 4.2, no magnifican los errores.

En el ejemplo 4.12 se encontraron los vectores unitarios ortogonales $q_1 = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{bmatrix}$ y $q_2 = \begin{bmatrix} -\frac{14}{15} \\ \frac{1}{3} \\ \frac{2}{15} \end{bmatrix}$.

Al agregar un tercer vector $A_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ se llega a

$$y_3 = A_3 - q_1 q_1^T A_3 - q_2 q_2^T A_3$$
$$= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{bmatrix} \frac{1}{3} - \begin{bmatrix} -\frac{14}{15} \\ \frac{1}{3} \\ -\frac{2}{15} \end{bmatrix} \left(-\frac{14}{15}\right) = \frac{2}{225} \begin{bmatrix} 2 \\ 10 \\ -11 \end{bmatrix}$$

y $q_3 = y_3/||y_3|| = \begin{bmatrix} \frac{2}{15} \\ \frac{10}{15} \\ -\frac{11}{15} \end{bmatrix}$. Si se reúnen las partes se obtiene la factorización QR completa

$$A = \begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 1/3 & -14/15 & 2/15 \\ 2/3 & 1/3 & 2/3 \\ 2/3 & 2/15 & -11/15 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 0 & 5 \\ 0 & 0 \end{bmatrix} = QR.$$

Tenga en cuenta que la elección de A_3 fue arbitraria. Pudo haberse usado cualquier tercer vector columna linealmente independiente de las dos primeras columnas. Compare este resultado con la factorización QR reducida del ejemplo 4.12. ◀

El comando `qr` de MATLAB realiza la factorización QR de una matriz de $m \times n$. No utiliza la ortogonalización de Gram-Schmidt, sino que emplea métodos más eficientes y estables que se presentarán en una subsección posterior. El comando

```
>> [Q,R]=qr(A,0)
```

devuelve la factorización QR reducida y

```
>> [Q,R]=qr(A)
```

devuelve la factorización QR completa.

Existen tres aplicaciones principales de la factorización QR. Aquí se describirán dos de ellas, y la tercera es el algoritmo QR para el cálculo de valores propios (o característicos), que se presentará en el capítulo 12.

En primer lugar, la factorización QR puede utilizarse para resolver un sistema de n ecuaciones con n incógnitas $Ax = b$. Sólo es necesario factorizar $A = QR$ y la ecuación $Ax = b$ se convierte en $QRx = b$ y $Rx = Q^T b$. Si se supone que A es no singular, las entradas de la diagonal de la matriz triangular superior R son distintas de cero, de modo que R es no singular. Una sustitución triangular hacia atrás da la solución x. Como se mencionó antes, este enfoque es más difícil de entender comparado con el enfoque LU.

La segunda aplicación es a los mínimos cuadrados. Sea A una matriz de $m \times n$ con $m \geq n$. Para minimizar $||Ax - b||_2$, vuelva a escribirla como $||QRx - b||_2 = ||Rx - Q^T b||_2$ por el lema 4.2.

El vector dentro de la norma euclidiana es

$$\begin{bmatrix} e_1 \\ \vdots \\ e_n \\ \hdashline e_{n+1} \\ \vdots \\ e_m \end{bmatrix} = \left[\begin{array}{cccc} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \\ \hdashline 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{array}\right] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} d_1 \\ \vdots \\ d_n \\ \hdashline d_{n+1} \\ \vdots \\ d_m \end{bmatrix} \tag{4.28}$$

donde $d = Q^T b$. Suponga que $r_{ii} \neq 0$. Entonces la parte superior $(e_1, \ldots, e_n)$ del vector de error e puede hacerse cero mediante una sustitución hacia atrás. La elección de la x_i no hace ninguna diferencia para la parte inferior del vector de error; resulta claro que $(e_{n+1}, \ldots, e_m) = (-d_{n+1}, \ldots, -d_m)$. Por lo tanto, la solución de mínimos cuadrados se reduce al mínimo al usar la x de la solución hacia atrás de la parte superior y el error de mínimos cuadrados es $||e||_2^2 = d_{n+1}^2 + \cdots + d_m^2$.

Mínimos cuadrados mediante la factorización QR

Dado el sistema indeterminado de $m \times n$

$$Ax = b,$$

encuentre la factorización completa QR $A = QR$ y establezca

$\hat{R}$ = submatriz superior de R de $n \times n$
$\hat{d}$ = n entradas superiores de $d = Q^T b$

Resuelva $\hat{R}\bar{x} = \hat{d}$ y obtenga la solución por mínimos cuadrados $\bar{x}$.

► **EJEMPLO 4.14** Use la factorización completa QR para resolver el problema de mínimos cuadrados

$$\begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -3 \\ 15 \\ 9 \end{bmatrix}.$$

Es necesario resolver $Rx = Q^T b$, o

$$\begin{bmatrix} 3 & 2 \\ 0 & 5 \\ \hdashline 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{15} \begin{bmatrix} 5 & 10 & 10 \\ -14 & 5 & 2 \\ 2 & 10 & -11 \end{bmatrix} \begin{bmatrix} -3 \\ 15 \\ 9 \end{bmatrix} = \begin{bmatrix} 15 \\ 9 \\ \hdashline 3 \end{bmatrix}.$$

El error de mínimos cuadrados será $||e||_2 = ||(0, 0, 3)||_2 = 3$. Al igualar las partes superiores se obtiene

$$\begin{bmatrix} 3 & 2 \\ 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 15 \\ 9 \end{bmatrix},$$

cuya solución es $\bar{x}_1 = 3.8, \bar{x}_2 = 1.8$. Este problema de mínimos cuadrados se resolvió mediante las ecuaciones normales en el ejemplo 4.2. ◄

Por último, se regresa al problema del ejemplo 4.5 que condujo a un sistema de ecuaciones normales mal condicionado.

ANOTACIÓN **Condicionamiento** En el capítulo 2, se encontró que la mejor manera de manejar problemas mal condicionados es evitarlos. El ejemplo 4.15 es un ejemplo clásico donde se sigue ese consejo. Debido a que las ecuaciones normales del ejemplo 4.5 están mal condicionadas, el método QR resuelve por mínimos cuadrados sin construir $A^T A$.

► EJEMPLO 4.15 Use la factorización completa QR para resolver el problema de mínimos cuadrados del ejemplo 4.5.

Las ecuaciones normales tuvieron muy poco éxito en la solución de este problema de mínimos cuadrados de 11 ecuaciones con 8 incógnitas. Se usará el comando `qr` de MATLAB para aplicar un enfoque alternativo:

```
>> x=(2+(0:10)/5)';
>> y=1+x+x.^2+x.^3+x.^4+x.^5+x.^6+x.^7;
>> A=[x.^0 x x.^2 x.^3 x.^4 x.^5 x.^6 x.^7];
>> [Q,R]=qr(A);
>> b=Q'*y;
>> c=R(1:8,1:8)\b(1:8)

c=
   0.99999991014308
   1.00000021004107
   0.99999979186557
   1.00000011342980
   0.99999996325039
   1.00000000708455
   0.99999999924685
   1.00000000003409
```

La solución tiene seis cifras decimales correctas $c = [1, \ldots, 1]$ empleando la factorización QR. Este enfoque encuentra la solución de mínimos cuadrados sin la formación de ecuaciones normales, que tienen un número de condición aproximado de 10^{19}. ◄

4.3.2 Ortogonalización de Gram-Schmidt modificado

Una ligera modificación a Gram-Schmidt resulta en una mejora de su exactitud en los cálculos en computadora. El nuevo algoritmo llamado Gram-Schmidt modificado es matemáticamente equivalente al algoritmo de Gram-Schmidt original, o "clásico".

Ortogonalización de Gram-Schmidt modificado

Sean $A_j, j = 1, \ldots, n$ vectores linealmente independientes.

for $j = 1, 2, \ldots, n$
  $y = A_j$
  for $i = 1, 2, \ldots, j-1$
    $r_{ij} = q_i^T y$
    $y = y - r_{ij} q_i$
  end
  $r_{jj} = \|y\|_2$
  $q_j = y / r_{jj}$
end

La única diferencia con el Gram-Schmidt clásico es que A_j se sustituye por y en el ciclo interior. Geométricamente hablando, cuando se proyecta la parte del vector A_j, en la dirección de q_2, por ejemplo, debe restarse la proyección de la y restante de A_j con la parte q_1 ya eliminada, en vez de la proyección de la propia A_j sobre q_2. Gram-Schmidt modificado es la versión que se empleará en el algoritmo GMRES de la sección 4.4.

► EJEMPLO 4.16 Compare los resultados de los algoritmos de Gram-Schmidt clásico y modificado, calculados en doble precisión, sobre la matriz de vectores casi paralelos

$$\begin{bmatrix} 1 & 1 & 1 \\ \delta & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & \delta \end{bmatrix}$$

donde $\delta = 10^{-10}$.

En primer lugar, se aplica el Gram-Schmidt clásico.

$$y_1 = A_1 = \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix} \quad \text{y} \quad q_1 = \frac{1}{\sqrt{1+\delta^2}} \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix}.$$

Observe que $\delta^2 = 10^{-20}$ es un número aceptable en doble precisión, pero $1 + \delta^2 = 1$ después del redondeo. Entonces

$$y_2 = \begin{bmatrix} 1 \\ 0 \\ \delta \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix} q_1^T A_2 = \begin{bmatrix} 1 \\ 0 \\ \delta \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\delta \\ \delta \\ 0 \end{bmatrix} \quad \text{y} \quad q_2 = \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$

después de dividir entre $||y_2||_2 = \sqrt{\delta^2 + \delta^2} = \sqrt{2}\delta$. Se completa el Gram-Schmidt clásico,

$$y_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \delta \end{bmatrix} - \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix} q_1^T A_3 - \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} q_2^T A_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \delta \end{bmatrix} - \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\delta \\ 0 \\ \delta \end{bmatrix} \quad \text{y} \quad q_3 = \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Desafortunadamente, debido al redondeo en doble precisión, hecho en el primer paso, q_2 y q_3 resultan ser no ortogonales:

$$q_2^T q_3 = \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}^T \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{2}.$$

Por otro lado, el Gram-Schmidt modificado se comporta mucho mejor. Mientras que q_1 y q_2 se calculan de la misma manera, se determina que q^3 es

$$y_3^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \delta \end{bmatrix} - \begin{bmatrix} 1 \\ \delta \\ 0 \\ 0 \end{bmatrix} q_1^T A_3 = \begin{bmatrix} 0 \\ -\delta \\ 0 \\ \delta \end{bmatrix},$$

$$y_3 = y_3^1 - \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} q_2^T y_3^1 = \begin{bmatrix} 0 \\ -\delta \\ 0 \\ \delta \end{bmatrix} - \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \frac{\delta}{\sqrt{2}}$$

$$= \begin{bmatrix} 0 \\ -\frac{\delta}{2} \\ -\frac{\delta}{2} \\ \delta \end{bmatrix} \quad \text{y} \quad q_3 = \begin{bmatrix} 0 \\ -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \end{bmatrix}.$$

Ahora $q_2^T q_3 = 0$ tal como se deseaba. Observe que tanto para el Gram-Schmidt clásico, como para el modificado, $q_1^T q_2$ está en el orden de δ, por lo que incluso el Gram-Schmidt modificado tiene un margen para mejorar. La ortogonalización mediante reflectores de Householder, que se describe en la siguiente sección, se utiliza mucho por ser más estable computacionalmente. ◄

4.3.3 Reflectores de Householder

Aunque el método de ortogonalización de Gram-Schmidt modificado es una forma mejorada para calcular la factorización QR de una matriz, no es la mejor manera. Un método alternativo que utiliza los reflectores de Householder requiere menos operaciones y es más estable, en el sentido de minimizar los errores de redondeo. En esta sección, se definirán los reflectores y se mostrará cómo se utilizan para factorizar una matriz.

Un reflector de Householder es una matriz ortogonal que refleja todos los m vectores a través de un plano con $m - 1$ dimensiones. Esto significa que la longitud de cada vector no se modifica al multiplicarlo por la matriz, por lo que los reflectores de Householder son ideales para mover vectores. Dado un vector x que se desea relocalizar hacia un vector w de igual longitud, los reflectores de Householder dan una matriz H de tal forma que $Hx = w$.

El origen queda claro en la figura 4.11. Dibuje el plano de $m - 1$ dimensiones bisectando a x y w, y perpendicular al vector que los conecta. Después refleje todos los vectores a través del plano.

LEMA 4.3 Suponga que x y w son vectores con la misma longitud euclidiana, $||x||_2 = ||w||_2$. Entonces $w - x$ y $w + x$ son perpendiculares. ■

Demostración. $(w - x)^T(w + x) = w^T w - x^T w + w^T x - x^T x = ||w||^2 - ||x||^2 = 0$. ❐

Defina el vector $v = w - x$, y considere la matriz de proyección

$$P = \frac{vv^T}{v^T v}. \tag{4.29}$$

Una **matriz de proyección** es una matriz que satisface $P^2 = P$. En el ejercicio 13 se pide al lector verificar que P en (4.29) es una matriz de proyección simétrica y que $Pv = v$. Geométricamente, para cualquier vector u, Pu es la proyección de u sobre v. En la figura 4.11 hay indicios de que si se resta el doble de la proyección Px de x, debe obtenerse w. Para comprobar esto, establezca $H = I - 2P$. Entonces

$$\begin{aligned} Hx &= x - 2Px \\ &= w - v - \frac{2vv^T x}{v^T v} \\ &= w - v - \frac{vv^T x}{v^T v} - \frac{vv^T(w - v)}{v^T v} \\ &= w - \frac{vv^T(w + x)}{v^T v} \\ &= w, \end{aligned} \tag{4.30}$$

la ultima igualdad que sigue del lema 4.3, significa que $w + x$ es ortogonal a $v = w - x$.

La matriz H se llama un **reflector de Householder**. Observe que H es una matriz simétrica (ejercicio 14) y ortogonal, dado que

$$\begin{aligned} H^T H = HH &= (I - 2P)(I - 2P) \\ &= I - 4P + 4P^2 \\ &= I. \end{aligned}$$

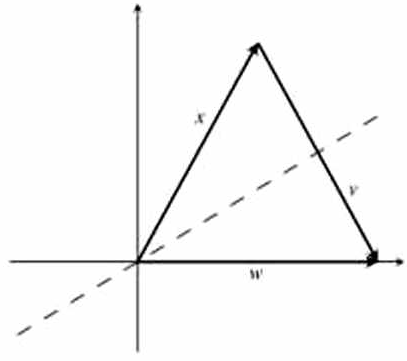


Figura 4.11 Reflector de Householder. Dados los vectores x y w de igual longitud, la reflexión a través de la bisectriz del ángulo entre ellos (línea punteada) los intercambia.

Estos hechos se resumen en el siguiente teorema:

TEOREMA 4.4 **Reflectores de Householder.** Sean x y w vectores con $||x||_2 = ||w||_2$ y defina $v = w - x$. Entonces $H = I - 2w^T/v^T v$ es una matriz simétrica ortogonal y $Hx = w$. ■

► EJEMPLO 4.17 Sea $x = [3, 4]$ y $w = [5, 0]$. Encuentre un reflector de Householder H que satisfaga $Hx = w$.

Establezca

$$v = w - x = \begin{bmatrix} 5 \\ 0 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 2 \\ -4 \end{bmatrix},$$

y defina la matriz de proyección

$$P = \frac{vv^T}{v^T v} = \frac{1}{20}\begin{bmatrix} 4 & -8 \\ -8 & 16 \end{bmatrix} = \begin{bmatrix} 0.2 & -0.4 \\ -0.4 & 0.8 \end{bmatrix}.$$

Entonces

$$H = I - 2P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.4 & -0.8 \\ -0.8 & 1.6 \end{bmatrix} = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}.$$

Verifique que H mueve a x hacia w y viceversa:

$$Hx = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix} = w$$

y

$$Hw = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}\begin{bmatrix} 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} = x.$$

◄

Como una primera aplicación de los reflectores de Householder, se desarrollará una nueva forma de realizar la factorización QR. En el capítulo 12 se aplica Householder al problema de los valores propios o característicos, para permitir matrices en la forma Hessenberg superior. En ambas aplicaciones, se emplearán los reflectores para un solo propósito: mover un vector columna x hacia los ejes de coordenadas como una forma de colocar ceros en una matriz.

Se inicia con una matriz A que debe escribirse en la forma $A = QR$. Sea x_1 la primera columna de A. Sea $w = \pm(||x_1||_2, 0, \ldots, 0)$ un vector a lo largo del primer eje de coordenadas de idéntica longitud euclidiana. (En teoría, cualquier signo funciona. Para la estabilidad numérica, con frecuencia el signo se elige como el opuesto del signo de la primera componente de x para evitar la posibilidad de una resta de números casi iguales al formarse v). Forme el reflector de Householder H_1 de manera que $H_1 x = w$. En el caso de 4×3, la multiplicación de H_1 por A resulta en

$$H_1 A = H_1 \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}.$$

Se han introducido algunos ceros en A. Se desea continuar de esta manera hasta que A se convierta en una triangular superior, entonces se tendrá R en la factorización QR. Encuentre el reflector de Householder $\hat{H}_2$ que mueve el vector x_2 de $(m-1)$, que contiene las $m-1$ entradas inferiores en la columna 2 de $H_1 A$ a $\pm(\|x_2\|_2, 0, \ldots, 0)$. Donde $\hat{H}_2$ es una matriz de $(m-1) \times (m-1)$, se defina H_2 como la matriz de $m \times m$ formada al colocar $\hat{H}_2$ en la parte inferior de la matriz identidad. Entonces

$$\left(\begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & & & \\ 0 & & \hat{H}_2 & \\ 0 & & & \end{array}\right)\left(\begin{array}{c|ccc} \times & \times & \times & \\ \hline 0 & \times & \times & \\ 0 & \times & \times & \\ 0 & \times & \times & \end{array}\right) = \left(\begin{array}{c|cc} \times & \times & \times \\ \hline 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{array}\right)$$

El resultado $H_2 H_1 A$ está a un paso de la triangulación superior. Un paso adicional da

$$\left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & & \\ 0 & 0 & & \hat{H}_3 \end{array}\right)\left(\begin{array}{cc|c} \times & \times & \times \\ 0 & \times & \times \\ \hline 0 & 0 & \times \\ 0 & 0 & \times \end{array}\right) = \left(\begin{array}{cc|c} \times & \times & \times \\ 0 & \times & \times \\ \hline 0 & 0 & \times \\ 0 & 0 & 0 \end{array}\right)$$

y el resultado

$$H_3 H_2 H_1 A = R,$$

es una matriz triangular superior. Al multiplicar sobre la izquierda por los inversos de los reflectores de Householder es posible reescribir el resultado como

$$A = H_1 H_2 H_3 R = QR,$$

donde $Q = H_1 H_2 H_3$. Tenga en cuenta que $H_i^{-1} = H_i$ supuesto que H_i es simétrica ortogonal. El problema de computadora 3 pide al lector que escriba el código para la factorización mediante los reflectores de Householder.

► **EJEMPLO 4.18** Use los reflectores de Householder para encontrar la factorización QR de

$$A = \begin{bmatrix} 3 & 1 \\ 4 & 3 \end{bmatrix}.$$

Es necesario encontrar un reflector de Householder que mueva la primera columna $[3, 4]$ sobre el eje x. Tal reflector H_1 se encontró en el ejemplo 4.17, y

$$H_1 A = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 0 & -1 \end{bmatrix}.$$

Al multiplicar ambos lados desde la izquierda por $H_1^{-1} = H_1$ se obtiene

$$A = \begin{bmatrix} 3 & 1 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}\begin{bmatrix} 5 & 3 \\ 0 & -1 \end{bmatrix} = QR,$$

donde $Q = H_1^T = H_1$. ◄

► EJEMPLO 4.19 Use los reflectores de Householder para encontrar la factorización QR de $A = \begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix}$.

Debe encontrarse un reflector de Householder que mueva la primera columna $x = [1, 2, 2]$ hacia el vector $w = [||x||_2, 0, 0]$. Establezca $v = w - x = [3, 0, 0] - [1, 2, 2] = [2, -2, -2]$. En referencia al teorema 4.4, se tiene

$$H_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{2}{12}\begin{bmatrix} 4 & -4 & -4 \\ -4 & 4 & 4 \\ -4 & 4 & 4 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{2}{3} \\ \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

y

$$H_1 A = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{2}{3} \\ \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \end{bmatrix}\begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 0 & -3 \\ 0 & -4 \end{bmatrix}.$$

El paso restante es mover el vector $\hat{x} = [-3, -4]$ hacia $\hat{w} = [5, 0]$. El cálculo de $\hat{H}_2$ a partir del teorema 4.4 resulta en

$$\begin{bmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{bmatrix}\begin{bmatrix} -3 \\ -4 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix},$$

que conduce a

$$H_2 H_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.6 & -0.8 \\ 0 & -0.8 & 0.6 \end{bmatrix}\begin{bmatrix} \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{2}{3} \\ \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \end{bmatrix}\begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 0 & 5 \\ 0 & 0 \end{bmatrix} = R.$$

Al multiplicar ambos lados desde la izquierda por $H_1^{-1} H_2^{-1} = H_1 H_2$ se obtiene la factorización QR

$$\begin{bmatrix} 1 & -4 \\ 2 & 3 \\ 2 & 2 \end{bmatrix} = H_1 H_2 R = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{2}{3} \\ \frac{2}{3} & -\frac{2}{3} & \frac{1}{3} \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.6 & -0.8 \\ 0 & -0.8 & 0.6 \end{bmatrix}\begin{bmatrix} 3 & 2 \\ 0 & 5 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1/3 & -14/15 & -2/15 \\ 2/3 & 1/3 & -2/3 \\ 2/3 & 2/15 & 11/15 \end{bmatrix}\begin{bmatrix} 3 & 2 \\ 0 & 5 \\ 0 & 0 \end{bmatrix} = QR.$$

Compare este resultado con la factorización mediante la ortogonalización de Gram-Schmidt del ejemplo 4.13. ◄

La factorización QR no es única para una determinada matriz A de $m \times n$. Por ejemplo, defina $D = \text{diag}(d_1, \ldots, d_m)$, donde cada d_i es $+1$ o bien -1. Entonces $A = QR = QDDR$, y se verifica que QD es ortogonal y que DR es triangular superior.

En el ejercicio 12 se pide un conteo de operaciones de la factorización QR mediante reflectores de Householder, que resulta ser de $(2/3)m^3$ multiplicaciones y el mismo número de sumas (una menor complejidad que la de la ortogonalización de Gram-Schmidt). Además, el método de Householder es conocido por ofrecer una mejor ortogonalidad en los vectores unitarios y por tener menores requisitos de memoria. Por estas razones, es el método más elegido para la factorización QR de matrices típicas.

4.3 Ejercicios

1. Aplique la ortogonalización de Gram-Schmidt clásica para encontrar la factorización QR completa de las siguientes matrices:

$$\text{(a)} \begin{bmatrix} 4 & 0 \\ 3 & 1 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \quad \text{(c)} \begin{bmatrix} 2 & 1 \\ 1 & -1 \\ 2 & 1 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} 4 & 8 & 1 \\ 0 & 2 & -2 \\ 3 & 6 & 7 \end{bmatrix}$$

2. Aplique la ortogonalización de Gram-Schmidt clásica para encontrar la factorización QR completa de las siguientes matrices:

$$\text{(a)} \begin{bmatrix} 2 & 3 \\ -2 & -6 \\ 1 & 0 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} -4 & -4 \\ -2 & 7 \\ 4 & -5 \end{bmatrix}$$

3. Aplique la ortogonalización de Gram-Schmidt modificada para encontrar la factorización QR completa de las matrices del ejercicio 1.

4. Aplique la ortogonalización de Gram-Schmidt modificada para encontrar la factorización QR completa de las matrices del ejercicio 2.

5. Aplique los reflectores de Householder para encontrar la factorización QR completa de las matrices del ejercicio 1.

6. Aplique los reflectores de Householder para encontrar la factorización QR completa de las matrices del ejercicio 2.

7. Use la factorización QR del ejercicio 2, 4 o 6 para resolver el problema de mínimos cuadrados.

$$\text{(a)} \begin{bmatrix} 2 & 3 \\ -2 & -6 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \\ 6 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} -4 & -4 \\ -2 & 7 \\ 4 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 9 \\ 0 \end{bmatrix}$$

8. Encuentre la factorización QR y úsela para resolver el problema de mínimos cuadrados.

$$\text{(a)} \begin{bmatrix} 1 & 4 \\ -1 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ -3 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 2 & 4 \\ 0 & -1 \\ 2 & -1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

9. Demuestre que una matriz cuadrada es ortogonal si y sólo si sus columnas son vectores unitarios ortogonales por parejas.

10. Demuestre que el producto de dos matrices ortogonales de $m \times m$ es de nuevo ortogonal.

11. Muestre que la ortogonalización de Gram-Schmidt de una matriz de $m \times m$ requiere aproximadamente m^3 multiplicaciones y m^3 sumas.

12. Muestre que el método de los reflectores de Householder para la factorización QR requiere aproximadamente $(2/3)m^3$ multiplicaciones y $(2/3)m^3$ sumas.

13. Sea P la matriz definida en (4.29). Muestre que (a) $P^2 = P$ (b) P es simétrica (c) $Pv = v$.

14. Demuestre que los reflectores de Householder son matrices simétricas.

15. Verifique que los métodos de Gram-Schmidt clásico y modificado son matemáticamente idénticos (en aritmética exacta).

4.3 Problemas de computadora

1. Escriba un programa en MATLAB que implemente el método de Gram-Schmidt clásico para encontrar la factorización QR reducida. Revise su trabajo comparando la factorizaciones de las matrices del ejercicio 1 con el comando `qr(A,0)` de MATLAB u otro equivalente. La factorización es única según los signos de las entradas de Q y R.
2. Repita el problema de computadora 1, pero implemente el método de Gram-Schmidt modificado.
3. Repita el problema de computadora 1, pero implemente los reflectores de Householder.
4. Escriba un programa en MATLAB que implemente el método de Gram-Schmidt (a) clásico y (b) modificado para encontrar la factorización QR completa. Revise su trabajo comparando las factorizaciones de las matrices del ejercicio 1 con el comando de MATLAB `qr(A)` u otro equivalente.
5. Use la factorización QR de MATLAB para encontrar las soluciones por mínimos cuadrados y el error de la norma 2 para los siguientes sistemas inconsistentes:

$$\text{(a)}\ \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 2 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 5 \\ 5 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 1 & 2 & 2 \\ 2 & -1 & 2 \\ 3 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \\ 10 \\ 3 \end{bmatrix}$$

6. Use la factorización QR de MATLAB para encontrar las soluciones por mínimos cuadrados y el error de la norma 2 para los siguientes sistemas inconsistentes:

$$\text{(a)}\ \begin{bmatrix} 3 & -1 & 2 \\ 4 & 1 & 0 \\ -3 & 2 & 1 \\ 1 & 1 & 5 \\ -2 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ -5 \\ 15 \\ 0 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 4 & 2 & 3 & 0 \\ -2 & 3 & -1 & 1 \\ 1 & 3 & -4 & 2 \\ 1 & 0 & 1 & -1 \\ 3 & 1 & 3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ 2 \\ 0 \\ 5 \end{bmatrix}$$

7. Sea A la matriz de $10 \times n$ formada por las primeras n columnas de la matriz de Hilbert de 10×10. Sea c el vector de n entradas $[1, \ldots, 1]$ y establezca $B = Ac$. Use la factorización QR para resolver el problema de mínimos cuadrados $Ax = b$ para (a) $n = 6$ (b) $n = 8$ y compárela con la solución correcta de mínimos cuadrados $\overline{x} = c$. ¿Cuántos decimales correctos pueden calcularse? Vea el problema de computadora 8 de la sección 4.1, donde se utilizan las ecuaciones normales.
8. Sean $x_1, \ldots, x_{11}$ 11 puntos espaciados uniformemente en $[2, 4]$ y $y_i = 1 + x_i + x_i^2 + \cdots + x_i^d$. Use la factorización QR para calcular el mejor polinomio de grado d, donde (a) $d = 5$ (b) $d = 6$ (c) $d = 8$. Compare su solución con la del ejemplo 4.5 y el problema de computadora 9 de la sección 4.1 ¿Cuántas posiciones decimales correctas de los coeficientes pueden calcularse?

4.4 Método del residuo mínimo generalizado (GMRES)

En el capítulo 2 se vio que el método del gradiente conjugado puede considerarse un método iterativo especialmente diseñado para resolver el sistema matricial $Ax = b$ para una matriz cuadrada simétrica A. Si A no es simétrica, la teoría del gradiente conjugado falla. Sin embargo, existen varias alternativas que funcionan para un problema no simétrico. Uno de los más populares es el método del residuo mínimo generalizado, o GMRES para abreviar. Este método es una buena opción para la solución de grandes sistemas dispersos, no simétricos, lineales $Ax = b$.

A primera vista, podría parecer extraño estar hablando de un método para resolver sistemas lineales en el capítulo de los mínimos cuadrados. ¿Por qué podría importar la ortogonalidad en un

problema que no tiene ninguna conexión aparente con ella? La respuesta está en el hecho, como se vio en el capítulo 2, de que las matrices con vectores columna casi paralelos tienden a estar mal condicionadas, lo que a su vez causa un gran aumento del error en la solución de $Ax = b$.

De hecho, la ortogonalización se integra en el GMRES de dos maneras. En primer lugar, el error hacia atrás del sistema se reduce al mínimo en cada paso de iteración si se usa una formulación de mínimos cuadrados. En segundo lugar, y de manera más sutil, la base del espacio de búsqueda se reortogonaliza en cada paso con el fin de evitar inexactitudes del mal condicionamiento. El GMRES es un ejemplo interesante de un método que aprovecha las ideas de la ortogonalidad en sitios donde no están presentes de manera evidente.

4.4.1 Métodos de Krylov

El GMRES es un miembro de la familia de los **métodos de Krylov**. Estos métodos se basan en el cálculo exacto del **espacio de Krylov**, que es el espacio vectorial generado por $\{r, Ar, \ldots, A^k r\}$, donde $r = b - Ax_0$ es el vector residual de la estimación inicial. Como los vectores $A^k r$ tienden hacia una dirección común para una k grande, la base para el espacio de Krylov debe calcularse con cuidado. La determinación de una base precisa para el espacio de Krylov requiere el uso de métodos como la ortogonalización de Gram-Schmidt o las reflexiones de Householder.

La idea detrás del GMRES es la búsqueda de mejoras en la estimación inicial x_0 en un espacio vectorial particular, el espacio de Krylov generado por la r residual y sus productos bajo la matriz no singular A. En el paso k del método, se amplía el espacio de Krylov añadiendo $A^k r$, se reortogonaliza la base y después se usan los mínimos cuadrados para encontrar la mejor manera para añadir a x_0.

Método del residuo mínimo generalizado (GMRES)

$x_0 =$ valor inicial
$r = b - Ax_0$
$q_1 = r/||r||_2$
for $k = 1, 2, \ldots, m$
  $y = Aq_k$
  for $j = 1, 2, \ldots, k$
    $h_{jk} = q_j^T y$
    $y = y - h_{jk} q_j$
  end
  $h_{k+1,k} = ||y||_2$ (Si $h_{k+1,k} = 0$,se omite la siguiente línea y termina en la parte inferior)
  $q_{k+1} = y/h_{k+1,k}$
  Disminuir al mínimo $||Hc_k - [||r||_2\ 0\ 0 \ldots\ 0]^T||_2$ para c_k
  $x_k = Q_k c_k + x_0$
end

Las iteraciones x_k son soluciones aproximadas al sistema $Ax = b$. En el paso k-ésimo del pseudocódigo, la matriz H es una matriz de $(k + 1) \times k$. El paso de minimización que produce el programa en lenguaje C es un problema de mínimos cuadrados de $k + 1$ ecuaciones con k incóg-

ANOTACIÓN

Ortogonalidad El GMRES es el primer ejemplo de un método de Krylov, que depende de un cálculo preciso del espacio de Krylov. En el capítulo 2 se encontró que los vectores columna casi paralelos de una matriz causan el mal condicionamiento. Los vectores $A^k r$ que definen el espacio de Krylov tienden a ser más paralelos a medida que k aumenta, por lo que el uso de las técnicas de ortogonalización de la sección 4.3 es esencial para construir algoritmos estables y eficientes como el GMRES.

nitas, que puede resolverse utilizando técnicas de este capítulo. La matriz Q_k en el código es de $n \times k$, y consiste en las k columnas ortonormales $q_1, \ldots, q_k$. Si $h_{k+1,k} = 0$, entonces el paso k es el paso final y la minimización llegará a la solución exacta de $Ax = b$.

Para aproximar el espacio, el enfoque más directo no es el mejor. En el capítulo 12 se explotará el hecho de que los vectores $A^k r$ tienden asintóticamente hacia la misma dirección para calcular valores propios. Con el fin de generar una base eficiente para el espacio de Krylov $\{r, Ar, \ldots, A^k r\}$, se confiará en la potencia de la ortogonalización de Gram-Schmidt como el enfoque más sencillo.

La aplicación del método de de Gram-Schmidt modificado a $\{r, Ar, \ldots, A^k r\}$, comenzando con $q_1 = r/||r||_2$, se realiza en el ciclo interior del pseudocódigo. Resulta en la igualdad matricial $AQ_k = Q_{k+1} H_k$, o bien

$$A \begin{bmatrix} & & & \\ q_1 & \cdots & q_k & \\ & & & \end{bmatrix} = \begin{bmatrix} & & & & \\ q_1 & \cdots & q_k & q_{k+1} & \\ & & & & \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2k} \\ & h_{32} & \cdots & h_{3k} \\ & & \ddots & \vdots \\ & & & h_{k+1,k} \end{bmatrix}$$

Aquí A es de $n \times n$, Q_k es de $n \times k$, y H_k es de $(k \times 1) \times k$. En la mayoría de los casos, k será mucho menor que n.

Las columnas de Q_k abarcan el espacio k-dimensional de Krylov, que se buscará para las x_{ad} añadidas a la aproximación inicial x_0. Los vectores en este espacio se escriben como $x_{ad} = Q_k c$. Para disminuir al mínimo el residuo

$$b - A(x_0 + x_{ad}) = r - Ax_{ad},$$

del problema original $Ax = b$ significa encontrar la c que minimiza

$$||Ax_{ad} - r||_2 = ||AQ_k c - r||_2 = ||Q_{k+1} H_k c - r||_2 = ||H_k c - Q^T_{k+1} r||_2,$$

donde la última igualdad cumple con la norma de las columnas ortonormales. Observe que $Q^T_{k+1} r = [||r||_2\, 0\, 0 \ldots 0]^T$, puesto que $q_1 = r/||r||_2$ como se señaló con anterioridad, y todas las columnas excepto la primera de Q_{k+1} son ortogonales a r. Ahora, el problema de mínimos cuadrados es

$$\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2k} \\ & h_{32} & \cdots & h_{3k} \\ & & \ddots & \vdots \\ & & & h_{k+1,k} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \\ \vdots \\ \\ c_k \end{bmatrix} = \begin{bmatrix} ||r||_2 \\ 0 \\ \\ \vdots \\ \\ 0 \end{bmatrix}.$$

Usando el programa en c se obtiene la solución aproximada del c-ésimo paso $x_k = x_0 + x_{ad} = x_0 + Q_k c$ para el problema original $Ax = b$.

En el GMRES, es importante tener en cuenta los tamaños respectivos de los subproblemas. La parte del algoritmo con mayor complejidad computacional es el cálculo por mínimos cuadrados, que disminuye al mínimo el error de $k + 1$ ecuaciones con k incógnitas. El tamaño k será pequeño en comparación con el tamaño total del problema n en la mayoría de las aplicaciones. En el caso especial cuando $h_{k+1,k} = 0$, el problema de mínimos cuadrados se vuelve cuadrado y la solución aproximada x_k es exacta.

Una característica conveniente del GMRES es que el error hacia atrás $||b - Ax_k||_2$ disminuye de forma monótona con k. La razón es clara por el hecho de que el problema de mínimos cuadrados en el paso k minimiza $||r - Ax_{ad}||_2$ para x_{ad} para x_{ad} en el espacio k-dimensional de Krylov. A medida que el GMRES procede, el espacio de Krylov se amplía, por lo que la siguiente aproximación no puede ser peor.

En cuanto al pseudocódigo presentado para el GMRES, vale la pena mencionar otros detalles de su implementación. En primer lugar, cabe destacar que el paso de minimización por mínimos cuadrados sólo se justifica cuando se requiere una solución aproximada x_k. Por lo tanto, puede realizarse sólo en forma intermitente, con el fin de monitorear el progreso hacia la solución, o en el caso extremo, el cálculo por mínimos cuadrados puede sacarse del ciclo y realizarse sólo al final, puesto que $x_{ad} = Q_k c$ no depende de los cálculos anteriores por mínimos cuadrados. Esto corresponde a mover el enunciado final **end** hacia arriba de las dos líneas anteriores. En segundo lugar, el paso de la ortogonalización de Gram-Schmidt que se lleva a cabo en el ciclo interior puede sustituirse por la ortogonalización de Householder con un aumento ligero de la complejidad computacional, si el condicionamiento es un asunto significativo.

El uso típico del GMRES es en una matriz A grande y dispersa de $n \times n$. En teoría, el algoritmo termina después de n pasos en la solución x siempre que A no sea singular. Sin embargo, en la mayoría de los casos, el objetivo es ejecutar el método durante k pasos, donde k es mucho menor que n. Tenga en cuenta que la matriz Q_k es de $n \times k$ y no se garantiza que sea dispersa. Por lo tanto, las consideraciones de memoria también pueden limitar el número k de pasos del GMRES.

Estas condiciones conducen a una variación del algoritmo conocida como **GMRES reiniciado**. Si no se alcanza un progreso suficiente hacia la solución después de k iteraciones y si la matriz Q_k de $n \times k$ se vuelve demasiado grande como para ser manejada, la idea es simple: eliminar Q_k y empezar el GMRES desde el principio, *utilizando la mejor estimación actual* x_k como la nueva x_0.

4.4.2 GMRES precondicionado

El concepto detrás del precondicionamiento del GMRES es muy similar al caso del gradiente conjugado. Comience con un sistema lineal no simétrico $Ax = b$. Una vez más, trate de resolver $M^{-1}Ax = M^{-1}b$, donde M es uno de los precondicionadores analizados en la sección 2.

Se requieren muy pocos cambios en el pseudocódigo en el lenguaje del GMRES de la sección anterior. En la versión precondicionada, el residuo de partida es $r = M^{-1}(b - Ax_0)$. El paso de la iteración del espacio de Krylov se cambia a $w = M^{-1}Aq_k$. Observe que ninguno de estos pasos requiere la formación explícita de M^{-1}. Deben llevarse a cabo por sustitución hacia atrás, suponiendo que M está en una forma simple o factorizada. Con estos cambios, el algoritmo es el siguiente.

GMRES precondicionado

$x_0 =$ estimación inicial
$r = M^{-1}(b - Ax_0)$
$q_1 = r/||r||_2$
for $k = 1, 2, \ldots, m$
  $w = M^{-1}Aq_k$
  for $j = 1, 2, \ldots, k$
    $h_{jk} = w^T q_j$
    $w = w - h_{jk}q_j$
  end
  $h_{k+1,k} = ||w||_2$
  $q_{k+1} = w/h_{k+1,k}$
  Disminuir al mínimo $||Hc_k - [||r||_2\ 0\ 0 \ldots\ 0]^T||_2$ para c_k
  $x_k = Qc_k + x_0$
end

► **EJEMPLO 4.20** Sea A la matriz con entradas diagonales $A_{ii} = \sqrt{i}$ para $i = 1, \ldots, n$ y $A_{i,i+10} = \cos i$, $A_{i+10,i} = \text{sen } i$ para $i = 1, \ldots, n - 10$, con todas las otras entradas iguales a cero. Establezca x como el vector de n unos y defina $Ax = b$. Para $n = 500$, resuelva $Ax = b$ con GMRES de tres maneras: sin usar precondicionadores, utilizando el precondicionador de Jacobi y empleando el precondicionador de Gauss-Seidel.

La matriz puede definirse en MATLAB como

```
A=diag(sqrt(1:n))+diag(cos(1:(n-10)),10)
                 +diag(sin(1:(n-10)),-10).
```

En la figura 4.12 se muestran los tres resultados diferentes. El GMRES converge lentamente sin precondicionamiento. El precondicionador de Jacobi hace una mejora significativa y el GMRES con el precondicionador de Gauss-Seidel sólo requiere alrededor de 10 pasos para alcanzar la precisión de máquina.

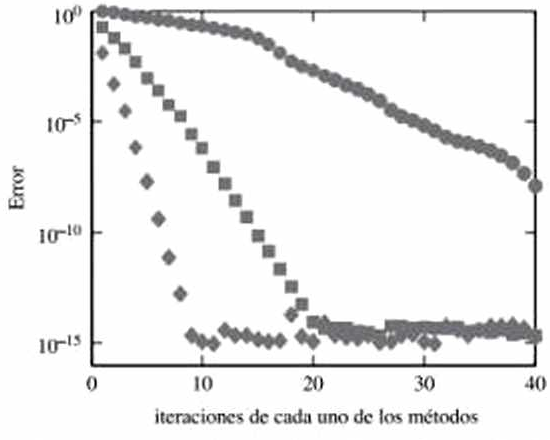


Figura 4.12 Eficiencia del método GMRES precondicionado para la solución del ejemplo 4.20. El error se grafica contra el número de iteraciones. Círculos: sin precondicionador. Cuadros: precondicionador de Jacobi. Rombos: precondicionador de Gauss-Seidel.

4.4 Ejercicios

1. Resuelva $Ax = b$ para las siguientes A y $b = [1, 0, 0]^T$, empleando el GMRES con $x_0 = [0, 0, 0]^T$. Reporte todas las aproximaciones x_k incluyendo la solución correcta.

$$(a)\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (b)\begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \quad (c)\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

2. Repita el ejercicio 1 con $b = [0, 0, 1]^T$.

3. Sea $A = \begin{bmatrix} 1 & 0 & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$. Demuestre que para cualquier x_0 y b, el GMRES converge a la solución exacta después de dos pasos.

4. Generalice el ejercicio 3, mostrando que para $A = \left[\begin{array}{c:c} I & C \\ \hdashline 0 & I \end{array}\right]$ y cualesquiera x_0 y b, el GMRES converge a la solución exacta después de dos pasos. Aquí C es una submatriz de $m_1 \times m_2$, 0 denota la matriz de ceros de $m_2 \times m_1$, e I denota la matriz identidad del tamaño apropiado.

4.4 Problemas de computadora

1. Sea A la matriz de $n \times n$ con $n = 1000$ y las entradas
$A(i,i) = i$, $A(i,i+1) = A(i+1,i) = 1/2$, $A(i,i+2) = A(i+2,i) = 1/2$ para toda i que se

ajuste dentro de la matriz. (a) Imprima la estructura distinta de cero `spy(A)`. (b) Sea x_e el vector de n unos. Establezca $b = Ax_e$ y aplique el método del gradiente conjugado, sin precondicionador, con el precondicionador de Jacobi y con el precondicionador de Gauss-Seidel. Compare los errores de las tres corridas en una gráfica contra las iteraciones.

2. Sea $n = 1000$. Comience con la matriz A de $n \times n$ del problema de computadora 1 y añada las entradas distintas de cero $A(i, 2i) = A(2i, i) = 1/2$ para $1 \leq i \leq n/2$. Realice los pasos (a) y (b) como en el problema 1.

3. Sea $n = 500$ y A la matriz de $n \times n$ con entradas $A(i,i) = 2, A(i,i+2) = A(i+2,i) = 1/2, A(i,i+4) = A(i+4,i) = 1/2$ para toda i y $A(500, i) = A(i, 500) = -0.1$ para $1 \leq i \leq 495$. Realice los pasos (a) y (b) como en el problema de computadora 1.

4. Sea A la matriz del problema de computadora 3, pero con los elementos de la diagonal sustituidos por $A(i,i) = \sqrt[3]{i}$. Realice los pasos (a) y (b) como en el problema 3.

5. Sea C el bloque matricial de 195×195 con $C(i,i) = 2, C(i,i+3) = C(i+3,i) = 0.1, C(i,i+39) = C(i+39,i) = 1/2, C(i,i+42) = C(i+42,i) = 1/2$ para toda i. Defina A como la matriz de $n \times n$ con $n = 780$, formada por cuatro bloques C dispuestos en diagonal, y con bloques de $\frac{1}{2}C$ en la superdiagonal y en la subdiagonal. Realice los pasos (a) y (b) como en el problema de computadora 1 para resolver $Ax = b$.

4.5 MÍNIMOS CUADRADOS NO LINEALES

La solución por mínimos cuadrados de un sistema de ecuaciones lineales $Ax = b$ disminuye al mínimo la norma euclidiana del residuo $||Ax - b||_2$. Se han aprendido dos métodos para encontrar la solución $\bar{x}$, basado uno en las ecuaciones normales y otro en la factorización QR.

Ninguno de estos métodos puede aplicarse si las ecuaciones son no lineales. En esta sección se desarrolla el método de Gauss-Newton para resolver problemas no lineales de mínimos cuadrados. Además de ilustrar el uso del método para resolver los problemas de intersección de círculos, se aplica Gauss-Newton para el ajuste de datos a modelos con coeficientes no lineales.

4.5.1 Método de Gauss-Newton

Considere el sistema de m ecuaciones con n incógnitas

$$\begin{aligned} r_1(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ r_m(x_1, \ldots, x_n) &= 0. \end{aligned} \tag{4.31}$$

La suma de los cuadrados de los errores está representada por la función

$$E(x_1, \ldots, x_n) = \frac{1}{2}(r_1^2 + \cdots + r_m^2) = \frac{1}{2} r^T r,$$

donde $r = [r_1, \ldots, r_m]^T$. La constante 1/2 se ha incluido en la definición para simplificar las fórmulas posteriores. Para minimizar E, se establece el gradiente $F(x) = \nabla E(x)$ como cero:

$$0 = F(x) = \nabla E(x) = \nabla\left(\frac{1}{2} r(x)^T r(x)\right) = r(x)^T Dr(x). \tag{4.32}$$

Observe que se ha usado la regla del producto punto para el gradiente (vea el apéndice A).

Se inicia por recordar el método de Newton multivariado, para después aplicarlo a la función vista como un vector columna $F(x)^T = (r^T Dr)^T = (Dr)^T r$. Puede aplicarse la regla del producto de matrices/vectores (vea el apéndice A) para obtener

$$DF(x)^T = D((Dr)^T r) = (Dr)^T \cdot Dr + \sum_{i=1}^{m} r_i Dc_i,$$

donde c_i es la i-ésima columna de Dr. Observe que $Dc_i = H_{r_i}$, la matriz de las segundas derivadas parciales, o **Hessiana**, de r_i:

$$H_{r_i} = \begin{bmatrix} \frac{\partial^2 r_i}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 r_i}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 r_i}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 r_i}{\partial x_n \partial x_n} \end{bmatrix}.$$

La aplicación del método de Newton puede simplificarse eliminando algunos de los términos. Sin la sumatoria superior de m términos, se tiene lo siguiente.

Método de Gauss-Newton

Para disminuir al mínimo

$$r_1(x)^2 + \cdots + r_m(x)^2.$$

Establezca x_0 = vector inicial,
for $k = 0, 1, 2, \ldots$

$$A = Dr(x^k) \tag{4.33}$$

$$A^T A v^k = -A^T r(x^k)$$

$$x^{k+1} = x^k + v^k \tag{4.34}$$

end

Observe que cada paso del método de Gauss-Newton es una reminiscencia de las ecuaciones normales, donde la matriz de coeficientes ha sido reemplazada por Dr. El método de Gauss-Newton resuelve para una raíz del gradiente del error cuadrático. Aunque el gradiente debe ser cero en el mínimo, lo inverso no es cierto, por lo que es posible que el método converja a un máximo o a un punto neutro. Debe tenerse cuidado en la interpretación del resultado del algoritmo.

Los tres ejemplos siguientes ilustran el uso del método de Gauss-Newton, así como el método de Newton multivariado del capítulo 2. Dos círculos secantes se cortan en uno o dos puntos, a menos que los círculos coincidan. Sin embargo, tres círculos en el plano por lo general no tienen puntos de intersección común. En tal caso, puede pedirse el punto en el plano que se aproxime más a ser un punto de intersección en el sentido de los mínimos cuadrados. Para tres círculos, éste es un problema de tres ecuaciones lineales con dos incógnitas x, y.

En el ejemplo 4.21 se muestra cómo resuelve el método de Gauss-Newton este problema no lineal de mínimos cuadrados. En el ejemplo 4.22 se define el mejor punto de una manera diferente: se encuentra el único punto de intersección de los tres círculos, lo que permite que su radio pueda cambiarse por una cantidad común K. Se trata de un problema de tres ecuaciones con tres incógnitas x, y, K; no es un problema de mínimos cuadrados y se resuelve usando el método multivariado de Newton.

Por último, en el ejemplo 4.23 se añade un cuarto círculo. La solución de cuatro ecuaciones con tres incógnitas x, y, K es de nuevo un problema de mínimos cuadrados que requiere de Gauss-Newton. Esta última formulación es relevante para los cálculos en GPS, como se muestra en la comprobación en la realidad 4.

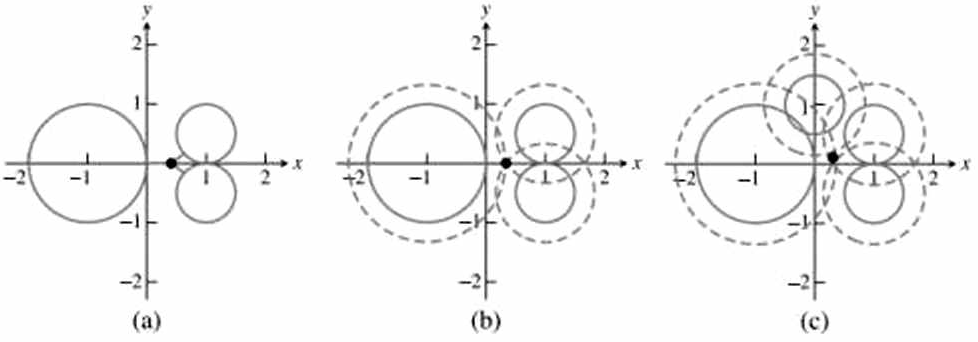


Figura 4.13 Puntos cercanos a la intersección de tres círculos. (a) Puntos cercanos a la intersección por mínimos cuadrados que se encuentran mediante el método de Gauss-Newton. (b) Al expandir los radios en una cantidad común se obtiene un tipo diferente de punto cercano a la intersección mediante el método de Newton multivariado. (c) Los cuatro círculos del ejemplo 4.23 con un punto de solución por mínimos cuadrados, encontrado mediante el método de Gauss-Newton.

► **EJEMPLO 4.21** Considere los tres círculos en el plano con centros $(x_1, y_1) = (-1, 0)$, $(x_2, y_2) = (1, 1/2)$, $(x_3, y_3) = (1, -1/2)$ y los radios $R_1 = 1$, $R_2 = 1/2$, $R_3 = 1/2$, respectivamente. Utilice el método de Gauss-Newton para encontrar el punto en el que se minimiza la suma de los cuadrados de las distancias a los tres círculos.

Los círculos se muestran en la figura 4.13(a). El punto (x, y) en cuestión minimiza la suma de los cuadrados de los errores residuales:

$$r_1(x, y) = \sqrt{(x - x_1)^2 + (y - y_1)^2} - R_1$$

$$r_2(x, y) = \sqrt{(x - x_2)^2 + (y - y_2)^2} - R_2$$

$$r_3(x, y) = \sqrt{(x - x_3)^2 + (y - y_3)^2} - R_3.$$

Esto se deduce del hecho de que la distancia de un punto (x, y) a un círculo con centro en (x_1, y_1) y radio R_1 es $|\sqrt{(x - x_1)^2 + (y - y_1)^2} - R_1|$ (vea el ejercicio 3). El jacobiano de $r(x, y)$ es

$$Dr(x, y) = \begin{bmatrix} \frac{x - x_1}{S_1} & \frac{y - y_1}{S_1} \\ \frac{x - x_2}{S_2} & \frac{y - y_2}{S_2} \\ \frac{x - x_3}{S_3} & \frac{y - y_3}{S_3} \end{bmatrix},$$

donde $S_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ para $i = 1, 2, 3$. La iteración de Gauss-Newton con vector inicial $(x^0, y^0) = (0, 0)$ converge a $(\bar{x}, \bar{y}) = (0.412891, 0)$ con seis posiciones decimales correctas después de siete pasos. ◄

Un problema relacionado con los tres círculos da un tipo diferente de respuesta. En lugar de buscar los puntos que más se asemejan a los puntos de intersección, es posible expandir (o reducir) los radios de los círculos en una cantidad común hasta que tengan una intersección común. Esto es equivalente a resolver el sistema

$$r_1(x, y, K) = \sqrt{(x - x_1)^2 + (y - y_1)^2} - (R_1 + K) = 0$$
$$r_2(x, y, K) = \sqrt{(x - x_2)^2 + (y - y_2)^2} - (R_2 + K) = 0$$
$$r_3(x, y, K) = \sqrt{(x - x_3)^2 + (y - y_3)^2} - (R_3 + K) = 0. \qquad (4.35)$$

El punto (x, y) identificado de esta manera suele ser diferente de la solución por mínimos cuadrados del ejemplo 4.21.

► **EJEMPLO 4.22** Resuelva el sistema (4.35) para (x, y, K), empleando los círculos del ejemplo 4.21.

El sistema consta de tres ecuaciones no lineales con tres incógnitas, si se invoca el método multivariado de Newton. El jacobiano es

$$Dr(x, y, K) = \begin{bmatrix} \frac{x-x_1}{S_1} & \frac{y-y_1}{S_1} & -1 \\ \frac{x-x_2}{S_2} & \frac{y-y_2}{S_2} & -1 \\ \frac{x-x_3}{S_3} & \frac{y-y_3}{S_3} & -1 \end{bmatrix}.$$

El método de Newton genera la solución $(x, y, K) = (1/3, 0, 1/3)$ en tres pasos. El punto de intersección $(1/3, 0)$ y los tres círculos con radios expandidos en $K = 1/3$ se presentan en la figura 4.13(b). ◄

Los ejemplos 4.21 y 4.22 muestran dos puntos de vista diferentes sobre el significado de "cercano al punto de intersección" para un grupo de círculos. En el ejemplo 4.23 se combinan los dos distintos enfoques.

► **EJEMPLO 4.23** Considere los cuatro círculos con centros $(-1, 0), (1, 1/2), (1, -1/2), (0, 1)$ y los radios respectivos de $1, 1/2, 1/2, 1/2$. Encuentre el punto (x, y) y la constante K para la cual la suma de los cuadrados de las distancias desde el punto a los cuatro círculos con radios aumentados en K (así, $1 + K$, $1/2 + K$, $1/2 + K$, $1/2 + K$, respectivamente) se minimiza.

Ésta es una combinación directa de los dos ejemplos anteriores. Existen cuatro ecuaciones con tres incógnitas x, y, K. Los residuos de mínimos cuadrados son similares a (4.35), pero con cuatro términos, y su jacobiano es

$$Dr(x, y, K) = \begin{bmatrix} \frac{x-x_1}{S_1} & \frac{y-y_1}{S_1} & -1 \\ \frac{x-x_2}{S_2} & \frac{y-y_2}{S_2} & -1 \\ \frac{x-x_3}{S_3} & \frac{y-y_3}{S_3} & -1 \\ \frac{x-x_4}{S_4} & \frac{y-y_4}{S_4} & -1 \end{bmatrix}.$$

El método de Gauss-Newton proporciona la solución $(\overline{x}, \overline{y}) = (0.311385, 0.112268)$ con $\overline{K} = 0.367164$, que se ilustra en la figura 4.13(c). ◄

El análogo del ejemplo 4.23 para esferas en tres dimensiones es la base matemática del sistema de posicionamiento global (GPS). Vea la comprobación en la realidad 4.

4.5.2 Modelos con parámetros no lineales

Una aplicación importante del método de Gauss-Newton es el ajuste de los modelos que no lineales en los coeficientes. Sean $(t_1, y_1), \ldots, (t_m, y_m)$ el conjunto de datos y $y = f_c(x)$ la función que debe

ajustarse, donde $c = [c_1, \ldots, c_p]$ es un conjunto de parámetros de se eligen para minimizar la suma de los cuadrados de los residuos

$$r_1(c) = f_c(t_1) - y_1$$
$$\vdots$$
$$r_m(c) = f_c(t_m) - y_m.$$

Este caso particular de (4.31) se ve con suficiente frecuencia como para justificar un tratamiento especial aquí.

Si los parámetros $c_1, \ldots, c_p$ entran al modelo de una manera lineal, entonces éste es un conjunto de ecuaciones lineales en los c_i y las ecuaciones normales, o la solución por la factorización QR, proporciona la elección óptima de los parámetros c. Si los parámetros c_i son no lineales en el modelo, el mismo tratamiento resulta en un sistema de ecuaciones que es no lineal en los c_i. Por ejemplo, el ajuste del modelo $y = c_1 t^{c_2}$ a los puntos de datos (t_i, y_i) produce las ecuaciones no lineales

$$y_1 = c_1 t_1^{c_2}$$
$$y_2 = c_1 t_2^{c_2}$$
$$\vdots$$
$$y_m = c_1 t_m^{c_2}.$$

Debido a que c_2 entra al modelo en forma no lineal, el sistema de ecuaciones no puede expresarse en forma matricial.

En la sección 4.2 se manejó esta dificultad cambiando el problema: se "linealizó el modelo" aplicando escalas logarítmicas en ambos lados del modelo y minimizando el error en la transformación logarítmicamente por mínimos cuadrados. En los casos donde las coordenadas transformadas logarítmicamente son en realidad las coordenadas adecuadas en las que se minimiza el error, esto es lo apropiado.

Sin embargo, para resolver el problema de mínimos cuadrados original, se regresa al método de Gauss-Newton. Se usa para reducir al mínimo la función de error E como una función del vector de parámetros c. La matriz Dr es la matriz de derivadas parciales de los errores r_i con respecto a los parámetros c_j, que son

$$(Dr)_{ij} = \frac{\partial r_i}{\partial c_j} = f_{c_j}(t_i).$$

Con esta información, es posible implementar el método de Gauss-Newton (4.33).

► EJEMPLO 4.24 Use el método de Gauss-Newton para ajustar los datos de la oferta mundial de automóviles del ejemplo 4.8 a un modelo exponencial (no linealizado).

Encontrar el mejor ajuste de los datos a un modelo exponencial por mínimos cuadrados significa encontrar c_1, c_2, que minimicen la RMEC para los errores $r_i = c_1 e^{c_2 t_i} - y_i$, $i = 1, \ldots, m$. Al usar la linealización del modelo en la sección anterior se ha minimizado la RMEC para los errores del modelo logarítmico $\ln y_i - (\ln c_1 - c_2 t_i)$. Los valores de c_i que minimizan la RMEC en los dos sentidos, por lo general son distintos.

Para calcular el mejor ajuste de mínimos cuadrados por el método de Gauss-Newton, defina

$$r = \begin{bmatrix} c_1 e^{c_2 t_1} - y_1 \\ \vdots \\ c_1 e^{c_2 t_m} - y_m \end{bmatrix},$$

y aplique derivadas con respecto a los parámetros c_1 y c_2 para obtener

$$Dr = -\begin{bmatrix} e^{c_2 t_1} & c_1 t_1 e^{c_2 t_1} \\ \vdots & \vdots \\ e^{c_2 t_m} & c_1 t_m e^{c_2 t_m} \end{bmatrix}.$$

ANOTACIÓN

Convergencia La no linealidad en los problemas de mínimos cuadrados ocasiona desafíos adicionales. Las ecuaciones normales y el enfoque QR encuentran la solución única, siempre y cuando la matriz A de coeficientes tenga un rango completo. Por otro lado, la iteración de Gauss-Newton aplicada a un problema no lineal puede converger a uno de varios mínimos relativos diferentes del error de mínimos cuadrados. El uso de una aproximación razonable para el vector inicial, si ésta existe, agrega convergencia al mínimo absoluto.

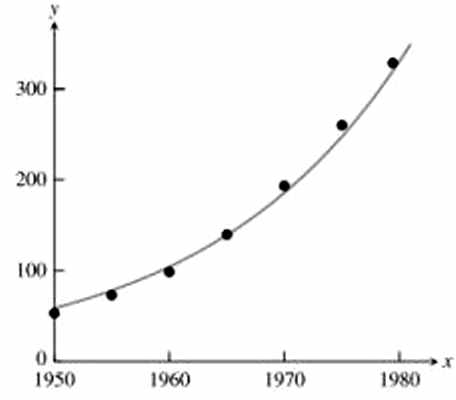


Figura 4.14 Ajuste exponencial de los datos de la oferta mundial de automóviles usando linealización. El mejor ajuste por mínimos cuadrados es $y = 58.51e^{0.05772t}$.

Este modelo se ajusta con los datos de la oferta mundial de automóviles, donde t se mide en años desde 1970 y los automóviles en millones. Cinco pasos del método de Gauss-Newton (4.33) a partir de la estimación inicial $(c_1, c_2) = (50, 0.1)$ producen $(c_1, c_2) \approx (58.51, 0.05772)$ con cuatro dígitos de precisión. El mejor modelo exponencial de mínimos cuadrados para los datos es

$$y = 58.51e^{0.05772t}. \tag{4.36}$$

La RMEC es 7.68, que significa un error promedio de modelado, en el sentido de los mínimos cuadrados, de 7.68 millones de automóviles (vea la figura 4.14).

El mejor modelo (4.36) puede compararse con el mejor modelo exponencial linealizado

$$y = 54.03e^{0.06152t}$$

calculado en el ejemplo 4.8. Esto se obtuvo de las ecuaciones normales aplicadas al modelo linealizado $\ln y = \ln c_1 + c_2 t$. La RMEC de los errores r_i del modelo linealizado es 9.56, mayor que la RMEC de (4.36), como se requería. Sin embargo, el modelo linealizado disminuye al mínimo la RMEC de los errores $\ln y_i - (\ln c_1 + c_2 t_i)$, dando un valor de 0.0357, menor que el valor correspondiente de 0.0568 para el modelo (4.36), también como se requería. Cada uno de los modelos es el ajuste óptimo en su espacio de datos.

La moraleja es que existen algoritmos de cálculo para resolver cualquier problema. La minimización de r_i es el problema estándar de mínimos cuadrados, pero el usuario debe decidir sobre la base del contexto de lo datos si es más apropiado minimizar los errores o linealizarlos. ◀

4.5.3 Método de Levenberg-Marquardt

La minimización por mínimos cuadrados es muy difícil cuando la matriz de coeficientes resulta estar mal condicionada. En el ejemplo 4.5 se encontraron grandes errores en la solución por mínimos cuadrados de $Ax = b$ cuando se emplearon las ecuaciones normales, puesto que $A^T A$ tenía un número de condición grande.

El problema suele ser peor para la minimización no lineal de mínimos cuadrados. Muchas definiciones plausibles del modelo producen matrices Dr mal condicionadas. El método de Levenberg-Marquardt utiliza un "término de regularización" para remediar en parte el problema de condicionamiento. Puede considerarse como una mezcla de Gauss-Newton y el método del descenso más inclinado, que se presentará para los problemas de optimización generales del capítulo 13.

El algoritmo es una modificación simple del método de Gauss-Newton.

Método de Levenberg-Marquardt

Para minimizar

$$r_1(x)^2 + \cdots + r_m(x)^2.$$

Establezca $x^0 =$ vector inicial, $l =$ constante

for $k = 0, 1, 2, \ldots$

$$A = Dr(x^k)$$
$$(A^T A + \lambda \operatorname{diag}(A^T A))v^k = -A^T r(x^k)$$
$$x^{k+1} = x^k + v^k$$

end

El caso de $\lambda = 0$ es idéntico a Gauss-Newton. Al aumentar el parámetro de regularización λ se acentúa el efecto de la diagonal de la matriz $A^T A$, lo que mejora el número de condición y por lo general permite que el método converja a partir de un conjunto más amplio de aproximaciones iniciales x_0 en comparación con Gauss-Newton.

► **EJEMPLO 4.25** Utilice Levenberg-Marquardt para ajustar el modelo $y = c_1 e^{-c_2(t-c_3)^2}$ a los puntos $(t_i, y_i) = \{(1,3), (2,5), (2,7), (3,5), (4,1)\}$.

Es necesario encontrar los c_1, c_2, c_3 que minimizan la RMEC para el vector de errores

$$r = \begin{bmatrix} c_1 e^{-c_2(t_1-c_3)^2} - y_1 \\ \vdots \\ c_1 e^{-c_2(t_5-c_3)^2} - y_5 \end{bmatrix}.$$

La derivada de r evaluada en los cinco puntos de datos es la matriz de 5×3

$$Dr = \begin{bmatrix} e^{-c_2(t_1-c_3)^2} & -c_1(t_1-c_3)^2 e^{-c_2(t_1-c_3)^2} & 2c_1c_2(t_1-c_3)e^{-c_2(t_1-c_3)^2} \\ \vdots & \vdots & \vdots \\ e^{-c_2(t_5-c_3)^2} & -c_1(t_5-c_3)^2 e^{-c_2(t_5-c_3)^2} & 2c_1c_2(t_5-c_3)e^{-c_2(t_5-c_3)^2} \end{bmatrix}.$$

Levenberg-Marquardt con estimación inicial $(c_1, c_2, c_3) = (1, 1, 1)$ y λ fija en 50 iteraciones converge al mejor modelo de mínimos cuadrados

$$y = 6.301e^{-0.5088(t-2.249)^2}.$$

En la figura 4.15 se grafica la mejor aproximación de los datos. El método de Gauss-Newton correspondiente diverge al infinito con la estimación inicial. ◄

El método se originó por la sugerencia de Levenberg [1944] de agregar λI a $A^T A$ en Gauss-Newton para mejorar su condicionamiento. Algunos años después, D. Marquardt, un estadístico de DuPont, mejoró la sugerencia de Levenberg al sustituir la matriz identidad con la diagonal de $A^T A$ (Marquardt [1963]).

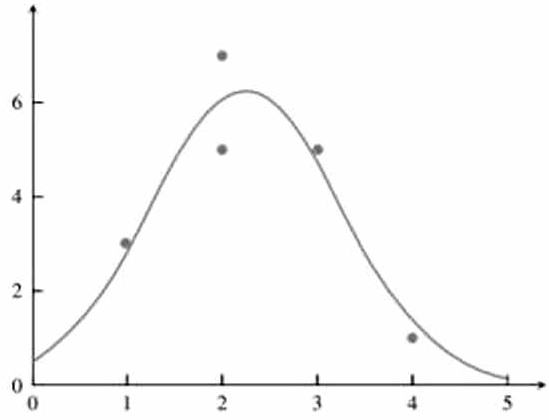


Figura 4.15 Ajuste del modelo del ejemplo 4.25. El método de Levenberg-Marquardt se usa para encontrar el mejor modelo de mínimos cuadrados $y = 6.301e^{-0.5088(t-2.249)^2}$, graficado como la curva sólida.

Aunque λ se ha tratado como una constante para mayor simplicidad, el método se aplica a menudo de manera adaptable con una λ variable. Una estrategia común consiste en seguir disminuyendo λ por un factor de 10 en cada paso de la iteración, siempre y cuando la suma residual de los errores cuadráticos disminuya en el paso, y si la suma aumenta, se rechaza el paso y se aumenta λ por un factor de 10.

4.5 Ejercicios

1. El método de Gauss-Newton puede aplicarse para encontrar el punto $\bar{x}, \bar{y}$ para el cual la suma de los cuadrados de las distancias hasta los tres círculos se minimiza. Utilizando un vector inicial $(x_0, y_0) = (0, 0)$, realice el primer paso para encontrar (x_1, y_1) (a) centros $(0, 1), (1, 1), (0, -1)$ y todos los radios iguales a 1 (b) centros $(-1, 0), (1, 1), (1, -1)$ y todos los radios iguales a 1. (El problema de computadora 1 pide $\bar{x}, \bar{y}$).
2. Realice el primer paso del método multivariado de Newton aplicado al sistema (4.35) para los tres círculos del ejercicio 1. Utilice $(x_0, y_0, K_0) = (0, 0, 0)$. (El problema de computadora 2 pide la solución (x, y, K)).
3. Demuestre que la distancia desde el punto (x, y) hasta un círculo $(x - x_1)^2 + (y - y_1)^2 = R_1^2$ es $|\sqrt{(x - x_1)^2 + (y - y_1)^2} - R_1|$.
4. Demuestre que el método de Gauss-Newton aplicado al sistema lineal $Ax = b$ converge en un paso a la solución de las ecuaciones normales.
5. Encuentre la matriz Dr necesaria para la aplicación de la iteración de Gauss-Newton a cada problema de ajuste del modelo con tres puntos $(t_1, y_1), (t_2, y_2), (t_3, y_3)$, (a) ley de potencia $y = c_1 t^{c_2}$ (b) $y = c_1 t e^{c_2 t}$.
6. Encuentre la matriz Dr necesaria para la aplicación de la iteración de Gauss-Newton a cada problema de ajuste del modelo con tres puntos de datos $(t_1, y_1), (t_2, y_2), (t_3, y_3)$, (a) exponencial trasladado $y = c_3 + c_1 e^{c_2 t}$ (b) ley de potencia trasladada $y = c_3 + c_1 t^{c_2}$.
7. Demuestre que el número de soluciones reales (x, y, K) de (4.35) es infinito o al menos dos.

4.5 Problemas de computadora

1. Aplique el método de Gauss-Newton para encontrar el punto $\bar{x}, \bar{y}$ para el cual la suma de los cuadrados de las distancias a los tres círculos se minimiza. Utilice el vector inicial $(x_0, y_0) = (0, 0)$. (a) Centros $(0, 1), (1, 1), (0, -1)$ y todos los radios iguales a 1. (b) Centros $(-1, 0), (1, 1), (1, -1)$ y todos los radios iguales a 1.

2. Aplique el método multivariado de Newton al sistema (4.35) para los tres círculos del problema de computadora 1. Use el vector inicial $(x_0, y_0, K_0) = (0, 0, 0)$.

3. Encuentre el punto (x, y) y la distancia K que minimiza la suma de los cuadrados de la distancia a los círculos con los radios aumentados por K, como en el ejemplo 4.23 (a) círculos con centros $(-1, 0), (1, 0), (0, 1), (0, -2)$ y todos los radios iguales a 1 (b) círculos con centros $(-2, 0), (3, 0), (0, 2), (0, -2)$ y todos los radios iguales a 1.

4. Realice los pasos del problema de computadora 3 con los siguientes círculos y grafique los resultados (a) centros $(-2, 0), (2, 0), (0, 2), (0, -2)$ y $(2, 2)$, con radios de 1, 1, 1, 1, 2 respectivamente (b) centros $(1, 1), (1, -1), (-1, 1), (-1, -1), (2, 0)$ y todos los radios iguales a 1.

5. Use el método de Gauss-Newton para ajustarse a una ley de potencia para los datos de peso y altura del ejemplo 4.10 sin linealización. Calcule la RMEC.

6. Use el método de Gauss-Newton para ajustar el modelo de concentración en la sangre (4.21) a los datos del ejemplo 4.11 sin linealización.

7. Utilice el método de Levenberg-Marquardt con $\lambda = 1$ para ajustarse a una ley de potencia para los datos de altura y peso del ejemplo de 4.10 sin linealización. Calcule la RMEC.

8. Use el método de Levenberg-Marquardt con $\lambda = 1$ para ajustar el modelo de concentración en la sangre (4.21) a los datos del ejemplo 4.11 sin linealización.

9. Aplique Levenberg-Marquardt para ajustar el modelo $y = c_1 e^{-c_2(t-c_3)^2}$ a los siguientes conjuntos de datos, con una estimación inicial adecuada. Indique el valor inicial, el parámetro de regularización λ usado y la RMEC. Grafique la mejor curva de mínimos cuadrados con sus datos.

   (a) $(t_i, y_i) = \{(-1, 1), (0, 5), (1, 10), (3, 8), (6, 1)\}$

   (b) $(t_i, y_i) = \{(1, 1), (2, 3), (4, 7), (5, 12), (6, 13), (8, 5), (9, 2)(11, 1)\}$

10. Investigue a mayor profundidad el ejemplo 4.25 al determinar las estimaciones iniciales a partir de la malla $0 \le c_1 \le 10$ con un espaciamiento de 1 y $0 \le c_2 \le 1$ con un espaciamiento de 0.1, $c_3 = 1$, para las cuales el método de Levenberg-Marquardt converge a la solución correcta de mínimos cuadrados. Utilice el comando `mesh` de MATLAB para graficar sus respuestas, 1 para un valor inicial convergente y 0 en el caso contrario. Grafique para $\lambda = 50$, $\lambda = 1$, y para el caso de Gauss-Newton $\lambda = 0$. Comente las diferencias encontradas.

11. Aplique Levenberg-Marquardt para ajustar el modelo $y = c_1 e^{-c_2 t} \cos(c_3 t + c_4)$ a los siguientes puntos, con una estimación inicial adecuada. Indique el valor inicial, el parámetro de regularización λ utilizado y la RMEC. Grafique la mejor curva de mínimos cuadrados con los datos. Este problema tiene soluciones múltiples con la misma RMEC, puesto que c_4 sólo está determinado por el módulo 2π.

   (a) $(t_i, y_i) = \{(0, 3), (2, -5), (3, -2), (5, 2), (6, 1), (8, -1), (10, 0)\}$

   (b) $(t_i, y_i) = \{(1, 2), (3, 6), (4, 4), (5, 2), (6, -1), (8, -3)\}$

Comprobación en la realidad **4** **GPS, condicionamiento y mínimos cuadrados no lineales**

El sistema de posicionamiento global (GPS) se compone de 24 satélites que llevan relojes atómicos y que orbitan la Tierra a una altitud de 20 200 km. Cuatro satélites en cada uno de los seis planos, inclinados a 55° respecto a los polos, realizan dos revoluciones por día. En cualquier momento, desde cualquier punto en la Tierra, entre cinco y ocho satélites, están en una línea de visión directa. Cada satélite tiene una sola misión: transmitir señales cuidadosamente sincronizadas desde posiciones predeterminadas en el espacio, que serán recogidas por los receptores GPS en tierra. Los receptores utilizan la información, realizan algunas operaciones matemáticas (descritas brevemente) y determinan las coordenadas (x, y, z) precisas del receptor.

En un instante dado, el receptor recibe la señal sincronizada desde el i-ésimo satélite y determina su tiempo de transmisión t_i, la diferencia entre los tiempos en que la señal fue transmitida y recibida. La velocidad nominal de la señal es la velocidad de la luz, $c \approx 299792.458$ km/seg. Al multiplicar el tiempo de transmisión por c se obtiene la distancia al satélite desde el receptor, colocando el receptor en la superficie de una esfera centrada en la posición del satélite y con radio ct_i. Si hay tres satélites disponibles, entonces se conocen tres esferas, cuya intersección consiste de dos puntos como se muestra en la figura 4.16. Un punto de intersección es la ubicación del receptor. El otro normalmente está alejado de la superficie terrestre y puede descartarse de forma segura. En teoría, el problema se reduce a calcular esta intersección, la solución común de tres ecuaciones de esfera.

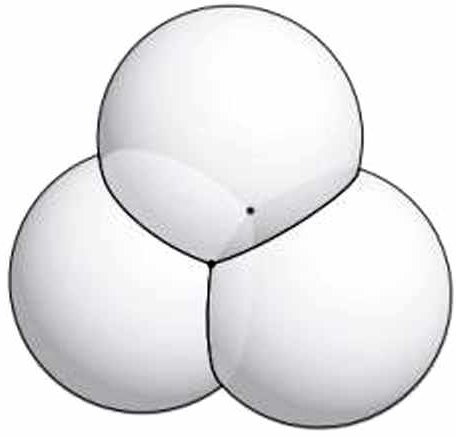


Figura 4.16 Tres esferas intersecantes. Por lo general, sólo dos puntos se encuentran en las tres esferas.

Sin embargo, hay un problema importante con este análisis. Primero, aunque las transmisiones de los satélites están programadas casi hasta un nanosegundo mediante los relojes atómicos a bordo, el reloj en el típico receptor de bajo costo en tierra tiene una precisión relativamente pobre. Si se resuelven las tres ecuaciones con una sincronización ligeramente inexacta, la posición calculada podría estar equivocada por varios kilómetros. Por fortuna, hay una manera de arreglar este problema. El precio a pagar es un satélite extra. Defina d como la diferencia entre la hora sincronizada en los (ahora cuatro) relojes de los satélites y el reloj del receptor en tierra. Denote la ubicación del satélite i por (A_i, B_i, C_i). Entonces, el verdadero punto de intersección (x, y, z) satisface

$$\begin{aligned}
r_1(x,y,z,d) &= \sqrt{(x-A_1)^2+(y-B_1)^2+(z-C_1)^2} - c(t_1-d) = 0\\
r_2(x,y,z,d) &= \sqrt{(x-A_2)^2+(y-B_2)^2+(z-C_2)^2} - c(t_2-d) = 0\\
r_3(x,y,z,d) &= \sqrt{(x-A_3)^2+(y-B_3)^2+(z-C_3)^2} - c(t_3-d) = 0\\
r_4(x,y,z,d) &= \sqrt{(x-A_4)^2+(y-B_4)^2+(z-C_4)^2} - c(t_4-d) = 0 \quad (4.37)
\end{aligned}$$

que debe resolverse para las incógnitas x, y, z, d. La solución del sistema revela no sólo la ubicación del receptor, sino también la hora correcta de los relojes de los satélites, debido a que ahora se conoce d. Por lo tanto, la inexactitud en el reloj receptor GPS puede fijarse usando un satélite adicional.

Geométricamente hablando, cuatro esferas no pueden tener un punto de intersección común, pero lo tendrán si los radios se expanden o contraen en la cantidad común correcta. El sistema

(4.37) que representa la intersección de cuatro esferas es el análogo tridimensional de (4.35), que representa el punto de intersección de tres círculos en el plano.

Puede verse que el sistema (4.37) tienen dos soluciones (x, y, z, d). Las ecuaciones pueden escribirse en forma equivalente

$$\begin{aligned}
(x - A_1)^2 + (y - B_1)^2 + (z - C_1)^2 &= [c(t_1 - d)]^2 \\
(x - A_2)^2 + (y - B_2)^2 + (z - C_2)^2 &= [c(t_2 - d)]^2 \\
(x - A_3)^2 + (y - B_3)^2 + (z - C_3)^2 &= [c(t_3 - d)]^2 \\
(x - A_4)^2 + (y - B_4)^2 + (z - C_4)^2 &= [c(t_4 - d)]^2.
\end{aligned} \tag{4.38}$$

Observe que al restar las últimas tres ecuaciones de la primera, se obtienen tres ecuaciones lineales. Cada ecuación lineal puede usarse para eliminar una variable x, y, z, y al sustituir en cualquiera de las ecuaciones originales, resulta una ecuación cuadrática con una sola variable d. Por lo tanto, el sistema (4.37) tiene a lo sumo dos soluciones reales y pueden encontrarse mediante la fórmula cuadrática.

Cuando se despliega el GPS surgen dos nuevos problemas. El primero es el condicionamiento del sistema de ecuaciones (4.37). Se encontrará que la solución para (x, y, z, d) está mal condicionada cuando los satélites se agrupan estrechamente en el cielo.

La segunda dificultad es que la velocidad de transmisión de las señales no es precisamente c. Las señales pasan a través de los 100 km de la ionosfera y los 10 km de la troposfera, cuyas propiedades electromagnéticas pueden afectar la velocidad de transmisión. Por otra parte, las señales pueden encontrar obstáculos en tierra antes de llegar al receptor, un efecto llamado interferencia de ruta múltiple. En la medida en que estos obstáculos tengan el mismo impacto en cada ruta satelital, la introducción de la corrección de tiempo d en el lado derecho de (4.37) resulta útil. Sin embargo, este supuesto suele no ser viable y conducirá a agregar información con más satélites y a considerar la aplicación de Gauss-Newton para resolver un problema de mínimos cuadrados.

Considere un sistema tridimensional de coordenadas cuyo origen es el centro de la tierra (radio $\approx$6370 kilómetros). Los receptores GPS convierten estas coordenadas en datos de latitud, longitud y altitud para poderlos leer y aplicarlos en procesos de mapeo más sofisticados que se usan en los sistemas mundiales de información geográfica (SIG), un proceso que no se estudiará aquí.

Actividades sugeridas

1. Resuelva el sistema (4.37) usando el método de Newton multivariado. Encuentre la posición del receptor (x, y, z) cerca de la Tierra y la corrección de tiempo d para las posiciones satelitales simultáneas conocidas (15600, 7540, 20140), (18760, 2750, 18610), (17610, 14630, 13480), (19170, 610, 18390) en km, y los intervalos de tiempo medidos 0.07074, 0.07220, 0.07690, 0.07242 en segundos, respectivamente. Establezca el vector inicial $(x_0, y_0, z_0, d_0) = (0, 0, 6370, 0)$. Como verificación, las respuestas son aproximadamente $(x, y, z) = (-41.77271, -16.78919, 6370.0596)$ y $d = -3.201566 \times 10^{-3}$ segundos.

2. Escriba un programa en MATLAB para encontrar una solución mediante la fórmula cuadrática. Sugerencia: al restar las tres últimas ecuaciones de (4.37) de la primera se obtienen tres ecuaciones lineales con cuatro incógnitas $x\vec{u}_x + y\vec{u}_y + z\vec{u}_z + d\vec{u}_d + \vec{w} = 0$, expresada en forma vectorial. Es posible obtener una fórmula para x en términos de d a partir de

$$0 = \det[\vec{u}_y | \vec{u}_z | x\vec{u}_x + y\vec{u}_y + z\vec{u}_z + d\vec{u}_d + \vec{w}],$$

donde se observa que el determinante es lineal en sus columnas y que una matriz con una columna repetida tiene determinante cero. De manera similar pueden deducirse fórmulas en términos de d para y y z, respectivamente, que pueden sustituirse en la primera ecuación cuadrática de (4.37) a fin de obtener una ecuación de una variable.

3. Si el cuadro de herramientas simbólicas de MATLAB está disponible (o un paquete simbólico como Maple o Mathematica), existe una alternativa al paso 2. Defina variables simbólicas usando el comando `syms` y resuelva las ecuaciones simultáneas con el comando del cuadro de herramientas simbólicas `solve`. Use `subs` para evaluar el resultado simbólico como un número de punto flotante.

4. Ahora configure una prueba del condicionamiento del problema del GPS. Defina las posiciones satelitales (A_i, B_i, C_i) a partir de las coordenadas esféricas (ρ, ϕ_i, θ_i) como

$$\begin{aligned} A_i &= \rho \cos\phi_i \cos\theta_i \\ B_i &= \rho \cos\phi_i \operatorname{sen}\theta_i \\ C_i &= \rho \operatorname{sen}\phi_i, \end{aligned}$$

donde $\rho = 26570$ kilómetros es fijo, mientras que $0 \le \phi_i \le \pi/2$ y $0 \le \theta_i \le 2\pi$ para $i = 1, \ldots, 4$ se eligen de manera arbitraria. La coordenada ϕ se restringe de modo que los cuatro satélites estén en el hemisferio superior. Establezca $x = 0$, $y = 0$, $z = 6370$, $d = 0.0001$ y calcule los rangos satelitales correspondientes $R_i = \sqrt{A_i^2 + B_i^2 + (C_i - 6370)^2}$, así como los tiempos de viaje $t_i = d + R_i/c$.

Se definirá un factor de magnificación del error especialmente adaptado a la situación. Los relojes atómicos a bordo de los satélites son correctos hasta 10 nanosegundos, o 10^{-8} segundos. Por lo tanto, es importante estudiar el efecto de los cambios en los tiempos de transmisión de esta magnitud. Sea el error hacia atrás, o el error de entrada, el cambio de la entrada en metros. A la velocidad de la luz, $\Delta t_i = 10^{-8}$ segundos corresponde a $10^{-8}\, c \approx 3$ metros. Sea el error hacia adelante, o el error de salida, el cambio en la posición $\|(\Delta x, \Delta y\, \Delta z)\|_\infty$, causado por un cambio en t_i, también en metros. Entonces es posible definir el

$$\text{factor de magnificación del error} = \frac{\|(\Delta x, \Delta y, \Delta z)\|_\infty}{c\|(\Delta t_1, \ldots, \Delta t_m)\|_\infty},$$

sin dimensiones y el número de condición del problema como el factor de magnificación del error máximo para todos los Δt_i pequeños, (por ejemplo, 10^{-8} o menos).

Cambie cada t_i definida anteriormente por $\Delta t_i = 10^{-8}$ o -10^{-8}, no todas del mismo modo. Denote la nueva solución de las ecuaciones (4.37) por $(\bar{x}, \bar{y}, \bar{z}, \bar{d})$, y calcule la diferencia en la posición $\|(\Delta x, \Delta y\, \Delta z)\|_\infty$, así como el factor de magnificación del error. Pruebe diferentes variaciones de los Δt_i. ¿Cuál es el máximo error de posición encontrado, en metros? Estime el número de condición del problema, sobre la base de los factores de magnificación del error que ha calculado.

5. Ahora repita el paso 4 con un conjunto de satélites agrupados de manera más estrecha. Elija todos los ϕ_i con una separación de 5 por ciento entre sí, y todos los θ_i dentro del mismo marco de separación del 5 por ciento. Resuelva con el mismo error de entrada que en el paso 4, y sin él. Encuentre el error máximo de posición y el factor máximo del error. Compare el condicionamiento del problema cuando los satélites de GPS están agrupados de manera muy estrecha o muy dispersa.

6. Decida si el error y el número de condición del GPS puede reducirse mediante la adición de satélites. Retome la configuración no agrupada de satélites del paso 4, y añada cuatro más. (En todo momento y en todas las posiciones sobre la Tierra, entre 5 y 12 satélites GPS visibles). Diseñe una iteración de Gauss-Newton para resolver el sistema de mínimos cuadrados con ocho ecuaciones y cuatro incógnitas (x, y, z, d). ¿Cuál es un buen vector inicial? Encuentre el máximo error de posición del GPS y estime el número de condiciones. Resuma sus resultados para cuatro satélites no agrupados, cuatro satélites agrupados y ocho satélites no agrupados. ¿Cuál es la mejor configuración y cuál es el máximo error del GPS, en metros, que usted debe esperar sólo sobre la base de las señales de satélite? ✔

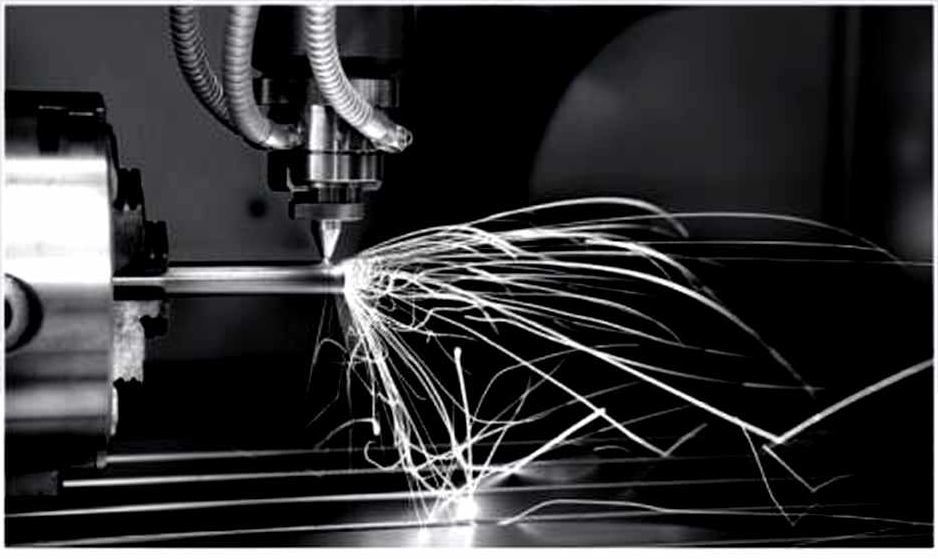
Software y lecturas adicionales

La aproximación por mínimos cuadrados data del siglo XIX. Al igual que la interpolación polinomial, pueden verse como una forma, con la que se encuentra una representación simple de un conjunto de datos complicado considerando su error. Los modelos que se implementan con frecuencia son rectas, polinomios, funciones exponenciales y leyes de potencia. Los datos periódicos requieren representaciones trigonométricas que, llevadas al extremo, conducen a la interpolación trigonométrica y a los ajustes por mínimos cuadrados trigonométricos, los cuales se estudian en el capítulo 10.

Cualquier función que sea lineal en sus coeficientes puede utilizarse para ajustar datos mediante la aplicación del método de tres pasos que se presentó en la sección 4.2, lo cual resulta en una solución de las ecuaciones normales. Para los problemas mal condicionados, no se recomiendan las ecuaciones normales, debido a que en este enfoque el número de condición es aproximadamente cuadrático. La factorización matricial recomendada en este caso es la factorización QR y, en algunos casos, la descomposición del valor singular que se presenta en el capítulo 12. Golub y Van Loan [1996] es una excelente referencia para la factorización QR y otras factorizaciones matriciales. Lawson y Hanson [1995] es una buena fuente de fundamentos para los mínimos cuadrados. Los aspectos estadísticos del ajuste por mínimos cuadrados a la regresión lineal y múltiple se cubren en textos más especializados como Draper y Smith [2001], Fox [1997] y Ryan [1997].

El comando de diagonal invertida en MATLAB aplicado a $Ax = b$ realiza la eliminación gaussiana si el sistema es consistente y resuelve el problema de mínimos cuadrados por factorización QR si es inconsistente. El comando `qr` de MATLAB se basa en la rutina DGEQRF de LAPACK. IMSL proporciona la rutina RLINE para el ajuste de datos por mínimos cuadrados. La rutina E02ADF de la biblioteca NAG realiza una aproximación a los polinomios por mínimos cuadrados, al igual que `polyfit` de MATLAB. Algunos paquetes estadísticos como S^+, SAS, SPSS y Minitab realizan una gran cantidad de análisis de regresión.

Los mínimos cuadrados no lineales se refieren al ajuste de coeficientes que no son lineales en el modelo. El método de Gauss-Newton y sus variantes, como Levenberg-Marquardt, son las herramientas recomendadas para este cálculo, aunque la convergencia no está garantizada, e incluso si se produce la convergencia, esto no implica un grado óptimo único. Vea en Strang y Borre [1997] una introducción a las matemáticas del GPS y en Hoffman-Wellenhof et al. [2001] información general sobre el tema.



CAPÍTULO

5

Diferenciación e integración numérica

La manufactura asistida por computadora depende del control preciso del movimiento a lo largo de una trayectoria prescrita. Por ejemplo, los tornos o fresadoras de control numérico dependen de las curvas paramétricas, a menudo dadas por splines cúbicas o curvas de Bézier delineadas en software de diseño asistido por computadora, para describir la trayectoria de las herramientas de corte o perfilado. La animación generada por computadora en el cine, los juegos de computadora y las aplicaciones de realidad virtual enfrentan problemas similares.

Comprobación en la realidad En la página 278 se considera el problema de controlar la velocidad a lo largo de una trayectoria arbitraria. Para que se recorra la curva a una velocidad deseada, la curva se parametriza con respecto a la longitud del arco. Una manera de controlar la velocidad es integrarla numéricamente adaptando la información de la longitud del arco.

El derivar o integrar es uno de los problemas principales del cálculo. Existen dos direcciones que pueden tomarse para resolver este tipo de problemas. El cálculo numérico y el cálculo simbólico. En este capítulo se analizarán ambos, pero se estudiarán a mayor detalle los aspectos del cálculo numérico. Tanto las derivadas como las integrales tienen definiciones matemáticas claras, pero a menudo el tipo de respuesta deseado por un usuario depende de la manera en la que se especifica la función.

Las derivadas de las funciones como $f(x) = \operatorname{sen} x$ son el tema del cálculo introductorio. Si la función se conoce en términos de funciones elementales, por ejemplo, $f(x) = \operatorname{sen}^3(x^{\tan x} \cosh x)$, su tercera derivada puede encontrarse de manera más rápida mediante los métodos de cálculo simbólico, donde las reglas del cálculo se realizan por medio de una computadora. Esto mismo es cierto para las antiderivadas, en los casos donde la respuesta puede expresarse en términos de funciones elementales.

En la práctica, existen otras dos formas comunes para que una función sea conocida. Una función puede expresarse en forma tabular; por ejemplo, un conjunto de datos $\{(t_1, T_1), \ldots, (t_n, T_n)\}$

de tiempo/temperatura medidos en un experimento, quizá con intervalos de tiempo iguales. En este caso, encontrar la derivada o la antiderivada a partir de las reglas de cálculo es imposible. Finalmente, una función puede especificarse como la salida de un experimento o una simulación por computadora cuya entrada la da el usuario. En los dos últimos casos, los métodos simbólicos de cálculo no pueden aplicarse y se requieren la diferenciación y la integración numérica para resolver el problema.

5.1 DIFERENCIACIÓN NUMÉRICA

Para comenzar, se desarrollan las fórmulas de diferencias finitas para aproximar derivadas. En algunos casos, éste es el objetivo del cálculo. En los capítulos 7 y 8 tales fórmulas se utilizan para discretizar las ecuaciones diferenciales ordinarias y parciales.

5.1.1 Fórmulas de las diferencias finitas

Por definición, la derivada de $f(x)$ en un valor x es

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}, \tag{5.1}$$

siempre y cuando el límite exista. Esto conduce a una fórmula útil para aproximar la derivada en x. El teorema de Taylor dice que si f es dos veces continuamente diferenciable, entonces

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(c), \tag{5.2}$$

donde c se encuentra entre x y $x + h$. La ecuación (5.2) implica la siguiente fórmula:

Fórmula de la diferencia hacia adelante de dos puntos

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(c), \tag{5.3}$$

donde c está entre x y $x + h$.

En un cálculo finito, no puede obtenerse el límite en (5.1), pero (5.3) implica que el cociente se aproximará a la derivada si h es pequeño. Entonces (5.3) calcula la aproximación

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{5.4}$$

y el segundo término de (5.3) se considera un error. Debido a que el error cometido por la aproximación es proporcional al incremento h, es posible reducir el error al disminuir h. La fórmula de la diferencia hacia adelante de dos puntos es un método de primer orden para aproximar la primera derivada. En general, si el error es $O(h^n)$, la fórmula se considerará una aproximación de **orden** n.

Un punto sutil de llamar a la fórmula de "primer orden" es que depende de h. La idea del primer orden es que el error debe ser proporcional a h cuando $h \to 0$. Si $h \to 0$, c es un objetivo en movimiento y, en consecuencia, la proporcionalidad constante cambia. Pero mientras f'' sea continua, la constante de proporcionalidad $f''(c)$ tiende hacia $f''(x)$ cuando $h \to 0$, por lo que es legítimo decir que esta fórmula es de primer orden.

ANOTACIÓN

Convergencia ¿De qué sirve la fórmula del error $-hf''(c)/2$ del método de la diferencia hacia adelante de dos puntos? Se pretende aproximar $f'(x)$, por lo que es probable que $f''(x)$ esté fuera de alcance. Hay dos respuestas. En primer lugar, cuando se verifica el código y el software, una buena comprobación es ejecutarlo con un ejemplo completamente resuelto, donde las respuestas correctas ya se conocen e incluso los errores pueden compararse con lo esperado. En tal caso es posible conocer tanto $f''(x)$ como $f'(x)$. En segundo lugar, aun cuando no puede evaluarse la fórmula completa, a menudo resulta útil saber cómo se escala el error con h. El hecho de que la fórmula sea de primer orden significa que al reducir h a la mitad el error debe disminuir aproximadamente a la mitad, incluso si no se tiene manera de calcular la constante de proporcionalidad $f''(c)/2$.

► EJEMPLO 5.1 Use la fórmula de la diferencia hacia adelante de dos puntos con $h = 0.1$ para aproximar la derivada de $f(x) = 1/x$ en $x = 2$.

La fórmula de la diferencia hacia adelante de dos puntos (5.4) se evalúa como

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} = \frac{\frac{1}{2.1} - \frac{1}{2}}{0.1} \approx -0.2381.$$

La diferencia entre esta aproximación y la derivada correcta $f'(x) = -x^{-2}$ en $x = 2$ es el error

$$-0.2381 - (-0.2500) = 0.0119.$$

Compare esto con el error predicho por la fórmula, que es $hf''(c)/2$ para alguna c entre 2 y 2.1. Dado que $f''(x) = 2x^{-3}$, el error debe estar entre

$$(0.1)2^{-3} \approx 0.0125 \text{ y } (0.1)(2.1)^{-3} \approx 0.0108,$$

lo cual es consistente con el resultado obtenido. Sin embargo, esta información no suele estar disponible. ◄

Mediante una estrategia más avanzada, es posible desarrollar una fórmula de segundo orden. De acuerdo con el teorema de Taylor, si f es tres veces continuamente diferenciable, entonces

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(c_1)$$

y

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(c_2),$$

donde $x - h < c_2 < x < c_1 < x + h$. Al restar las dos ecuaciones se obtiene la siguiente fórmula de tres puntos con un término de error explícito:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{12}f'''(c_1) - \frac{h^2}{12}f'''(c_2). \tag{5.5}$$

A fin de tener más precisión acerca del término de error para la nueva fórmula, se utilizará el siguiente teorema:

TEOREMA 5.1 **Teorema del valor intermedio.** Sea f una función continua en el intervalo $[a, b]$. Sean $x_1, \ldots, x_n$, puntos del intervalo $[a, b]$, y $a_1, \ldots, a_n > 0$. Entonces existe un número c entre a y b de tal forma que

$$(a_1 + \cdots + a_n) f(c) = a_1 f(x_1) + \cdots + a_n f(x_n). \tag{5.6}$$

■

Demostración. Sean $f(x_i)$ el mínimo y $f(x_j)$ el máximo de los valores de la función. Entonces

$$a_1 f(x_i) + \cdots + a_n f(x_i) \le a_1 f(x_1) + \cdots + a_n f(x_n) \le a_1 f(x_j) + \cdots + a_n f(x_j)$$

implica que

$$f(x_i) \le \frac{a_1 f(x_1) + \cdots + a_n f(x_n)}{a_1 + \cdots + a_n} \le f(x_j).$$

Por el teorema del valor intermedio, existe un número c entre x_i y x_j de tal forma que

$$f(c) = \frac{a_1 f(x_1) + \cdots + a_n f(x_n)}{a_1 + \cdots + a_n},$$

y (5.6) se satisface. ❐

El teorema 5.1 dice que es posible combinar los dos últimos términos de (5.5), de donde se obtiene una fórmula de segundo orden:

Fórmula de la diferencia centrada de tres puntos

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(c), \tag{5.7}$$

donde $x - h < c < x + h$.

► EJEMPLO 5.2 Use la fórmula de las diferencias centrales de tres puntos con $h = 0.1$ para aproximar la derivada de $f(x) = 1/x$ en $x = 2$.

La fórmula de la diferencia centrada de tres puntos se evalúa como

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} = \frac{\frac{1}{2.1} - \frac{1}{1.9}}{0.2} \approx -0.2506.$$

El error es 0.0006, que representa una mejora respecto a la fórmula de la diferencia hacia adelante de dos puntos usada en el ejemplo 5.1. ◄

Las fórmulas de aproximación para derivadas de orden superior pueden obtenerse de la misma forma. Por ejemplo, la expansión de Taylor

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \frac{h^3}{6} f'''(x) + \frac{h^4}{24} f^{(\mathrm{iv})}(c_1)$$

y

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) - \frac{h^3}{6} f'''(x) + \frac{h^4}{24} f^{(\mathrm{iv})}(c_2),$$

ANOTACIÓN

Convergencia Las aproximaciones de dos y tres puntos convergen a la derivada a medida que $h \to 0$, aunque a velocidades diferentes. Las fórmulas rompen la regla de oro del cálculo en punto flotante al restar números casi iguales, pero no puede evitarse, puesto que la búsqueda de derivadas es un proceso inherentemente inestable. Para valores muy pequeños de h, el error de redondeo afecta el cálculo, como se muestra en el ejemplo 5.3.

donde $x - h < c_2 < x < c_1 < x + h$ pueden sumarse entre sí para eliminar los términos de la primera derivada, para obtener

$$f(x+h) + f(x-h) - 2f(x) = h^2 f''(x) + \frac{h^4}{24} f^{(\text{iv})}(c_1) + \frac{h^4}{24} f^{(\text{iv})}(c_2).$$

Si se usa el teorema 5.1 para combinar los términos del error y dividir entre h^2, resulta la siguiente fórmula:

Fórmula de la diferencia centrada de tres puntos para la segunda derivada

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} - \frac{h^2}{12} f^{(\text{iv})}(c) \tag{5.8}$$

para alguna c entre $x - h$ y $x + h$.

5.1.2 Error de redondeo

Hasta el momento, todas las fórmulas de este capítulo rompen la regla del capítulo 0 que aconseja no restar números casi iguales. Ésta es la mayor dificultad que presenta la diferenciación numérica, pero en esencia es imposible de evitar. Para entender mejor el problema, considere el siguiente ejemplo:

► **EJEMPLO 5.3** Aproxime la derivada de $f(x) = e^x$ en $x = 0$.

La fórmula de dos puntos (5.4) da

$$f'(x) \approx \frac{e^{x+h} - e^x}{h}, \tag{5.9}$$

y con la fórmula de tres puntos (5.7) se obtiene

$$f'(x) \approx \frac{e^{x+h} - e^{x-h}}{2h}. \tag{5.10}$$

En la siguiente tabla se presentan los resultados de estas fórmulas para $x = 0$ y un amplio rango de tamaños h, junto con los errores comparados con el valor correcto $e^0 = 1$:

h	fórmula (5.9)	error	fórmula (5.10)	error
10^{-1}	1.05170918075648	−0.05170918075648	1.00166750019844	−0.00166750019844
10^{-2}	1.00501670841679	−0.00501670841679	1.00001666674999	−0.00001666674999
10^{-3}	1.00050016670838	−0.00050016670838	1.00000016666668	−0.00000016666668
10^{-4}	1.00005000166714	−0.00005000166714	1.00000000166689	−0.00000000166689
10^{-5}	1.00000500000696	−0.00000500000696	1.00000000001210	−0.00000000001210
10^{-6}	1.00000049996218	−0.00000049996218	0.99999999997324	0.00000000002676
10^{-7}	1.00000004943368	−0.00000004943368	0.99999999947364	0.00000000052636
10^{-8}	0.99999999392253	0.00000000607747	0.99999999392253	0.00000000607747
10^{-9}	1.00000008274037	−0.00000008274037	1.00000002722922	−0.00000002722922

Al principio, el error decrece a medida que h disminuye, siguiendo de cerca los errores esperados $O(h)$ y $O(h^2)$, respectivamente, para la fórmula de la diferencia hacia adelante de dos puntos (5.4) y la fórmula de la diferencia centrada de tres puntos (5.7). Sin embargo, observe el deterioro de las aproximaciones a medida que h disminuye.

La razón por la que las aproximaciones pierden precisión para las h muy pequeñas es la pérdida de significancia. Ambas fórmulas restan números casi iguales, pierden dígitos significativos y además, para empeorar las cosas, aumenta el efecto de dividir entre un número pequeño. ◀

Para tener una mejor idea del grado en que las fórmulas de diferenciación numérica son susceptibles a la pérdida de significancia, se analiza a detalle la fórmula de las diferencias centrales de tres puntos. Indique la versión de punto flotante de la entrada de $f(x+h)$ por $\hat{f}(x+h)$, que diferirá del valor correcto $f(x+h)$ por un número épsilon de máquina en términos relativos. Para el presente análisis, se asumirá que los valores de la función son del orden de 1, por lo que los errores absolutos y relativos son aproximadamente iguales.

Puesto que $\hat{f}(x+h) = f(x+h) + \epsilon_1$ y $\hat{f}(x-h) = f(x-h) + \epsilon_2$, donde $|\epsilon_1|, |\epsilon_2| \approx \epsilon_{\text{máq}}$, la diferencia entre la $f'(x)$ correcta y la versión de máquina de la fórmula de la diferencia centrada de tres puntos (5.7) es

$$\begin{aligned} f'(x)_{\text{correcta}} - f'(x)_{\text{máquina}} &= f'(x) - \frac{\hat{f}(x+h) - \hat{f}(x-h)}{2h} \\ &= f'(x) - \frac{f(x+h) + \epsilon_1 - (f(x-h) + \epsilon_2)}{2h} \\ &= \left(f'(x) - \frac{f(x+h) - f(x-h)}{2h}\right) + \frac{\epsilon_2 - \epsilon_1}{2h} \\ &= \left(f'(x)_{\text{correcta}} - f'(x)_{\text{fórmula}}\right) + \text{error}_{\text{redondeo}} \end{aligned}$$

El error total puede verse como una suma del error de truncamiento, la diferencia entre la derivada correcta y la fórmula de aproximación correcta, y el error de redondeo, que representa la pérdida de significancia de la fórmula implementada por computadora. El error de redondeo tiene un valor absoluto

$$\left|\frac{\epsilon_2 - \epsilon_1}{2h}\right| \leq \frac{2\epsilon_{\text{máq}}}{2h} = \frac{\epsilon_{\text{máq}}}{h},$$

donde $\epsilon_{\text{máq}}$ representa el épsilon máquina. Por lo tanto, el valor absoluto del error de la aproximación por máquina de $f'(x)$ está acotado en la parte superior por

$$E(h) \equiv \frac{h^2}{6} f'''(c) + \frac{\epsilon_{\text{máq}}}{h}, \tag{5.11}$$

donde $x - h < c < x + h$. Previamente se había considerado sólo el primer término del error, el error matemático. La tabla presentada con anterioridad obliga a considerar también el término de la pérdida de significancia.

Resulta instructivo graficar la función $E(h)$, que se muestra en la figura 5.1. El mínimo de $E(h)$ se produce en la solución de

$$0 = E'(h) = -\frac{\epsilon_{\text{máq}}}{h^2} + \frac{M}{3}h, \tag{5.12}$$

donde se ha aproximado $|f'''(c)| \approx |f'''(x)|$ a través de M. Al resolver (5.12) se obtiene

$$h = (3\epsilon_{\text{máq}}/M)^{1/3}$$

para el tamaño del incremento h que da el menor error global, incluyendo los efectos del redondeo de la computadora. En doble precisión, esto es aproximadamente $\epsilon_{\text{máq}}{}^{1/3} \approx 10^{-5}$, que es consistente con la tabla.

El mensaje principal es que la fórmula de la diferencia centrada de tres puntos mejorará la precisión a medida que h se reduzca, hasta que h tenga aproximadamente el tamaño de la raíz cúbica del épsilon de máquina. Cuando h cae por debajo de ese tamaño, el error puede empezar a aumentar de nuevo.

Al realizar un análisis del redondeo, es posible obtener resultados similares para otras fórmulas. En el ejercicio 18 se pide al lector que analice los efectos del redondeo para la fórmula de la diferencia hacia adelante de dos puntos.

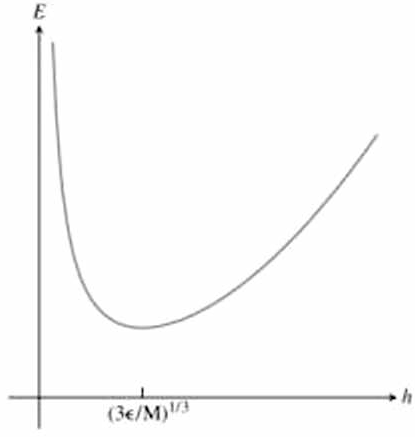


Figura 5.1 El efecto del error de redondeo en la diferenciación numérica. Para una h suficientemente pequeña, el error está dominado por el error de redondeo.

5.1.3 Extrapolación

Suponga que se presenta una fórmula $F(h)$ de orden n para aproximar una cantidad dada Q. El orden significa que

$$Q \approx F(h) + Kh^n,$$

donde K es aproximadamente constante en el rango de h que tiene interés para este análisis. Un ejemplo relevante es

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f'''(c_h)}{6}h^2, \tag{5.13}$$

donde se ha enfatizado el hecho de que el punto desconocido c_h se encuentra entre x y $x + h$, pero depende de h. Aunque c_h no es constante, si la función f es razonablemente suave y h no es demasiado grande, los valores del coeficiente de error $f'''(c_h)$ no debe variar mucho de $f'''(x)/6$.

En un caso como este, puede usarse un poco de álgebra para convertir una fórmula de orden n en una de orden superior. Como se sabe que el orden de la fórmula $F(h)$ es n, al aplicar de nuevo la fórmula con $h/2$ en vez de h, el error se reduce desde una constante multiplicada por h^n hasta una constante multiplicada por $(h/2)^n$, o se reduce por un factor de 2^n. En otras palabras, se espera

$$Q - F(h/2) \approx \frac{1}{2^n}(Q - F(h)). \tag{5.14}$$

Se confía en el supuesto de que K es aproximadamente constante. Observe que (5.14) se resuelve con facilidad para la cantidad Q en cuestión obteniendo la siguiente fórmula:

Fórmula de la extrapolación de orden *n*

$$Q \approx \frac{2^n F(h/2) - F(h)}{2^n - 1}. \tag{5.15}$$

Ésta es la fórmula de **extrapolación** para $F(h)$. Por lo general, la extrapolación, llamada en ocasiones **extrapolación de Richardson**, proporciona una aproximación de Q con un orden superior al de $F(h)$. Para entender por qué, suponga que la fórmula de n-ésimo orden $F_n(h)$ puede escribirse como

$$Q = F_n(h) + Kh^n + O(h^{n+1}).$$

Entonces al dividir h a la mitad se obtiene

$$Q = F_n(h/2) + K\frac{h^n}{2^n} + O(h^{n+1}),$$

y la versión extrapolada, que se denomina $F_{n+1}(h)$, satisfará

$$\begin{aligned} F_{n+1}(h) &= \frac{2^n F_n(h/2) - F_n(h)}{2^n - 1} \\ &= \frac{2^n(Q - Kh^n/2^n - O(h^{n+1})) - (Q - Kh^n - O(h^{n+1}))}{2^n - 1} \\ &= Q + \frac{-Kh^n + Kh^n + O(h^{n+1})}{2^n - 1} = Q + O(h^{n+1}). \end{aligned}$$

Por lo tanto, $F_{n+1}(h)$ es una fórmula de orden $n + 1$ (al menos) para aproximar la cantidad Q.

► EJEMPLO 5.4 Aplique la extrapolación a la fórmula (5.13).

Se inicia con la fórmula de las diferencias centrales de segundo orden $F_2(h)$ para la derivada $f'(x)$. La fórmula de extrapolación (5.15) proporciona una nueva fórmula para $f'(x)$ como

$$\begin{aligned} F_4(x) &= \frac{2^2 F_2(h/2) - F_2(h)}{2^2 - 1} \\ &= \left[4\frac{f(x + h/2) - f(x - h/2)}{h} - \frac{f(x + h) - f(x - h)}{2h}\right]/3 \\ &= \frac{f(x - h) - 8f(x - h/2) + 8f(x + h/2) - f(x + h)}{6h}. \end{aligned} \tag{5.16}$$

Ésta es una fórmula de las diferencias centrales de cinco puntos. El argumento anterior garantiza que esta fórmula tiene un orden de por lo menos tres, pero resulta que tiene orden cuatro, porque los términos del error de orden tres se anulan. De hecho, por inspección $F_4(h) = F_4(-h)$, y por lo tanto el error debe ser el mismo para h y para $-h$. Entonces, los términos del error pueden ser sólo potencias pares de h. ◄

► EJEMPLO 5.5 Aplique la extrapolación a la fórmula de la segunda derivada (5.8).

De nuevo, el método es de segundo orden, por lo que se utiliza la fórmula de extrapolación (5.15) con $n = 2$. La fórmula extrapolada es

$$\begin{aligned} F_4(x) &= \frac{2^2 F_2(h/2) - F_2(h)}{2^2 - 1} \\ &= \left[4\frac{f(x + h/2) - 2f(x) + f(x - h/2)}{h^2/4}\right. \\ &\quad \left. - \frac{f(x + h) - 2f(x) + f(x - h)}{h^2}\right]/3 \\ &= \frac{-f(x - h) + 16f(x - h/2) - 30f(x) + 16f(x + h/2) - f(x + h)}{3h^2}. \end{aligned}$$

El nuevo método para aproximar la segunda derivada es de cuarto orden, por la misma razón que el ejemplo anterior. ◄

5.1.4 Diferenciación e integración simbólica

Las herramientas simbólicas (Symbolic Toolbox) de MATLAB contienen comandos para obtener la derivada de funciones escritas simbólicamente. Los siguientes comandos son ilustrativos:

```
>> syms x;
>> f=sin(3*x);
>> f1=diff(f)

f1=

3*cos(3*x)

>>
```

La tercera derivada también es fácil de encontrar:

```
>>f3=diff(f,3)

f3=

-27*cos(3*x)
```

Para la integración se utiliza el comando simbólico `int` de MATLAB:

```
>>syms x
>>f=sin(x)

f=

sin(x)

>>int(f)

ans=

-cos(x)

>>int(f,0,pi)

ans=

2
```

Con funciones más complicadas, los comandos `pretty` de MATLAB, para ver la respuesta resultante, y `simple`, para simplificarla, son muy útiles como en el siguiente código:

```
>>syms x

>>f=sin(x)^7

f=

sin(x)^7

>>int(f)

ans=

-1/7*sin(x)^6*cos(x)-6/35*sin(x)^4*cos(x)-8/35*sin(x)^2*cos(x)
      -16/35*cos(x)
```

```
>>pretty(simple(int(f)))
                                  3                5                7
      -cos(x) + cos(x)  - 3/5 cos(x)  + 1/7 cos(x)
```

Por supuesto, para algunos integrandos, no existe una expresión de la integral indefinida en términos de funciones elementales. Pruebe la función $f(x) = e^{\text{sen}x}$ para ver como MATLAB se da por vencido. En un caso como éste, sólo existe la alternativa de los métodos numéricos de la siguiente sección.

5.1 Ejercicios

1. Utilice la fórmula de la diferencia hacia adelante de dos puntos para aproximar $f'(1)$, y encuentre el error de aproximación, donde $f(x) = \ln x$, para (a) $h = 0.1$ (b) $h = 0.01$ (c) $h = 0.001$.
2. Utilice la fórmulas de la diferencia centrada de tres puntos para aproximar $f'(0)$, donde $f(x) = e^x$, para (a) $h = 0.1$ (b) $h = 0.01$ (c) $h = 0.001$.
3. Utilice la fórmula de la diferencia hacia adelante de dos puntos para aproximar $f'(\pi/3)$, donde $f(x) = \text{sen } x$, y encuentre el error de aproximación. Además, determine los límites implícitos en el término de error y demuestre que el error de aproximación se encuentra entre ellos (a) $h = 0.1$ (b) $h = 0.01$ (c) $h = 0.001$.
4. Realice los pasos del ejercicio 3 empleando la fórmula de la diferencia centrada de tres puntos.
5. Use la fórmula de la diferencia centrada de tres puntos para la segunda derivada y aproxime $f''(1)$, donde $f(x) = x^{-1}$, para (a) $h = 0.1$ (b) $h = 0.01$ (c) $h = 0.001$. Encuentre el error de aproximación.
6. Use la fórmula de la diferencia centrada de tres puntos para la segunda derivada y aproxime $f''(0)$, donde $f(x) = \cos x$, para (a) $h = 0.1$ (b) $h = 0.01$ (c) $h = 0.001$. Encuentre el error de aproximación.
7. Desarrolle una fórmula de la diferencia hacia atrás de dos puntos para aproximar $f'(x)$, incluyendo el término del error.
8. Demuestre la fórmula de segundo orden para la primera derivada

$$f'(x) = \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + O(h^2).$$

9. Desarrolle una fórmula de segundo orden para la primera derivada $f'(x)$ en términos de $f(x)$, $f(x-h)$ y $f(x-2h)$.
10. Encuentre el término del error y el orden para la fórmula de aproximación

$$f'(x) = \frac{4f(x+h) - 3f(x) - f(x-2h)}{6h}.$$

11. Desarrolle una fórmula de segundo orden para aproximar $f'(x)$ aplicando la extrapolación a la fórmula de la diferencia hacia adelante de dos puntos.
12. (a) Calcule la fórmula de aproximación a $f'(x)$ de la diferencia hacia adelante de dos puntos para $f(x) = 1/x$, donde x y h son arbitrarias. (b) Reste la respuesta correcta para obtener el error explícitamente y demuestre que es proporcional a h. (c) Repita los incisos (a) y (b) usando la fórmula de la diferencia centrada de tres puntos. Ahora el error debe ser proporcional a h^2.
13. Desarrolle un método de segundo orden para aproximar $f'(x)$ que utilice sólo los datos $f(x-h)$, $f(x)$ y $f(x+3h)$.

14. (a) Extrapole la fórmula desarrollada en el ejercicio 13. (b) Demuestre el orden de la nueva fórmula al aproximar $f'(\pi/3)$, donde $f(x) = \operatorname{sen} x$, con $h = 0.1$ y $h = 0.01$.

15. Desarrolle un método de primer orden para aproximar $f'(x)$ que sólo utilice los datos $f(x - h)$, $f(x)$ y $f(x + 3h)$.

16. (a) Aplique extrapolación a la fórmula desarrollada en el ejercicio 15 a fin de obtener una fórmula de segundo orden para $f''(x)$. (b) Demuestre el orden de la nueva fórmula al aproximar $f''(0)$, donde $f(x) = \cos x$, con $h = 0.1$ y $h = 0.01$.

17. Desarrolle un método de segundo orden para aproximar $f'(x)$ que utilice sólo los datos $f(x - 2h)$, $f(x)$ y $f(x + 3h)$.

18. Encuentre $E(h)$, una cota superior para el error de la aproximación por máquina de la fórmula de la diferencia hacia adelante de dos puntos para la primera derivada. Siga el razonamiento anterior (5.11). Encuentre la h correspondiente al mínimo de $E(h)$.

19. Demuestre la fórmula de segundo orden para la tercera derivada

$$f'''(x) = \frac{-f(x-2h) + 2f(x-h) - 2f(x+h) + f(x+2h)}{2h^3} + O(h^2).$$

20. Demuestre la fórmula de segundo orden para la tercera derivada

$$f'''(x) = \frac{f(x-3h) - 6f(x-2h) + 12f(x-h) - 10f(x) + 3f(x+h)}{2h^3} + O(h^2).$$

21. Demuestre la fórmula de segundo orden para la cuarta derivada

$$f^{(iv)}(x) = \frac{f(x-2h) - 4f(x-h) + 6f(x) - 4f(x+h) + f(x+2h)}{h^4} + O(h^2).$$

Esta fórmula se usa en la comprobación en la realidad 2.

22. Este ejercicio justifica las ecuaciones de viga (2.33) y (2.34) en la comprobación en la realidad 2. Sea $f(x)$ una función seis veces diferenciable continuamente.

(a) Demuestre que si $f(x) = f'(x) = 0$, entonces

$$f^{(iv)}(x+h) - \frac{16f(x+h) - 9f(x+2h) + \frac{8}{3}f(x+3h) - \frac{1}{4}f(x+4h)}{h^4} = O(h^2).$$

(Sugerencia: primero demuestre que si $f(x) = f'(x) = 0$, entonces $f(x-h) - 10f(x+h) + 5f(x+2h) - \frac{5}{3}f(x+3h) + \frac{1}{4}f(x+4h) = O(h^6)$. Después aplique el ejercicio 21).

(b) Demuestre que si $f''(x) = f'''(x) = 0$, entonces

$$f^{(iv)}(x+h) - \frac{-28f(x) + 72f(x+h) - 60f(x+2h) + 16f(x+3h)}{17h^4} = O(h^2).$$

(Sugerencia: primero demuestre que si $f''(x) = f'''(x) = 0$, entonces $17f(x-h) - 40f(x) + 30f(x+h) - 8f(x+2h) + f(x+3h) = O(h^6)$. Después aplique el ejercicio 21).

(c) Demuestre que si $f''(x) = f'''(x) = 0$, entonces

$$f^{(iv)}(x) - \frac{72f(x) - 156f(x+h) + 96f(x+2h) - 12f(x+3h)}{17h^4} = O(h^2).$$

(Sugerencia: primero demuestre que si $f''(x) = f'''(x) = 0$, entonces $17f(x-2h) - 130f(x) + 208f(x+h) - 111f(x+2h) + 16f(x+3h) = O(h^6)$. Después aplique el inciso (b) junto con el ejercicio 21).

23. Use la expansión de la serie de Taylor para demostrar que (5.16) es una fórmula de cuarto orden.

24. El término del error en la fórmula de la diferencia hacia adelante de dos puntos para $f'(x)$ puede escribirse de otras maneras. Demuestre el resultado alternativo

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(x) - \frac{h^2}{6}f'''(c),$$

donde c está entre x y $x + h$. Esta forma del error se utilizará en la obtención del método de Crank-Nicolson del capítulo 8.

25. Investigue la razón del nombre de extrapolación. Suponga que $F(h)$ es una fórmula de n-ésimo orden para aproximar una cantidad Q, y considere los puntos $(Kh^2, F(h))$ y $(K(h/2)^2, F(h/2))$ en el plano x-y, donde el error se grafica en el eje x y la salida de la fórmula en el eje y. Encuentre la recta que pasa a través de los dos puntos (la mejor aproximación funcional para la relación entre el error y F). La intersección en y de esta línea es el valor de la fórmula al extrapolar el error a cero. Demuestre que este valor extrapolado está dado por la fórmula (5.15).

5.1 Problemas de computadora

1. Haga una tabla del error de la fórmula de la diferencia centrada de tres puntos para $f'(0)$, donde $f(x) = \operatorname{sen} x - \cos x$, con $h = 10^{-1}, \ldots, 10^{-12}$, como en la tabla de la sección 5.1.2. Dibuje una gráfica de los resultados. ¿El error mínimo corresponde a la expectativa teórica?

2. Haga una tabla y una gráfica del error de la fórmula de la diferencia centrada de tres puntos para $f'(1)$, como en el problema de computadora 1, donde $f(x) = (1 + x)^{-1}$.

3. Haga una tabla y una gráfica del error de la fórmula de la diferencia hacia adelante de dos puntos para $f'(0)$, como en el problema de computadora 1, donde $f(x) = \operatorname{sen} x - \cos x$. Compare sus respuestas con la teoría desarrollada en el ejercicio 18.

4. Haga una tabla y una gráfica como en el problema 3, pero aproxime $f'(1)$, donde $f(x) = x^{-1}$. Compare sus respuestas con la teoría desarrollada en el ejercicio 18.

5. Haga una gráfica como en el problema 1 a fin de aproximar $f''(0)$ para (a) $f(x) = \cos x$ (b) $f(x) = x^{-1}$; para ello utilice la fórmula de la diferencia centrada de tres puntos. ¿Dónde parece ocurrir el error mínimo en términos de epsilon máquina?

5.2 FÓRMULAS DE NEWTON-COTES PARA LA INTEGRACIÓN NUMÉRICA

El cálculo numérico de integrales definidas se basa en muchas de las herramientas que ya se han estudiado. En los capítulos 3 y 4 se han desarrollado métodos para tratar de aproximar una función a un conjunto de puntos, utilizando la interpolación y el modelado por mínimos cuadrados. A continuación se analizarán los métodos para la **integración numérica**, o en **cuadratura**, con base a estas dos ideas.

Por ejemplo, dada una función $f$definida en un intervalo $[a, b]$, es posible encontrar un polinomio de interpolación a través de algunos de los puntos de $f(x)$. Dado que resulta sencillo evaluar la integral definida de un polinomio, este cálculo puede usarse para aproximar la integral de $f(x)$. Éste es el método de Newton-Cotes para aproximar integrales. De manera alternativa, podría encontrarse un polinomio de grado menor que se aproxime bien a la función en el sentido de los mínimos cuadrados y usar la integral como la aproximación, en un método llamado cuadratura gaussiana. Los dos enfoques se describirán en este capítulo.

Para desarrollar las fórmulas de Newton-Cotes se necesitan los valores de tres integrales definidas simples, representadas en la figura 5.2.

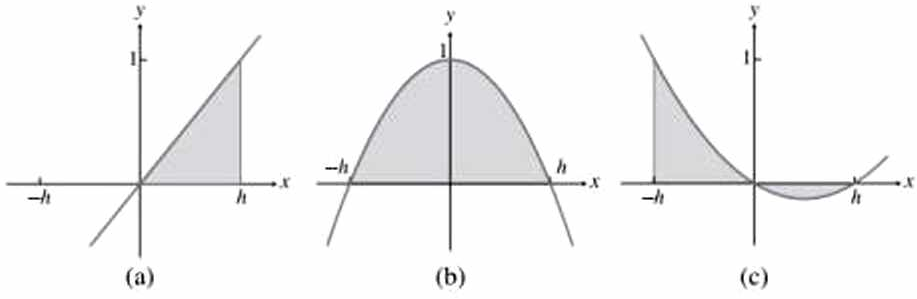


Figura 5.2 Tres integrales simples (5.17), (5.18) y (5.19). Las áreas netas positivas son (a) $h/2$, (b) $4h/3$ y (c) $h/3$.

En la figura 5.2(a) se muestra el área bajo la curva (una recta) que interpola los puntos $(0, 0)$ y $(h, 1)$. La región es un triángulo de altura 1 y base h, por lo que el área es

$$\int_0^h \frac{x}{h}\,dx = h/2. \tag{5.17}$$

En la figura 5.2(b) se muestra el área bajo la curva (una parábola cuadrática) $P(x)$ que interpola los puntos $(-h, 0)$, $(0, 1)$ y $(h, 0)$, con el área

$$\int_{-h}^h P(x)\,dx = x - \frac{x^3}{3h^2} = \frac{4}{3}h. \tag{5.18}$$

En la figura 5.2(c) se muestra el área bajo la curva (parábola cúbica) que interpola los puntos $(-h, 1)$, $(0, 0)$ y $(h, 0)$, con área neta positiva

$$\int_{-h}^h P(x)\,dx = \frac{1}{3}h. \tag{5.19}$$

5.2.1 Regla del trapecio

Se iniciará con la aplicación más simple de la interpolación basada en la integración numérica. Sea $f(x)$ una función con una segunda derivada continua, definida en el intervalo $[x_0, x_1]$, como se muestra en la figura 5.3(a). Los valores de la función están dados como $y_0 = f(x_0)$ y $y_1 = f(x_1)$. Considere el polinomio de interpolación $P_1(x)$ de primer grado que pasa por (x_0, y_0) y (x_1, y_1). Usando la fórmula de Lagrange, se encuentra que el polinomio de interpolación con su término del error es

$$f(x) = y_0\frac{x - x_1}{x_0 - x_1} + y_1\frac{x - x_0}{x_1 - x_0} + \frac{(x - x_0)(x - x_1)}{2!}f''(c_x) = P(x) + E(x).$$

Puede demostrarse que el "punto desconocido" c, depende de la continuidad de x.

Al integrar ambos miembros en el intervalo de interés $[x_0, x_1]$ se obtiene

$$\int_{x_0}^{x_1} f(x)\,dx = \int_{x_0}^{x_1} P(x)\,dx + \int_{x_0}^{x_1} E(x)\,dx.$$

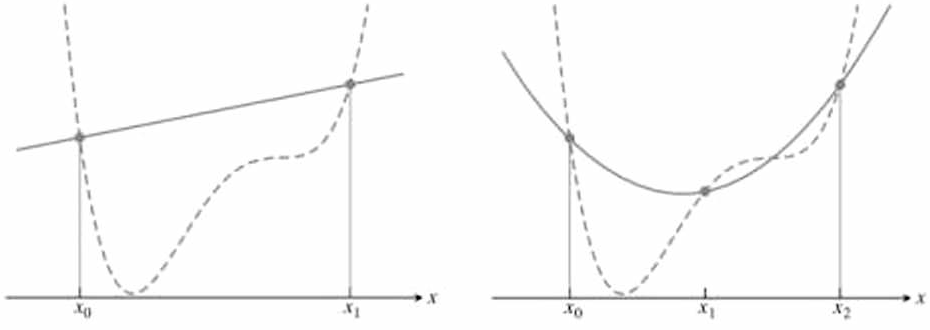


Figura 5.3 Las fórmulas de Newton-Cotes se basan en la interpolación. (a) La regla del trapecio sustituye a la función con la recta que interpola $(x_0, f(x_0))$ y $(x_1, f(x_1))$. (b) La regla de Simpson utiliza la parábola que interpola la función en tres puntos $(x_0, f(x_0))$, $(x_1, f(x_1))$, y $(x_2, f(x_2))$.

Si se calcula la primera integral resulta

$$\begin{aligned}\int_{x_0}^{x_1} P(x)\,dx &= y_0 \int_{x_0}^{x_1} \frac{x - x_1}{x_0 - x_1}\,dx + y_1 \int_{x_0}^{x_1} \frac{x - x_0}{x_1 - x_0}\,dx \\ &= y_0 \frac{h}{2} + y_1 \frac{h}{2} = h\frac{y_0 + y_1}{2},\end{aligned} \tag{5.20}$$

donde se ha definido $h = x_1 - x_0$ como la longitud del intervalo y se han calculado las integrales usando el hecho (5.17). Por ejemplo, al sustituir $w = -x + x_1$ en la primera integral se obtiene

$$\int_{x_0}^{x_1} \frac{x - x_1}{x_0 - x_1}\,dx = \int_h^0 \frac{-w}{-h}(-dw) = \int_0^h \frac{w}{h}\,dw = \frac{h}{2},$$

y la segunda integral, después de sustituir $w = x - x_0$, es

$$\int_{x_0}^{x_1} \frac{x - x_0}{x_1 - x_0}\,dx = \int_0^h \frac{w}{h}\,dw = \frac{h}{2}.$$

La fórmula (5.20) calcula el área de un trapecio, lo que da el nombre a la regla.

El término del error es

$$\begin{aligned}\int_{x_0}^{x_1} E(x)\,dx &= \frac{1}{2!}\int_{x_0}^{x_1} (x - x_0)(x - x_1) f''(c(x))\,dx \\ &= \frac{f''(c)}{2}\int_{x_0}^{x_1} (x - x_0)(x - x_1)\,dx \\ &= \frac{f''(c)}{2}\int_0^h u(u - h)\,du \\ &= -\frac{h^3}{12} f''(c),\end{aligned}$$

donde se ha utilizado el teorema 0.9, el teorema del valor medio para las integrales. Se ha demostrado:

Regla del trapecio

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{h}{2}(y_0 + y_1) - \frac{h^3}{12}f''(c), \tag{5.21}$$

donde $h = x_1 - x_0$ y c está entre x_0 y x_1.

5.2.2 Regla de Simpson

En la figura 5.3(b) se ilustra la **regla de Simpson**, que es similar a la regla del trapecio, excepto que el polinomio de interpolación de primer grado se sustituye por una parábola (polinomio de segundo grado). Como antes, es posible escribir el integrando $f(x)$ como la suma de la parábola de interpolación y su error:

$$\begin{aligned} f(x) &= y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \\ &\quad + y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} + \frac{(x-x_0)(x-x_1)(x-x_2)}{3!}f'''(c_x) \\ &= P(x) + E(x). \end{aligned}$$

La integración da

$$\int_{x_0}^{x_2} f(x)\,dx = \int_{x_0}^{x_2} P(x)\,dx + \int_{x_0}^{x_2} E(x)\,dx,$$

donde

$$\begin{aligned} \int_{x_0}^{x_2} P(x)\,dx &= y_0\int_{x_0}^{x_2}\frac{(x-x_1)(x-x_2)\,dx}{(x_0-x_1)(x_0-x_2)} + y_1\int_{x_0}^{x_2}\frac{(x-x_0)(x-x_2)\,dx}{(x_1-x_0)(x_1-x_2)} \\ &\quad + y_2\int_{x_0}^{x_2}\frac{(x-x_0)(x-x_1)\,dx}{(x_2-x_0)(x_2-x_1)} \\ &= y_0\frac{h}{3} + y_1\frac{4h}{3} + y_2\frac{h}{3}. \end{aligned}$$

Se estableció que $h = x_2 - x_1 = x_1 - x_0$ y se utilizó (5.18) para la integral media y (5.19) para la primera y tercera integrales. El término del error puede calcularse (se omite la demostración) como

$$\int_{x_0}^{x_2} E(x)\,dx = -\frac{h^5}{90}f^{(iv)}(c)$$

para alguna c en el intervalo $[x_0, x_2]$, siempre que $f^{(iv)}$ exista y sea continua. Al concluir la deducción se obtiene la regla de Simpson:

Regla de Simpson

$$\int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}(y_0 + 4y_1 + y_2) - \frac{h^5}{90}f^{(iv)}(c), \tag{5.22}$$

donde $h = x_2 - x_1 = x_1 - x_0$ y c está entre x_0 y x_2.

► EJEMPLO 5.6 Aplique la regla del trapecio y la regla de Simpson para aproximar

$$\int_1^2 \ln x \, dx,$$

y encuentre una cota superior para el error en sus aproximaciones.

La regla del trapecio estima que

$$\int_1^2 \ln x \, dx \approx \frac{h}{2}(y_0 + y_1) = \frac{1}{2}(\ln 1 + \ln 2) = \frac{\ln 2}{2} \approx 0.3466.$$

El error para la regla del trapecio es $-h^3 f''(c)/12$, donde $1 < c < 2$. Dado que $f''(x) = -1/x^2$, la magnitud del error es a lo sumo

$$\frac{1^3}{12c^2} \leq \frac{1}{12} \approx 0.0834.$$

En otras palabras, la regla del trapezoide dice que

$$\int_1^2 \ln x \, dx = 0.3466 \pm 0.0834.$$

La integral puede calcularse exactamente usando la integración por partes:

$$\begin{aligned} \int_1^2 \ln x \, dx &= x \ln x|_1^2 - \int_1^2 dx \\ &= 2\ln 2 - 1\ln 1 - 1 \approx 0.386294. \end{aligned} \tag{5.23}$$

La aproximación de la regla del trapecio y la cota del error son consistentes con este resultado.

La regla de Simpson produce la estimación

$$\int_1^2 \ln x \, dx \approx \frac{h}{3}(y_0 + 4y_1 + y_2) = \frac{0.5}{3}\left(\ln 1 + 4\ln \frac{3}{2} + \ln 2\right) \approx 0.3858.$$

El error para la regla de Simpson es $-h^5 f^{(iv)}(c)/90$, donde $1 < c < 2$. Dado que $f^{(iv)}(x) = -6/x^4$, el error es como máximo

$$\frac{6(0.5)^5}{90c^4} \leq \frac{6(0.5)^5}{90} = \frac{1}{480} \approx 0.0021.$$

Por lo tanto, la regla de Simpson dice que

$$\int_1^2 \ln x \, dx = 0.3858 \pm 0.0021,$$

de nuevo es consistente con el valor correcto y más precisa que la aproximación por la regla del trapecio. ◄

Una forma de comparar las reglas de integración numérica del trapecio o la de Simpson consiste en comparar sus términos del error. Lo que queda establecido a través de la siguiente definición:

DEFINICIÓN 5.2 El **grado de precisión** de un método de integración numérica es el grado k o menor, del polinomio que se usa para integrar. ❐

Por ejemplo, el término del error para la regla del trapecio, $-h^3 f''(c)/12$, muestra que si $f(x)$ es un polinomio de primer grado o menor, el error será cero y el polinomio se integrará exactamente. Así la precisión de la regla del trapecio es de primer grado. Esto es intuitivamente evidente a partir de la geometría, puesto que el área bajo una función lineal se aproxima exactamente mediante un trapecio.

Resulta menos evidente que el grado de precisión de la regla de Simpson es tres, pero eso es lo que muestra el término del error en (5.22). La base geométrica de este sorprendente resultado es el hecho de que una parábola que interseca a una curva cúbica en tres puntos igualmente espaciados, tiene la misma integral que la curva cúbica en ese intervalo (ejercicio 17).

► EJEMPLO 5.7 Encuentre el grado de precisión de la fórmula de Newton-Cotes de tercer grado, llamada la **regla 3/8 de Simpson**.

$$\int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8}(y_0 + 3y_1 + 3y_2 + y_3).$$

Resulta suficiente probar los monomios en sucesión. Los detalles se dejarán al lector. Por ejemplo, si $f(x) = x^2$, se verifica la identidad

$$\frac{3h}{8}(x^2 + 3(x + h)^2 + 3(x + 2h)^2 + (x + 3h)^2) = \frac{(x + 3h)^3 - x^3}{3},$$

donde esta última es la integral correcta de x^2 en $[x, x + 3h]$. La igualdad se cumple para $1, x, x^2$, x^3, pero no para x^4. Por lo tanto, el grado de precisión de la regla es 3. ◄

La regla del trapecio y la regla de Simpson son ejemplos de fórmulas de Newton-Cotes "cerradas", debido a que se incluyen evaluaciones del integrando en los extremos finales del intervalo. Las fórmulas de Newton-Cotes abiertas son útiles para circunstancias en las que no es posible, por ejemplo, aproximar una integral impropia. Las fórmulas abiertas se analizan en la sección 5.2.4.

5.2.3 Fórmulas de Newton-Cotes compuestas

Las reglas del trapecio y de Simpson están limitadas a operar en un solo intervalo. Por supuesto, como las integrales definidas son aditivas en los subintervalos, es posible evaluar una integral al dividir el intervalo en varios subintervalos, aplicando la regla por separado en cada uno para después obtener un total. Esta estrategia se llama **integración numérica compuesta**.

La regla del trapecio compuesta es simplemente la suma de las aproximaciones de la regla del trapecio en subintervalos, o **paneles**, adyacentes. Para aproximar

$$\int_a^b f(x)\,dx,$$

considere una cuadrícula uniformemente espaciada

$$a = x_0 < x_1 < x_2 < \cdots < x_{m-2} < x_{m-1} < x_m = b$$

a lo largo del eje x, donde $h = x_{i+1} - x_i$ para cada i, como se muestra en la figura 5.4. En cada subintervalo, se hace la aproximación con un término de error

$$\int_{x_i}^{x_{i+1}} f(x)\,dx = \frac{h}{2}(f(x_i) + f(x_{i+1})) - \frac{h^3}{12}f''(c_i),$$

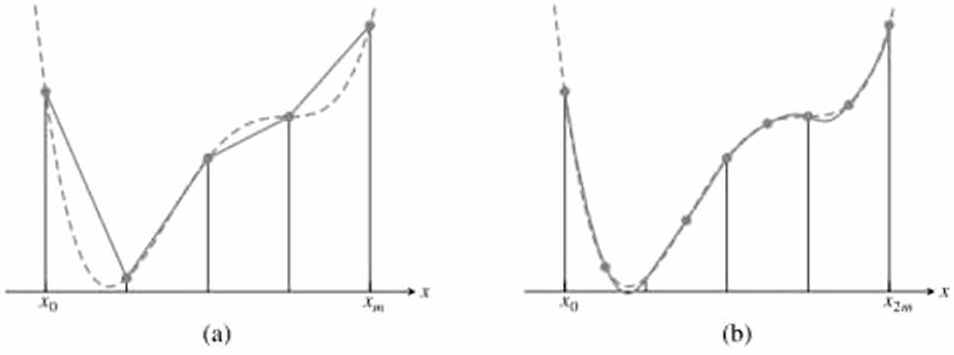


Figura 5.4 Fórmulas de Newton-Cotes compuestas. (a) La regla del trapecio compuesta suma la fórmula de la regla del trapecio (línea sólida) en m subintervalos adyacentes. (b) La regla de Simpson compuesta hace lo mismo.

asumiendo que f'' es continua. La suma de todos los subintervalos (observe la sobreposición en los subintervalos interiores) resulta en

$$\int_a^b f(x)\,dx = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{m-1} f(x_i)\right] - \sum_{i=0}^{m-1} \frac{h^3}{12} f''(c_i).$$

El término de error puede escribirse como

$$\frac{h^3}{12}\sum_{i=0}^{m-1} f''(c_i) = \frac{h^3}{12} m f''(c),$$

de acuerdo con el teorema 5.1, para alguna $a < c < b$. Dado que $mh = (b - a)$, el término del error es $(b - a)h^2 f''(c)/12$. En resumen, si f'' es continua en $[a, b]$, entonces se cumple lo siguiente:

Regla del trapecio compuesta

$$\int_a^b f(x)\,dx = \frac{h}{2}\left(y_0 + y_m + 2\sum_{i=1}^{m-1} y_i\right) - \frac{(b-a)h^2}{12} f''(c) \tag{5.24}$$

donde $h = (b - a)/m$ y c está entre a y b.

La regla compuesta de Simpson sigue la misma estrategia. Considere una cuadrícula uniformemente espaciada

$$a = x_0 < x_1 < x_2 < \cdots < x_{2m-2} < x_{2m-1} < x_{2m} = b$$

a lo largo del eje x, donde $h = x_{i+1} - x_i$ para toda i. En cada panel de longitud $2h$ $[x_{2i}, x_{2i+2}]$, para $i = 0, \ldots, m - 1$, se lleva a cabo un método de Simpson. En otras palabras, el integrando $f(x)$ se aproxima en cada subintervalo mediante el ajuste de la parábola de interpolación en x_{2i}, x_{2i+1} y x_{2i+2}, que se integra y se añade a la suma. La aproximación con el término de error en el subintervalo es

$$\int_{x_{2i}}^{x_{2i+2}} f(x)\,dx = \frac{h}{3}[f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})] - \frac{h^5}{90} f^{(iv)}(c_i).$$

Esta vez, la sobreposición se da sólo en los números pares x_j. Al sumar todos los subintervalos se obtiene

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[f(a) + f(b) + 4\sum_{i=1}^{m} f(x_{2i-1}) + 2\sum_{i=1}^{m-1} f(x_{2i})\right] - \sum_{i=0}^{m-1} \frac{h^5}{90} f^{(iv)}(c_i).$$

El término del error puede escribirse como

$$\frac{h^5}{90}\sum_{i=0}^{m-1} f^{(iv)}(c_i) = \frac{h^5}{90} m f^{(iv)}(c),$$

de acuerdo con el teorema 5.1, para alguna $a < c < b$. Dado que $m \cdot 2h = (b - a)$, el término del error es $(b - a)h^4 f^{(iv)}(c)/180$. Suponiendo que $f^{(iv)}$ es continua en $[a, b]$, se cumple lo siguiente:

Regla de Simpson compuesta

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[y_0 + y_{2m} + 4\sum_{i=1}^{m} y_{2i-1} + 2\sum_{i=1}^{m-1} y_{2i}\right] - \frac{(b-a)h^4}{180} f^{(iv)}(c), \quad (5.25)$$

donde c está entre a y b.

► **EJEMPLO 5.8** Realice las aproximaciones de cuatro paneles de

$$\int_1^2 \ln x\,dx,$$

usando la regla del trapecio compuesta y la regla de Simpson compuesta.

Para la regla del trapecio compuesta en $[1, 2]$, cuatro paneles significa que $h = 1/4$. La aproximación es

$$\begin{aligned}\int_1^2 \ln x\,dx &\approx \frac{1/4}{2}\left[y_0 + y_4 + 2\sum_{i=1}^{3} y_i\right] \\ &= \frac{1}{8}[\ln 1 + \ln 2 + 2(\ln 5/4 + \ln 6/4 + \ln 7/4)] \\ &\approx 0.3837.\end{aligned}$$

El error es a lo sumo

$$\frac{(b-a)h^2}{12}|f''(c)| = \frac{1/16}{12}\frac{1}{c^2} \le \frac{1}{(16)(12)(1^2)} = \frac{1}{192} \approx 0.0052.$$

Una regla de Simpson de cuatro paneles establece que $h = 1/8$. La aproximación es

$$\begin{aligned}\int_1^2 \ln x\,dx &\approx \frac{1/8}{3}\left[y_0 + y_8 + 4\sum_{i=1}^{4} y_{2i-1} + 2\sum_{i=1}^{3} y_{2i}\right] \\ &= \frac{1}{24}[\ln 1 + \ln 2 + 4(\ln 9/8 + \ln 11/8 + \ln 13/8 + \ln 15/8) \\ &\quad + 2(\ln 5/4 + \ln 6/4 + \ln 7/4)] \\ &\approx 0.386292.\end{aligned}$$

Esto concuerda dentro de cinco posiciones decimales con el valor correcto de 0.386294 de (5.23). De hecho, el error no puede ser mayor que

$$\frac{(b-a)h^4}{180}|f^{(iv)}(c)| = \frac{(1/8)^4}{180}\frac{6}{c^4} \le \frac{6}{8^4 \cdot 180 \cdot 1^4} \approx 0.000008.$$ ◄

► EJEMPLO 5.9 Encuentre el número m de paneles necesarios en la regla de Simpson compuesta para aproximar

$$\int_0^{\pi} \operatorname{sen}^2 x \, dx$$

con seis posiciones decimales correctas.

Se requiere que el error satisfaga

$$\frac{(\pi - 0)h^4}{180} |f^{(iv)}(c)| < 0.5 \times 10^{-6}.$$

Como la cuarta derivada de $\operatorname{sen}^2 x$ es $(-8\cos 2x)$, se necesita

$$\frac{\pi h^4}{180} 8 < 0.5 \times 10^{-6},$$

o $h < 0.0435$. Por lo tanto, $m = \texttt{ceil}(\pi/(2h)) = 37$ paneles serán suficientes. ◄

5.2.4 Métodos de Newton-Cotes abiertos

Los métodos de Newton-Cotes denominados cerrados como las reglas del trapecio y la de Simpson requieren valores de entrada en los extremos del intervalo de integración. Algunos integrandos que tienen la singularidad que se pueden quitar en un punto extremo del intervalo, y pueden manejarse con mayor facilidad con un método de Newton-Cotes abierto, el cual no utiliza valores en los puntos extremos. La siguiente regla se aplica a las funciones f cuya segunda derivada f'' es continua en $[a, b]$:

Regla del punto medio

$$\int_{x_0}^{x_1} f(x) \, dx = hf(w) + \frac{h^3}{24} f''(c), \tag{5.26}$$

donde $h = (x_1 - x_0)$, w es el punto medio $x_0 + h/2$ y c está entre x_0 y x_1.

La regla del punto medio también es útil para reducir el número necesario de evaluaciones de la función. En comparación con la regla del trapecio, el método de Newton-Cotes cerrado del mismo orden requiere una evaluación de la función en lugar de dos. Además, el término del error tiene la mitad del tamaño del término del error para la regla del trapecio.

La demostración de (5.26) sigue las mismas líneas que en la obtención de la regla del trapecio. Establezca $h = x_1 - x_0$. La expansión de la serie de Taylor de primer grado para $f(x)$ alrededor del punto medio $w = x_0 + h/2$ en el intervalo es

$$f(x) = f(w) + (x - w) f'(w) + \frac{1}{2}(x - w)^2 f''(c_x),$$

donde c_x depende de x y se encuentra entre x_0 y x_1. Al integrar ambos lados se obtiene

$$\begin{aligned}
\int_{x_0}^{x_1} f(x) \, dx &= (x_1 - x_0) f(w) + f'(w) \int_{x_0}^{x_1} (x - w) \, dx + \frac{1}{2} \int_{x_0}^{x_1} f''(c_x)(x - w)^2 \, dx \\
&= hf(w) + 0 + \frac{f''(c)}{2} \int_{x_0}^{x_1} (x - w)^2 \, dx \\
&= hf(w) + \frac{h^3}{24} f''(c),
\end{aligned}$$

donde $x_0 < c < x_1$. Una vez más, se ha utilizado el teorema del valor medio para las integrales a fin de obtener la segunda derivada de la integral. Esto completa la ecuación (5.26).

La demostración de la versión compuesta se deja al lector (ejercicio 12).

Regla compuesta del punto medio

$$\int_a^b f(x)\,dx = h\sum_{i=1}^{m} f(w_i) + \frac{(b-a)h^2}{24} f''(c), \tag{5.27}$$

donde $h = (b-a)/m$ y c está entre a y b. Las w_i son los puntos medios de los m subintervalos iguales de $[a, b]$.

► EJEMPLO 5.10 Aproxime $\int_0^1 \operatorname{sen} x/x\,dx$ usando la regla compuesta del punto medio con $m = 10$ paneles.

Primero observe que no es posible aplicar directamente un método cerrado al problema, sin un tratamiento especial cuando $x = 0$. El método del punto medio sí puede aplicarse en forma directa. Los puntos medios son 0.05, 0.15, ... , 0.95, por lo que la regla del punto medio compuesta entrega

$$\int_0^1 f(x)\,dx \approx 0.1\sum_{1}^{10} f(m_i) = 0.94620858.$$

La respuesta correcta hasta ocho posiciones es 0.94608307. ◄

Otra regla abierta de Newton-Cotes que resulta útil es

$$\int_{x_0}^{x_4} f(x)\,dx = \frac{4h}{3}[2f(x_1) - f(x_2) + 2f(x_3)] + \frac{14h^5}{45} f^{(iv)}(c), \tag{5.28}$$

donde $h = (x_4 - x_0)/4$, $x_1 = x_0 + h$, $x_2 = x_0 + 2h$, $x_3 = x_0 + 3h$, y donde $x_0 < c < x_4$. La regla tiene un grado de precisión tres. En el ejercicio 11 se le pide que la extienda a una regla compuesta.

5.2 Ejercicios

1. Aplique la regla compuesta del trapecio con $m = 1$, 2 y 4 paneles para aproximar la integral. Calcule el error comparando la aproximación con el valor exacto del cálculo.

$$\text{(a)} \int_0^1 x^2\,dx \quad \text{(b)} \int_0^{\pi/2} \cos x\,dx \quad \text{(c)} \int_0^1 e^x\,dx$$

2. Aplique la regla compuesta del punto medio con $m = 1$, 2 y 4 paneles para aproximar las integrales del ejercicio 1 e indique los errores.

3. Aplique la regla compuesta de Simpson con $m = 1$, 2 y 4 paneles a las integrales del ejercicio 1 e indique los errores.

4. Aplique la regla compuesta de Simpson con $m = 1$, 2 y 4 paneles a las integrales e indique los errores.

$$\text{(a)} \int_0^1 xe^x\,dx \quad \text{(b)} \int_0^1 \frac{dx}{1+x^2}\,dx \quad \text{(c)} \int_0^{\pi} x\cos x\,dx$$

5. Aplique la regla compuesta del punto medio con $m = 1$, 2 y 4 paneles para aproximar las integrales. Calcule el error comparando la aproximación con el valor exacto del cálculo.

$$\text{(a)} \int_0^1 \frac{dx}{\sqrt{x}} \quad \text{(b)} \int_0^1 x^{-1/3}\,dx \quad \text{(c)} \int_0^2 \frac{dx}{\sqrt{2-x}}$$

6. Aplique la regla compuesta del punto medio con $m = 1$, 2 y 4 paneles para aproximar las integrales.

$$\text{(a)} \int_0^{\pi/2} \frac{1-\cos x}{x^2}\,dx \quad \text{(b)} \int_0^1 \frac{e^x-1}{x}\,dx \quad \text{(c)} \int_0^{\pi/2} \frac{\cos x}{\frac{\pi}{2}-x}\,dx$$

7. Aplique la regla de Newton-Cotes abierta (5.28) para aproximar las integrales del ejercicio 5 e indique los errores.

8. Aplique la regla de Newton-Cotes abierta (5.28) para aproximar las integrales del ejercicio 6.

9. Aplique la regla de aproximación de Simpson a $\int_0^1 x^4\,dx$, y demuestre que el error de aproximación coincide con el término del error (5.22).

10. Integre el polinomio de interpolación de la diferencia dividida de Newton para probar la fórmula (a) (5.18) (b) (5.19).

11. Encuentre el grado de precisión de la siguiente aproximación para $\int_{-1}^1 f(x)\,dx$:
(a) $f(1) + f(-1)$ (b) $2/3[f(-1) + f(0) + f(1)]$ (c) $f(-1/\sqrt{3}) + f(1/\sqrt{3})$.

12. Encuentre c_1, c_2 y c_3 tales que la regla

$$\int_0^1 f(x)\,dx \approx c_1 f(0) + c_2 f(0.5) + c_3 f(1)$$

tenga un grado de precisión mayor que uno. (Sugerencia: sustituya $f(x) = 1$, x y x^2). ¿Reconoce el método que resulta?

13. Desarrolle una versión compuesta de la regla de la expresión (5.28), con término del error.

14. Pruebe la regla compuesta del punto medio de la expresión (5.27).

15. Encuentre el grado de precisión de la regla de Newton-Cotes de cuarto grado (con frecuencia llamada la regla de Boole)

$$\int_{x_0}^{x_4} f(x)\,dx \approx \frac{2h}{45}(7y_0 + 32y_1 + 12y_2 + 32y_3 + 7y_4).$$

16. Use el hecho de que el término del error de la regla de Boole es proporcional a $f^{(6)}(c)$ para encontrar el término exacto del error, siguiendo la siguiente estrategia: Calcule la aproximación de Boole para $\int_0^{4h} x^6\,dx$, encuentre el error de aproximación y escríbalo en términos de h y $f^{(6)}(c)$.

17. Sea $P_3(x)$ un polinomio de tercer grado y $P_2(x)$ su polinomio de interpolación en los tres puntos $x = -h$, 0 y h. Pruebe directamente que $\int_{-h}^{h} P_3(x)\,dx = \int_{-h}^{h} P_2(x)\,dx$. ¿Qué dice este hecho acerca de la regla de Simpson?

5.2 Problemas de computadora

1. Utilice la regla del trapecio compuesta con $m = 16$ y 32 paneles para aproximar la integral definida. Compare con la integral correcta e indique los dos errores.

$$\text{(a)} \int_0^4 \frac{x\,dx}{\sqrt{x^2+9}} \quad \text{(b)} \int_0^1 \frac{x^3\,dx}{x^2+1} \quad \text{(c)} \int_0^1 xe^x\,dx \quad \text{(d)} \int_1^3 x^2 \ln x\,dx$$

$$\text{(e)} \int_0^{\pi} x^2 \operatorname{sen} x\,dx \quad \text{(f)} \int_2^3 \frac{x^3\,dx}{\sqrt{x^4-1}} \quad \text{(g)} \int_0^{2\sqrt{3}} \frac{dx}{\sqrt{x^2+4}}\,dx \quad \text{(h)} \int_0^1 \frac{x\,dx}{\sqrt{x^4+1}}$$

2. Aplique la regla de Simpson compuesta a las integrales del problema de computadora 1. Use $m = 16$ y 32, e indique los errores.

3. Utilice la regla del trapecio compuesta con $m = 16$ y 32 paneles para aproximar la integral definida.

$$\text{(a)} \int_0^1 e^{x^2}\,dx \quad \text{(b)} \int_0^{\sqrt{\pi}} \operatorname{sen} x^2\,dx \quad \text{(c)} \int_0^{\pi} e^{\cos x}\,dx \quad \text{(d)} \int_0^1 \ln(x^2+1)\,dx$$

$$\text{(e)} \int_0^1 \frac{x\,dx}{2e^x - e^{-x}} \quad \text{(f)} \int_0^{\pi} \cos e^x\,dx \quad \text{(g)} \int_0^1 x^x\,dx \quad \text{(h)} \int_0^{\pi/2} \ln(\cos x + \operatorname{sen} x)\,dx$$

4. Aplique la regla de Simpson compuesta a las integrales del problema de computadora 3, utilizando $m = 16$ y 32.

5. Aplique la regla del punto medio compuesta a las integrales impropias del ejercicio 5, usando $m = 10$, 100 y 1000. Calcule el error comparando con el valor exacto.

6. Aplique la regla del punto medio compuesta a las integrales impropias del ejercicio 6, usando $m = 16$ y 32.

7. Aplique la regla del punto medio compuesta a las integrales impropias

$$\text{(a)} \int_0^{\frac{\pi}{2}} \frac{x}{\operatorname{sen} x}\,dx \quad \text{(b)} \int_0^{\frac{\pi}{2}} \frac{e^x - 1}{\operatorname{sen} x}\,dx \quad \text{(c)} \int_0^1 \frac{\arctan x}{x}\,dx,$$

usando $m = 16$ y 32.

8. La longitud de arco de la curva definida por $y = f(x)$ desde $x = a$ hasta $x = b$ está dada por la integral $\int_a^b \sqrt{1 + f'(x)^2}\,dx$. Utilice la regla de Simpson compuesta con $m = 32$ paneles para aproximar las longitudes de las curvas

(a) $y = x^3$ en $[0, 1]$ (b) $y = \tan x$ en $[0, \pi/4]$ (c) $y = \arctan x$ en $[0, 1]$.

9. Para las integrales del problema de computadora 1, calcule el error de aproximación de la regla del trapecio compuesta para $h = b - a$, $h/2$, $h/4$, ... , $h/2^8$ y grafique. Haga una gráfico log-log, usando, por ejemplo, el comando `loglog` de MATLAB. ¿Cuál es la pendiente de la gráfica?, ¿concuerda con la teoría?

10. Resuelva el problema de computadora 9, pero use la regla de Simpson compuesta en vez de la regla del trapecio compuesta.

5.3 INTEGRACIÓN DE ROMBERG

En esta sección se inicia con el análisis de los métodos eficaces para el cálculo de integrales definidas, que pueden ampliarse al añadir datos hasta alcanzar la precisión requerida. La integración de Romberg es el resultado de aplicar la extrapolación a la regla compuesta del trapecio. Recuerde de la sección 5.1 que, dada una regla $N(h)$ para aproximar una cantidad M, en función de un tamaño de paso h, la regla puede extrapolarse si se conoce el orden de dicha regla. La ecuación (5.24) muestra que la regla compuesta del trapezoide es una regla de segundo orden en h. Por lo tanto, es posible aplicar la extrapolación para obtener una nueva regla de (al menos) tercer orden.

Al examinar con mayor cuidado el error de la regla del trapecio (5.24), puede demostrarse que, para una función f infinitamente diferenciable,

$$\int_a^b f(x)\,dx = \frac{h}{2}\left(y_0 + y_m + 2\sum_{i=1}^{m-1} y_i\right) + c_2h^2 + c_4h^4 + c_6h^6 + \cdots, \tag{5.29}$$

donde las c_i dependen sólo de las derivadas más altas de f en a y b, y no de h. Por ejemplo, $c_2 = (f'(a) - f'(b))/12$. La ausencia de potencias impares en el error proporciona una ventaja adicional cuando se realiza la extrapolación. Dado que no hay términos con potencias impares, la extrapolación con la fórmula de segundo orden dada por la regla del trapecio compuesta produce una fórmula de cuarto orden; la extrapolación con la fórmula de cuarto orden resultante da una fórmula de sexto orden, y así sucesivamente.

La extrapolación implica combinar la fórmula evaluada una vez en h y una vez más en $h/2$, tamaño de paso a la mitad. Pronosticando hacia dónde se dirige este proceso, defina la siguiente serie de tamaños de paso:

$$\begin{aligned} h_1 &= b - a \\ h_2 &= \frac{1}{2}(b - a) \\ &\vdots \\ h_j &= \frac{1}{2^{j-1}}(b - a). \end{aligned} \tag{5.30}$$

La cantidad que se aproxima es $M = \int_a^b f(x)\,dx$. Defina las fórmulas de aproximación R_{j1} como la regla del trapecio compuesta, usando h_j. Así, $R_{j+1,1}$ es exactamente R_{j1} con un tamaño de paso reducido a la mitad, según sea necesario al aplicar la extrapolación. En segundo lugar, observe la sobreposición de las fórmulas. Algunas de las mismas evaluaciones de función $f(x)$ se requieren tanto en R_{j1} como en $R_{j+1,1}$. Por ejemplo, se tiene

$$\begin{aligned} R_{11} &= \frac{h_1}{2}(f(a) + f(b)) \\ R_{21} &= \frac{h_2}{2}\left(f(a) + f(b) + 2f\left(\frac{a+b}{2}\right)\right) \\ &= \frac{1}{2}R_{11} + h_2 f\left(\frac{a+b}{2}\right). \end{aligned}$$

Se demuestra por inducción (vea el ejercicio 5) que para $j = 2, 3, \ldots,$

$$R_{j1} = \frac{1}{2}R_{j-1,1} + h_j \sum_{i=1}^{2^{j-2}} f(a + (2i - 1)h_j). \tag{5.31}$$

La ecuación (5.31) proporciona una manera eficiente de calcular la regla del trapecio compuesta agregando términos. La segunda característica de la integración de Romberg es la extrapolación. Forme la tabla

$$\begin{array}{cccccc} R_{11} & & & & \\ R_{21} & R_{22} & & & \\ R_{31} & R_{32} & R_{33} & & \\ R_{41} & R_{42} & R_{43} & R_{44} & \\ \vdots & & & & \ddots \end{array} \tag{5.32}$$

donde se define la segunda columna R_{j2} como las extrapolaciones de la primera columna:

$$\begin{aligned} R_{22} &= \frac{2^2R_{21} - R_{11}}{3} \\ R_{32} &= \frac{2^2R_{31} - R_{21}}{3} \\ R_{42} &= \frac{2^2R_{41} - R_{31}}{3}. \end{aligned} \tag{5.33}$$

La tercera columna consta de aproximaciones de cuarto orden a M, por lo que pueden extrapolarse como

$$R_{33} = \frac{4^2 R_{32} - R_{22}}{4^2 - 1}$$
$$R_{43} = \frac{4^2 R_{42} - R_{32}}{4^2 - 1}$$
$$R_{53} = \frac{4^2 R_{52} - R_{42}}{4^2 - 1}, \tag{5.34}$$

y así sucesivamente. La jk-ésima entrada general está dada por la fórmula (vea el ejercicio 6)

$$R_{jk} = \frac{4^{k-1} R_{j,k-1} - R_{j-1,k-1}}{4^{k-1} - 1}. \tag{5.35}$$

La tabla es una matriz triangular inferior que se extiende infinitamente hacia abajo y de manera transversal. La mejor aproximación para la integral definida M es R_{jj}, la entrada inferior más a la derecha calculada hasta el momento, que es una aproximación de $2j$-ésimo orden. El cálculo de la integración de Romberg consiste sólo en escribir las fórmulas (5.31) y (5.35) en un ciclo.

Integración de Romberg

$$R_{11} = (b-a)\frac{f(a)+f(b)}{2}$$
for $j = 2, 3, \ldots$
$$h_j = \frac{b-a}{2^{j-1}}$$
$$R_{j1} = \frac{1}{2}R_{j-1,1} + h_j \sum_{i=1}^{2^{j-2}} f(a + (2i-1)h_j)$$
for $k = 2, \ldots, j$
$$R_{jk} = \frac{4^{k-1} R_{j,k-1} - R_{j-1,k-1}}{4^{k-1} - 1}$$
end
end

El siguiente código de MATLAB es una implementación directa del algoritmo anterior.

```
%Programa 5.1 Integración de Romberg
% Calcula una aproximación a la integral definida
% Entradas: función de MATLAB especificando el integrando f,
%   el intervalo de integración a, b, n = número de filas
% Salida: tabla r de Romberg
function r=romberg(f,a,b,n)
h=(b-a)./(2.^(0:n-1));
r(1,1)=(b-a)*(f(a)+f(b))/2;
for j=2:n
  subtotal = 0;
  for i=1:2^(j-2)
    subtotal = subtotal + f(a+(2*i-1)*h(j));
  end
  r(j,1) = r(j-1,1)/2+h(j)*subtotal;
  for k=2:j
    r(j,k)=(4^(k-1)*r(j,k-1)-r(j-1,k-1))/(4^(k-1)-1);
  end
end
```

► EJEMPLO 5.11 Aplique la integración de Romberg para aproximar $\int_1^2 \ln x \, dx$.

Se usa la función `log` predefinida de MATLAB. Esta función se designa por `@log`. La ejecución del código anterior resulta en

```
>> romberg(@log,1,2,4)

ans =

   0.34657359027997   0                  0                  0
   0.37601934919407   0.38583460216543   0                  0
   0.38369950940944   0.38625956281457   0.38628789352451   0
   0.38564390995210   0.38629204346631   0.38629420884310   0.38629430908625
```

Observe la similitud entre R_{43} y R_{44} en las primeras seis posiciones decimales. Ésta es una señal de convergencia del método de Romberg hacia el valor correcto de la integral definida. Compare con el valor exacto de $21n2 - 1 \approx 0.38629436$. ◄

Al comparar los resultados del ejemplo 5.11 con los del ejemplo 5.8 se presenta una similitud entre la última entrada en la segunda columna de Romberg y los resultados de la regla de Simpson compuesta. Esto no es una coincidencia. De hecho, al igual que la primera columna de Romberg está definida como las entradas sucesivas de la regla del trapecio compuesta, la segunda columna se forma con las entradas de la regla Simpson compuesta. En otras palabras, la extrapolación de la regla del trapecio compuesta es la regla de Simpson compuesta. Vea el ejercicio 3.

Un criterio de terminación común para la integración de Romberg es calcular nuevas filas hasta que dos entradas diagonales sucesivas R_{jj} difieran en menos de una tolerancia al error preestablecida.

5.3 Ejercicios

1. Aplique la integración de Romberg y encuentre la R_{33} para las integrales.

$$\text{(a)} \int_0^1 x^2 \, dx \quad \text{(b)} \int_0^{\pi/2} \cos x \, dx \quad \text{(c)} \int_0^1 e^x \, dx$$

2. Aplique la integración de Romberg y encuentre la R_{33} para las integrales.

$$\text{(a)} \int_0^1 xe^x \, dx \quad \text{(b)} \int_0^1 \frac{dx}{1+x^2} \, dx \quad \text{(c)} \int_0^{\pi} x \cos x \, dx$$

3. Demuestre que la extrapolación de las reglas compuestas del trapecio en R_{11} y R_{21} genera la regla compuesta de Simpson (con tamaño de paso h_2) en R_{22}.
4. Demuestre que la R_{33} de la integración de Romberg puede expresarse como la regla de Boole (con tamaño de paso h_3), definida en el ejercicio 13 de la sección 5.2.
5. Demuestre la fórmula (5.31).
6. Demuestre la fórmula (5.35).

5.3 Problemas de computadora

1. Utilice la aproximación R_{55} de la integración de Romberg para aproximar la integral definida. Compárela con la integral correcta e indique el error.

$$\text{(a)} \int_0^4 \frac{x \, dx}{\sqrt{x^2+9}} \quad \text{(b)} \int_0^1 \frac{x^3 \, dx}{x^2+1} \quad \text{(c)} \int_0^1 xe^x \, dx \quad \text{(d)} \int_1^3 x^2 \ln x \, dx$$

(e) $\int_0^{\pi} x^2 \operatorname{sen} x \, dx$ (f) $\int_2^3 \frac{x^3 \, dx}{\sqrt{x^4 - 1}}$ (g) $\int_0^{2\sqrt{3}} \frac{dx}{\sqrt{x^2 + 4}} \, dx$ (h) $\int_0^1 \frac{x \, dx}{\sqrt{x^4 + 1}} \, dx$

2. Utilice la integración de Romberg para aproximar la integral definida. Como un criterio de detención, continúe hasta que dos entradas diagonales sucesivas difieran en menos de 0.5×10^{-8}.

(a) $\int_0^1 e^{x^2} \, dx$ (b) $\int_0^{\sqrt{\pi}} \operatorname{sen} x^2 \, dx$ (c) $\int_0^{\pi} e^{\cos x} \, dx$ (d) $\int_0^1 \ln(x^2 + 1) \, dx$

(e) $\int_0^1 \frac{x \, dx}{2e^x - e^{-x}}$ (f) $\int_0^{\pi} \cos e^x \, dx$ (g) $\int_0^1 x^x \, dx$ (h) $\int_0^{\pi/2} \ln(\cos x + \operatorname{sen} x) \, dx$

3. (a) Pruebe el orden de la segunda columna de Romberg. Si son aproximaciones de cuarto orden, ¿cómo debería verse una gráfica log-log del error contra h? Realice esto para la integral del ejemplo 5.11. (b) Pruebe el orden de la tercera columna de Romberg.

5.4 CUADRATURA ADAPTATIVA

Los métodos de integración aproximada que se han estudiado hasta ahora usan tamaños de paso iguales. En general, los tamaños de paso más pequeños mejoran la precisión. Una función que varía sin patrón definido requerirá más pasos y, por lo tanto, más tiempo de cálculo, debido a los pasos más pequeños que se requerirán para seguir la pista de las variaciones.

A pesar de que se tienen fórmulas del error para los métodos compuestos, su uso para calcular directamente el valor de h que cumpla con una tolerancia al error dada suele ser difícil. Las fórmulas implican derivadas más altas, que puede ser complicadas y difíciles de estimar en el intervalo en cuestión. Las derivadas más altas pueden incluso no ser calculables si la función se conoce sólo a través de una tabla de valores.

Un segundo problema con la aplicación de las fórmulas compuestas con tamaños de paso iguales es que, con frecuencia, las funciones varían mucho sobre parte de su dominio y varían con mayor lentitud en otras secciones (vea la figura 5.5). Un tamaño de paso que sea suficiente para cumplir con la tolerancia del error en la sección anterior puede ser un exceso en la sección actual.

Por fortuna, hay una manera de solucionar ambos problemas. Si se usa la información surgida de las fórmulas del error de integración, puede desarrollarse un criterio para decidir durante el cálculo qué tamaño de paso es apropiado para un subintervalo particular. La idea detrás de este método, llamado **cuadratura adaptativa**, está estrechamente relacionada con las ideas de extrapolación que se han estudiado en este capítulo.

De acuerdo con (5.21), la regla del trapecio $S_{[a,\,b]}$ en el intervalo $[a, b]$ satisface la fórmula

$$\int_a^b f(x) \, dx = S_{[a,b]} - h^3 \frac{f''(c_0)}{12} \tag{5.36}$$

para alguna $a < c_0 < b$, donde $h = b - a$. Si se establece c como el punto medio de $[a, b]$, podría aplicarse la regla del trapecio para las dos mitades del intervalo y, por la misma fórmula, obtener

$$\begin{aligned} \int_a^b f(x) \, dx &= S_{[a,c]} - \frac{h^3}{8} \frac{f''(c_1)}{12} + S_{[c,b]} - \frac{h^3}{8} \frac{f''(c_2)}{12} \\ &= S_{[a,c]} + S_{[c,b]} - \frac{h^3}{4} \frac{f''(c_3)}{12}, \end{aligned} \tag{5.37}$$

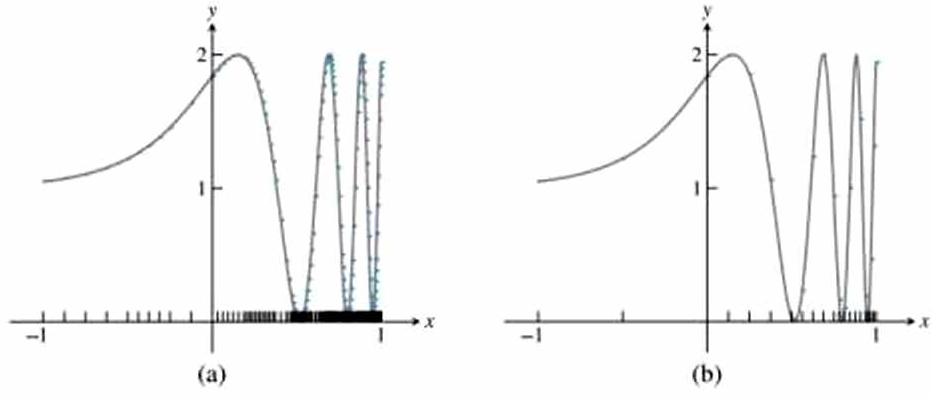


Figura 5.5 Cuadratura adaptativa aplicada a $f(x) = 1 + \operatorname{sen} e^{3x}$. La tolerancia se establece en TOL = 0.005. (a) La regla adaptativa del trapecio requiere 140 subintervalos. (b) La regla adaptativa de Simpson requiere 20 subintervalos.

donde c_1 y c_2 se encuentran en $[a, c]$ y $[c, b]$, respectivamente. Se ha aplicado el teorema 5.1 para consolidar los términos de error. Al restar (5.37) de (5.36) se obtiene

$$\begin{aligned} S_{[a,b]} - (S_{[a,c]} + S_{[c,b]}) &= -\frac{h^3}{4}\frac{f''(c_3)}{12} + h^3\frac{f''(c_0)}{12} \\ &\approx \frac{3}{4}h^3\frac{f''(c_3)}{12}, \end{aligned} \tag{5.38}$$

donde se logra la aproximación $f''(c_3) \approx f''(c_0)$.

Al restar la integral exacta de la ecuación, se escribe el error (aproximadamente) en términos de cosas que pueden calcularse. Por ejemplo, observe que $S_{[a,b]} - (S_{[a,c]} + S_{[c,b]})$ es aproximadamente tres veces el tamaño del error de integración de la fórmula $S_{[a,c]} + S_{[c,b]}$ en $[a, b]$, a partir de (5.37). Por lo tanto, es posible comprobar si la expresión anterior es inferior a 3 *TOL para alguna tolerancia al error, como una forma aproximada de comprobar si la expresión se aproxima la integral exacta desconocida dentro de TOL.

Si el criterio no se cumple, es posible subdividir de nuevo. Ahora que existe un criterio para aceptar una aproximación en un subintervalo dado, pueden seguirse partiendo intervalos a la mitad y aplicando el criterio de las mitades de forma recursiva. Para cada mitad, la tolerancia al error requerida disminuye en un factor de 2, mientras que el error (por la regla del trapecio) debe disminuir en un factor de $2^3 = 8$, por lo que un número suficiente de particiones a la mitad debe permitir que la tolerancia original concuerde con un enfoque compuesto adaptativo.

Cuadratura adaptativa

Para aproximar $\int_a^b f(x)\,dx$ dentro de la tolerancia TOL:

$$c = \frac{a+b}{2}$$

$$S_{[a,b]} = (b-a)\frac{f(a)+f(b)}{2}$$

if $|S_{[a,b]} - S_{[a,c]} - S_{[c,b]}| < 3 \cdot \text{TOL} \cdot \left(\dfrac{b-a}{b_{\text{orig}} - a_{\text{orig}}}\right)$

aceptar $S_{[a,c]} + S_{[c,b]}$ como aproximación sobre $[a, b]$

else

repetir recursivamente lo anterior para $[a, c]$ y $[c, b]$

end

La estrategia de programación de MATLAB funciona de la manera siguiente: se establece una lista de subintervalos que aún no se han procesado. La lista original se compone de un intervalo, $[a, b]$. En general, se elige el último subintervalo de la lista y se aplica el criterio. Si éste se cumple, la aproximación de la integral sobre el subintervalo se agrega a una suma continua, y el intervalo se borra de la lista. Si el criterio no se cumple, el subintervalo se sustituye en la lista por dos subintervalos, alargando la lista en uno, se desplaza al final de la lista y se repite el proceso. El siguiente código de MATLAB lleva a cabo esta estrategia:

```
%Programa 5.2  Cuadratura adaptativa
% Calcula una aproximación a la integral definida
% Entradas: función de Matlab f, intervalo [a0, b0],
% tolerancia al error tol0
% Salida: integral definida aproximada
function int=adapquad(f,a0,b0,tol0)
int=0; n=1; a(1)=a0; b(1)=b0; tol(1)=tol0; app(1)=trap(f,a,b);
while n>0                % n es la posición actual al final de la lista
    c=(a(n)+b(n))/2; oldapp=app(n);
    app(n)=trap(f,a(n),c);app(n+1)=trap(f,c,b(n));
    if abs(oldapp-(app(n)+app(n+1)))<3*tol(n)
        int=int+app(n)+app(n+1);        % éxito
        n=n-1;                          % intervalo terminado
    else                                % dividir en dos intervalos
        b(n+1)=b(n); b(n)=c;            % establecer nuevos intervalos
        a(n+1)=c;
   tol(n)=tol(n)/2; tol(n+1)=tol(n);
   n=n+1;                          % ir al final de la lista, repetir
    end
end

function s=trap(f,a,b)
s=(f(a)+f(b))*(b-a)/2;
```

► EJEMPLO 5.12 Use la cuadratura adaptativa para aproximar la integral

$$\int_{-1}^{1} (1 + \operatorname{sen} e^{3x})\, dx.$$

En la figura 5.5(a) se muestra el resultado del algoritmo de la cuadratura adaptativa para $f(x)$, con una tolerancia al error de 0.005. Aunque se requieren 140 intervalos, sólo 11 de ellos se encuentran en la región "calmada" $[-1, 0]$. La integral definida aproximada es de 2.502 ± 0.005. En una segunda corrida, se cambia la tolerancia al error a 0.5×10^{-4} y se obtiene 2.5008, exacta solo en cuatro posiciones decimales, calculada con 1316 subintervalos. ◄

Por supuesto, la regla del trapecio puede sustituirse por reglas más sofisticadas. Por ejemplo, suponga que $S_{[a,b]}$ denota la regla de Simpson (5.22) en el intervalo $[a, b]$:

$$\int_a^b f(x)\, dx = S_{[a,b]} - \frac{h^5}{90} f^{(iv)}(c_0). \tag{5.39}$$

Al aplicar la regla de Simpson a las dos mitades de $[a, b]$ se obtiene

$$\begin{aligned} \int_a^b f(x)\, dx &= S_{[a,c]} - \frac{h^5}{32}\frac{f^{(iv)}(c_1)}{90} + S_{[c,b]} - \frac{h^5}{32}\frac{f^{(iv)}(c_2)}{90} \\ &= S_{[a,c]} + S_{[c,b]} - \frac{h^5}{16}\frac{f^{(iv)}(c_3)}{90}, \end{aligned} \tag{5.40}$$

donde se ha aplicado el teorema 5.1 para consolidar los términos de error. Si se resta (5.40) de (5.39) resulta

$$S_{[a,b]} - (S_{[a,c]} + S_{[c,b]}) = h^5 \frac{f^{(iv)}(c_0)}{90} - \frac{h^5}{16}\frac{f^{(iv)}(c_3)}{90}$$
$$\approx \frac{15}{16} h^3 \frac{f^{(iv)}(c_3)}{90}, \tag{5.41}$$

donde se realizó la aproximación $f^{(iv)}(c_3) \approx f^{(iv)}(c_0)$.

Como $S_{[a,b]} - (S_{[a,c]} + S_{[c,b]})$ es ahora 15 veces el error de la aproximación $S_{[a,c]} + S_{[c,b]}$ para la integral, puede establecerse un nuevo criterio

$$|S_{[a,b]} - (S_{[a,c]} + S_{[c,b]})| < 15 * \text{TOL} \tag{5.42}$$

y proceder como antes. Es típico reemplazar el 15 por 10 en el criterio para hacer que el algoritmo sea más conservador. En la figura 5.5(b) se muestra una aplicación de la cuadratura adaptativa de Simpson a la misma integral. La integral aproximada es 2.500 cuando se usa una tolerancia de 0.005, con 20 subintervalos, un ahorro considerable sobre la cuadratura adaptativa de la regla del trapecio. Al disminuir la tolerancia a 0.5×10^{-4} se obtiene 2.5008, usando sólo 58 subintervalos.

5.4 Ejercicios

1. Aplique la cuadratura adaptativa en forma manual, usando la regla del trapecio con tolerancia TOL = 0.05 para aproximar las integrales. Encuentre el error de aproximación.

$$\text{(a)} \int_0^1 x^2\,dx \quad \text{(b)} \int_0^{\pi/2} \cos x\,dx \quad \text{(c)} \int_0^1 e^x\,dx$$

2. Aplique la cuadratura adaptativa en forma manual, usando la regla de Simpson con tolerancia TOL = 0.01 para aproximar las integrales. Encuentre el error de aproximación.

$$\text{(a)} \int_0^1 xe^x\,dx \quad \text{(b)} \int_0^1 \frac{dx}{1+x^2}\,dx \quad \text{(c)} \int_0^{\pi} x\cos x\,dx$$

3. Desarrolle un método de cuadratura adaptativa para la regla del punto medio (5.26). Empiece por encontrar un criterio para cumplir con la tolerancia en los subintervalos.

4. Desarrolle un método de cuadratura adaptativa para la regla (5.28).

5.4 Problemas de computadora

1. Use la cuadratura adaptativa del trapecio para aproximar la integral definida con precisión de 0.5×10^{-8}. Imprima la respuesta correcta con ocho posiciones decimales y el número de subintervalos necesarios.

$$\text{(a)} \int_0^4 \frac{x\,dx}{\sqrt{x^2+9}} \quad \text{(b)} \int_0^1 \frac{x^3\,dx}{x^2+1} \quad \text{(c)} \int_0^1 xe^x\,dx \quad \text{(d)} \int_1^3 x^2 \ln x\,dx$$

$$\text{(e)} \int_0^{\pi} x^2 \operatorname{sen} x\,dx \quad \text{(f)} \int_2^3 \frac{x^3\,dx}{\sqrt{x^4-1}} \quad \text{(g)} \int_0^{2\sqrt{3}} \frac{dx}{\sqrt{x^2+4}}\,dx \quad \text{(h)} \int_0^1 \frac{x\,dx}{\sqrt{x^4+1}}\,dx$$

2. Modifique el código de MATLAB para la cuadratura adaptativa de la regla del trapecio de modo que utilice la regla de Simpson, aplicando el criterio (5.42) con el 15 reemplazado por 10. Aproxime la integral del ejemplo 5.12 con precisión de 0.005 y compare su respuesta con la figura 5.5(b). ¿Cuántos subintervalos se requieren?

3. Realice los pasos del problema de computadora 1 para la regla adaptativa de Simpson, desarrollada en el problema de computadora 2.
4. Realice los pasos del problema de computadora 1 para la regla adaptativa del punto medio, desarrollada en el ejercicio 3.
5. Realice los pasos del problema de computadora 1 para la regla adaptativa abierta de Newton-Cotes desarrollada en el ejercicio 4. Use el criterio (5.42) con el 15 sustituido por 10.
6. Use la cuadratura adaptativa del trapecio para aproximar la integral definida con precisión de 0.5×10^{-8}.

   (a) $\int_0^1 e^{x^2}\,dx$ (b) $\int_0^{\sqrt{\pi}} \operatorname{sen} x^2\,dx$ (c) $\int_0^{\pi} e^{\cos x}\,dx$ (d) $\int_0^1 \ln(x^2+1)\,dx$

   (e) $\int_0^1 \frac{x\,dx}{2e^x - e^{-x}}$ (f) $\int_0^{\pi} \cos e^x\,dx$ (g) $\int_0^1 x^x\,dx$ (h) $\int_0^{\pi/2} \ln(\cos x + \operatorname{sen} x)\,dx$

7. Realice los pasos del problema 6, utilizando la cuadratura adaptativa de Simpson.
8. La probabilidad dentro de σ desviaciones estándar de la media de la distribución normal es

$$\frac{1}{\sqrt{2\pi}} \int_{-\sigma}^{\sigma} e^{-x^2/2}\,dx.$$

   Use la cuadratura adaptativa de Simpson para encontrar, con ocho posiciones decimales correctas, la probabilidad dentro de (a) 1 (b) 2 (c) 3 desviaciones estándar.
9. Escriba una función de MATLAB llamada `myerf.m` que utilice la regla adaptativa de Simpson para calcular el valor de

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-s^2}\,ds$$

   con ocho posiciones decimales correctas para una entrada arbitraria x. Pruebe su programa para $x = 1$ y $x = 3$, comparando con la función `erf` de MATLAB.

5.5 CUADRATURA GAUSSIANA

El grado de precisión de un método de cuadratura es el grado para el que todas las funciones polinomiales se integran mediante ese método sin error. Los métodos de Newton-Cotes de grado n tienen un grado de precisión n (para n impar) y $n + 1$ (para n par). La regla del trapecio (Newton-Cotes para $n = 1$) tiene grado de precisión uno. La regla de Simpson ($n = 2$) es correcta hasta los polinomios de tercer grado inclusive.

Para alcanzar este grado de precisión, las fórmulas de Newton-Cotes usan $n + 1$ evaluaciones de la función, realizadas en puntos espaciados uniformemente. La pregunta que surge es una reminiscencia del análisis del capítulo 3 sobre los polinomios de Chebyshev. ¿Las fórmulas de Newton-Cotes son óptimas para su grado de precisión o pueden desarrollarse fórmulas más poderosas? En particular, si el requisito de que los puntos de evaluación se espacien uniformemente se relaja, ¿existirán métodos mejores?

Por lo menos desde el punto de vista del grado de precisión, hay métodos más potentes y sofisticados. Se escogió el más famoso para ser analizado en esta sección. La cuadratura gaussiana tiene grado de precisión $2n + 1$ cuando se utilizan $n + 1$ puntos, el doble de Newton-Cotes. Los puntos de evaluación no están uniformemente espaciados. Se explicará cómo la cuadratura

gaussiana implica una breve digresión en funciones ortogonales, lo que no sólo es interesante por derecho propio, sino la punta del iceberg de los métodos numéricos inspirados en los beneficios de la ortogonalidad.

DEFINICIÓN 5.3 El conjunto de funciones diferentes de cero $\{p_0, \ldots, p_n\}$ en el intervalo $[a, b]$ es **ortogonal** en $[a, b]$ si

$$\int_a^b p_j(x)\, p_k(x)\, dx = \begin{cases} 0 & j \neq k \\ \neq 0 & j = k. \end{cases}$$

❐

TEOREMA 5.4 Si $\{p_0, p_1, \ldots, p_n\}$ es un conjunto ortogonal de polinomios en el intervalo $[a, b]$, donde el grado $p_i = i$, entonces $\{p_0, p_1, \ldots, p_n\}$ es una base para el espacio vectorial de los polinomios a lo sumo de grado n en $[a, b]$. ■

Demostración. Debe demostrarse que los polinomios cubren el espacio vectorial y son linealmente independientes. Un argumento de inducción simple demuestra que cualquier conjunto de polinomios $\{p_0, p_1, \ldots, p_n\}$, donde $p_i = i$, cubre el espacio de los polinomios a lo sumo de grado n. Para mostrar la independencia lineal, se supondrá que existe una dependencia lineal $\sum_{i=0}^{n} c_i p_i(x) = 0$ y se demostrará que toda c_i debe ser cero, utilizando el supuesto de ortogonalidad. Para cualquier $0 \leq k \leq n$, dado que p_k es ortogonal a cualquier polinomio menos a sí mismo, se obtiene

$$0 = \int_a^b p_k \sum_{i=0}^{n} c_i p_i(x)\, dx = \sum_{i=0}^{n} c_i \int_a^b p_k p_i\, dx = c_k \int_a^b p_k^2\, dx. \tag{5.43}$$

Por lo tanto, $c_k = 0$. ❐

La demostración del siguiente teorema se omite.

TEOREMA 5.5 Si $\{p_0, \ldots, p_n\}$ es un conjunto ortogonal de polinomios en $[a, b]$ y si el grado $p_i = i$, entonces p_i tiene i raíces distintas en el intervalo (a, b). ■

► EJEMPLO 5.13 Encuentre un conjunto de tres polinomios ortogonales en el intervalo $[-1, 1]$.

Se estima que $p_0(x) = 1$ y $p_1(x) = x$ es un buen comienzo, porque

$$\int_{-1}^{1} 1 \cdot x\, dx = 0.$$

Si se prueba $p_2(x) = x^2$ no funciona del todo, puesto que carece de ortogonalidad con $p_0(x)$:

$$\int_{-1}^{1} p_0(x) x^2\, dx = 2/3 \neq 0.$$

ANOTACIÓN

Ortogonalidad En el capítulo 4 se encontró que la ortogonalidad de los vectores de dimensión finita era útil para formular y resolver problemas de mínimos cuadrados. Para la cuadratura, se requiere la ortogonalidad en espacios de dimensión infinita como el espacio vectorial de polinomios en una variable. Una base es la de los monomios $\{1, x, x^2, \ldots\}$. Sin embargo, una base más útil es aquella que también es un conjunto ortogonal. Por ortogonalidad en el intervalo $[-1, 1]$, la mejor opción la constituyen los polinomios de Legendre.

Al ajustarse a $p_2(x) = x^2 + c$, se tiene que

$$\int_{-1}^{1} p_0(x)(x^2 + c)\, dx = 2/3 + 2c = 0,$$

siempre y cuando $c = -1/3$. Asegúrese de que p_1 y p_2 sean ortogonales (vea el ejercicio 7). Por lo tanto, el conjunto $\{1, x, x^2 - 1/3\}$ es un conjunto ortogonal en $[-1, 1]$. ◄

Los tres polinomios del ejemplo 5.13 pertenecen a un conjunto descubierto por Legendre.

► EJEMPLO 5.14 Demuestre que el conjunto de **polinomios de Legendre**

$$p_i(x) = \frac{1}{2^i i!} \frac{d^i}{dx^i}[(x^2 - 1)^i]$$

para $0 \leq i \leq n$ es ortogonal en $[-1, 1]$.

Observe primero que $p_i(x)$ es un polinomio de grado i (como la derivada i-ésima de un polinomio de grado $2i$). Segundo, note que la derivada i-ésima de $(x^2 - 1)^j$ es divisible entre $(x^2 - 1)$ si $i < j$.

Se desea demostrar que si $i < j$, entonces la integral

$$\int_{-1}^{1} [(x^2 - 1)^i]^{(i)} [(x^2 - 1)^j]^{(j)}\, dx$$

es igual a cero. Al integrar por partes con $u = [(x^2 - 1)^i]^{(i)}$ y $dv = [(x^2 - 1)^j]^{(j)}\, dx$ se obtiene

$$\begin{aligned} uv - \int_{-1}^{1} v\, du &= [(x^2 - 1)^i]^{(i)} [(x^2 - 1)^j]^{(j-1)} \Big|_{-1}^{1} \\ &\quad - \int_{-1}^{1} [(x^2 - 1)^i]^{(i+1)} [(x^2 - 1)^j]^{(j-1)}\, dx \\ &= - \int_{-1}^{1} [(x^2 - 1)^i]^{(i+1)} [(x^2 - 1)^j]^{(j-1)}\, dx, \end{aligned}$$

puesto que $[(x^2 - 1)^j]^{(j-1)}$ es divisible entre $(x^2 - 1)$.

Después de $i + 1$ integraciones por partes repetidas, queda

$$(-1)^{i+1} \int_{-1}^{1} [(x^2 - 1)^i]^{(2i+1)} [(x^2 - 1)^j]^{(j-i-1)}\, dx = 0,$$

porque la derivada $(2i + 1)$-ésima de $(x^2 - 1)^i$ es cero. ◄

Por el teorema 5.5, el n-ésimo polinomio de Legendre tiene n raíces $x_1, \ldots, x_n$ en $[-1, 1]$. La cuadratura gaussiana de una función es tan sólo una combinación lineal de las evaluaciones de las funciones en las raíces de Legendre. Esto se logra al aproximar la integral de la función deseada mediante la integral del polinomio de interpolación, cuyos nodos son las raíces de Legendre.

Fije una n y sea $Q(x)$ el polinomio de interpolación para el integrando $f(x)$ en los nodos $x_1, \ldots, x_n$. Si se usa la formulación de Lagrange, es posible escribir

$$Q(x) = \sum_{i=1}^{n} L_i(x) f(x_i), \text{ donde } L_i(x) = \frac{(x - x_1) \cdots \overline{(x - x_i)} \cdots (x - x_n)}{(x_i - x_1) \cdots \overline{(x_i - x_i)} \cdots (x_i - x_n)}.$$

Al integrar ambos lados se obtiene la siguiente aproximación de la integral:

n	raíces x_i	coeficientes c_i
2	$-\sqrt{1/3} = -0.57735026918963$	$1 = 1.00000000000000$
	$\sqrt{1/3} = 0.57735026918963$	$1 = 1.00000000000000$
3	$-\sqrt{3/5} = -0.77459666924148$	$5/9 = 0.55555555555555$
	$0 = 0.00000000000000$	$8/9 = 0.88888888888888$
	$\sqrt{3/5} = 0.77459666924148$	$5/9 = 0.55555555555555$
4	$-\sqrt{\frac{15+2\sqrt{30}}{35}} = -0.86113631159405$	$\frac{90-5\sqrt{30}}{180} = 0.34785484513745$
	$-\sqrt{\frac{15-2\sqrt{30}}{35}} = -0.33998104358486$	$\frac{90+5\sqrt{30}}{180} = 0.65214515486255$
	$\sqrt{\frac{15-2\sqrt{30}}{35}} = 0.33998104358486$	$\frac{90+5\sqrt{30}}{180} = 0.65214515486255$
	$\sqrt{\frac{15+2\sqrt{30}}{35}} = 0.86113631159405$	$\frac{90-5\sqrt{30}}{180} = 0.34785484513745$

Tabla 5.1 Coeficientes de la cuadratura gaussiana. Raíces x_i de los n-ésimos polinomios de Legendre, y coeficientes c_i en (5.44).

Cuadratura gaussiana

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} c_i f(x_i), \tag{5.44}$$

donde

$$c_i = \int_{-1}^{1} L_i(x)\,dx, \quad i = 1, \ldots, n.$$

Las c_i se tabulan con gran exactitud. Los valores se dan en la tabla 5.1 hasta $n = 4$.

► **EJEMPLO 5.15** Aproxime

$$\int_{-1}^{1} e^{-\frac{x^2}{2}}\,dx,$$

usando la cuadratura gaussiana.

La respuesta correcta hasta 14 dígitos es 1.71124878378430. Para el integrando $f(x) = e^{-x^2/2}$, la aproximación de la cuadratura gaussiana con $n = 2$ es

$$\begin{aligned}\int_{-1}^{1} e^{-\frac{x^2}{2}}\,dx &\approx c_1 f(x_1) + c_2 f(x_2)\\ &= 1 \cdot f(-\sqrt{1/3}) + 1 \cdot f(\sqrt{1/3}) \approx 1.69296344978123.\end{aligned}$$

La aproximación para $n = 3$ es

$$\frac{5}{9} f(-\sqrt{3/5}) + \frac{8}{9} f(0) + \frac{5}{9} f(\sqrt{3/5}) \approx 1.71202024520191,$$

y la aproximación con $n = 4$ es

$$c_1 f(x_1) + c_2 f(x_2) + c_3 f(x_3) + c_4 f(x_4) \approx 1.71122450459949.$$

Esta aproximación, que emplea cuatro evaluaciones de la función, es mucho más cercana que la aproximación R_{33} de Romberg, que utiliza cinco evaluaciones de funciones uniformemente espaciadas en $[-1, 1]$:

```
1.21306131942527   0                  0
1.60653065971263   1.73768710647509   0
1.68576223244091   1.71217275668367   1.71047180003091
```

◄

El secreto de la precisión de la cuadratura gaussiana se evidencia con el siguiente teorema.

TEOREMA 5.6 El método de la cuadratura gaussiana, usando el polinomio de Legendre de grado n en $[-1, 1]$, tiene un grado de precisión $2n - 1$. ■

Demostración. Sea $P(x)$ un polinomio a lo sumo de grado $2n - 1$. Debe demostrarse que puede integrarse exactamente mediante la cuadratura gaussiana.

Si se usa la división larga de polinomios, puede expresarse

$$P(x) = S(x)p_n(x) + R(x), \tag{5.45}$$

donde los $S(x)$ y $R(x)$ son polinomios de grado menor que n. Tenga en cuenta que la cuadratura gaussiana será exacta en el polinomio $R(x)$, puesto que sólo es la integración del polinomio de interpolación de grado $n - 1$, que es idéntico a $R(x)$.

En las raíces x_i del n-ésimo polinomio de Legendre, $P(x_i) = R(x_i)$, dado que $p_n(x_i) = 0$ para todas las i. Esto implica que sus aproximaciones por la cuadratura gaussiana serán las mismas. Pero sus integrales también son idénticas: al integrar (5.45) resulta

$$\int_{-1}^{1} P(x)\,dx = \int_{-1}^{1} S(x)\,p_n(x)\,dx + \int_{-1}^{1} R(x)\,dx = 0 + \int_{-1}^{1} R(x)\,dx,$$

ya que por el teorema 5.4, $S(x)$ puede escribirse como una combinación lineal de polinomios de grado menor que n, que son ortogonales a $p_n(x)$. Dado que la cuadratura gaussiana es exacta en $R(x)$, también debe serlo para $P(x)$. ❐

Para aproximar las integrales en un intervalo general $[a, b]$, el problema debe trasladarse de nuevo a $[-1, 1]$. Si se usa la sustitución $t = (2x - a - b)/(b - a)$, resulta fácil comprobar que

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f\left(\frac{(b-a)t + b + a}{2}\right)\frac{b-a}{2}\,dt. \tag{5.46}$$

Esto se demuestra con un ejemplo.

► EJEMPLO 5.16 Aproxime la integral

$$\int_1^2 \ln x\,dx,$$

usando la cuadratura gaussiana.

A partir de (5.46),

$$\int_1^2 \ln x\,dx = \int_{-1}^{1} \ln\left(\frac{t+3}{2}\right)\frac{1}{2}\,dt.$$

Ahora es posible establecer $f(t) = \ln((t + 3)/2)/2$ y emplear las raíces y los coeficientes estándar. El resultado para $n = 4$ es 0.38629449693871, comparado con el valor correcto $2 \ln 2 - 1 \approx$ 0.38629436111989. De nuevo, esto es más preciso que la integración de Romberg utilizando cuatro puntos del ejemplo 5.11. ◄

5.5 Ejercicios

1. Aproxime las integrales usando la cuadratura gaussiana con $n = 2$. Compare su respuesta con el valor correcto e indique el error de aproximación.

$$\text{(a)} \int_{-1}^{1} (x^3 + 2x)\,dx \quad \text{(b)} \int_{-1}^{1} x^4\,dx \quad \text{(c)} \int_{-1}^{1} e^x\,dx \quad \text{(d)} \int_{-1}^{1} \cos \pi x\,dx$$

2. Aproxime las integrales del ejercicio 1 usando la cuadratura gaussiana con $n = 3$. También indique el error.
3. Aproxime las integrales del ejercicio 1 usando la cuadratura gaussiana con $n = 4$. También indique el error.
4. Cambie las variables empleando la sustitución (5.46) para reescribir la integral en $[-1, 1]$.

$$\text{(a)} \int_{0}^{4} \frac{x\,dx}{\sqrt{x^2 + 9}} \quad \text{(b)} \int_{0}^{1} \frac{x^3\,dx}{x^2 + 1} \quad \text{(c)} \int_{0}^{1} xe^x\,dx \quad \text{(d)} \int_{1}^{3} x^2 \ln x\,dx$$

5. Aproxime las integrales del ejercicio 4 usando la cuadratura gaussiana con $n = 3$.
6. Aproxime las integrales usando la cuadratura gaussiana con $n = 4$.

$$\text{(a)} \int_{0}^{1} (x^3 + 2x)\,dx \quad \text{(b)} \int_{1}^{4} \ln x\,dx \quad \text{(c)} \int_{-1}^{2} x^5\,dx \quad \text{(d)} \int_{-3}^{3} e^{-\frac{x^2}{2}}\,dx$$

7. Demuestre que los polinomios de Legendre $p_1(x) = x$ y $p_2(x) = x^2 - 1/3$ son ortogonales en $[-1, 1]$.
8. Encuentre los polinomios de Legendre hasta de grado 3 y compárelos con el ejemplo 5.13.
9. Verifique los coeficientes c_i y las x_i en la tabla 5.1 para el grado $n = 3$.
10. Verifique los coeficientes c_i y las x_i en la tabla 5.1 para el grado $n = 4$.

Comprobación en la realidad **5** **Control de movimiento en el modelado asistido por computadora**

El modelado y la manufactura asistidos por computadora requieren un control preciso de la posición espacial a lo largo de una trayectoria de movimiento prescrita. Se ilustrará el uso de la integración numérica para resolver una pieza fundamental del problema: la subdivisión, o la división de una trayectoria arbitraria en subtrayectorias de la misma longitud.

En los problemas numéricos de maquinado, es preferible mantener la velocidad constante a lo largo de la trayectoria. Durante cada segundo, se debe avanzar a lo largo de una longitud igual de la interfase del material con la máquina. En otras aplicaciones de planificación del movimiento, incluyendo la animación por computadora, pueden requerirse curvas de avance más complicadas: una mano que trata de alcanzar la perilla de una puerta puede comenzar y terminar con baja velocidad y tener una mayor velocidad en el intervalo de ambos. Las aplicaciones de robótica y de realidad virtual requieren la construcción de curvas y superficies parametrizadas sobre las cuales se navega. Con frecuencia, el primer paso requerido es la construcción de una tabla con pequeños incrementos iguales sobre la distancia de la trayectoria.

Suponga que se tiene una trayectoria paramétrica $P = \{x(t), y(t) | 0 \le t \le 1\}$. En la figura 5.6 se muestra la trayectoria de ejemplo

$$P = \begin{cases} x(t) = 0.5 + 0.3t + 3.9t^2 - 4.7t^3 \\ y(t) = 1.5 + 0.3t + 0.9t^2 - 2.7t^3 \end{cases},$$

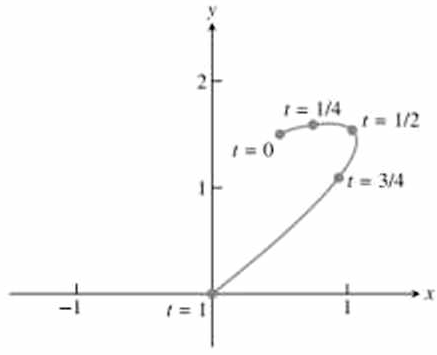


Figura 5.6 Curva parametrizada dada por la spline de Bézier. Por lo general, los intervalos iguales del parámetro t no dividen la trayectoria en segmentos de igual longitud.

que es la curva de Bézier definida por los cuatro puntos (0.5, 1.5), (0.6, 1.6), (2, 2), (0, 0) (vea la sección 3.5). Se muestran los puntos definidos para valores igualmente espaciados en $t = 0$, 1/4, 1/2, 3/4, 1. Observe que un espaciado uniforme de los valores en t no implica un espaciado uniforme en la longitud del arco. El objetivo es aplicar los métodos de integración para dividir esta trayectoria en n partes iguales.

Recuerde del cálculo que la longitud del arco de una trayectoria desde t_1 hasta t_2 es

$$\int_{t_1}^{t_2} \sqrt{x'(t)^2 + y'(t)^2}\, dt.$$

Sólo en raras ocasiones la integral produce una expresión en forma cerrada, y normalmente se utiliza una técnica de cuadratura adaptativa para controlar la parametrización de la ruta.

Actividades sugeridas:

1. Escriba una función en MATLAB que utilice la cuadratura adaptativa para calcular la longitud del arco desde $t = 0$ hasta $t = T$ para una $T \leq 1$ dada.

2. Escriba un programa que, para cualquier s de entrada entre 0 y 1, encuentre el parámetro $t^*(s)$ que sea la s de la forma a lo largo de la curva. En otras palabras, la longitud del arco desde $t = 0$ hasta $t = t^*(s)$ dividida entre la longitud del arco desde $t = 0$ hasta $t = 1$ debe ser igual a s. Utilice el método de bisección para localizar el punto $t^*(s)$ hasta tres cifras decimales correctas. ¿Qué función se establece en cero? ¿Qué intervalo de confinamiento debe utilizarse para iniciar el método de bisección?

3. Divida uniformemente la trayectoria de la figura 5.6 en n partes de igual longitud, para $n = 4$ y $n = 20$. Grafique los análogos de la figura 5.6 que muestren las subdivisiones. Si sus cálculos son demasiado lentos, considere acelerar la cuadratura adaptativa mediante la regla de Simpson, como se sugiere en el problema de computadora 2 de la sección 5.4.

4. Cambie el método de bisección del paso 2 por el método de Newton, y repita los pasos 2 y 3. ¿Cuál es la derivada necesaria?, ¿cuál es una buena opción para el valor inicial? ¿El tiempo de cálculo se reduce con esta sustitución?

5. En el apéndice A se muestran los comandos de animación disponibles en MATLAB. Por ejemplo, los comandos

```
set(gca,'XLim',[-2 2],'YLim',[-2 2],'Drawmode','fast',...
   'Visible','on');
cla
axis square
```

```
ball=line('color','r','Marker','o','MarkerSize',10,...
   'LineWidth',2, 'erase','xor','xdata',[],'ydata',[]);
```

defina un objeto "ball" (bola) con la posición asignada (x, y) mediante los siguientes comandos:

```
set(ball,'xdata',x,'ydata',y); drawnow;pause(0.01)
```

Al poner esta línea en un ciclo que cambie x y y hace que la bola se mueva a lo largo de la trayectoria en la ventana de la figura de MATLAB.

Utilice los comandos de animación de MATLAB para demostrar el viaje a lo largo de la trayectoria, primero con el parámetro original de velocidad $0 \le t \le 1$ y luego a velocidad (constante) dada por $t^*(s)$ para $0 \le s \le 1$.

6. Experimente con la subdivisión de una trayectoria de su elección. Construya una curva de Bézier de su elección con diseño, inicio, etcétera; divídala en segmentos con la misma longitud de arco y anímela como en el paso 5.

7. Escriba un programa que intercepte la trayectoria P de acuerdo con una **curva de progreso** arbitraria $C(s)$, $0 \le s \le 1$, con $C(0) = 0$ y $C(1) = 1$. El objeto debe moverse a lo largo de la curva C de modo que la proporción $C(s)$ de longitud de arco total de la trayectoria cruce entre 0 y s. Por ejemplo, la velocidad constante a lo largo de la trayectoria podría estar representada por $C(s) = s$. Pruebe la curvas de progreso $C(s) = s^{1/3}$, $C(s) = s^2$, $C(s) = \text{sen } s\pi/2$ o $C(s) = 1/2 + (1/2)\text{sen}(2s - 1)\pi/2$.

Consulte Wang *et al.* [2003] y Guenter y Parent [1990] para obtener más detalles y aplicaciones de la integración numérica de curvas en el plano y en el espacio. ✔

Software y lecturas adicionales

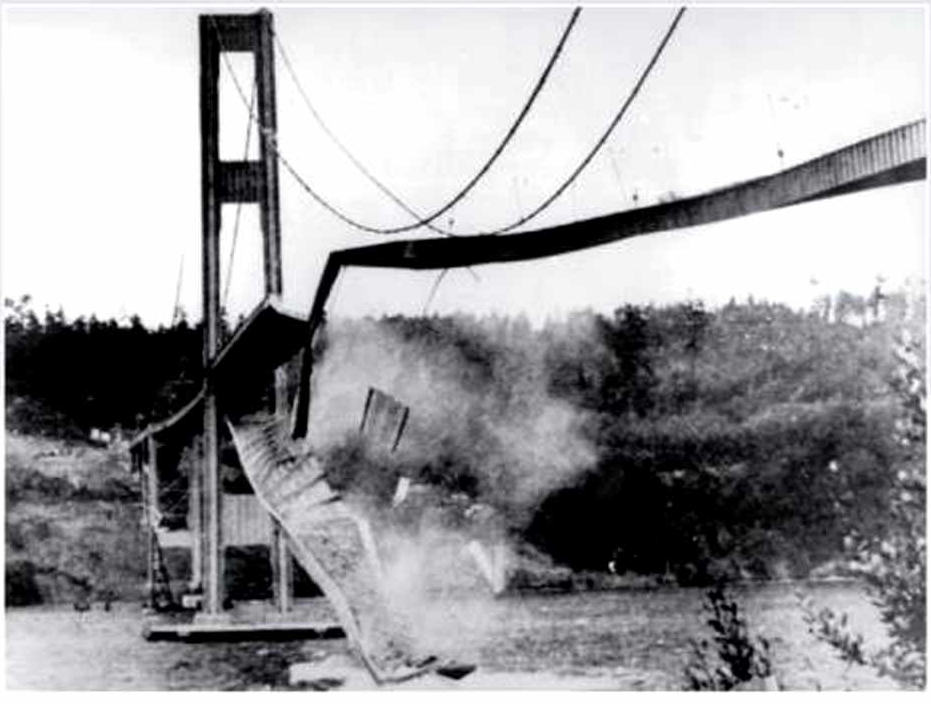
Los métodos de Newton-Cotes abiertos y cerrados son herramientas básicas para aproximar integrales definidas. La integración de Romberg es una versión acelerada. Las implementaciones de software más comerciales involucran la integración numérica en alguna de sus formas. Los textos clásicos sobre la diferenciación y la integración numérica incluyen Davis y Rabinowitz [1984], Stroud y Secrest [1966], Krommer y Ueberhuber [1998], Engels [1980], y Evans [1993].

Existen muchas técnicas de integración numérica eficaces que se implementan mediante subrutinas de Fortran en el software de dominio público Quadpack (Piessens *et al.* [1983]), disponible en Netlib (`www.netlib.org/quadpack`). El método de Gauss-Kronrod es una técnica adaptativa que se basa en la cuadratura gaussiana. Quadpack proporciona los métodos adaptativo y no adaptativo QNG y QAG, respectivamente; el último se basa en Gauss-Kronrod. Los programas en IMSL y NAG se basan en las subrutinas de Quadpack. Por ejemplo, la clase `quadrature` en IMSL es la implementación en Java.

El comando `quad` de MATLAB es una implementación de la cuadratura adaptativa compuesta de Simpson, y `dblquad` trata con integrales dobles. La caja de herramientas simbólica de MATLAB tiene los comandos `diff` e `int` para diferenciar e integrar en forma simbólica, respectivamente.

La integración de funciones con varias variables puede hacerse al extender los métodos unidimensionales en forma directa, siempre y cuando la región de integración sea simple; por ejemplo, vea Davis y Rabinowitz [1984] y Haber [1970]. Para algunas regiones complicadas, se recomienda la integración Monte Carlo. Monte Carlo es más fácil de implementar, pero por lo general converge con mayor lentitud. Estos temas se analizan en el capítulo 9.

CAPÍTULO



6 Ecuaciones diferenciales ordinarias

El 7 de noviembre de 1940, el puente Tacoma Narrows, el tercer puente colgante más largo del mundo, se hizo famoso por sus pronunciadas oscilaciones verticales durante los vendavales. Alrededor de las 11 A.M. de ese día, entró en resonancia.

Pero el movimiento que precedió al colapso fue principalmente torsional, moviéndose de lado a lado. Este movimiento, que casi nunca había sido visto antes de ese día, duró 45 minutos antes de colapsar. Con el tiempo, el movimiento de torsión se volvió lo suficientemente grande para romper un cable de soporte y el puente se desintegró con rapidez.

El debate entre los arquitectos e ingenieros sobre el motivo de la caída no ha cesado desde entonces. Los fuertes vientos causaron la oscilación vertical por razones aerodinámicas, con el puente actuando como un ala de avión, pero la integridad del puente no estaba en peligro por los movimientos estrictamente verticales. El misterio es cómo surgió la oscilación torsional.

Comprobación en la realidad En la página 322 se propone un modelo de ecuaciones diferenciales que explora los posibles mecanismos para la oscilación torsional.

Una ecuación diferencial es una ecuación que involucra derivadas. En la forma

$$y'(t) = f(t, y(t)),$$

una ecuación diferencial de primer orden expresa la tasa de cambio de una cantidad y en términos del tiempo presente y el valor presente de la inversión de una cantidad de dinero. Las ecuaciones diferenciales se utilizan para modelar, comprender y predecir los sistemas que cambian con el tiempo.

Una gran mayoría de las ecuaciones interesantes no tienen solución en forma directa, lo que hace que las aproximaciones sean el único recurso. Este capítulo trata de la solución aproximada de ecuaciones diferenciales ordinarias (EDO) mediante métodos computacionales. Después de las

ideas de introducción sobre las ecuaciones diferenciales, se describe y analiza con detalle el método de Euler. Aunque es demasiado simple como para usarse a gran escala en las aplicaciones, el método de Euler es fundamental, puesto que la mayoría de los aspectos importantes del tema pueden entenderse con facilidad en su sencillo contexto.

Después se presentan métodos más sofisticados y se exploran ejemplos interesantes de los sistemas de ecuaciones diferenciales. Los protocolos con tamaño de paso variable son importantes para la solución eficiente de los problemas de rigidez, los cuales requieren métodos especiales. El capítulo termina con una introducción a los métodos implícitos y de pasos múltiples.

6.1 PROBLEMAS DE VALOR INICIAL

Muchas leyes físicas que han tenido éxito en el modelado de la naturaleza se expresan en la forma de ecuaciones diferenciales. Sir Isaac Newton escribió sus leyes del movimiento en esa forma: $F = ma$ es una ecuación que implica la fuerza que actúa sobre un objeto y la aceleración del objeto, que es la segunda derivada de la posición. De hecho, la postulación de las leyes de Newton, junto con el desarrollo de la infraestructura necesaria para escribirlas (cálculo), constituyó una de las revoluciones más importantes en la historia de la ciencia.

Un modelo sencillo conocido como la **ecuación logística** modela la tasa de cambio de una población como

$$y' = cy(1 - y), \tag{6.1}$$

donde y' denota la derivada con respecto al tiempo t. Si se piensa en y como la representación de la población como una proporción de la capacidad de carga del hábitat del animal, entonces se espera que crezca hasta cerca de esa capacidad y después se estabilice. La ecuación diferencial (6.1) muestra la tasa de cambio y' como proporcional al producto de la población actual y y la "capacidad restante" $1 - y$. Por lo tanto, la tasa de cambio es pequeña, tanto cuando la población es pequeña (y cercana a 0) como cuando la población se acerca a la capacidad (y cercana a 1).

La ecuación diferencial ordinaria (6.1) es típica ya que tiene un número infinito de soluciones $y(t)$. Al especificar una condición inicial, es posible identificar cuál de las familias infinitas es de interés (en la siguiente sección se obtendrá más precisión acerca de la existencia y la unicidad). Un **problema de valor inicial** para una ecuación diferencial ordinaria de primer orden es la ecuación junto con una condición inicial en un intervalo específico $a \le t \le b$:

$$\begin{cases} y' = f(t, y) \\ y(a) = y_a \\ t \text{ en } [a, b] \end{cases}. \tag{6.2}$$

Será de gran ayuda pensar en una ecuación diferencial como en un campo de pendientes, como se muestra en la figura 6.1(a). La ecuación (6.1) puede verse como la especificación de una pendiente para cualesquiera valores actuales de (t, y). Si se utiliza una flecha para graficar la pendiente en cada punto del plano, se obtiene el **campo de pendientes**, o **campo de direcciones** de la ecuación diferencial. La ecuación es **autónoma** si el lado derecho $f(t, y)$ es independiente de t. Esto es evidente en la figura 6.1.

Cuando una condición inicial se especifica en un campo de pendientes, es posible identificar alguna solución en la familia infinita de soluciones. En la figura 6.1(b) se grafican dos soluciones diferentes a partir de dos valores iniciales diferentes, $y(0) = 0.2$ y $y(0) = 1.4$, respectivamente.

La ecuación (6.1) tiene una solución que puede escribirse en términos de funciones elementales. Al diferenciar y sustituir se comprueba que, siempre y cuando la condición inicial $y_0 \neq 1$,

$$y(t) = 1 - \frac{1}{1 + \frac{y_0}{1 - y_0} e^{ct}} \tag{6.3}$$

es la solución del problema de valor inicial

$$\begin{cases} y' = cy(1 - y) \\ y(0) = y_0 \\ t \text{ en } [0, T] \end{cases}. \tag{6.4}$$

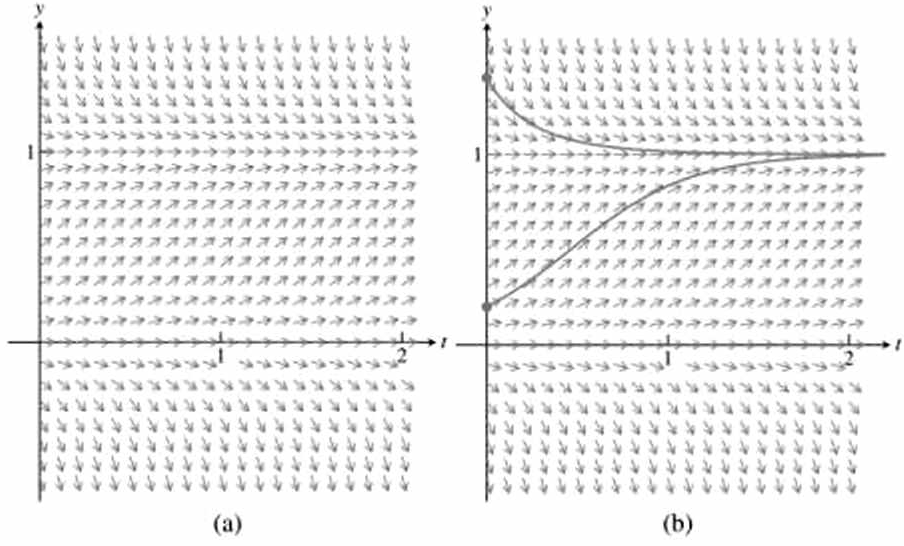


Figura 6 1 La ecuación diferencial logística. (a) El campo de pendientes varía en la dirección *y* pero es constante para toda *t*, la definición de una ecuación autónoma. (b) Dos soluciones de la ecuación diferencial.

La solución sigue las flechas de la figura 6.1(b). Si $y_0 = 1$, la solución es $y(t) = 1$, que se verifica de la misma manera.

6.1.1 Método de Euler

La ecuación logística tenía una solución explícita bastante simple. Un escenario mucho más común es una ecuación diferencial con ninguna fórmula de solución explícita. La geometría de la figura 6.1 sugiere un enfoque alternativo: "resolver" computacionalmente la ecuación diferencial siguiendo las flechas. Comience en la condición inicial (t_0, y_0) y siga la dirección especificada allí. Después desplácese una distancia corta, vuelva a evaluar la pendiente en el punto nuevo (t_1, y_1), aléjese aún más de acuerdo con la nueva pendiente y repita el proceso. Habrá algún error asociado con el proceso, puesto que, entre las evaluaciones de la pendiente, no se moverá a lo largo de una pendiente totalmente exacta. Pero si las pendientes cambian poco a poco, es posible obtener una buena aproximación a la solución del problema de valor inicial.

► **EJEMPLO 6.1** Dibuje el campo de pendientes del problema de valor inicial

$$\begin{cases} y' = ty + t^3 \\ y(0) = y_0 \\ t \text{ en } [0,1] \end{cases}. \tag{6.5}$$

En la figura 6.2(a) se muestra el campo de pendientes. Para cada punto (t, y) en el plano, se grafica una flecha con pendiente igual a $ty + y^3$. Este problema de valor inicial no es autónomo porque t aparece de manera explícita del lado derecho de la ecuación. Lo anterior también queda claro a partir del campo de pendientes, el cual varía de acuerdo con t y con y. Se muestra la solución exacta $y(t) = 3e^{t^2/2} - t^2 - 2$ para la condición inicial $y(0) = 1$. Vea el ejemplo 6.6 para la obtención de la solución explícita. ◄

En la figura 6.2(b) se muestra una implementación del método para seguir computacionalmente el campo de pendientes, que se conoce como método de Euler. Se inicia con una malla de $n + 1$ puntos

$$t_0 < t_1 < t_2 < \cdots < t_n$$

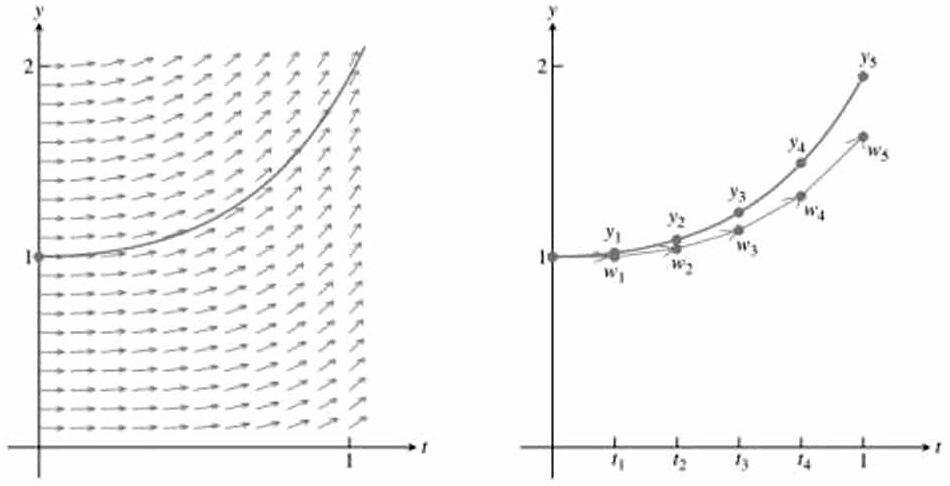


Figura 6.2 Solución del problema de valor inicial (6.5). (a) El campo de pendientes para una ecuación no autónoma varía con t. Se muestra la solución que satisface $y(0) = 1$. (b) Aplicación del método de Euler a la ecuación, con tamaño de paso $h = 0.2$.

a lo largo del eje t con el mismo tamaño de paso h. En la figura 6.2(b), se seleccionaron los valores de t como

$$t_0 = 0.0 \quad t_1 = 0.2 \quad t_2 = 0.4 \quad t_3 = 0.6 \quad t_4 = 0.8 \quad t_5 = 1.0 \tag{6.6}$$

con tamaño de paso $h = 0.2$.

Comenzando con $w_0 = y_0$. Al seguir el campo de pendientes en cada t_i se obtiene la aproximación

$$w_{i+1} = w_i + hf(t_i, w_i)$$

en t_{i+1}, puesto que $f(t_i, w_i)$ representa la pendiente de la solución. Observe que el cambio en y es la distancia horizontal h multiplicada por la pendiente. Como se muestra en la figura 6.2(b), cada w_i es una aproximación a la solución en t_i.

La fórmula para este método puede expresarse de la siguiente manera:

Método de Euler

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + hf(t_i, w_i). \end{aligned} \tag{6.7}$$

► EJEMPLO 6.2 Aplique el método de Euler al problema de valor inicial (6.5), con la condición inicial $y_0 = 1$.

El lado derecho de la ecuación diferencial es $f(t, y) = ty + t^3$. Por lo tanto, el método de Euler será la iteración

$$\begin{aligned} w_0 &= 1 \\ w_{i+1} &= w_i + h(t_i w_i + t_i^3). \end{aligned} \tag{6.8}$$

Si se usa la malla (6.6) con tamaño de paso $h = 0.2$, se calcula de manera iterativa la solución aproximada de (6.8). Los valores w_i dados por el método de Euler, y graficados en la figura 6.2(b), se comparan con los valores verdaderos de y_i, en la siguiente tabla:

iteración	t_i	w_i	y_i	e_i
0	0.0	1.0000	1.0000	0.0000
1	0.2	1.0000	1.0206	0.0206
2	0.4	1.0416	1.0899	0.0483
3	0.6	1.1377	1.2317	0.0939
4	0.8	1.3175	1.4914	0.1739
5	1.0	1.6306	1.9462	0.3155

La tabla también muestra el error $e_i = |y_i - w_i|$ en cada paso. El error tiende a crecer, desde cero en la condición inicial hasta su valor más grande en el extremo del intervalo, aunque el error máximo no siempre se encuentra en dicho extremo.

La aplicación del método de Euler con tamaño de paso $h = 0.1$ hace que el error disminuya, como se hace evidente en la figura 6.3(a). Al usar de nuevo (6.8), se calculan los valores siguientes:

iteración	t_i	w_i	y_i	e_i
0	0.0	1.0000	1.0000	0.0000
1	0.1	1.0000	1.0050	0.0050
2	0.2	1.0101	1.0206	0.0105
3	0.3	1.0311	1.0481	0.0170
4	0.4	1.0647	1.0899	0.0251
5	0.5	1.1137	1.1494	0.0357
6	0.6	1.1819	1.2317	0.0497
7	0.7	1.2744	1.3429	0.0684
8	0.8	1.3979	1.4914	0.0934
9	0.9	1.5610	1.6879	0.1269
10	1.0	1.7744	1.9462	0.1718

Compare el error e_{10} para el cálculo $h = 0.1$ con el error e_5 para el cálculo $h = 0.2$. Observe que al reducir el tamaño de paso h a la mitad resulta una disminución del error en $t = 1.0$ aproximadamente de la mitad. ◀

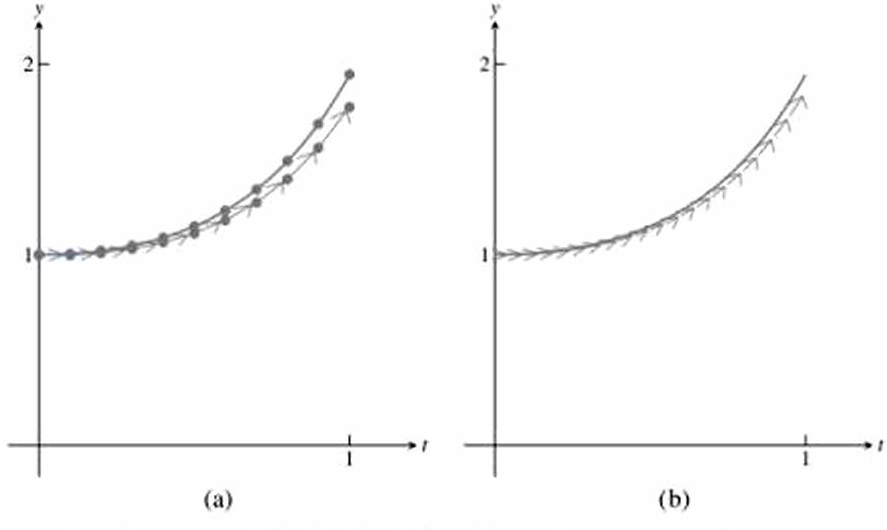


Figura 6.3 Método de Euler aplicado al PVI (6.5). Las flechas muestran los pasos de Euler, exactamente como en la figura 6.2, excepto por el tamaño del paso. (a) Diez pasos de tamaño de $h = 0.1$ (b) Veinte pasos de tamaño $h = 0.05$.

En el siguiente código de MATLAB se implementa el método de Euler, el cual se ha escrito en forma modular para resaltar los tres componentes individuales. El programa de graficado invoca a un subprograma para ejecutar cada paso de Euler, el cual a su vez llama a la función f que contiene el lado derecho de la ecuación diferencial. En esta forma, después será fácil intercambiar el lado derecho por otra ecuación diferencial y el método de Euler por otro método más sofisticado. A continuación se presenta el código:

```
%Programa 6.1 Método de Euler para resolver problemas de valor inicial
%Uso con ydot.m para evaluar el lado derecho de la ecuación diferencial
% Entrada: intervalo inter, valor inicial y0, número de pasos n
% Salida: tiempo de pasos t, solución y
% Uso de ejemplo: euler([0 1],1,10);
function [t,y]=euler(inter,y0,n)
t(1)=inter(1); y(1)=y0;
h=(inter(2)-inter(1))/n;
for i=1:n
  t(i+1)=t(i)+h;
  y(i+1)=eulerstep(t(i),y(i),h);
end
plot(t,y)

function y=eulerstep(t,y,h)
%un paso del método de Euler
%Entrada: tiempo actual t, valor actual y, tamaño de paso h
%Salida: valor aproximado de la solución en el tiempo t+h
y=y+h*ydot(t,y);

function z=ydot(t,y)
%lado derecho de la ecuación diferencial
z=t*y+t^3;
```

Si se compara la aproximación del método de Euler para (6.5) con la solución exacta en $t = 1$, se obtiene la siguiente tabla que extiende los resultados previos para $n = 5$ y 10:

n iteraciones	tamaño de paso h	error en $t = 1$
5	0.20000	0.3155
10	0.10000	0.1718
20	0.05000	0.0899
40	0.02500	0.0460
80	0.01250	0.0233
160	0.00625	0.0117
320	0.00312	0.0059
640	0.00156	0.0029

En la tabla y en las figuras 6.3 y 6.4 se observan dos hechos evidentes. En primer lugar, el error es distinto de cero. Como el método de Euler realiza pasos no infinitesimales, la pendiente cambia a lo largo del paso y la aproximación no se encuentra exactamente en la curva de solución. En segundo lugar, el error disminuye a medida que el tamaño de paso se reduce, como puede verse también en la figura 6.3. De la tabla se deduce que el error es proporcional a h; este hecho se confirmará en la siguiente sección.

► **EJEMPLO 6.3** Encuentre la fórmula del método de Euler para el siguiente problema de valor inicial:

$$\begin{cases} y' = cy \\ y(0) = y_0 \\ t \text{ en } [0, 1] \end{cases}. \tag{6.9}$$

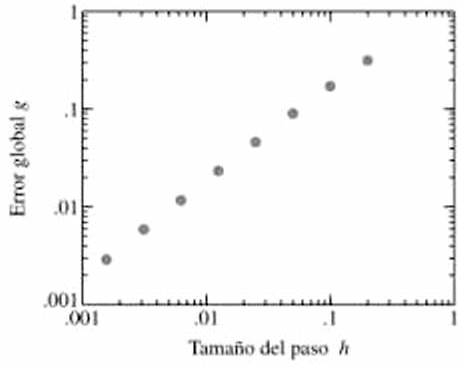


Figura 6.4 Error en función del tamaño de paso para el método de Euler. La diferencia entre la solución aproximada de (6.5) y la solución correcta en $t = 1$ tiene una pendiente de 1 en una gráfica log-log, por lo que es proporcional al tamaño de paso h, para una h pequeña.

Para $f(t, y) = cy$ donde c es una constante, el método de Euler da

$$w_0 = y_0$$
$$w_{i+1} = w_i + hcw_i = (1 + hc)w_i \text{ para } i = 1, 2, 3, \ldots. \quad \blacktriangleleft$$

La solución exacta de la ecuación $y' = cy$ puede encontrarse mediante el método de **separación de variables**. Suponiendo que $y \neq 0$, divida ambos lados entre y, separe variables e integre de la manera siguiente:

$$\frac{dy}{y} = c\,dt$$
$$\ln|y| = ct + k.$$
$$|y| = e^{ct+k} = e^k e^{ct}$$

La condición inicial $y(0) = y_0$ implica $y = y_0 e^{ct}$.

En este caso simple, puede mostrarse que el método de Euler converge a la solución correcta cuando el número de pasos $n \to \infty$. Tenga en cuenta que

$$w_i = (1 + hc)w_{i-1} = (1 + hc)^2 w_{i-2} = \cdots = (1 + hc)^i w_0.$$

Para una t fija, establezca el tamaño de paso $h = t/n$ para un entero n. Entonces el valor aproximado en t es

$$w_n = (1 + hc)^n y_0$$
$$= \left(1 + \frac{ct}{n}\right)^n y_0.$$

La fórmula clásica dice que

$$\lim_{n\to\infty} \left(1 + \frac{ct}{n}\right)^n = e^{ct},$$

lo que demuestra que, cuando $n \to \infty$, el método de Euler converge a la solución correcta.

6.1.2 Existencia, unicidad y continuidad de las soluciones

En esta sección se proporcionan algunos antecedentes teóricos para los métodos de solución para problemas de valores iniciales. Antes de empezar a calcular la solución de un problema, es útil saber que (1) la solución existe y (2) sólo hay una solución, de modo que el algoritmo no se confunda

con cuál solución calcular. En circunstancias adecuadas, los problemas de valores iniciales tienen exactamente una solución.

DEFINICIÓN 6.1 Una función $f(t, y)$ es **continua de Lipschitz** en la variable y sobre el rectángulo $S = [a, b] \times [\alpha, \beta]$ si existe una constante L (llamada la **constante de Lipschitz**) que satisface

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|$$

para cada $(t, y_1), (t, y_2)$ en S.

Una función que es continua de Lipschitz en y es continua en y, pero no necesariamente diferenciable.

► EJEMPLO 6.4 Encuentre la constante de Lipschitz para el lado derecho $f(t, y) = ty + t^3$ de (6.5).

La función $f(t, y) = ty + t^3$ es continua de Lipschitz en la variable y sobre el conjunto $0 \leq t \leq 1$, $-\infty < y < \infty$. Compruebe que

$$|f(t, y_1) - f(t, y_2)| = |ty_1 - ty_2| \leq |t||y_1 - y_2| \leq |y_1 - y_2| \tag{6.10}$$

en el conjunto. La constante de Lipschitz es $L = 1$. ◄

Aunque la definición 6.1 especifica el conjunto S como un rectángulo, de manera más general S puede ser un conjunto **convexo**, que contiene al segmento de línea que une a cualesquiera dos puntos en el conjunto. Si la función f es continuamente diferenciable en la variable y, el valor máximo absoluto de la derivada parcial $\partial f/\partial y$ es una constante de Lipschitz. De acuerdo con el teorema del valor medio, para cada t fija, hay una c entre y_1 y y_2 tal que

$$\frac{f(t, y_1) - f(t, y_2)}{y_1 - y_2} = \frac{\partial f}{\partial y}(t, c).$$

Por lo tanto, L puede tomarse como el máximo de

$$\left|\frac{\partial f}{\partial y}(t, c)\right|$$

en el conjunto.

La hipótesis de continuidad de Lipschitz garantiza la existencia y unicidad de las soluciones a problemas de valores iniciales. En referencia a Birkhoff y Rota [1989] se demostrará el teorema siguiente:

TEOREMA 6.2 Suponga que $f(t, y)$ es continua de Lipschitz en la variable y sobre el conjunto $[a, b] \times [\alpha, \beta]$ y que $\alpha < y_a < \beta$. Entonces existe una c entre a y b de tal forma que el problema de valor inicial

$$\begin{cases} y' = f(t, y) \\ y(a) = y_a \\ t \text{ en } [a, c] \end{cases} \tag{6.11}$$

tiene exactamente una solución $y(t)$. Por otra parte, si f es continua de Lipschitz en $[a, b] \times (-\infty, \infty)$, entonces existe exactamente una solución en $[a, b]$. ■

Es importante realizar una lectura cuidadosa del teorema 6.2, sobre todo si el objetivo es calcular la solución de manera numérica. El hecho de que el problema de valor inicial satisfaga una condición de Lipschitz en $[a, b] \times [\alpha, \beta]$ que contiene la condición inicial, no garantiza una

solución para t en *todo el intervalo* $[a, b]$. La razón simple es que la solución puede estar fuera del rango de y en $[\alpha, \beta]$ para el cual la constante de Lipschitz es válida. Lo mejor que puede decirse es que la solución existe en algún intervalo más corto $[a, c]$. Este punto se ilustra mediante el siguiente ejemplo:

► **EJEMPLO 6.5** ¿En qué intervalos $[0, c]$ el problema de valor inicial tiene una solución única?

$$\begin{cases} y' = y^2 \\ y(0) = 1 \\ t \text{ en } [0, 2]. \end{cases} \tag{6.12}$$

La derivada parcial de f respecto a y es $2y$. La constante de Lipschitz máx $|2y| = 20$ es válida en el conjunto $0 \le t \le 2$, $-10 \le y \le 10$. El teorema 6.2 garantiza una solución iniciando en $t = 0$ y existente en algún intervalo $[a, c]$ para $c > 0$, pero no garantiza una solución en todo el intervalo $[0, 2]$.

De hecho, la solución única de la ecuación diferencial (6.12) es $y(t) = 1/(1 - t)$, que puede encontrarse mediante la separación de variables. Esta solución tiende a infinito cuando t se aproxima a 1. En otras palabras, la solución existe en el intervalo $0 \le t \le c$ para cualquier $0 < c < 1$, pero no para cualquier c más grande. En este ejemplo se explica la función de c en el teorema 6.2: la constante de Lipschitz 20 es válida para $|y| \le 10$, pero la solución y supera a 10 antes de que t llegue a 2. ◄

El teorema 6.3 es el hecho básico de la estabilidad (amplificación del error) para ecuaciones diferenciales ordinarias. Si una constante de Lipschitz existe para el lado derecho de la ecuación diferencial, entonces la solución se vuelve una función de Lipschitz del valor inicial, con una nueva constante de Lipschitz que es exponencial en la función original. Ésta es una versión de la desigualdad de Gronwall.

TEOREMA 6.3 Suponga que $f(t, y)$ es continua de Lipschitz en la variable y sobre el conjunto $S = [a, b] \times [\alpha, \beta]$. Si $Y(t)$ y $Z(t)$ son soluciones en S de la ecuación diferencial

$$y' = f(t, y)$$

con las condiciones iniciales $Y(a)$ y $Z(a)$, respectivamente, entonces

$$|Y(t) - Z(t)| \le e^{L(t-a)} |Y(a) - Z(a)|. \tag{6.13}$$

■

Demostración. Si $Y(a) = Z(a)$, entonces $Y(t) = Z(t)$ por la unicidad de las soluciones, y (6.13) es trivialmente satisfecha. Es posible suponer que $Y(a) \neq Z(a)$, en cuyo caso $Y(t) \neq Z(t)$ para toda t en el intervalo, para evitar contradecir la unicidad.

Defina $u(t) = Y(t) - Z(t)$. Dado que $u(t)$ es estrictamente positiva o bien estrictamente negativa, y porque (6.13) sólo depende de $|u|$, puede suponerse que $u > 0$. Entonces $u(a) = Y(a) - Z(a)$,

ANOTACIÓN **Condicionamiento** La magnificación del error se analizó en los capítulos 1 y 2 como una forma de cuantificar los efectos de los cambios pequeños en las entradas sobre la solución. El análogo de esta cuestión para los problemas de valores iniciales recibe una respuesta precisa mediante el teorema 6.3. Cuando la condición inicial (los datos de entrada) $Y(a)$ se cambia a $Z(a)$, el mayor cambio posible en la salida t unidades de tiempo después, $Y(t) - Z(t)$, es exponencial en t y lineal en la diferencia de la condición inicial. Esto último implica que puede hablarse de un "número de condición" igual a $e^{L(t-a)}$ durante un tiempo fijo t.

y la derivada es $u'(t) = Y'(t) - Z'(t) = f(t, Y(t)) - f(t, Z(t))$. La condición de Lipschitz implica que

$$u' = |f(t, Y) - f(t, Z)| \leq L|Y(t) - Z(t)| = L|u(t)| = Lu(t),$$

y por lo tanto $(\ln u)' = u'/u \leq L$. Por el teorema del valor medio,

$$\frac{\ln u(t) - \ln u(a)}{t - a} \leq L,$$

que se simplifica como

$$\ln \frac{u(t)}{u(a)} \leq L(t - a)$$
$$u(t) \leq u(a)e^{L(t-a)}.$$

Éste es el resultado deseado. ❐

De regreso al ejemplo 6.4, el teorema 6.3 implica que las soluciones $Y(t)$ y $Z(t)$, a partir de valores iniciales diferentes, no deben crecer por separado más rápido que un factor multiplicativo de e^t para $0 \leq t \leq 1$. De hecho, la solución en el valor inicial Y_0 es $Y(t) = (2 + Y_0)e^{t^2/2} - t^2 - 2$, y entonces la diferencia entre las dos soluciones es

$$\begin{aligned} |Y(t) - Z(t)| &\leq |(2 + Y_0)e^{t^2/2} - t^2 - 2 - ((2 + Z_0)e^{t^2/2} - t^2 - 2)| \\ &\leq |Y_0 - Z_0|e^{t^2/2}, \end{aligned} \tag{6.14}$$

que es menor que $|Y_0 - Z_0|e^t$ para $0 \leq t \leq 1$, según lo enunciado por el teorema 6.3.

6.1.3 Ecuaciones lineales de primer orden

Una clase especial de ecuaciones diferenciales ordinarias que pueden resolverse con facilidad proporciona un conjunto práctico de ejemplos ilustrativos. Son las ecuaciones de primer orden cuyos lados derechos son lineales en la variable y. Considere el problema de valor inicial

$$\begin{cases} y' = g(t)y + h(t) \\ y(a) = y_a \\ t \text{ en } [a, b] \end{cases}. \tag{6.15}$$

Primero observe que si $g(t)$ es continua en $[a, b]$, existe una solución única por el teorema 6.2, usando $L = \max_{[a,b]} g(t)$ como la constante de Lipschitz. La solución se encuentra mediante un truco, multiplicar toda la ecuación por un "factor de integración".

El factor de integración es $e^{-\int g(t)\,dt}$. Al multiplicar ambos lados por éste se obtiene

$$\begin{aligned} (y' - g(t)y)e^{-\int g(t)\,dt} &= e^{-\int g(t)\,dt}h(t) \\ \left(ye^{-\int g(t)\,dt}\right)' &= e^{-\int g(t)\,dt}h(t) \\ ye^{-\int g(t)\,dt} &= \int e^{-\int g(t)\,dt}h(t)\,dt, \end{aligned}$$

que puede resolverse como

$$y(t) = e^{\int g(t)\,dt} \int e^{-\int g(t)\,dt}h(t)\,dt. \tag{6.16}$$

Si el factor de integración puede expresarse de manera simple, este método permite una solución explícita de la ecuación lineal de primer orden (6.15).

► EJEMPLO 6.6 Resuelva la ecuación diferencial lineal de primer orden

$$\begin{cases} y' = ty + t^3 \\ y(0) = y_0 \end{cases}. \tag{6.17}$$

El factor de integración es

$$e^{-\int g(t)\,dt} = e^{-\frac{t^2}{2}}.$$

De acuerdo con (6.16), la solución es

$$\begin{aligned} y(t) &= e^{\frac{t^2}{2}} \int e^{-\frac{t^2}{2}} t^3\,dt \\ &= e^{\frac{t^2}{2}} \int e^{-u}(2u)\,du \\ &= 2e^{\frac{t^2}{2}} \left[-\frac{t^2}{2} e^{-\frac{t^2}{2}} - e^{-\frac{t^2}{2}} + C \right] \\ &= -t^2 - 2 + 2Ce^{\frac{t^2}{2}}, \end{aligned}$$

donde se hizo la sustitución $u = t^2/2$. Al despejar la constante de integración C se obtiene $y_0 = -2 + 2C$, por lo que $C = (2 + y_0)/2$. Por lo tanto,

$$y(t) = (2 + y_0)e^{\frac{t^2}{2}} - t^2 - 2. \quad \blacktriangleleft$$

6.1 Ejercicios

1. Demuestre que la función $y(t) = t \operatorname{sen} t$ es una solución de las ecuaciones diferenciales
   (a) $y + t^2 \cos t = ty'$ (b) $y'' = 2\cos t - y$ (c) $t(y'' + y) = 2y' - 2\operatorname{sen} t$.

2. Demuestre que la función $y(t) = e^{\operatorname{sen} t}$ es una solución de los problemas de valores iniciales
   (a) $y' = y\cos t$, $y(0) = 1$ (b) $y'' = (\cos t)y' - (\operatorname{sen} t)y$, $y(0) = 1$, $y'(0) = 1$
   (c) $y'' = y(1 - \ln y - (\ln y)^2)$, $y(\pi) = 1$, $y'(\pi) = -1$.

3. Use la separación de variables para encontrar las soluciones del PVI dado por $y(0) = 1$ y las siguientes ecuaciones diferenciales:

   (a) $y' = t$ (b) $y' = t^2 y$ (c) $y' = 2(t + 1)y$

   (d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

4. Encuentre las soluciones del PVI dado por $y(0) = 0$ y las siguientes ecuaciones diferenciales lineales de primer orden:

   (a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

5. Aplique el método de Euler con tamaño de paso $h = 1/4$ a los PVI del ejercicio 3 en el intervalo $[0, 1]$. Enliste las w_i, $i = 0, \ldots, 4$, y encuentre el error en $t = 1$, comparando su respuesta con la solución correcta.

6. Aplique el método de Euler con tamaño de paso $h = 1/4$ a los PVI del ejercicio 4 en el intervalo $[0, 1]$. Encuentre el error en $t = 1$ al comparar su respuesta con la solución correcta.

7. (a) Muestre que $y = \tan(t + c)$ es una solución de la ecuación diferencial $y' = 1 + y^2$ para cada c. (b) Para cada número real y_0, encuentre c en el intervalo $(-\pi/2, \pi/2)$ de manera que el problema del valor inicial $y' = 1 + y^2$, $y(0) = y_0$ tenga una solución $y = \tan(t + c)$.

8. (a) Muestre que $y = \tanh(t + c)$ es una solución de la ecuación diferencial $y' = 1 - y^2$ para cada c. (b) Para cada número real y_0 en el intervalo $(-1, 1)$, encuentre c tal que el problema de valor inicial $y' = 1 - y^2$, $y(0) = y_0$ tenga una solución $y = \tanh(t + c)$.

9. ¿Para cuál de estos problemas de valores iniciales en [0, 1] el teorema 6.2 garantiza una solución única? Encuentre las constantes de Lipschitz si existen (a) $y' = t$ (b) $y' = y$ (c) $y' = -y$ (d) $y' = -y^3$.

10. Dibuje el campo de pendientes para las ecuaciones diferenciales del ejercicio 9 y encuentre aproximaciones a las soluciones, a partir de las condiciones iniciales $y(0) = 1$, $y(0) = 0$, y $y(0) = -1$.

11. Encuentre las soluciones de los problemas de valores iniciales del ejercicio 9. Para cada ecuación, utilice las constantes de Lipschitz del ejercicio 9 y verifique, si es posible, la desigualdad del teorema 6.3 para el par de soluciones con condiciones iniciales $y(0) = 0$ y $y(0) = 1$.

12. (a) Demuestre que si $a \neq 0$, la solución del problema de valor inicial $y' = ay + b$, $y(0) = y_0$ es $y(t) = (b/a)(e^{at} - 1) + y_0 e^{at}$. (b) Verifique la desigualdad del teorema 6.3 para las soluciones $y(t)$, $z(t)$ con valores iniciales y_0 y z_0, respectivamente.

13. Use la separación de variables para resolver el problema de valor inicial $y' = y^2$, $y(0) = 1$.

14. Encuentre la solución del problema de valor inicial $y' = ty^2$ con $y(0) = 1$. ¿Cuál es el mayor intervalo $[0, b]$ para el que existe la solución?

15. Considere el problema de valor inicial $y' = \operatorname{sen} y$, $y(a) = y_a$ en $a \leq t \leq b$.

    (a) ¿En qué subintervalo de $[a, b]$ el teorema 6.2 garantiza una solución única?

    (b) Demuestre que $y(t) = 2\arctan(e^{t-a}\tan(y_a/2)) + 2\pi[(y_a + \pi)/2\pi]$ es la solución del problema de valor inicial, donde $[\ ]$ denota la mayor función entera.

16. Considere el problema de valor inicial $y' = \operatorname{senh} y$, $y(a) = y_a$ en $a \leq t \leq b$.

    (a) ¿En qué subintervalo de $[a, b]$ el teorema 6.2 garantiza una solución única?

    (b) Demuestre que $y(t) = 2\operatorname{arctanh}(e^{t-a}\tanh(y_a/2))$ es una solución al problema de valor inicial.

    (c) ¿En qué intervalo $[a, c]$ existe la solución?

6.1 Problemas de computadora

1. Aplique el método de Euler con tamaño de paso $h = 0.1$ en [0, 1] a los problemas de valores iniciales del ejercicio 3. Imprima una tabla con los valores de t, las aproximaciones de Euler, y el error (diferencia con la solución exacta) para cada paso.

2. Grafique las soluciones aproximadas del método de Euler para los PVI del ejercicio 3 en [0, 1], con tamaños de paso $h = 0.1$, 0.05 y 0.025, junto con la solución exacta.

3. Grafique las soluciones aproximadas del método de Euler para los PVI del ejercicio 4 en [0, 1], con tamaños de paso $h = 0.1$, 0.05 y 0.025, junto con la solución exacta.

4. Para los PVI del ejercicio 3, haga una gráfica log-log del error del método de Euler en $t = 1$ como una función de $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$. Utilice el comando `loglog` de MATLAB como en la figura 6.4.

5. Para los PVI del ejercicio 4, haga una gráfica log-log del error del método de Euler en $t = 1$ como una función de $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$.

6. Para los problemas de valores iniciales del ejercicio 4, haga una gráfica log-log del error del método de Euler en el instante $t = 2$ como una función de $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$.

7. Grafique la solución aproximada del método de Euler en [0, 1] para la ecuación diferencial $y' = 1 + y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 1$, junto con la solución exacta (vea el ejercicio 7). Utilice los tamaños de paso $h = 0.1$ y 0.05.

8. Grafique la solución aproximada del método de Euler en [0, 1] para la ecuación diferencial $y' = 1 - y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = -1/2$, junto con la solución exacta (vea el ejercicio 8). Use los tamaños de paso $h = 0.1$ y 0.05.

9. Calcule la solución aproximada del método de Euler en [0, 4] para la ecuación diferencial $y' =$ sen y y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 100$, usando tamaños de paso $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$, junto con la solución exacta (vea el ejercicio 15), y haga una gráfica log-log del error en $t = 4$ como una función de h.

10. Calcule la solución aproximada del método de Euler para la ecuación diferencial $y' =$ senh y y la condición inicial (a) $y_0 = 1/4$ en el intervalo [0, 2] (b) $y_0 = 2$ en el intervalo [0, 1/4], usando tamaños de paso $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$, junto con la solución exacta (vea el ejercicio 16), y haga una gráfica log-log del error en el extremo del intervalo de tiempo como una función de h.

6.2 ANÁLISIS DEL ERROR EN LA SOLUCIÓN DE PVI

En la figura 6.4 se muestra un error continuamente decreciente en la aproximación del método de Euler como una función de la disminución del tamaño de paso decreciente para el ejemplo 6.1. ¿Esto es por lo general cierto? ¿El error puede hacerse tan pequeño como se desee, con tan sólo reducir el tamaño de paso? Una investigación cuidadosa del error en el método de Euler ilustrará las interrogantes para los solucionadores de PVI en general.

6.2.1 Error de truncamiento local y total

En la figura 6.5 se muestra una imagen esquemática del método de Euler con un paso al resolver un problema de valor inicial de la forma

$$\begin{cases} y' = f(t, y) \\ y(a) = y_a \\ t \text{ en } [a, b] \end{cases}. \tag{6.18}$$

En el paso i, el error acumulado de los pasos anteriores se conserva y tal vez se amplifica, al mismo tiempo que se añade el error de la nueva aproximación de Euler. En concreto, se define el **error de truncamiento total**

$$g_i = |w_i - y_i|$$

como la diferencia entre la aproximación a la solución de la EDO (por ejemplo, el método de Euler) y la solución correcta del problema de valor inicial. Además, se define el **error de truncamiento local**, o error de un solo paso, como

$$e_{i+1} = |w_{i+1} - z(t_{i+1})|, \tag{6.19}$$

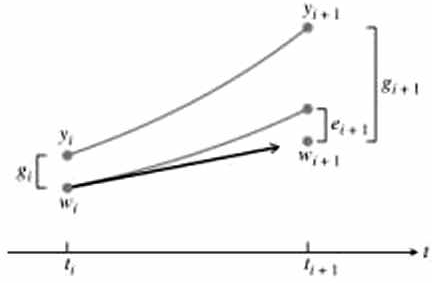


Figura 6.5 Solución de un paso de una EDO. El método de Euler sigue un segmento de línea con la pendiente del campo de vectores en el punto actual hasta el siguiente punto (t_{i+1}, w_{i+1}). La curva superior representa la solución verdadera a la ecuación diferencial. El error de truncamiento total g_{i+1} es la suma del error de truncamiento local e_{i+1} y el error acumulado de los pasos anteriores.

la diferencia entre el valor de la solución de la EDO en ese intervalo y la solución correcta del "problema de valor inicial de un solo paso"

$$\begin{cases} y' = f(t, y) \\ y(t_i) = w_i \\ t \text{ en } [t_i, t_{i+1}] \end{cases} . \tag{6.20}$$

(Se da a la solución el nombre de z, puesto que y ya se utilizó para la solución del problema del mismo valor inicial a partir de la condición inicial exacta $y(t_i) = y_i$). El error de truncamiento local es el error que se produce a partir de un solo paso, tomando la aproximación anterior w_i a la solución como punto de inicio. El error de truncamiento total es el error acumulado en los primeros pasos i. Los errores de truncamiento local y total se ilustran en la figura 6.5. En cada paso, el nuevo error total es la combinación del error de la etapa anterior y el nuevo error local. Debido a la acumulación, el error total no es sólo la suma de los errores de truncamiento locales.

► **EJEMPLO 6.7** Encuentre el error de truncamiento local para el método de Euler.

De acuerdo con la definición, éste es el nuevo error cometido por el método de Euler en un solo paso. Suponga que el paso anterior w_i es correcto, resuelva el problema de valor inicial (6.20) y compare la solución exacta $y(t_{i+1})$ con el método de aproximación de Euler.

Suponiendo que y'' es continua, la solución exacta en $t_{i+1} = t_i + h$ es

$$y(t_i + h) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(c),$$

de acuerdo con el teorema de Taylor, para una cierta c (desconocida) que satisface $t_i < c < t_{i+1}$. Puesto que $y(t_i) = w_i$ y $y'(t_i) = f(t_i, w_i)$, esto puede escribirse como

$$y(t_{i+1}) = w_i + hf(t_i, w_i) + \frac{h^2}{2} y''(c).$$

Mientras tanto, el método de Euler dice que

$$w_{i+1} = w_i + hf(t_i, w_i).$$

Al restar las dos expresiones se obtiene el error local de truncamiento

$$e_{i+1} = |w_{i+1} - y(t_{i+1})| = \frac{h^2}{2} |y''(c)|$$

para una cierta c en el intervalo. Si M es una cota superior para y'' en $[a, b]$, entonces el error de truncamiento local satisface $e_i \leq Mh^2/2$. ◀

Ahora se investigará la forma en que se acumulan los errores locales para formar los errores globales. En la condición inicial $y(a) = y_a$, el error global es $g_0 = |w_0 - y_0| = |y_a - y_a| = 0$. Después de un paso, no hay error acumulado de los pasos anteriores, y el error global es igual al primer error local, $g_1 = e_1 = |w_1 - y_1|$. Después de dos pasos, descomponga g_2 en el error de truncamiento local más el error acumulado de la etapa anterior, como en la figura 6.5. Defina $z(t)$ como la solución del problema de valor inicial

$$\begin{cases} y' = f(t, y) \\ y(t_1) = w_1 \\ t \text{ en } [t_1, t_2] \end{cases}. \tag{6.21}$$

Por lo tanto, $z(t_2)$ es el valor exacto de la solución comenzando con la condición inicial (t_1, w_1). Observe que si se utiliza la condición inicial (t_1, y_1) se obtiene y_2, que está sobre la curva de solución real, a diferencia de $z(t_2)$. Entonces $e_2 = |w_2 - z(t_2)|$ es el error local de truncamiento del paso $i = 2$. La otra diferencia $|z(t_2) - y_2|$ está cubierta por el teorema 6.3, puesto que es la diferencia entre dos soluciones de la misma ecuación con diferentes condiciones iniciales w_1 y y_1. Por lo tanto,

$$\begin{aligned} g_2 = |w_2 - y_2| &= |w_2 - z(t_2) + z(t_2) - y_2| \\ &\leq |w_2 - z(t_2)| + |z(t_2) - y_2| \\ &\leq e_2 + e^{Lh} g_1 \\ &= e_2 + e^{Lh} e_1. \end{aligned}$$

El argumento es el mismo para el paso $i = 3$, de donde resulta

$$g_3 = |w_3 - y_3| \leq e_3 + e^{Lh} g_2 \leq e_3 + e^{Lh} e_2 + e^{2Lh} e_1. \tag{6.22}$$

Del mismo modo, el error de truncamiento global en la etapa i satisface

$$g_i = |w_i - y_i| \leq e_i + e^{Lh} e_{i-1} + e^{2Lh} e_{i-2} + \cdots + e^{(i-1)Lh} e_1. \tag{6.23}$$

En el ejemplo 6.7, se encontró que el método de Euler tiene un error de truncamiento local proporcional a h^2. De manera más general, se supone que el error de truncamiento local satisface

$$e_i \leq Ch^{k+1}$$

para algún entero k y una constante $C > 0$. Entonces

$$\begin{aligned} g_i &\leq Ch^{k+1}\left(1 + e^{Lh} + \cdots + e^{(i-1)Lh}\right) \\ &= Ch^{k+1} \frac{e^{iLh} - 1}{e^{Lh} - 1} \\ &\leq Ch^{k+1} \frac{e^{L(t_i - a)} - 1}{Lh} \\ &= \frac{Ch^k}{L}(e^{L(t_i - a)} - 1). \end{aligned} \tag{6.24}$$

Observe cómo el error de truncamiento local se relaciona con el error de truncamiento global. El error de truncamiento local es proporcional a h^{k+1} para alguna k. En términos generales, el error de truncamiento global "suma" los errores de truncamiento local durante un número de pasos proporcional a h^{-1}, el recíproco del tamaño de paso. Por lo tanto, el error global resulta ser

ANOTACIÓN **Convergencia** El teorema 6.4 es el teorema principal en la convergencia de los métodos para la solución de ecuaciones diferenciales de un solo paso. La dependencia del error global en h muestra que puede esperarse que el error disminuya a medida que h se reduce, de modo que (al menos en la aritmética exacta) el error puede hacerse tan pequeño como se desee. Esto conduce a otro punto importante: la dependencia exponencial del error global en b. En la medida que aumenta el número de pasos, la cota del error global puede crecer mucho. Para t_j grandes, el tamaño de paso h requerido para mantener pequeño al error global puede ser tan pequeño que no resulte práctico.

proporcional a h^k. Este hallazgo es el más importante del cálculo anterior, y se establecerá en el siguiente teorema:

TEOREMA 6.4 Suponga que $f(t, y)$ tiene una constante de Lipschitz L para la variable y y que el valor de y_i de la solución del problema de valor inicial (6.2) en t_i se aproxima mediante la w_i de un método para resolver una EDO de un solo paso con error local de truncamiento $e_i \leq Ch^{k+1}$, para alguna constante C y $k \geq 0$. Entonces, para cada $a < t_i < b$, el método tiene un error de truncamiento global

$$g_i = |w_i - y_i| \leq \frac{Ch^k}{L}(e^{L(t_i-a)} - 1). \tag{6.25}$$

■

Si un método para resolver una EDO satisface (6.25) cuando $h \to 0$, se dice que el método tiene **orden** k. El ejemplo 6.7 demuestra que el error de truncamiento local del método de Euler es de un tamaño limitado por $Mh^2/2$, por lo que el orden del método de Euler es 1. Al aplicar el teorema en el caso del método de Euler se da el siguiente corolario:

COROLARIO 6.5 (Convergencia del método de Euler) Suponga que $f(t, y)$ tiene una constante de Lipschitz L para la variable y y que la solución y_i del problema de valor inicial (6.2) en t_i se aproxima mediante w_i, usando el método de Euler. Sea M una cota superior para $|y''(t)|$ en $[a, b]$. Entonces

$$|w_i - y_i| \leq \frac{Mh}{2L}(e^{L(t_i-a)} - 1). \tag{6.26}$$

■

► EJEMPLO 6.8 Encuentre una cota de error para el método de Euler aplicado al ejemplo 6.1.

La constante de Lipschitz en $[0, 1]$ es $L = 1$. Ahora que se conoce la solución $y(t) = 3e^{t^2/2} - t^2 - 2$, se determina que la segunda derivada es $y''(t) = (t^2 + 2)e^{t^2/2} - 2$, cuyo valor absoluto está limitado por arriba en $[0, 1]$ mediante $M = y''(1) = 3\sqrt{e} - 2$. El corolario 6.5 implica que el error de truncamiento global en $t = 1$ debe ser menor que

$$\frac{Mh}{2L}e^L(1 - 0) = \frac{(3\sqrt{e} - 2)}{2}eh \approx 4.004h. \tag{6.27}$$

Este límite superior se confirma mediante los errores de truncamiento totales reales, que se muestran en la figura 6.4, los cuales son aproximadamente 2 veces h para h pequeñas. ◄

Hasta ahora, el método de Euler parece ser un buen método. Es intuitivo en su construcción y los errores que comete se hacen más pequeños cuando disminuye el tamaño del paso, de acuerdo con el corolario 6.5. Sin embargo, para PVI más difíciles, el método de Euler casi no se utiliza. Existen métodos más sofisticados cuyo orden, o potencia de h en (6.25), es mayor que uno. Esto conduce a un error total reducido, como se verá más adelante. Esta sección se cierra con un ejemplo de aspecto inocente en el que se requiere una reducción del error.

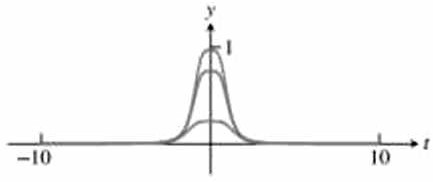


Figura 6.6 Aproximación del ejemplo 6.9 mediante el método de Euler. De abajo hacia arriba, soluciones aproximadas con tamaños de paso $h = 10^{-3}, 10^{-4}$ y 10^{-5}. La solución correcta tiene $y(0) = 1$. Se requieren pasos extremadamente pequeños para obtener una aproximación razonable.

► EJEMPLO 6.9 Aplique el método de Euler al problema de valor inicial

$$\begin{cases} y' = -4t^3y^2 \\ y(-10) = 1/10001 \\ t \text{ en } [-10, 0]. \end{cases} \tag{6.28}$$

Es fácil comprobar por sustitución que la solución exacta es $y(t) = 1/(t^4 + 1)$. La solución se comporta muy bien en el intervalo de interés. Se evaluará la capacidad del método de Euler para aproximar la solución en $t = 0$.

En la figura 6.6 se muestran las aproximaciones del método de Euler a la solución, con tamaños de paso $h = 10^{-3}, 10^{-4}$ y 10^{-5}, de abajo hacia arriba. El valor de la solución correcta en $t = 0$ es $y(0) = 1$. Incluso la mejor aproximación, que utiliza un millón de iteraciones para llegar a $t = 0$ a partir de la condición inicial, es notablemente correcta. ◄

Este ejemplo muestra que se requieren métodos más exactos para lograr la precisión en una cantidad razonable de cálculos. El resto del capítulo está dedicado al desarrollo de métodos más sofisticados que requieren menos pasos para obtener la misma o mejor precisión.

6.2.2 Método explícito del trapecio

Un pequeño ajuste en la fórmula del método de Euler hace una gran mejora en la precisión. Considere el método al interpretarlo geométricamente:

Método explícito del trapecio

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i + \frac{h}{2}(f(t_i, w_i) + f(t_i + h, w_i + hf(t_i, w_i))). \end{aligned} \tag{6.29}$$

Para el método de Euler, la pendiente $y'(t_i)$ que rige el paso discreto se toma del campo de pendientes en el extremo izquierdo del intervalo $[t_i, t_{i+1}]$. Para el método del trapecio como se ilustra en la figura 6.7, esta pendiente se sustituye por el promedio entre $y'(t_i)$ desde el extremo izquierdo y la pendiente $f(t_i + h, w_i + hf(t_i, w_i))$ desde el punto derecho que da el método de Euler. La "predicción" del método de Euler se utiliza como el valor w para evaluar la pendiente de la función f en $t_{i+1} = t_i + h$. En cierto sentido, la predicción del método de Euler se corrige mediante el método del trapecio, que es más preciso, como se demostrará más adelante.

El método del trapecio se llama explícito porque la nueva aproximación w_{i+1} puede determinarse mediante una fórmula explícita en términos de las w_i, t_i y h anteriores. El método de Euler también es un método explícito.

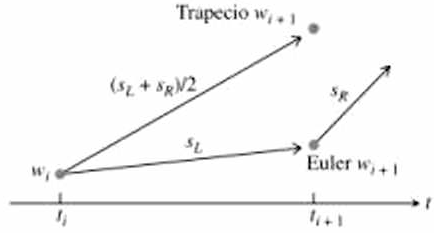


Figura 6.7 Vista esquemática de un solo paso del método del trapecio explícito. Las pendientes $s_L = f(t_i, w_i)$ y $S_R = f(t_i + h, w_i + hf(t_i, w_i))$ se promedian para definir la pendiente usada para avanzar a la solución hasta t_{i+1}.

La razón para el nombre de "método del trapecio" es que en el caso especial donde $f(t, y)$ es independiente de y, el método

$$w_{i+1} = w_i + \frac{h}{2}[f(t_i) + f(t_i + h)]$$

puede verse como la suma de una aproximación por la regla del trapecio de la integral $\int_{t_i}^{t_i+h} f(t)\,dt$ a la w_i actual. Como

$$\int_{t_i}^{t_i+h} f(t)\,dt = \int_{t_i}^{t_i+h} y'(t)\,dt = y(t_i + h) - y(t_i),$$

esto corresponde a resolver la ecuación diferencial $y' = f(t)$ integrando ambos lados mediante la regla del trapecio (5.21). El método del trapecio explícito también se le conoce como método de Euler mejorado y método de Heun, pero se utilizará el título más descriptivo y más fácil de recordar.

► EJEMPLO 6.10 Aplique el método del trapecio explícito al problema de valor inicial (6.5) con condición inicial $y(0) = 1$.

La fórmula (6.29) para $f(t, y) = ty + t^3$ es

$$\begin{aligned} w_0 &= y_0 = 1 \\ w_{i+1} &= w_i + \frac{h}{2}(f(t_i, w_i) + f(t_i + h, w_i + hf(t_i, w_i))) \\ &= w_i + \frac{h}{2}(t_i y_i + t_i^3 + (t_i + h)(w_i + h(t_i y_i + t_i^3)) + (t_i + h)^3). \end{aligned}$$

Si se usa el tamaño de paso $h = 0.1$, la iteración produce la tabla siguiente:

iteración	t_i	w_i	y_i	e_i
0	0.0	1.0000	1.0000	0.0000
1	0.1	1.0051	1.0050	0.0001
2	0.2	1.0207	1.0206	0.0001
3	0.3	1.0483	1.0481	0.0002
4	0.4	1.0902	1.0899	0.0003
5	0.5	1.1499	1.1494	0.0005
6	0.6	1.2323	1.2317	0.0006
7	0.7	1.3437	1.3429	0.0008
8	0.8	1.4924	1.4914	0.0010
9	0.9	1.6890	1.6879	0.0011
10	1.0	1.9471	1.9462	0.0010

◄

La comparación del ejemplo 6.10 con los resultados del método de Euler en el mismo problema del ejemplo 6.2 es sorprendente. Con el fin de cuantificar la mejora que el método del trapecio trae consigo para la solución de problemas de valor inicial, es necesario calcular su error local de truncamiento (6.19).

El error local de truncamiento es el error cometido en un solo paso. A partir de un punto supuesto de la solución correcta (t_i, y_i), la extensión correcta de la solución en t_{i+1} puede estar dado por la expansión de Taylor

$$y_{i+1} = y(t_i + h) = y_i + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \frac{h^3}{6}y'''(c), \tag{6.30}$$

para algún número c entre t_i y t_{i+1}, suponiendo que y''' es continua. Con el fin de comparar estos términos con el método del trapecio, se escribirán un poco diferente. A partir de la ecuación diferencial $y'(t) = f(t, y)$, diferencie ambos lados con respecto a t, utilizando la regla de la cadena:

$$\begin{aligned} y''(t) &= \frac{\partial f}{\partial t}(t, y) + \frac{\partial f}{\partial y}(t, y)y'(t) \\ &= \frac{\partial f}{\partial t}(t, y) + \frac{\partial f}{\partial y}(t, y)f(t, y). \end{aligned}$$

La nueva versión de (6.30) es

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t}(t_i, y_i) + \frac{\partial f}{\partial y}(t_i, y_i)f(t_i, y_i)\right) + \frac{h^3}{6}y'''(c). \tag{6.31}$$

Se desea comparar esta expresión con el método del trapecio explícito, usando el teorema de Taylor bidimensional para expandir el término

$$f(t_i + h, y_i + hf(t_i, y_i)) = f(t_i, y_i) + h\frac{\partial f}{\partial t}(t_i, y_i) + hf(t_i, y_i)\frac{\partial f}{\partial y}(t_i, y_i) + O(h^2).$$

El método del trapecio puede escribirse como

$$\begin{aligned} w_{i+1} &= y_i + \frac{h}{2}\left(f(t_i, y_i) + f(t_i + h, y_i + hf(t_i, y_i))\right) \\ &= y_i + \frac{h}{2}f(t_i, y_i) + \frac{h}{2}\left(f(t_i, y_i) + h\left(\frac{\partial f}{\partial t}(t_i, y_i)\right.\right. \\ &\quad \left.\left. + f(t_i, y_i)\frac{\partial f}{\partial y}(t_i, y_i)\right) + O(h^2)\right) \\ &= y_i + hf(t_i, y_i) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t}(t_i, y_i) + f(t_i, y_i)\frac{\partial f}{\partial y}(t_i, y_i)\right) + O(h^3). \end{aligned} \tag{6.32}$$

ANOTACIÓN

Resumen ¿Un método de segundo orden es más eficiente o menos eficiente que un método de primer orden? En cada paso el error es menor, pero el trabajo de cálculo es mayor, puesto que de manera ordinaria se requieren dos evaluaciones de la función ($f(t, y)$) en lugar de uno. Una comparación simple es la siguiente: suponga que se ha calculado una aproximación con tamaño de paso h, y se desea mejorar la aproximación. Para un mismo número de evaluaciones de la función, se puede (a) reducir a la mitad el tamaño de paso del método de primer orden, multiplicando el error global por 1/2, o (b) mantener el mismo tamaño de paso, pero usar un método de segundo orden, sustituyendo h en el teorema 6.4 por h^2, o sea multiplicar el error global por h. Para las h pequeñas, la opción (b) gana.

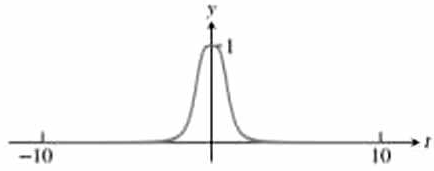


Figura 6.8 Aproximación del ejemplo 6.9 por el método del trapecio. El tamaño de paso es $h = 10^{-3}$. Observe la mejora significativa en la precisión comparada con el método de Euler de la figura 6.6.

Si se resta (6.32) de (6.31) se obtiene el error de truncamiento local

$$y_{i+1} - w_{i+1} = O(h^3).$$

El teorema 6.4 demuestra que el error global del método del trapecio es proporcional a h^2, lo que significa que el método es de segundo orden, en comparación con el primer orden del método de Euler. Para h pequeñas, ésta es una diferencia significativa, como se observa al revisar el ejemplo 6.9.

► **EJEMPLO 6.11** Aplique el método del trapecio al ejemplo 6.9:

$$\begin{cases} y' = -4t^3y^2 \\ y(-10) = 1/10001. \\ t \text{ en } [-10, 0] \end{cases}$$

Al abordar de nuevo el ejemplo 6.9 con un método más poderoso se obtiene una gran mejora en la aproximación a la solución; por ejemplo, en $x = 0$. Con el método del trapecio, el valor correcto $y(0) = 1$ se alcanza con precisión de .0015 y un tamaño de paso $h = 10^{-3}$, como se muestra en la figura 6.8. Esto ya es mejor que Euler con un tamaño de paso $h = 10^{-5}$. Si se usa el método del trapecio con $h = 10^{-5}$ se obtiene un error en el orden de 10^{-7} para este problema de valor inicial relativamente difícil. ◄

6.2.3 Métodos de Taylor

Hasta ahora, se han estudiado dos métodos para aproximar soluciones de ecuaciones diferenciales ordinarias. El método de Euler es de primer orden, y el método del trapecio aparentemente superior es de segundo orden. En esta sección se muestra que existen métodos de todos los órdenes. Para cada entero positivo k existe un método de Taylor de orden k, que se describirá a continuación.

La idea básica es una explotación directa de la expansión de Taylor. Suponga que la solución $y(t)$ es $(k + 1)$ veces continuamente diferenciable. Dado el punto actual $(t, y(t))$ en la curva de solución, la meta es expresar $y(t + h)$ en términos de $y(t)$ para algún tamaño de paso h, utilizando información acerca de la ecuación diferencial. La expansión de Taylor de $y(t)$ respecto a t es

$$y(t+h) = y(t) + hy'(t) + \frac{1}{2}h^2y''(t) + \cdots + \frac{1}{k!}h^k y^{(k)}(t) + \frac{1}{(k+1)!}h^{k+1}y^{(k+1)}(c), \tag{6.33}$$

donde c se encuentra entre t y $t + h$. El último término es el término residual de Taylor. Esta ecuación motiva el método siguiente:

Método de Taylor de orden *k*

$$w_0 = y_0$$

$$w_{i+1} = w_i + hf(t_i, w_i) + \frac{h^2}{2}f'(t_i, w_i) + \cdots + \frac{h^k}{k!}f^{(k-1)}(t_i, w_i). \tag{6.34}$$

La notación prima se refiere a la derivada total de $f(t, y(t))$ con respecto a t. Por ejemplo,

$$\begin{aligned} f'(t, y) &= f_t(t, y) + f_y(t, y)y'(t) \\ &= f_t(t, y) + f_y(t, y)f(t, y). \end{aligned}$$

Se usa la notación f_t para denotar la derivada parcial de f con respecto a t, algo similar ocurre con f_y. Para encontrar el error de truncamiento local del método de Taylor, establezca $w_i = y_i$ en (6.34) y compárela con la expansión de Taylor (6.33) para obtener

$$y_{i+1} - w_{i+1} = \frac{h^{k+1}}{(k+1)!}y^{(k+1)}(c).$$

Se llega a la conclusión de que el método de Taylor de orden k tiene un error de truncamiento local h^{k+1} y tiene un orden k, de acuerdo con el teorema 6.4.

El método de Taylor de primer orden es

$$w_{i+1} = w_i + hf(t_i, w_i),$$

que se identifica como método de Euler. El método de Taylor de segundo orden es

$$w_{i+1} = w_i + hf(t_i, w_i) + \frac{1}{2}h^2(f_t(t_i, w_i) + f_y(t_i, w_i)f(t_i, w_i)).$$

► EJEMPLO 6.12 Determine el método de Taylor de segundo orden para la ecuación lineal de primer orden

$$\begin{cases} y' = ty + t^3 \\ y(0) = y_0 \end{cases} \tag{6.35}$$

Como $f(t, y) = ty + t^3$, se deduce que

$$\begin{aligned} f'(t, y) &= f_t + f_y f \\ &= y + 3t^2 + t(ty + t^3), \end{aligned}$$

y el método queda

$$w_{i+1} = w_i + h(t_i w_i + t_i^3) + \frac{1}{2}h^2(w_i + 3t_i^2 + t_i(t_i w_i + t_i^3)).$$

◄

Aunque este método de Taylor es de segundo orden, observe que se requirió trabajo manual por parte del usuario para determinar las derivadas parciales. Compare esto con el otro método de segundo orden que se ha estudiado, donde (6.29) requiere sólo invocar una rutina que calcule los valores de $f(t, y)$.

Conceptualmente, la lección representada por los métodos de Taylor es que existen métodos en la solución de una EDO de orden arbitrario, como se muestra en (6.34). Sin embargo, tienen el problema del trabajo extra necesario para calcular las derivadas parciales de f que se muestran en la fórmula. Como hay fórmulas de los mismos órdenes que pueden desarrollarse de modo que no requieran estas derivadas parciales, los métodos de Taylor sólo se utilizan para fines especializados.

6.2 Ejercicios

1. Use la condición inicial $y(0) = 1$ y el tamaño de paso $h = 1/4$ para calcular la aproximación $w_0, \ldots, w_4$ mediante el método del trapecio en el intervalo $[0, 1]$. Encuentre el error en $t = 1$ comparando su respuesta con la solución correcta que se encontró en el ejercicio 6.1.3.

   (a) $y' = t$ (b) $y' = t^2 y$ (c) $y' = 2(t + 1)y$

(d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

2. Use la condición inicial $y(0) = 0$ y el tamaño de paso $h = 1/4$ para calcular la aproximación por el método del trapecio en el intervalo [0, 1]. Encuentre el error en $t = 1$ comparando su respuesta con la solución correcta que se encontró en el ejercicio 4 de la sección 6.1.

(a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

3. Encuentre la fórmula del método de Taylor de segundo orden para las siguientes ecuaciones diferenciales:
(a) $y' = ty$ (b) $y' = ty^2 + y^3$ (c) $y' = y \operatorname{sen} y$ (d) $y' = e^{yt^2}$

4. Aplique el método de Taylor de segundo orden a los problemas con los valores iniciales del ejercicio 1. Use el tamaño de paso $h = 1/4$ y calcule la aproximación por el método de Taylor de segundo orden en el intervalo [0, 1]. Compare su respuesta con la solución correcta hallada en el ejercicio 3 de la sección 6.1 y encuentre el error en $t = 1$.

5. (a) Pruebe (6.22) (b) Pruebe (6.23).

6.2 Problemas de computadora

1. Aplique el método del trapecio explícito en una malla con tamaño de paso $h = 0.1$ en [0, 1] para los problemas de valor inicial del ejercicio 1. Imprima una tabla con los valores de t, las aproximaciones y los errores de truncamiento global en cada paso.

2. Grafique las soluciones aproximadas para los PVI del ejercicio 1 en [0, 1] para tamaños de paso $h = 0.1, 0.05$ y 0.025, junto con la solución verdadera.

3. Para los PVI del ejercicio 1, grafique el error de truncamiento global del método del trapecio explícito en $t = 1$ como una función de $h = 0.1 \times 2^{-k}$ para $0 \le k \le 5$. Utilice una gráfica log-log como en la figura 6.4.

4. Para los PVI del ejercicio 1, grafique el error de truncamiento global del método de Taylor de segundo orden en $t = 1$ como una función de $h = 0.1 \times 2^{-k}$ para $0 \le k \le 5$.

5. Grafique la solución aproximada del método del trapecio en [0, 1] para la ecuación diferencial $y' = 1 + y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 1$, junto con la solución exacta (vea el ejercicio 7 de la sección 6.1). Utilice tamaños de paso $h = 0.1$ y 0.05.

6. Grafique la solución aproximada del método del trapecio en [0, 1] para la ecuación diferencial $y' = 1 - y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = -1/2$, junto con la solución exacta (vea el ejercicio 8 de la sección 6.1). Utilice tamaños de paso $h = 0.1$ y 0.05.

7. Calcule la solución aproximada del método del trapecio en [0, 4] para la ecuación diferencial $y' = \operatorname{sen} y$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 100$, con tamaños de paso $h = 0.1 \times 2^{-k}$ para $0 \le k \le 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$, junto con la solución exacta (vea el ejercicio 15 de la sección 6.1), y haga una gráfica log-log del error en $t = 4$ como una función de h.

8. Calcule la solución aproximada del método del trapecio para la ecuación diferencial $y' = \operatorname{senh} y$ y la condición inicial (a) $y_0 = 1/4$ en el intervalo [0, 2] (b) $y_0 = 2$ en el intervalo [0, 1/4], con tamaños de paso $h = 0.1 \times 2^{-k}$ para $0 \le k \le 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$ junto con la solución exacta (vea el ejercicio 16 de la sección 6.1), y haga una gráfica log-log del error en el extremo del intervalo de tiempo como una función de h.

6.3 SISTEMAS DE ECUACIONES DIFERENCIALES ORDINARIAS

La aproximación a los sistemas de ecuaciones diferenciales puede hacerse como una extensión sencilla de la metodología para una sola ecuación diferencial. El tratamiento de los sistemas de ecuaciones amplía en gran medida la capacidad para modelar los comportamientos dinámicos más interesantes.

La capacidad para resolver sistemas de ecuaciones diferenciales ordinarias está en el centro del arte y la ciencia de la simulación en computadora. En esta sección se presentan dos sistemas físicos cuya simulación ha motivado un gran desarrollo de solucionadores de EDO: el péndulo y la mecánica orbital. El estudio de estos ejemplos proporcionará al lector una experiencia práctica sobre las capacidades y limitaciones de los métodos.

El **orden** de una ecuación diferencial se refiere a la derivada de mayor orden que aparece en la ecuación. Un sistema de primer orden tiene la forma

$$\begin{aligned} y_1' &= f_1(t, y_1, \ldots, y_n) \\ y_2' &= f_2(t, y_1, \ldots, y_n) \\ &\vdots \\ y_n' &= f_n(t, y_1, \ldots, y_n). \end{aligned}$$

En un problema de valor inicial, cada variable tiene su propia condición inicial.

► EJEMPLO 6.13 Aplique el método de Euler al sistema de primer orden con dos ecuaciones:

$$\begin{aligned} y_1' &= y_2^2 - 2y_1 \\ y_2' &= y_1 - y_2 - t y_2^2 \\ y_1(0) &= 0 \\ y_2(0) &= 1. \end{aligned} \tag{6.36}$$

Verifique que la solución del sistema (6.36) es la función valorada del vector

$$\begin{aligned} y_1(t) &= te^{-2t} \\ y_2(t) &= e^{-t}. \end{aligned}$$

Por el momento, olvide que la solución ya se conoce y aplique el método de Euler. La fórmula escalar del método de Euler se aplica a cada uno de los componentes de la siguiente manera:

$$\begin{aligned} w_{i+1,1} &= w_{i,1} + h(w_{i,2}^2 - 2w_{i,1}) \\ w_{i+1,2} &= w_{i,2} + h(w_{i,1} - w_{i,2} - t_i w_{i,2}^2). \end{aligned}$$

En la figura 6.9 se muestran las aproximaciones del método de Euler de y_1 y y_2, junto con la solución correcta. El código de MATLAB que realiza esto es en esencia igual que el programa 6.1, con unos pocos ajustes para tratar a y como un vector:

```
% Programa 6.2 Versión vectorial del método de Euler
% Entrada: intervalo inter, vector inicial y0, número de pasos n
% Salida: pasos de tiempo t, solución y
% Uso de ejemplo: euler2 ([0 1], [0 1], 10);
function [t,y]=euler2(inter,y0,n)
t(1)=inter(1); y(1,:)=y0;
h=(inter(2)-inter(1))/n;
for i=1:n
  t(i+1)=t(i)+h;
```

```
  y(i+1,:)=eulerstep(t(i),y(i,:),h);
end
plot(t,y(:,1),t,y(:,2));

function y=eulerstep(t,y,h)
%un paso del método de Euler
%Entrada: tiempo actual t, vector actual y, tamaño de paso h
%Salida: el vector solución aproximado en el tiempo t+h
y=y+h*ydot(t,y);

function z=ydot(t,y)
%lado derecho de la ecuación diferencial
z(1)=y(2)^2-2*y(1);
z(2)=y(1)-y(2)-t*y(2)^2;
```

6.3.1 Ecuaciones de orden superior

Una sola ecuación diferencial de orden superior puede convertirse en un sistema. Sea

$$y^{(n)} = f(t, y, y', y'', \ldots, y^{(n-1)})$$

una ecuación diferencial ordinaria de orden n. Defina la nuevas variables

$$\begin{aligned} y_1 &= y \\ y_2 &= y' \\ y_3 &= y'' \\ &\vdots \\ y_n &= y^{(n-1)}, \end{aligned}$$

y observe que la ecuación diferencial original puede escribirse como

$$y_n' = f(t, y_1, y_2, \ldots, y_n).$$

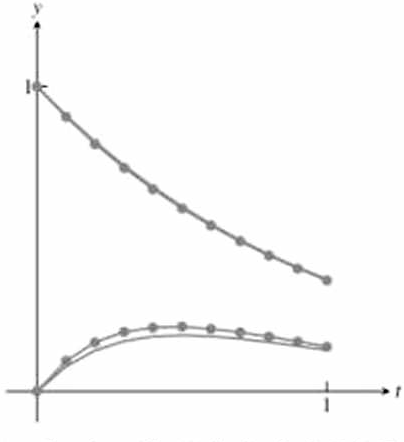


Figura 6.9 Ecuación (6.36) aproximada mediante el método de Euler. Tamaño de paso $h = 0.1$. La curva superior es $y_1(t)$, junto con su solución aproximada $w_{i,1}$ (círculos), mientras que la curva inferior es $y_2(t)$ y $w_{i,2}$.

En conjunto, las ecuaciones

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ y_3' &= y_4 \\ &\vdots \\ y_{n-1}' &= y_n, \\ y_n' &= f(t, y_1, \ldots, y_n) \end{aligned}$$

convierten la ecuación diferencial de orden n en un sistema de ecuaciones de primer orden, que puede resolverse usando métodos como el de Euler o el del trapecio.

► EJEMPLO 6.14 Convierta la ecuación diferencial de tercer orden

$$y''' = a(y'')^2 - y' + yy'' + \operatorname{sen} t \tag{6.37}$$

en un sistema.

Establezca $y_1 = y$, y defina las nuevas variables

$$\begin{aligned} y_2 &= y' \\ y_3 &= y''. \end{aligned}$$

Después, en términos de las primeras derivadas, (6.37) es equivalente a

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ y_3' &= ay_3^2 - y_2 + y_1 y_3 + \operatorname{sen} t. \end{aligned} \tag{6.38}$$

La solución $y(t)$ de la ecuación de tercer orden (6.37) puede encontrarse al resolver el sistema (6.38) para $y_1(t)$, $y_2(t)$, $y_3(t)$. ◄

Debido a la posibilidad de convertir ecuaciones de orden superior en sistemas de ecuaciones, se limitará la atención a los sistemas de ecuaciones de primer orden. Tenga en cuenta también que un sistema de varias ecuaciones de orden superior puede convertirse en un sistema de ecuaciones de primer orden en la misma forma.

6.3.2 Simulación en computadora: el péndulo

En la figura 6.10 se muestra un péndulo oscilante bajo la influencia de la gravedad. Suponga que el péndulo está colgado de una barra rígida que es libre de oscilar a través de 360 grados. Indique mediante y el ángulo del péndulo con respecto a la vertical, de modo que $y = 0$ corresponda a una línea recta hacia abajo. Por lo tanto, y y $y + 2\pi$ se consideran el mismo ángulo.

Para encontrar la ecuación del péndulo puede usarse la segunda ley del movimiento de Newton $F = ma$. El movimiento de la masa del péndulo está restringida a lo largo de un círculo de radio l, donde l es la longitud de la varilla del péndulo. Si y se mide en radianes, entonces la componente de la aceleración tangencial al círculo es ly'', porque la componente de la tangente al círculo es ly. La componente de fuerza a lo largo de la dirección del movimiento es mg sen y. Es una fuerza de restauración, lo que significa que se dirige en dirección opuesta al desplazamiento de la variable y. Por lo tanto, la ecuación diferencial que rige al péndulo sin fricción es

$$mly'' = F = -mg \operatorname{sen} y. \tag{6.39}$$

Ésta es una ecuación diferencial de segundo orden para el ángulo y del péndulo. Las condiciones iniciales están dadas por el ángulo inicial $y(0)$ y la velocidad angular $y'(0)$.

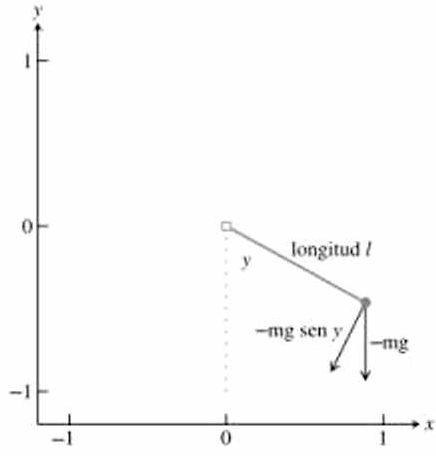


Figura 6.10 El péndulo. La componente de fuerza en la dirección tangencial es $F = -mg \operatorname{sen} y$, donde y es el ángulo del péndulo con la vertical.

Si se establece $y_1 = y$ y se introduce la nueva variable $y_2 = y'$, la ecuación de segundo orden se convierte en un sistema de primer orden:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -\frac{g}{l} \operatorname{sen} y_1. \end{aligned} \tag{6.40}$$

El sistema es autónomo porque no hay dependencia de t en el lado derecho. Si el péndulo se inicia desde una posición recta a la derecha, las condiciones iniciales son $y_1(0) = \pi/2$ y $y_2(0) = 0$. En unidades MKS, la aceleración de la gravedad en la superficie de la Tierra es de aproximadamente 9.81 m/s^2. Si se usan estos parámetros, puede probarse que el método de Euler proporciona una solución de este sistema.

En la figura 6.11 se muestran las aproximaciones del método de Euler a las ecuaciones del péndulo con dos diferentes tamaños de paso. A la varilla del péndulo se la asigna $l = 1$ metro de longitud. La curva más pequeña representa el ángulo y como una función del tiempo, y la curva de mayor amplitud es la velocidad angular instantánea. Observe que los ceros del ángulo, que representan la posición vertical del péndulo, corresponden a la mayor velocidad angular, positiva o negativa. El péndulo viaja más rápido cuando se balancea a través del punto más bajo. Cuando el péndulo se extiende al extremo derecho, el pico de la curva más pequeña, la velocidad es cero al pasar de positiva a negativa.

La incompetencia del método de Euler es evidente en la figura 6.11. El tamaño de paso $h = 0.01$ es claramente demasiado grande para lograr incluso una exactitud cualitativa. Un péndulo no amortiguado que inicia con velocidad cero debe oscilar adelante y atrás para siempre, volviendo a su posición de partida con una periodicidad regular. La amplitud del ángulo en la figura 6.11(a) está creciendo, lo que viola la ley de la conservación de la energía. Si se usan 10 veces más pasos, como en la figura 6.11(b), la situación mejora, al menos visualmente, pero se requieren un total de 10^4 pasos, un número extremo para el comportamiento dinámico rutinario mostrado por el péndulo.

Un método de segundo orden como el método del trapecio mejora la exactitud notablemente. Se reescribirá el código de MATLAB para que utilice el método del trapecio y así se tendrá la oportunidad de ilustrar la capacidad de MATLAB para hacer animaciones simples.

El código `pend.m` que sigue contiene la misma información de la ecuación diferencial, pero `eulerstep` se sustituye por `trapstep`. Además, se introducen las variables `rod` y `bob` para representar la varilla y la masa del péndulo, respectivamente. El comando `set` de MATLAB asigna atributos a las variables. El comando `drawnow` grafica las variables `rod` y `bob`. Tenga en cuenta que el modo de borrado de ambas variables se establece como `xor`, lo que significa que cuando la

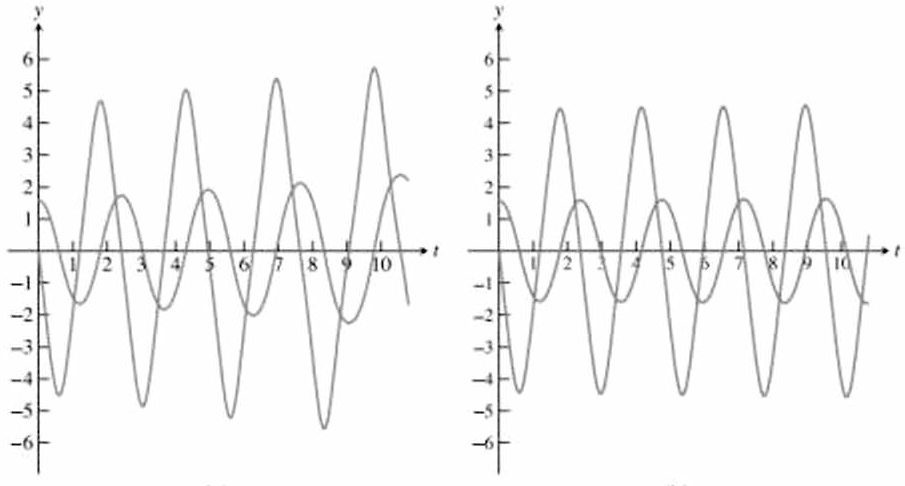


Figura 6.11 Método de Euler aplicado a la ecuación del péndulo (6.40). La curva de menor amplitud es el ángulo y_1 en radianes; la curva de mayor amplitud es la velocidad angular y_2. (a) El tamaño del paso $h = 0.01$ es demasiado grande, la energía está creciendo. (b) El tamaño de paso $h = 0.001$ muestra trayectorias más precisas.

variable graficada se vuelva a trazar en otro lugar, se borra la posición anterior. La figura 6.10 es una captura de pantalla de la animación. A continuación se presenta el código:

```
% Programa 6.3 Programa de animación para el péndulo
% Entradas: intervalo de tiempo inter,
%   valores iniciales ic= [y(1,1) y(1,2)], número de pasos n
% Llama un método de un solo paso, como trapstep.m
% Uso de ejemplo: pend([0 10], [pi/2 0], 200)
function pend(inter,ic,n)
h=(inter(2)-inter(1))/n;     % grafica n puntos en total
y(1,:)=ic;                   % introduce condiciones iniciales en y
t(1)=inter(1);
set(gca,'xlim',[-1.2 1.2],'ylim',[-1.2 1.2], ...
  'XTick',[-1 0 1],'YTick',[-1 0 1], ...
  'Drawmode','fast','Visible','on','NextPlot','add');
cla;
axis square                  % hace aspecto de razón 1 - 1
bob=line('color','r','Marker','.','markersize',40,...
    'erase','xor','xdata',[],'ydata',[]);
rod=line('color','b','LineStyle','-','LineWidth',3,...
    'erase','xor','xdata',[],'ydata',[]);
for k=1:n
  t(k+1)=t(k)+h;
  y(k+1,:)=trapstep(t(k),y(k,:),h);
  xbob=sin(y(k+1,1)); ybob= -cos(y(k+1,1));
  xrod=[0 xbob]; yrod=[0 ybob];
  set(rod,'xdata',xrod,'ydata',yrod)
  set(bob,'xdata',xbob,'ydata',ybob)
  drawnow; pause(h)
end

function y=trapstep(t,x,h)
%un paso del método del trapecio
```

```
z1=ydot(t,x);
g=x+h*z1;
z2=ydot(t+h,g);
y=x+h*(z1+z2)/2;

function z=ydot(t,y)
g=9.81;length=1;
z(1)=y(2);
z(2)=-(g/length)*sin(y(1));
```

El uso del método del trapecio en la ecuación del péndulo permite encontrar soluciones más exactas con tamaños de paso más grandes. Esta sección termina con algunas variaciones interesantes en la simulación básica del péndulo, con las cuales el lector podrá experimentar en los problemas de computadora.

► **EJEMPLO 6.15** El péndulo amortiguado.

La fuerza de amortiguación, tal como la resistencia del aire o la fricción, suelen modelarse como proporcionales y en la dirección opuesta a la velocidad. La ecuación del péndulo se convierte en

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -\frac{g}{l}\operatorname{sen} y_1 - d y_2, \end{aligned} \tag{6.41}$$

donde $d > 0$ es el coeficiente de amortiguación. A diferencia del péndulo no amortiguado, éste perderá energía a través del amortiguamiento y en el tiempo se aproximará a la solución de equilibrio limitante $y_1 = y_2 = 0$, a partir de cualquier condición inicial. El problema de computadora 3 le pide que corra una versión amortiguada de `pend.m`. ◄

► **EJEMPLO 6.16** El péndulo amortiguado forzado.

Al añadir un término en función del tiempo a (6.41) representa un esfuerzo externo del péndulo amortiguado. Considere añadir el término sinusoidal A sen t al lado derecho de y_2', de donde se obtiene

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -\frac{g}{l}\operatorname{sen} y_1 - d y_2 + A \operatorname{sen} t. \end{aligned} \tag{6.42}$$

Esto puede considerarse como un modelo de un péndulo que se ve afectado, por ejemplo, por un campo magnético oscilante.

Cuando se agrega este esfuerzo se presentan una serie de nuevos comportamientos dinámicos. Para un sistema autónomo bidimensional de ecuaciones diferenciales, el teorema de Poincaré-Bendixson (de la teoría de ecuaciones diferenciales) dice que las trayectorias sólo pueden tender hacia movimientos regulares, del tipo de los equilibrios estables como la posición vertical hacia abajo del péndulo, o a ciclos periódicos estables como el péndulo que oscila infinitamente. El esfuerzo hace que el sistema no sea autónomo (puede reescribirse como un sistema autónomo de tres dimensiones, pero no como uno bidimensional), de modo que se permite un tercer tipo de trayectorias; a saber, trayectorias caóticas.

Si se establece el coeficiente de amortiguamiento como $d = 1$ y el coeficiente de esfuerzo como $A = 10$, resulta un comportamiento periódico interesante, que se explora en el problema de computadora 4. Si el parámetro se establece como $A = 15$, resultan las trayectorias caóticas. ◄

► EJEMPLO 6.17 El péndulo doble.

El péndulo doble se compone de un péndulo simple, con otro péndulo simple que cuelga de su masa. Si y_1 y y_3 son los ángulos de las dos masas con respecto a la vertical, el sistema de ecuaciones diferenciales es

$$\begin{aligned}
y_1' &= y_2 \\
y_2' &= \frac{-3g\,\mathrm{sen}\,y_1 - g\,\mathrm{sen}(y_1 - 2y_3) - 2\,\mathrm{sen}(y_1 - y_3)(y_4^2 - y_2^2\cos(y_1 - y_3))}{3 - \cos(2y_1 - 2y_3)} - d\,y_2 \\
y_3' &= y_4 \\
y_4' &= \frac{2\,\mathrm{sen}(y_1 - y_3)[2y_2^2 + 2g\cos y_1 + y_4^2\cos(y_1 - y_3)]}{3 - \cos(2y_1 - 2y_3)},
\end{aligned}$$

donde $g = 9{,}81$ y la longitud de las dos barras se ha establecido en 1. El parámetro d representa la fricción en el pivote. Para $d = 0$, el doble péndulo exhibe una no periodicidad sostenida para muchas condiciones iniciales y es fascinante de observar. Vea el problema de computadora 8. ◄

6.3.3 Simulación en computadora: la mecánica orbital

Como un segundo ejemplo se simula el movimiento de un satélite en órbita. La segunda ley de Newton del movimiento dice que la aceleración a del satélite está relacionada con la fuerza F aplicada al satélite como $F = ma$, donde m es la masa. La ley de la gravitación expresa la fuerza sobre un cuerpo de masa m_1 debida a un cuerpo de masa m_2 mediante una ley inversa al cuadrado

$$F = \frac{gm_1m_2}{r^2},$$

donde r es la distancia que separa las masas. En el **problema de un cuerpo**, una de las masas se considera insignificante en comparación con la otra, como en el caso de un pequeño satélite que orbita a un planeta grande. Esta simplificación permite pasar por alto la fuerza del satélite sobre el planeta, por lo que el planeta puede considerarse como fijo.

Coloque la masa grande en el origen e indique la posición del satélite mediante (x, y). La distancia entre las masas es $r = \sqrt{x^2 + y^2}$, y la fuerza sobre el satélite es central (es decir, en dirección de la masa grande). El vector de dirección, un vector unitario en esta dirección, es

$$\left(-\frac{x}{\sqrt{x^2 + y^2}}, -\frac{y}{\sqrt{x^2 + y^2}}\right).$$

Por lo tanto, la fuerza sobre el satélite en términos de sus componentes es

$$(F_x, F_y) = \left(\frac{gm_1m_2}{x^2 + y^2}\frac{-x}{\sqrt{x^2 + y^2}}, \frac{gm_1m_2}{x^2 + y^2}\frac{-y}{\sqrt{x^2 + y^2}}\right). \tag{6.43}$$

Al sustituir estas fuerzas en la ley del movimiento de Newton se obtienen las dos ecuaciones de segundo orden

$$\begin{aligned}
m_1x'' &= -\frac{gm_1m_2x}{(x^2 + y^2)^{3/2}} \\
m_1y'' &= -\frac{gm_1m_2y}{(x^2 + y^2)^{3/2}}.
\end{aligned}$$

La introducción de las variables $v_x = x'$ y $v_y = y'$ permite que las dos ecuaciones de segundo orden se reduzcan a un sistema de cuatro ecuaciones de primer orden:

$$
\begin{aligned}
x' &= v_x \\
v_x' &= -\frac{gm_2 x}{(x^2+y^2)^{3/2}} \\
y' &= v_y \\
v_y' &= -\frac{gm_2 y}{(x^2+y^2)^{3/2}}
\end{aligned} \tag{6.44}
$$

El siguiente programa `orbit.m` de MATLAB llama a `eulerstep.m` y grafica de manera secuencial la órbita del satélite.

```
%Programa 6.4 Programa de graficado para el problema de un cuerpo
%Entradas: intervalo de tiempo inter, condiciones iniciales
% ic =[x0 vx0 y0 vy0], posición x, velocidad x, pos y, vel y,
% número de pasos n, pasos por punto graficado p
% Llama un método de un solo paso, como trapstep.m
% Uso de ejemplo: órbita ([0 100], [0 1 2 0],10000,5)
function z=orbit(inter,ic,n,p)
h=(inter(2)-inter(1))/n;                    % grafica n puntos
x0=ic(1);vx0=ic(2);y0=ic(3);vy0=ic(4);      % toma condiciones iniciales
y(1,:)=[x0 vx0 y0 vy0];t(1)=inter(1);       % construye el vector y
set(gca,'XLim',[-5 5],'YLim',[-5 5],'XTick',[-5 0 5],'YTick',...
  [-5 0 5],'Drawmode','fast','Visible','on');
cla;
sun=line('color','y','Marker','.','markersize',25,...
  'xdata',0,'ydata',0);
drawnow;
head=line('color','r','Marker','.','markersize',25,...
  'erase','xor','xdata',[],'ydata',[]);
tail=line('color','b','LineStyle','-','erase','none', ...
  'xdata',[],'ydata',[]);
%[px,py]=ginput(1);              %  incluya estas tres líneas
%[px1,py1]=ginput(1);            %  para permitir el uso del ratón
%y(1,:)=[px px1-px py py1-py];   %  2 clics establecen la dirección
for k=1:n/p
  for i=1:p
    t(i+1)=t(i)+h;
    y(i+1,:)=eulerstep(t(i),y(i,:),h);
  end
  y(1,:)=y(p+1,:);t(1)=t(p+1);
  set(head,'xdata',y(1,1),'ydata',y(1,3))
  set(tail,'xdata',y(2:p,1),'ydata',y(2:p,3))
  drawnow;
end

function y=eulerstep(t,x,h)
%un paso del método de Euler
y=x+h*ydot(t,x);

function z=ydot(t,x)
m2=3;g=1;mg2=m2*g;px2=0;py2=0;
px1=x(1);py1=x(3);vx1=x(2);vy1=x(4);
dist=sqrt((px2-px1)^2+(py2-py1)^2);
z=zeros(1,4);
z(1)=vx1;
z(2)=(mg2*(px2-px1))/(dist^3);
z(3)=vy1;
z(4)=(mg2*(py2-py1))/(dist^3);
```

Al correr el script `orbit.m` de MATLAB inmediatamente se muestran las limitaciones del método de Euler para este tipo de problemas. En la figura 6.12(a) se presenta el resultado de la ejecución de `orbit ([0 100],[0 1 2 0],10000,5)`. En otras palabras, se sigue la órbita durante el intervalo de tiempo $[a, b] = [0, 100]$, la posición inicial es $(x_0, y_0) = (0, 2)$, la velocidad inicial es $(v_x, v_y) = (1, 0)$ y el tamaño de paso de Euler es $h = 100/10000 = 0.01$.

Las soluciones al problema de un cuerpo deben ser secciones cónicas (ya sea elipses, parábolas o hipérbolas). La espiral que se ve en la figura 6.12(a) es un artificio numérico, es decir, una distorsión causada por los errores de cálculo. En este caso, el error de truncamiento del método de Euler conduce a que la órbita no se cierre en una elipse. Si el tamaño de paso se divide por un factor de 10 para obtener $h = 0.001$, resulta una mejora en el resultado, como se muestra en la figura 6.12(b). Es evidente que incluso con la gran disminución del tamaño de paso, el error acumulado es notable.

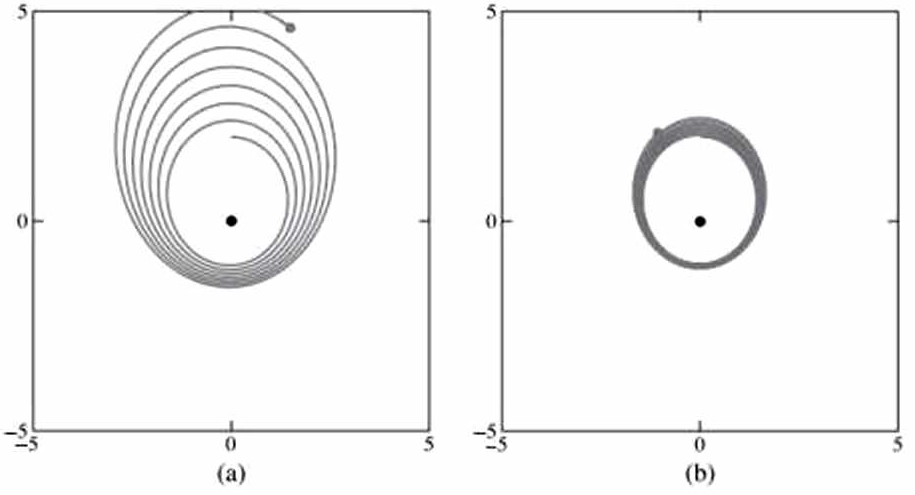


Figura 6.12 Método de Euler aplicado al problema de un cuerpo. (a) $h = 0.01$ y (b) $h = 0.001$.

El corolario 6.5 dice que el método de Euler, en principio, puede aproximarse a una solución con tanta precisión como se desee, si el tamaño de paso h es lo suficientemente pequeño. Sin embargo, resultados como los representados en las figuras 6.6 y 6.12 demuestran que, en la práctica, el método está muy limitado.

En la figura 6.13 se muestra una clara mejora en el problema de un cuerpo como consecuencia del remplazo del paso de Euler con el paso del trapecio. La gráfica se elaboró sustituyendo la función `eulerstep` por `trapstep` en el código anterior.

El problema de un cuerpo es ficticio, en el sentido de que no toma en cuenta la fuerza del satélite sobre el planeta (mucho más grande). Cuando dicha fuerza sí se incluye, el movimiento de los dos objetos se denomina problema de los dos cuerpos.

El caso de tres objetos que interactúan de manera gravitacional, llamado el **problema de los tres cuerpos**, tiene una posición importante en la historia de la ciencia. Aun cuando todo el movimiento se limite a un plano (problema de tres cuerpos **restringido**), las trayectorias a gran escala pueden ser demasiado impredecibles. En 1889, el rey Óscar II de Suecia y Noruega organizó un concurso para un trabajo en el que se demostrara la estabilidad del sistema solar. El premio fue otorgado a Henri Poincaré, que mostró que sería imposible probar tal cosa, debido a los fenómenos observados incluso para tres cuerpos en interacción.

La imprevisibilidad deriva de la **dependencia sensible a las condiciones iniciales**, un término que denota el hecho de que pequeñas incertidumbres en las posiciones y velocidades iniciales

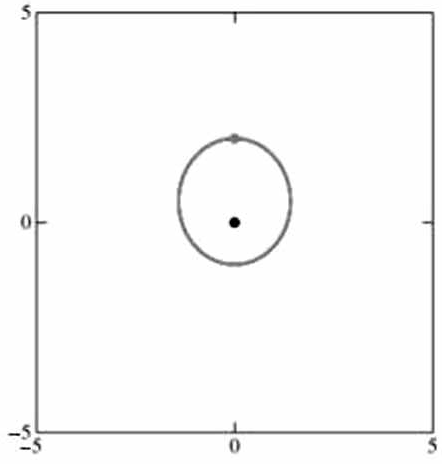


Figura 6.13 Problema de un cuerpo aproximado mediante el método del trapecio. Tamaño de paso $h = 0.01$. La órbita parece cerrarse, al menos con la resolución visible de la gráfica.

pueden conducir a grandes desviaciones en un momento posterior. En los términos utilizados aquí, esto es una afirmación de que la solución del sistema de ecuaciones diferenciales está mal condicionada con respecto a la entrada de las condiciones iniciales.

El problema de tres cuerpos restringido es un sistema de 12 ecuaciones, 4 para cada cuerpo, que también provienen de la segunda ley de Newton. Por ejemplo, las ecuaciones del primer cuerpo son

$$\begin{aligned} x_1' &= v_{1x} \\ v_{1x}' &= \frac{gm_2(x_2 - x_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}} + \frac{gm_3(x_3 - x_1)}{((x_3 - x_1)^2 + (y_3 - y_1)^2)^{3/2}} \\ y_1' &= v_{1y} \\ v_{1y}' &= \frac{gm_2(y_2 - y_1)}{((x_2 - x_1)^2 + (y_2 - y_1)^2)^{3/2}} + \frac{gm_3(y_3 - y_1)}{((x_3 - x_1)^2 + (y_3 - y_1)^2)^{3/2}}. \end{aligned} \tag{6.45}$$

Los cuerpos segundo y tercero, en (x_2, y_2) y (x_3, y_3), respectivamente, satisfacen ecuaciones similares.

Los programas de computadora 9 y 10 piden que el lector resuelva problemas de dos y tres cuerpos. El problema 10 ilustra sensiblemente la gran dependencia de las condiciones iniciales.

6.3 Ejercicios

1. Aplique el método de Euler con tamaño de paso $h = 1/4$ para el problema de valor inicial en $[0, 1]$.

(a) $\begin{cases} y_1' = y_1 + y_2 \\ y_2' = -y_1 + y_2 \\ y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$ (b) $\begin{cases} y_1' = -y_1 - y_2 \\ y_2' = y_1 - y_2 \\ y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$

(c) $\begin{cases} y_1' = -y_2 \\ y_2' = y_1 \\ y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$ (d) $\begin{cases} y_1' = y_1 + 3y_2 \\ y_2' = 2y_1 + 2y_2 \\ y_1(0) = 5 \\ y_2(0) = 0 \end{cases}$

Encuentre los errores de truncamiento totales de y_1 y y_2 en $t = 1$ comparando sus respuestas con las soluciones correctas

(a) $y_1(t) = e^t \cos t$, $y_2(t) = -e^t \operatorname{sen} t$ (b) $y_1(t) = e^{-t} \cos t$, $y_2(t) = e^{-t} \operatorname{sen} t$

(c) $y_1(t) = \cos t$, $y_2(t) = \operatorname{sen} t$ (d) $y_1(t) = 3e^{-t} + 2e^{4t}$, $y_2(t) = -2e^{-t} + 2e^{4t}$.

2. Aplique el método del trapecio con $h = 1/4$ a los problemas de valor inicial del ejercicio 1. Encuentre el error de truncamiento total en $t = 1$ comparando con las soluciones correctas.

3. Convierta la ecuación diferencial ordinaria de orden superior a un sistema de ecuaciones de primer orden.
(a) $y'' - ty = 0$ (ecuación de Airy) (b) $y'' - 2ty' + 2y = 0$ (ecuación de Hermite)
(c) $y'' - ty'' - y = 0$

4. Aplique el método del trapecio con $h = 1/4$ a los problemas de valor inicial del ejercicio 3, con $y(0) = y'(0) = 1$.

5. (a) Demuestre que $y(t) = (e^t + e^{-t} - t^2)/2 - 1$ es la solución del problema de valor inicial $y''' - y' = t$, con $y(0) = y'(0) = y''(0) = 0$. (b) Convierta la ecuación diferencial en un sistema de tres ecuaciones de primer orden. (c) Utilice el método de Euler con tamaño de paso $h = 1/4$ para aproximar la solución en [0, 1]. (d) Determine el error de truncamiento total en $t = 1$.

6.3 Problemas de computadora

1. Aplique el método de Euler con tamaño de paso $h = 0.1$ y $h = 0.01$ a los problemas de valor inicial del ejercicio 1. Grafique las soluciones aproximadas y la solución correcta en [0, 1], asimismo encuentre el error de truncamiento total en $t = 1$. ¿La reducción del error para $h = 0.01$ es consistente con el orden del método de Euler?

2. Resuelva el problema de computadora 1 con el método del trapecio.

3. Adapte `pend.m` para modelar el péndulo amortiguado. Ejecute el código resultante con $d = 0.1$. Excepto por la condición inicial $y_1(0) = \pi$, $y_2(0) = 0$, todas las trayectorias se mueven hacia la posición vertical hacia abajo conforme avanza el tiempo. Compruebe la condición inicial excepcional: ¿la simulación concuerda con la teoría?, ¿con un péndulo físico?

4. Adapte `pend.m` para construir una versión forzada y amortiguada del péndulo. Ejecute el método de trapecio para lo siguiente: (a) establezca la amortiguación $d = 1$ y el parámetro de esfuerzo $A = 10$. Fije el tamaño de paso $h = 0.005$ y la condición inicial de su elección. Después de moverse a través de un comportamiento transitorio, el péndulo se asentará en una trayectoria periódica (repetida). Describa esta trayectoria de manera cualitativa. Pruebe diferentes condiciones iniciales. ¿Todas las soluciones finalizan en la misma trayectoria periódica "atrayente"? (b) Ahora, aumente el tamaño de paso a $h = 0.01$, y repita el experimento. Pruebe la condición inicial $[\pi/2, 0]$ y otras. Describa lo que ocurre y dé una explicación razonable para el comportamiento anómalo con este tamaño de paso.

5. Ejecute el péndulo forzado amortiguado como en el problema de computadora 4, pero establezca $A = 12$. Utilice el método del trapecio con $h = 0.005$. Ahora hay dos atracciones periódicas que son imágenes entre sí. Describa las dos trayectorias de atracción y encuentre dos condiciones iniciales $(y_1, y_2) = (a, 0)$ y $(b, 0)$, donde $|a - b| \leq 0.1$, que son atraídas hacia diferentes trayectorias periódicas. Fije $A = 15$ para ver el movimiento caótico del péndulo forzado amortiguado.

6. Adapte `pend.m` para construir un péndulo amortiguado con pivote oscilante. El objetivo es investigar el fenómeno de la resonancia paramétrica, mediante el cual el péndulo invertido ¡se vuelve estable! La ecuación es

$$y'' + dy' + \left(\frac{g}{l} + A\cos 2\pi t\right)\operatorname{sen} y = 0,$$

donde A es el impulso de esfuerzo. Establezca $d = 0.1$ y la longitud del péndulo igual a 2.5 metros. En ausencia de esfuerzo ($A = 0$), el péndulo hacia abajo $y = 0$ es un equilibrio estable y el péndulo invertido $y = \pi$ es un equilibrio inestable. Encuentre, con la mayor precisión posible, el rango del

parámetro A para el cual el péndulo invertido se vuelve estable. (Por supuesto, $A = 0$ es demasiado pequeño y $A = 30$ es demasiado grande). Utilice la condición inicial $y = 3.1$ para su prueba y considere que la posición invertida es "estable" si el péndulo no pasa a través de la posición hacia abajo.

7. Use los parámetros establecidos en el problema de computadora 6 para demostrar el otro efecto de la resonancia paramétrica: el equilibrio estable puede llegar a ser inestable con un pivote oscilante. Encuentre el menor valor (positivo) del impulso del esfuerzo A para el cual ocurre esto. Clasifique la posición hacia abajo como inestable si el péndulo viaja en algún momento hasta la posición invertida.

8. Adapte `pend.m` para construir el péndulo doble. Debe definirse un nuevo par de `rod` y `bob` para el segundo péndulo. Tenga en cuenta que el extremo del pivote de la segunda barra es igual al antiguo extremo libre de la primera barra: la posición (x, y) del extremo libre de la segunda barra puede calcularse usando la trigonometría simple.

9. Adapte `orbit.m` para resolver el problema de dos cuerpos. Establezca las masas $m_1 = 0.3$, $m_2 = 0.03$, y grafique las trayectorias con condiciones iniciales $(x_1, y_1) = (2, 2)$, $(x_1', y_1') = (0.2, -0.2)$ y $(x_2, y_2) = (0, 0)$, $(x_2', y_2') = (-0.01, 0.01)$.

10. Adapte `orbit.m` para resolver el problema de tres cuerpos. Establezca las masas $m_1 = 0.3$, $m_2 = m_3 = 0.03$. (a) Grafique las trayectorias con condiciones iniciales $(x_1, y_1) = (2, 2)$, $(x_1', y_1') = (0.2, -0.2)$, $(x_2, y_2) = (0, 0)$, $(x_2', y_2') = (0, 0)$ y $(x_3, y_3) = (-2, -2)$, $(x_3', y_3') = (-0.2, 0.2)$. (b) Cambie la condición inicial de x_1' a 0.20001 y compare las trayectorias resultantes. Éste es un claro ejemplo visual de la dependencia sensible.

11. C. Moore descubrió en 1993 una notable órbita de tres cuerpos en forma de ocho. En esta configuración, tres cuerpos de igual masa se persiguen entre sí a lo largo de una figura en forma de ocho con un solo ciclo. Establezca las masas $m_1 = m_2 = m_3 = 1$ y la gravedad $g = 1$. (a) Adapte `orbit.m` para graficar la trayectoria con las condiciones iniciales $(x_1, y_1) = (-0.970, 0.243)$, $(x_1', y_1') = (-0.466, -0.433)$, $(x_2, y_2) = (-x_1, -y_1)$, $(x_2', y_2') = (x_1', y_1')$ y $(x_3, y_3) = (0, 0)$, $(x_3', y_3') = (-2x_1', -2y_1')$. (b) ¿Las trayectorias son sensibles a pequeños cambios en las condiciones iniciales? Investigue el efecto de cambiar x_3' en 10^{-k} para $1 \le k \le 5$. Para cada k, decida si el patrón en forma de ocho persiste, o si con el tiempo se presenta un cambio catastrófico.

6.4 MÉTODOS Y APLICACIONES DE RUNGE-KUTTA

Los métodos de Runge-Kutta forman una familia de métodos en la solución de EDO que incluye a los métodos de Euler y del trapecio, así como a los métodos más sofisticados de orden superior. En esta sección, se presenta una variedad de métodos de un solo paso, los cuales se aplican a la simulación de trayectorias de algunas aplicaciones clave.

6.4.1 La familia Runge-Kutta

Se ha visto que el método de Euler es de primer orden y que el método del trapecio es de segundo orden. Además del método del trapecio, existen otros métodos de segundo orden del tipo Runge-Kutta. Un ejemplo importante es el método del punto medio.

Método del punto medio

$$w_0 = y_0$$
$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2} f(t_i, w_i)\right). \tag{6.46}$$

Para verificar el orden del método del punto medio, es necesario calcular el error de truncamiento local. Cuando se hizo esto para el método del trapecio, se encontró la útil expresión (6.31):

$$y_{i+1} = y_i + hf(t_i, y_i) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t}(t_i, y_i) + \frac{\partial f}{\partial y}(t_i, y_i) f(t_i, y_i)\right) + \frac{h^3}{6} y'''(c). \qquad (6.47)$$

Para calcular el error de truncamiento local en el paso i, suponga que $w_i = y_i$ y calcule $y_{i+1} - w_{i+1}$. Si se repite el uso de la expansión de la serie de Taylor como en el método del trapecio, puede escribirse

$$\begin{aligned} w_{i+1} &= y_i + hf\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} f(t_i, y_i)\right) \\ &= y_i + h\left(f(t_i, y_i) + \frac{h}{2}\frac{\partial f}{\partial t}(t_i, y_i) + \frac{h}{2} f(t_i, y_i)\frac{\partial f}{\partial y}(t_i, y_i) + O(h^2)\right). \end{aligned} \qquad (6.48)$$

Al comparar (6.47) y (6.48) se obtiene

$$y_{i+1} - w_{i+1} = O(h^3),$$

por lo que el método del punto medio es de segundo orden, según el teorema 6.4.

Cada evaluación de función del lado derecho de la ecuación diferencial se denomina una **etapa** del método. Los métodos del trapecio y del punto medio son miembros de la familia de dos etapas, de los métodos de Runge-Kutta de segundo orden, que tienen la forma

$$w_{i+1} = w_i + h\left(1 - \frac{1}{2\alpha}\right) f(t_i, w_i) + \frac{h}{2\alpha} f(t_i + \alpha h, w_i + \alpha h f(t_i, w_i)) \qquad (6.49)$$

para alguna $\alpha \neq 0$. Si se establece $\alpha = 1$ corresponde al método del trapecio explícito y $\alpha = 1/2$ al método del punto medio. En el ejercicio 5 se le pide que compruebe el orden de los métodos de esta familia.

La figura 6.14 ilustra la intuición que está detrás de los métodos del trapecio y del punto medio. El método del trapecio utiliza un paso de Euler para el extremo derecho del intervalo, se evalúa la pendiente en ese punto y después se promedia con la pendiente del extremo izquierdo. El método del punto medio utiliza un paso de Euler para trasladarse al punto medio del intervalo, se evalúa la pendiente como $f(t_i + h/2, w_i + (h/2) f(t_i, w_i))$ y utiliza esa pendiente para pasar de w_i a la nueva aproximación w_{i+1}. Estos métodos utilizan diferentes enfoques para resolver el mismo problema: la obtención de una pendiente que represente a todo el intervalo de mejor manera que el método de Euler, el cual utiliza sólo la estimación de la pendiente desde el extremo izquierdo del intervalo.

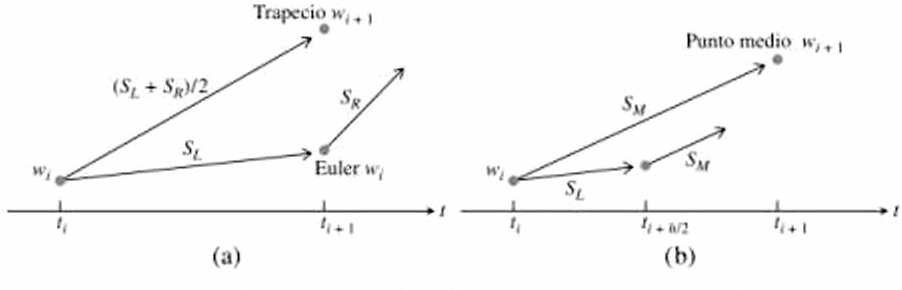


Figura 6.14 Vista esquemática de dos miembros de la familia RK2. (a) El método del trapecio usa un promedio de los extremos izquierdo y derecho para dividir el intervalo. (b) El método del punto medio usa una pendiente desde el punto medio del intervalo.

ANOTACIÓN

Convergencia Las propiedades de convergencia de un método de cuarto orden, como RK4, son muy superiores a las de los métodos de primer y segundo orden que se han analizado hasta ahora. La convergencia aquí significa lo rápido que el error (total) de la aproximación de la EDO en un tiempo t fijo tiende a cero cuando el tamaño de paso h tiende a cero. Cuarto orden significa que por cada reducción a la mitad del tamaño de paso, el error se reduce en aproximadamente un factor de $2^4 = 16$, como resulta claro en la figura 6.15.

Hay métodos de Runge-Kutta de todos los órdenes. Un ejemplo en particular es el método de cuarto orden.

Método de Runge-Kutta de cuarto orden (RK4)

$$w_{i+1} = w_i + \frac{h}{6}(s_1 + 2s_2 + 2s_3 + s_4) \tag{6.50}$$

donde

$$\begin{aligned} s_1 &= f(t_i, w_i) \\ s_2 &= f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}s_1\right) \\ s_3 &= f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}s_2\right) \\ s_4 &= f(t_i + h, w_i + hs_3). \end{aligned}$$

La popularidad de este método radica en su simplicidad y facilidad de programación. Se trata de un método de un solo paso, por lo que sólo requiere una condición inicial para empezar; sin embargo, como un método de cuarto orden, es mucho más exacto que los métodos de Euler o del trapecio.

La cantidad $h(s_1 + 2s_2 + 2s_3 + s_4)/6$ en el método de Runge-Kutta de cuarto orden toma el lugar de la pendiente en el método de Euler. Esta cantidad puede considerarse como una estimación mejorada de la pendiente de la solución en el intervalo $[t_i, t_i + h]$. Tenga en cuenta que s_1 es la pendiente en el extremo izquierdo del intervalo, s_2 es la pendiente utilizada en el método del punto medio, s_3 es una pendiente mejorada en el punto medio, y s_4 es una pendiente aproximada en el punto extremo derecho $t_i + h$. El álgebra que se requiere para probar que este método es de cuarto orden es similar a la obtención de los métodos del trapecio y del punto medio, pero es un poco larga y puede encontrarse, por ejemplo, en Henrici [1962]. Ahora se abordará de nuevo la ecuación diferencial (6.5) para fines de comparación.

► **EJEMPLO 6.18** Aplique Runge-Kutta de cuarto orden para el problema de valor inicial

$$\begin{cases} y' = ty + t^3 \\ y(0) = 1 \end{cases}. \tag{6.51}$$

Al calcular el error de truncamiento total en $t = 1$ para una variedad de tamaños de paso se obtiene la siguiente tabla:

iteración n	tamaño de paso h	error en $t = 1$
5	0.20000	2.3788×10^{-5}
10	0.10000	1.4655×10^{-6}
20	0.05000	9.0354×10^{-8}
40	0.02500	5.5983×10^{-9}
80	0.01250	3.4820×10^{-10}
160	0.00625	2.1710×10^{-11}
320	0.00312	1.3491×10^{-12}
640	0.00156	7.2609×10^{-14}

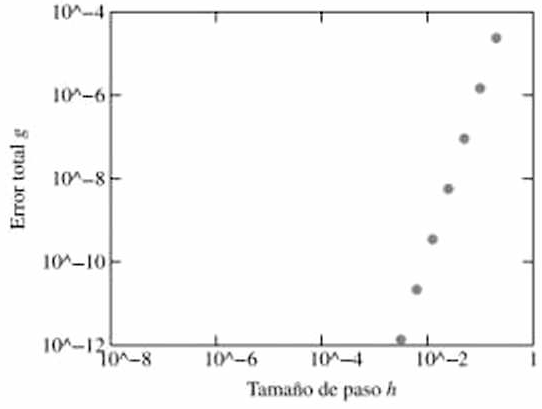


Figura 6.15 Error como una función del tamaño de paso para Runge-Kutta de cuarto orden. La diferencia entre la solución aproximada de (6.5) y la solución correcta en $t = 1$ tiene pendiente 4 en una gráfica log-log, por lo tanto es proporcional a h^4, para h pequeñas.

Compare con la tabla correspondiente para el método de Euler en la página 286. La diferencia es notable y fácilmente compensa la complejidad adicional de RK4, que requiere de cuatro evaluaciones de función por paso, en comparación con una sola de Euler. En la figura 6.15 se muestra la misma información en una forma que evidencia el hecho de que el error de truncamiento total es proporcional a h^4, como se esperaba para un método de cuarto orden. ◀

6.4.2 Simulación en computadora: la neurona de Hodgkin-Huxley

Las computadoras estaban en sus primeras etapas de desarrollo a mediados del siglo XX. Algunas de las primeras aplicaciones se hicieron para ayudar a resolver los hasta ese momento intratables sistemas de ecuaciones diferenciales.

A. L. Hodgkin y A. F. Huxley dieron nacimiento al campo de la neurociencia computacional mediante el desarrollo de un nuevo modelo realista para las células nerviosas, o neuronas. Fueron capaces de aproximar soluciones al modelo de ecuaciones diferenciales, incluso con los equipos rudimentarios que existían en ese momento. Por este trabajo ganaron el Premio Nobel de Biología en 1963.

El modelo es un sistema de cuatro ecuaciones diferenciales acopladas, una de las cuales modela la diferencia de voltaje entre el interior y el exterior de la célula. Las otras tres ecuaciones modelan los niveles de activación de los canales iónicos, que hacen el trabajo de intercambiar iones de sodio y potasio entre el interior y el exterior. Las **ecuaciones de Hodgkin-Huxley** son

$$\begin{aligned} Cv' &= -g_1 m^3 h(v - E_1) - g_2 n^4 (v - E_2) - g_3 (v - E_3) + I_{\text{in}} \\ m' &= (1 - m)\alpha_m(v - E_0) - m\beta_m(v - E_0) \\ n' &= (1 - n)\alpha_n(v - E_0) - n\beta_n(v - E_0) \\ h' &= (1 - h)\alpha_h(v - E_0) - h\beta_h(v - E_0), \end{aligned} \tag{6.52}$$

donde

$$\alpha_m(v) = \frac{2.5 - 0.1v}{e^{2.5-0.1v} - 1},\ \beta_m(v) = 4e^{-v/18},$$

$$\alpha_n(v) = \frac{0.1 - 0.01v}{e^{1-0.1v} - 1},\ \beta_n(v) = \frac{1}{8}e^{-v/80},$$

y

$$\alpha_h(v) = 0.07e^{-v/20},\ \beta_h(v) = \frac{1}{e^{3-0.1v} + 1}.$$

El coeficiente C representa la capacitancia de la célula e I_{in} indica la entrada de corriente de otras células. Los valores típicos del coeficiente son la capacitancia $C = 1$ microfaradios, las conductancias $g_1 = 120$, $g_2 = 36$, $g_3 = 0.3$ siemens y los voltajes $E_0 = -65$, $E_1 = 50$, $E_2 = -77$, $E_3 = -54.4$ milivoltios.

La ecuación v' es una ecuación de la corriente por unidad de área, en unidades de miliamperios/cm^2, mientras que las otras tres activaciones m, n y h no tienen unidades. El coeficiente C es la capacitancia de la membrana neuronal, g_1, g_2 y g_3 son las conductancias y E_1, E_2 y E_3 son los "potenciales inversos", que son los niveles de voltaje que forman el límite entre las corrientes que fluyen hacia el interior y el exterior.

Hodgkin y Huxley eligieron cuidadosamente la forma de las ecuaciones para que coincidieran con los datos experimentales, que se obtuvieron del axón gigante del calamar. También ajustaron los parámetros al modelo. Aunque los detalles del axón del calamar difieren de las neuronas de los mamíferos, el modelo se ha mantenido como una representación realista de la dinámica neural. De manera más general, resulta útil como un ejemplo de los medios excitables que traducen las entradas continuas en una respuesta del tipo todo o nada. El código de MATLAB que implementa el modelo es el siguiente:

```
% Programa 6.5 ecuaciones de Hodgkin-Huxley
% Entradas: Intervalo de tiempo inter,
% ic = voltaje inicial v y 3 variables de compuerta, pasos n
% Salida: solución y
% Llama un método de un paso como rk4step.m
% Uso de ejemplo: hh([0,100],[-65,0,0.3,0.6],2000);
function y=hh(inter,ic,n)
global pa pb pulse
inp=input('pulse start, end, muamps in [ ], e.g. [50 51 7]: ');
pa=inp(1);pb=inp(2);pulse=inp(3);
a=inter(1); b=inter(2); h=(b-a)/n; % grafica n puntos en total
y(1,:)=ic;                        % ingresa cond iniciales en y
t(1)=a;
for i=1:n
  t(i+1)=t(i)+h;
  y(i+1,:)=rk4step(t(i),y(i,:),h);
end
subplot(3,1,1);
plot([a pa pa pb pb b],[0 0 pulse pulse 0 0]);
grid;axis([0 100 0 2*pulse])
ylabel('input pulse')
subplot(3,1,2);
plot(t,y(:,1));grid;axis([0 100 -100 100])
ylabel('voltage (mV)')
subplot(3,1,3);
plot(t,y(:,2),t,y(:,3),t,y(:,4));grid;axis([0 100 0 1])
ylabel('gating variables')
legend('m','n','h')
xlabel('time (msec)')

function y=rk4step(t,w,h)
%un paso del método de Runge-Kutta de cuarto orden
s1=ydot(t,w);
s2=ydot(t+h/2,w+h*s1/2);
s3=ydot(t+h/2,w+h*s2/2);
s4=ydot(t+h,w+h*s3);
y=w+h*(s1+2*s2+2*s3+s4)/6;

function z=ydot(t,w)
global pa pb pulse
```

```
c=1;g1=120;g2=36;g3=0.3;T=(pa+pb)/2;len=pb-pa;
e0=-65;e1=50;e2=-77;e3=-54.4;
in=pulse*(1-sign(abs(t-T)-len/2))/2;
%entrada de pulso cuadrado en el intervalo [pa,pb] del pulso muamps
v=w(1);m=w(2);n=w(3);h=w(4);
z=zeros(1,4);
z(1)=(in-g1*m*m*m*h*(v-e1)-g2*n*n*n*n*(v-e2)-g3*(v-e3))/c;
v=v-e0;
z(2)=(1-m)*(2.5-0.1*v)/(exp(2.5-0.1*v)-1)-m*4*exp(-v/18);
z(3)=(1-n)*(0.1-0.01*v)/(exp(1-0.1*v)-1)-n*0.125*exp(-v/80);
z(4)=(1-h)*0.07*exp(-v/20)-h/(exp(3-0.1*v)+1);
```

Sin la entrada, la neurona de Hodgkin-Huxley permanece en reposo, con un voltaje aproximado de E_0. Si se establece I_{in} como un pulso cuadrado de corriente con longitud de 1 ms y fuerza de 7 microamperios que es suficiente para causar un pico, una desviación grande de la tensión de despolarización del voltaje. Esto se ilustra en la figura 6.16. Ejecute el programa para comprobar que 6.9 μA no es suficiente para causar un pico completo. De ahí, la respuesta de todo o nada. Esta propiedad de magnificar el efecto de las pequeñas diferencias en la entrada es la que puede explicar el éxito de la neurona en el procesamiento de información. En la figura 6.16(b) se muestra que si la corriente de entrada se mantiene, la neurona dispara una descarga periódica de picos. El problema de computadora 10 es el umbral de la investigación virtual del estudio de las neuronas.

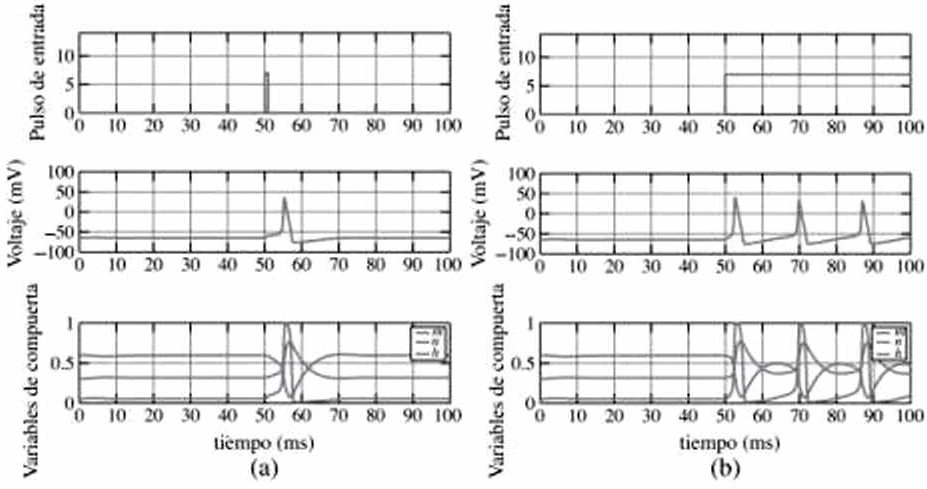


Figura 6.16 Capturas de pantalla del programa de Hodgin-Huxley. (a) Una entrada de onda cuadrada de tamaño $I_{in} = 7$ μA en un tiempo de 50 ms, con 1 ms de duración, hace que el modelo de neurona dispare una vez. (b) Una onda cuadrada sostenida, con $I_{in} = 7$ μA, hace que el modelo de neurona dispare de forma periódica.

6.4.3 Simulación en computadora: las ecuaciones de Lorenz

A finales de la década de 1950, el meteorólogo E. Lorenz del MIT adquirió una de las primeras computadoras comerciales. Era del tamaño de un refrigerador y funcionaba a una velocidad de 60 multiplicaciones por segundo. Este caché de poder de cómputo personal sin precedentes le permitió desarrollar y evaluar de manera significativa modelos meteorológicos que consistían en varias ecuaciones diferenciales que, al igual que las ecuaciones de Hodgkin-Huxley, no podían resolverse analíticamente.

Las **ecuaciones de Lorenz** son una simplificación de un modelo atmosférico en miniatura que se diseñó para estudiar la convección de Rayleigh-Bénard, el movimiento del calor en un fluido,

como el aire, a partir de un medio de calentamiento inferior (como el suelo) hacia un medio fresco más alto (como la atmósfera superior). En este modelo de la atmósfera en dos dimensiones, el desarrollo de una circulación de aire puede describirse mediante el siguiente sistema de tres ecuaciones:

$$\begin{aligned} x' &= -sx + sy \\ y' &= -xz + rx - y \\ z' &= xy - bz. \end{aligned} \tag{6.53}$$

La variable x denota la velocidad de circulación en el sentido horario y mide la diferencia de temperatura entre las columnas de aire ascendente y descendente; por otro lado, z mide la desviación de un perfil de temperatura estrictamente lineal en la dirección vertical. El número de Prandtl s, el número de Rayleigh r y b son los parámetros del sistema. La configuración más común para los parámetros es $s = 10$, $r = 28$ y $b = 8/3$. Estos valores se utilizaron para la trayectoria mostrada en la figura 6.17, que se calculó mediante Runge-Kutta de cuarto orden, utilizando el código siguiente para describir la ecuación diferencial.

```
function z=ydot(t,y)
%Ecuaciones de lorenz
s=10; r=28; b=8/3;
z(1)=-s*y(1)+s*y(2);
z(2)=-y(1)*y(3)+r*y(1)-y(2)
z(3)=y(1)*y(2)-b*y(3)
```

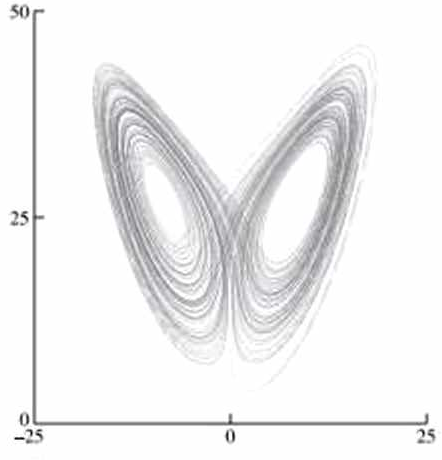


Figura 6.17 Trayectoria uno de las ecuaciones de Lorenz (6.53), proyectada al plano xz. Los parámetros se fijan en $s = 10$, $r = 28$ y $b = 8/3$.

Las ecuaciones de Lorenz son un ejemplo importante del porque las trayectorias muestran una gran complejidad, a pesar del hecho de que las ecuaciones son deterministas y bastante simples (casi lineales). La explicación de la complejidad es similar a la del péndulo doble o el problema de tres cuerpos: dependencia sensible sobre las condiciones iniciales. Los problemas de computadora 12 y 13 exploran la dependencia sensible de la llamada atracción caótica.

6.4 Ejercicios

1. Aplique el método del punto medio para los PVI

   (a) $y' = t$ (b) $y' = t^2 y$ (c) $y' = 2(t+1)y$

   (d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

con la condición inicial $y(0) = 1$. Use el tamaño de paso $h = 1/4$ y calcule la aproximación del método del punto medio en el intervalo [0, 1]. Compare su respuesta con la solución correcta hallada en el ejercicio 3 de la sección 6.1 y encuentre el error de truncamiento global en $t = 1$.

2. Realice los pasos del ejercicio 1 para los PVI

$$\text{(a)}\quad y' = t + y \quad \text{(b)}\quad y' = t - y \quad \text{(c)}\quad y' = 4t - 2y$$

con la condición inicial $y(0) = 0$. Las soluciones exactas se encontraron en el ejercicio 4 de la sección 6.1.

3. Aplique Runge-Kutta de cuarto orden para los PVI del ejercicio 1. Con el tamaño de paso $h = 1/4$, calcule la aproximación en el intervalo [0, 1]. Compare su respuesta con la solución correcta hallada en el ejercicio 3 de la sección 6.1 y encuentre el error de truncamiento total en $t = 1$.

4. Realice los pasos del ejercicio 3 para los PVI del ejercicio 2.

5. Demuestre que para cualquier $\alpha \neq 0$, el método de (6.49) es de segundo orden.

6. Considere el problema de valor inicial $y' = \lambda y$. La solución es $y(t) = y_0 e^{\lambda t}$. (a) Calcule w_1 con RK4 en términos de w_0 para esta ecuación diferencial. (b) Calcule el error de truncamiento local estableciendo $w_0 = y_0 = 1$ y determinando $y_1 - w_1$. Demuestre que el error de truncamiento local es de tamaño $O(h^5)$, como se espera de un método de cuarto orden.

7. Suponga que el lado derecho de $f(t, y) = f(t)$ no depende de y. Demuestre que $s_2 = s_3$ en Runge-Kutta de cuarto orden y que RK4 es equivalente a la regla de Simpson para la integral $\int_{t_i}^{t_i+h} f(s)\, ds$.

6.4 Problemas de computadora

1. Aplique el método del punto medio en una malla con tamaño de paso $h = 0.1$ en [0, 1] para los problemas de valor inicial del ejercicio 1. Imprima una tabla con los valores de t, las aproximaciones y los errores de truncamiento total en cada paso.

2. Aplique el método de Runge-Kutta de cuarto orden usando una malla con tamaño de paso $h = 0.1$ en [0, 1] para los problemas de valor inicial del ejercicio 1. Imprima una tabla con los valores de t, las aproximaciones y los errores de truncamiento global en cada paso.

3. Realice los pasos del problema de computadora 2, pero grafique las soluciones aproximadas en [0, 1] para tamaños de paso $h = 0.1$, 0.05 y 0.025, junto con la solución verdadera.

4. Realice los pasos del problema de computadora 2 para las ecuaciones del ejercicio 2.

5. Grafique la solución aproximada del método de Runge-Kutta de cuarto orden en [0, 1] para la ecuación diferencial $y' = 1 + y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 1$, junto con la solución exacta (vea el ejercicio 7 de la sección 6.1). Utilice tamaños de paso $h = 0.1$ y 0.05.

6. Grafique la solución aproximada del método de Runge-Kutta de cuarto orden en [0, 1] para la ecuación diferencial $y' = 1 - y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = -1/2$, junto con la solución exacta (vea el ejercicio 8 de la sección 6.1). Utilice tamaños de paso $h = 0.1$ y 0.05.

7. Calcule la solución aproximada del método Runge-Kutta de cuarto orden en [0, 4] para la ecuación diferencial $y' = \text{sen } y$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 100$, con tamaños de paso $h = 0.1 \times 2^{-k}$ para $k \leq 0 \leq 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$ junto con la solución exacta (vea el ejercicio 15 de la sección 6.1). También haga una gráfica log-log del error como una función de h.

8. Calcule la solución aproximada del método de Runge-Kutta de cuarto orden para la ecuación diferencial $y' = \text{senh}\, y$ con la condición inicial (a) $y_0 = 1/4$ en el intervalo $[0, 2]$ (b) $y_0 = 2$ en el intervalo $[0, 1/4]$, utilizando tamaños de paso $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$ junto con la solución exacta (vea el ejercicio 16 de la sección 6.1) y haga una gráfica log-log del error como una función de h.

9. Para los PVI del ejercicio 1, grafique el error total del método RK4 en $t = 1$ como una función de h, de la misma manera que en la figura 6.4.

10. Considere las ecuaciones de Hodgkin-Huxley (6.52) con los parámetros predeterminados. (A) Encuentre tan exactamente como sea posible el umbral mínimo, en microamperios, para la generación de un pico con un pulso de 1 ms. (b) ¿La respuesta cambia si el pulso dura 5 ms? (c) Experimente con la forma del pulso. ¿Un pulso triangular de área cerrada idéntica causa el mismo efecto que un pulso cuadrado? (d) Discuta la existencia de un umbral para una entrada sostenida constante.

11. Adapte el programa `orbit.m` de MATLAB para animar una solución a las ecuaciones de Lorenz mediante el método de Runge-Kutta con tamaño de paso $h = 0.001$. Dibuje la trayectoria con condición inicial $(x_0, y_0, z_0) = (5, 5, 5)$.

12. Evalúe el condicionamiento de las ecuaciones de Lorenz siguiendo dos trayectorias de dos condiciones iniciales cercanas. Considere las condiciones iniciales $(x, y, z) = (5, 5, 5)$ y otra condición inicial a una distancia $\Delta = 10^{-5}$ de la primera. Calcule ambas trayectorias mediante Runge-Kutta de cuarto orden con tamaño de paso $h = 0.001$ y calcule el factor de magnificación del error después de $t = 10$ y $t = 20$ unidades de tiempo.

13. Siga dos trayectorias de las ecuaciones de Lorenz con condiciones iniciales cercanas, como en el problema de computadora 12. Para cada una, construya una secuencia de símbolos binarios usando 0 si la trayectoria cruza el ciclo "x negativo" de la figura 6.17 y 1 si cruza el ciclo positivo. ¿Para cuántas unidades de tiempo las secuencias de símbolos de las dos trayectorias coinciden?

Comprobación en la realidad ✔ **6** ***El puente Tacoma Narrows***

Un modelo matemático que intenta capturar el incidente del puente Tacoma Narrows fue propuesto por McKenna y Tuama [2001]. El objetivo es explicar cómo las oscilaciones torsionales, o de torcimiento, pueden magnificarse mediante fuerzas que son estrictamente verticales.

Considere una carretera con anchura $2l$ que cuelga entre dos cables suspendidos, como en la figura 6.18(a). Se considerará un corte bidimensional del puente, haciendo caso omiso de la dimensión de la longitud del puente para este modelo, puesto que sólo se tiene interés en el movimiento de lado a lado. En reposo, la carretera cuelga a cierta altura de equilibrio debido a la gravedad; sea y la distancia actual a la que el centro de la carretera cuelga por debajo de este equilibrio.

La ley de Hooke postula una respuesta lineal, lo que significa que la fuerza de recuperación que aplican los cables será proporcional a la desviación. Sea θ el ángulo que forma la carretera con la horizontal. Hay dos cables de suspensión, estirados $y - l \operatorname{sen}\theta$ y $y + l \operatorname{sen}\theta$ desde el equilibrio, respectivamente. Suponga un término de amortiguamiento viscoso que es proporcional a la velocidad. Si se usa la ley de Newton $F = ma$ y se denota la constante de Hooke con K, las ecuaciones de movimiento para y y θ son las siguientes:

$$y'' = -dy' - \left[\frac{K}{m}(y - l\operatorname{sen}\theta) + \frac{K}{m}(y + l\operatorname{sen}\theta)\right]$$

$$\theta'' = -d\theta' + \frac{3\cos\theta}{l}\left[\frac{K}{m}(y - l\operatorname{sen}\theta) - \frac{K}{m}(y + l\operatorname{sen}\theta)\right].$$

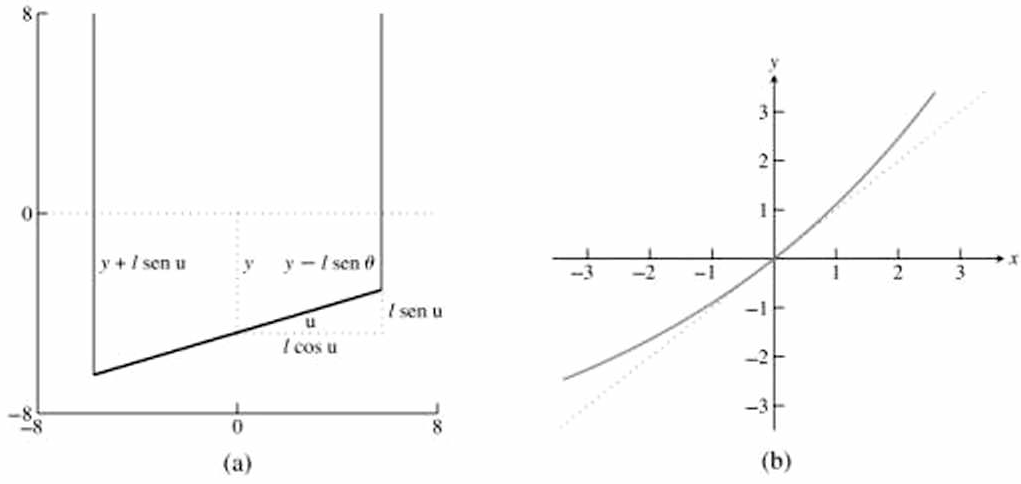


Figura 6.18 Esquema del modelo McKenna-Tuama para el puente Tacoma Narrows. (a) Indica mediante y la distancia desde el centro de masa de la carretera hasta su posición de equilibrio, y con θ el ángulo que forma la carretera con la horizontal. (b) Curva de la ley exponencial de Hooke, $f(y) = (K/a)(e^{ay} - 1)$.

Sin embargo, la ley de Hooke está diseñada para resortes, donde la fuerza de restauración es más o menos igual si los resortes se comprimen o se estiran. McKenna y Tuama suponen que los cables jalan con más fuerza cuando se estiran que lo que empujan cuando se comprimen. (Piense en una cadena como un ejemplo extremo). Ellos reemplazan la fuerza de restauración lineal de la ley Hooke $f(y) = Ky$ por una fuerza no lineal $f(y) = (K/a)(e^{ay} - 1)$, como se muestra en la figura 6.18(b). Ambas funciones tienen la misma pendiente K en $y = 0$, pero para la fuerza no lineal, una y positiva (cable estirado) causa una fuerza de restauración mayor que la correspondiente y negativa (cable aflojado). Al hacer esta sustitución en las ecuaciones anteriores se obtiene

$$y'' = -dy' - \frac{K}{ma}\left[e^{a(y-l\,\mathrm{sen}\,\theta)} - 1 + e^{a(y+l\,\mathrm{sen}\,\theta)} - 1\right]$$
$$\theta'' = -d\theta' + \frac{3\cos\theta}{l}\frac{K}{ma}\left[e^{a(y-l\,\mathrm{sen}\,\theta)} - e^{a(y+l\,\mathrm{sen}\,\theta)}\right]. \qquad (6.54)$$

Como lo indican estas ecuaciones, la condición $y = y' = \theta = \theta' = 0$ es un equilibrio. Ahora es necesario considerar al viento. Añada el término de esfuerzo $0{,}2W$ sen wt al lado derecho de la ecuación y, donde W es la velocidad del viento en km/h. Esto añade una oscilación estrictamente vertical al puente.

Pueden hacerse estimaciones útiles de las constantes físicas. La masa de la carretera con longitud de un pie era de unos 2500 kg y la constante de resorte K se estimó en 1000 newtons. La carretera tenía unos 12 metros de ancho. Para esta simulación, el coeficiente de amortiguación se fijó en $d = 0.01$ y el coeficiente de no linealidad de Hooke en $a = 0.2$. Un observador contó 38 oscilaciones verticales del puente en un minuto poco antes del colapso —considere $w = 2\pi(38/60)$. Estos coeficientes son sólo supuestos, pero son suficientes para mostrar los rangos de movimiento que tienden a coincidir con la evidencia fotográfica de las oscilaciones finales del puente. El código de MATLAB que ejecuta el modelo (6.54) es el siguiente:

```
%Programa 6.6 Programa de animación para puente usando solucionador de PVI
%Entradas: Intervalo de tiempo inter,
% ic= [y(1,1) y(1,2) y(1,3) y(1,4)],
% número de pasos n, pasos por punto graficado p
%Llama un método de un paso como trapstep.m
%Uso de ejemplo: tacoma([0 1000],[1 0 0.001 0],25000,5)
function tacoma(inter,ic,n,p)
```

```
clf                          % limpia ventana de figuras
h=(inter(2)-inter(1))/n;
y(1,:)=ic;                   % ingresa conds iniciales en y
t(1)=inter(1);len=6;
set(gca,'XLim',[-8 8],'YLim',[-8 8], ...
   'XTick',[-8 0 8],'YTick',[-8 0 8], ...
   'Drawmode','fast','Visible','on','NextPlot','add');
cla;                         % limpia pantalla
axis square                  % hace relación de aspecto 1-1
road=line('color','b','LineStyle','-','LineWidth',5,...
    'erase','xor','xdata',[],'ydata',[]);
lcable=line('color','r','LineStyle','-','LineWidth',1,...
    'erase','xor','xdata',[],'ydata',[]);
rcable=line('color','r','LineStyle','-','LineWidth',1,...
    'erase','xor','xdata',[],'ydata',[]);
for k=1:n
  for i=1:p
    t(i+1)=t(i)+h;
    y(i+1,:)=trapstep(t(i),y(i,:),h);
  end
  y(1,:)=y(p+1,:);t(1)=t(p+1);
  z1(k)=y(1,1);z3(k)=y(1,3);
  c=len*cos(y(1,3));s=len*sin(y(1,3));
  set(road,'xdata',[-c c],'ydata',[-s-y(1,1) s-y(1,1)])
  set(lcable,'xdata',[-c -c],'ydata',[-s-y(1,1) 8])
  set(rcable,'xdata',[c c],'ydata',[s-y(1,1) 8])
  drawnow; pause(h)
  end

function y=trapstep(t,x,h)
%un paso del método del trapecio
z1=ydot(t,x);
g=x+h*z1;
z2=ydot(t+h,g);
y=x+h*(z1+z2)/2;

function ydot=ydot(t,y)
len=6;a=0.2; W=80; omega=2*pi*38/60;
a1=exp(a*(y(1)-len*sin(y(3))));
a2=exp(a*(y(1)+len*sin(y(3))));
ydot(1)=y(2);
ydot(2)=-0.01*y(2)-0.4*(a1+a2-2)/a+0.2*W*sin(omega*t);
ydot(3)=y(4);
ydot(4)=-0.01*y(4)+1.2*cos(y(3))*(a1-a2)/(len*a);
```

Ejecute `tacoma.m` con los valores predeterminados de los parámetros para ver el fenómeno postulado anteriormente. Si el ángulo θ de la carretera se fija en cualquier valor pequeño distinto de cero, las fuerzas verticales causan que θ alcance gradualmente un valor macroscópico, lo que conduce a la torsión significativa de la carretera. El punto interesante es que no hay ninguna fuerza de torsión aplicada en la ecuación; el "modo torsional" inestable se excita por completo mediante un forzamiento vertical.

Actividades sugeridas:

1. Ejecute `tacoma.m` con la velocidad del viento $W = 80$ km/h y las condiciones iniciales $y = y' = \theta' = 0$, $\theta = 0{,}001$. El puente es estable en la dimensión torsional si las pequeñas perturbaciones en θ se terminan; y es inestable si éstas crecen mucho más allá de su tamaño original. ¿Qué ocurre para este valor de W?

2. Reemplace el método del trapecio por Runge-Kutta de cuarto orden para mejorar la exactitud. Además, agregue nuevas ventanas de figuras para graficar $y(t)$ y $\theta(t)$.

3. El sistema es torsionalmente estable para $W = 50$ km/h. Encuentre el factor de magnificación para un ángulo inicial pequeño. Es decir, establezca $\theta(0) = 10^{-3}$ y encuentre la relación del ángulo máximo $\theta(t)$, $0 \leq t < \infty$, con $\theta(0)$. ¿El factor de magnificación es aproximadamente constante para los ángulos iniciales $\theta(0) = 10^{-4}, 10^{-5}, \ldots$?

4. Encuentre la velocidad mínima W del viento para que una pequeña perturbación $\theta(0) = 10^{-3}$ tenga un factor de magnificación de 100 o más. ¿Puede definirse un factor de magnificación constante para esta W?

5. Diseñe e implemente un método para el cálculo de la velocidad mínima del viento del paso 4, hasta 0.5×10^{-3} km/h. Es posible que desee utilizar un método del capítulo 1.

6. Pruebe algunos de los valores más grandes de W. ¿Todos los ángulos iniciales muy pequeños crecen con el tiempo hasta tamaños catastróficos?

7. ¿Cuál es el efecto de aumentar el coeficiente de amortiguación? Duplique el valor actual y compare la A crítica cuando $w = 3$. ¿Puede sugerir posibles cambios en el diseño que podrían haber hecho al puente menos susceptible a la torsión?

Este proyecto es un ejemplo de las matemáticas experimentales. Con estas ecuaciones es demasiado difícil obtener soluciones de forma cerrada e incluso es demasiado difícil comprobar los resultados cualitativos. Si se emplean métodos confiables para resolver EDO, es posible generar trayectorias numéricas para parámetros diferentes a fin de ilustrar los tipos de fenómenos disponibles para el modelo. Utilizados de esta manera, los modelos de ecuaciones diferenciales pueden predecir el comportamiento y arrojar luz sobre los mecanismos de los problemas científicos e ingenieriles. ✔

6.5 MÉTODOS CON TAMAÑO DE PASO VARIABLE

Hasta este punto, el tamaño de paso h se ha tratado como una constante en la aplicación de los métodos para resolver EDO. Sin embargo, no hay ninguna razón por la que h no pueda cambiar durante el proceso de solución. Una buena razón para querer cambiar el tamaño del paso es una solución que se mueve entre periodos de cambio lento y periodos de cambio rápido. El establecimiento de un tamaño de paso fijo suficientemente pequeño como para seguir con precisión los cambios rápidos, puede significar que el resto de la solución se vuelva demasiado lenta.

En esta sección, se analizan las estrategias para controlar el tamaño de paso de los métodos para resolver EDO. El enfoque más común utiliza dos métodos de diferentes órdenes, llamados pares integrados.

6.5.1 Pares integrados de Runge-Kutta

La idea clave de un método de tamaño de paso variable es monitorear el error producido por el paso actual. El usuario establece una tolerancia al error que debe cumplirse para el paso actual. Después, el método está diseñado para (1) rechazar el paso y reducir el tamaño de paso si la tolerancia al error se excede, o (2) si la tolerancia al error se cumple, aceptar el paso y luego elegir un tamaño de paso h que debe ser apropiado para el siguiente paso. La necesidad clave es encontrar algún modo de aproximar el error cometido en cada paso. Primero se supondrá que se ha encontrado tal modo y se explicará cómo cambiar el tamaño de paso.

La forma más sencilla de variar el tamaño de paso es duplicar o partir a la mitad el tamaño de paso, dependiendo del error actual. Compare la estimación del error e_i, o la estimación del error relativo $e_i/|w_i|$, con la tolerancia al error. (Aquí, como en el resto de esta sección, se supondrá que

el sistema de EDO que se está resolviendo consiste en una ecuación. Resulta bastante fácil generalizar las ideas de esta sección a dimensiones superiores). Si la tolerancia no se cumple, el paso se repite con un tamaño de paso nuevo igual a $h_i/2$. Si la tolerancia se cumple demasiado bien (por ejemplo, si el error es menor a 1/10 de la tolerancia) se acepta el paso y se duplica el tamaño de paso para el siguiente paso.

De esta manera, el tamaño de paso se ajusta automáticamente a un tamaño que mantiene el error (relativo) de truncamiento local cerca del nivel requerido por el usuario. El uso del error absoluto o relativo depende del contexto; una buena técnica de propósito general es utilizar el híbrido $e_i/\text{máx}(|w_i|, \theta)$ para ser comparado con la tolerancia al error, en donde la constante $\theta > 0$ protege contra valores muy pequeños de w_i.

Una forma más sofisticada de elegir el tamaño de paso apropiado se deduce del conocimiento del orden del método utilizado. Suponga que el método tiene orden p, de modo que el error de truncamiento local $e_i = O(h^{p+1})$. Sea T la tolerancia al error relativo permitida por el usuario para cada paso. Eso significa que el objetivo es asegurar $e_i/|w_i| < T$.

Si se cumple el objetivo $e_i/|w_i| < T$, entonces el paso se acepta y se requiere un tamaño de paso nuevo para el siguiente paso. Si se supone que

$$e_i \approx ch_i^{p+1} \tag{6.55}$$

para alguna constante c, el tamaño de paso h que mejor cumple la tolerancia satisface

$$T|w_i| = ch^{p+1}. \tag{6.56}$$

Al resolver las ecuaciones (6.55) y (6.56) para h y c se obtiene

$$h_* = 0.8\left(\frac{T|w_i|}{e_i}\right)^{\frac{1}{p+1}} h_i, \tag{6.57}$$

donde se ha añadido un factor de seguridad de 0.8 para ser conservadores. Por lo tanto, el siguiente tamaño de paso se establece en $h_{i+1} = h_*$.

Por otro lado, si el objetivo $e_i/|w_i| < T$ no se cumple para el error relativo, entonces h_i se establece en h_* para un segundo intento. Esto debe ser suficiente, debido al factor de seguridad. Sin embargo, si el segundo intento tampoco cumple el objetivo, entonces el tamaño de paso simplemente se corta a la mitad. Esto continúa hasta que se logre el objetivo. Como se ha indicado para los propósitos generales, el error relativo debe sustituirse por $e_i/\text{máx}(|w_i|, \theta)$.

Tanto los métodos simples como los sofisticados que se han descrito dependen en gran medida de algún modo de estimar el error en el paso actual del método utilizado $e_i = |w_{i+1} - y_{i+1}|$. Una limitación importante es obtener la estimación sin requerir una gran cantidad de cálculo adicional.

La forma más utilizada para obtener dicha estimación del error consiste en ejecutar un método en la solución de la EDO de orden superior en paralelo con otro método que se considere el principal. El método de estimación de orden superior para w_{i+1} —llámelo z_{i+1}— será mucho más preciso que el original w_{i+1}, de modo que la diferencia

$$e_{i+1} \approx |z_{i+1} - w_{i+1}| \tag{6.58}$$

se utiliza como una estimación del error para el paso actual de t_i a t_{i+1}.

Siguiendo esta idea se han desarrollado varios "pares" de Runge-Kutta, uno de orden p y otro de orden $p + 1$, que comparten la mayoría de los cálculos necesarios. De este modo, el costo adicional por controlar el tamaño de paso se mantiene bajo. Dicho par suele llamarse **par Runge-Kutta integrado**.

► EJEMPLO 6.19 RK2/3, un ejemplo de un par de Runge-Kutta integrado de orden 2/orden 3.

El método del trapecio explícito puede combinarse con un método de RK de tercer orden para crear un par integrado adecuado a fin de controlar el tamaño de paso. Establezca

$$w_{i+1} = w_i + h\frac{s_1 + s_2}{2}$$
$$z_{i+1} = w_i + h\frac{s_1 + 4s_3 + s_2}{6},$$

donde

$$s_1 = f(t_i, w_i)$$
$$s_2 = f(t_i + h, w_i + hs_1)$$
$$s_3 = f\left(t_i + \frac{1}{2}h, w_i + \frac{1}{2}h\frac{s_1 + s_2}{2}\right).$$

En las ecuaciones anteriores, w_{i+1} es el paso del trapecio y z_{i+1} representa un método de tercer orden, que requiere las tres etapas de Runge-Kutta mostradas. El método de tercer orden es sólo una aplicación de la regla de Simpson para la integración numérica en el contexto de las ecuaciones diferenciales. A partir de los dos métodos, puede encontrarse una estimación del error restando las dos aproximaciones:

$$e_{i+1} \approx |w_{i+1} - z_{i+1}| = \left|h\frac{s_1 - 2s_3 + s_2}{3}\right|. \tag{6.59}$$

El uso de esta estimación para el error de truncamiento local permite aplicar cualquiera de los protocolos de control del tamaño de paso anteriormente descritos. Tenga en cuenta que la estimación del error de truncamiento local para el método del trapecio se logra a costa de una sola evaluación adicional de f, utilizada para calcular S_3. ◄

Aunque el protocolo del tamaño de paso se trabajó para w_{i+1}, tiene incluso un mejor sentido utilizar la aproximación de orden superior z_{i+1} para avanzar de paso, puesto que ya está disponible. Esto se llama **extrapolación local**.

► EJEMPLO 6.20 El par integrado de Bogacki-Shampine de segundo/tercer orden.

MATLAB utiliza un par integrado diferente en su comando `ode23`. Sean

$$s_1 = f(t_i, w_i)$$
$$s_2 = f\left(t_i + \frac{1}{2}h, w_i + \frac{1}{2}hs_1\right)$$
$$s_3 = f\left(t_i + \frac{3}{4}h, w_i + \frac{3}{4}hs_2\right)$$
$$z_{i+1} = w_i + \frac{h}{9}(2s_1 + 3s_2 + 4s_3)$$
$$s_4 = f(t + h, z_{i+1})$$
$$w_{i+1} = w_i + \frac{h}{24}(7s_1 + 6s_2 + 8s_3 + 3s_4). \tag{6.60}$$

Puede comprobarse que z_{i+1} es una aproximación de tercer orden y que w_{i+1}, a pesar de tener cuatro etapas, es de segundo orden. La estimación del error necesaria para controlar el tamaño de paso es

$$e_{i+1} = |z_{i+1} - w_{i+1}| = \frac{h}{72}|-5s_1 + 6s_2 + 8s_3 - 9s_4|. \tag{6.61}$$

Observe que si se acepta, s_4 se convierte en s_1 en el siguiente paso, por lo que no existen etapas desperdiciadas (de cualquier modo, se requieren al menos tres etapas para un Runge-Kutta de tercer orden). Este diseño de un método de segundo orden se llama FSAL, por "primero igual al último" (First Same As Last). ◄

6.5.2 Métodos de cuarto y quinto orden

► EJEMPLO 6.21 Par integrado Runge-Kutta-Fehlberg de cuarto y quinto orden.

$$
\begin{aligned}
s_1 &= f(t_i, w_i)\\
s_2 &= f\left(t_i + \frac{1}{4}h, w_i + \frac{1}{4}hs_1\right)\\
s_3 &= f\left(t_i + \frac{3}{8}h, w_i + \frac{3}{32}hs_1 + \frac{9}{32}hs_2\right)\\
s_4 &= f\left(t_i + \frac{12}{13}h, w_i + \frac{1932}{2197}hs_1 - \frac{7200}{2197}hs_2 + \frac{7296}{2197}hs_3\right)\\
s_5 &= f\left(t_i + h, w_i + \frac{439}{216}hs_1 - 8hs_2 + \frac{3680}{513}hs_3 - \frac{845}{4104}hs_4\right)\\
s_6 &= f\left(t_i + \frac{1}{2}h, w_i - \frac{8}{27}hs_1 + 2hs_2 - \frac{3544}{2565}hs_3 + \frac{1859}{4104}hs_4 - \frac{11}{40}hs_5\right)\\
w_{i+1} &= w_i + h\left(\frac{25}{216}s_1 + \frac{1408}{2565}s_3 + \frac{2197}{4104}s_4 - \frac{1}{5}s_5\right)\\
z_{i+1} &= w_i + h\left(\frac{16}{135}s_1 + \frac{6656}{12825}s_3 + \frac{28561}{56430}s_4 - \frac{9}{50}s_5 + \frac{2}{55}s_6\right). \qquad (6.62)
\end{aligned}
$$

Puede comprobarse que z_{i+1} es una aproximación de quinto orden y que w_{i+1} es de cuarto orden. La estimación del error necesaria para controlar el tamaño de paso es

$$
e_{i+1} = |z_{i+1} - w_{i+1}| = h\left|\frac{1}{360}s_1 - \frac{128}{4275}s_3 - \frac{2197}{75240}s_4 + \frac{1}{50}s_5 + \frac{2}{55}s_6\right|. \qquad (6.63)
$$

◄

El método de Runge-Kutta-Fehlberg (RKF45) es actualmente el método de tamaño de paso variable de un paso más conocido. La implementación es sencilla dadas las fórmulas anteriores. El usuario debe definir una tolerancia T al error relativo y un tamaño de paso h inicial. Después de calcular w_1, z_1 y e_1, la prueba del error relativo

$$
\frac{e_i}{|w_i|} < T \qquad (6.64)
$$

se verifica para $i = 1$. Si tiene éxito, la nueva w_1 se sustituye por la versión extrapolada localmente de z_1 y el programa pasa a la etapa siguiente. Por otro lado, si la prueba del error relativo (6.64) falla, el paso se intenta de nuevo con el tamaño de paso h dado por (6.57) con $p = 4$, el orden del método que produce w_i. (La repetición de la falla, que es poco probable, se trata reduciendo el tamaño de paso a la mitad hasta que se alcanza el éxito). En cualquier caso, el tamaño de paso h_1 para el paso siguiente debe calcularse a partir de (6.57).

► EJEMPLO 6.22 Par integrado de Dormand-Prince de cuarto/quinto orden.

$$
\begin{aligned}
s_1 &= f(t_i, w_i)\\
s_2 &= f\left(t_i + \frac{1}{5}h, w_i + \frac{1}{5}hs_1\right)\\
s_3 &= f\left(t_i + \frac{3}{10}h, w_i + \frac{3}{40}hs_1 + \frac{9}{40}hs_2\right)
\end{aligned}
$$

$$
\begin{aligned}
s_4 &= f\left(t_i + \frac{4}{5}h, w_i + \frac{44}{45}hs_1 - \frac{56}{15}hs_2 + \frac{32}{9}hs_3\right)\\
s_5 &= f\left(t_i + \frac{8}{9}h, w_i + h\left(\frac{19372}{6561}s_1 - \frac{25360}{2187}s_2 + \frac{64448}{6561}s_3 - \frac{212}{729}s_4\right)\right)\\
s_6 &= f\left(t_i + h, w_i + h\left(\frac{9017}{3168}s_1 - \frac{355}{33}s_2 + \frac{46732}{5247}s_3 + \frac{49}{176}s_4 - \frac{5103}{18656}s_5\right)\right)\\
z_{i+1} &= w_i + h\left(\frac{35}{384}s_1 + \frac{500}{1113}s_3 + \frac{125}{192}s_4 - \frac{2187}{6784}s_5 + \frac{11}{84}s_6\right)\\
s_7 &= f(t_i + h, z_{i+1})\\
w_{i+1} &= w_i + h\left(\frac{5179}{57600}s_1 + \frac{7571}{16695}s_3 + \frac{393}{640}s_4 - \frac{92097}{339200}s_5 + \frac{187}{2100}s_6 + \frac{1}{40}s_7\right).
\end{aligned}
\tag{6.65}
$$

Puede comprobarse que z_{i+1} es una aproximación de quinto orden y que w_{i+1} es de cuarto orden. La estimación del error necesaria para controlar el tamaño de paso es

$$
\begin{aligned}
e_{i+1} &= |z_{i+1} - w_{i+1}|\\
&= h\left|\frac{71}{57600}s_1 - \frac{71}{16695}s_3 + \frac{71}{1920}s_4 - \frac{17253}{339200}s_5 + \frac{22}{525}s_6 - \frac{1}{40}s_7\right|.
\end{aligned}
\tag{6.66}
$$

Una vez más, se usa la extrapolación local, lo que significa que el paso se hace avanzar con z_{i+1} en vez de w_{i+1}. Tenga en cuenta que, de hecho, no hay necesidad de calcular w_{i+1} (para el control del error sólo se requiere e_{i+1}). Éste es un método FSAL, como el método Bogacki-Shampine, ya que s_7 se convierte en s_1 en el siguiente paso, si éste es aceptado. No hay etapas desperdiciadas: puede demostrarse que se necesitan al menos seis etapas para un Runge-Kutta de quinto orden. ◀

El comando `ode45` de MATLAB utiliza el par integrado de Dormand-Prince junto con el control del tamaño de paso, así como se acaba de describir. El usuario puede establecer la tolerancia T como desee. El lado derecho de la ecuación diferencial debe especificarse como una función de MATLAB. Por ejemplo, los comandos

```
>> opts=odeset('RelTol',1e-4,'Refine',1,'MaxStep',1);
>> [t,y]=ode45(@(t,y) t*y+t^3,[0 1],1,opts);
```

resolverán el problema de valor inicial del ejemplo 6.1 con condición inicial $y_0 = 1$ y tolerancia al error relativo $T = 0.0001$. Si el parámetro `RelTol` no está indicado, se usa el valor predeterminado 0.001. Tenga en cuenta que la entrada a la función *f* `ode45` debe ser una función de dos variables, en este caso t y y, incluso si una de ellas está ausente en la definición de la función.

La salida de `ode45` para este problema, utilizando la definición de parámetros anterior, es

iteración	t_i	w_i	y_i	e_i
0	0.00000000	1.00000000	1.00000000	0.00000000
1	0.54021287	1.17946818	1.17946345	0.00000473
2	1.00000000	1.94617812	1.94616381	0.00001431

Si se usa una tolerancia relativa de 10^{-6}, resulta la siguiente salida:

iteración	t_i	w_i	y_i	e_i
0	0.00000000	1.00000000	1.00000000	0.00000000
1	0.21506262	1.02393440	1.02393440	0.00000000
2	0.43012524	1.10574441	1.10574440	0.00000001
3	0.68607729	1.32535658	1.32535653	0.00000005
4	0.91192246	1.71515156	1.71515144	0.00000012
5	1.00000000	1.94616394	1.94616381	0.00000013

Las soluciones aproximadas cumplen sobradamente la tolerancia al error relativo debido a la extrapolación local, lo que significa que se usará z_{i+1} en vez de w_{i+1}, aun cuando el tamaño de paso esté diseñado como suficiente para w_{i+1}. Esto es lo mejor que puede hacerse; si se tuviera una estimación del error para z_{i+1}, ésta podría emplearse para ajustar el tamaño de paso de mejor manera, pero no se cuenta con ella. Tenga en cuenta también que las soluciones se detienen exactamente en el extremo del intervalo [0, 1], puesto que `ode45` detecta el extremo del intervalo y trunca el paso según se requiera.

Para ver cómo `ode45` hace su selección del tamaño de paso, fue necesario desactivar algunas configuraciones básicas predeterminadas, con el comando `odeset`. Por lo general, el parámetro `Refine` aumenta el número de valores reportados para la solución más allá de lo que calcula el método, para hacer una gráfica más atractiva, cuando la salida se utiliza para ese propósito. El valor predeterminado es 4, lo que hace que el número de puntos que deben proporcionarse como salida se multipliquen por cuatro. El parámetro `MaxStep` establece un límite superior en el tamaño de paso h, y define por omisión la longitud del intervalo como un décimo. Si se emplearan los valores predeterminados para estos dos parámetros se tendría un tamaño de paso $h = 0.1$ y después de refinarse por un factor de 4, la solución se mostraría con un tamaño de paso de 0.025. De hecho, la ejecución del comando sin una variable de salida específica, como en el código

```
>> opts=odeset('RelTol',1e-6);
>> ode45(@(t,y) t*y+t^3,[0 1],1,opts);
```

ocasiona que MATLAB grafique automáticamente la solución en una malla con tamaño de paso constante de 0.025, como se muestra en la figura 6.19.

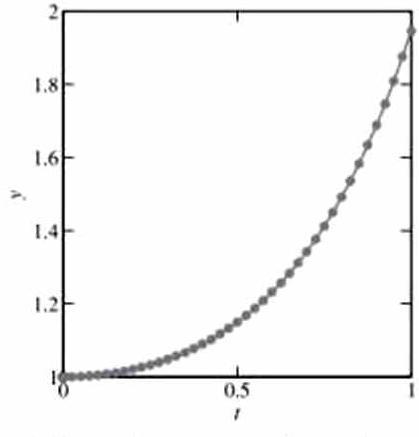


Figura 6.19 Comando `ode45` de MATLAB. Cálculo de la solución al problema de valor inicial del ejemplo 6.1, correcto hasta 10^{-6} decimales.

Una forma alternativa de definir el lado derecho de la función f es crear un archivo de función, por ejemplo `f.m` y utilizar el carácter @ para designar la función de inicio:

```
function y=f(t,y)
y=t*y+t^3;
```

El comando

```
>> [t,y]=ode45(@f,[0 1],1,opts);
```

causa que `ode45` funcione como antes. Esta alternativa será conveniente cuando aumente el número de variables independientes en las ecuaciones diferenciales.

Si bien es tentador coronar a los métodos de Runge-Kutta con tamaño de paso variable como los campeones entre los métodos para resolver EDO, hay algunos tipos de ecuaciones que no son muy bien manejadas por estos métodos. A continuación se presenta un ejemplo muy sencillo pero problemático:

► **EJEMPLO 6.23** Utilice `ode45` para resolver el problema de valor inicial con una tolerancia relativa de 10^{-4}.

$$\begin{cases} y' = 10(1 - y) \\ y(0) = 1/2 \\ t \text{ en } [0, 100]. \end{cases} \tag{6.67}$$

Esto puede lograrse con las siguientes tres líneas de código en MATLAB:

```
>> opts=odeset('RelTol',1e-4);
>> [t,y]=ode45(@(t,y) 10*(1-y),[0 100],.5,opts);
>> length(t)

ans=        1241
>>
```

Se tiene impreso el número de iteraciones pero parece excesivo. La solución del problema de valor inicial es fácil de determinar: $y(t) = 1 - e^{-10t}/2$. Para $t > 1$, la solución ya ha alcanzado su equilibrio 1 dentro de 4 posiciones decimales y nunca se mueve más allá de 1. Sin embargo, `ode45` se mueve a paso de tortuga, utilizando un tamaño de paso promedio de menos de 0.1. ¿Por qué una selección conservadora del tamaño de paso para una solución fácil?

Parte de la respuesta se hace evidente al ver la salida de `ode45` en la figura 6.20. Aunque la solución está muy cerca de 1, el método se sobrepasa continuamente al tratar de aproximarse aún más. La ecuación diferencial es "rígida", un término que se define con mayor detalle en la siguiente sección. Para las ecuaciones rígidas, una estrategia diferente en la solución numérica aumenta en

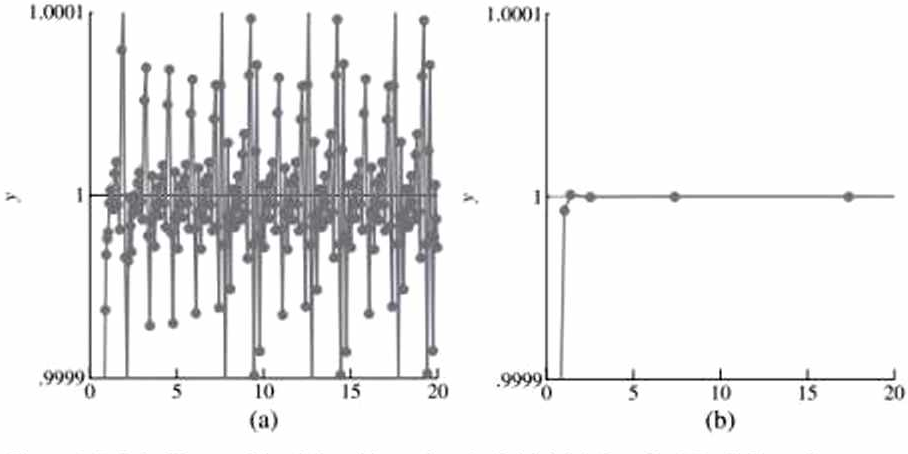


Figura 6.20 Solución numérica del problema de valor inicial del ejemplo 6.23. (a) El uso de `ode45` requiere más de 10 pasos por unidad de tiempo para permanecer dentro de la tolerancia relativa 10^{-4}. (b) Con `ode23s`, se necesitan muchos menos pasos.

gran medida la eficiencia de los problemas. Por ejemplo, observe la diferencia en las iteraciones necesarias cuando se utiliza uno de los métodos rígidos usando MATLAB:

```
>> opts=odeset('RelTol',1e-4);
>> [t,y]=ode23s(@(t,y) 10*(1-y),[0 100],.5,opts);
>> length(t)

ans=
        39
```

En la figura 6.20(b) se representan los puntos de la solución del código `ode23s` de MATLAB. Se requieren relativamente pocos puntos para mantener la solución numérica dentro de la tolerancia. En la siguiente sección se investigará cómo construir métodos que controlen este tipo de dificultades. ◄

6.5 Problemas de computadora

1. Escriba una implementación de RK23 (ejemplo 6.19) en MATLAB y aplíquela para aproximar las soluciones de los PVI del ejercicio 3 de la sección 6.1 con una tolerancia relativa de 10^{-8} en [0, 1]. Haga que el programa se detenga justo en el punto extremo $t = 1$. Registre el tamaño de paso máximo utilizado y el número de iteraciones.
2. Compare los resultados del problema de computadora 1 con la aplicación de `ode23` de MATLAB para el mismo problema.
3. Realice los pasos del problema de computadora 1 para el método de Runge-Kutta-Fehlberg RKF45.
4. Compare los resultados del problema de computadora 3 con la aplicación de `ode45` de MATLAB para el mismo problema.
5. Aplique una implementación de RKF45 en MATLAB para aproximar las soluciones a los sistemas del ejercicio 1 de la sección 6.3 con una tolerancia relativa de 10^{-6} en [0, 1]. Reporte el tamaño de paso máximo utilizado y el número de iteraciones.

6.6 MÉTODOS IMPLÍCITOS Y ECUACIONES RÍGIDAS

Los métodos para resolver ecuaciones diferenciales que se han presentado hasta ahora son **explícitos**, lo que significa que existe una fórmula explícita para la aproximación a la nueva w_{i+1} en términos de los datos conocidos, como h, t_i y w_i. En ocasiones algunas ecuaciones diferenciales son mal abordadas por los métodos explícitos y el primer objetivo consiste en explicar por qué. En el ejemplo 6.23, un método sofisticado con tamaño de paso variable parece emplear la mayor parte de su potencia pasando de largo la solución correcta en una u otra dirección.

El fenómeno de la rigidez puede entenderse con más facilidad en un contexto más simple. En consecuencia, se inicia con el método de Euler.

► **EJEMPLO 6.24** Aplique el método de Euler al ejemplo 6.23.

El método de Euler para el lado derecho de $f(t, r) = 10(1 - y)$ con tamaño de paso h es

$$\begin{aligned} w_{i+1} &= w_i + hf(t_i, w_i) \\ &= w_i + h(10)(1 - w_i) \\ &= w_i(1 - 10h) + 10h. \end{aligned} \tag{6.68}$$

Como la solución es $y(t) = 1 - e^{-10t}/2$, la solución aproximada debe acercarse a 1 en el largo plazo. Aquí se obtiene un poco de ayuda del capítulo 1. Observe que (6.68) puede verse como una iteración de punto fijo con $g(x) = x(1 - 10h) - 10h$. Esta iteración converge al punto fijo en $x = 1$ siempre que $|g'(1)| = |1 - 10h| < 1$. Al resolver esta desigualdad se obtiene $0 < h < 0.2$. Para cualquier h más grande, el punto fijo 1 se aleja y no se tendrá la solución. ◀

En la figura 6.21 se muestra este efecto para el ejemplo 6.24. La solución es muy dócil: un equilibrio atrayente en $y = 1$. Un paso de Euler de tamaño $h = 0.3$ tiene dificultad para encontrar el equilibrio debido a que la pendiente cercana a la solución cambia mucho entre el comienzo y el final del intervalo h. Esto causa un alejamiento a la solución numérica.

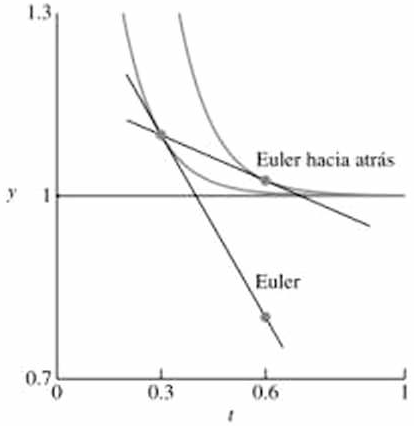


Figura 6.21 Comparación de los pasos de Euler y Euler hacia atrás. La ecuación diferencial del ejemplo 6.23 es rígida. La solución de equilibrio $y = 1$ está rodeada por otras soluciones de gran curvatura (cuya pendiente cambia con rapidez). El paso de Euler rebasa la solución, mientras que el paso de Euler hacia atrás es más consistente con la dinámica del sistema.

Las ecuaciones diferenciales con esta propiedad (que las soluciones atrayentes están rodeadas de soluciones cercanas que cambian con rapidez) se denominan **rígidas**. Esto suele ser una señal de la existencia de múltiples escalas de tiempo en el sistema. Cuantitativamente, esto corresponde a una parte lineal del lado derecho f de la ecuación diferencial, en la variable y, que es grande y negativa. (Para un sistema de ecuaciones, esto corresponde a un valor propio de la parte lineal que es grande y negativo). Esta definición es un poco relativa, pero es la naturaleza de la rigidez (entre más negativa sea, menor será el tamaño de paso para evitar el alejarse). Para el ejemplo 6.24, la rigidez se mide mediante la evaluación de $\partial f/\partial y = -10$ en la solución de equilibrio $y = 1$.

Una manera de resolver el problema que se representa en la figura 6.21 consiste en transferir de algún modo información desde el lado derecho del intervalo $[t_i, t_i + \mathrm{h}]$, en vez de confiar sólo en la información del lado izquierdo. Ésa es la motivación detrás de la siguiente variación del método de Euler:

Método de Euler hacia atrás

$$w_0 = y_0$$
$$w_{i+1} = w_i + hf(t_{i+1}, w_{i+1}). \tag{6.69}$$

Observe la diferencia: mientras que el método de Euler emplea la pendiente del extremo izquierdo para pasar a través del intervalo, Euler hacia atrás intenta cruzar el intervalo de modo que la pendiente en el extremo derecho sea la correcta.

Para lograr esta mejora debe pagarse un precio. Euler hacia atrás es el primer ejemplo de un método **implícito**, lo que significa que el método no da directamente una fórmula para la nueva

aproximación w_{i+1}. En vez de eso, es necesario trabajar un poco para conseguirla. Para el ejemplo $y' = 10(1 - y)$ el método de Euler hacia atrás da

$$w_{i+1} = w_i + 10h(1 - w_{i+1}),$$

que, después de algo de álgebra, puede expresarse como

$$w_{i+1} = \frac{w_i + 10h}{1 + 10h}.$$

Si, por ejemplo, se establece $h = 0.3$ el método de Euler da $w_{i+1} = (w_i + 3)/4$. Una vez más, puede evaluarse el comportamiento como una iteración de punto fijo $w \to g(w) = (w + 3)/4$. Existe un punto fijo en 1, y $g'(1) = 1/4 < 1$, lo que verifica la convergencia a la verdadera solución de equilibrio $y = 1$. A diferencia del método de Euler con $h = 0.3$, al menos la solución numérica sigue el comportamiento cualitativo correcto. De hecho, observe que la solución del método de Euler hacia atrás converge a $y = 1$, sin importar cuán grande sea el tamaño de paso h (ejercicio 3).

Debido al mejor comportamiento de los métodos implícitos como Euler hacia atrás en la presencia de ecuaciones rígidas, vale la pena realizar un trabajo adicional para evaluar el siguiente paso, a pesar de que no esté explícitamente disponible. El ejemplo 6.24 no fue difícil de resolver para w_{i+1}, debido a que la ecuación diferencial era lineal y que era posible cambiar la fórmula implícita original por una fórmula explícita para su evaluación. Sin embargo, en general esto no es posible y es necesario utilizar medios más indirectos.

Si el método implícito deja una ecuación no lineal por resolver, es necesario hacer referencia al capítulo 1. Tanto la iteración de punto fijo como el método de Newton se utilizan con frecuencia para resolver w_{i+1}. Esto significa que hay un ciclo de resolución de ecuaciones dentro del ciclo de avance de la ecuación diferencial. El siguiente ejemplo muestra cómo puede hacerse esto.

► **EJEMPLO 6.25** Aplique el método de Euler al problema de valor inicial

$$\begin{cases} y' = y + 8y^2 - 9y^3 \\ y(0) = 1/2 \\ t \text{ en } [0,3]. \end{cases}$$

Esta ecuación, como en el ejemplo anterior, tiene una solución de equilibrio $y = 1$. La derivada parcial $\partial f/\partial y = 1 + 16y - 27y^2$ se evalúa como -10 en $y = 1$, lo que identifica a esta ecuación como moderadamente rígida. Habrá un límite superior para h, similar al del ejemplo anterior, de modo que el método de Euler sea exitoso. Por lo tanto, existe una motivación para probar el método de Euler

$$\begin{aligned} w_{i+1} &= w_i + hf(t_{i+1}, w_{i+1}) \\ &= w_i + h(w_{i+1} + 8w_{i+1}^2 - 9w_{i+1}^3). \end{aligned}$$

Ésta es una ecuación no lineal en w_{i+1}, la cual debe resolverse para avanzar en la solución numérica. Si se renombra $z = w_{i+1}$, es necesario resolver la ecuación $z = w_i + h(z + 8z^2 - 9z^3)$, o bien

$$9hz^3 - 8hz^2 + (1 - h)z - w_i = 0 \tag{6.70}$$

para la incógnita z. Se usará el método de Newton.

Para iniciar el método de Newton se necesita una estimación inicial. Dos opciones que vienen a la mente son la aproximación w_i anterior y la aproximación del método de Euler para w_{i+1}. Aunque la segunda opción es accesible puesto que Euler es explícito, puede no ser la mejor opción para los problemas de rigidez, como se muestra en la figura 6.21. En este caso, se utilizará w_i como la estimación de inicio.

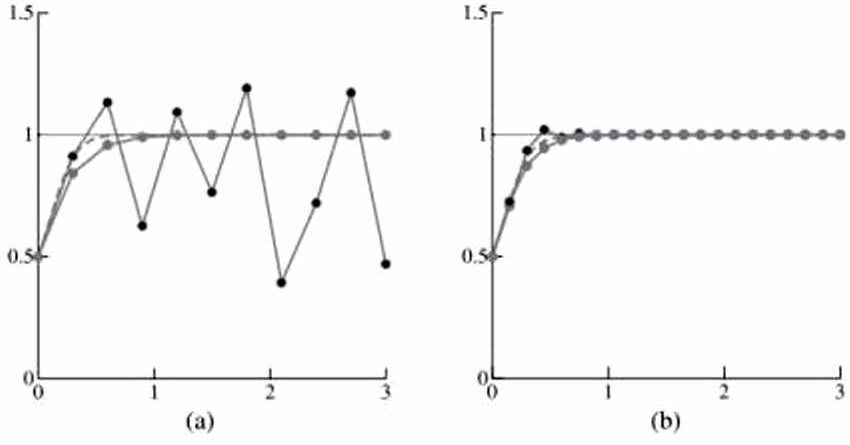


Figura 6.22 Solución numérica del problema de valor inicial del ejemplo 6.25. La solución verdadera es la curva discontinua. Los círculos negros indican la aproximación del método de Euler; los círculos grises indican Euler hacia atrás. (a) $h = 0.3$ (b) $h = 0.15$.

Al aplicar el método de Newton para (6.70) produce

$$z_{\text{nueva}} = z - \frac{9hz^3 - 8hz^2 + (1-h)z - w_i}{27hz^2 - 16hz + 1 - h}. \tag{6.71}$$

Después de evaluar (6.71), remplace z con z_{nueva} y repita. Para cada paso de Euler hacia atrás, el método de Newton se ejecuta hasta que $z_{\text{nueva}} - z$ sea menor que una tolerancia predefinida (más pequeña que los errores que se cometen al aproximar la solución de la ecuación diferencial).

En la figura 6.22 se muestran los resultados para dos tamaños de paso diferentes. Además, se muestran las soluciones numéricas del método de Euler. Es evidente que $h = 0.3$ es demasiado grande para Euler en este problema rígido. Por otro lado, cuando h se corta a 0.15, ambos métodos se desempeñan aproximadamente al mismo nivel. ◀

Los llamados métodos rígidos como Euler hacia atrás permiten un control suficiente de los errores con tamaños de paso relativamente grandes, lo que aumenta la eficiencia. `ode23s` de MATLAB es una versión de orden superior con una estrategia integrada de tamaño de paso variable.

6.6 Ejercicios

1. Use la condición inicial $y(0) = 0$ y el tamaño de paso $h = 1/4$ para calcular la aproximación de Euler hacia atrás en el intervalo [0, 1]. Encuentre el error en $t = 1$ comparando su respuesta con la solución correcta hallada en el ejercicio 4 de la sección 6.1.

   (a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

2. Encuentre todas las soluciones de equilibrio y el valor del jacobiano en el equilibrio. ¿La ecuación es rígida? (a) $y' = y - y^2$ (b) $y' = 10y - 10y^2$ (c) $y' = -10 \operatorname{sen} y$

3. Demuestre que para cada tamaño de paso h, la solución aproximada de Euler hacia atrás converge a la solución de equilibrio $y = 1$ cuando $t_i \to \infty$ para el ejemplo 6.24.

4. Considere la ecuación diferencial lineal $y' = ay + b$ para $a < 0$. (a) Encuentre el equilibrio. (b) Escriba el método de Euler hacia atrás para la ecuación. (c) Vea el método de Euler hacia atrás como una iteración de punto fijo para demostrar que la solución aproximada del método convergerá al equilibrio cuando $t \to \infty$.

6.6 Problemas de computadora

1. Aplique Euler hacia atrás, utilizando el método de Newton, para los problemas de valor inicial. ¿A cuál de las soluciones de equilibrio se acerca la solución aproximada? Aplique el método de Euler. ¿Para cuál rango aproximado de h puede usarse Euler con éxito en la convergencia hacia el equilibrio? Grafique las soluciones aproximadas dadas por Euler hacia atrás y por Euler con un tamaño de paso grande.

$$\text{(a)} \begin{cases} y' = y^2 - y^3 \\ y(0) = 1/2 \\ t \text{ en } [0, 20] \end{cases} \qquad \text{(b)} \begin{cases} y' = 6y - 6y^2 \\ y(0) = 1/2 \\ t \text{ en } [0, 20] \end{cases}$$

2. Realice los pasos del problema de computadora 1 para los siguientes problemas de valor inicial:

$$\text{(a)} \begin{cases} y' = 6y - 3y^2 \\ y(0) = 1/2 \\ t \text{ en } [0, 20] \end{cases} \qquad \text{(b)} \begin{cases} y' = 10y^3 - 10y^4 \\ y(0) = 1/2 \\ t \text{ en } [0, 20] \end{cases}$$

6.7 MÉTODOS DE VARIOS PASOS

La familia Runge-Kutta que se ha estudiado consiste en métodos de un solo paso, lo que significa que el paso más nuevo w_{i+1} se produce sobre la base de la ecuación diferencial y el valor de la w_i del paso anterior. Éste es el espíritu de los problemas de valor inicial, para los cuales el teorema 6.2 garantiza una solución única a partir de una w_0 arbitraria.

Los métodos de varios pasos sugieren un enfoque diferente: usar el conocimiento de más de una de las w_i anteriores para ayudar a producir el siguiente paso. Esto conduce a métodos para resolver EDO que tienen un orden tan alto como los métodos de un solo paso, pero gran parte del cálculo necesario se reemplazará con la interpolación de los valores ya calculados en el transcurso de la resolución.

6.7.1 Generación de métodos de varios pasos

Como primer ejemplo, se considerará el siguiente método de dos pasos:

Método de dos pasos de Adams-Bashforth

$$w_{i+1} = w_i + h\left[\frac{3}{2} f(t_i, w_i) - \frac{1}{2} f(t_{i-1}, w_{i-1})\right]. \tag{6.72}$$

Mientras que el método del punto medio de segundo orden,

$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2} f(t_i, w_i)\right),$$

necesita dos evaluaciones de función del lado derecho f de la EDO por paso, el método de dos pasos de Adams-Bashforth requiere solamente una nueva evaluación por paso (una evaluación se recupera del paso anterior). Se verá posteriormente que (6.72) también es un método de segundo orden. Por lo tanto, los métodos de varios pasos pueden lograr el mismo orden con menos esfuerzo de cálculo (por lo general, sólo una evaluación de función por paso).

Dado que los métodos de varios pasos usan más de un valor anterior w, necesitan algo de ayuda para empezar. La fase inicio (o de puesta en marcha) de un método de s pasos consiste por lo regular en un proceso que usa w_0 para generar $s - 1$ valores $w_1, w_2, \ldots, w_{s-1}$, antes de que el método de varios pasos pueda utilizarse. El método de Adams-Bashforth de dos pasos (6.72) necesita w_1, junto con la condición inicial dada w_0, para comenzar. El siguiente código de MATLAB utiliza el método del trapecio para proporcionar el valor de w_1 de la fase de inicio.

```
% Programa 6.7 Método de varios pasos
% Entradas: intervalo de tiempo inter,
%  ic = [y0] condición inicial, número de pasos n,
%  s= número de pasos (varios), por ejemplo 2 para el método de 2 pasos
% Salida: pasos de tiempo t, solución y
% Llama a un método de varios pasos, como ab2step.m
% Uso de ejemplo: [t, y]=exmultistep([0,1],1,20,2)
function [t,y]=exmultistep(inter,ic,n,s)
h=(inter(2)-inter(1))/n;
% Fase de inicio
y(1,:)=ic;t(1)=inter(1);
for i=1:s-1                % fase de inicio, usando método de un solo paso
  t(i+1)=t(i)+h;
  y(i+1,:)=trapstep(t(i),y(i,:),h);
  f(i,:)=ydot(t(i),y(i,:));
end
for i=s:n                  % ciclo del método de varios pasos
  t(i+1)=t(i)+h;
  f(i,:)=ydot(t(i),y(i,:));
  y(i+1,:)=ab2step(t(i),i,y,f,h);
end
plot(t,y)

function y=trapstep(t,x,h)
%un paso del método del trapecio de la sección 6.2
z1=ydot(t,x);
g=x+h*z1;
z2=ydot(t+h,g);
y=x+h*(z1+z2)/2;

function z=ab2step(t,i,y,f,h)
%un paso del método de Adams-Bashforth de 2 pasos
z=y(i,:)+h*(3*f(i,:)/2-f(i-1,:)/2);

function z=unstable2step(t,i,y,f,h)
%un paso de un método inestable de 2 pasos
z=-y(i,:)+2*y(i-1,:)+h*(5*f(i,:)/2+f(i-1,:)/2);

function z=weaklystable2step(t,i,y,f,h)
%un paso de un método débilmente estable de 2 pasos
z=y(i-1,:)+h*2*f(i,:);

function z=ydot(t,y)  % PVI de la sección 6.1
z=t*y+t^3;
```

En la figura 6.23(a) se muestra el resultado de la aplicación del método de Adams-Bashforth de dos pasos al problema de valor inicial (6.5) que se presentó con anterioridad en este capítulo, con tamaño de paso $h = 0.05$ y aplicando el método del trapecio para la fase de inicio. El inciso (b) de la figura muestra el uso de un método diferente de dos pasos. Su inestabilidad será el tema del análisis de la estabilidad en las siguientes secciones.

Un método general de s pasos tiene la forma

$$w_{i+1} = a_1 w_i + a_2 w_{i-1} + \cdots + a_s w_{i-s+1} + h[b_0 f_{i+1} + b_1 f_i + b_2 f_{i-1} + \cdots + b_s f_{i-s+1}]. \tag{6.73}$$

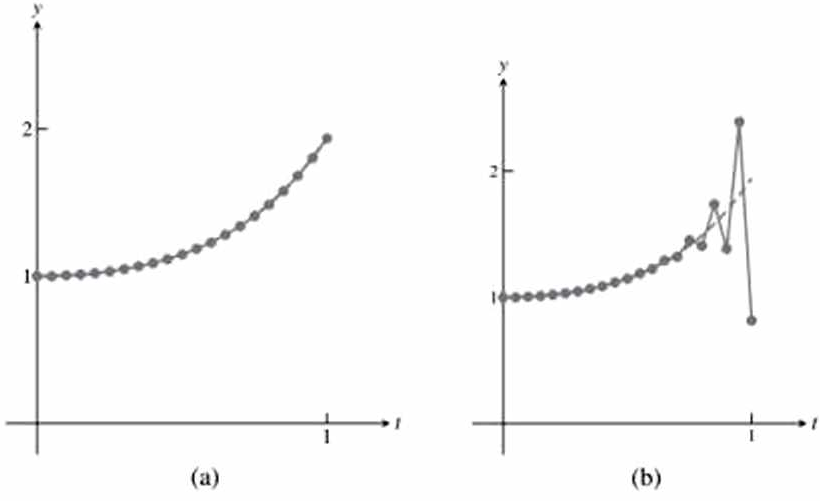


Figura 6.23 Métodos de dos pasos aplicados al PVI (6.5). La curva discontinua muestra la solución correcta. Tamaño del paso $h = 0.05$. (a) El método de Adams-Bashforth de dos pasos se representa con círculos. (b) El método inestable (6.81) se representa con círculos.

El tamaño del paso es h, y se usa la notación

$$f_i \equiv f(t_i, w_i).$$

Si $b_0 = 0$, el método es explícito. Si $b_0 \neq 0$, el método es implícito. En breve se estudiará la manera de utilizar los métodos implícitos.

En primer lugar, se quiere mostrar la manera de obtener los métodos de varios pasos y cómo decidir cuáles funcionan mejor. Los principales problemas que surgen con los métodos de pasos múltiples pueden introducirse mediante un caso relativamente simple de los métodos de dos pasos, por lo que se iniciará allí. Un método general de dos pasos ($s = 2$ en (6.73)) tiene la forma

$$w_{i+1} = a_1 w_i + a_2 w_{i-1} + h[b_0 f_{i+1} + b_1 f_i + b_2 f_{i-1}]. \tag{6.74}$$

Para desarrollar un método de varios pasos, es necesario hacer referencia al teorema de Taylor, puesto que el juego sigue siendo hacer coincidir el mayor número posible de términos de la expansión de Taylor para la solución con los términos del método. El residuo será el error de truncamiento local.

Se supone que toda w_i anterior es correcta —es decir, $w_i = y_i$ y $w_{i-1} = y_{i-1}$ en (6.74). La ecuación diferencial dice que $y_i' = f_i$, por lo que todos los términos puede expandirse en serie de Taylor de la siguiente manera:

$$\begin{aligned} w_{i+1} &= a_1 w_i + a_2 w_{i-1} + h[b_0 f_{i+1} + b_1 f_i + b_2 f_{i-1}] \\ &= a_1[y_i] \\ &\quad + a_2[y_i - hy_i' + \tfrac{h^2}{2}y_i'' - \tfrac{h^3}{6}y_i''' + \tfrac{h^4}{24}y_i''' - \cdots] \\ &\quad + b_0[hy_i' + h^2 y_i'' + \tfrac{h^3}{2}y_i''' + \tfrac{h^4}{6}y_i''' + \cdots] \\ &\quad + b_1[hy_i'] \\ &\quad + b_2[hy_i' - h^2 y_i'' + \tfrac{h^3}{2}y_i''' - \tfrac{h^4}{6}y_i''' + \cdots]. \end{aligned}$$

Al sumar se obtiene

$$\begin{aligned} w_{i+1} = (a_1 + a_2)y_i + (b_0 + b_1 + b_2 - a_2)hy_i' + (a_2 - 2b_2 + 2b_0)\frac{h^2}{2}y_i'' \\ + (-a_2 + 3b_0 + 3b_2)\frac{h^3}{6}y_i''' + (a_2 + 4b_0 - 4b_2)\frac{h^4}{24}y_i''' + \cdots. \end{aligned} \tag{6.75}$$

Si se eligen a_i y b_i apropiadamente, el error de truncamiento local $y_{i+1} - w_{i+1}$, donde

$$y_{i+1} = y_i + hy_i' + \frac{h^2}{2}y_i'' + \frac{h^3}{6}y_i''' + \cdots, \tag{6.76}$$

puede hacerse tan pequeño como sea posible, en el supuesto de que las derivadas involucradas realmente existan. A continuación se investigarán las posibilidades.

6.7.2 Métodos de varios pasos explícitos

Para buscar los métodos explícitos, establezca $b_0 = 0$. Un método de segundo orden puede desarrollarse agrupando los términos coincidentes en (6.75) y (6.76) incluyendo el término h^2, con lo que el error de truncamiento local tendrá un tamaño $O(h^3)$. Al comparar términos se obtiene el sistema

$$\begin{aligned} a_1 + a_2 &= 1 \\ -a_2 + b_1 + b_2 &= 1 \\ a_2 - 2b_2 &= 1. \end{aligned} \tag{6.77}$$

Existen tres ecuaciones con cuatro incógnitas a_1, a_2, b_1, b_2, por lo que será posible encontrar una infinidad de métodos explícitos de orden dos diferentes. (Una de las soluciones corresponde a un método de orden tres. Vea el ejercicio 3). Observe que las ecuaciones pueden escribirse en términos de a_1 como sigue:

$$\begin{aligned} a_2 &= 1 - a_1 \\ b_1 &= 2 - \frac{1}{2}a_1 \\ b_2 &= -\frac{1}{2}a_1. \end{aligned} \tag{6.78}$$

El error de truncamiento local será

$$\begin{aligned} y_{i+1} - w_{i+1} &= \frac{1}{6}h^3 y_i''' - \frac{3b_2 - a_2}{6}h^3 y_i''' + O(h^4) \\ &= \frac{1 - 3b_2 + a_2}{6}h^3 y_i''' + O(h^4) \\ &= \frac{4 + a_1}{12}h^3 y_i''' + O(h^4). \end{aligned} \tag{6.79}$$

Existe la libertad de fijar arbitrariamente a_1 (cualquier elección conduce a un método de segundo orden, como se acaba de mostrar). Si se fija $a_1 = 1$ resulta el método de segundo orden de Adams-Bashforth (6.72). Observe que $a_2 = 0$ por la primera ecuación, asimismo $b_2 = -1/2$ y $b_1 = 3/2$. De acuerdo con (6.79), el error de truncamiento local es $5/12h^3\, y'''(t_i) + O(h^4)$.

De manera alternativa, podría establecerse $a_1 = 1/2$ para obtener otro método de dos pasos de segundo orden con $a_2 = 1/2$, $b_1 = 7/4$ y $b_2 = -1/4$:

$$w_{i+1} = \frac{1}{2}w_i + \frac{1}{2}w_{i-1} + h\left[\frac{7}{4}f_i - \frac{1}{4}f_{i-1}\right]. \tag{6.80}$$

Este método tiene un error de truncamiento local de $3/8h^3\, y'''(t_i) + O(h^4)$.

ANOTACIÓN

Resumen La ventaja de los métodos de varios pasos sobre los métodos de un solo paso es clara. Después de unos pocos pasos, sólo se requiere realizar una nueva evaluación de la función del lado derecho. Para los métodos de un solo paso, resulta típico que se requieran varias evaluaciones de función. Por ejemplo, el método Runge-Kutta de cuarto orden requiere cuatro evaluaciones por paso, mientras que el método de cuarto orden de Adams-Bashforth requiere sólo una después de la fase de inicio.

Una tercera opción, $a_1 = -1$, da el método de segundo orden de dos pasos

$$w_{i+1} = -w_i + 2w_{i-1} + h\left[\frac{5}{2}f_i + \frac{1}{2}f_{i-1}\right] \tag{6.81}$$

que se utilizó en la figura 6.23(b). La falla de (6.81) pone de manifiesto una situación de estabilidad importante que deben cumplir los métodos de varios pasos. Considere el PVI aún más simple

$$\begin{cases} y' = 0 \\ y(0) = 0 \\ t \text{ en } [0,1] \end{cases}. \tag{6.82}$$

Al aplicar el método (6.81) a este ejemplo, se obtiene

$$w_{i+1} = -w_i + 2w_{i-1} + h[0]. \tag{6.83}$$

Una solución $\{w_i\}$ para (6.83) es $w_i \equiv 0$. Sin embargo, hay otras. Si se sustituye la forma $w_i = c\lambda^i$ en (6.83) resulta

$$\begin{aligned} c\lambda^{i+1} + c\lambda^i - 2c\lambda^{i-1} &= 0 \\ c\lambda^{i-1}(\lambda^2 + \lambda - 2) &= 0. \end{aligned} \tag{6.84}$$

Las raíces del "polinomio característico" $\lambda^2 + \lambda - 2 = 0$ son 1 y -2. El último es un problema (significa que las soluciones de la forma $(-2)^i c$ son soluciones del método para la constante c). Esto permite que los pequeños errores de redondeo y de truncamiento crezcan con rapidez y dificulten el cálculo, como se ve en la figura 6.23. Para evitar esta posibilidad, es importante que las raíces del polinomio característico del método estén delimitadas por 1 en valor absoluto. Lo anterior conduce a la siguiente definición:

DEFINICIÓN 6.6 El método de varios pasos (6.73) es **estable** si las raíces del polinomio $P(x) = x^s - a_1x^{s-1} - \ldots - a_s$, están delimitadas por 1 en valor absoluto, y las raíces en valor absoluto 1 son raíces simples. Un método estable para el que 1 sea la única raíz en valor absoluto 1 se llama **fuertemente estable**; en caso contrario, es **débilmente estable**.

El método de Adams-Bashforth (6.72) tiene raíces 0 y 1, por lo que es muy estable, mientras que (6.81) tiene raíces -2 y 1, por lo que es inestable.

El polinomio característico de la fórmula general de dos pasos, utiliza el hecho de que $a_1 = 1 - a_2$ de (6.78), es

$$\begin{aligned} P(x) &= x^2 - a_1x - a_2 \\ &= x^2 - a_1x - 1 + a_1 \\ &= (x-1)(x - a_1 + 1), \end{aligned}$$

cuyas raíces son 1 y $a_1 - 1$. De regreso a (6.78), puede encontrarse un método de segundo orden débilmente estable si se define $a_1 = 0$. En tal caso las raíces son 1 y -1, lo que conduce al siguiente método débilmente estable de segundo orden en dos pasos:

$$w_{i+1} = w_{i-1} + 2hf_i. \tag{6.85}$$

► EJEMPLO 6.26 Aplique el método fuertemente estable (6.72), el método débilmente estable (6.85), y el método inestable (6.81) al problema de valor inicial

$$\begin{cases} y' = -3y \\ y(0) = 1 \\ t \text{ en } [0,2] \end{cases}. \tag{6.86}$$

La solución es la curva $y = e^{-3t}$. Se usará el programa 6.7 para seguir las soluciones, donde `ydot.m` se ha cambiado por

```
function z=ydot(t,y)
z=-3*y;
```

y `ab2step` se sustituye por una de las tres llamadas `ab2step`, `weaklystable2step` o `unstable2step`.

En la figura 6.24 se muestran las tres aproximaciones a la solución para el tamaño de paso $h = 0.1$. Los métodos débilmente estables e inestables parecen seguir de cerca durante un tiempo y después se alejan con rapidez de la solución correcta. La reducción del tamaño de paso no elimina el problema, aunque puede retrasar la aparición de la inestabilidad. ◄

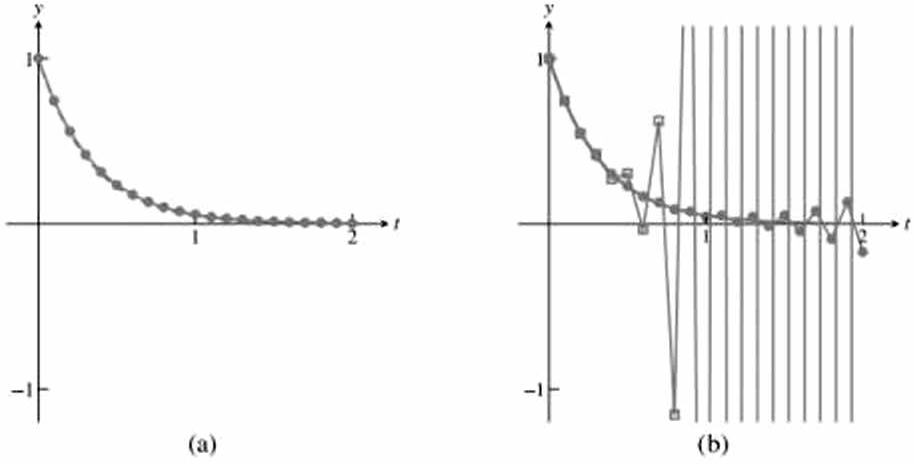


Figura 6.24 Comparación de métodos de segundo orden, de dos pasos, aplicados al PVI (6.86).
(a) Método de Adams-Bashforth. (b) Método débilmente estable (en círculos) y método inestable (en cuadros).

Con dos nuevas definiciones puede enunciarse el teorema fundamental de los métodos de varios pasos.

DEFINICIÓN 6.7 Un método de varios pasos es **consistente** si tiene por lo menos un orden de 1. Un método es **convergente** si las soluciones aproximadas convergen a la solución exacta para cada t, cuando $h \to 0$. ❐

TEOREMA 6.8 **(Dahlquist)** Suponga que los valores iniciales son correctos. Entonces, un método de varios pasos (6.73) es convergente si y sólo si es estable y consistente. ■

Para una demostración del teorema de Dahlquist, vea Hairer y Wanner [1996]. El teorema 6.8 indica que evitar una catástrofe como la de la figura 6.24(b) para un método de segundo orden de dos pasos es tan simple como comprobar la estabilidad del método.

Una de las raíces del polinomio característico debe estar en 1 (vea el ejercicio 6). Los métodos de Adams-Bashforth son aquellos cuyas otras raíces están en 0. Por esta razón, el método de Adams-Bashforth de dos pasos se considera el más estable de los métodos de dos pasos.

La obtención de métodos de orden superior, usando más pasos, es precisamente análoga a la derivación anterior de los métodos de dos pasos. Los ejercicios 13 y 14 piden verificar que los siguientes métodos sean muy estables:

Método de Adams-Bashforth de tres pasos (tercer orden)

$$w_{i+1} = w_i + \frac{h}{12}[23f_i - 16f_{i-1} + 5f_{i-2}]. \tag{6.87}$$

Método de Adams-Bashforth de cuatro pasos (cuarto orden)

$$w_{i+1} = w_i + \frac{h}{24}[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}]. \tag{6.88}$$

6.7.3 Métodos de varios pasos implícitos

Cuando el coeficiente b_0 en (6.73) es distinto de cero, el método es implícito. El método implícito más simple de segundo orden (vea el ejercicio 5) es el método del trapecio implícito:

Método del trapecio implícito (segundo orden)

$$w_{i+1} = w_i + \frac{h}{2}[f_{i+1} + f_i]. \tag{6.89}$$

Si el término f_{i+1} se sustituye al evaluar f en la "predicción" para w_{i+1} realizada mediante el método de Euler, entonces ésta se convierte en el método del trapecio explícito. El método del trapecio implícito también se conoce como el método de Adams-Moulton de un paso, por analogía con lo que sigue. Un ejemplo de un método implícito de dos pasos es el método de Adams-Moulton de dos pasos:

Método de Adams-Moulton de dos pasos (tercer orden)

$$w_{i+1} = w_i + \frac{h}{12}[5f_{i+1} + 8f_i - f_{i-1}]. \tag{6.90}$$

Existen diferencias significativas entre los métodos implícitos y explícitos. En primer lugar, es posible obtener un método implícito estable de tercer orden usando sólo dos pasos anteriores, a diferencia del caso explícito. En segundo lugar, la fórmula correspondiente del error local de truncamiento es menor para los métodos implícitos. Por otra parte, el método implícito tiene la dificultad inherente de que requiere un procesamiento adicional para evaluar la parte implícita.

Por estas razones, los métodos implícitos se utilizan a menudo como correctores en un par "predictor-corrector"; es decir, métodos implícitos y explícitos del mismo orden que se usan juntos. Cada paso es la combinación de una predicción por el método explícito y una corrección por el método implícito, donde el método implícito utiliza la w_{i+1} predicha para calcular f_{i+1}. Los métodos predictor-corrector implican el doble del esfuerzo de cálculo, puesto que tanto en la parte de predicción como en la parte de corrección del paso, se realiza una evaluación del lado derecho f de la ecuación diferencial. Sin embargo, la precisión y estabilidad obtenidas suelen hacer que valga la pena hacerlo.

Un método simple de predictor-corrector empareja el método explícito de Adams-Bashforth de dos pasos como predictor con el método implícito de Adams-Moulton de un paso como corrector. Ambos son métodos de segundo orden. El código de MATLAB es similar al código de Adams-Bashforth utilizado con anterioridad, pero se agrega una etapa de corrección:

```
% Programa 6.8 Adams-Bashforth-Moulton de segundo orden p-c
% Entradas: intervalo de tiempo inter,
%  ic=[y0] condición inicial
%  número de pasos n, número de pasos (varios) s para método explícito
% Salida: pasos de tiempo t, solución y
% Llama métodos de varios pasos como ab2step.m y am1step.m
% Uso de ejemplo: [t,y]=predcorr([0 1],1,20,2)
function [t,y]=predcorr(inter,ic,n,s)
h=(inter(2)-inter(1))/n;
% Fase de inicio
y(1,:)=ic;t(1)=inter(1);
for i=1:s-1                  % fase inicio, usando método de un paso
  t(i+1)=t(i)+h;
  y(i+1,:)=trapstep(t(i),y(i,:),h);
  f(i,:)=ydot(t(i),y(i,:));
end
for i=s:n                    % ciclo del método de varios pasos
  t(i+1)=t(i)+h;
```

```
  f(i,:)=ydot(t(i),y(i,:));
  y(i+1,:)=ab2step(t(i),i,y,f,h);  % predice
  f(i+1,:)=ydot(t(i+1),y(i+1,:));
  y(i+1,:)=am1step(t(i),i,y,f,h);  % corrige
end
plot(t,y)

function y=trapstep(t,x,h)
%un paso del método del trapecio de la sección 6.2
z1=ydot(t,x);
g=x+h*z1;
z2=ydot(t+h,g);
y=x+h*(z1+z2)/2;

function z=ab2step(t,i,y,f,h)
%un paso del método de Adams-Bashforth de 2 pasos
z=y(i,:)+h*(3*f(i,:)-f(i-1,:))/2;

function z=am1step(t,i,y,f,h)
%un paso del método de Adams-Moulton de 1 paso
z=y(i,:)+h*(f(i+1,:)+f(i,:))/2;

function z=ydot(t,y)  % IVP
z=t*y+t^3;
```

El método de Adams-Moulton de dos pasos se obtiene de la misma forma que los métodos explícitos. Vuelva a hacer el conjunto de ecuaciones (6.77), pero sin requerir que $b_0 = 0$. Puesto que hay un parámetro adicional ahora (b_0), es posible hacer coincidir (6.75) y (6.76) a través de los términos de tercer grado con sólo un método de dos pasos, colocando el error local de truncamiento en el término h^4. El análogo a (6.77) es

$$\begin{aligned} a_1 + a_2 &= 1 \\ -a_2 + b_0 + b_1 + b_2 &= 1 \\ a_2 + 2b_0 - 2b_2 &= 1 \\ -a_2 + 3b_0 + 3b_2 &= 1. \end{aligned} \tag{6.91}$$

Al satisfacer estas ecuaciones se obtiene un método implícito de tercer orden y dos pasos.

Las ecuaciones pueden escribirse en términos de a_1 como sigue:

$$\begin{aligned} a_2 &= 1 - a_1 \\ b_0 &= \frac{1}{3} + \frac{1}{12}a_1 \\ b_1 &= \frac{4}{3} - \frac{2}{3}a_1 \\ b_2 &= \frac{1}{3} - \frac{5}{12}a_1. \end{aligned} \tag{6.92}$$

El error de truncamiento local es

$$\begin{aligned} y_{i+1} - w_{i+1} &= \frac{1}{24}h^4 y_i^{iv} - \frac{4b_0 - 4b_2 + a_2}{24}h^4 y_i^{iv} + O(h^5) \\ &= \frac{1 - a_2 - 4b_0 + 4b_2}{24}h^4 y_i^{iv} + O(h^5) \\ &= -\frac{a_1}{24}h^4 y_i^{iv} + O(h^5). \end{aligned}$$

El orden del método será tres, siempre que $a_1 \neq 0$. Como a_1 es un parámetro libre, hay un número infinito métodos implícitos de tercer orden y dos pasos. El método de Adams-Moulton de dos pasos

utiliza la opción $a_1 = 1$. El ejercicio 8 pide una verificación de que este método es fuertemente estable. En el ejercicio 9 se exploran otras opciones de a_1.

Tenga en cuenta una opción más especial, $a_1 = 0$. A partir de la fórmula del truncamiento local, se observa que este método de dos pasos será de cuarto orden.

Método de Milne-Simpson

$$w_{i+1} = w_{i-1} + \frac{h}{3}[f_{i+1} + 4f_i + f_{i-1}]. \tag{6.93}$$

En el ejercicio 10 se le pide que compruebe que éste sólo sea débilmente estable. Por esta razón, es susceptible a la magnificación del error.

La sugestiva terminología del método implícito del trapecio (6.89) y del método de Milne-Simpson (6.93) debe recordarle al lector las fórmulas de integración numérica del capítulo 5. De hecho, aunque no se ha hecho hincapié en este enfoque, muchas de las fórmulas de varios pasos que se han presentado aquí pueden obtenerse de manera alternativa mediante la integración de polinomios que se aproximan, en una estrecha analogía con los esquemas de integración numérica.

La idea básica detrás de este enfoque es que la ecuación diferencial $y' = f(t, y)$ puede integrarse en el intervalo $[t_i, t_{i+1}]$ para dar

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y)\, dt. \tag{6.94}$$

La aplicación de un esquema de integración numérica para aproximar la integral en (6.94) da como resultado un método de EDO con varios pasos. Por ejemplo, si se usa la regla del trapecio para la integración numérica del capítulo 5, se obtiene

$$y(t_{i+1}) - y(t_i) = \frac{h}{2}(f_{i+1} + f_i) + O(h^2),$$

que es el método del trapecio de segundo orden para ecuaciones diferenciales ordinarias. Si se aproxima la integral mediante la regla de Simpson, el resultado es

$$y(t_{i+1}) - y(t_i) = \frac{h}{3}(f_{i+1} + 4f_i + f_{i-1}) + O(h^4),$$

que es el método Milne-Simpson de cuarto orden (6.93). En esencia, la parte derecha de la EDO se aproxima mediante un polinomio y se integra, tal como se hizo en la integración numérica. Este enfoque puede extenderse para recuperar algunos de los métodos de varios pasos que ya se han presentado, cambiando el grado de interpolación y la ubicación de los puntos de interpolación. Aunque este enfoque es una forma geométrica adicional para obtener algunos métodos de varios pasos, no proporciona ninguna visión particular de la estabilidad de los solucionadores de EDO que resultan.

Al extender los métodos anteriores, pueden obtenerse los métodos de Adams-Moulton de orden superior, en cada caso utilizando $a_1 = 1$:

Método de Adams-Moulton de tres pasos (cuarto orden)

$$w_{i+1} = w_i + \frac{h}{24}[9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}]. \tag{6.95}$$

Método de Adams-Moulton de cuatro pasos (quinto orden)

$$w_{i+1} = w_i + \frac{h}{720}[251f_{i+1} + 646f_i - 264f_{i-1} + 106f_{i-2} - 19f_{i-3}]. \tag{6.96}$$

Éstos son muy utilizados en los métodos de predictor-corrector, con un predictor de Adams-Bashforth del mismo orden. Los problemas de computadora 9 y 10 le piden el código de MATLAB para implementar esta idea.

6.7 Ejercicios

1. Aplique el método de Adams-Bashforth de dos pasos a los PVI

   (a) $y' = t$ (b) $y' = t^2 y$ (c) $y' = 2(t+1)y$

   (d) $y' = 5t^4 y$ (e) $y' = 1/y^2$ (f) $y' = t^3/y^2$

   con la condición inicial $y(0) = 1$. Utilice el tamaño de paso $h = 1/4$ en el intervalo $[0, 1]$. Utilice el método del trapecio explícito para crear w_1. Usando la solución correcta del ejercicio 3 de la sección 6.1, encuentre el error de truncamiento global en $t = 1$.

2. Realice los pasos del ejercicio 1 sobre los PVI

   (a) $y' = t + y$ (b) $y' = t - y$ (c) $y' = 4t - 2y$

   con la condición inicial $y(0) = 0$. Use la solución correcta del ejercicio 4 de la sección 6.1 para encontrar el error de truncamiento total en $t = 1$.

3. Encuentre un método explícito de tercer orden y dos pasos. ¿El método es estable?

4. Encuentre un método explícito de segundo orden y dos pasos, cuyo polinomio característico tenga una doble raíz en 1.

5. Demuestre que el método del trapecio implícito (6.89) es un método de segundo orden.

6. Explique por qué el polinomio característico de un método explícito o implícito de s pasos, para $s \geq 2$, debe tener una raíz en 1.

7. (a) ¿Para qué a_1 existe un método explícito de segundo orden y dos pasos fuertemente estable? (b) Responda la misma pregunta para un método débilmente estable.

8. Demuestre que los coeficientes del método implícito de Adams-Moulton de dos pasos satisfacen (6.92) y que el método es fuertemente estable.

9. Encuentre el orden y el tipo de estabilidad de los siguientes métodos implícitos de dos pasos:

   (a) $w_{i+1} = 3w_i - 2w_{i-1} + \frac{h}{12}[13f_{i+1} - 20f_i - 5f_{i-1}]$

   (b) $w_{i+1} = \frac{4}{3}w_i - \frac{1}{3}w_{i-1} + \frac{2}{3}hf_{i+1}$

   (c) $w_{i+1} = \frac{4}{3}w_i - \frac{1}{3}w_{i-1} + \frac{h}{9}[4f_{i+1} + 4f_i - 2f_{i-1}]$

   (d) $w_{i+1} = 3w_i - 2w_{i-1} + \frac{h}{12}[7f_{i+1} - 8f_i - 11f_{i-1}]$

   (e) $w_{i+1} = 2w_i - w_{i-1} + \frac{h}{2}[f_{i+1} - f_{i-1}]$

10. Deduzca el método de Milne-Simpson (6.93) a partir de (6.92), y demuestre que es de cuarto orden y débilmente estable.

11. Encuentre un método implícito de segundo orden y dos pasos que sea débilmente estable.

12. El método de Milne-Simpson es un método implícito de cuarto orden y dos pasos, débilmente estable. ¿Hay algún método implícito de tercer orden y dos pasos que sea débilmente estable?

13. (a) Encuentre las condiciones (análogas a (6.77)) en a_i, b_i necesarias para un método explícito de tercer orden y tres pasos. (b) Demuestre que el método de Adams-Bashforth de tres pasos satisface las condiciones. (c) Demuestre que el método de Adams-Bashforth de tres pasos es fuertemente

estable. (d) Encuentre un método explícito de tercer orden y tres pasos que sea débilmente estable, y verifique estas propiedades.

14. (a) Encuentre las condiciones (análogas a (6.77)) en a_i, b_i necesarias para un método explícito de cuarto orden y cuatro pasos. (b) Demuestre que el método de Adams-Bashforth de cuatro pasos satisface las condiciones. (c) Demuestre que el método de Adams-Bashforth de cuatro pasos es fuertemente estable.

15. (a) Encuentre las condiciones (análogas a (6.77)) en a_i, b_i necesarias para un método implícito de cuarto orden y tres pasos. (b) Demuestre que el método de Adams-Moulton de tres pasos satisface las condiciones. (c) Demuestre que el método de Adams-Moulton de tres pasos es fuertemente estable.

6.7 Problemas de computadora

1. Adapte el programa `exmultistep.m` para aplicar el método de Adams-Bashforth de dos pasos a los PVI del ejercicio 1. Usando el tamaño de paso $h = 0.1$, calcule la aproximación en el intervalo [0, 1]. Imprima una tabla con los valores de t, las aproximaciones y el error de truncamiento global en cada iteración.

2. Adapte el programa `exmultistep.m` para aplicar el método de Adams-Bashforth de dos pasos a los PVI del ejercicio 2. Usando el tamaño de paso $h = 0.1$, calcule la aproximación en el intervalo [0, 1]. Imprima una tabla con los valores de t, las aproximaciones y los errores de truncamiento global en cada iteración.

3. Realice los pasos del problema de computadora 2, usando el método inestable de dos pasos (6.81).

4. Realice los pasos del problema de computadora 2, usando el método de Adams-Bashforth de tres pasos. Use Runge-Kutta de cuarto orden para calcular w_1 y w_2.

5. Grafique la solución aproximada del método de Adams-Bashforth de tres pasos en [0, 1] para la ecuación diferencial $y' = 1 + y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 1$, junto con la solución exacta (vea el ejercicio 7 de la sección 6.1). Utilice tamaños de paso $h = 0.1$ y 0.05.

6. Grafique la solución aproximada del método de Adams-Bashforth de tres pasos en [0, 1] para la ecuación diferencial $y' = 1 - y^2$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = -1/2$, junto con la solución exacta (vea el ejercicio 8 de la sección 6.1). Utilice tamaños de paso $h = 0.1$ y 0.05.

7. Calcule la solución aproximada del método de Adams-Bashforth de tres pasos en [0, 4] para la ecuación diferencial $y' = \text{sen } y$ y la condición inicial (a) $y_0 = 0$ (b) $y_0 = 100$, con tamaños de paso $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$, junto con la solución exacta (vea el ejercicio 15 de la sección 6.1); asimismo, haga una gráfica log-log del error como una función de h.

8. Calcule la solución aproximada del método de Adams-Bashforth de tres pasos para la ecuación diferencial $y' = \text{senh } y$ y la condición inicial (a) $y_0 = 1/4$ en el intervalo [0, 2] (b) $y_0 = 2$ en el intervalo [0, 1/4], con tamaños de paso $h = 0.1 \times 2^{-k}$ para $0 \leq k \leq 5$. Grafique las soluciones aproximadas $k = 0$ y $k = 5$, junto con la solución exacta (vea el ejercicio 16 de la sección 6.1); asimismo, haga una gráfica log-log del error en función de h.

9. Convierta el programa 6.8 en un método predictor-corrector de tercer orden, utilizando el método de Adams-Bashforth de tres pasos y el método de Adams-Moulton de dos pasos con tamaño de paso 0.05. Grafique la aproximación y la solución correcta del PVI (6.5) en el intervalo [0, 5].

10. Convierta el programa 6.8 en un método predictor-corrector de tercer orden, utilizando el método de Adams-Bashforth de cuatro pasos y el método de Adams-Moulton de tres pasos con tamaño de paso 0.05. Grafique la aproximación y la solución correcta del PVI (6.5) en el intervalo [0, 5].

Software y lecturas adicionales

Las fuentes tradicionales sobre ecuaciones diferenciales ordinarias son Blanchard *et al.* [2002], Boyce y DiPrima [2008], Braun [1993], Edwards y Penny [2004] y Kostelich y Armbruster [1997]. Hay muchos libros que enseñan los conceptos básicos de las EDO con amplia ayuda de cálculo y gráfica; se puede mencionar a *ODE Architect* [1999] como un buen ejemplo. Los códigos de MATLAB en Polking [1999] son una excelente manera de aprender y visualizar los conceptos de las EDO.

A fin de complementar el viaje realizado en este libro por los métodos numéricos de uno y varios pasos para la resolución de ecuaciones diferenciales ordinarias, pueden recomendarse muchos textos intermedios y avanzados. Henrici [1962] y Gear [1971] son clásicos. Shampine *et al.* [2003] abordan un enfoque contemporáneo con MATLAB. Otros textos recomendables son Iserles [1996], Shampine [1994], Ascher y Petzold [1998], Lambert [1991], Dormand [1996], Butcher [1987] y un amplio tratado de dos volúmenes de Hairer *et al.* [1993] y Hairer y Wanner [1996].

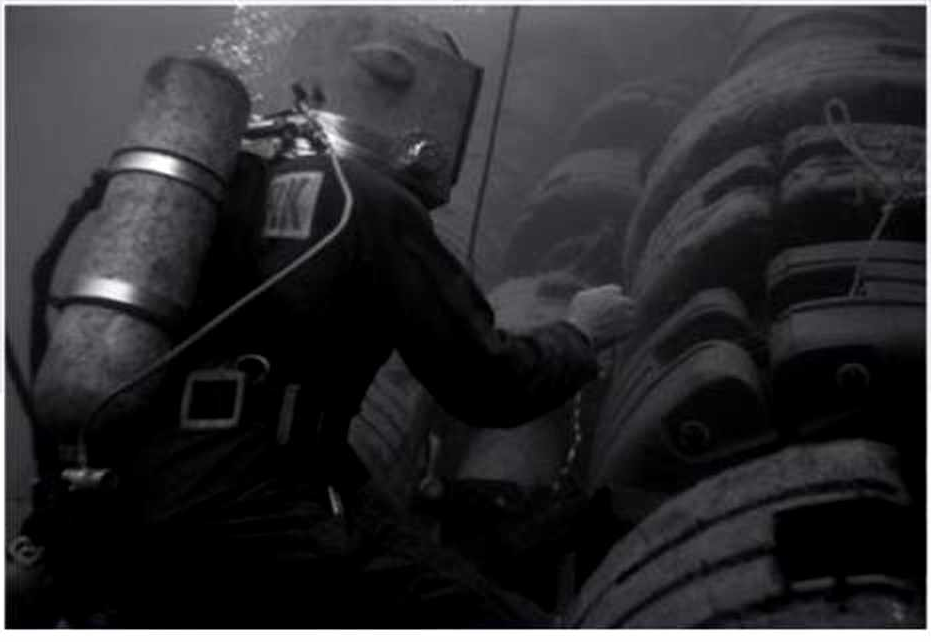
Existe una gran cantidad de software sofisticado disponible para la resolución de ecuaciones diferenciales ordinarias. Los detalles sobre los métodos utilizados en MATLAB pueden encontrarse en Shampine y Reichelt [1997], y Ashino *et al.* [2000]. Los métodos explícitos con tamaño de paso variable del tipo Runge-Kutta suelen tener éxito con los problemas no rígidos o ligeramente rígidos. Además de Runge-Kutta-Fehlberg y Dormand-Prince, con frecuencia se usa la variante Runge-Kutta-Verner, un método de quinto/sexto orden. Para los problemas rígidos, se requieren los métodos de diferencia hacia atrás y los métodos de extrapolación.

El IMSL incluye la rutina de doble precisión DIVPRK, basada en el método de Runge-Kutta-Verner, y DIVPAG para un método del tipo Adams con varios pasos, que puede manejar los problemas rígidos. La biblioteca NAG ofrece una rutina conductora D02BJF que ejecuta los pasos estándar de Runge-Kutta. El conductor de varios pasos D02CJF incluye programas de control del error del tipo Adams. Para los problemas rígidos, se recomienda la rutina D02EJF donde el usuario tiene la opción de especificar el jacobiano de cálculo más rápido.

El compendio Netlib contiene una rutina RKF45 en Fortran para el método de Runge-Kutta-Fehlberg y DVERK para el método de Runge-Kutta-Verner. El paquete para EDO de Netlib contiene algunas rutinas de varios pasos. La rutina VODE maneja los problemas rígidos.

La colección ODEPACK es un paquete de dominio público en código Fortran que implementa métodos de EDO. Fue desarrollada en el Lawrence Livermore National Laboratory (LLNL). El método básico LSODE y sus variantes resultan adecuados para los problemas rígidos y no rígidos. Las rutinas están disponibles de manera gratuita en el sitio web de LLNL `http://www.llnl.gov/CASC/odepack`.

CAPÍTULO



7 Problemas de valor de frontera

Las tuberías subterráneas y submarinas deben diseñarse para soportar la presión del ambiente exterior. Cuanto más profunda sea la tubería, más cotosa será una falla debida a un colapso. Los oleoductos que conectan las plataformas del mar del Norte con la costa se encuentran a una profundidad de 70 metros. La creciente importancia del gas natural, y el peligro y el costo de su transporte por vía marítima, pueden dar lugar a la construcción de gasoductos intercontinentales. Las profundidades del Atlántico Medio exceden los 5 kilómetros, donde la presión hidrostática de 7000 psi requerirá la innovación en los materiales y los procesos de construcción de las tuberías para evitar su deformación.

La teoría de la deformación de tuberías es crucial en una gran variedad de aplicaciones, desde los soportes arquitectónicos hasta los stents coronarios. Los modelos numéricos sobre deformaciones resultan valiosos cuando la experimentación directa es costosa y difícil.

Comprobación en la realidad En la página 355 se representa una parte de la sección transversal de una tubería en forma de anillo circular y examina cómo y cuándo se produce la deformación.

En el capítulo 6 se describieron los métodos de cálculo en la solución a un problema de valor inicial (PVI), una ecuación diferencial junto con los datos iniciales, que se especifican en el extremo izquierdo del intervalo de solución. Todos los métodos propuestos se basaban en técnicas de "paso a paso" (la solución aproximada comenzaba en el extremo izquierdo y avanzaba hacia delante con la variable independiente t). Cuando una ecuación diferencial se presenta junto con los datos de frontera, especificados en ambos extremos del intervalo de solución, surge un conjunto de problemas igualmente importante.

El capítulo 7 describe los métodos para aproximar las soluciones a un problema de valores en la frontera (PVF). Los métodos pueden ser de tres tipos. En primer lugar, se presentan los métodos de disparo, una combinación de los métodos de PVI del capítulo 6 y los métodos de ecuaciones del capítulo 1. Después, se exploran los métodos de diferencias finitas, que convierten la ecuación diferencial y las condiciones de frontera en un sistema de ecuaciones lineales o no lineales que debe resolverse. La sección final se centra en los métodos de colocación y el método del elemento finito, que resuelve el problema al expresar la solución en términos de funciones elementales.

7.1 MÉTODO DE DISPARO

Este primer método convierte el problema de valores en la frontera en un problema de valor inicial mediante la determinación de los valores iniciales faltantes que son consistentes con los valores de frontera. Los métodos que se desarrollaron en los capítulos 1 y 6 pueden combinarse para formar este método.

7.1.1 Soluciones a problemas de valor de frontera

Un problema general de valores en la frontera de segundo orden pide una solución de

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b \end{cases} \tag{7.1}$$

en el intervalo $a \leq t \leq b$, como se muestra en la figura 7.1. En el capítulo 6 se vio que una ecuación diferencial típica tiene una infinidad de soluciones y que se requieren datos adicionales para determinar una solución particular. En (7.1), la ecuación es de segundo orden, y se necesitan dos restricciones adicionales. Se dan como condiciones de frontera para la solución de $y(t)$ en a y b.

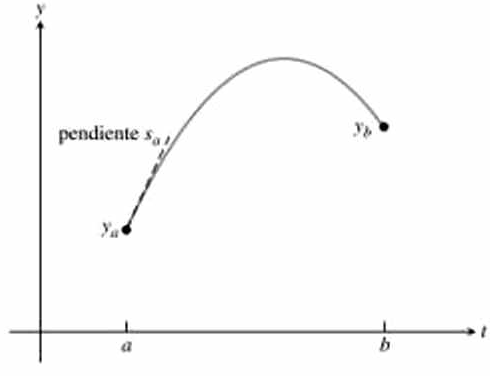


Figura 7.1 Comparación de los PVI y los PVF. En un problema de valor inicial, el valor inicial $y_a = y(a)$ y la pendiente inicial $s_a = y'(a)$ se especifican como parte del problema. En cambio, en un problema de valor de frontera, se especifican los valores de frontera y_a y y_b, con s_a como incógnita.

Consideremos un proyectil que satisface la ecuación diferencial de segundo orden $y''(t) = -g$ a medida que se mueve, donde y es la altura proyectil y g es la aceleración de la gravedad. La especificación de la posición y la velocidad iniciales sólo determinan el movimiento del proyectil como un problema de valor inicial. Por otra parte, podrían especificarse un intervalo de tiempo $[a, b]$ y las posiciones $y(a)$ y $y(b)$. Este último problema, un problema de valores en la frontera, también tiene una solución única en este caso.

► **EJEMPLO 7.1** Encuentre la altura máxima de un proyectil que se lanza desde la parte superior de un edificio con 30 metros de altura y llega al suelo 4 segundos más tarde.

La ecuación diferencial se deduce de la segunda ley de Newton $F = ma$, donde la fuerza de la gravedad es $F = -mg$ y $g = 9.81$ m/s^2. Sea $y(t)$ la altura en el momento t. La trayectoria puede expresarse como la solución del PVI

$$\begin{cases} y'' = -g \\ y(0) = 30 \\ y'(0) = v_0 \end{cases}$$

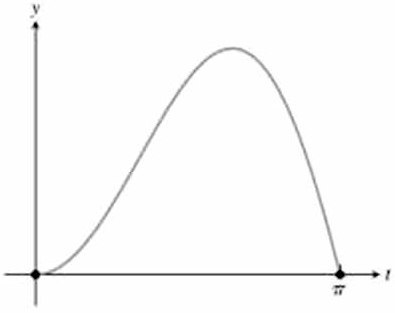


Figura 7.2 Solución del PVF (7.2). Gráfica de la solución $y(t) = t \operatorname{sen} t$ junto con los valores de frontera $y(0) = 0$ y $y(\pi) = 0$.

o el PVF

$$\begin{cases} y'' = -g \\ y(0) = 30 \\ y(4) = 0 \end{cases}$$

Dado que no se conoce la velocidad inicial v_0, debe resolverse el problema de valor de frontera. Al integrar dos veces resulta

$$y(t) = -\frac{1}{2}gt^2 + v_0 t + y_0.$$

Usando las condiciones de frontera se obtiene

$$\begin{aligned} 30 = y(0) &= y_0 \\ 0 = y(4) &= -\frac{16}{2}g + 4v_0 + 30, \end{aligned}$$

lo que implica que $v_0 \approx 12.12$ m/s. La trayectoria es $y(t) = -\frac{1}{2}gt^2 + 12.12t + 30$. Ahora es fácil usar el cálculo para encontrar el máximo desplazamiento horizontal del proyectil, que es aproximadamente 37.5 m. ◄

► EJEMPLO 7.2 Demuestre que $y(t) = t \operatorname{sen} t$ es una solución al problema de valores en la frontera

$$\begin{cases} y'' = -y + 2\cos t \\ y(0) = 0 \\ y(\pi) = 0 \end{cases} \tag{7.2}$$

En la figura 7.2 se muestra la función $y(t) = t \operatorname{sen} t$. Esta función resuelve la ecuación diferencial porque

$$y''(t) = -t \operatorname{sen} t + 2\cos t.$$

Al verificar las condiciones de frontera se obtiene $y(0) = 0 \operatorname{sen} 0 = 0$ y $y(\pi) = \pi \operatorname{sen} \pi = 0$. ◄

La teoría de existencia y unicidad de los problemas de valores en la frontera es más complicada que la correspondiente para los problemas de valor inicial. Algunos PVF aparentemente razonables pueden no tener solución o tener un número infinito de soluciones, situación que casi no ocurre con los PVI.

La situación de existencia y unicidad es análoga al arco de una bala de cañón humana que actúa bajo la gravedad terrestre. Suponga que el cañón tiene una velocidad de salida fija, pero que el ángulo del cañón puede variarse. Cualesquiera valores para la posición y la velocidad ori-

ginales determinan una trayectoria debida a la gravedad de la Tierra. Una solución al problema de valor inicial siempre existe y siempre es única. El problema de valores en la frontera tiene propiedades diferentes. Si la red para recibir al ejecutante se fija más allá del rango del cañón, no puede existir ninguna solución. Además, para cualquier condición de frontera dentro del alcance del cañón, hay dos soluciones: un viaje corto (con el ángulo de disparo del cañón inferior a 45°) y un viaje más largo (con un ángulo mayor a 45°), con lo que se viola la unicidad. Los dos ejemplos siguientes muestran las posibles soluciones para una ecuación diferencial muy sencilla.

► **EJEMPLO 7.3** Demuestre que el problema de valores en la frontera

$$\begin{cases} y'' = -y \\ y(0) = 0 \\ y(\pi) = 1 \end{cases}$$

no tiene soluciones.

La ecuación diferencial tiene una familia de soluciones, generadas por las soluciones linealmente independientes de cos t y sen t. Todas las soluciones de la ecuación deben tener la forma $y(t) = a\cos t + b \operatorname{sen} t$. Al sustituir la primera condición de frontera, $0 = y(0) = a$ implica que $a = 0$ y $y(t) = b \operatorname{sen} t$. La segunda condición de frontera $1 = y(\pi) = b \operatorname{sen} \pi = 0$ da una contradicción. No hay solución, y la existencia falla.

► **EJEMPLO 7.4** Demuestre que el problema de valores en la frontera

$$\begin{cases} y'' = -y \\ y(0) = 0 \\ y(\pi) = 0 \end{cases}$$

tiene un número infinito de soluciones.

Verifique que $y(t) = k \operatorname{sen} t$ es una solución de la ecuación diferencial y satisface las condiciones de frontera, para cada número real k. En particular, no hay unicidad de las soluciones para este ejemplo.

► **EJEMPLO 7.5** Encuentre todas las soluciones del problema de valores en la frontera

$$\begin{cases} y'' = 4y \\ y(0) = 1 \\ y(1) = 3 \end{cases} \tag{7.3}$$

Este ejemplo es suficientemente simple como para resolverse de manera exacta, pero lo suficientemente interesante para servir de ejemplo de los métodos de solución de PVF que se presentarán enseguida. Es posible imaginar dos soluciones de la ecuación diferencial, $y = e^{2t}$ y $y = e^{-2t}$. Como las soluciones no son múltiplos entre sí, son linealmente independientes; por lo tanto, a partir de la teoría elemental de las ecuaciones diferenciales, todas las soluciones de la ecuación diferencial son combinaciones lineales de $c_1e^{2t} + c_2e^{-2t}$. Las dos constantes c_1 y c_2 se determinan al aplicar las condiciones de frontera

$$1 = y(0) = c_1 + c_2$$

y

$$3 = y(1) = c_1e^2 + c_2e^{-2}.$$

Resolviendo se obtiene que:

$$y(t) = \frac{3 - e^{-2}}{e^2 - e^{-2}}e^{2t} + \frac{e^2 - 3}{e^2 - e^{-2}}e^{-2t}. \tag{7.4}$$

◄

7.1.2 Implementación del método de disparo

El método de disparo resuelve el PVF (7.1) al encontrar el PVI que tiene la misma solución. Se produce una secuencia de PVI que converge al correcto. La secuencia comienza con un valor inicial s_a de la pendiente, que se proporciona para acompañar al valor inicial y_a. El PVI resultante de esta pendiente inicial se resuelve y se compara con el valor de frontera y_b. La pendiente inicial se mejora mediante el ensayo y error hasta alcanzar la coincidencia con el valor de frontera. Para establecer una estructura más formal sobre este método, defina la siguiente función:

$$F(s) = \begin{cases} \text{diferencia entre } y_b \text{ y} \\ y(b), \text{ donde } y(t) \text{ es la} \\ \text{solución del PVI con} \\ y(a) = y_a \text{ y } y'(a) = s. \end{cases}$$

Con esta definición, el problema de valor de frontera se reduce al resolver la ecuación

$$F(s) = 0, \tag{7.5}$$

como se muestra en la figura 7.3.

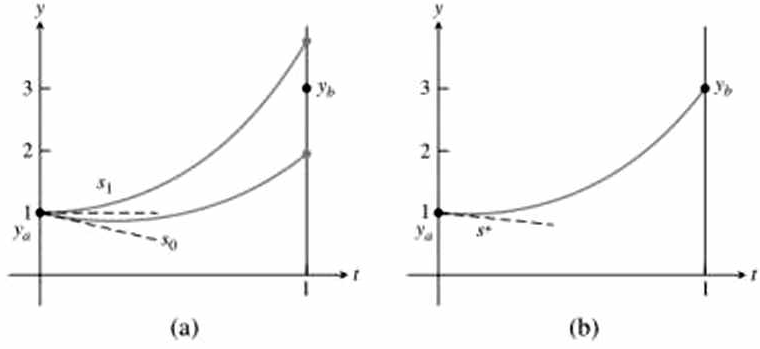


Figura 7.3 El método de disparo. (a) Para resolver el PVF, se resuelve el PVI con condiciones iniciales $y(a) = y_a$, $y'(a) = s_0$ con la estimación inicial s_0. El valor de $F(s_0)$ es $y(b) - y_b$. Después se elige una nueva s_1 y el proceso se repite con la meta de resolver $F(s) = 0$ para s. (b) Se usa el comando ode45 de MATLAB con raíz s^* para graficar la solución del PVF (7.7).

Ahora puede usarse un método de resolución de ecuaciones del capítulo 1 para resolver la ecuación. Puede elegirse el método de bisección o un método más sofisticado como el método de Brent. Deben encontrarse dos valores de s, llamados s_0 y s_1, para los cuales $F(s_0)F(s_1) < 0$. Entonces s_0 y s_1 confinan una raíz de (7.5), y es posible localizar una raíz s^* dentro de la tolerancia requerida por el solucionador de ecuaciones elegido. Por último, la solución al PVF (7.1) puede rastrearse (por ejemplo, mediante un método de PVI del capítulo) como la solución al problema de valor inicial

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y'(a) = s^* \end{cases}. \tag{7.6}$$

En el siguiente ejemplo se muestra una implementación en MATLAB del método de disparo.

► **EJEMPLO 7.6** Aplique el método de disparo al problema de valor de frontera

$$\begin{cases} y'' = 4y \\ y(0) = 1. \\ y(1) = 3 \end{cases} \tag{7.7}$$

Escriba la ecuación diferencial como un sistema de primer orden con el fin de utilizar el solucionador de PVI `ode45` de MATLAB:

$$\begin{aligned} y' &= v \\ v' &= 4y. \end{aligned} \tag{7.8}$$

Escriba un archivo de la función `F.m` que represente la función en (7.5):

```
function z=F(s)
a=0;b=1;yb=3;
ydot=@(t,y) [y(2);4*y(1)];
[t,y]=ode45(ydot,[a,b],[1,s]);
z=y(end,1)-yb; % end significa la última entrada de la solución y
```

Calcule $F(-1) \approx -1.05$ y $F(0) \approx 0.76$, como puede verse en la figura 7.3(a). Por lo tanto, hay una raíz de F entre -1 y 0. Aplique el método `bisect.m` del capítulo 1 o el comando `fzero` de MATLAB con intervalo de inicio $[-1, 0]$ para encontrar a s dentro de la precisión deseada. Por ejemplo,

```
>> sstar=fzero(@F,[-1,0])
```

Regresa aproximadamente -0.4203. (Recuerde que `fzero` requiere como entrada el inicio de la función `F`, que es `@F`). Entonces, la solución puede representarse como la solución de un problema de valor inicial (vea la figura 7.3(b)). La solución exacta de (7.7) se da en (7.4) y $s^* = y'(0) \approx -0.4203$. ◄

Para los sistemas de ecuaciones diferenciales ordinarias, los problemas de valores en la frontera surgen en muchas formas. Para concluir esta sección, se explora una posible forma y se remite al lector a los ejercicios y la comprobación en la realidad 7 para ver más ejemplos.

► EJEMPLO 7.7 Aplique el método de disparo al problema de valor de frontera

$$\begin{cases} y_1' = (4 - 2y_2)/t^3 \\ y_2' = -e^{y_1} \\ y_1(1) = 0 \\ y_2(2) = 0 \\ t \text{ in } [1, 2]. \end{cases} \tag{7.9}$$

Si la condición inicial $y_2(1)$ estuviera presente, esto sería un problema de valor inicial. Se aplicará el método de disparo para determinar la incógnita $y_2(1)$, utilizando la rutina `ode45` de MATLAB como en el ejemplo 7.6 para resolver los problemas de valor inicial. Defina la función $F(s)$

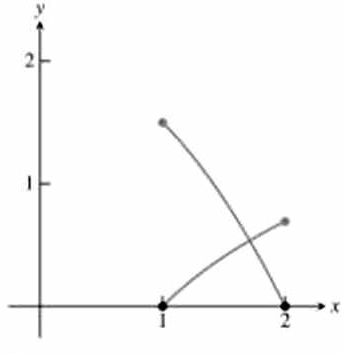


Figura 7.4 Solución del ejemplo 7.7 para el método del disparo. Se muestran las curvas $y_1(t)$ y $y_2(t)$. Los círculos negros denotan los datos frontera dados.

como la condición final $y_2(2)$, en la que se resuelve el PVI con las condiciones iniciales $y_1(1) = 0$ y $y_2(1) = s$. El objetivo es resolver $F(s) = 0$.

La solución se encierra al observar que $F(0) \approx -3.97$ y $F(2) \approx 0.87$. Una aplicación de `fzero(@F, [0 2])` encuentra $s^* = 1.5$. Si se usa `ode45` con valores iniciales $y_1(1) = 0$ y $y_2(1) = 1.5$ se obtiene la solución mostrada en la figura 7.4. Las soluciones exactas son $y_1(t) = \ln t$, $y_2(t) = 2 - t^2/2$.

7.1 Ejercicios

1. Muestre que las soluciones para los PVF lineales

$$\text{(a)} \begin{cases} y'' = y + 2e^t \\ y(0) = 0 \\ y(1) = e \end{cases} \qquad \text{(b)} \begin{cases} y'' = (2 + 4t^2)y \\ y(0) = 1 \\ y(1) = e \end{cases}$$

$$\text{(c)} \begin{cases} y'' = -y + 2\cos t \\ y(0) = 0 \\ y(\frac{\pi}{2}) = \frac{\pi}{2} \end{cases} \qquad \text{(d)} \begin{cases} y'' = 2 - 4y \\ y(0) = 0 \\ y(\frac{\pi}{2}) = 1 \end{cases}$$

son (a) $y = te^t$, (b) $y = e^{t^2}$, (c) $y = t \operatorname{sen} t$, (d) $y = \operatorname{sen}^2 t$, respectivamente.

2. Muestre que las soluciones a los PVF

$$\text{(a)} \begin{cases} y'' = \frac{3}{2}y^2 \\ y(1) = 4 \\ y(2) = 1 \end{cases} \quad \text{(b)} \begin{cases} y'' = 2yy' \\ y(0) = 0 \\ y(\frac{\pi}{4}) = 1 \end{cases} \quad \text{(c)} \begin{cases} y'' = -e^{-2y} \\ y(1) = 0 \\ y(e) = 1 \end{cases} \quad \text{(d)} \begin{cases} y'' = 6y^{\frac{1}{3}} \\ y(1) = 1 \\ y(2) = 8 \end{cases}$$

son (a) $y = 4t^{-2}$, (b) $y = \tan t$, (c) $y = \ln t$, (d) $y = t^3$, respectivamente.

3. Considere el problema de valor de frontera

$$\begin{cases} y'' = -4y \\ y(a) = y_a, \\ y(b) = y_b \end{cases}$$

(a) Encuentre dos soluciones linealmente independientes de la ecuación diferencial. (b) Suponga que $a = 0$ y $b = \pi$. ¿Qué condiciones sobre y_a, y_b deben cumplirse para que exista una solución? (c) Responda la misma pregunta que en (b), para $b = \pi/2$. (d) Responda la misma pregunta que en (b), para $b = \pi/4$.

4. Exprese, como la solución de un problema de valores en la frontera de segundo orden, la altura de un proyectil que se lanza desde la parte superior de un edificio con 60 metros de altura y tarda 5 segundos en llegar al suelo. Resuelva el problema de valor de frontera y encuentre la altura máxima alcanzada por el proyectil.

5. Encuentre todas las soluciones del PVF $y'' = ky$, $y(0) = y_0$, $y(1) = y_1$, para $k \geq 0$.

7.1 Problemas de computadora

1. Aplique el método de disparo a los PVF lineales. Empiece por encontrar un intervalo $[s_0, s_1]$ que encierre la solución. Utilice el comando `fzero` de MATLAB o el método de bisección para encontrar la solución. Grafique la solución aproximada en el intervalo especificado.

$$\text{(a)} \begin{cases} y'' = y + \frac{2}{3}e^t \\ y(0) = 0 \\ y(1) = \frac{1}{3}e \end{cases} \qquad \text{(b)} \begin{cases} y'' = (2 + 4t^2)y \\ y(0) = 1 \\ y(1) = e \end{cases}$$

2. Realice los pasos del problema de computadora 1 para los PVF.

$$\text{(a)} \begin{cases} 9y'' + \pi^2 y = 0 \\ y(0) = -1 \\ y\left(\frac{3}{2}\right) = 3 \end{cases} \qquad \text{(b)} \begin{cases} y'' = 3y - 2y' \\ y(0) = e^3 \\ y(1) = 1 \end{cases}$$

3. Aplique el método de disparo a los PVF no lineales. Encuentre un intervalo que encierre $[s_0, s_1]$ y aplique un método de aproximaciones para encontrar y graficar la solución.

$$\text{(a)} \begin{cases} y'' = 18y^2 \\ y(1) = \frac{1}{3} \\ y(2) = \frac{1}{12} \end{cases} \qquad \text{(b)} \begin{cases} y'' = 2e^{-2y}(1 - t^2) \\ y(0) = 0 \\ y(1) = \ln 2 \end{cases}$$

4. Realice los pasos del problema de computadora 3 para los PVF no lineales.

$$\text{(a)} \begin{cases} y'' = e^y \\ y(0) = 1 \\ y(1) = 3 \end{cases} \qquad \text{(b)} \begin{cases} y'' = \operatorname{sen} y' \\ y(0) = 1 \\ y(1) = -1 \end{cases}$$

5. Aplique el método de disparo a los sistemas no lineales de problemas de valores en la frontera. Siga el método del ejemplo 7.7.

$$\text{(a)} \begin{cases} y_1' = 1/y_2 \\ y_2' = t + \tan y_1 \\ y_1(0) = 0 \\ y_2(1) = 2 \end{cases} \qquad \text{(b)} \begin{cases} y_1' = y_1 - 3y_1 y_2 \\ y_2' = -6(ty_2 + \ln y_1) \\ y_1(0) = 1 \\ y_2(1) = -\frac{2}{3} \end{cases}$$

Comprobación ✔ en la realidad

7 Deformación de un anillo circular

Los problemas de valores en la frontera son modelos naturales para el cálculo de estructuras. Un sistema de siete ecuaciones diferenciales sirve como un modelo para un anillo circular con compresibilidad c, bajo presión hidrostática p proveniente de todas direcciones. Por simplicidad se considera que el modelo no tiene unidades y se asumirá que el anillo tiene radio 1 con simetría horizontal y vertical en ausencia de una presión externa. Aunque simplificado, el modelo resulta útil para estudiar el fenómeno de la **deformación**, o colapso de una forma anular circular. En Huddleston [2000] puede encontrarse este ejemplo y muchos otros problemas estructurales de valores en la frontera.

El modelo toma en cuenta sólo la **cuarta parte superior izquierda del anillo** (el resto puede llenarse mediante el supuesto de simetría). La variable independiente s representa la longitud de arco a lo largo de la línea central original del anillo, que va de $s = 0$ a $s = \pi/2$. Las variables dependientes en el punto especificado por la longitud de arco s son como sigue:

$y_1(s)$ = ángulo de la línea central con respecto a la horizontal
$y_2(s)$ = coordenada x
$y_3(s)$ = coordenada y
$y_4(s)$ = longitud de arco a lo largo de la línea central deformada
$y_5(s)$ = fuerza axial interna
$y_6(s)$ = fuerza normal interna
$y_7(s)$ = momento flexionante.

En la figura 7.5(a) se muestra el anillo y las primeras cuatro variables. El problema de valores en la frontera (vea, por ejemplo, Huddleston [2000]) es

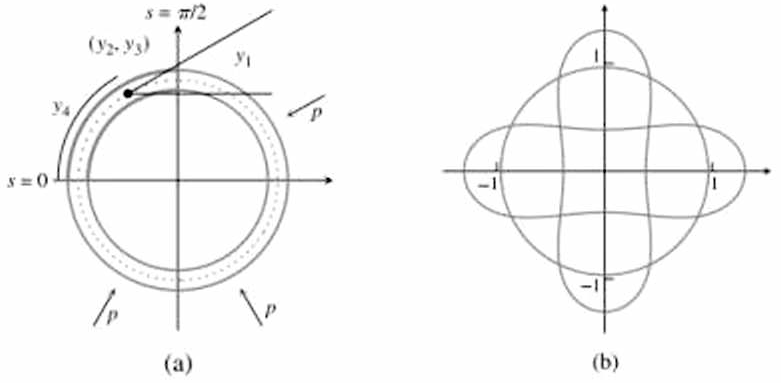


Figura 7.5 Esquema de la deformación de un anillo. (a) La variable s representa la longitud de arco a lo largo de la línea central punteada de la cuarta parte superior izquierda del anillo. (b) Tres soluciones diferentes para el PVF con parámetros $c = 0.01$, $p = 3.8$. Las dos soluciones con deformación son estables.

$$
\begin{aligned}
y_1' &= -1 - cy_5 + (c+1)y_7 && y_1(0) = \tfrac{\pi}{2} && y_1(\tfrac{\pi}{2}) = 0\\
y_2' &= (1 + c(y_5 - y_7))\cos y_1 && && y_2(\tfrac{\pi}{2}) = 0\\
y_3' &= (1 + c(y_5 - y_7))\operatorname{sen} y && y_3(0) = 0 &&\\
y_4' &= 1 + c(y_5 - y_7) && y_4(0) = 0 &&\\
y_5' &= -y_6(-1 - cy_5 + (c+1)y_7) && && \\
y_6' &= y_7 y_5 - (1 + c(y_5 - y_7))(y_5 + p) && y_6(0) = 0 && y_6(\tfrac{\pi}{2}) = 0\\
y_7' &= (1 + c(y_5 - y_7))y_6. && &&
\end{aligned}
$$

Bajo ninguna presión ($p = 0$), observe que $y_1 = \pi/2 - s$, $(y_2, y_3) = (-\cos s, \operatorname{sen} s)$, $y_4 = s$, $y_5 = y_6 = y_7 = 0$ es una solución. Esta solución es un perfecto cuarto de círculo, que corresponde a un anillo perfectamente circular debido a las simetrías.

De hecho, la siguiente solución circular para el problema de valores en la frontera existe para cualquier elección de los parámetros c y p:

$$
\begin{aligned}
y_1(s) &= \frac{\pi}{2} - s\\
y_2(s) &= \frac{c+1}{cp+c+1}(-\cos s)\\
y_3(s) &= \frac{c+1}{cp+c+1}\operatorname{sen} s\\
y_4(s) &= \frac{c+1}{cp+c+1}s\\
y_5(s) &= -\frac{c+1}{cp+c+1}p\\
y_6(s) &= 0\\
y_7(s) &= -\frac{cp}{cp+c+1}.
\end{aligned}
\tag{7.10}
$$

A medida que la presión aumenta desde cero, el radio del círculo disminuye. Cuando el parámetro p de la presión se incrementa aún más, hay una **bifurcación**, o cambios de estado, del anillo. La forma circular del anillo permanece matemáticamente posible, pero inestable, lo que significa

que hay pequeñas perturbaciones que hacen que el anillo pase a otra posible configuración (solución del PVF) que sea estable.

Para la presión p aplicada por abajo del punto de bifurcación, o **presión crítica** p_c, solo existe la solución (7.10). Para $p > p_c$, hay tres soluciones diferentes del PVF, que se muestran en la figura 7.5(b). Más allá de la presión crítica, el papel del anillo circular como un estado inestable es similar a la del péndulo invertido (problema de computadora 6.3.6) o el puente sin torsión en la comprobación en la realidad 6.

La presión crítica depende de la compresibilidad del anillo. Cuanto menor es el parámetro c, menos compresible es el anillo y menor es la presión crítica a la que cambia de forma, en vez de comprimirse con su forma original. El trabajo del lector consiste en utilizar el método de disparo junto con el método de Broyden para encontrar la presión crítica p_c y las deformaciones resultantes obtenidas para el anillo.

Actividades sugeridas:

1. Compruebe que (7.10) es una solución del PVF para cada compresibilidad c y presión p.
2. Establezca la compresibilidad en el valor moderado $c = 0.01$. Resuelva el PVF mediante el método del disparo para las presiones $p = 0$ y 3. La función F en el método de disparo debe utilizar los tres valores iniciales faltantes $(y_2(0), y_5(0), y_7(0))$ como entrada y los tres valores finales $(y_1(\pi/2), y_2(\pi/2), y_6(\pi/2))$ como salida. Para resolver las raíces de F, puede utilizarse el método multivariado de Broyden II del capítulo 2. Compare con la solución correcta (7.10). Observe que, para ambos valores de p, las diferentes condiciones iniciales del método de Broyden resultan en la misma trayectoria de solución. ¿Cuál es la disminución del radio cuando p aumenta de 0 a 3?
3. Grafique las soluciones del paso 2. La curva $(y_2(s), y_3(s))$ representa la cuarta parte superior izquierda del anillo. Use la simetría horizontal y vertical para graficar todo el anillo.
4. Cambie la presión a $p = 3.5$ y resuelva el PVF. Tenga en cuenta que la solución obtenida depende de la condición inicial que se utilice para el método de Broyden. Grafique cada solución diferente que encuentre.
5. Determine la presión crítica p_c para la compresibilidad $c = 0.01$, con una precisión de dos decimales. Para $p > p_c$, hay tres soluciones diferentes. Para $p < p_c$, sólo hay una solución (7.10).
6. Realice el paso 5 para la compresibilidad reducida $c = 0.001$. El anillo es más frágil. ¿El cambio en p_c para el caso de la compresibilidad reducida es consistente con su intuición?
7. Realice el paso 5 para una compresibilidad aumentada $c = 0.05$. ✔

7.2 MÉTODOS DE DIFERENCIAS FINITAS

La idea fundamental detrás de los métodos de diferencias finitas es reemplazar las derivadas en la ecuación diferencial por aproximaciones discretas y evaluar sobre una malla para desarrollar un sistema de ecuaciones. El enfoque de discretizar la ecuación diferencial también se utilizará en el capítulo 8 sobre EDP.

7.2.1 Problemas de valor de frontera lineales

Sea $y(t)$ una función con por lo menos cuatro derivadas continuas. En el capítulo 5 se desarrollaron aproximaciones discretas para la primera derivada

$$y'(t) = \frac{y(t+h) - y(t-h)}{2h} - \frac{h^2}{6}y'''(c) \tag{7.11}$$

y para la segunda derivada

$$y''(t) = \frac{y(t+h) - 2y(t) + y(t-h)}{h^2} + \frac{h^2}{12} f^{iv}(c). \tag{7.12}$$

Ambas son precisas hasta un error proporcional a h^2.

El método de diferencia finita consiste en sustituir las derivadas en la ecuación diferencial por sus versiones discretas y resolver las ecuaciones algebraicas resultantes, que son más simples, con aproximaciones w_i a los valores correctos y_i, como se muestra en la figura 7.6. Las condiciones de frontera se sustituyen en el sistema de ecuaciones donde sean necesarias.

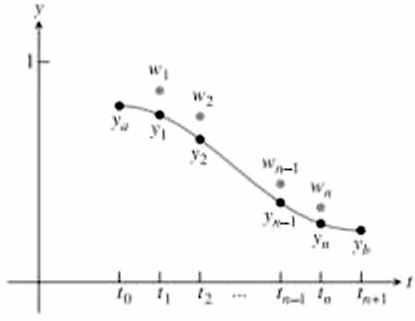


Figura 7.6 Método de diferencias finitas para PVF. Las aproximaciones w_i, $i = 1, \ldots, n$ para los valores correctos y_i en los puntos discretos t_i se calculan resolviendo un sistema lineal de ecuaciones.

Después de las sustituciones hay dos posibles situaciones. Si el problema de valor de frontera original era lineal, entonces el sistema resultante de ecuaciones es lineal y puede resolverse mediante la eliminación de Gauss o por medio de métodos iterativos. Si el problema original era no lineal, entonces el sistema algebraico es un sistema de ecuaciones no lineales, que requiere técnicas más sofisticadas. Se iniciará con un ejemplo lineal.

► EJEMPLO 7.8 Resuelve el PVF (7.7)

$$\begin{cases} y'' = 4y \\ y(0) = 1, \\ y(1) = 3 \end{cases}$$

usando diferencias finitas.

Considere la forma discreta de la ecuación diferencial $y'' = 4y$, utilizando el formulario de diferencias centradas para la segunda derivada. La versión de diferencias finitas en t_i es

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} - 4w_i = 0$$

o de forma equivalente

$$w_{i-1} + (-4h^2 - 2)w_i + w_{i+1} = 0.$$

Para $n = 3$, el tamaño del intervalo es $h = 1/(n+1) = 1/4$ y se tienen tres ecuaciones. Si se insertan las condiciones de frontera $w_0 = 1$ y $w_4 = 3$, queda el siguiente sistema que debe resolverse para w_1, w_2 y w_3:

$$\begin{aligned} 1 + (-4h^2 - 2)w_1 + w_2 &= 0 \\ w_1 + (-4h^2 - 2)w_2 + w_3 &= 0 \\ w_2 + (-4h^2 - 2)w_3 + 3 &= 0. \end{aligned}$$

Al sustituir h se obtiene la ecuación de la matriz tridiagonal

$$\begin{bmatrix} -\frac{9}{4} & 1 & 0 \\ 1 & -\frac{9}{4} & 1 \\ 0 & 1 & -\frac{9}{4} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -3 \end{bmatrix}.$$

La solución de este sistema por eliminación gaussiana da los valores aproximados de la solución en tres puntos 1.0249, 1.3061, 1.9138. La siguiente tabla muestra los valores aproximados w_i de la solución en t_i comparados con los valores de la solución correcta y_i (tenga en cuenta que los valores de frontera, w_0 y w_4, ya se conocen y no se calculan):

i	t_i	w_i	y_i
0	0.00	1.0000	1.0000
1	0.25	1.0249	1.0181
2	0.50	1.3061	1.2961
3	0.75	1.9138	1.9049
4	1.00	3.0000	3.0000

Las diferencias son del orden de 10^{-2}. Para obtener errores más pequeños, deben usarse n más grandes. En general, $h = (b - a)/(n + 1) = 1/(n + 1)$, y la ecuación de la matriz tridiagonal es

$$\begin{bmatrix} -4h^2 - 2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -4h^2 - 2 & \ddots & & 0 & 0 & 0 \\ 0 & 1 & \ddots & \ddots & 0 & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & 0 & \ddots & \ddots & 1 & 0 \\ 0 & 0 & 0 & & \ddots & -4h^2 - 2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -4h^2 - 2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{n-1} \\ w_n \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ -3 \end{bmatrix}.$$

A medida que se agregan más subintervalos, se espera que las aproximaciones w_i estén más cerca de las correspondientes y_i. ◀

Las posibles fuentes de error en el método de diferencias finitas son el error de truncamiento cometido por las fórmulas de diferencias centradas y el error cometido en la resolución del sistema de ecuaciones. Para los tamaños de paso h superiores a la raíz cuadrada del épsilon de la máquina, el último error domina. Este error es $O(h^2)$, por lo que se espera que el error disminuya a $O(n^{-2})$ a medida que crezca el número de subintervalos $n + 1$.

Esta expectativa se prueba para el problema (7.7). En la figura 7.7 se muestra la magnitud del error E de la solución en $t = 3/4$, para diversos números de subintervalos $n + 1$. En una gráfica log-log, el error en función del número de subintervalos es en esencia una línea recta con pendiente -2, lo que significa que $\log E \approx a + b \log n$, donde $b = -2$; en otras palabras, el error $E \approx Kn^{-2}$, tal como se esperaba.

7.2.2 Problemas de valor de frontera no lineales

Cuando el método de diferencias finitas se aplica a una ecuación diferencial no lineal, el resultado es un sistema de ecuaciones algebraicas no lineales que debe resolverse. En el capítulo 2 se utilizó

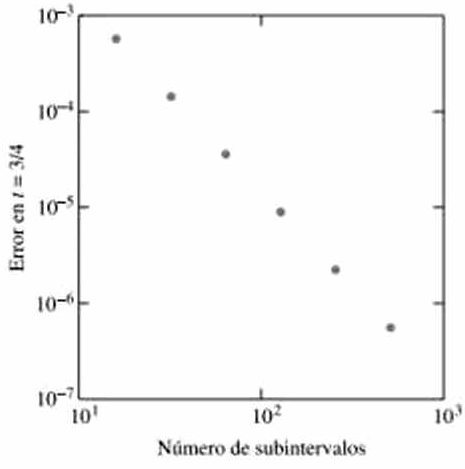


Figura 7.7 Convergencia del método de diferencias finitas. Gráfica del error $|w_i - y_i|$ en $t_i = 3/4$ del ejemplo 7.8 contra el número de subintervalos n. La pendiente es -2, lo que confirma que el error es $O(n^{-2}) = O(h^2)$.

el método multivariado de Newton para resolver dichos sistemas. Se demostró el uso de este método para aproximar el siguiente problema de valor de frontera no lineal:

► **EJEMPLO 7.9** Resuelva el PVF no lineal

$$\begin{cases} y'' = y - y^2 \\ y(0) = 1 \\ y(1) = 4 \end{cases} \tag{7.13}$$

por diferencias finitas.

La forma discretizada de la ecuación diferencial en t_i es

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} - w_i + w_i^2 = 0$$

o bien

$$w_{i-1} - (2 + h^2)w_i + h^2 w_i^2 + w_{i+1} = 0$$

para $2 \le i \le n - 1$, junto con la primera y última ecuaciones

$$y_a - (2 + h^2)w_1 + h^2 w_1^2 + w_2 = 0$$
$$w_{n-1} - (2 + h^2)w_n + h^2 w_n^2 + y_b = 0$$

que contienen la información de las condiciones de frontera.

ANOTACIÓN

Convergencia En la figura 7.7 se ilustra la convergencia de segundo orden del método de diferencias finitas. Esto se deduce del uso de las fórmulas de segundo orden (7.11) y (7.12). El conocimiento del orden permite aplicar la extrapolación, como se estudió en el capítulo 5. Para cada t y tamaño de paso h fijos, la aproximación $w_h(t)$ a partir del método de diferencias finitas es de segundo orden en h y puede extrapolarse mediante una fórmula sencilla. Los problemas de computadora 7 y 8 exploran esta oportunidad de acelerar la convergencia.

La resolución de la versión discretizada del problema de valores en la frontera F implica resolver $F(w) = 0$, lo que se hace mediante el método de Newton. El método multivariado de Newton es la iteración $w^{k+1} = w^k - DF(w^k)^{-1} F(w^k)$. Como siempre, lo mejor es llevar a cabo la iteración resolviendo para $\Delta w = w^{k+1} - w^k$ en la ecuación $DF(w^k)\Delta w = -F(w^k)$.

La función $F(w)$ está dada por

$$F\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{bmatrix} = \begin{bmatrix} y_a - (2+h^2)w_1 + h^2 w_1^2 + w_2 \\ w_1 - (2+h^2)w_2 + h^2 w_2^2 + w_3 \\ \vdots \\ w_{n-2} - (2+h^2)w_{n-1} + h^2 w_{n-1}^2 + w_n \\ w_{n-1} - (2+h^2)w_n + h^2 w_n^2 + y_b \end{bmatrix},$$

donde $y_a = 1$ y $y_b = 4$. El jacobiano $DF(w)$ de F es

$$\begin{bmatrix} 2h^2 w_1 - (2+h^2) & 1 & 0 & \cdots & 0 \\ 1 & 2h^2 w_2 - (2+h^2) & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & 1 & 0 \\ \vdots & \ddots & \ddots & 2h^2 w_{n-1} - (2+h^2) & 1 \\ 0 & \cdots & 0 & 1 & 2h^2 w_n - (2+h^2) \end{bmatrix}.$$

La i-ésima fila del jacobiano se determina tomando la derivada parcial de la i-ésima ecuación (el i-ésimo componente de F) con respecto a cada w_j.

En la figura 7.8(a) se muestra el resultado de la aplicación del método multivariado de Newton para resolver $F(w) = 0$, con $n = 40$. El código de MATLAB se da en el programa 7.1. Para alcanzar la convergencia dentro de la precisión de máquina, bastan veinte iteraciones del método de Newton.

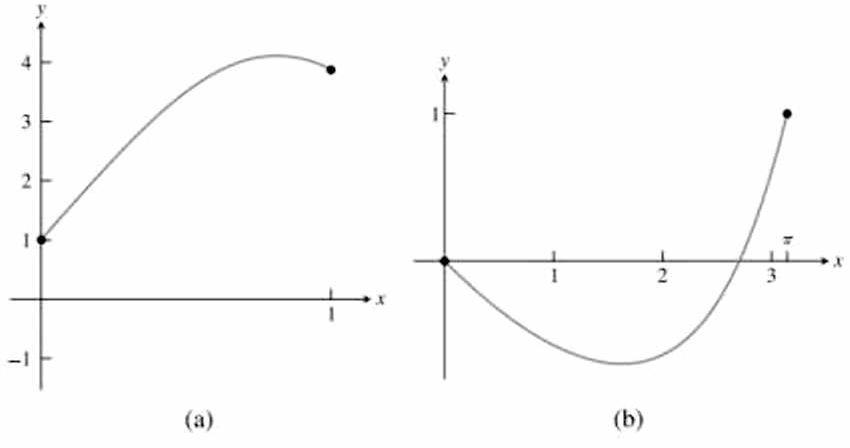


Figura 7.8 Soluciones de PVF no lineales mediante el método de diferencias finitas. (a) Solución del ejemplo 7.9 con $n = 40$, después de la convergencia del método de Newton. (b) Lo mismo, para el ejemplo 7.10.

```
% Programa 7.1 Método de diferencias finitas no lineal para PVF
% Usa el método Multivariado de Newton para resolver la ecuación no lineal
% Entradas: intervalo inter, valores de frontera bv, número de pasos n
% Salida: solución w
% Uso de ejemplo: w=nlbvpfd([0 1],[1 4],40)
function w=nlbvpfd(inter,bv,n);
```

```
a=inter(1); b=inter(2); ya=bv(1); yb=bv(2);
h=(b-a)/(n+1);                    % h es el tamaño de paso
w=zeros(n,1);                     % inicializa arreglo w de solución
for i=1:20                        % ciclo del paso de Newton
  w=w-jac(w,inter,bv,n)\f(w,inter,bv,n);
end
plot([a a+(1:n)*h b],[ya w' yb]);  % grafica w con datos de frontera

function y=f(w,inter,bv,n)
y=zeros(n,1);h=(inter(2)-inter(1))/(n+1);
y(1)=bv(1)-(2+h^2)*w(1)+h^2*w(1)^2+w(2);
y(n)=w(n-1)-(2+h^2)*w(n)+h^2*w(n)^2+bv(2);
for i=2:n-1
    y(i)=w(i-1)-(2+h^2)*w(i)+h^2*w(i)^2+w(i+1);
end

function a=jac(w,inter,bv,n)
a=zeros(n,n);h=(inter(2)-inter(1))/(n+1);
for i=1:n
  a(i,i)=2*h^2*w(i)-2-h^2;
end
for i=1:n-1
  a(i,i+1)=1;
  a(i+1,i)=1;
end
```

◄

► **EJEMPLO 7.10** Use diferencias finitas para resolver el problema de valor de frontera no lineal

$$\begin{cases} y'' = y' + \cos y \\ y(0) = 0 \\ y(\pi) = 1. \end{cases} \tag{7.14}$$

La forma discretizada de la ecuación diferencial en t_i es

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} - \frac{w_{i+1} - w_{i-1}}{2h} - \cos(w_i) = 0,$$

o bien

$$(1 + h/2)w_{i-1} - 2w_i + (1 - h/2)w_{i+1} - h^2 \cos w_i = 0,$$

para $2 \le i \le n - 1$, junto con la primera y última ecuaciones,

$$\begin{aligned} (1 + h/2)y_a - 2w_1 + (1 - h/2)w_2 - h^2 \cos w_1 &= 0 \\ (1 + h/2)w_{n-1} - 2w_n + (1 - h/2)y_b - h^2 \cos w_n &= 0, \end{aligned}$$

donde $y_a = 0$ y $y_b = 1$. Los lados izquierdos de las n ecuaciones forman una función valuada vectorialmente

$$F(w) = \begin{bmatrix} (1 + h/2)y_a - 2w_1 + (1 - h/2)w_2 - h^2 \cos w_1 \\ \vdots \\ (1 + h/2)w_{i-1} - 2w_i + (1 - h/2)w_{i+1} - h^2 \cos w_i \\ \vdots \\ (1 + h/2)w_{n-1} - 2w_n + (1 - h/2)y_b - h^2 \cos w_n \end{bmatrix}.$$

El jacobiano $DF(w)$ de F es

$$\begin{bmatrix} -2+h^2\operatorname{sen} w_1 & 1-h/2 & 0 & \cdots & 0 \\ 1+h/2 & -2+h^2\operatorname{sen} w_2 & \ddots & \ddots & \vdots \\ 0 & 1+h/2 & \ddots & 1-h/2 & 0 \\ \vdots & \ddots & \ddots & -2+h^2\operatorname{sen} w_{n-1} & 1-h/2 \\ 0 & \cdots & 0 & 1+h/2 & -2+h^2\operatorname{sen} w_n \end{bmatrix}.$$

El siguiente código puede insertarse en el programa 7.1, junto con los cambios apropiados en la información de las condiciones de frontera, para manejar el problema de valor de frontera no lineal:

```
function y=f(w,inter,bv,n)
  y=zeros(n,1);h=(inter(2)-inter(1))/(n+1);
  y(1)=-2*w(1)+(1+h/2)*bv(1)+(1-h/2)*w(2)-h*h*cos(w(1));
  y(n)=(1+h/2)*w(n-1)-2*w(n)-h*h*cos(w(n))+(1-h/2)*bv(2);
  for j=2:n-1
    y(j)=-2*w(j)+(1+h/2)*w(j-1)+(1-h/2)*w(j+1)-h*h*cos(w(j));
  end

function a=jac(w,inter,bv,n)
  a=zeros(n,n);h=(inter(2)-inter(1))/(n+1);
  for j=1:n
    a(j,j)=-2+h*h*sin(w(j));
  end
  for j=1:n-1
    a(j,j+1)=1-h/2;
    a(j+1,j)=1+h/2;
  end
```

En la figura 7.8(b) se muestra la curva $y(t)$ de la solución resultante. ◀

7.2 Problemas de computadora

1. Utilice diferencias finitas para aproximar las soluciones a los PVF lineales con $n = 9$, 19 y 39.

$$\text{(a)} \begin{cases} y'' = y + \frac{2}{3}e^t \\ y(0) = 0 \\ y(1) = \frac{1}{3}e \end{cases} \qquad \text{(b)} \begin{cases} y'' = (2 + 4t^2)y \\ y(0) = 1 \\ y(1) = e \end{cases}$$

Grafique las soluciones aproximadas junto con las soluciones exactas (a) $y(t) = te^t/3$ y (b) $y(t) = e^{t^2}$; además, muestre los errores como una función de t en una gráfica semilogarítmica por separado.

2. Utilice diferencias finitas para aproximar las soluciones a los PVF lineales con $n = 9$, 19 y 39.

$$\text{(a)} \begin{cases} 9y'' + \pi^2 y = 0 \\ y(0) = -1 \\ y(\frac{3}{2}) = 3 \end{cases} \qquad \text{(b)} \begin{cases} y'' = 3y - 2y' \\ y(0) = e^3 \\ y(1) = 1 \end{cases}$$

Grafique las soluciones aproximadas junto con las soluciones exactas (a) $y(t) = 3\operatorname{sen}\frac{\pi t}{3} - \cos\frac{\pi t}{3}$ y (b) $y(t) = e^{3-3t}$; además, muestre los errores como una función de t en una gráfica semilogarítmica por separado.

3. Use diferencias finitas para aproximar las soluciones a los problemas de valor de frontera no lineales con $n = 9, 19$ y 39.

$$\text{(a)} \begin{cases} y'' = 18y^2 \\ y(1) = \frac{1}{3} \\ y(2) = \frac{1}{12} \end{cases} \qquad \text{(b)} \begin{cases} y'' = 2e^{-2y}(1 - t^2) \\ y(0) = 0 \\ y(1) = \ln 2 \end{cases}$$

Grafique las soluciones aproximadas junto con las soluciones exactas (a) $y(t) = 1/(3t^2)$ y (b) $y(t) = \ln(t^2 + 1)$; además, muestre los errores como una función de t en una gráfica semilogarítmica por separado.

4. Use diferencias finitas para graficar las soluciones a los PVF no lineales con $n = 9, 19$ y 39.

$$\text{(a)} \begin{cases} y'' = e^y \\ y(0) = 1 \\ y(1) = 3 \end{cases} \qquad \text{(b)} \begin{cases} y'' = \operatorname{sen} y' \\ y(0) = 1 \\ y(1) = -1 \end{cases}$$

5. (a) Encuentre la solución del PVF $y'' = y$, $y(0) = 0$, $y(1) = 1$ analíticamente. (b) Implemente la versión de diferencias finitas de la ecuación y grafique la solución aproximada para $n = 15$. (c) Compare la aproximación con la solución exacta mediante una gráfica log-log del error en $t = 1/2$ contra n para $n = 2^p - 1, p = 2, \ldots, 7$.

6. Resuelva el PVF no lineal $4y'' = ty^4$, $y(1) = 2$, $y(2) = 1$ por diferencias finitas. Grafique la solución aproximada para $n = 15$. Compare su aproximación con la solución exacta $y(t) = 2/t$ para hacer una gráfica log-log del error en $t = 3/2$ para $n = 2^p - 1, p = 2, \ldots, 7$.

7. Extrapole las soluciones aproximadas del problema de computadora 5. Aplique la extrapolación de Richardson (sección 5.1) a la fórmula $N(h) = w_h(1/2)$, la aproximación por diferencias finitas con tamaño de paso h. ¿Qué tanto puede acercarse la extrapolación al valor exacto $y(1/2)$ si se usan sólo los valores aproximados de $h = 1/4, 1/8$ y $1/16$?

8. Extrapole las soluciones aproximadas del problema de computadora 6. Utilice la fórmula $N(h) = w_h(3/2)$, la aproximación por diferencias finitas con tamaño de paso h. ¿Qué tanto puede acercarse la extrapolación al valor exacto $y(3/2)$ si se usan sólo los valores aproximados de $h = 1/4, 1/8$ y $1/16$?

9. Resuelva el problema de valores en la frontera no lineal $y'' = \operatorname{sen} y$, $y(0) = 1$, $y(\pi) = 0$ por diferencias finitas. Grafique las aproximaciones para $n = 9, 19$ y 39.

10. Utilice diferencias finitas para resolver la ecuación

$$\begin{cases} y'' = 10y(1 - y) \\ y(0) = 0 \\ y(1) = 1 \end{cases}.$$

Grafique las aproximaciones para $n = 9, 19$ y 39.

11. Resuelva

$$\begin{cases} y'' = cy(1 - y) \\ y(0) = 0 \\ y(1/2) = 1/4 \\ y(1) = 1 \end{cases}$$

para $c > 0$, con tres posiciones decimales correctas. (Sugerencia: considere el PVF que se forma al fijar dos de las tres condiciones de frontera. Sea $G(c)$ la discrepancia en la tercera condición de frontera y utilice el método de bisección para resolver $G(c) = 0$).

7.3 COLOCACIÓN Y EL MÉTODO DEL ELEMENTO FINITO

Al igual que el método de diferencias finitas, la idea detrás de la colocación y el método del elemento finito es reducir el problema de valores en la frontera a la solución de un conjunto de ecuaciones algebraicas. Sin embargo, en vez de discretizar la ecuación diferencial sustituyendo las derivadas con diferencias finitas, a la solución se le da una forma funcional cuyos parámetros se ajustan mediante el método.

Elija un conjunto de funciones de base $\phi_1(t), \ldots, \phi_n(t)$, que pueden ser polinomios, funciones trigonométricas, splines u otras funciones simples. Después, considere la posible solución

$$y(t) = c_1\phi_1(t) + \cdots + c_n\phi_n(t). \tag{7.15}$$

La determinación de una solución aproximada se reduce a encontrar los valores de c_i. Se considerarán dos formas de encontrar los coeficientes.

El enfoque de colocación es sustituir (7.15) en el problema de valores en la frontera y evaluar en una malla de puntos. Este método es sencillo, reduciendo el problema a resolver un sistema de ecuaciones en c_i, lineales si el problema original era lineal. Cada punto da una ecuación, y su solución para c_i es un tipo de interpolación.

Un segundo enfoque, el método del elemento finito, procede tratando al ajuste como un problema de mínimos cuadrados en vez de una interpolación. La proyección de Galerkin se emplea para disminuir al mínimo la diferencia entre (7.15) y la solución exacta en el sentido del error cuadrado. El método del elemento finito se revisa en el capítulo 8 para resolver problemas de valor de frontera en ecuaciones diferenciales parciales.

7.3.1 Colocación

Considere el PVF

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b. \end{cases} \tag{7.16}$$

Elija n puntos, comenzando y terminando con los puntos en la frontera a y b, por ejemplo,

$$a = t_1 < t_2 < \cdots < t_n = b. \tag{7.17}$$

El método de colocación funciona sustituyendo la solución candidata (7.15) en la ecuación diferencial (7.16) y la evaluación de la ecuación diferencial en los puntos (7.17) para obtener n ecuaciones con n incógnitas $c_1, \ldots, c_n$.

Para empezar de la manera más sencilla posible, se eligen las funciones de base $\phi_j(t) = t^{j-1}$ para $1 \leq j \leq n$. La solución será de la forma

$$y(t) = \sum_{j=1}^{n} c_j\phi_j(t) = \sum_{j=1}^{n} c_j t^{j-1}. \tag{7.18}$$

Se escribirán n ecuaciones con n incógnitas $c_1, \ldots, c_n$. La primera y la última son las condiciones de frontera:

$$i = 1: \sum_{j=1}^{n} c_j a^{j-1} = y(a)$$

$$i = n: \sum_{j=1}^{n} c_j b^{j-1} = y(b).$$

Las restantes $n-2$ ecuaciones provienen de la ecuación diferencial evaluada en t_i para $2 \le i \le n-1$. La ecuación diferencial $y'' = f(t, y, y')$ aplicada a $y(t) = \sum_{j=1}^{n} c_j t^{j-1}$ es

$$\sum_{j=1}^{n}(j-1)(j-2)c_j t^{j-3} = f\left(t, \sum_{j=1}^{n} c_j t^{j-1}, \sum_{j=1}^{n} c_j (j-1) t^{j-2}\right) \tag{7.19}$$

La evaluación en t_i para cada i produce n ecuaciones que deben resolverse para las c_i. Si la ecuación diferencial es lineal, entonces las ecuaciones en las c_i serán lineales y podrán resolverse con facilidad. Este enfoque se ilustra con el siguiente ejemplo.

► **EJEMPLO 7.11** Resuelva el problema de valores en la frontera

$$\begin{cases} y'' = 4y \\ y(0) = 1 \\ y(1) = 3 \end{cases}$$

mediante el método de colocación.

La primera y última ecuaciones son las condiciones de frontera

$$c_1 = \sum_{j=1}^{n} c_j \phi_j(0) = y(0) = 1$$

$$c_1 + \cdots + c_n = \sum_{j=1}^{n} c_j \phi_j(1) = y(1) = 3.$$

Las otras $n-2$ ecuaciones provienen de (7.19), las cuales tienen la forma

$$\sum_{j=1}^{n}(j-1)(j-2)c_j t^{j-3} - 4\sum_{j=1}^{n} c_j t^{j-1} = 0.$$

Evaluando t_i para cada i, se obtiene

$$\sum_{j=1}^{n}[(j-1)(j-2)t_i^{j-3} - 4t_i^{j-1}]c_j = 0.$$

Las n ecuaciones forman un sistema lineal $Ac = g$, donde la matriz de coeficientes A está definida por

$$A_{ij} = \begin{cases} 1 \;\; 0 \;\; 0 \;\; \ldots \;\; 0 & \text{renglón } i = 1 \\ (j-1)(j-2)t_i^{j-3} - 4t_i^{j-1} & \text{renglones } i = 2 \text{ hasta } n-1 \\ 1 \;\; 1 \;\; 1 \;\; \ldots \;\; 1 & \text{renglón } i = n \end{cases}$$

y $g = (1, 0, 0, \ldots, 0, 3)^T$. Con frecuencia se utilizan los puntos de la malla espaciados de manera uniforme

$$t_i = a + \frac{i-1}{n-1}(b-a) = \frac{i-1}{n-1}.$$

Después de resolver para las c_j, se obtiene la solución aproximada $y(t) = \sum c_j t^{j-1}$.

Para $n = 2$, el sistema $Ac = g$ es

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix},$$

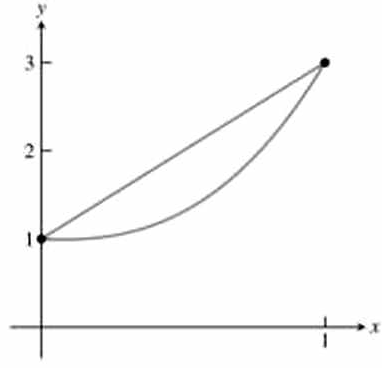


Figura 7.9 Soluciones de la PVF lineal del ejemplo 7.11 mediante el método de colocación. Se muestran las soluciones con $n = 2$ (curva superior) y $n = 4$ (curva inferior).

y la solución es $c = [1, 2]^T$. La solución aproximada (7.18) es la línea recta $y(t) = c_1 + c_2 t = 1 + 2t$. Al calcular para $n = 4$ se obtiene la solución aproximada $y(t) \approx 1 - 0.1886t + 1.0273t^2 + 1.1613t^3$. Las soluciones para $n = 2$ y $n = 4$ se representan gráficamente en la figura 7.9. Ya para $n = 4$ la aproximación es muy cercana a la solución exacta (7.4), como se muestra en la figura 7.3(b). Es posible lograr mayor precisión si se incrementa n.

Las ecuaciones que deben resolverse para c_i en el ejemplo 7.11 son lineales porque la ecuación diferencial es lineal. Los problemas de valores en la frontera no lineales pueden resolverse por colocación de una manera similar. Para resolver el sistema resultante de ecuaciones no lineales, se utiliza el método de Newton, tal como en el método de diferencias finitas.

Aunque se ha ilustrado el uso de la colocación con funciones de base monomiales debido a su sencillez, existen muchas mejores opciones. Por lo general, no se recomiendan las bases polinomiales. Dado que la colocación realiza una interpolación de la solución, el uso de funciones de base polinomial hace que el método sea susceptible al fenómeno de Runge (capítulo 3). El hecho de que los elementos de base monomiales t^i no sean ortogonales entre sí como funciones, hace que la matriz de coeficientes de las ecuaciones lineales esté mal condicionada cuando n es grande. Una forma de mejorar el condicionamiento consiste en usar las raíces de los polinomios de Chebyshev como puntos de evaluación, en vez de los puntos espaciados de manera uniforme.

La elección de funciones trigonométricas como funciones de base en la colocación conduce al análisis de Fourier y a los **métodos espectrales**, que se utilizan tanto en los problemas de valores en la frontera como para las ecuaciones diferenciales parciales. Éste es un enfoque "global", donde las funciones de base son distintas de cero en un amplio rango de t, pero tiene buenas propiedades de ortogonalidad. En el capítulo 10 se estudiarán las aproximaciones discretas de Fourier.

7.3.2 Elementos finitos y el método de Galerkin

La elección de splines como funciones de base conduce al **método del elemento finito**. En este enfoque, cada función de base es distinta de cero sólo en un intervalo corto de t. Los métodos del elemento finito se utilizan mucho para PVF y EDP en las dimensiones superiores, en especial cuando las fronteras irregulares hacen que la parametrización mediante funciones de base estándar resulte inconveniente.

En la colocación se supuso una forma funcional $y(t) = \sum c_i \phi_i(t)$ y se resolvió para los coeficientes c_i forzando la solución para satisfacer las condiciones de frontera y satisfacer exactamente la ecuación diferencial en puntos discretos. Por otro lado, el enfoque de Galerkin disminuye al mínimo el error cuadrático de la ecuación diferencial a lo largo de la solución. Esto conduce a un sistema diferente de ecuaciones para las c_i.

El método del elemento finito para PVF

$$\begin{cases} y'' = f(t, y, y') \\ y(a) = y_a \\ y(b) = y_b. \end{cases}$$

consiste en elegir la solución aproximada y de forma que el **residuo** $r = y'' - f$, la diferencia en los dos lados de la ecuación diferencial, sea lo más pequeño posible. En analogía con los métodos de mínimos cuadrados del capítulo 4, esto se logra al elegir y de modo que el residuo sea ortogonal al espacio vectorial de las posibles soluciones.

Para un intervalo $[a, b]$, defina el espacio vectorial de funciones integrables cuadradas

$$L^2[a, b] = \left\{ \text{funciones } y(t) \text{ en } [a, b] \;\middle|\; \int_a^b y(t)^2\, dt \text{ existe y es finita} \right\}.$$

El espacio funcional L^2 tiene un **producto interno**

$$\langle y_1, y_2 \rangle = \int_a^b y_1(t) y_2(t)\, dt$$

que tiene las propiedades usuales:

1. $\langle y_1, y_1 \rangle \geq 0$;
2. $\langle \alpha y_1 + \beta y_2, z \rangle = \alpha \langle y_1, z \rangle + \beta \langle y_2, z \rangle$ para los escalares α, β;
3. $\langle y_1, y_2 \rangle = \langle y_2, y_1 \rangle$.

Dos funciones y_1 y y_2 son **ortogonales** en $L^2[a, b]$ si $\langle y_1, y_2 \rangle = 0$. Como $L^2[a, b]$ es un espacio vectorial de dimensión infinita, no es posible hacer que el residuo $r = y'' - f$ sea ortogonal a todas las $L^2[a, b]$ por medio de un cálculo finito. Sin embargo, puede elegirse una base que cubra tanto de L^2 como sea posible con los recursos computacionales disponibles. Sea el conjunto de $n + 2$ funciones de base indicado por $\phi_0(t), \ldots, \phi_{n+1}(t)$. Esto se especificará más adelante.

El método de Galerkin consiste en dos ideas principales. La primera es disminuir al mínimo r forzándola a ser ortogonal a las funciones de base, en el sentido del producto interno L^2. Esto significa forzar $\int_a^b (y'' - f)\phi_i\, dt = 0$, o

$$\int_a^b y''(t)\phi_i(t)\, dt = \int_a^b f(t, y, y')\phi_i(t)\, dt \tag{7.20}$$

para cada $0 \leq i \leq n + 1$. La forma (7.20) se llama la **forma débil** del problema de valor de frontera.

La segunda idea de Galerkin es utilizar la integración por partes para eliminar las segundas derivadas. Tenga en cuenta que

$$\begin{aligned} \int_a^b y''(t)\phi_i(t)\, dt &= \phi_i(t)y'(t)\Big|_a^b - \int_a^b y'(t)\phi_i'(t)\, dt \\ &= \phi_i(b)y'(b) - \phi_i(a)y'(a) - \int_a^b y'(t)\phi_i'(t)\, dt. \end{aligned} \tag{7.21}$$

Si se usan (7.20) y (7.21) juntas resulta un conjunto de ecuaciones

$$\int_a^b f(t, y, y')\phi_i(t)\, dt = \phi_i(b)y'(b) - \phi_i(a)y'(a) - \int_a^b y'(t)\phi_i'(t)\, dt \tag{7.22}$$

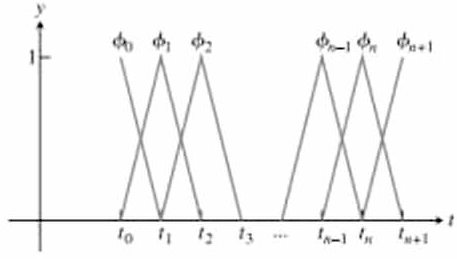


Figura 7.10 Splines B lineales por partes usadas como elementos finitos. Cada $\phi_i(t)$, para $1 \le i \le n$, tiene soporte en el intervalo que va de t_{i-1} a t_{i+1}.

para cada i que puede resolverse para las c_i en la forma funcional

$$y(t) = \sum_{i=0}^{n+1} c_i \phi_i(t). \tag{7.23}$$

Las dos ideas de Galerkin hacen conveniente el uso de funciones muy simples, como los elementos finitos $\phi_i(t)$. Las splines B lineales por partes se introducirán directamente y sólo se remitirá al lector a la bibliografía para conocer opciones más elaboradas.

Comience con una malla $t_0 < t_1 < \ldots < t_n < t_{n+1}$ de puntos sobre el eje t. Para $i = 1, \ldots, n$ defina

$$\phi_i(t) = \begin{cases} \dfrac{t - t_{i-1}}{t_i - t_{i-1}} & \text{para } t_{i-1} < t \le t_i \\ \dfrac{t_{i+1} - t}{t_{i+1} - t_i} & \text{para } t_i < t < t_{i+1}. \\ 0 & \text{en otro caso} \end{cases}$$

También defina

$$\phi_0(t) = \begin{cases} \dfrac{t_1 - t}{t_1 - t_0} & \text{para } t_0 \le t < t_1 \\ 0 & \text{en otro caso} \end{cases} \quad \text{y} \quad \phi_{n+1}(t) = \begin{cases} \dfrac{t - t_n}{t_{n+1} - t_n} & \text{para } t_n < t \le t_{n+1} \\ 0 & \text{en otro caso} \end{cases}.$$

Las funciones de "techo" lineales por partes ϕ_i, que se muestran en la figura 7.10, satisfacen la siguiente propiedad interesante:

$$\phi_i(t_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \ne j \end{cases}. \tag{7.24}$$

Para un conjunto de puntos de datos (t_i, c_i), defina la **spline B lineal por partes**

$$S(t) = \sum_{i=0}^{n+1} c_i \phi_i(t).$$

Se deduce de inmediato a partir de (7.24) que $S(t_j) = \sum_{i=0}^{n+1} c_i \phi_i(t_j) = c_j$. Por lo tanto, $S(t)$ es una función lineal por partes que interpola los puntos de datos (t_i, c_i). En otras palabras, ¡las coordenadas y son los coeficientes! Esto simplificará la interpretación de la solución (7.23). Las c_i no son sólo los coeficientes, sino también los valores de la solución en los puntos de la malla t_i.

ANOTACIÓN **Ortogonalidad** En el capítulo 4 se vio que la distancia de un punto a un plano se disminuye al mínimo al dibujar el segmento perpendicular desde el punto hasta el plano. El plano representa los candidatos para aproximar el punto, la distancia entre ellos es el error de aproximación. Este simple hecho sobre la ortogonalidad impregna el análisis numérico. Es el núcleo de la aproximación por mínimos cuadrados y es fundamental para el método de Galerkin en problemas de valor de frontera y de ecuaciones diferenciales parciales, así como de la cuadratura gaussiana (capítulo 5), la compresión (vea los capítulos 10 y 11), y las soluciones a problemas de valores propios (capítulo 12).

Ahora se muestra cómo calcular las c_i para resolver el PVF (7.16). La primera y la última de las c_i se encuentran mediante la colocación:

$$y(a) = \sum_{i=0}^{n+1} c_i \phi_i(a) = c_0 \phi_0(a) = c_0$$
$$y(b) = \sum_{i=0}^{n+1} c_i \phi_i(b) = c_{n+1} \phi_{n+1}(b) = c_{n+1}.$$

Para $i = 1, \ldots, n$, utilice las ecuaciones de elemento finito (7.22):

$$\int_a^b f(t, y, y') \phi_i(t)\, dt + \int_a^b y'(t) \phi_i'(t)\, dt = 0,$$

o al sustituir la forma funcional $y(t) = \sum c_i \phi_i(t)$,

$$\int_a^b \phi_i(t) f(t, \sum c_j \phi_j(t), \sum c_j \phi_j'(t))\, dt + \int_a^b \phi_i'(t) \sum c_j \phi_j'(t)\, dt = 0. \tag{7.25}$$

Observe que los términos de frontera de (7.22) son cero para $i = 1, \ldots, n$.

Suponga que la malla está espaciada de manera uniforme con tamaño de paso h. Se requerirán las siguientes integrales, para $i = 1, \ldots, n$:

$$\int_a^b \phi_i(t) \phi_{i+1}(t)\, dt = \int_0^h \frac{t}{h}\left(1 - \frac{t}{h}\right) dt = \int_0^h \left(\frac{t}{h} - \frac{t^2}{h^2}\right) dt$$
$$= \frac{t^2}{2h} - \frac{t^3}{3h^2}\bigg|_0^h = \frac{h}{6} \tag{7.26}$$

$$\int_a^b (\phi_i(t))^2\, dt = 2\int_0^h \left(\frac{t}{h}\right)^2 dt = \frac{2}{3}h \tag{7.27}$$

$$\int_a^b \phi_i'(t) \phi_{i+1}'(t)\, dt = \int_0^h \frac{1}{h}\left(-\frac{1}{h}\right) dt = -\frac{1}{h} \tag{7.28}$$

$$\int_a^b (\phi_i'(t))^2\, dt = 2\int_0^h \left(\frac{1}{h}\right)^2 dt = \frac{2}{h}. \tag{7.29}$$

Las fórmulas (7.26) a (7.29) se utilizan para simplificar (7.25) un vez sustituidos los detalles de la ecuación diferencial $y'' = f(t, y, y')$. Siempre que la ecuación diferencial sea lineal, las ecuaciones resultantes para las c_i serán lineales.

► EJEMPLO 7.12 Aplique el método del elemento finito para el PVF

$$\begin{cases} y'' = 4y \\ y(0) = 1. \\ y(1) = 3 \end{cases}$$

Al sustituir la ecuación diferencial en (7.25) se obtiene, para cada i, la ecuación

$$\begin{aligned} 0 &= \int_0^1 \left(4\phi_i(t) \sum_{j=0}^{n+1} c_j \phi_j(t) + \sum_{j=0}^{n+1} c_j \phi_j'(t) \phi_i'(t) \right) dt \\ &= \sum_{j=0}^{n+1} c_j \left[4 \int_0^1 \phi_i(t) \phi_j(t)\, dt + \int_0^1 \phi_j'(t) \phi_i'(t)\, dt \right]. \end{aligned}$$

Si se usan las relaciones de spline B (7.26) a (7.29) para $i = 1, \ldots, n$, así como las relaciones $c_0 = f(a)$, $c_{n+1} = f(b)$, se encuentra que las ecuaciones son

$$\begin{aligned} \left[\frac{2}{3}h - \frac{1}{h}\right] c_0 + \left[\frac{8}{3}h + \frac{2}{h}\right] c_1 + \left[\frac{2}{3}h - \frac{1}{h}\right] c_2 &= 0 \\ \left[\frac{2}{3}h - \frac{1}{h}\right] c_1 + \left[\frac{8}{3}h + \frac{2}{h}\right] c_2 + \left[\frac{2}{3}h - \frac{1}{h}\right] c_3 &= 0 \\ &\vdots \\ \left[\frac{2}{3}h - \frac{1}{h}\right] c_{n-1} + \left[\frac{8}{3}h + \frac{2}{h}\right] c_n + \left[\frac{2}{3}h - \frac{1}{h}\right] c_{n+1} &= 0. \end{aligned} \tag{7.30}$$

Observe que se tiene $c_0 = y_a = 1$ y $c_{n+1} = y_b = 3$, por lo que la forma matricial de las ecuaciones es simétrica tridiagonal

$$\begin{bmatrix} \alpha & \beta & 0 & \cdots & 0 \\ \beta & \alpha & \ddots & \ddots & \vdots \\ 0 & \beta & \ddots & \beta & 0 \\ \vdots & \ddots & \ddots & \alpha & \beta \\ 0 & \cdots & 0 & \beta & \alpha \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} -y_a\beta \\ 0 \\ \vdots \\ 0 \\ -y_b\beta \end{bmatrix}$$

donde

$$\alpha = \frac{8}{3}h + \frac{2}{h} \quad \text{y} \quad \beta = \frac{2}{3}h - \frac{1}{h}.$$

Recuerde el comando `spdiags` de MATLAB que se usó en el capítulo 2; con éste puede escribirse una implementación simple que es muy compacta:

```
% Programa 7.2 Solución por elemento finito de un PVF lineal
% Entradas: intervalo inter, valores de frontera bv, número de pasos n.
% Salida: valores de solución c
% Uso de ejemplo: c=bvpfem ([0 1],[1 3],9);
function c=bvpfem(inter,bv,n)
a=inter(1); b=inter(2); ya=bv(1); yb=bv(2);
h=(b-a)/(n+1);
alpha=(8/3)*h+2/h; beta=(2/3)*h-1/h;
e=ones(n,1);
M=spdiags([beta*e alpha*e beta*e],-1:1,n,n);
```

```
d=zeros(n,1);
d(1)=-ya*beta;
d(n)=-yb*beta;
c=M\d;
```

Para $n = 3$, el código MATLAB da las siguientes c_i:

i	t_i	$w_i = c_i$	y_i
0	0.00	1.0000	1.0000
1	0.25	1.0109	1.0181
2	0.50	1.2855	1.2961
3	0.75	1.8955	1.9049
4	1.00	3.0000	3.0000

La solución aproximada w_i en t_i tiene el valor c_i, que se compara con la solución exacta y_i. Los errores son aproximadamente de 10^{-2}, el mismo tamaño de error que para el método de diferencias finitas. De hecho, la figura 7.11 muestra que la ejecución del método del elemento finito con valores mayores de n da una curva de convergencia casi idéntica a la del método de diferencias finitas de la figura 7.7, mostrando la convergencia $O(n^{-2})$.

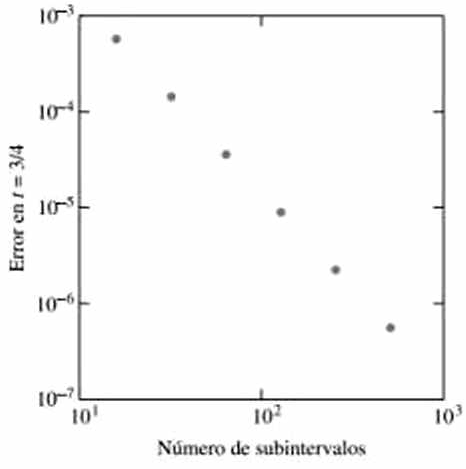


Figura 7.11 Convergencia del método del elemento finito. Gráfica del error $|w_i - y_i|$ para el ejemplo 7.12 en $t_i = 3/4$ contra el número de subintervalos n. De acuerdo con la pendiente, el error es $O(n^{-2}) = O(h^2)$.

7.3 Problemas de computadora

1. Utilice el método de colocación con $n = 8$ y 16 para aproximar las soluciones a los problemas de valor de frontera lineales

$$\text{(a)} \begin{cases} y'' = y + \frac{2}{3}e^t \\ y(0) = 0 \\ y(1) = \frac{1}{3}e \end{cases} \qquad \text{(b)} \begin{cases} y'' = (2 + 4t^2)y \\ y(0) = 1 \\ y(1) = e \end{cases}$$

   Grafique las soluciones aproximadas junto con las soluciones exactas (a) $y(t) = te^t/3$ y (b) $y(t) = e^{t^2}$, asimismo, muestre los errores como una función de t en una gráfica semilogarítmica por separado.

2. Utilice el método de colocación con $n = 8$ y 16 para aproximar las soluciones a los problemas de valor de frontera lineales

$$\text{(a)} \quad \begin{cases} 9y'' + \pi^2 y = 0 \\ y(0) = -1 \\ y(\frac{3}{2}) = 3 \end{cases} \qquad \text{(b)} \quad \begin{cases} y'' = 3y - 2y' \\ y(0) = e^3 \\ y(1) = 1 \end{cases}$$

Grafique las soluciones aproximadas junto con las soluciones exactas (a) $y(t) = 3 \operatorname{sen} \pi t/3 - \cos \pi t/3$ y (b) $y(t) = e^{3-3t}$. Asimismo, muestre los errores como una función de t en una gráfica semilogarítmica por separado.

3. Realice los pasos del problema de computadora 1, utilizando el método del elemento finito.
4. Realice los pasos del problema de computadora 2, utilizando el método del elemento finito.

Software y lecturas adicionales

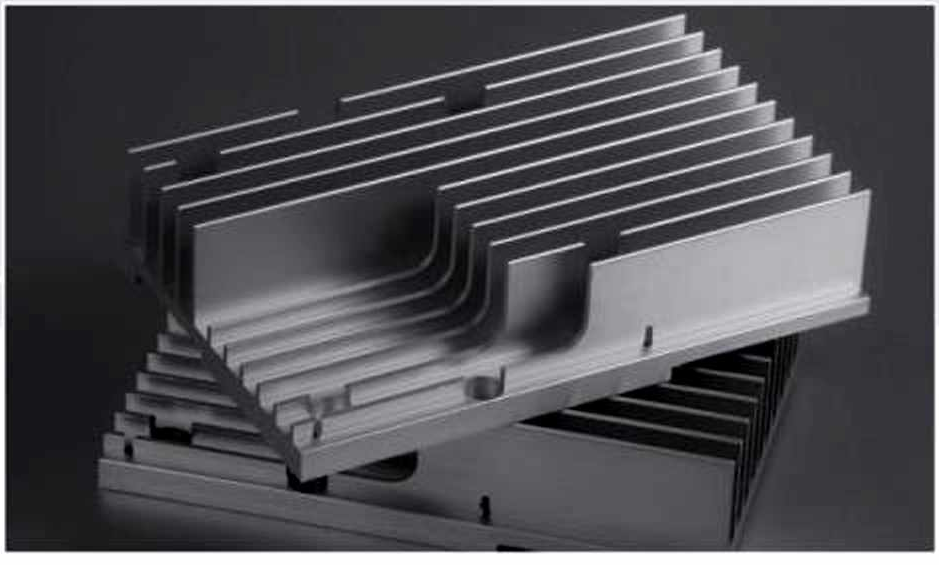
Los problemas de valor de frontera se analizan en la mayoría de los textos que tratan sobre ecuaciones diferenciales ordinarias. Ascher *et al.* [1995] es un estudio exhaustivo de las técnicas usadas en problemas de valor de frontera con EDO, incluyendo los métodos de disparo múltiple que no se cubrieron en este capítulo. Otras referencias buenas sobre los métodos de disparo y los métodos de diferencias finitas para PVF son Keller [1968], Bailey *et al.* [1968] y Roberts y Shipman [1972].

Las rutinas BVPMS y BVPFD del IMSL son implementaciones de los métodos de disparo y los métodos de diferencias finitas, respectivamente, para PVF de dos puntos. BVPFD utiliza un método de diferencias finitas con orden y tamaño de paso variables.

El programa D02HAF de NAG implementa un método de disparo para PVF de dos puntos, utilizando el método de Runge-Kutta-Merson y la iteración de Newton. La rutina D02GAF implementa una técnica de diferencias finitas con iteración de Newton para resolver las ecuaciones resultantes. La matriz jacobiana se calcula por diferenciación numérica. Por último, D02JAF resuelve un PVF lineal para una sola EDO de orden n mediante la colocación.

La biblioteca Netlib contiene dos subrutinas de Fortran que el usuario puede invocar: MUSL, para problemas lineales, y MUSN, para problemas no lineales. Ambas se basan en métodos de disparo.

CAPÍTULO



8

Ecuaciones diferenciales parciales

Las unidades centrales de procesamiento 8086 fabricadas por Intel Corp. en la década de 1970 corrían a 5 MHz y requerían menos de 5 watts de potencia. Hoy en día, las velocidades aumentaron en un factor de varios cientos, y los chips disipan más de 50 watts. Para evitar los daños que pueden causar las temperaturas excesivamente altas en el procesador, es esencial distribuir el calor usando un ventilador y un disipador. Las consideraciones de enfriamiento son un obstáculo constante para extender la ley de Moore a velocidades de procesamiento más rápidas.

El transcurso de tiempo para la disipación de calor se modela bien mediante una EDP parabólica. Cuando el calor alcanza un equilibrio, la distribución en estado estacionario se puede modelar con una ecuación elíptica.

Comprobación en la realidad En la página 403 se muestra cómo se modela una configuración sencilla del disipador de calor, utilizando una ecuación diferencial parcial elíptica con condiciones de frontera de convección térmica.

Una ecuación diferencial parcial es una ecuación diferencial con más de una variable independiente. Aunque el tema es muy amplio, el análisis aquí se limitará a las ecuaciones con dos variables independientes que tienen la forma

$$Au_{xx} + Bu_{xy} + Cu_{yy} + F(u_x, u_y, u, x, y) = 0, \tag{8.1}$$

donde las derivadas parciales se indican mediante los subíndices x y y para las variables independientes, y u indica la solución. Cuando una de las variables representa el tiempo, como en la ecuación de calor, se prefiere llamar las variables independientes x y t.

Dependiendo de los términos de orden principal de (8.1), las soluciones tendrán propiedades bastante diferentes. Las EDP de segundo orden con dos variables independientes se clasifican como sigue:

(1) Parabólica si $B^2 - 4AC = 0$
(2) Hiperbólica si $B^2 - 4AC > 0$
(3) Elíptica si $B^2 - 4AC < 0$

La diferencia práctica es que las ecuaciones parabólicas e hiperbólicas están definidas en una región abierta. Las condiciones de frontera para una variable (en la mayoría de los casos el tiempo)

se especifican en un extremo de la región y la solución del sistema se encuentra alejándose de esa frontera. Por otro lado, las ecuaciones elípticas suelen especificarse con condiciones de frontera en todo el contorno de una región cerrada. Se estudiarán algunos ejemplos de cada tipo y se ilustrarán los métodos numéricos disponibles para obtener soluciones aproximadas.

8.1 ECUACIONES PARABÓLICAS

La ecuación de calor

$$u_t = Du_{xx} \tag{8.2}$$

representa la temperatura x medida a lo largo de una placa homogénea unidimensional. La constante $D > 0$ se denomina **coeficiente de difusión**, que representa la difusividad térmica del material que forma la placa. La ecuación de calor modela la propagación del calor desde las regiones de mayor concentración hasta las de menor concentración. Las variables independientes son x y t.

En (8.2) se usa la variable t en vez de y porque representa el tiempo. A partir de la clasificación anterior, se tiene $B^2 - 4AC = 0$, por lo que la ecuación de calor es parabólica. La denominada ecuación de calor es un ejemplo de una **ecuación de difusión**, que modela la difusión de una sustancia. En la ciencia de materiales, es la misma ecuación que se conoce como segunda ley de Fick y describe la difusión de una sustancia dentro de un medio.

Al igual que en el caso de las EDO, la EDP (8.2) tiene soluciones infinitas y se requieren condiciones adicionales para determinar una solución particular. Los capítulos 6 y 7 trataron la solución de ecuaciones diferenciales ordinarias, donde se usaron condiciones iniciales o condiciones de frontera, respectivamente. Con el fin de representar correctamente una EDP, pueden emplearse varias combinaciones de condiciones iniciales y de frontera.

Para la ecuación de calor, un análisis sencillo puede sugerir cuáles condiciones deben exigirse. Para especificar la situación de manera única, es necesario conocer la distribución de la temperatura inicial a lo largo de la placa y lo que está sucediendo en los extremos de la misma a medida que avanza el tiempo. La ecuación de calor correctamente ubicada sobre un intervalo finito tiene la forma

$$\begin{cases} u_t = Du_{xx} \text{ para toda } a \le x \le b, t \ge 0 \\ u(x,0) = f(x) \text{ para toda } a \le x \le b \\ u(a,t) = l(t) \text{ para toda } t \ge 0 \\ u(b,t) = r(t) \text{ para toda } t \ge 0 \end{cases}, \tag{8.3}$$

donde la placa se extiende a lo largo del intervalo $a \le x \le b$. El coeficiente de difusión D controla la tasa de transferencia de calor. La función $f(x)$ en $[a, b]$ da la distribución de la temperatura inicial a lo largo de la placa, y $l(t)$, $r(t)$ para $t \ge 0$ proporciona la temperatura en los extremos. Aquí, se ha utilizado una combinación de condiciones iniciales $f(x)$ y condiciones de frontera $l(t)$ y $r(t)$para especificar una solución única de la EDP.

8.1.1 Método de las diferencias hacia adelante

El uso de los métodos de diferencias finitas para aproximar la solución de una ecuación diferencial parcial sigue la dirección establecida en los dos capítulos anteriores. La idea es establecer una malla en las variables independientes y discretizar la EDP. El problema continuo se ha convertido en un problema discreto con un número finito de ecuaciones. Si la EDP es lineal, las ecuaciones discretas son lineales y pueden resolverse mediante los métodos del capítulo 2.

Para discretizar la ecuación de calor en el intervalo de tiempo $[0, T]$, se considera una cuadrícula o malla de puntos, como la que se muestra en la figura 8.1. Los círculos rellenos representan valores ya conocidos de la solución $u(x, t)$ a partir de las condiciones iniciales y de frontera, y los círculos no rellenos son los puntos de la malla que deben rellenarse mediante el método. La solución exacta se indicará mediante $u(x_i, t_j)$ y su aproximación en (x_i, t_j) mediante w_{ij}. Sean M y N el

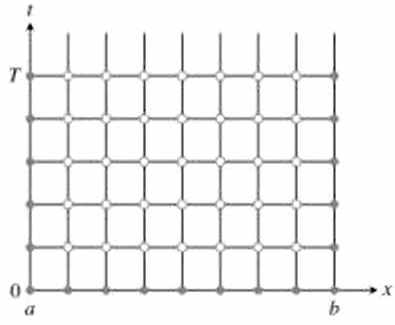


Figura 8.1 Malla para el método de diferencias finitas. Los círculos rellenos representan las condiciones iniciales y de frontera conocidas. Los círculos no rellenos representan los valores desconocidos que deben determinarse.

número total de pasos en las direcciones x y t, además sean $h = (b - a)/M$ y $k = T/N$ los tamaños de paso en las direcciones x y t.

Para aproximar las derivadas en las direcciones x y t pueden usarse las fórmulas de discretización del capítulo 5. Por ejemplo, la aplicación de la fórmula de la diferencia centrada para la segunda derivada sobre la variable x resulta en

$$u_{xx}(x,t) \approx \frac{1}{h^2}(u(x+h,t) - 2u(x,t) + u(x-h,t)), \tag{8.4}$$

con error $h^2\, u_{xxxx}(c_1, t)/12$; y la fórmula de la diferencia hacia adelante para la primera derivada usada para la variable del tiempo da

$$u_t(x,t) \approx \frac{1}{k}(u(x,t+k) - u(x,t)), \tag{8.5}$$

con error $ku_{tt}(x, c_2)/2$, donde $x - h < c_1 < x + h$ y $t < c_2 < t + h$. Al sustituir en la ecuación de calor en el punto (x_i, t_j) se obtiene

$$\frac{D}{h^2}(w_{i+1,j} - 2w_{ij} + w_{i-1,j}) \approx \frac{1}{k}(w_{i,j+1} - w_{ij}), \tag{8.6}$$

con los errores de truncamiento local dado por $O(k) + O(h^2)$. Al igual que en el estudio de las ecuaciones diferenciales ordinarias, los errores de truncamiento proporcionan una buena imagen de los errores totales, siempre que el método sea estable. La estabilidad del método de las diferencias finitas se investigará después de presentar los detalles de implementación.

Observe que las condiciones iniciales y de frontera dan cantidades conocidas w_{i0} para $i = 0, \ldots, M$ y w_{0j} y w_{Mj} para $j = 0, \ldots, N$, lo que corresponde a la parte inferior y los lados del rectángulo de la figura 8.1. La versión discreta (8.6) puede resolverse dando un paso adelante en el tiempo. Reordene (8.6) como

$$\begin{aligned} w_{i,j+1} &= w_{ij} + \frac{Dk}{h^2}(w_{i+1,j} - 2w_{ij} + w_{i-1,j}) \\ &= \sigma w_{i+1,j} + (1 - 2\sigma)w_{ij} + \sigma w_{i-1,j}, \end{aligned} \tag{8.7}$$

donde se ha definido $\sigma = Dk/h^2$. La figura 8.2 muestra el conjunto de puntos de malla que participan en (8.7), con frecuencia llamados la **plantilla** del método.

El método de la diferencia hacia adelante (8.7) es **explícito**, puesto que existe una manera de determinar los nuevos valores (en el sentido del tiempo) directamente a partir de los valores antes

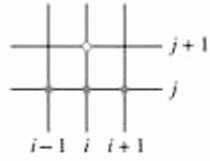


Figura 8.2 Plantilla para el método de la diferencia hacia adelante. El círculo no relleno representa $w_{i,j+1}$, que puede determinarse mediante (8.7) a partir de los valores $w_{i-1,j}$, w_{ij} y $w_{i+1,j}$ en los círculos rellenos.

conocidos. Un método que no sea explícito se llama **implícito**. La plantilla del método demuestra que este método es explícito. En términos matriciales, pueden obtenerse los valores de $w_{i,j+1}$ en el tiempo t_{j+1} al calcular una multiplicación matricial $w_{j+1} = Aw_j + s_j$, o bien

$$\begin{bmatrix} w_{1,j+1} \\ \vdots \\ w_{m,j+1} \end{bmatrix} = \begin{bmatrix} 1-2\sigma & \sigma & 0 & \cdots & 0 \\ \sigma & 1-2\sigma & \sigma & \ddots & \vdots \\ 0 & \sigma & 1-2\sigma & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \sigma \\ 0 & \cdots & 0 & \sigma & 1-2\sigma \end{bmatrix} \begin{bmatrix} w_{1j} \\ \vdots \\ w_{mj} \end{bmatrix} + \sigma \begin{bmatrix} w_{0,j} \\ 0 \\ \vdots \\ 0 \\ w_{m+1,j} \end{bmatrix}. \tag{8.8}$$

Aquí, la matriz A es de $m \times m$, donde $m = M - 1$. El vector s_j de la derecha representa las condiciones laterales impuestas por el problema, en este caso la temperatura en los extremos de la placa.

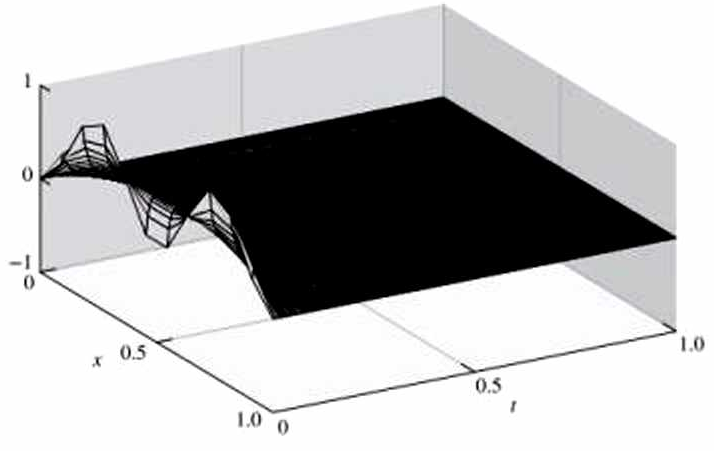
La solución se reduce a iterar una fórmula matricial, lo que permite rellenar los círculos vacíos en la figura 8.1, renglón por renglón. La iteración de la fórmula matricial $w_{j+1} = Aw_j + s_j$ es similar a los métodos iterativos para sistemas lineales descritos en el capítulo 2. Allí se aprendió que la convergencia de la iteración depende de los valores propios de la matriz. En la situación actual se tiene interés en los valores propios para analizar la magnificación del error.

Considere la ecuación de calor para $D = 1$, con la condición inicial $f(x) = \text{sen}^2\, 2\pi x$ y condiciones de frontera $u(0, t) = u(1, t) = 0$ para todo tiempo t. El código de MATLAB para llevar a cabo el cálculo en (8.8) se da en el programa 8.1.

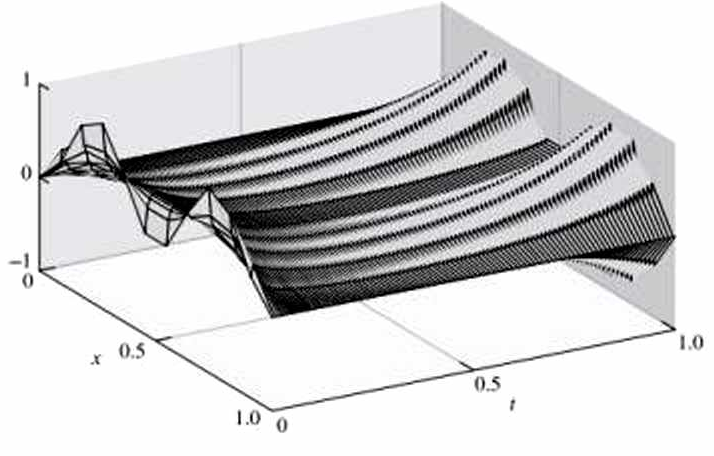
```
% Programa 8.1 Método de la diferencia hacia adelante para la ecuación de calor
% entrada: intervalo de espacio [xl, xr], intervalo de tiempo [yb, yt],
%          número de pasos de espacio M, número de pasoas de tiempo N
% salida: solución w
% Uso de ejemplo: w =heatfd(0,1,0,1,10,250)
function w=heatfd(xl,xr,yb,yt,M,N)
f=@(x) sin(2*pi*x).^2;
l=@(t) 0*t;
r=@(t) 0*t;
D=1;                                % coeficiente de difusión
h=(xr-xl)/M; k=(yt-yb)/N; m=M-1; n=N;
sigma=D*k/(h*h);
a=diag(1-2*sigma*ones(m,1))+diag(sigma*ones(m-1,1),1);
a=a+diag(sigma*ones(m-1,1),-1);     % define la matriz a
lside=l(yb+(0:n)*k); rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';              % condiciones iniciales
for j=1:n
  w(:,j+1)=a*w(:,j)+sigma*[lside(j);zeros(m-2,1);rside(j)];
end
w=[lside;w;rside];                  % adjunta conds de frontera
x=(0:m+1)*h;t=(0:n)*k;
mesh(x,t,w')                        % gráfica 3D de la solución w
view(60,30);axis([xl xr yb yt -1 1])
```

Los picos de temperatura iniciales deben suavizarse con el tiempo, produciendo una gráfica como la que se muestra en la figura 8.3(a). En esa gráfica, se usan las fórmulas (8.8) con tamaños de paso $h = 0.1$ a lo largo de la placa y $k = 0.004$ en el tiempo. El método de las diferencias hacia adelante explícito (8.7) da una solución aproximada en la figura 8.3(a), que muestra el flujo suave del calor hasta cerca del equilibrio después de menos de una unidad de tiempo. Esto corresponde a que la temperatura de la barra $u \to 0$ cuando $t \to \infty$.

En la figura 8.3(b) se usa un paso de tiempo un poco más grande $k > .005$. Al principio, los picos de calor comienzan a desvanecerse tal como se esperaba, pero después de un tiempo, los pequeños errores en la aproximación son magnificados por el método de las diferencias hacia adelante, haciendo que la solución se aleje del equilibrio correcto en cero. Éste es un artificio del proceso de solución, una señal de que el método es inestable. Si se permite que la simulación continúe, estos errores crecen sin límite. Por lo tanto, es necesario mantener el paso de tiempo k lo suficientemente pequeño para asegurar la convergencia.



(a)



(b)

Figura 8.3 Aproximación a la ecuación de calor (8.2) por el método de las diferencias finitas hacia adelante del programa 8.1. El parámetro de difusión es $D = 1$, con la condición inicial $f(x) = \text{sen}^2 2\pi x$. El tamaño de paso en el espacio es $h = 0.1$. El método de las diferencias hacia adelante es (a) estable para el paso de tiempo $k = 0.0040$, (b) inestable para $k > .005$.

8.1.2 Análisis de estabilidad del método de las diferencias hacia adelante

El comportamiento extraño mostrado por la anterior simulación de la ecuación del calor conduce a la raíz del problema. En la resolución de ecuaciones diferenciales parciales mediante el método de las diferencias hacia adelante, el control de la magnificación del error para tamaños de paso prácticos resulta ser un aspecto crucial en la solución eficiente de problemas.

Al igual que en el caso de las EDO estudiado con anterioridad, hay dos tipos de error involucrados. La propia discretización contribuye con errores de truncamiento debido a las aproximaciones a las derivadas. El tamaño de estos errores se conoce a partir de la fórmula del error de Taylor, como en (8.4) y (8.5). Además, existe una magnificación de los errores debida al método mismo. Para investigar esta magnificación, es necesario mirar más de cerca lo que está haciendo el método de las diferencias finitas. El **análisis de estabilidad de Von Neumann** mide la magnificación, o amplificación, del error. Para un método estable, los tamaños de paso deben elegirse de modo que el factor de magnificación no sea mayor que 1.

Sea y_j la solución exacta que satisface $y_{j+1} = Ay_j + s_j$ en la ecuación (8.8), y sea w_j la aproximación calculada, que satisface $w_{j+1} = Aw_j + s_j$. La diferencia $e_j = w_j - y_j$ satisface

$$\begin{aligned} e_j = w_j - y_j &= Aw_{j-1} + s_{j-1} - (Ay_{j-1} + s_{j-1}) \\ &= A(w_{j-1} - y_{j-1}) \\ &= Ae_{j-1}. \end{aligned} \tag{8.9}$$

El teorema A.7 del apéndice A dice que, para asegurar que los errores e_j no se amplifiquen, debe requerirse el radio espectral $\rho(A) < 1$. Este requisito pone límites a los tamaños de paso h y k del método de las diferencias finitas. Para determinar estos límites, se requiere información sobre los valores propios de las matrices simétricas tridiagonales.

Considere el ejemplo fundamental siguiente:

$$T = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 1 & -1 & \ddots & \vdots \\ 0 & -1 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}. \tag{8.10}$$

TEOREMA 8.1 Los vectores propios de la matriz T en (8.10) son los vectores v_j en (8.12) para $j = 1, \ldots, m$ con los valores propios correspondientes $\lambda_j = 1 - 2\cos j\pi/(m+1)$. ■

Demostración. Primero, recuerde de la trigonometría la fórmula de adición de senos. Para cualquier entero i y número real x, pueden sumarse las dos ecuaciones

$$\begin{aligned} \operatorname{sen}(i-1)x &= \operatorname{sen} ix \cos x - \cos ix \operatorname{sen} x \\ \operatorname{sen}(i+1)x &= \operatorname{sen} ix \cos x + \cos ix \operatorname{sen} x \end{aligned}$$

para obtener

$$\operatorname{sen}(i-1)x + \operatorname{sen}(i+1)x = 2\operatorname{sen} ix \cos x,$$

que puede reescribirse como

$$-\operatorname{sen}(i-1)x + \operatorname{sen} ix - \operatorname{sen}(i+1)x = (1 - 2\cos x)\operatorname{sen} ix. \tag{8.11}$$

La ecuación (8.11) puede verse como un hecho acerca de la multiplicación de matrices por T. Fije un entero j y defina el vector

$$v_j = \left[\text{sen}\frac{j\pi}{m+1}, \text{sen}\frac{2\pi j}{m+1}, \ldots, \text{sen}\frac{m\pi j}{m+1}\right]. \tag{8.12}$$

Observe el patrón: las entradas son de la forma sen ix como en (8.11), donde $x = \pi j/(m+1)$. Ahora (8.11) implica que

$$Tv_j = \left(1 - 2\cos\frac{\pi j}{m+1}\right)v_j \tag{8.13}$$

para $j = 1, \ldots, m$, que muestra los m vectores propios y valores propios. ❐

Para una j que inicia en $m+1$, los vectores v_j se repiten, por lo que hay exactamente m vectores propios, como se esperaba (vea el ejercicio 6). Todos los valores propios de T se encuentran entre -1 y 3.

El teorema 8.1 puede explotarse para encontrar los valores propios de cualquier matriz simétrica tridiagonal cuya diagonal principal y superdiagonal sean constantes. Por ejemplo, la matriz A (8.8) puede expresarse como $A = -\sigma T + (1-\sigma)I$. De acuerdo con el teorema 8.1, los valores propios de A son $-\sigma(1 - 2\cos \pi j/(m+1)) + 1 - \sigma = 2\sigma(\cos \pi j/(m+1) - 1) + 1$ para $j = 1, \ldots, m$. Aquí se ha utilizado el hecho de que los valores propios de una matriz que cambian mediante la adición de un múltiplo de la matriz identidad se modificarán mediante el mismo múltiplo.

Ahora puede aplicarse el criterio del teorema A.7. Dado que $-2 < \cos x - 1 < 0$ para los argumentos dados $x = \pi j/(m+1)$, donde $1 \le j \le m$, los valores propios de A pueden ir desde $-4\sigma + 1$ hasta 1. Suponiendo que el coeficiente de difusión $D > 0$, es necesario restringir $\sigma < 1/2$ para asegurar que los valores absolutos de todos los valores propios de A sean menores que 1 —es decir, que $\rho(A) < 1$.

El resultado del análisis de estabilidad de Von Neumann puede establecerse de la siguiente manera:

TEOREMA 8.2 Sea h el paso de espacio y k el paso de tiempo para el método de la diferencia hacia adelante aplicado a la ecuación del calor (8.2) con $D > 0$. Si $\frac{Dk}{h^2} < \frac{1}{2}$, el método de las diferencias hacia adelante es estable. ■

El análisis confirma lo que se observó en la figura 8.3. Por definición, $\sigma = Dk/h^2 = (1)(0.004)/(0.1)^2 = 0.4 < 1/2$ en la figura 8.3(a), mientras que k es ligeramente mayor que 0.005 en la figura 8.3(b), lo que conduce a $\sigma > (1)(0.005)/(0.1)^2 = 1/2$ y una notable magnificación del error. El método de las diferencias hacia adelante explícito se llama **condicionalmente estable**, porque su estabilidad depende de la elección de los tamaños de paso.

8.1.3 Método de la diferencia hacia atrás

Como alternativa, el método de la diferencia finita puede rehacerse con mejores propiedades de magnificación del error utilizando un método implícito. Al igual que antes, se sustituye u_{xx} en la ecuación del calor con la fórmula de las diferencias centradas, pero esta vez se usa la fórmula de las diferencias hacia atrás

$$u_t = \frac{1}{k}(u(x,t) - u(x,t-k)) + \frac{k}{2}u_{tt}(x, c_0),$$

donde $t - k < c_0 < t$, para aproximar u_t. La motivación aquí se sigue del capítulo 6, donde se han mejorado las características de estabilidad del método de Euler (explícito) utilizando el método de Euler hacia atrás (implícito), que utiliza una diferencia hacia atrás.

Al sustituir las fórmulas de diferencias en la ecuación del calor en el punto (x_i, t_j) resulta

$$\frac{1}{k}(w_{ij} - w_{i,j-1}) = \frac{D}{h^2}(w_{i+1,j} - 2w_{ij} + w_{i-1,j}), \tag{8.14}$$

con error local de truncamiento de $O(k) + O(h^2)$, el mismo error que da el método de la diferencia hacia adelante. La ecuación (8.14) puede reordenarse como

$$-\sigma w_{i+1,j} + (1+2\sigma)w_{ij} - \sigma w_{i-1,j} = w_{i,j-1},$$

con $\sigma = Dk/h^2$ y escribirse como la ecuación matricial de $m \times m$

$$\begin{bmatrix} 1+2\sigma & -\sigma & 0 & \cdots & 0 \\ -\sigma & 1+2\sigma & -\sigma & \ddots & \vdots \\ 0 & -\sigma & 1+2\sigma & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\sigma \\ 0 & \cdots & 0 & -\sigma & 1+2\sigma \end{bmatrix} \begin{bmatrix} w_{1j} \\ \vdots \\ w_{mj} \end{bmatrix} = \begin{bmatrix} w_{1,j-1} \\ \vdots \\ w_{m,j-1} \end{bmatrix} + \sigma \begin{bmatrix} w_{0j} \\ 0 \\ \vdots \\ 0 \\ w_{m+1,j} \end{bmatrix}. \tag{8.15}$$

Con pequeñas modificaciones, el programa de 8.1 puede adaptarse para seguir el **método de la diferencia hacia atrás**.

```
% Programa 8.2 Método de la diferencia hacia atrás para la ecuación de calor
% entrada: intervalo de espacio [xl,xr], intervalo de tiempo [yb, yt],
%          número de pasos de espacio M, número de pasos de tiempo N
% salida: solución w
% Uso de ejemplo: w=heatbd (0,1,0,1,10,10)
function w=heatbd(xl,xr,yb,yt,M,N)
f=@(x) sin(2*pi*x).^2;
l=@(t) 0*t;
r=@(t) 0*t;
D=1;                                    % coeficiente de difusión
h=(xr-xl)/M; k=(yt-yb)/N; m=M-1; n=N;
sigma=D*k/(h*h);
a=diag(1+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1);
a=a+diag(-sigma*ones(m-1,1),-1);      % define la matriz a
lside=l(yb+(0:n)*k); rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';                  %  condiciones iniciales
for j=1:n
  w(:,j+1)=a\(w(:,j)+sigma*[lside(j);zeros(m-2,1);rside(j)]);
end
w=[lside;w;rside];                      % adjunta conds de frontera
x=(0:m+1)*h;t=(0:n)*k;
mesh(x,t,w')                            % gráfica 3D de la solución w
view(60,30);axis([xl xr yb yt -1 2])
```

► **EJEMPLO 8.1** Aplique el método de la diferencia hacia atrás a la ecuación de calor

$$\begin{cases} u_t = u_{xx} \text{ para toda } 0 \le x \le 1, t \ge 0 \\ u(x,0) = \text{sen}^2 2\pi x \text{ para toda } 0 \le x \le 1 \\ u(0,t) = 0 \text{ para toda } t \ge 0 \\ u(1,t) = 0 \text{ para toda } t \ge 0 \end{cases}$$

Usando tamaños de paso $h = k = 0.1$, se llega a la solución aproximada que se muestra en la figura 8.4. Compare esto con el desempeño del método de la diferencia hacia adelante de la figura 8.3, donde $h = 0.1$ y k debe ser mucho menor para evitar la inestabilidad. ◄

¿Cuál es la razón de la mejora del rendimiento del método implícito? El análisis de estabilidad para el método de la diferencia hacia atrás procede de manera similar al caso explícito. El método de la diferencia hacia atrás (8.15) puede verse como la iteración matricial

$$w_j = A^{-1} w_{j-1} + b,$$

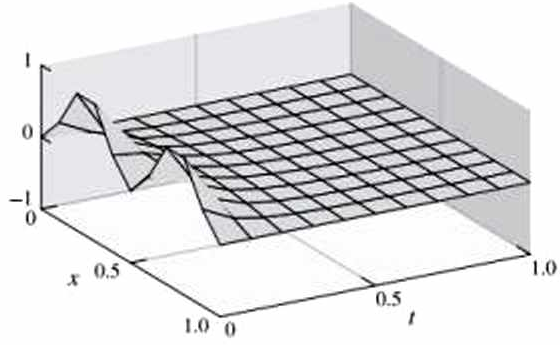


Figura 8.4 Solución aproximada del ejemplo 8.1 mediante el método de las diferencias hacia atrás. El coeficiente de difusión es $D = 1$ y los tamaños de paso son $h = 0.1$ y $k = 0.1$.

donde

$$A = \begin{bmatrix} 1+2\sigma & -\sigma & 0 & \cdots & 0 \\ -\sigma & 1+2\sigma & -\sigma & \ddots & \vdots \\ 0 & -\sigma & 1+2\sigma & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\sigma \\ 0 & \cdots & 0 & -\sigma & 1+2\sigma \end{bmatrix}. \tag{8.16}$$

Al igual que en el análisis de estabilidad de Von Neumann para el método de la diferencia hacia adelante, las cantidades relevantes son los valores propios de A^{-1}. Dado que $A = \sigma T + (1 + \sigma)I$, el lema 8.1 implica que los valores propios de A son

$$\sigma\left(1 - 2\cos\frac{\pi j}{m+1}\right) + 1 + \sigma = 1 + 2\sigma - 2\sigma\cos\frac{\pi j}{m+1},$$

y los valores propios de A^{-1} son recíprocos. Para asegurarse de que el radio espectral de A^{-1} es menor que 1, es necesario que

$$|1 + 2\sigma(1 - \cos x)| > 1, \tag{8.17}$$

lo cual es verdadero para todos las σ, puesto que $1 - \cos x > 0$ y $\sigma = Dk/h^2 > 0$. Por lo tanto, el método implícito es estable para toda σ y, por ende, para todas las opciones de tamaños de paso h y k, que es la definición de **incondicionalmente estable**. Entonces, el tamaño de paso puede hacerse mucho más grande, limitado sólo por consideraciones del error de truncamiento local.

TEOREMA 8.3 Sea h el paso de espacio y k el paso de tiempo para el método de las diferencias hacia atrás aplicado a la ecuación del calor (8.2) con $D > 0$. Para cualquier h, k, el método de la diferencia hacia atrás es estable. ■

► EJEMPLO 8.2 Aplique el método de las diferencias hacia atrás para resolver la ecuación de calor

$$\begin{cases} u_t = 4u_{xx} \text{ para toda } 0 \le x \le 1, 0 \le t \le 1 \\ u(x,0) = e^{-x/2} \text{ para toda } 0 \le x \le 1 \\ u(0,t) = e^t \text{ para toda } 0 \le t \le 1 \\ u(1,t) = e^{t-1/2} \text{ para toda } 0 \le t \le 1 \end{cases}.$$

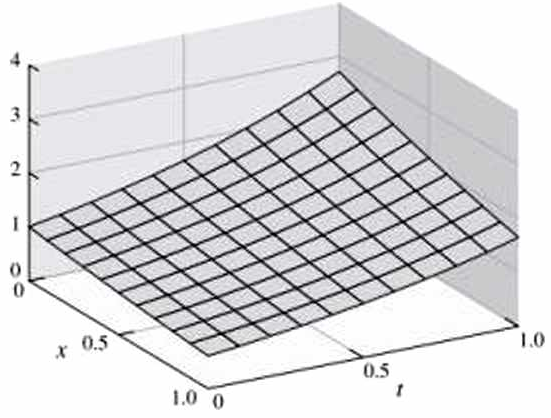


Figura 8.5 Solución aproximada del ejemplo 8.2 mediante el método de las diferencias hacia atrás. Los tamaños de paso son $h = 0.1$, $k = 0.1$.

Compruebe que la solución correcta es $u(x, t) = e^{t-x/2}$. Si se establece $h = k = 0.1$ y $D = 4$, entonces $\sigma = Dk/h^2 = 40$. La matriz A es de 9×9, y en cada uno de los 10 pasos de tiempo, (8.15) se resuelve mediante la eliminación gaussiana. La solución se muestra en la figura 8.5. ◄

Como el método de las diferencias hacia atrás es estable para cualquier tamaño de paso, puede discutirse el tamaño de los errores de truncamiento que se cometen por la discretización en el espacio y el tiempo. Los errores por la discretización en el tiempo son de orden $O(k)$, y los errores por la discretización en el tiempo son de orden $O(h^2)$. Esto significa que, para tamaños de paso pequeños $h \approx k$, dominará el error del paso de tiempo, dado que $O(h^2)$ será insignificante comparado con $O(k)$. En otras palabras, el error en el método de las diferencias hacia atrás puede describirse a grandes rasgos como $O(k) + O(h^2) \approx O(k)$.

Para demostrar esta conclusión, se utilizó el método de las diferencias finitas implícito a fin de producir las soluciones del ejemplo 8.2 para $h = 0.1$ fija y una serie decreciente de k. La tabla adjunta muestra que el error medido en $(x, t) = (0.5, 1)$ disminuye linealmente con k; es decir, cuando k se reduce a la mitad, ocurre lo mismo con el error. Si el tamaño de h se redujera, la cantidad de cálculo aumentaría, pero los errores para una k dada serían prácticamente iguales.

h	k	$u(0.5, 1)$	$w(0.5, 1)$	error
0.10	0.10	2.11700	2.12015	0.00315
0.10	0.05	2.11700	2.11861	0.00161
0.10	0.01	2.11700	2.11733	0.00033

Las condiciones de frontera que se han estado aplicando a la ecuación del calor se denominan condiciones de frontera de **Dirichlet**. Éstas especifican los valores de la solución $u(x, t)$ en la frontera del dominio de la solución. En el último ejemplo, las condiciones de Dirichlet $u(0, t) = e^t$ y $u(1, t) = e^{t-1/2}$ establecen los valores de temperatura requeridos en las fronteras del dominio $[0, 1]$. Si se considera la ecuación de calor como un modelo de conducción de calor, esto corresponde a mantener la temperatura en la frontera a un nivel prescrito.

Un tipo alternativo de condición de frontera corresponde a una frontera aislada. Aquí la temperatura no se especifica, pero se supone que el calor no puede conducirse a través de la frontera. En general, una condición de frontera de **Neumann** especifica el valor de una derivada en la frontera. Por ejemplo, en el dominio $[a, b]$, que requiere $u_x(a, t) = u_x(b, t) = 0$ para toda t corresponde a una frontera aislada, o sin flujo. En general, las condiciones de frontera que se establecen en cero se denominan condiciones de frontera **homogéneas**.

► EJEMPLO 8.3 Aplique el método de las diferencias hacia atrás para resolver la ecuación de calor con condiciones de frontera de Neumann homogéneas

$$\begin{cases} u_t = u_{xx} \text{ para toda } 0 \le x \le 1, 0 \le t \le 1 \\ u(x,0) = \text{sen}^2 2\pi x \text{ para toda } 0 \le x \le 1 \\ u_x(0,t) = 0 \text{ para toda } 0 \le t \le 1 \\ u_x(1,t) = 0 \text{ para toda } 0 \le t \le 1. \end{cases} \tag{8.18}$$

Del capítulo 5, recuerde la fórmula de segundo orden para la primera derivada

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + O(h^2). \tag{8.19}$$

Esta fórmula es útil para situaciones en las que no están disponibles los valores de la función a ambos lados de x. Con las condiciones de frontera de Neumann se presenta precisamente esta posición. Por lo tanto, se utilizarán las aproximaciones de segundo orden

$$u_x(0,t) \approx \frac{-3u(0,t) + 4u(0+h,t) - u(0+2h,t)}{2h}$$
$$u_x(1,t) \approx \frac{-u(1-2h,t) + 4u(1-h,t) - 3u(1,t)}{-2h}$$

para las condiciones de Neumann. El establecimiento de estas aproximaciones a la derivada en cero se traduce en las fórmulas

$$-3w_0 + 4w_1 - w_2 = 0$$
$$-w_{M-2} + 4w_{M-1} - 3w_M = 0$$

que se deben añadir a las partes de las ecuaciones que no están en la frontera. Para efectos de conteo, observe que al pasar de las condiciones de frontera de Dirichlet a las de Neumann, la novedad es que deben resolverse los dos puntos en la frontera w_0 y w_M. Esto significa que mientras que para Dirichlet, el tamaño de la matriz en el método de la diferencia hacia atrás es $m \times m$ donde $m = M - 1$ cuando se pasa a las condiciones de frontera de Neumann, $m = M + 1$, y la matriz es ligeramente más grande. Estos detalles pueden verse en el siguiente programa 8.3. La primera y última ecuaciones se reemplazan por las condiciones de Neumann.

```
% Programa 8.3 Método de la diferencia hacia atrás para la ecuación de calor
%              con condiciones de frontera de Neumann
% entrada: intervalo de espacio [xl, xr], intervalo de tiempo [yb, yt],
%        número de pasos de espacio M, número de pasos de tiempo N
% salida: solución w
% Uso de ejemplo: w=heatbdn(0,1,0,1,20,20)
function w=heatbdn(xl,xr,yb,yt,M,N)
f=@(x) sin(2*pi*x).^2;
D=1;                                % coeficiente de difusión
h=(xr-xl)/M; k=(yt-yb)/N; m=M+1; n=N;
sigma=D*k/(h*h);
a=diag(1+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1);
a=a+diag(-sigma*ones(m-1,1),-1);  % define la matriz a
a(1,:)=[-3 4 -1 zeros(1,m-3)];    % condiciones de Neumann
a(m,:)=[zeros(1,m-3) -1 4 -3];
w(:,1)=f(xl+(0:M)*h)';            % condiciones iniciales
for j=1:n
  b=w(:,j);b(1)=0;b(m)=0;
  w(:,j+1)=a\b;
end
x=(0:M)*h;t=(0:n)*k;
mesh(x,t,w')                       % gráfica 3D de la solución w
view(60,30);axis([xl xr yb yt -1 1])
```

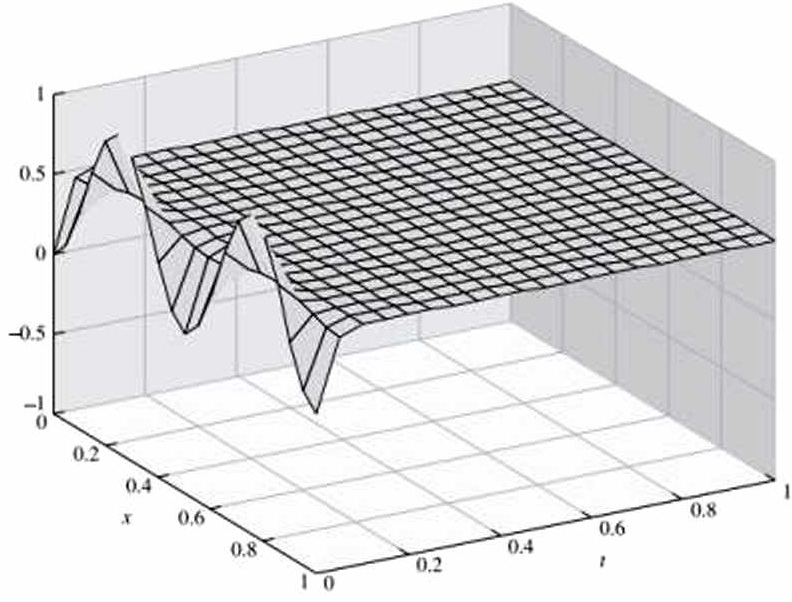


Figura 8.6 Solución aproximada del problema de Neumann (8.18) por el método de las diferencias hacia atrás. Los tamaños de paso son $h = k = 0.05$.

La figura 8.6 muestra los resultados del programa 8.3 con las condiciones de Neumann, los valores de frontera ya no están fijos en cero y la solución flota para satisfacer el valor de los datos iniciales, ésta se promedia por difusión y tiene un valor de 1/2. ◀

8.1.4 Método de Crank-Nicolson

Hasta ahora, los métodos para la ecuación de calor ha consistido en un método explícito que a veces es estable y un método implícito que es siempre estable. Ambos tienen errores de tamaño $O(k + h^2)$ cuando se estabilizan. El tamaño del paso de tiempo k debe ser bastante pequeño para obtener una buena precisión.

El **método de Crank-Nicolson** es una combinación de los métodos implícitos y explícitos, es incondicionalmente estable, y tiene error $O(h^2) + O(k^2)$. Las fórmulas son un poco más complicadas, pero valen la pena debido a la mayor precisión y estabilidad que garantizan.

Crank-Nicolson utiliza la fórmula de las diferencias hacia atrás para la derivada en el tiempo, y una combinación ponderada uniformemente de las aproximaciones por las diferencias hacia adelante y hacia atrás para el resto de la ecuación. Por ejemplo, en la ecuación de calor (8.2) se reemplaza u_t con la fórmula de la diferencia hacia atrás

$$\frac{1}{k}(w_{ij} - w_{i,j-1})$$

y u_{xx} con la diferencia mixta

$$\frac{1}{2}\left(\frac{w_{i+1,j} - 2w_{ij} + w_{i-1,j}}{h^2}\right) + \frac{1}{2}\left(\frac{w_{i+1,j-1} - 2w_{i,j-1} + w_{i-1,j-1}}{h^2}\right).$$

De nuevo, al ajustar $\sigma = Dk/h^2$, es posible reordenar la aproximación a la ecuación del calor en la forma

$$2w_{ij} - 2w_{i,j-1} = \sigma[w_{i+1,j} - 2w_{ij} + w_{i-1,j} + w_{i+1,j-1} - 2w_{i,j-1} + w_{i-1,j-1}],$$

o bien

$$-\sigma w_{i-1,j} + (2+2\sigma)w_{ij} - \sigma w_{i+1,j} = \sigma w_{i-1,j-1} + (2-2\sigma)w_{i,j-1} + \sigma w_{i+1,j-1},$$

lo que conduce a la plantilla que se muestra en la figura 8.7.

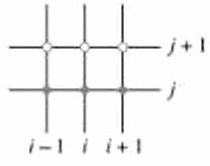


Figura 8.7 Puntos de la malla para el método de Crank-Nicolson. En cada paso de tiempo, los círculos no rellenos son las incógnitas del paso anterior.

Establezca $w_j = [w_{1j}, \ldots, w_{mj}]^T$. En forma matricial, el método de Crank-Nicolson es

$$Aw_j = Bw_{j-1} + \sigma(s_{j-1} + s_j),$$

donde

$$A = \begin{bmatrix} 2+2\sigma & -\sigma & 0 & \cdots & 0 \\ -\sigma & 2+2\sigma & -\sigma & \ddots & \vdots \\ 0 & -\sigma & 2+2\sigma & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\sigma \\ 0 & \cdots & 0 & -\sigma & 2+2\sigma \end{bmatrix},$$

$$B = \begin{bmatrix} 2-2\sigma & \sigma & 0 & \cdots & 0 \\ \sigma & 2-2\sigma & \sigma & \ddots & \vdots \\ 0 & \sigma & 2-2\sigma & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \sigma \\ 0 & \cdots & 0 & \sigma & 2-2\sigma \end{bmatrix},$$

y $s_j = [w_{0j}, 0, \ldots, 0, w_{m+1,j}]^T$. Si se aplica Crank-Nicolson a la ecuación de calor, se obtiene el resultado mostrado en la figura 8.8, para tamaños de paso $h = 0.1$ y $k = 0.1$. El código de MATLAB para el método se da en el programa 8.4.

```
% Programa 8.4 Método de Crank-Nicolson
%        con condiciones de frontera de Dirichlet
% entrada: intervalo de espacio [xl, xr], intervalo de tiempo [yb, yt],
%        número de pasos de espacio M, número de pasos de tiempo N
% salida: solución w
% Uso de ejemplo: w=crank(0,1,0,1,10,10)
function w=crank(xl,xr,yb,yt,M,N)
f=@(x) sin(2*pi*x).^2;
l=@(t) 0*t;
r=@(t) 0*t;
D=1;                                  % coeficiente de difusión
h=(xr-xl)/M;k=(yt-yb)/N;              % tamaños de paso
```

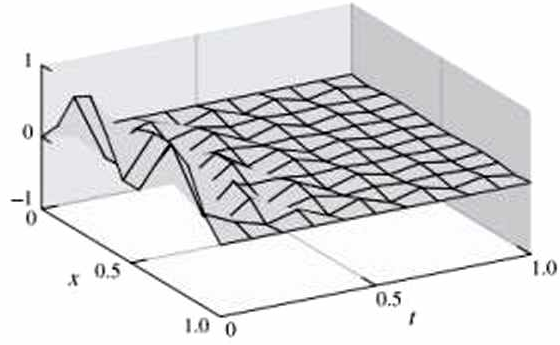


Figura 8.8 Solución aproximada de la ecuación de calor (8.2) calculada mediante el método de Crank-Nicolson. Tamaños de paso $h = 0.1, k = 0.1$.

```
sigma=D*k/(h*h); m=M-1; n=N;
a=diag(2+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1);
a=a+diag(-sigma*ones(m-1,1),-1);     % define la matriz tridiagonal a
b=diag(2-2*sigma*ones(m,1))+diag(sigma*ones(m-1,1),1);
b=b+diag(sigma*ones(m-1,1),-1);      % define la matriz tridiagonal b
lside=l(yb+(0:n)*k); rside=r(yb+(0:n)*k);
w(:,1)=f(xl+(1:m)*h)';               % condiciones iniciales
for j=1:n
    sides=[lside(j)+lside(j+1);zeros(m-2,1);rside(j)+rside(j+1)];
    w(:,j+1)=a\(b*w(:,j)+sigma*sides);
end
w=[lside;w;rside];
x=xl+(0:M)*h;t=yb+(0:N)*k;
mesh(x,t,w');
view (60,30); axis([xl xr yb yt -1 1])
```

Para investigar la estabilidad de Crank-Nicolson debe hallarse el radio espectral de la matriz $A^{-1}B$, para A y B que se dio en el párrafo anterior. Una vez más, la matriz en cuestión puede reescribirse en términos de T. Observe que $A = \sigma T + (2 + \sigma)I$ y $B = -\sigma T + (2 - \sigma)I$. Al multiplicar $A^{-1}B$ por el j-ésimo vector propio v_j de T se obtiene

$$\begin{aligned} A^{-1}Bv_j &= (\sigma T + (2+\sigma)I)^{-1}(-\sigma\lambda_j v_j + (2-\sigma)v_j) \\ &= \frac{1}{\sigma\lambda_j + 2 + \sigma}(-\sigma\lambda_j + 2 - \sigma)v_j, \end{aligned}$$

donde λ_j es el valor propio de T asociado con v_j. Los valores propios de $A^{-1}B$ son

$$\frac{-\sigma\lambda_j + 2 - \sigma}{\sigma\lambda_j + 2 + \sigma} = \frac{4 - (\sigma(\lambda_j + 1) + 2)}{\sigma(\lambda_j + 1) + 2} = \frac{4}{L} - 1, \tag{8.20}$$

donde $L = \sigma(\lambda_j + 1) + 2 > 2$, puesto que $\lambda_j > -1$. Por lo tanto, los valores propios (8.20) están entre -1 y 1. El método de Crank-Nicolson, al igual que el método de las diferencias finitas implícito, es incondicionalmente estable.

ANOTACIÓN

Convergencia Crank-Nicolson es un método de diferencias finitas conveniente para la ecuación de calor debido a su estabilidad incondicional (teorema 8.4) y la convergencia de segundo orden, que se muestra en (8.23). La deducción de un método de este tipo no es sencilla, debido a la primera derivada parcial u_t en la ecuación. En la ecuación de onda y la ecuación de Poisson descritas más adelante en este capítulo, sólo aparecen derivadas de segundo orden y es mucho más fácil encontrar métodos estables de segundo orden.

TEOREMA 8.4 El método de Crank-Nicolson aplicado a la ecuación del calor (8.2) con $D > 0$ es estable para cualesquiera tamaños de paso $h, k > 0$. ■

Para terminar esta sección, se deduce el error de truncamiento para el método de Crank-Nicolson, que es $O(h^2) + O(k^2)$. Además de su estabilidad incondicional, esto hace que el método sea superior en lo general a los métodos de las diferencias hacia adelante y hacia atrás para la ecuación de calor $u_t = Du_{xx}$.

Para la deducción del método se requieren las siguientes cuatro ecuaciones. Se supone la existencia de derivadas más altas de la solución u, según se requieran. Del ejercicio 5.1.24, se tiene la fórmula de las diferencias hacia atrás

$$u_t(x,t) = \frac{u(x,t) - u(x,t-k)}{k} + \frac{k}{2}u_{tt}(x,t) - \frac{k^2}{6}u_{ttt}(x,t_1), \tag{8.21}$$

donde $t - k < t_1 < t$, suponiendo que las derivadas parciales existen. Al expandir u_{xx} en serie de Taylor de la variable t se obtiene

$$u_{xx}(x,t-k) = u_{xx}(x,t) - ku_{xxt}(x,t) + \frac{k^2}{2}u_{xxtt}(x,t_2),$$

donde $t - k < t_2 < t$, o bien

$$u_{xx}(x,t) = u_{xx}(x,t-k) + ku_{xxt}(x,t) - \frac{k^2}{2}u_{xxtt}(x,t_2). \tag{8.22}$$

La fórmula de la diferencia centrada para las segundas derivadas da tanto

$$u_{xx}(x,t) = \frac{u(x+h,t) - 2u(x,t) + u(x-h,t)}{h^2} + \frac{h^2}{12}u_{xxxx}(x_1,t) \tag{8.23}$$

como

$$u_{xx}(x,t-k) = \frac{u(x+h,t-k) - 2u(x,t-k) + u(x-h,t-k)}{h^2} + \frac{h^2}{12}u_{xxxx}(x_2,t-k), \tag{8.24}$$

donde x_1 y x_2 se encuentran entre x y $x + h$.

Sustituyendo las últimas cuatro ecuaciones en la ecuación de calor

$$u_t = D\left(\frac{1}{2}u_{xx} + \frac{1}{2}u_{xx}\right),$$

donde se ha dividido el lado derecho en dos. La estrategia consiste en reemplazar el lado izquierdo usando (8.21), la primera mitad del lado derecho por (8.23) y la segunda mitad del lado derecho por (8.22) en combinación con (8.24). Esto da como resultado

$$\frac{u(x,t)-u(x,t-k)}{k}+\frac{k}{2}u_{tt}(x,t)-\frac{k^2}{6}u_{ttt}(x,t_1)$$
$$=\frac{1}{2}D\left[\frac{u(x+h,t)-2u(x,t)+u(x-h,t)}{h^2}+\frac{h^2}{12}u_{xxxx}(x_1,t)\right]$$
$$+\frac{1}{2}D\left[ku_{xxt}(x,t)-\frac{k^2}{2}u_{xxtt}(x,t_2)\right.$$
$$\left.+\frac{u(x+h,t-k)-2u(x,t-k)+u(x-h,t-k)}{h^2}+\frac{h^2}{12}u_{xxxx}(x_2,t-k)\right].$$

Por lo tanto, el error asociado con la igualación de los cocientes de las diferencias es el residuo

$$-\frac{k}{2}u_{tt}(x,t)+\frac{k^2}{6}u_{ttt}(x,t_1)+\frac{Dh^2}{24}[u_{xxxx}(x_1,t)+u_{xxxx}(x_2,t-k)]$$
$$+\frac{Dk}{2}u_{xxt}(x,t)-\frac{Dk^2}{4}u_{xxtt}(x,t_2).$$

Esta expresión puede simplificarse usando el hecho de que $u_t = Du_{xx}$. Por ejemplo, observe que $Du_{xxt} = (Du_{xx})_t = u_{tt}$, con lo cual se cancelan el primero y cuarto términos de la expresión del error. El error de truncamiento es

$$\frac{k^2}{6}u_{ttt}(x,t_1)-\frac{Dk^2}{4}u_{xxtt}(x,t_2)+\frac{Dh^2}{24}[u_{xxxx}(x_1,t)+u_{xxxx}(x_2,t-k)]$$
$$=\frac{k^2}{6}u_{ttt}(x,t_1)-\frac{k^2}{4}u_{ttt}(x,t_2)+\frac{h^2}{24D}[u_{tt}(x_1,t)+u_{tt}(x_2,t-k)].$$

Una expansión de Taylor en la variable t resulta en

$$u_{tt}(x_2,t-k)=u_{tt}(x_2,t)-ku_{ttt}(x_2,t_4),$$

con lo que el error de truncamiento es igual a $O(h^2) + O(k^2)$ + los términos de orden superior. Se llega a la conclusión de que el método de Crank-Nicolson es de segundo orden e incondicionalmente estable para la ecuación de calor.

Para ilustrar la convergencia rápida de Crank-Nicolson, regrese a la ecuación del ejemplo 8.2. Vea también los problemas de computadora 5 y 6 para explorar la razón de convergencia.

► EJEMPLO 8.4 Aplique el método de Crank-Nicolson a la ecuación de calor

$$\begin{cases} u_t = 4u_{xx} \text{ para toda } 0 \le x \le 1, 0 \le t \le 1 \\ u(x,0) = e^{-x/2} \text{ para toda } 0 \le x \le 1 \\ u(0,t) = e^t \text{ para toda } 0 \le t \le 1 \\ u(1,t) = e^{t-1/2} \text{ para toda } 0 \le t \le 1 \end{cases} \tag{8.25}$$

La tabla siguiente muestra el error convergencia $O(h^2) + O(k^2)$ predicho por el cálculo anterior. La solución correcta $u(x, t) = e^{t-x/2}$ evaluada en $(x, t) = (0.5, 1)$ es $u = e^{3/4}$. Observe que el error se reduce en un factor de 4 cuando los tamaños de paso h y k se reducen a la mitad. Compare los errores con la tabla del ejemplo 8.2.

h	k	$u(0.5, 1)$	$w(0.5, 1)$	error
0.10	0.10	2.11700002	2.11706765	0.00006763
0.05	0.05	2.11700002	2.11701689	0.00001687
0.01	0.01	2.11700002	2.11700069	0.00000067

◄

En resumen, se han presentado tres métodos numéricos para ecuaciones parabólicas utilizando la ecuación del calor como primer ejemplo. El método de las diferencias hacia adelante es el más

sencillo, el método de las diferencias hacia atrás es tan incondicionalmente estable como exacto y Crank-Nicolson es incondicionalmente estable y exacto de segundo orden, tanto en el espacio como en el tiempo. Aunque la ecuación de calor es representativa, existen una gran variedad de ecuaciones parabólicas para las que estos métodos son aplicables.

Un área de aplicación importante para las ecuaciones difusivas se refiere a la evolución espacio-temporal de poblaciones biológicas. Considere una población (de bacterias, perros de la pradera, etcétera) que vive en una fracción o sustrato de terreno. Para comenzar en forma simple, la fracción será un segmento de recta $[0, L]$. Se utilizará una ecuación diferencial parcial para modelar $u(x, t)$, la densidad de población para cada punto $0 \leq x \leq L$. Las poblaciones tienden a actuar como el calor en el sentido de que se extienden, o difuminan, de las zonas de alta densidad a las áreas de densidad reducida siempre que es posible. También pueden crecer o morir, como se representa en el siguiente ejemplo.

► EJEMPLO 8.5 Considere la ecuación de difusión con crecimiento proporcional

$$\begin{cases} u_t = Du_{xx} + Cu \\ u(x, 0) = \operatorname{sen}^2 \frac{\pi}{L} x \text{ para toda } 0 \leq x \leq L \\ u(0, t) = 0 \text{ para toda } t \geq 0 \\ u(L, t) = 0 \text{ para toda } t \geq 0. \end{cases} \tag{8.26}$$

La densidad de población en el tiempo t y en la posición x se indica $u(x, t)$. El uso que se hace aquí de las condiciones de frontera de Dirichlet representa el supuesto de que la población no puede vivir fuera de la fracción $0 \leq x \leq L$. ◄

Quizás éste es el ejemplo más simple posible de una ecuación de **reacción-difusión**. El término de difusión Du_{xx} hace que la población se extienda a lo largo de la dirección x, mientras que el término de reacción Cu contribuye al crecimiento de la población a una tasa C. Debido a las condiciones de frontera de Dirichlet, la población es aniquilada a medida que llega a la frontera. En las ecuaciones de reacción-difusión, existe una competencia entre la tendencia de uniformidad de la difusión y la contribución al crecimiento de la reacción. Si la población sobrevive o avanza hacia la extinción depende de la competencia entre el parámetro de difusión D, la tasa de crecimiento C y el tamaño de la fracción L.

Se aplicará el método Crank-Nicolson al problema. El lado izquierdo de la ecuación se sustituye con

$$\frac{1}{k}(w_{ij} - w_{i,j-1})$$

y el lado derecho con la diferencia mezclada hacia adelante/hacia atrás

$$\frac{1}{2}\left(D\frac{w_{i+1,j} - 2w_{ij} + w_{i-1,j}}{h^2} + Cw_{ij}\right)$$
$$+\frac{1}{2}\left(D\frac{w_{i+1,j-1} - 2w_{i,j-1} + w_{i-1,j-1}}{h^2} + Cw_{i,j-1}\right).$$

Si se establece $\sigma = Dk/h^2$, es posible reordenar como

$$-\sigma w_{i-1,j} + (2 + 2\sigma - kC)w_{ij} - \sigma w_{i+1,j} = \sigma w_{i-1,j-1} + (2 - 2\sigma + kC)w_{i,j-1} + \sigma w_{i+1,j-1}.$$

Si se comparan con las ecuaciones de Crank-Nicolson para la ecuación de calor que se mostraron antes, sólo es necesario restar kC de las entradas diagonales de la matriz A y añadir kC a los elementos de la diagonal de la matriz B. Esto conduce a cambios en dos líneas del programa 8.4.

En la figura 8.9 se muestran los resultados de Crank-Nicolson aplicados a (8.26) con coeficiente de difusión $D = 1$, en la fracción $[0, 1]$. Para la elección $C = 9.5$, la densidad de población

original tiende a cero en el largo plazo. Para $C = 10$, la población crece. Aunque está más allá del alcance del análisis actual, puede demostrarse que el modelo de población sobrevive mientras

$$C > \pi^2 D/L^2. \tag{8.27}$$

En el caso actual, esto se traduce a $C > \pi^2$, que está entre 9.5 y 10, lo cual explica el resultado que se ve en la figura 8.9. En el modelado de poblaciones biológicas, con frecuencia la información se utiliza en sentido inverso: dada la tasa de crecimiento de la población y la velocidad de difusión conocidas, un ecologista que estudia la supervivencia de las especies podría desear saber cuál es la fracción más pequeña que es capaz de acoger a la población.

Los problemas de computadora 7 y 8 le piden al lector investigar este sistema de reacción-difusión a mayor profundidad. Las ecuaciones de reacción-difusión no lineales son un tema importante en la sección 8.4.

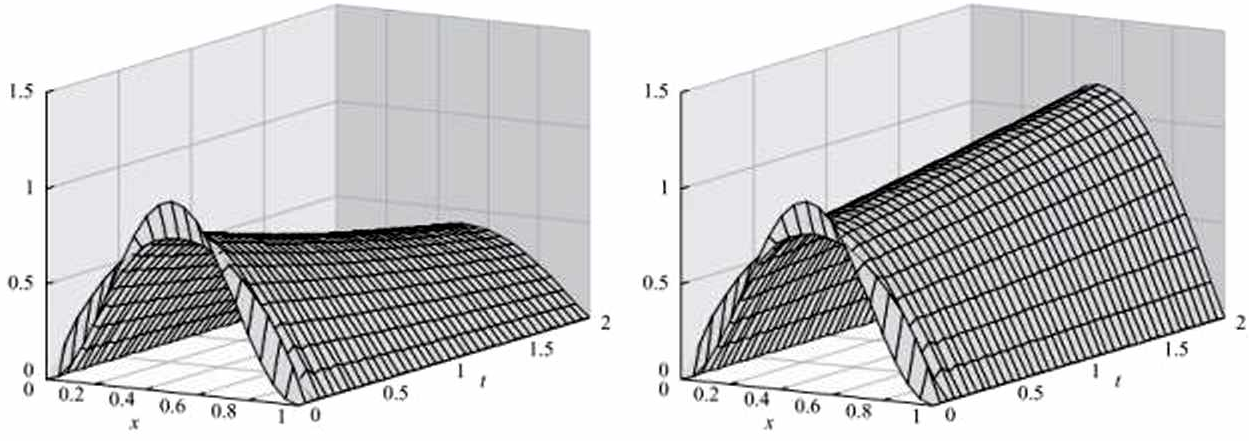


Figura 8.9 Soluciones aproximadas de la ecuación (8.26) calculadas mediante el método de Crank-Nicolson. Los parámetros son $D = 1, L = 1$, y los tamaños de paso utilizados son $h = k = 0.05$. (a) $C = 9.5$ (b) $C = 10$.

8.1 Ejercicios

1. Demuestre que las funciones $u(x,t) = e^{2t+x} + e^{2t-x}$, (b) $u(x,t) = e^{2t+x}$ son soluciones de la ecuación del calor $u_t = 2u_{xx}$ con las condiciones de frontera iniciales que se indican:

$$\text{(a)} \begin{cases} u(x,0) = 2\cosh x \text{ para } 0 \le x \le 1 \\ u(0,t) = 2e^{2t} \text{ para } 0 \le t \le 1 \\ u(1,t) = (e^2+1)e^{2t-1} \text{ para } 0 \le t \le 1 \end{cases} \qquad \text{(b)} \begin{cases} u(x,0) = e^x \text{ para } 0 \le x \le 1 \\ u(0,t) = e^{2t} \text{ para } 0 \le t \le 1 \\ u(1,t) = e^{2t+1} \text{ para } 0 \le t \le 1 \end{cases}$$

2. Demuestre que las funciones (a) $u(x,t) = e^{-\pi t} \operatorname{sen} \pi x$, (b) $u(x,t) = e^{-\pi t} \cos \pi x$ son soluciones de la ecuación de calor $\pi u_t = u_{xx}$ con las condiciones de frontera iniciales que se indican:

$$\text{(a)} \begin{cases} u(x,0) = \operatorname{sen} \pi x \text{ para } 0 \le x \le 1 \\ u(0,t) = 0 \text{ para } 0 \le t \le 1 \\ u(1,t) = 0 \text{ para } 0 \le t \le 1 \end{cases} \qquad \text{(b)} \begin{cases} u(x,0) = \cos \pi x \text{ para toda } 0 \le x \le 1 \\ u(0,t) = e^{-\pi t} \text{ para } 0 \le t \le 1 \\ u(1,t) = -e^{-\pi t} \text{ para } 0 \le t \le 1 \end{cases}$$

3. Demuestre que si $f(x)$ es un polinomio de tercer grado, entonces $u(x,t) = f(x) + ctf''(x)$ es una solución del problema de valor inicial $u_t = cu_{xx}$, $u(x,0) = f(x)$.

4. ¿El método de las diferencias hacia atrás es incondicionalmente estable para la ecuación de calor si $c < 0$? Explique su respuesta.

5. Verifique la ecuación del vector propio (8.13).

6. Demuestre que los vectores v_j diferentes de cero en (8.12), para todos los números enteros m, consisten solamente en m vectores distintos, hasta el cambio de signo.

8.1 Problemas de computadora

1. Resuelva la ecuación $u_t = 2u_{xx}$ para $0 \le x \le 1$, $0 \le t \le 1$, con las condiciones iniciales y de frontera siguientes, utilizando el método de las diferencias hacia adelante con tamaños de paso $h = 0.1$ y $k = 0.002$. Grafique la solución aproximada, empleando el comando de MATLAB mesh. ¿Qué sucede si se usa $k > 0.003$? Compare con las soluciones exactas del ejercicio 1.

(a)
$$\begin{cases} u(x,0) = 2\cosh x \text{ para } 0 \le x \le 1 \\ u(0,t) = 2e^{2t} \text{ para } 0 \le t \le 1 \\ u(1,t) = (e^2+1)e^{2t-1} \text{ para } 0 \le t \le 1 \end{cases}$$

(b)
$$\begin{cases} u(x,0) = e^x \text{ para } 0 \le x \le 1 \\ u(0,t) = e^{2t} \text{ para } 0 \le t \le 1 \\ u(1,t) = e^{2t+1} \text{ para } 0 \le t \le 1 \end{cases}$$

2. Considere la ecuación $\pi u_t = u_{xx}$ para $0 \le x \le 1$, $0 \le t \le 1$, con las condiciones iniciales y de frontera indicadas. Establezca el tamaño de paso $h = 0.1$. ¿Para qué tamaños de paso k el método de la diferencia hacia adelante es estable? Aplique el método de la diferencia hacia adelante con tamaños de paso $h = 0.1$ y $k = 0.01$, asimismo compare su respuesta con la solución exacta del ejercicio 2.

(a)
$$\begin{cases} u(x,0) = \operatorname{sen} \pi x \text{ para } 0 \le x \le 1 \\ u(0,t) = 0 \text{ para } 0 \le t \le 1 \\ u(1,t) = 0 \text{ para } 0 \le t \le 1 \end{cases}$$

(b)
$$\begin{cases} u(x,0) = \cos \pi x \text{ para toda } 0 \le x \le 1 \\ u(0,t) = e^{-\pi t} \text{ para } 0 \le t \le 1 \\ u(1,t) = -e^{-\pi t} \text{ para } 0 \le t \le 1 \end{cases}$$

3. Utilice el método de las diferencias hacia atrás para resolver los ejercicios del problema de computadora 1. Haga una tabla con el valor exacto, el valor aproximado y el error en $(x, t) = (0.5, 1)$ para tamaños de paso $h = 0.02$ y $k = 0.02, 0.01, 0.005$.

4. Utilice el método de las diferencias hacia atrás para resolver los ejercicios del problema de computadora 2. Haga una tabla con el valor exacto, el valor aproximado y el error en $(x, t) = (0.3, 1)$ para tamaños de paso $h = 0.1$ y $k = 0.02, 0.01, 0.005$.

5. Utilice el método de Crank-Nicolson para resolver los ejercicios del problema de computadora 1. Haga una tabla con el valor exacto, el valor aproximado y el error en $(x, t) = (0.5, 1)$ para tamaños de paso $h = k = 0.02, 0.01, 0.005$.

6. Utilice el método de Crank-Nicolson para resolver los ejercicios del problema de computadora 2. Haga una tabla con el valor exacto, el valor aproximado y el error en $(x, t) = (0.3, 1)$ para tamaños de paso $h = k = 0.1, 0.05, 0.025$.

7. Establezca $D = 1$ y encuentre la C más pequeña para la cual la población de (8.26), en la fracción $[0, 10]$, sobrevive en el largo plazo. Utilice el método de Crank-Nicolson para aproximar la solución, y trate de confirmar que sus resultados no dependen de las elecciones del tamaño de paso. Compare sus resultados con la regla de supervivencia (8.27).

8. Establezca $C = D = 1$ en el modelo de población (8.26). Use el método de Crank-Nicolson para encontrar el tamaño de fracción mínima que permite que la población pueda sobrevivir. Compare con la regla (8.27).

8.2 ECUACIONES HIPERBÓLICAS

Las ecuaciones hiperbólicas ponen menos restricciones rigurosas en los métodos explícitos. En esta sección se explora la estabilidad de los métodos de diferencias finitas en el contexto de una ecuación representativa hiperbólica llamada la ecuación de onda. Se introducirá la condición CFL que es, en general, una condición necesaria para la estabilidad de los solucionadores de EDP.

8.2.1 La ecuación de onda

Considere la ecuación diferencial parcial

$$u_{tt} = c^2 u_{xx} \tag{8.28}$$

para $a \leq x \leq b$, $t \geq 0$. Al comparar con la forma normal (8.1), se calcula $B^2 - 4AC = 4c^2 > 0$, por lo que la ecuación es hiperbólica. Este ejemplo se llama la **ecuación de onda** con velocidad de onda c. Las condiciones iniciales y de frontera típicas que se requieren para especificar una solución única son

$$\begin{cases} u(x,0) = f(x) \text{ para toda } a \leq x \leq b \\ u_t(x,0) = g(x) \text{ para toda } a \leq x \leq b \\ u(a,t) = l(t) \text{ para toda } t \geq 0 \\ u(b,t) = r(t) \text{ para toda } t \geq 0 \end{cases} \tag{8.29}$$

En comparación con el ejemplo de la ecuación de calor, se requieren datos iniciales adicionales debido a la derivada del tiempo de mayor orden en la ecuación. Hablando de manera intuitiva, la ecuación de onda describe la evolución en el tiempo de una onda que se propaga a lo largo de la dirección x. Para especificar lo que pasa, es necesario conocer la forma inicial de la onda y su velocidad inicial en cada punto.

La ecuación de onda modela una amplia variedad de fenómenos, desde las ondas magnéticas en la atmósfera del Sol hasta la oscilación de una cuerda de violín. La ecuación implica una amplitud u, que para el violín representa el desplazamiento físico de la cuerda. Para una onda de sonido que viaja en el aire, u representa la presión local del aire.

Se aplicará el método de las diferencias finitas a la ecuación de onda (8.28) y se analizará su estabilidad. El método de las diferencias finitas opera sobre una malla como la de la figura 8.1, del mismo modo que en el caso parabólico. Los puntos de la malla son (x_i, t_j), donde $x_i = a + ih$ y $t_j = jk$, para los tamaños de paso h y k. Al igual que antes, se representará la aproximación a la solución $u(x_i, t_j)$ mediante w_{ij}.

Para discretizar la ecuación de onda, las segundas derivadas parciales se sustituyen por la fórmula de las diferencias centrales (8.4), en las direcciones x y t:

$$\frac{w_{i,j+1} - 2w_{ij} + w_{i,j-1}}{k^2} - c^2 \frac{w_{i-1,j} - 2w_{ij} + w_{i+1,j}}{h^2} = 0.$$

Si se establece $\sigma = ck/h$, es posible obtener la solución en el próximo paso de tiempo y escribir la ecuación discretizada como

$$w_{i,j+1} = (2 - 2\sigma^2)w_{ij} + \sigma^2 w_{i-1,j} + \sigma^2 w_{i+1,j} - w_{i,j-1}. \tag{8.30}$$

La fórmula (8.30) no puede usarse para el primer paso de tiempo, puesto que se requieren los valores de dos momentos anteriores, $j-1$ y j. Esto es semejante al problema con el inicio de los métodos de varios pasos para las EDO. A fin de resolver este problema, puede introducirse la fórmula de las diferencias centrales de tres puntos para aproximar la primera derivada del tiempo de la solución u:

$$u_t(x_i, t_j) \approx \frac{w_{i,j+1} - w_{i,j-1}}{2k}.$$

Al sustituir los datos iniciales en el primer paso de tiempo (x_i, t_1) se obtiene

$$g(x_i) = u_t(x_i, t_0) \approx \frac{w_{i1} - w_{i,-1}}{2k},$$

o en otras palabras,

$$w_{i,-1} \approx w_{i1} - 2kg(x_i). \tag{8.31}$$

Si se sustituye (8.31) en la fórmula de las diferencias finitas (8.30) para $j = 0$ resulta

$$w_{i1} = (2 - 2\sigma^2)w_{i0} + \sigma^2 w_{i-1,0} + \sigma^2 w_{i+1,0} - w_{i1} + 2kg(x_i),$$

que puede resolverse para w_{i1} y obtener

$$w_{i1} = (1 - \sigma^2)w_{i0} + kg(x_i) + \frac{\sigma^2}{2}(w_{i-1,0} + w_{i+1,0}). \tag{8.32}$$

La fórmula (8.32) se utiliza para el primer paso de tiempo. Ésta es la forma en la que la información de la velocidad inicial g entra en el cálculo. Para todos los pasos de tiempo posteriores, se emplea la fórmula (8.30). Dado que se han utilizado fórmulas de segundo orden, tanto para las derivadas de espacio como para las de tiempo, el error de este método de diferencia finita será $O(h^2) + O(k^2)$ (vea los problemas de computadora 3 y 4).

Para escribir el método de la diferencia finita en términos matriciales, defina

$$A = \begin{bmatrix} 2-2\sigma^2 & \sigma^2 & 0 & \cdots & 0 \\ \sigma^2 & 2-2\sigma^2 & \sigma^2 & \ddots & \vdots \\ 0 & \sigma^2 & 2-2\sigma^2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \sigma^2 \\ 0 & \cdots & 0 & \sigma^2 & 2-2\sigma^2 \end{bmatrix}. \tag{8.33}$$

La ecuación inicial (8.32) puede escribirse

$$\begin{bmatrix} w_{11} \\ \vdots \\ w_{m1} \end{bmatrix} = \frac{1}{2}A \begin{bmatrix} w_{10} \\ \vdots \\ w_{m0} \end{bmatrix} + k \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_m) \end{bmatrix} + \frac{1}{2}\sigma^2 \begin{bmatrix} w_{00} \\ 0 \\ \vdots \\ 0 \\ w_{m+1,0} \end{bmatrix},$$

y los pasos subsiguientes de (8.30) están dados por

$$\begin{bmatrix} w_{1,j+1} \\ \vdots \\ w_{m,j+1} \end{bmatrix} = A \begin{bmatrix} w_{1j} \\ \vdots \\ w_{mj} \end{bmatrix} - \begin{bmatrix} w_{1,j-1} \\ \vdots \\ w_{m,j-1} \end{bmatrix} + \sigma^2 \begin{bmatrix} w_{0j} \\ 0 \\ \vdots \\ 0 \\ w_{m+1,j} \end{bmatrix}.$$

Al insertar el resto de los datos adicionales, las dos ecuaciones se escriben

$$\begin{bmatrix} w_{11} \\ \vdots \\ w_{m1} \end{bmatrix} = \frac{1}{2}A \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix} + k \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_m) \end{bmatrix} + \frac{1}{2}\sigma^2 \begin{bmatrix} l(t_0) \\ 0 \\ \vdots \\ 0 \\ r(t_0) \end{bmatrix},$$

y los pasos subsiguientes de (8.30) están dados por

$$\begin{bmatrix} w_{1,j+1} \\ \vdots \\ w_{m,j+1} \end{bmatrix} = A \begin{bmatrix} w_{1j} \\ \vdots \\ w_{mj} \end{bmatrix} - \begin{bmatrix} w_{1,j-1} \\ \vdots \\ w_{m,j-1} \end{bmatrix} + \sigma^2 \begin{bmatrix} l(t_j) \\ 0 \\ \vdots \\ 0 \\ r(t_j) \end{bmatrix}. \tag{8.34}$$

► EJEMPLO 8.6 Aplique el método de las diferencias finitas explícito para la ecuación de onda con velocidad de onda $c = 2$ y las condiciones iniciales $f(x) = \text{sen } \pi x$ y $g(x) = l(x) = r(x) = 0$.

En la figura 8.10 se muestran las soluciones aproximadas de la ecuación de onda con $c = 2$. El método explícito de las diferencias finitas es condicionalmente estable; los tamaños de paso deben elegirse con cuidado para evitar la inestabilidad del método. El inciso (a) de la figura muestra una elección estable de $h = 0.05$ y $k = 0.025$, mientras que el inciso (b) muestra la elección inestable $h = 0.05$ y $k = 0.032$. El método explícito de las diferencias finitas aplicado a la ecuación de onda es inestable cuando el paso de tiempo k es demasiado grande en relación con el paso de espacio h.

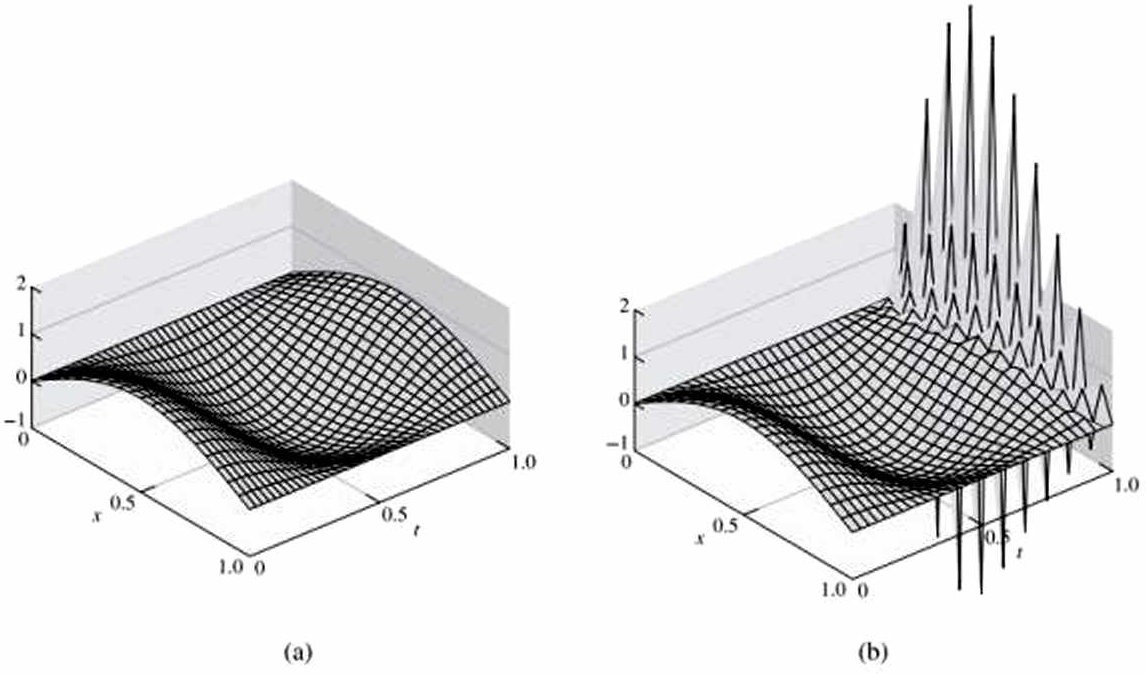


Figura 8.10 Ecuación de onda del ejemplo 8.6 aproximada mediante el método explícito de las diferencias finitas. El tamaño del paso de espacio es $h = 0.05$. (a) El método es estable para el paso de tiempo $k = 0.025$ y (b) es inestable para $k = 0.032$.

8.2.2 La condición CFL

La forma matricial permite analizar las características de estabilidad del método explícito de las diferencias finitas aplicado a la ecuación de onda. El resultado del análisis, indicado como el teorema 8.5, explica la figura 8.10.

TEOREMA 8.5 El método de las diferencias finitas aplicado a la ecuación de onda con velocidad de onda $c > 0$ es estable si $\sigma = ck/h \leq 1$. ■

Demostración. La ecuación (8.34) en forma vectorial es

$$w_{j+1} = Aw_j - w_{j-1} + \sigma^2 s_j, \tag{8.35}$$

donde s_j mantiene las condiciones laterales. Como w_{j+1} depende tanto de w_j como de w_{j-1}, para estudiar la magnificación del error se reescribe (8.35) como

$$\begin{bmatrix} w_{j+1} \\ w_j \end{bmatrix} = \begin{bmatrix} A & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} w_j \\ w_{j-1} \end{bmatrix} + \sigma^2 \begin{bmatrix} s_j \\ 0 \end{bmatrix}, \tag{8.36}$$

para ver el método como una recursión de un solo paso. El error no se magnificará siempre que los valores propios de

$$A' = \begin{bmatrix} A & -I \\ I & 0 \end{bmatrix}$$

están delimitados por 1 en valor absoluto.

Sea $\lambda \neq 0$, $(y, z)^T$ puede ser un par valor propio/vector propio de A', de modo que

$$\lambda y = Ay - z$$
$$\lambda z = y,$$

lo que implica que

$$Ay = \left(\frac{1}{\lambda} + \lambda\right) y,$$

de manera que $\mu = 1/\lambda + \lambda$ es un valor propio de A. Los valores propios de A se encuentran entre $2 - 4\sigma^2$ y 2 (ejercicio 5). El supuesto de que $\sigma \leq 1$ implica que $-2 \leq \mu \leq 2$. Para terminar, basta con demostrar que, para un número complejo, el hecho de que $1/\lambda + \lambda$ es real y tiene magnitud a lo sumo de 2 implica que $|\lambda| = 1$ (ejercicio 6).

La cantidad ck/h se llama el **número CFL** del método, en honor a R. Courant, K. Friedrichs y H. Lewy [1928]. En general, el número CFL debe tener un valor máximo de 1 para que el método de EDP sea estable. Como c es la velocidad de onda, la distancia ck recorrida por la solución en un paso de tiempo no debe exceder el paso de espacio h. En las figuras 8.10(a) y (b) se ilustran los números CFL de 1 y 1.28, respectivamente. La restricción $ck \leq h$ se denomina la **condición CFL** para la ecuación de onda.

El teorema 8.5 establece que, para la ecuación de onda, la condición CFL implica la estabilidad del método de las diferencias finitas. La condición CFL es necesaria para la estabilidad de las ecuaciones hiperbólicas más generales, pero no siempre es suficiente. Para mayores detalles consulte Morton y Mayers [1996].

El parámetro de la velocidad de onda c en la ecuación de onda regula la velocidad de la propagación de la onda. En la figura 8.11 se muestra que, para $c = 6$, la condición inicial de una onda sinusoidal oscila tres veces durante una unidad de tiempo, tres veces más rápido que el caso $c = 2$.

8.2 Ejercicios

1. Demuestre que las funciones (a) $u(x,t) = \operatorname{sen} \pi x \cos 4\pi t$, (b) $u(x,t) = e^{-x-2t}$, (c) $u(x,t) = \ln(1 + x + t)$ son soluciones de la ecuación de onda con las condiciones iniciales de frontera indicadas:

(a)
$$\begin{cases} u_{tt} = 16u_{xx} \\ u(x,0) = \operatorname{sen} \pi x \text{ para } 0 \leq x \leq 1 \\ u_t(x,0) = 0 \text{ para } 0 \leq x \leq 1 \\ u(0,t) = 0 \text{ para } 0 \leq t \leq 1 \\ u(1,t) = 0 \text{ para } 0 \leq t \leq 1 \end{cases}$$

(b)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = e^{-x} \text{ para } 0 \leq x \leq 1 \\ u_t(x,0) = -2e^{-x} \text{ para } 0 \leq x \leq 1 \\ u(0,t) = e^{-2t} \text{ para } 0 \leq t \leq 1 \\ u(1,t) = e^{-1-2t} \text{ para } 0 \leq t \leq 1 \end{cases}$$

(c)
$$\begin{cases} u_{tt} = u_{xx} \\ u(x,0) = \ln(1 + x) \text{ para } 0 \leq x \leq 1 \\ u_t(x,0) = 1/(1 + x) \text{ para } 0 \leq x \leq 1 \\ u(0,t) = \ln(1 + t) \text{ para } 0 \leq t \leq 1 \\ u(1,t) = \ln(2 + t) \text{ para } 0 \leq t \leq 1 \end{cases}$$

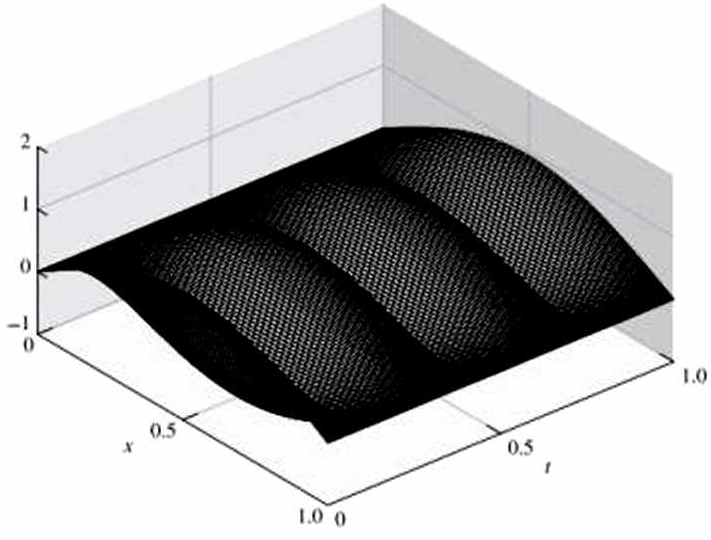


Figura 8.11 Método explícito de las diferencias finitas aplicado a la ecuación de onda, $c = 6$. Los tamaños de paso $h = 0.05$, $k = 0.008$ satisfacen la condición CFL.

2. Demuestre que las funciones (a) $u(x, t) = \operatorname{sen} \pi x \operatorname{sen} 2\pi t$, (b) $u(x, t) = (x + 2t)^5$, (c) $u(x, t) = \operatorname{senh} x \cosh 2t$ son soluciones de la ecuación de onda con las condiciones iniciales de frontera indicadas:

(a)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u_t(x,0) = 2\pi \operatorname{sen} \pi x \text{ para } 0 \le x \le 1 \\ u(0,t) = 0 \text{ para } 0 \le t \le 1 \\ u(1,t) = 0 \text{ para } 0 \le t \le 1 \end{cases}$$

(b)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = x^5 \text{ para } 0 \le x \le 1 \\ u_t(x,0) = 10x^4 \text{ para } 0 \le x \le 1 \\ u(0,t) = 32t^5 \text{ para } 0 \le t \le 1 \\ u(1,t) = (1+2t)^5 \text{ f para } 0 \le t \le 1 \end{cases}$$

(c)
$$\begin{cases} u_{tt} = 4u_{xx} \\ u(x,0) = \operatorname{senh} x \text{ para } 0 \le x \le 1 \\ u_t(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u(0,t) = 0 \text{ para } 0 \le t \le 1 \\ u(1,t) = \frac{1}{2}(e - \frac{1}{e}) \cosh 2t \text{ para } 0 \le t \le 1 \end{cases}$$

3. Demuestre que $u_1(x, t) = \operatorname{sen} \alpha x \cos c\alpha t$ y $u_2(x, t) = e^{x+\alpha}$ son soluciones de la ecuación de onda (8.28).

4. Demuestre que si $s(x)$ es dos veces diferenciable, entonces $u(x, t) = s(\alpha x + c\alpha t)$ es una solución de la ecuación de onda (8.28).

5. Demuestre que los valores propios de A en (8.33) se encuentran entre $2 - 4\sigma^2$ y 2.

6. Sea λ un número complejo. (a) Demuestre que si $\lambda + 1/\lambda$ es un número real, entonces $|\lambda| = 1$ o λ es real. (b) Demuestre que si λ es real y $|\lambda + 1/\lambda| \le 2$, entonces $|\lambda| = 1$.

8.2 Problemas de computadora

1. Resuelva los problemas de valor inicial de frontera del ejercicio 1 en $0 \le x \le 1$, $0 \le t \le 1$ mediante el método de las diferencias finitas con $h = 0.05$ y $k = h/c$. Utilice el comando `mesh` de MATLAB para graficar la solución.

2. Resuelva los problemas de valor inicial de frontera del ejercicio 2 en $0 \le x \le 1$, $0 \le t \le 1$ mediante el método de las diferencias finitas con $h = 0.05$ y k suficientemente pequeño como para satisfacer la condición CFL. Grafique la solución.
3. Para las ecuaciones de onda del ejercicio 1, haga una tabla de la aproximación y el error en $(x, t) = (1/4, 3/4)$ como una función del tamaño de paso $h = ck = 2^{-p}$ para $p = 4, \ldots, 8$.
4. Para las ecuaciones de onda del ejercicio 2, haga una tabla de la aproximación y el error en $(x, t) = (1/4, 3/4)$ como una función del tamaño de paso $h = ck = 2^{-p}$ para $p = 4, \ldots, 8$.

8.3 ECUACIONES ELÍPTICAS

En las secciones anteriores se trataron ecuaciones dependientes del tiempo. La ecuación de difusión modela el flujo de calor como una función del tiempo, y la ecuación de onda sigue el movimiento de una onda. Las ecuaciones elípticas, el tema principal de esta sección, modelan los sistemas cerrados (**estados estables**). Por ejemplo, la distribución del estado estacionario del calor en una región plana cuya frontera se mantiene a una temperatura específica, se modela mediante una ecuación elíptica. Dado que el tiempo no suele ser un factor en las ecuaciones elípticas, se utilizará x y y para indicar las variables independientes.

DEFINICIÓN 8.6 Sea $u(x, y)$ una función dos veces diferenciable, y defina el **Laplaciano** de u como

$$\Delta u = u_{xx} + u_{yy}.$$

Para una función continua $f(x, y)$, la ecuación diferencial parcial

$$\Delta u(x, y) = f(x, y) \tag{8.37}$$

se llama la **ecuación de Poisson**. La ecuación de Poisson con $f(x, y) = 0$ se denomina **ecuación de Laplace**. Una solución de la ecuación de Laplace se llama una función **armónica**.

Al comparar con la forma normal (8.1), se calcula $B^2 - 4AC < 0$, por lo que la ecuación de Poisson es elíptica. Por lo general, las condiciones adicionales dadas para determinar una solución única son condiciones de frontera. Se aplican dos tipos comunes de condiciones de frontera. Las condiciones de Dirichlet especifican los valores de la solución $u(x, y)$ en la frontera ∂R de una región R. Las condiciones de frontera de Neumann especifican los valores de la derivada direccional $\partial u/\partial n$ en la frontera, donde n indica el vector unitario normal externo.

► EJEMPLO 8.7 Demuestre que $u(x, y) = x^2 - y^2$ es una solución de la ecuación de Laplace en $[0, 1] \times [0, 1]$ con las condiciones de frontera de Dirichlet

$$\begin{aligned} u(x, 0) &= x^2 \\ u(x, 1) &= x^2 - 1 \\ u(0, y) &= -y^2 \\ u(1, y) &= 1 - y^2. \end{aligned}$$

El laplaciano es $\Delta u = u_{xx} + u_{yy} = 2 - 2 = 0$. Las condiciones de frontera se presentan para la parte inferior, superior, izquierda y derecha del cuadrado unitario, respectivamente, y se comprueban con facilidad por sustitución. ◄

Las ecuaciones de Poisson y Laplace están siempre presentes en la física clásica debido a que sus soluciones representan la energía potencial. Por ejemplo, un campo eléctrico E es el gradiente de un potencial electrostático u, o

$$E = -\nabla u.$$

A su vez, el gradiente del campo eléctrico está relacionado con la densidad de carga ρ mediante la ecuación de Maxwell

$$\nabla E = \frac{\rho}{\epsilon},$$

donde ϵ es la permisividad eléctrica. Si se igualan las dos ecuaciones resulta

$$\Delta u = \nabla(\nabla u) = -\frac{\rho}{\epsilon},$$

la ecuación de Poisson para el potencial u. En el caso especial de carga cero, el potencial satisface la ecuación de Laplace $\Delta u = 0$.

Existen muchos otros ejemplos de energía potencial que se modelan mediante la ecuación de Poisson. La aerodinámica de los perfiles de ala a bajas velocidades, conocida como flujo irrotacional incompresible, es una solución de la ecuación de Laplace. El potencial gravitatorio u generado por una distribución de masa de densidad ρ satisface la ecuación de Poisson

$$\Delta u = 4\pi G\rho,$$

donde G indica la constante gravitacional. Una distribución de calor en estado estacionario, como el límite de una solución de la ecuación de calor cuando el tiempo $t \to \infty$, se modela mediante la ecuación de Poisson. En la comprobación en la realidad 8 se utiliza una variante de la ecuación de Poisson para modelar la distribución del calor en una aleta de enfriamiento.

A continuación se presentan dos métodos para resolver ecuaciones elípticas. El primero es un método de diferencias finitas que sigue de cerca el desarrollo de las ecuaciones parabólicas e hiperbólicas. El segundo generaliza el método del elemento finito presentado en el capítulo 7 para resolver problemas de frontera. En la mayoría de las ecuaciones elípticas que se considerarán, el dominio tiene dos dimensiones, lo que hará necesario un poco de trabajo extra en el conteo.

8.3.1 Método de las diferencias finitas para ecuaciones elípticas

Se resolverá la ecuación de Poisson $\Delta u = f$ sobre un rectángulo $[x_l, x_r] \times [y_b, y_t]$ en el plano, con las condiciones de contorno de Dirichlet

$$\begin{aligned} u(x, y_b) &= g_1(x) \\ u(x, y_t) &= g_2(x) \\ u(x_l, y) &= g_3(y) \\ u(x_r, y) &= g_4(y) \end{aligned}$$

En la figura 8.12(a) se muestra una malla rectangular de puntos utilizando $M = m - 1$ pasos en la dirección horizontal y $N = n - 1$ pasos en la dirección vertical. Los tamaños de malla en las direcciones x y y son $h = (x_r - x_l)/M$ y $k = (y_t - y_b)/N$, respectivamente.

Un método de diferencias finitas implica aproximar derivadas mediante cocientes de diferencias. La fórmula de las diferencias centrales (8.4) puede usarse para las dos segundas derivadas en el operador Laplaciano. La ecuación de Poisson $\Delta u = f$ tiene la forma de diferencia finita

$$\begin{aligned} &\frac{u(x-h, y) - 2u(x, y) + u(x+h, y)}{h^2} + O(h^2) \\ &\quad + \frac{u(x, y-k) - 2u(x, y) + u(x, y+k)}{k^2} + O(k^2) = f(x, y), \end{aligned}$$

y en términos de la solución aproximada $w_{ij} \approx u(x_i, y_j)$ puede escribirse

$$\frac{w_{i-1,j} - 2w_{ij} + w_{i+1,j}}{h^2} + \frac{w_{i,j-1} - 2w_{i,j} + w_{i,j+1}}{k^2} = f(x_i, y_j) \tag{8.38}$$

donde $x_i = x_l + (i-1)h$ y $y_j = y_b + (j-1)k$ para $1 \le i \le m$ y $1 \le j \le n$.

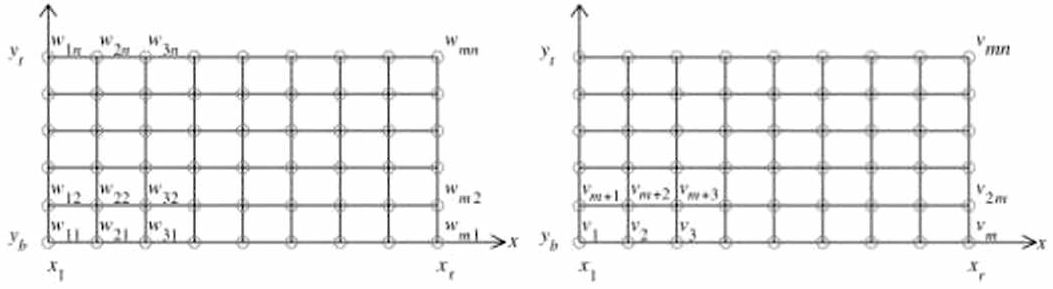


Figura 8.12 Malla para el método por diferencias finitas de la ecuación de Poisson con condiciones de frontera de Dirichlet. (a) Sistema original de numeración con subíndices dobles. (b) El sistema de numeración (8.39) para las ecuaciones lineales, con subíndices individuales, ordena los puntos a través de los renglones.

Dado que las ecuaciones de la w_{ij} son lineales, debe construirse una ecuación matricial para resolver las incógnitas mn. Esto presenta un problema de conteo: es necesario volver a etiquetar estas incógnitas doblemente indizadas en un orden lineal. En la figura 8.12(b) se muestra un sistema alternativo de numeración para los valores de la solución, donde se ha establecido

$$v_{i+(j-1)m} = w_{ij}. \tag{8.39}$$

A continuación, se construirá una matriz A y un vector b de tal forma que $Av = b$ pueda resolverse para v, y traducirse de nuevo en la solución w sobre la malla rectangular. Como v es un vector de longitud mn, A será una matriz de $mn \times mn$ y cada punto de la malla se corresponde con su propia ecuación lineal.

Por definición, la entrada A_{pq} es el q-ésimo coeficiente lineal de la p-ésima ecuación de $Av = b$. Por ejemplo, (8.38) representa la ecuación en el punto de la malla (i, j), que se llama el número de la ecuación $p = i + (j - 1)m$, de acuerdo con (8.39). Los coeficientes de los términos $w_{i-1,j}$, w_{ij}, ... en (8.38) también se numeran de acuerdo con (8.39) y se resumen en la tabla 8.1.

x	y	Ecuación número p
i	j	$i + (j - 1)m$

x	y	Coeficiente número q
i	j	$i + (j - 1)m$
$i + 1$	j	$i + 1 + (j - 1)m$
$i - 1$	j	$i - 1 + (j - 1)m$
i	$j + 1$	$i + jm$
i	$j - 1$	$i + (j - 2)m$

Tabla 8.1 Tabla de traducción para los dominios bidimensionales. La ecuación en el punto (i, j) de la malla se numera p y sus coeficientes son A_{pq} para las diferentes q, donde p y q se dan en la columna derecha de la tabla. La tabla es simplemente una ilustración de (8.39).

De acuerdo con la tabla 8.1, al etiquetar con el número de ecuación p y el número de coeficiente q, las entradas de la matriz A_{pq} de (8.38) son

$$A_{i+(j-1)m,i+(j-1)m} = -\frac{2}{h^2} - \frac{2}{k^2} \tag{8.40}$$

$$A_{i+(j-1)m,i+1+(j-1)m} = \frac{1}{h^2}$$

$$A_{i+(j-1)m,i-1+(j-1)m} = \frac{1}{h^2}$$
$$A_{i+(j-1)m,i+jm} = \frac{1}{k^2}$$
$$A_{i+(j-1)m,i+(j-2)m} = \frac{1}{k^2}.$$

El lado derecho de la ecuación correspondiente a (i, j) es

$$b_{i+(j-1)m} = f(x_i, y_j).$$

Estas entradas de A y b se cumplen para los puntos interiores $1 < i < m$, $1 < j < n$ de la malla de la figura 8.12.

Cada punto de la frontera necesita también una ecuación. Dado que se adoptan las condiciones de Dirichlet, éstas son muy simples:

Parte inferior	$w_{ij} = g_1(x_i)$	para	$j = 1,\ 1 \le i \le m$
Parte superior	$w_{ij} = g_2(x_i)$	para	$j = n,\ 1 \le i \le m$
Lado izquierdo	$w_{ij} = g_3(y_j)$	para	$i = 1,\ 1 < j < n$
Lado derecho	$w_{ij} = g_4(y_j)$	para	$i = m,\ 1 < j < n$

Las condiciones de Dirichlet se traducen a través de la tabla 8.1 como

Parte inferior	$A_{i+(j-1)m,i+(j-1)m} = 1,\ b_{i+(j-1)m} = g_1(x_i)$	para	$j = 1,\ 1 \le i \le m$
Parte superior	$A_{i+(j-1)m,i+(j-1)m} = 1,\ b_{i+(j-1)m} = g_2(x_i)$	para	$j = n,\ 1 \le i \le m$
Lado izquierdo	$A_{i+(j-1)m,i+(j-1)m} = 1,\ b_{i+(j-1)m} = g_3(y_j)$	para	$i = 1,\ 1 < j < n$
Lado derecho	$A_{i+(j-1)m,i+(j-1)m} = 1,\ b_{i+(j-1)m} = g_4(y_j)$	para	$i = m,\ 1 < j < n$

Todas las otras entradas de A y b son cero. El sistema lineal $Av = b$ pueden resolverse mediante el método apropiado del capítulo 2. En el ejemplo siguiente se ilustra este sistema de etiquetado.

► EJEMPLO 8.8 Aplique el método de las diferencias finitas con $m = n = 5$ para aproximar la solución de la ecuación de Laplace $\Delta u = 0$ en $[0, 1] \times [1, 2]$ con las siguientes condiciones de frontera de Dirichlet:

$$u(x, 1) = \ln(x^2 + 1)$$
$$u(x, 2) = \ln(x^2 + 4)$$
$$u(0, y) = 2\ln y$$
$$u(1, y) = \ln(y^2 + 1).$$

A continuación se presenta el código de MATLAB para el método de la diferencia finita:

```
% Programa 8.5 Solucionador de diferencia finita para la ecuación de Poisson en 2D
% con condiciones de frontera de Dirichlet en un rectángulo
% Entrada: rectángulo de dominio [xl,xr]x[yb,yt] con MxN pasos de espacio
% Salida: matriz w que contiene los valores de la solución
% Uso de ejemplo: w=poisson(0,1,1,2,4,4)
function w=poisson(xl,xr,yb,yt,M,N)
f=@(x,y) 0; % define los datos de la función de entrada
g1=@(x) log(x.^2+1); % define los valores de frontera
g2=@(x) log(x.^2+4); % muestra el ejemplo 8.8
g3=@(y) 2*log(y);
g4=@(y) log(y.^2+1);
```

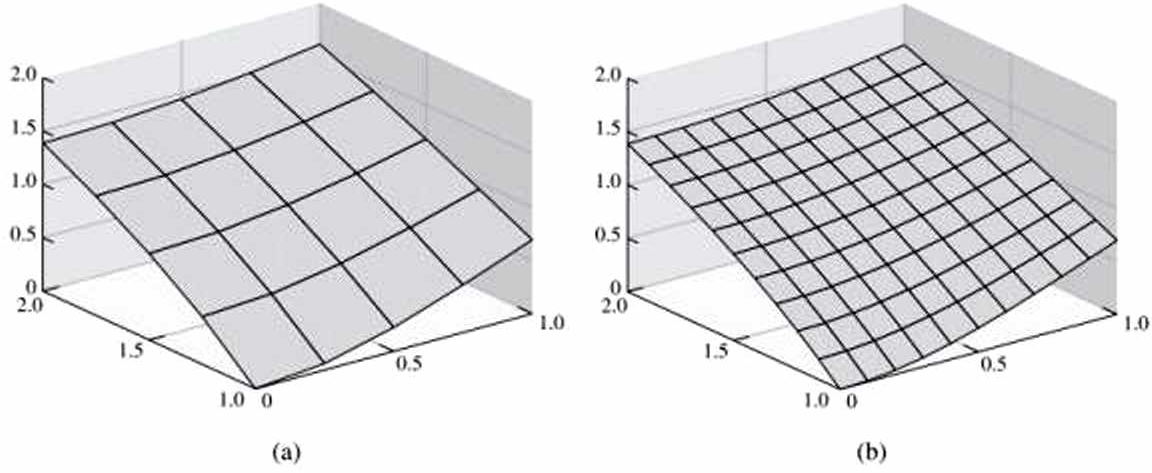


Figura 8.13 Solución del método de las diferencias finitas para la EDP elíptica del ejemplo 8.8. (a) $M = N = 4$, tamaños de malla $h = k = 0.25$ (b) $M = N = 10$, tamaños de malla $h = k = 0.1$.

```
m=M+1;n=N+1; mn=m*n;
h=(xr-xl)/M;h2=h^2;k=(yt-yb)/N;k2=k^2;
x=xl+(0:M)*h; %  establece valores de malla
y=yb+(0:N)*k;
A=zeros(mn,mn);b=zeros(mn,1);
for i=2:m-1 %  puntos inferiores
  for j=2:n-1
    A(i+(j-1)*m,i-1+(j-1)*m)=1/h2;A(i+(j-1)*m,i+1+(j-1)*m)=1/h2;
    A(i+(j-1)*m,i+(j-1)*m)=-2/h2-2/k2;
    A(i+(j-1)*m,i+(j-2)*m)=1/k2;A(i+(j-1)*m,i+j*m)=1/k2;
    b(i+(j-1)*m)=f(x(i),y(j));
  end
end
for i=1:m %  puntos de frontera inferior y superior
  j=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g1(x(i));
  j=n;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g2(x(i));
end
for j=2:n-1 %  puntos de frontera izquierda y derecha
  i=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g3(y(j));
  i=m;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g4(y(j));
end
v=A\b; %  resuelve para la solución en etiquetado v
w=reshape(v(1:mn),m,n);  %traduce de v a w
mesh(x,y,w')
```

Se utilizará la solución correcta $u(x, y) = \ln(x^2 + y^2)$ para compararla con la aproximación a los nueve puntos de la malla en el cuadrado. Como $m = n = 5$, los tamaños de malla son $h = k = 1/4$.

La solución encuentra los siguientes nueve valores interiores para u:

$$\begin{array}{lll} w_{24} = 1.1390 & w_{34} = 1.1974 & w_{44} = 1.2878 \\ w_{23} = 0.8376 & w_{33} = 0.9159 & w_{43} = 1.0341 \\ w_{22} = 0.4847 & w_{32} = 0.5944 & w_{42} = 0.7539 \end{array}$$

La solución aproximada w_{ij} se representa en la figura 8.13(a). Ésta se compara con la solución exacta $u(x, y) = \ln(x^2 + y^2)$ en los mismos puntos:

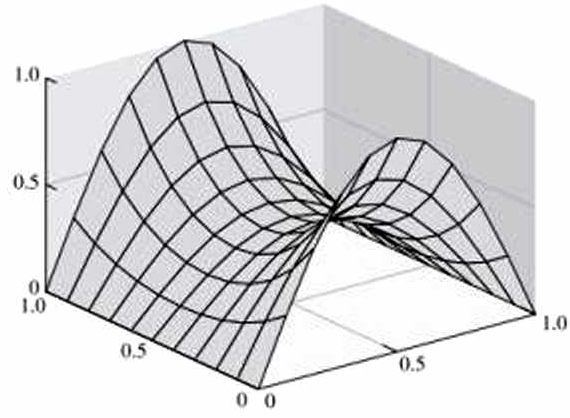


Figura 8.14 Potencial electrostático a partir de la ecuación de Laplace. Condiciones de frontera establecidas en el ejemplo 8.9.

$$u(\tfrac{1}{4}, \tfrac{7}{4}) = 1.1394 \quad u(\tfrac{2}{4}, \tfrac{7}{4}) = 1.1977 \quad u(\tfrac{3}{4}, \tfrac{7}{4}) = 1.2879$$
$$u(\tfrac{1}{4}, \tfrac{6}{4}) = 0.8383 \quad u(\tfrac{2}{4}, \tfrac{6}{4}) = 0.9163 \quad u(\tfrac{3}{4}, \tfrac{6}{4}) = 1.0341$$
$$u(\tfrac{1}{4}, \tfrac{5}{4}) = 0.4855 \quad u(\tfrac{2}{4}, \tfrac{5}{4}) = 0.5947 \quad u(\tfrac{3}{4}, \tfrac{5}{4}) = 0.7538$$

Dado que se usaron fórmulas de diferencias finitas de segundo orden, el error del método de diferencias finitas `poisson.m` es de segundo orden en h y k. La figura 8.13(b) muestra una solución aproximada más precisa, para $h = k = 0.1$. El código de MATLAB `poisson.m` está escrito para un dominio rectangular, pero es posible cambiarlo para abarcar dominios más generales. ◄

Como ejemplo adicional, se utiliza la ecuación de Laplace para calcular un potencial.

► **EJEMPLO 8.9** Encuentre el potencial electrostático en el cuadrado $[0, 1] \times [0, 1]$, suponiendo que no hay carga en el interior y asumiendo las condiciones siguientes:

$$u(x, 0) = \operatorname{sen} \pi x$$
$$u(x, 1) = \operatorname{sen} \pi x$$
$$u(0, y) = 0$$
$$u(1, y) = 0.$$

El potencial u satisface la ecuación de Laplace con condiciones de frontera de Dirichlet. Si se usa la malla de tamaño $h = k = 0.1$, o $M = N = 10$ en `poisson.m`, se obtiene la gráfica mostrada en la figura 8.14. ◄

Comprobación en la realidad **8** ***Distribución del calor en una aleta de enfriamiento***

Los disipadores de calor se utilizan para alejar el exceso de calor del punto donde se genera. En este proyecto se modela la distribución de un sistema cerrado a lo largo de una aleta rectangular de un disipador de calor. La energía térmica entra a la aleta a lo largo de uno de sus lados. El objetivo principal es diseñar las dimensiones de la aleta para mantener la temperatura dentro de las tolerancias de seguridad.

La forma de la aleta es una placa rectangular delgada, con dimensiones $L_x \times L_y$ y ancho de δ cm, donde δ es relativamente pequeño. Debido a lo delgado de la placa, se indicará la temperatura mediante $u(x, y)$ y se considerará que es constante a través del ancho de la placa.

El calor se mueve de las siguientes tres maneras: por conducción, por convección y por radiación. La conducción se refiere a la transmisión de energía entre las moléculas vecinas, tal vez

debido al movimiento de electrones, mientras que en la convección las moléculas son las que se mueven. La radiación, el movimiento de la energía a través de fotones, no se considerará aquí.

La conducción se presenta a través de un material conductor de acuerdo con la primera ley de Fourier

$$q = -KA\nabla u, \tag{8.41}$$

donde q es la energía calorífica por unidad de tiempo (medida en watts), A es el área de la sección transversal del material y ∇u es el gradiente de la temperatura. La constante K se llama la **conductividad térmica** del material. La convección se rige por la ley de enfriamiento de Newton,

$$q = -HA(u - u_b), \tag{8.42}$$

donde H es una constante de proporcionalidad llamada **coeficiente de transferencia convectiva del calor** y u_b es la temperatura ambiente, o **temperatura global**, del fluido circundante (en este caso, aire).

La aleta es un rectángulo $[0, L_x] \times [0, L_y]$ por δ cm en la dirección z, como se ilustra en la figura 8.15(a). El equilibrio de energía en una caja interior típica de la aleta de $\Delta x \times \Delta y \times \delta$, alineada a lo largo de los ejes x y y, dice que la energía que entra en la caja por unidad de tiempo es igual a la energía que sale. El flujo de calor en la caja a través de los dos lados de $\Delta y \times \delta$ y los dos lados de $\Delta x \times \delta$ es por conducción, y a través de los dos lados de $\Delta x \times \Delta y$ es por convección, de donde se obtiene la ecuación de estado estacionario

$$\begin{aligned} &-K\Delta y\delta u_x(x, y) + K\Delta y\delta u_x(x + \Delta x, y) - K\Delta x\delta u_y(x, y) \\ &\quad + K\Delta x\delta u_y(x, y + \Delta y) - 2H\Delta x\Delta y u(x, y) = 0. \end{aligned} \tag{8.43}$$

Aquí, por conveniencia se ha establecido la temperatura global $u_b = 0$, por lo que u indicará la diferencia entre la temperatura de la aleta y el entorno.

Al dividir entre $\Delta x\Delta y$ se obtiene

$$K\delta\frac{u_x(x + \Delta x, y) - u_x(x, y)}{\Delta x} + K\delta\frac{u_y(x, y + \Delta y) - u_y(x, y)}{\Delta y} = 2Hu(x, y),$$

y en el límite cuando $\Delta x, \Delta y \to 0$, resulta la ecuación diferencial parcial elíptica.

$$u_{xx} + u_{yy} = \frac{2H}{K\delta}u \tag{8.44}$$

resulta.

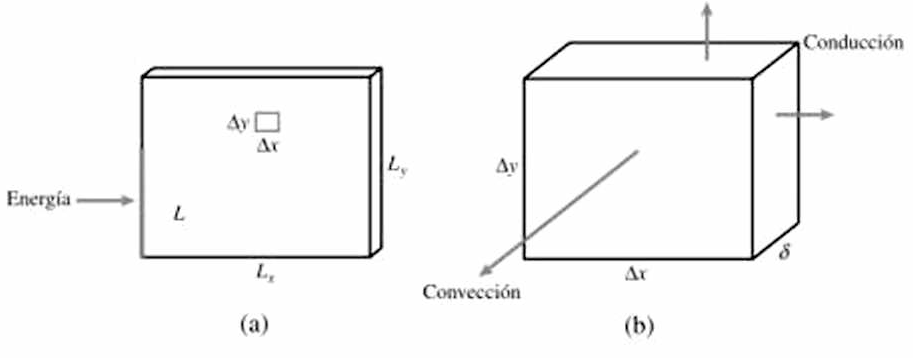


Figura 8.15 Aleta de enfriamiento de la comprobación en la realidad 8. (a) La entrada de energía se produce a lo largo del intervalo $[0, L]$ en el lado izquierdo de la aleta. (b) La transferencia de energía en la pequeña caja interior es por conducción a lo largo de las direcciones x y y por convección a lo largo de la interfaz aérea.

Argumentos similares implican la condición de frontera **convectiva**

$$Ku_{\text{normal}} = Hu$$

donde u_{normal} es la derivada parcial respecto a la dirección normal $\vec{n}$ hacia afuera. La condición de frontera convectiva que se conoce como una condición de frontera de **Robin**, la cual involucra tanto el valor de la función como el de su derivada. Por último, se supondrá que la energía entra en la aleta a lo largo de un lado de acuerdo a la ley Fourier,

$$u_{\text{normal}} = \frac{P}{L\delta K},$$

donde P es la energía total y L es la longitud de la entrada.

En una malla discreta con tamaños de paso h y k, respectivamente, puede utilizarse el método de las diferencias finitas (5.8) para aproximar la EDP (8.44) como

$$\frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{k^2} = \frac{2H}{K\delta}u_{ij}.$$

Esta discretización se usa para los puntos interiores (x_i, y_j), donde $1 < i < m$, $1 < j < n$ para los números enteros m y n. Los bordes de la aleta obedecen las condiciones de Robin empleando la aproximación a la primera derivada

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + O(h^2).$$

Para aplicar esta aproximación a los bordes de la aleta, tenga en cuenta que la dirección normal hacia afuera se traduce

$$\begin{aligned} u_{\text{normal}} &= -u_y \text{ en el borde inferior} \\ u_{\text{normal}} &= u_y \text{ en el borde superior} \\ u_{\text{normal}} &= -u_x \text{ en el borde izquierdo} \\ u_{\text{normal}} &= u_x \text{ en el borde derecho} \end{aligned}$$

Además, observe que la anterior aproximación de segundo orden a la primera derivada produce

$$\begin{aligned} u_y &\approx \frac{-3u(x,y) + 4u(x,y+k) - u(x,y+2k)}{2k} \text{ en el borde inferior} \\ u_y &\approx \frac{-3u(x,y) + 4u(x,y-k) - u(x,y-2k)}{-2k} \text{ en el borde superior} \\ u_x &\approx \frac{-3u(x,y) + 4u(x+h,y) - u(x+2h,y)}{2h} \text{ en el borde izquierdo} \\ u_x &\approx \frac{-3u(x,y) + 4u(x-h,y) - u(x-2h,y)}{-2h} \text{ en el borde derecho} \end{aligned}$$

Igualando ambas ecuaciones, la condición de frontera de Robin conduce a las ecuaciones en diferencias

$$\begin{aligned} \frac{-3u_{i1} + 4u_{i2} - u_{i3}}{2k} &= -\frac{H}{K}u_{i1} \text{ en el borde inferior} \\ \frac{-3u_{in} + 4u_{i,n-1} - u_{i,n-2}}{2k} &= -\frac{H}{K}u_{in} \text{ en el borde superior} \\ \frac{-3u_{1j} + 4u_{2j} - u_{3j}}{2h} &= -\frac{H}{K}u_{1j} \text{ en el borde izquierdo} \\ \frac{-3u_{mj} + 4u_{m-1,j} - u_{m-2,j}}{2h} &= -\frac{H}{K}u_{mj} \text{ en el borde derecho.} \end{aligned}$$

Si se supone que la energía entra por el lado izquierdo de la aleta, la ley de Fourier conduce a la ecuación

$$\frac{-3u_{1j} + 4u_{2j} - u_{3j}}{2h} = -\frac{P}{L\delta K}. \tag{8.45}$$

Hay mn ecuaciones con mn incógnitas u_{ij}, $1 \le i \le m$, $1 \le j \le n$, por resolver.

Suponga que la aleta está compuesta de aluminio, cuya conductividad térmica es $K = 1.68$ W/cm °C (watts por centímetro-grado Celsius). Suponga que el coeficiente de transferencia convectiva de calor es $H = 0.005$ W/cm^2 °C, y que la temperatura ambiente es $u_b = 20$ °C.

Actividades sugeridas

1. Comience con una aleta con dimensiones de 2×2 cm y un espesor de 1 mm. Suponga que entran 5W de potencia a lo largo de todo el borde izquierdo, como si la aleta se adjuntara para disipar la energía de un chip de CPU con longitud $L = 2$ cm por lado. Resuelva la EDP (8.44) con $M = N = 10$ pasos de las direcciones x y y. Utilice el comando `mesh` para graficar la distribución de calor resultante sobre el plano xy. ¿Cuál es la temperatura máxima de la aleta en °C?
2. Aumente el tamaño de la aleta hasta 4×4 cm. La entrada es de 5W de energía a lo largo del intervalo [0, 2] en el lado izquierdo de la aleta, como en el paso anterior. Grafique la distribución resultante y encuentre la temperatura máxima. Experimente con valores elevados de M y N. ¿Cuánto cambia la solución?
3. Encuentre la energía máxima que puede disipar una aleta de 4×4 cm, manteniendo la temperatura máxima por debajo de 80 °C. Suponga que la temperatura de global es 20 °C y la energía entra a lo largo de 2 cm, como en los pasos 1 y 2.
4. Reemplace la aleta de aluminio por una aleta de cobre, con una conductividad térmica $K = 3.85$ W/cm °C. Encuentre la energía máxima que puede disipar una aleta de 4×4 cm, con la entrada de energía de 2 cm colocada en forma óptima, mientras se mantiene la temperatura máxima por debajo de 80 °C.
5. Grafique la energía máxima que puede disiparse en el paso 4 (manteniendo la temperatura máxima por debajo de 80 grados) como una función de la conductividad térmica, para $1 \le K \le 5$ W/cm °C.
6. Vuelva a realizar el paso 4 para una aleta de agua congelada. Suponga que el agua tiene un coeficiente de transferencia convectiva del calor de $H = 0.1$ W/cm^2 °C, y que la temperatura ambiente del agua se mantiene en 20 °C.
7. Haga un corte rectangular del lado derecho de la aleta y rehaga el paso 4. ¿La aleta recortada disipa más o menos potencia que la original?

El diseño de las aletas de enfriamiento para computadoras portátiles y de escritorio es un problema de ingeniería fascinante. Para disipar cantidades cada vez mayores de calor, se requieren varias aletas en un espacio pequeño y se utilizan ventiladores para mejorar la convección cerca de los bordes de las aletas. La adición de ventiladores en la complicada geometría de las aletas desplaza la simulación al dominio de la dinámica computacional de fluidos, un área vital de las matemáticas aplicadas modernas. ✔

8.3.2 Método del elemento finito para ecuaciones elípticas

Un enfoque de alguna manera más flexible para la resolución de ecuaciones diferenciales parciales surgió de la comunidad de la ingeniería estructural a mediados del siglo XX. El método del elemento finito convierte la ecuación diferencial en una equivalente variacional llamada la forma débil de la ecuación, y utiliza la poderosa idea de la ortogonalidad en los espacios de las funciones para estabilizar sus cálculos. Además, el sistema de ecuaciones lineales resultante puede tener una simetría considerable en su matriz de estructura, incluso si la geometría subyacente es complicada.

Se aplicarán los elementos finitos utilizando el método de Galerkin, como se introdujo en el capítulo 7 para los problemas de ecuaciones diferenciales ordinarias de valores en la frontera. El método para las EDP sigue los mismos pasos, aunque los requisitos de conteo son más extensos. Considere el problema de Dirichlet para la ecuación elíptica

$$\begin{aligned} \Delta u + r(x, y)u &= f(x, y) \quad \text{en } R \\ u &= g(x, y) \quad \text{sobre } S \end{aligned} \tag{8.46}$$

donde la solución $u(x, y)$ se define sobre una región R en un plano limitado por una curva cerrada S uniforme por partes.

Se utilizará un espacio funcional L^2 en la región R, como en el capítulo 7. Sea

$$L^2(R) = \left\{ \text{funciones } \phi(x, y) \text{ sobre } R \;\middle|\; \int\int_R \phi(x, y)^2 \, dx \, dy \text{ existe y es finita} \right\}.$$

Se designa por $L_0^2(R)$ el subespacio de $L^2(R)$ que consta de las funciones que son cero en la frontera S de la región R.

El objetivo será disminuir al mínimo el error cuadrático de la ecuación elíptica en (8.46) al forzar el residuo $\Delta u(x, y) + r(x, y)u(x, y) - f(x, y)$ a ser ortogonal a un subespacio grande de $L^2(R)$. Sean $\phi_1(x, y), \ldots, \phi_P(x, y)$ elementos de $L^2(R)$. El supuesto de ortogonalidad toma la forma

$$\int\int_R (\Delta u + ru - f)\phi_p \, dx \, dy = 0,$$

o

$$\int\int_R (\Delta u + ru)\phi_p \, dx dy = \int\int_R f\phi_p \, dx \, dy \tag{8.47}$$

para cada $1 \leq p \leq P$. La forma (8.47) se llama la **forma débil** de la ecuación elíptica (8.46).

La versión de la integración por partes necesaria para aplicar el método de Galerkin está contenida en el siguiente hecho:

TEOREMA 8.7 **Primera identidad de Green.** Sea R una región acotada con la frontera S uniforme por partes. Sean u y v funciones suaves, y sea n la normal unitaria hacia afuera a lo largo de la frontera. Entonces

$$\int\int_R v\Delta u = \int_S v\frac{\partial u}{\partial n} dS - \int\int_R \nabla u \cdot \nabla v. \qquad \blacksquare$$

La derivada direccional puede calcularse como

$$\frac{\partial u}{\partial n} = \nabla u \cdot (n_x, n_y),$$

donde (n_x, n_y) indica el vector unitario normal hacia afuera en la frontera S de R. Al aplicar la identidad de Green a la forma débil (8.47) se obtiene

$$\int_S \phi_p \frac{\partial u}{\partial n} \, dS - \int\int_R (\nabla u \cdot \nabla \phi_p) \, dx \, dy + \int\int_R ru\phi_p \, dx \, dy = \int\int_R f\phi_p \, dx \, dy. \tag{8.48}$$

La esencia del método del elemento finito es sustituir

$$w(x, y) = \sum_{q=1}^{P} v_q \phi_q(x, y) \tag{8.49}$$

por u en la forma débil de la ecuación diferencial parcial, y después determinar las constantes desconocidas v_q. Suponga por el momento que ϕ_p pertenece a $L_0^2(R)$, es decir, $\phi_p(S) = 0$. Al sustituir la forma (8.49) en (8.48) resulta

$$-\int\int_R \left(\sum_{q=1}^{P} v_q \nabla\phi_q\right) \cdot \nabla\phi_p \, dx\,dy + \int\int_R r\left(\sum_{q=1}^{P} v_q\phi_q\right)\phi_p \, dx\,dy = \int\int_R f\phi_p \, dx\,dy$$

para cada ϕ_p en $L_0^2(R)$. Si se factorizan las constantes v_q se obtiene

$$\sum_{q=1}^{P} v_q \left[\int\int_R \nabla\phi_q \cdot \nabla\phi_p \, dx\,dy - \int\int_R r\phi_q\phi_p \, dx\,dy\right] = -\int_R f\phi_p dx\,dy. \tag{8.50}$$

Para cada ϕ_p perteneciente a $L_0^2(R)$, se ha desarrollado una ecuación lineal con las incógnitas $v_1, \ldots, v_p$. En forma matricial, la ecuación es $Av = b$, donde las entradas de la p-ésima fila de A y b son

$$A_{pq} = \int\int_R \nabla\phi_q \cdot \nabla\phi_p \, dx\,dy - \int\int_R r\phi_q\phi_p \, dx\,dy \tag{8.51}$$

y

$$b_p = -\int\int_R f\phi_p \, dx\,dy. \tag{8.52}$$

Ahora se tiene la posibilidad de elegir funciones explícitas para los elementos finitos ϕ_p y planificar un cálculo. Se seguirá el ejemplo del capítulo 7 en la elección de splines lineales en B, funciones lineales por partes de x, y representados por triángulos en el plano. De manera más específica, considere que la región R es un rectángulo y forme una triangulación con los nodos (x_i, y_j) obtenidos de una malla rectangular. Se volverá a utilizar la malla de $M \times N$ de la sección anterior, que se muestra en la figura 8.16(a), donde se estableció $m = \mathrm{M} + 1$ y $n = N + 1$. Al igual que antes, se indicará el tamaño de paso de la malla en las direcciones x y y como h y k, respectivamente. En la figura 8.16(b) se muestra la triangulación de la región rectangular que va a utilizarse.

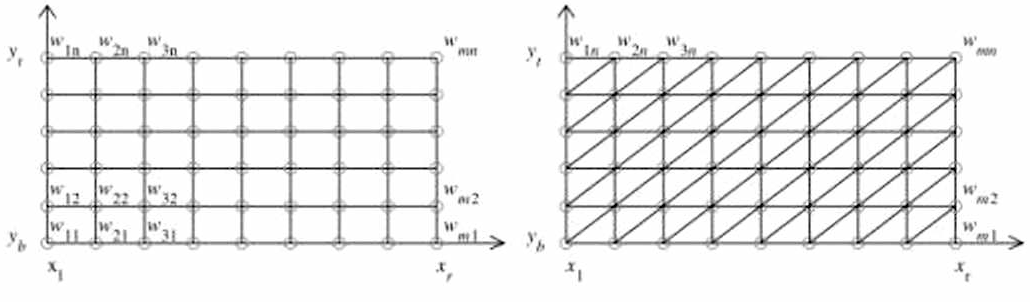


Figura 8.16 Solución por elementos finitos de la ecuación elíptica con condiciones de frontera de Dirichlet. (a) La malla es la misma que se utilizó para el método por diferencias finitas. (b) Una triangulación posible de la región. Cada punto interior es un vértice de seis triángulos diferentes.

La elección de las funciones de elemento finito ϕ_p a partir de $L^2(R)$ serán las $P = mn$ funciones lineales por partes, cada una de las cuales toma el valor 1 en un punto de la malla de la figura 8.16(a) y cero en los otros $mn - 1$ puntos de la misma. En otras palabras, $\phi_1, \ldots, \phi_{mn}$ se determinan mediante la igualdad $\phi_{i+(j-1)m}(x_i, y_i) = 1$ y $\phi_{i+(j-1)m}(x_{i'}, y_{j'}) = 0$ para todos los demás puntos de la malla $(x_{i'}, y_{j'})$, siempre que sean lineales en cada triángulo de la figura 8.16(b). Una vez más, se está usando el sistema de numeración de la tabla 8.1 en la página 400. Cada $\phi_p(x, y)$ es diferenciable, excepto a lo largo de los bordes del triángulo y, por lo tanto, es una función integrable de Riemann que pertenece a $L^2(R)$. Tenga en cuenta que cada punto que no está en la frontera (x_i, y_j) del rectángulo R, $\phi_{i+(j-1)m}$ pertenece a $L_0^2(R)$. Aún más, debido al supuesto (8.49), satisfacen

$$w(x_i, y_j) = \sum_{i=1}^{m}\sum_{j=1}^{n} v_{i+(j-1)m}\phi_{i+(j-1)m}(x_i, y_j) = v_{i+(j-1)m}$$

para $i = 1, \ldots, m, j = 1, \ldots, n$. Por lo tanto, la aproximación w a la solución correcta u en (x_i, v_j) estarán directamente disponibles una vez que se resuelva el sistema $Av = b$. Esta conveniencia es la razón por la que las splines en B se eligen como funciones de los elementos finitos.

Quedan por calcular las entradas de la matriz (8.51) y (8.52) y resolver $Av = b$. Para calcular estas entradas, se recopilan algunos datos sobre las splines en B en el plano. Las integrales de las funciones lineales por partes se aproximan con facilidad por la regla del punto medio bidimensional. Defina el **baricentro** de una región en el plano como el punto $(\overline{x}, \overline{y})$ donde

$$\overline{x} = \frac{\int\int_R x\, dx\, dy}{\int\int_R 1\, dx\, dy}, \quad \overline{y} = \frac{\int\int_R y\, dx\, dy}{\int\int_R 1\, dx\, dy}.$$

Si R es un triángulo con vértices (x_1, y_1), (x_2, y_2), (x_3, y_3), entonces el baricentro es (vea el ejercicio 8)

$$\overline{x} = \frac{x_1 + x_2 + x_3}{3}, \quad \overline{y} = \frac{y_1 + y_2 + y_3}{3}.$$

LEMA 8.8 El valor medio de una función lineal $L(x, y)$ en una región plana R es $L(\overline{x}, \overline{y})$, el valor en el baricentro. En otras palabras, $\int\int_R L(x, y)\, dx\, dy = L(\overline{x}, \overline{y}) \cdot$ área (R). ■

Demostración. Sea $L(x, y) = a + bx + cy$. Entonces

$$\begin{aligned}\int\int_R L(x, y)\, dx\, dy &= \int\int_R (a + bx + cy)\, dx\, dy \\ &= a\int\int_R dx\, dy + b\int\int_R x\, dx\, dy + c\int\int_R y\, dx\, dy \\ &= \text{área } (R) \cdot (a + b\overline{x} + c\overline{y}).\end{aligned}$$

❐

El lema 8.8 conduce a una generalización de la regla del punto medio del capítulo 5 que es útil para aproximar las entradas de (8.51) y (8.52). El teorema de Taylor para funciones de dos variables, dice que

$$\begin{aligned}f(x, y) &= f(\overline{x}, \overline{y}) + \frac{\partial f}{\partial x}(\overline{x}, \overline{y})(x - \overline{x}) + \frac{\partial f}{\partial y}(\overline{x}, \overline{y})(y - \overline{y}) \\ &\quad + O((x - \overline{x})^2, (x - \overline{x})(y - \overline{y}), (y - \overline{y})^2) \\ &= L(x, y) + O((x - \overline{x})^2, (x - \overline{x})(y - \overline{y}), (y - \overline{y})^2).\end{aligned}$$

Por lo tanto,

$$\int\int_R f(x, y)\, dx\, dy = \int\int_R L(x, y)\, dx\, dy + \int\int_R O((x - \bar{x})^2, (x - \bar{x})(y - \bar{y}), (y - \bar{y})^2)\, dx\, dy$$
$$= \text{área}\,(R) \cdot L(\bar{x}, \bar{y}) + O(h^4) = \text{área}\,(R) \cdot f(\bar{x}, \bar{y}) + O(h^4),$$

donde h es el **diámetro** de R, la distancia más grande entre dos puntos de R, y donde se ha usado el Lema 8.8. Ésta es la regla del punto medio en dos dimensiones.

Regla del punto medio en dos dimensiones

$$\int\int_R f(x, y)\, dx\, dy = \text{área}\,(R) \cdot f(\bar{x}, \bar{y}) + O(h^4), \tag{8.53}$$

donde $(\bar{x}, \bar{y})$ es el baricentro de la región acotada R y $h = \text{diám}(R)$.

La regla del punto medio indica que para aplicar el método del elemento finito con convergencia $O(h^2)$, sólo es necesario aproximar las integrales en (8.51) y (8.52) mediante la evaluación de los integrandos en los baricentros triangulares. Para las funciones spline en B ϕ_p, es particularmente fácil. Las pruebas de los próximos dos lemas se dejan para los ejercicios 9 y 10.

LEMA 8.9 Sea $\phi(x, y)$ una función lineal sobre el triángulo T con vértices (x_1, y_1), (x_2, y_2), (x_3, y_3), que satisfacen $\phi(x_1, y_1) = 1$, $\phi(x_2, y_2) = 0$ y $\phi(x_3, y_3) = 0$. Entonces, $\phi(\bar{x}, \bar{y}) = 1/3$. ■

LEMA 8.10 Sean $\phi_1(x, y)$ y $\phi_2(x, y)$ las funciones lineales sobre el triángulo T con vértices (x_1, y_1), (x_2, y_2) y (x_3, y_3), que satisfacen $\phi_1(x_1, y_1) = 1$, $\phi_1(x_2, y_2) = 0$, $\phi_1(x_3, y_3) = 0$, $\phi_2(x_1, y_1) = 0$, $\phi_2(x_2, y_2) = 1$, y $\phi_2(x_3, y_3) = 0$. Sea $f(x, y)$ una función dos veces diferenciable. Establezca

$$d = \det \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix}.$$

Entonces

(a) el triángulo T tiene un área de $|d|/2$

(b) $\nabla\phi_1(x, y) = \left(\dfrac{y_2 - y_3}{d}, \dfrac{x_3 - x_2}{d}\right)$

(c) $\int\int_T \nabla\phi_1 \cdot \nabla\phi_1\, dx\, dy = \dfrac{(x_2 - x_3)^2 + (y_2 - y_3)^2}{2|d|}$

(d) $\int\int_T \nabla\phi_1 \cdot \nabla\phi_2\, dx\, dy = \dfrac{-(x_1 - x_3)(x_2 - x_3) - (y_1 - y_3)(y_2 - y_3)}{2|d|}$

(e) $\int\int_T f\phi_1\phi_2\, dx\, dy = f(\bar{x}, \bar{y})|d|/18 + O(h^4) = \int\int_T f\phi_1^2\, dx\, dy$

(f) $\int\int_T f\phi_1\, dx\, dy = f(\bar{x}, \bar{y})|d|/6 + O(h^4)$

donde $(\bar{x}, \bar{y})$ es el baricentro de T y $h = \text{diám}(T)$. ■

Ahora es posible calcular las entradas matriciales de A. Considere un vértice (x_i, y_i) que no está en la frontera S del rectángulo. Entonces $\phi_{i+(j-1)m}$ pertenece a $L_0^2(R)$ y de acuerdo con (8.51) con $p = q = i + (j - 1)m$, la entrada matricial $A_{i+(j-1)m,i+(j-1)m}$ se compone de dos integrales. Los integrandos son cero fuera de los seis triángulos mostrados en la figura 8.17.

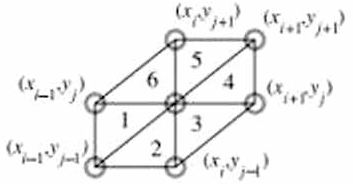


Figura 8.17 Detalle del punto interior (i, j) de la figura 8.16(b). Cada punto interior (x_i, y_j) está rodeado por seis triángulos, numerados como se muestra. La función spline en B $\phi_{i+(j-1)m}$ es lineal, toma el valor 1 en el centro y es cero fuera de estos seis triángulos.

Los triángulos tienen los lados horizontal y vertical h y k, respectivamente. Para la primera integral, la suma desde el triángulo 1 hasta el triángulo 6, respectivamente, puede usarse el lema 8.10(c) para incluir las seis contribuciones

$$\frac{k^2}{2hk} + \frac{h^2}{2hk} + \frac{h^2+k^2}{2hk} + \frac{k^2}{2hk} + \frac{h^2}{2hk} + \frac{h^2+k^2}{2hk} = \frac{2(h^2+k^2)}{hk}. \tag{8.54}$$

Para la segunda integral de (8.51) se utiliza el lema 8.10(e). De nuevo, las integrales son cero excepto para los seis triángulos mostrados. Los baricentros de los seis triángulos son

$$\begin{aligned}
B_1 &= (x_i - \frac{2}{3}h, y_j - \frac{1}{3}k)\\
B_2 &= (x_i - \frac{1}{3}h, y_j - \frac{2}{3}k)\\
B_3 &= (x_i + \frac{1}{3}h, y_j - \frac{1}{3}k)\\
B_4 &= (x_i + \frac{2}{3}h, y_j + \frac{1}{3}k)\\
B_5 &= (x_i + \frac{1}{3}h, y_j + \frac{2}{3}k)\\
B_6 &= (x_i - \frac{1}{3}h, y_j + \frac{1}{3}k).
\end{aligned} \tag{8.55}$$

La segunda integral contribuye con $-(hk/18)[r(B_1) + r(B_2) + r(B_3) + r(B_4) + r(B_5) + r(B_6)]$, por lo que al sumar (8.54) y (8.55),

$$\begin{aligned}
A_{i+(j-1)m,i+(j-1)m} = \frac{2(h^2+k^2)}{hk} - \frac{hk}{18}[&r(B_1) + r(B_2) + r(B_3)\\
&+ r(B_4) + r(B_5) + r(B_6)].
\end{aligned} \tag{8.56}$$

Un uso similar del lema 8.10 (vea el ejercicio 12) muestra que

$$\begin{aligned}
A_{i+(j-1)m,i-1+(j-1)m} &= -\frac{k}{h} - \frac{hk}{18}[r(B_6) + r(B_1)]\\
A_{i+(j-1)m,i-1+(j-2)m} &= -\frac{hk}{18}[r(B_1) + r(B_2)]\\
A_{i+(j-1)m,i+(j-2)m} &= -\frac{h}{k} - \frac{hk}{18}[r(B_2) + r(B_3)]\\
A_{i+(j-1)m,i+1+(j-1)m} &= -\frac{k}{h} - \frac{hk}{18}[r(B_3) + r(B_4)]\\
A_{i+(j-1)m,i+1+jm} &= -\frac{hk}{18}[r(B_4) + r(B_5)]
\end{aligned}$$

$$A_{i+(j-1)m,i+jm} = -\frac{h}{k} - \frac{hk}{18}[r(B_5) + r(B_6)] \tag{8.57}$$

Para calcular las entradas b_p se usa el lema 8.10(f), lo que implica que para $p = i - (j-1)m$,

$$b_{i+(j-1)m} = -\frac{hk}{6}[f(B_1) + f(B_2) + f(B_3) + f(B_4) + f(B_5) + f(B_6)]. \tag{8.58}$$

Para las funciones de elemento finito en la frontera, $\phi_{i+(j-1)m}$ no pertenecen a $L_0^2(R)$, y las ecuaciones

$$\begin{aligned} A_{i+(j-1)m,i+(j-1)m} &= 1 \\ b_{i+(j-1)m} &= g(x_i, y_j) \end{aligned} \tag{8.59}$$

se utilizará para garantizar la condición de contorno Dirichlet $v_{i+(j-1)m} = g(x_i, y_i)$, donde (x_i, y_j) es un punto en la frontera.

Con estas fórmulas, es sencillo de construir una aplicación en MATLAB del método de elemento finito sobre un rectángulo con las condiciones de contorno de Dirichlet. El programa consiste en crear la matriz A y el vector b usando (8.56) a (8.59), para después resolver $Av = b$. Aunque en el código de MATLAB se utiliza la operación de barra invertida, para las aplicaciones reales ésta podría sustituirse por un método disperso como en el capítulo 2.

```
% Programa 8.6 Solucionador de elemento finito para EDP en 2D
% con condiciones de frontera de Dirichlet en un rectángulo
% Entrada: rectángulo de dominio [xl,xr]x[yb,yt] con MxN pasos de espacio
% Salida: matriz w que contiene los valores de la solución
% Uso de ejemplo: w=poissonfem(0,1,1,2,4,4)
function w=poissonfem(xl,xr,yb,yt,M,N)
f=@(x,y) 0; % define los datos de la función de entrada
r=@(x,y) 0;
g1=@(x) log(x.^2+1); % define los valores de frontera en parte inferior
g2=@(x) log(x.^2+4); % superior
g3=@(y) 2*log(y);    % lado izquierdo
g4=@(y) log(y.^2+1); % lado derecho
m=M+1; n=N+1; mn=m*n;
h=(xr-xl)/M; h2=h^2; k=(yt-yb)/N; k2=k^2; hk=h*k;
x=xl+(0:M)*h; %  establece valores de malla
y=yb+(0:N)*k;
A=zeros(mn,mn); b=zeros(mn,1);
for i=2:m-1 %  puntos inferiores
 for j=2:n-1
  rsum=r(x(i)-2*h/3,y(j)-k/3)+r(x(i)-h/3,y(j)-2*k/3)...
       +r(x(i)+h/3,y(j)-k/3);
  rsum=rsum+r(x(i)+2*h/3,y(j)+k/3)+r(x(i)+h/3,y(j)+2*k/3)...
       +r(x(i)-h/3,y(j)+k/3);
  A(i+(j-1)*m,i+(j-1)*m)=2*(h2+k2)/(hk)-hk*rsum/18;
  A(i+(j-1)*m,i-1+(j-1)*m)=-k/h-hk*(r(x(i)-h/3,y(j)+k/3)...
                           +r(x(i)-2*h/3,y(j)-k/3))/18;
  A(i+(j-1)*m,i-1+(j-2)*m)=-hk*(r(x(i)-2*h/3,y(j)-k/3)...
                           +r(x(i)-h/3,y(j)-2*k/3))/18;
  A(i+(j-1)*m,i+(j-2)*m)=-h/k-hk*(r(x(i)-h/3,y(j)-2*k/3)...
                         +r(x(i)+h/3,y(j)-k/3))/18;
  A(i+(j-1)*m,i+1+(j-1)*m)=-k/h-hk*(r(x(i)+h/3,y(j)-k/3)...
                           +r(x(i)+2*h/3,y(j)+k/3))/18;
```

```
  A(i+(j-1)*m,i+1+j*m)=-hk*(r(x(i)+2*h/3,y(j)+k/3)...
                          +r(x(i)+h/3,y(j)+2*k/3))/18;
  A(i+(j-1)*m,i+j*m)=-h/k-hk*(r(x(i)+h/3,y(j)+2*k/3)...
                        +r(x(i)-h/3,y(j)+k/3))/18;
  fsum=f(x(i)-2*h/3,y(j)-k/3)+f(x(i)-h/3,y(j)-2*k/3)...
       +f(x(i)+h/3,y(j)-k/3);
  fsum=fsum+f(x(i)+2*h/3,y(j)+k/3)+f(x(i)+h/3,y(j)+2*k/3)...
       +f(x(i)-h/3,y(j)+k/3);
  b(i+(j-1)*m)=-h*k*fsum/6;
 end
end
for i=1:m %  puntos de frontera
  j=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g1(x(i));
  j=n;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g2(x(i));
end
for j=2:n-1
  i=1;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g3(y(j));
  i=m;A(i+(j-1)*m,i+(j-1)*m)=1;b(i+(j-1)*m)=g4(y(j));
end
v=A\b; %  resuelve para la solución en etiquetado v
w=reshape(v(1:mn),m,n);
mesh(x,y,w')
```

► **EJEMPLO 8.10** Aplique el método del elemento finito con $M = N = 4$ para aproximar la solución de la ecuación de Laplace $\Delta u = 0$ en $[0, 1] \times [1, 2]$ con las condiciones de frontera de Dirichlet:

$$\begin{aligned} u(x,1) &= \ln(x^2+1) \\ u(x,2) &= \ln(x^2+4) \\ u(0,y) &= 2\ln y \\ u(1,y) &= \ln(y^2+1) \end{aligned}$$

Como $M = N = 4$, hay un sistema lineal de $mn \times mn$ por resolver. Dieciséis de las 25 ecuaciones son la evaluación de las condiciones de frontera. Al resolver $Av = b$ se obtiene

$$\begin{array}{lll} w_{24} = 1.1390 & w_{34} = 1.1974 & w_{44} = 1.2878 \\ w_{23} = 0.8376 & w_{33} = 0.9159 & w_{43} = 1.0341 \\ w_{22} = 0.4847 & w_{32} = 0.5944 & w_{42} = 0.7539 \end{array}$$

de acuerdo con los resultados del ejemplo 8.8. ◄

► **EJEMPLO 8.11** Aplique el método del elemento finito con $M = N = 16$ para aproximar la solución del problema elíptico de Dirichlet

$$\begin{cases} \Delta u + 4\pi^2 u = 2\operatorname{sen} 2\pi y \\ u(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u(x,1) = 0 \text{ para } 0 \le x \le 1 \\ u(0,y) = 0 \text{ para } 0 \le y \le 1 \\ u(1,y) = \operatorname{sen} 2\pi y \text{ para } 0 \le y \le 1 \end{cases}$$

Se define $r(x, y) = 4\pi^2$ y $f(x, y) = 2\text{sen}2\pi y$. Como $m = n = 17$, la malla es de 17×17, lo que significa que la matriz A es de 289×289. La solución se calcula dentro de un error máximo de aproximadamente 0.023, en comparación con la solución correcta $u(x, y) = x^2\text{sen}2\pi y$. La solución aproximada w se muestra en la figura 8.18. ◄

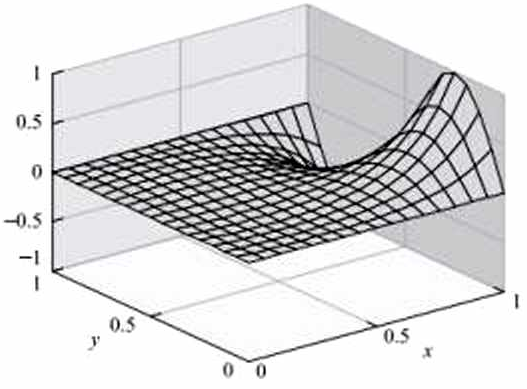


Figura 8.18 Solución de elemento finito para el ejemplo 8.11. El error máximo en [0, 1] × [0, 1] es 0.023.

8.3 Ejercicios

1. Demuestre que $u(x, y) = \ln(x^2 + y^2)$ es una solución a la ecuación de Laplace con las condiciones de frontera de Dirichlet del ejemplo 8.8.

2. Demuestre que (a) $u(x, y) = x^2y - 1/3\, y^3$ y (b) $u(x, y) = 1/6\, x^4 - x^2y^2 + 1/6\, y^4$ son funciones armónicas.

3. Demuestre que las funciones (a) $u(x, y) = e^{-\pi y} \operatorname{sen} \pi x$, (b) $u(x, y) = \operatorname{senh} \pi x \operatorname{sen} \pi y$ son soluciones de la ecuación de Laplace con las condiciones de frontera indicadas:

(a)
$$\begin{cases} u(x,0) = \operatorname{sen} \pi x \text{ para } 0 \le x \le 1 \\ u(x,1) = e^{-\pi} \operatorname{sen} \pi x \text{ para } 0 \le x \le 1 \\ u(0,y) = 0 \text{ para } 0 \le y \le 1 \\ u(1,y) = 0 \text{ para } 0 \le y \le 1 \end{cases}$$

(b)
$$\begin{cases} u(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u(x,1) = 0 \text{ para } 0 \le x \le 1 \\ u(0,y) = 0 \text{ para } 0 \le y \le 1 \\ u(1,y) = \operatorname{senh} \pi \operatorname{sen} \pi y \text{ para } 0 \le y \le 1 \end{cases}$$

4. Demuestre que las funciones (a) $u(x, y) = e^{-\pi y}$, (b) $u(x, y) = (x^2 + y^2)^{3/2}$ son soluciones de la ecuación de Poisson especificada con las condiciones de frontera dadas:

(a)
$$\begin{cases} \Delta u = e^{-xy}(x^2 + y^2) \\ u(x,0) = 1 \text{ para } 0 \le x \le 1 \\ u(x,1) = e^{-x} \text{ para } 0 \le x \le 1 \\ u(0,y) = 1 \text{ para } 0 \le y \le 1 \\ u(1,y) = e^{-y} \text{ para } 0 \le y \le 1 \end{cases}$$

(b)
$$\begin{cases} \Delta u = 9\sqrt{x^2 + y^2} \\ u(x,0) = x^3 \text{ para } 0 \le x \le 1 \\ u(x,1) = (1 + x^2)^{3/2} \text{ para } 0 \le x \le 1 \\ u(0,y) = y^3 \text{ para } 0 \le y \le 1 \\ u(1,y) = (1 + y^2)^{3/2} \text{ para } 0 \le y \le 1 \end{cases}$$

5. Demuestre que las funciones de (a) $u(x, y) = \operatorname{sen} \frac{\pi}{2} xy$, (b) $u(x, y) = e^{xy}$ son soluciones de la ecuación elíptica especificada con las condiciones de frontera de Dirichlet dadas:

(a)
$$\begin{cases} \Delta u + \frac{\pi^2}{4}(x^2 + y^2)u = 0 \\ u(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u(x,1) = \operatorname{sen} \frac{\pi}{2} x \text{ para } 0 \le x \le 1 \\ u(0,y) = 0 \text{ para } 0 \le y \le 1 \\ u(1,y) = \operatorname{sen} \frac{\pi}{2} y \text{ para } 0 \le y \le 1 \end{cases}$$

(b)
$$\begin{cases} \Delta u = (x^2 + y^2)u \\ u(x,0) = 1 \text{ para } 0 \le x \le 1 \\ u(x,1) = e^x \text{ para } 0 \le x \le 1 \\ u(0,y) = 1 \text{ para } 0 \le y \le 1 \\ u(1,y) = e^y \text{ para } 0 \le y \le 1 \end{cases}$$

6. Demuestre que las funciones (a) $u(x, y) = e^{x+2y}$, (b) $u(x, y) = y/x$ son soluciones de la ecuación elíptica especificada con las condiciones de frontera de Dirichlet dadas:

(a)
$$\begin{cases} \Delta u = 5u \\ u(x,0) = e^x \text{ para } 0 \le x \le 1 \\ u(x,1) = e^{x+2} \text{ para } 0 \le x \le 1 \\ u(0,y) = e^{2y} \text{ para } 0 \le y \le 1 \\ u(1,y) = e^{2y+1} \text{ para } 0 \le y \le 1 \end{cases}$$

(b)
$$\begin{cases} \Delta u = \dfrac{2u}{x^2} \\ u(x,0) = 0 \text{ para } 1 \le x \le 2 \\ u(x,1) = 1/x \text{ para } 1 \le x \le 2 \\ u(1,y) = y \text{ para } 0 \le y \le 1 \\ u(2,y) = y/2 \text{ para } 0 \le y \le 1 \end{cases}$$

7. Demuestre que las funciones (a) $u(x, y) = x^2 + y^2$, (b) $u(x, y) = y^2/x$ son soluciones de la ecuación elíptica especificada con las condiciones de frontera de Dirichlet dadas:

(a)
$$\begin{cases} \Delta u + \dfrac{u}{x^2+y^2} = 5 \\ u(x,1) = x^2 + 1 \text{ para } 1 \le x \le 2 \\ u(x,2) = x^2 + 4 \text{ para } 1 \le x \le 2 \\ u(1,y) = y^2 + 1 \text{ para } 1 \le y \le 2 \\ u(2,y) = y^2 + 4 \text{ para } 1 \le y \le 2 \end{cases}$$

(b)
$$\begin{cases} \Delta u - \dfrac{2u}{x^2} = \dfrac{2}{x} \\ u(x,0) = 0 \text{ para } 1 \le x \le 2 \\ u(x,2) = 4/x \text{ para } 1 \le x \le 2 \\ u(1,y) = y^2 \text{ para } 0 \le y \le 2 \\ u(2,y) = y^2/2 \text{ para } 0 \le y \le 2 \end{cases}$$

8. Demuestre que el baricentro de un triángulo con vértices (x_1, y_1), (x_2, y_2), (x_3, y_3) es $\bar{x} = (x_1 + x_2 + x_3)/3$, $\bar{y} = (y_1 + y_2 + y_3)/3$.

9. Demuestre el lema 8.9.

10. Demuestre el lema 8.10.

11. Obtenga las coordenadas del baricentro de (8.55).

12. Obtenga las entradas de la matriz en (8.57).

13. Demuestre que la ecuación de Laplace $\Delta T = 0$ sobre el rectángulo $[0, L] \times [0, H]$ con las condiciones de frontera de Dirichlet $T = T_0$ en los tres lados $x = 0$, $x = L$ y $y = 0$, así como $T = T_1$ en el lado $y = H$, tiene la solución

$$T(x,y) = T_0 + \sum_{k=0}^{\infty} C_k \operatorname{sen}\frac{(2k+1)\pi x}{L} \operatorname{senh}\frac{(2k+1)\pi y}{L}$$

donde

$$C_k = \frac{4(T_1 - T_0)}{(2k+1)\pi \operatorname{senh}\frac{(2k+1)\pi H}{L}}.$$

8.3 Problemas de computadora

1. Resuelva los problemas de la ecuación de Laplace del ejercicio 3 en $0 \le x \le 1$, $0 \le y \le 1$ mediante el método de las diferencias finitas con $h = k = 0.1$. Utilice el comando `mesh` de MATLAB para graficar la solución.

2. Resuelva los problemas de la ecuación de Poisson del ejercicio 4 en $0 \le x \le 1$, $0 \le y \le 1$ mediante el método de las diferencias finitas con $h = k = 0.1$. Grafique la solución.

3. Utilice el método de las diferencias finitas con $h = k = 0.1$ para aproximar el potencial electrostático en el cuadrado $0 \le x, y \le 1$ a partir de la ecuación de Laplace con las condiciones de frontera especificadas. Grafique la solución.

(a)
$$\begin{cases} u(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u(x,1) = \operatorname{sen}\pi x \text{ para } 0 \le x \le 1 \\ u(0,y) = 0 \text{ para } 0 \le y \le 1 \\ u(1,y) = 0 \text{ para } 0 \le y \le 1 \end{cases}$$

(b)
$$\begin{cases} u(x,0) = \operatorname{sen}\frac{\pi}{2}x \text{ para } 0 \le x \le 1 \\ u(x,1) = \cos\frac{\pi}{2}x \text{ para } 0 \le x \le 1 \\ u(0,y) = \operatorname{sin}\frac{\pi}{2}y \text{ para } 0 \le y \le 1 \\ u(1,y) = \cos\frac{\pi}{2}y \text{ para } 0 \le y \le 1 \end{cases}$$

4. Use el método de las diferencias finitas con $h = k = 0.1$ para aproximar el potencial electrostático en el cuadro $0 \le x, y \le 1$ de la ecuación de Laplace con las condiciones de contorno especificadas. Grafique la solución.

(a)
$$\begin{cases} u(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u(x,1) = x^3 \text{ para } 0 \le x \le 1 \\ u(0,y) = 0 \text{ para } 0 \le y \le 1 \\ u(1,y) = y^2 \text{ para } 0 \le y \le 1 \end{cases}$$

(b)
$$\begin{cases} u(x,0) = 0 \text{ para } 0 \le x \le 1 \\ u(x,1) = x \operatorname{sen} \frac{\pi}{2} x \text{ para } 0 \le x \le 1 \\ u(0,y) = 0 \text{ para } 0 \le y \le 1 \\ u(1,y) = y \text{ para } 0 \le y \le 1 \end{cases}$$

5. La presión hidrostática puede expresarse como la carga hidráulica, definida como la altura u equivalente de una columna de agua que ejerce esa presión. En un depósito subterráneo, el flujo de agua subterránea en ese sistema cerrado satisface la ecuación de Laplace $\Delta u = 0$. Suponga que el depósito tiene dimensiones de 2 km $\times$ 1 km, y las alturas de agua

$$\begin{cases} u(x,0) = 0.01 \text{ para } 0 \le x \le 2 \\ u(x,1) = 0.01 + 0.003x \text{ para } 0 \le x \le 2 \\ u(0,y) = 0.01 \text{ para } 0 \le y \le 1 \\ u(1,y) = 0.01 + 0.006y^2 \text{ para } 0 \le y \le 1 \end{cases}$$

en la frontera del depósito, en kilómetros. Calcule la carga $u(1, 1/2)$ en el centro del depósito.

6. La temperatura u en el sistema cerrado de una placa de cobre calentado satisface la ecuación de Poisson

$$\Delta u = -\frac{D(x,y)}{K},$$

donde $D(x, y)$ es la densidad de la potencia en (x, y) y K es la conductividad térmica. Suponga que la placa tiene la forma de un rectángulo de $[0, 4] \times [0, 2]$ cm cuya frontera se mantiene a una temperatura constante de 30 °C, y que la potencia se genera a la razón constante $D(x, y) = 5$ watts/cm^3. La conductividad térmica del cobre es $K = 3.85$ watts/cm °C. (a) Grafique la distribución de la temperatura en la placa. (b) Encuentre la temperatura en el centro $(x, y) = (2, 1)$.

7. Para las ecuaciones de Laplace del ejercicio 3, haga una tabla de la aproximación por diferencias finitas y el error en $(x, y) = (1/4, 3/4)$ como una función de los tamaños de paso $h = k = 2^{-p}$ para $p = 2, \ldots, 5$.

8. Para las ecuaciones de Poisson del ejercicio 4, haga una tabla de la aproximación por diferencias finitas y el error en $(x, y) = (1/4, 3/4)$ como una función de los tamaños de paso $h = k = 2^{-p}$ para $p = 2, \ldots, 5$.

9. Resuelva los problemas de la ecuación de Laplace del ejercicio 3 en $0 \le x \le 1$, $0 \le y \le 1$ mediante el método del elemento finito con $h = k = 0.1$. Utilice el comando `mesh` de MATLAB para graficar la solución.

10. Resuelva los problemas de la ecuación de Poisson del ejercicio 4 en $0 \le x \le 1$, $0 \le y \le 1$ mediante el método del elemento finito con $h = k = 0.1$. Grafique la solución.

11. Resuelva las ecuaciones diferenciales parciales elípticas del ejercicio 5 mediante el método del elemento finito con $h = k = 0.1$. Grafique la solución.

12. Resuelva las ecuaciones diferenciales parciales elípticas del ejercicio 6 mediante el método del elemento finito con $h = k = 1/16$. Grafique la solución.

13. Resuelva las ecuaciones diferenciales parciales elípticas del ejercicio 7 mediante el método del elemento finito con $h = k = 1/16$. Grafique la solución.

14. Resuelva las ecuaciones diferenciales parciales elípticas con las condiciones de frontera de Dirichlet dadas. Utilice el método del elemento finito con $h = k = 0.1$. Grafique la solución.

(a)
$$\begin{cases} \Delta u + \operatorname{sen} \pi x y = (x^2 + y^2)u \\ u(x, 0) = 0 \text{ para } 0 \le x \le 1 \\ u(x, 1) = 0 \text{ para } 0 \le x \le 1 \\ u(0, y) = 0 \text{ para } 0 \le y \le 1 \\ u(1, y) = 0 \text{ para } 0 \le y \le 1 \end{cases}$$

(b)
$$\begin{cases} \Delta u + (\operatorname{sen} \pi x y)u = e^{2xy} \\ u(x, 0) = 0 \text{ para } 0 \le x \le 1 \\ u(x, 1) = 0 \text{ para } 0 \le x \le 1 \\ u(0, y) = 0 \text{ para } 0 \le y \le 1 \\ u(1, y) = 0 \text{ para } 0 \le y \le 1 \end{cases}$$

15. Para las ecuaciones elípticas del ejercicio 5, haga una tabla de la aproximación por elemento finito y el error en $(x, y) = (1/4, 3/4)$ como una función de los tamaños de paso $h = k = 2^{-p}$ para $p = 2, \ldots, 5$.

16. Para las ecuaciones elípticas del ejercicio 6, haga una gráfica log-log del error máximo del método del elemento finito como una función del tamaño de paso $h = k = 2^{-p}$ para $p = 2, \ldots, 6$.

17. Para las ecuaciones elípticas del ejercicio 7, haga una gráfica log-log del error máximo del método del elemento finito como una función del tamaño de paso $h = k = 2^{-p}$ para $p = 2, \ldots, 6$.

18. Resuelva la ecuación de Laplace con las condiciones de frontera de Dirichlet del ejercicio 13 en $[0, 1] \times [0, 1]$, con $T_0 = 0$ y $T_1 = 10$. Utilice (a) una aproximación por diferencias finitas y (b) el método del elemento finito. Haga gráficas log-log del error en sitios particulares del rectángulo como una función de los tamaños de paso $h = k = 2^{-p}$ para una p tan grande como sea posible. Explique cualquier simplificación que haga para evaluar la solución correcta en esos sitios.

8.4 ECUACIONES DIFERENCIALES PARCIALES NO LINEALES

En las secciones anteriores de este capítulo se han analizado las diferencias finitas y los elementos finitos aplicados a EDP lineales. Para el caso no lineal, es necesario hacer un arreglo adicional para que los métodos anteriores resulten apropiados.

De manera concreta, se hará énfasis en el método implícito de las diferencias hacia atrás de la sección 8.1 y su aplicación a ecuaciones de difusión no lineal. Los cambios similares realizados a este método pueden aplicarse a cualquiera de las técnicas que se han estudiado para que puedan usarse en las ecuaciones no lineales.

8.4.1 Solucionador implícito de Newton

Se ilustra el método con un ejemplo no lineal típico

$$u_t + uu_x = Du_{xx}, \tag{8.60}$$

conocido como la **ecuación de Burgers**. La ecuación es no lineal debido al término del producto uu_x. Esta ecuación elíptica, nombrada en honor de J. M. Burgers (1895-1981), es un modelo simplificado del flujo de un fluido. Cuando el coeficiente de difusión $D = 0$, se denomina ecuación de Burgers no viscosa. La fijación de $D > 0$ corresponde a agregar viscosidad al modelo.

Esta ecuación de difusión se discretizará en la misma forma que la ecuación del calor de la sección 8.1. Considere la malla de puntos mostrada en la figura 8.1. La solución aproximada en (x_i, t_j) se indicará por w_{ij}. Sean M y N el número total de pasos en las direcciones x y t, y sean $h = (b - a)/M$ y $k = T/N$ los tamaños de paso en las direcciones x y t. Al aplicar las diferencias hacia atrás sobre u_t y las diferencias centrales sobre los otros términos se obtiene

$$\frac{w_{ij} - w_{i,j-1}}{k} + w_{ij}\left(\frac{w_{i+1,j} - w_{i-1,j}}{2h}\right) = \frac{D}{h^2}(w_{i+1,j} - 2w_{ij} + w_{i-1,j}),$$

o

$$w_{ij} + \frac{k}{2h}w_{ij}(w_{i+1,j} - w_{i-1,j}) - \sigma(w_{i+1,j} - 2w_{ij} + w_{i-1,j}) - w_{i,j-1} = 0 \tag{8.61}$$

donde se ha establecido $\sigma = Dk/h^2$. Tenga en cuenta que, debido a los términos cuadráticos de las variables w, no es posible resolver directamente para $w_{i+1,j}$, w_{ij}, $w_{i-1,j}$, en forma explícita o implícita. Por lo tanto, se invoca el método multivariado de Newton del capítulo 2 para obtener la solución.

Para clarificar la implementación, indique las incógnitas en (8.61) como $z_i = w_{ij}$. En el paso de tiempo j, se trata de resolver las ecuaciones

$$F_i(z_1, \ldots, z_m) = z_i + \frac{k}{2h}z_i(z_{i+1} - z_{i-1}) - \sigma(z_{i+1} - 2z_i + z_{i-1}) - w_{i,j-1} = 0 \tag{8.62}$$

para las m incógnitas $z_1, \ldots, z_m$. Tenga en cuenta que el último término w_{ij-1} se conoce a partir del paso de tiempo anterior y se trata como una cantidad conocida.

La primera y última ecuaciones se sustituyen por las condiciones de frontera adecuadas. Por ejemplo, en el caso de la ecuación de Burgers con condiciones de frontera de Dirichlet

$$\begin{cases} u_t + uu_x = Du_{xx} \\ u(x,0) = f(x) \text{ para } x_l \le x \le x_r \\ u(x_l,t) = l(t) \text{ para toda } t \ge 0 \\ u(x_r,t) = r(t) \text{ para toda } t \ge 0, \end{cases} \tag{8.63}$$

se añadirán las ecuaciones

$$\begin{aligned} F_1(z_1, \ldots, z_m) &= z_1 - l(t_j) = 0 \\ F_m(z_1, \ldots, z_m) &= z_m - r(t_j) = 0. \end{aligned} \tag{8.64}$$

Ahora hay m ecuaciones algebraicas no lineales con m incógnitas.

Para aplicar el método multivariado de Newton, debe calcularse el jacobiano $DF(\vec{z}) = \partial\vec{F}/\partial\vec{z}$, que de acuerdo con (8.62) y (8.64) tendrá la forma tridiagonal

$$\begin{bmatrix} 1 & 0 & \cdots & & \\ -\sigma - \frac{kz_2}{2h} & 1 + 2\sigma + \frac{k(z_3 - z_1)}{2h} & -\sigma + \frac{kz_2}{2h} & & \\ & -\sigma - \frac{kz_3}{2h} & 1 + 2\sigma + \frac{k(z_4 - z_2)}{2h} & -\sigma + \frac{kz_3}{2h} & \\ & & \ddots & \ddots & \ddots \\ & & & -\sigma - \frac{kz_{m-1}}{2h} & 1 + 2\sigma + \frac{k(z_m - z_{m-2})}{2h} & -\sigma + \frac{kz_{m-1}}{2h} \\ & & & \cdots & 0 & 1 \end{bmatrix}$$

En general, las ecuaciones superior e inferior de DF dependen de las condiciones de frontera. Una vez que se ha construido DF, se resuelve para $z_i = w_{ij}$ mediante la iteración multivariada de Newton

$$\vec{z}^{K+1} = \vec{z}^K - DF(\vec{z}^K)^{-1}F(\vec{z}^K). \tag{8.65}$$

► EJEMPLO 8.12 Utilice la ecuación de las diferencias hacia atrás con la iteración de Newton para resolver la ecuación de Burgers

$$\begin{cases} u_t + uu_x = Du_{xx} \\ u(x,0) = \dfrac{2D\beta\pi \operatorname{sen}\pi x}{\alpha + \beta\cos\pi x} \text{ para } 0 \le x \le 1 \\ u(0,t) = 0 \text{ para toda } t \ge 0 \\ u(1,t) = 0 \text{ para toda } t \ge 0. \end{cases} \tag{8.66}$$

A continuación se presenta el código de MATLAB para la versión de la condición de frontera de Dirichlet con el método de Newton, donde se ha establecido $\alpha = 5$, $\beta = 4$. El programa utiliza tres iteraciones de Newton para cada paso de tiempo. En los problemas típicos, esto sería suficiente, pero en los casos difíciles pueden requerirse más. Observe que la eliminación gaussiana o su equivalente se lleva a cabo en la iteración de Newton; como de costumbre, no es necesario invertir la matriz de manera explícita.

```
% Programa 8.7 Solucionador implícito de Newton para la ecuación de Newton
% entrada: intervalo de espacio [xl,xr], intervalo de tiempo [tb,te],
%          número de pasos de espacio M, número de pasos de tiempo N
% Salida: solución w
% Uso de ejemplo: w=burgers(0,1,0,2,20,40)
function w=burgers(xl,xr,tb,te,M,N)
alf=5;bet=4;D=.05;
f=@(x) 2*D*bet*pi*sin(pi*x)./(alf+bet*cos(pi*x));
l=@(t) 0*t;
r=@(t) 0*t;
h=(xr-xl)/M; k=(te-tb)/N; m=M+1; n=N;
sigma=D*k/(h*h);
w(:,1)=f(xl+(0:M)*h)';                    % condiciones iniciales
wl=w;
for j=1:n
  for it=1:3                              % iteración de Newton
    DF1=zeros(m,m);DF2=zeros(m,m);
    DF1=diag(1+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1);
    DF1=DF1+diag(-sigma*ones(m-1,1),-1);
    DF2=diag([0;k*wl(3:m)/(2*h);0])-diag([0;k*wl(1:(m-2))/(2*h);0]);
    DF2=DF2+diag([0;k*wl(2:m-1)/(2*h)],1)...
        -diag([k*wl(2:m-1)/(2*h);0],-1);
    DF=DF1+DF2;
    F=-w(:,j)+(DF1+DF2/2)*wl;     % Usando el lema 8.11
    DF(1,:)=[1 zeros(1,m-1)];     % Condiciones de Dirichlet para DF
    DF(m,:)=[zeros(1,m-1) 1];
    F(1)=wl(1)-l(j);F(m)=wl(m)-r(j);  % Condiciones de Dirichlet para F
    wl=wl-DF\F;
  end
  w(:,j+1)=wl;
end
x=xl+(0:M)*h;t=tb+(0:n)*k;
mesh(x,t,w')                              % Gráfica en 3D de la solución w
```

El código es una implementación directa de la iteración de Newton (8.65), junto con un hecho conveniente acerca de los polinomios homogéneos. Por ejemplo, considere el polinomio $P(x_1, x_2, x_3) = x_1x_2x_3^2 + x_1^4$, que se denomina homogéneo de cuarto grado, ya que consiste por completo en términos de cuarto grado en x_1, x_2, x_3. Las derivadas parciales de P con respecto a las tres variables están contenidas en el gradiente

$$\nabla P = (x_2x_3^2 + 4x_1^3, x_1x_3^2, 2x_1x_2x_3).$$

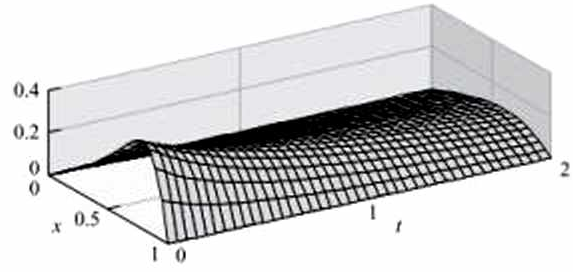


Figura 8.19 Solución aproximada a la ecuación de Burgers (8.66). Se suponen condiciones de frontera de Dirichlet homogéneas, con tamaños de paso $h = k = 0.05$.

El hecho notable es que P puede recuperarse al multiplicar el gradiente por el vector de las variables, con un múltiplo adicional de 4:

$$\nabla P \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = (x_2 x_3^2 + 4x_1^3)x_1 + x_1 x_3^2 x_2 + 2x_1 x_2 x_3 x_3 = 4x_1 x_2 x_3^2 + 4x_1^4 = 4P.$$

En general, defina el polinomio $P(x_1, \ldots, x_m)$ como **homogéneo** de grado d si

$$P(cx_1, \ldots, cx_m) = c^d P(x_1, \ldots, x_m) \tag{8.67}$$

para toda c.

LEMA 8.11 Sea $P(x_1, \ldots, x_m)$ un polinomio homogéneo de grado d. Entonces

$$\nabla P \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = dP.$$

■

Demostración. Diferenciando (8.67) con respecto a c resulta

$$x_1 P_{x_1}(cx_1, \ldots, cx_m) + \ldots + x_m P_{x_m}(cx_1, \ldots, cx_m) = dc^{d-1} P(x_1, \ldots, x_m)$$

utilizando la regla de la cadena multivariada. La evaluación en $c = 1$ da como resultado la conclusión deseada. ❐

El uso de este hecho permite escribir un código muy compacto para ecuaciones diferenciales parciales con términos polinomiales, siempre y cuando se agrupen los términos del mismo grado. Observe cómo la matriz `DF1` en el programa 8.7 agrupa los términos de las derivadas de primer grado en `F`; `DF2` agrupa los términos de las derivadas de segundo grado. Entonces es posible definir la matriz jacobiana `DF` como la suma de los términos de las derivadas de primer y segundo grado y, en esencia, definir la función `F` como la suma de los términos de grado 0, 1 y 2. El lema 8.11 se utiliza para identificar los términos de grado d de `F` como gradientes de tiempo variables, divididos por d. La conveniencia de esta simplificación será aún más aceptada cuando se aborden problemas más difíciles.

Para ciertas condiciones de frontera, se conoce una solución explícita de la ecuación de Burgers. La solución al problema de Dirichlet (8.66) es

$$u(x,t) = \frac{2D\beta\pi e^{-D\pi^2 t} \operatorname{sen} \pi x}{\alpha + \beta e^{-D\pi^2 t} \cos \pi x}. \tag{8.68}$$

La solución exacta puede utilizase para medir la exactitud de este método de aproximación, como una función de los tamaños de paso h y k. Si se usan los parámetros $\alpha = 5$, $\beta = 4$ y el coeficiente de difusión $D = 0{,}05$, se encuentra que los errores en $x = 1/2$ después de una unidad de tiempo son los siguientes:

h	k	$u(0.5, 1)$	$w(0.5, 1)$	error
0.01	0.04	0.153435	0.154624	0.001189
0.01	0.02	0.153435	0.154044	0.000609
0.01	0.01	0.153435	0.153749	0.000314

Se observa la disminución aproximadamente de primer orden en el error como una función del tamaño de paso de tiempo k, como se esperaba con el método implícito de las diferencias hacia atrás. ◄

Otra categoría interesante de las EDP no lineales está compuesta de las **ecuaciones de reacción-difusión**. Un ejemplo fundamental de una ecuación de reacción-difusión no lineal se debe al revolucionario biólogo y genetista R.A. Fisher (1890-1962), sucesor de Darwin, quien ayudó a crear las bases de la estadística moderna. La ecuación se obtuvo en un principio para modelar cómo se propagan los genes. La forma general de la **ecuación de Fisher** es

$$u_t = Du_{xx} + f(u), \tag{8.69}$$

donde $f(u)$ es un polinomio en u. La parte de reacción en la ecuación es la función f, la parte de difusión es Du_{xx}. Si se usan las condiciones de frontera homogéneas de Neumann, la constante o el estado de equilibrio $u(x, t) \equiv C$ es una solución siempre que $f(C) = 0$. El estado de equilibrio resulta ser estable si $f'(C) < 0$, lo que significa que las soluciones cercanas tienden hacia el estado de equilibrio.

► EJEMPLO 8.13 Use la ecuación de las diferencias hacia atrás con la iteración de Newton para resolver la ecuación de Fisher; utilice las condiciones de frontera homogéneas de Neumann

$$\begin{cases} u_t = Du_{xx} + u(1-u) \\ u(x,0) = \operatorname{sen} \pi x \text{ para } 0 \le x \le 1 \\ u_x(0,t) = 0 \text{ para toda } t \ge 0 \\ u_x(1,t) = 0 \text{ para toda } t \ge 0. \end{cases} \tag{8.70}$$

Observe que $f(u) = u(1-u)$, lo que implica que $f'(u) = 1 - 2u$. El equilibrio $u = 0$ satisface $f'(0) = 1$ y la otra solución de equilibrio $u = 1$ satisface $f'(1) = -1$. Por lo tanto, las soluciones tienden hacia el equilibrio $u = 1$.

La discretización repasa el proceso de obtención de la ecuación de Burgers:

$$\frac{w_{ij} - w_{i,j-1}}{k} = \frac{D}{h^2}(w_{i+1,j} - 2w_{ij} + w_{i-1,j}) + w_{ij}(1 - w_{ij}),$$

o

$$(1 + 2\sigma - k(1 - w_{ij}))w_{ij} - \sigma(w_{i+1,j} + w_{i-1,j}) - w_{i,j-1} = 0. \tag{8.71}$$

Esto resulta en las ecuaciones no lineales

$$F_i(z_1, \ldots, z_m) = (1 + 2\sigma - k(1 - z_i))z_i - \sigma(z_{i+1} + z_{i-1}) - w_{i,j-1} = 0 \tag{8.72}$$

para resolver para las $z_i = w_{ij}$ en el j-ésimo paso de tiempo. La primera y última ecuaciones establecerán las condiciones de frontera de Neumann:

$$F_1(z_1, \ldots, z_m) = (-3z_0 + 4z_1 - z_2)/(2h) = 0$$
$$F_m(z_1, \ldots, z_m) = (-z_{m-2} + 4z_{m-1} - 3z_m)/(-2h) = 0$$

El jacobiano DF tiene la forma

$$\begin{bmatrix} -3 & 4 & -1 & & & \\ -\sigma & 1+2\sigma-k+2kz_2 & -\sigma & & & \\ & -\sigma & 1+2\sigma-k+2kz_3 & -\sigma & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\sigma & 1+2\sigma-k+2kz_{m-1} & -\sigma \\ & & & -1 & 4 & -3 \end{bmatrix}.$$

Después de alternar la función F y el jacobiano DF, la iteración de Newton implementada en el programa 8.7 puede utilizarse para resolver la ecuación de Fisher. El lema 8.11 puede emplearse para separar las partes de primero y segundo grado de DF. También se aplican las condiciones de frontera de Neumann, como se muestra en el siguiente fragmento de código:

```
DF1=diag(1-k+2*sigma*ones(m,1))+diag(-sigma*ones(m-1,1),1);
DF1=DF1+diag(-sigma*ones(m-1,1),-1);
DF2=diag(2*k*w1);
DF=DF1+DF2;
F=-w(:,j)+(DF1+DF2/2)*w1;
DF(1,:)=[-3 4 -1 zeros(1,m-3)];F(1)=DF(1,:)*w1;
DF(m,:)=[zeros(1,m-3) -1 4 -3];F(m)=DF(m,:)*w1;
```

En la figura 8.20 se muestran las soluciones aproximadas de la ecuación de Fisher con $D = 1$, que demuestran la tendencia a estabilizarse hacia la atracción del equilibrio $u(x, t) \equiv 1$. Por supuesto, $u(x, t) \equiv 0$ es también una solución de (8.69) con $f(u) = u(1 - u)$, y se encontrará mediante los datos iniciales $u(x, 0) = 0$. Sin embargo, casi todos los demás datos iniciales, se acercarán con el tiempo a $u = 1$ a medida que t crece.

El ejemplo 8.13 cubre la ecuación original considerada por Fisher, pero hay muchas versiones generalizadas para otras elecciones del polinomio $f(u)$. Consulte otras exploraciones de esta ecuación de reacción-difusión en los problemas de computadora. A continuación, se investigará una versión de la ecuación de Fisher con mayor dimensión.

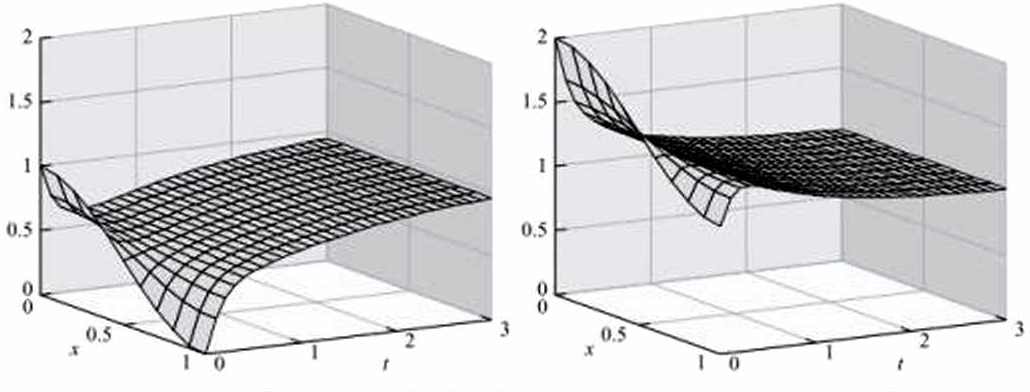


Figura 8.20 Dos soluciones a la ecuación de Fisher. Ambas soluciones tienden hacia la solución de equilibrio $u(x, t) = 1$ a medida que t crece. (a) Condición inicial $u(x, 0) = 0.5 + 0.5\cos\pi x$. (b) Condición inicial $u(x, 0) = 1.5 + 0.5\cos\pi x$. Se suponen condiciones de frontera homogéneas de Neumann, con tamaños de paso $h = k = 0.1$.

8.4.2 Ecuaciones no lineales en dos dimensiones espaciales

La solución de ecuaciones diferenciales parciales con dominios bidimensionales obliga a combinar las técnicas de las secciones anteriores. El método implícito de las diferencias hacia atrás con la iteración de Newton se encargará de la no linealidad, y será necesario aplicar las coordenadas al estilo acordeón de la tabla 8.1 para hacer el conteo para el dominio bidimensional.

Se inicia por extender la ecuación de Fisher de una dimensión a dos dimensiones en el espacio.

► EJEMPLO 8.14 Aplique el método de las diferencias hacia atrás con la iteración de Newton a la ecuación de Fisher en el cuadrado unitario $[0, 1] \times [0, 1]$:

$$\begin{cases} u_t = D\Delta u + u(1-u) \\ u(x, y, 0) = 2 + \cos \pi x \cos \pi y \text{ para } 0 \le x, y \le 1 \\ u_{\vec{n}}(x, y, t) = 0 \text{ en la frontera del rectángulo, para toda } t \ge 0. \end{cases} \tag{8.73}$$

Aquí D es el coeficiente de difusión y $u_{\vec{n}}$ indica la derivada direccional en el sentido normal hacia afuera. Se suponen condiciones de frontera de Neumann, o sin flujo, en el límite del rectángulo.

En esta sección, los dos subíndices de discretización representarán las dos coordenadas espaciales x y y, y se usarán superíndices para indicar los pasos de tiempo. Suponiendo M pasos en la dirección x y N pasos en la dirección y, se definirán los tamaños de paso $h = (x_r - x_l)/M$ y $k = (y_t - y_b)/N$. Las ecuaciones discretizadas en los puntos de la malla que no están en la frontera, para $1 < i < m = M + 1$, $1 < j < n = N + 1$, son

$$\begin{aligned} \frac{w_{ij}^t - w_{ij}^{t-\Delta t}}{\Delta t} &= \frac{D}{h^2}(w_{i+1,j}^t - 2w_{ij}^t + w_{i-1,j}^t) + \frac{D}{k^2}(w_{i,j+1}^t - 2w_{ij}^t + w_{i,j-1}^t) \\ &\quad + w_{ij}^t(1 - w_{ij}^t), \end{aligned} \tag{8.74}$$

que pueden reorganizarse en la forma $F_{ij}(w^t) = 0$, o bien

$$\begin{aligned} &\left(\frac{1}{\Delta t} + \frac{2D}{h^2} + \frac{2D}{k^2} - 1\right) w_{ij}^t - \frac{D}{h^2} w_{i+1,j}^t - \frac{D}{h^2} w_{i-1,j}^t - \frac{D}{k^2} w_{i,j+1}^t - \frac{D}{k^2} w_{i,j-1}^t \\ &+ (w_{ij}^t)^2 - \frac{w_{ij}^{t-\Delta t}}{\Delta t} = 0 \end{aligned} \tag{8.75}$$

Las ecuaciones F_{ij} deben resolverse en forma implícita. Las ecuaciones son no lineales, por lo que se utilizará el método de Newton como se hizo para la versión unidimensional de la ecuación de Fisher. Como ahora el dominio es bidimensional, es necesario recordar el sistema de coordenadas alternativo (8.39)

$$v_{i+(j-1)m} = w_{ij},$$

ilustrado en la tabla 8.1. Habrán mn ecuaciones F_{ij}, y en las coordenadas v, (8.75) representa la ecuación numerada $i + (j-1)m$. La matriz jacobiana DF tendrá el tamaño $mn \times mn$. Usando la tabla 8.1 para traducir a las coordenadas v, se tienen las entradas de la matriz jacobiana

$$\begin{aligned} DF_{i+(j-1)m,\, i+(j-1)m} &= \left(\frac{1}{\Delta t} + \frac{2D}{h^2} + \frac{2D}{k^2} - 1\right) + 2w_{ij} \\ DF_{i+(j-1)m,\, i+1+(j-1)m} &= -\frac{D}{h^2} \\ DF_{i+(j-1)m,\, i-1+(j-1)m} &= -\frac{D}{h^2} \end{aligned}$$

$$DF_{i+(j-1)m,i+jm} = -\frac{D}{k^2}$$

$$DF_{i+(j-1)m,i+(j-2)m} = -\frac{D}{k^2}$$

para los puntos interiores de la malla. Los puntos fuera de la malla se rigen por las condiciones de frontera homogéneas de Neumann

Parte inferior	$(3w_{ij} - 4w_{i,j+1} + w_{i,j+2})/(2k) = 0$	para $j = 1, 1 \le i \le m$
Parte superior	$(3w_{ij} - 4w_{i,j-1} + w_{i,j-2})/(2k) = 0$	para $j = n, 1 \le i \le m$
Lado izquierdo	$(3w_{ij} - 4w_{i+1,j} + w_{i+2,j})/(2h) = 0$	para $i = 1, 1 < j < n$
Lado derecho	$(3w_{ij} - 4w_{i-1,j} + w_{i-2,j})/(2h) = 0$	para $i = m, 1 < j < n$

Las condiciones de Neumann se traducen mediante la tabla 8.1 como

inferior $DF_{i+(j-1)m,i+(j-1)m} = 3,\ DF_{i+(j-1)m,i+jm} = -4,\ DF_{i+(j-1)m,i+(j+1)m} = 1,$
$b_{i+(j-1)m} = 0$ para $j = 1, 1 \le i \le m$

superior $DF_{i+(j-1)m,i+(j-1)m} = 3,\ DF_{i+(j-1)m,i+(j-2)m} = -4,\ DF_{i+(j-1)m,i+(j-3)m} = 1,$
$b_{i+(j-1)m} = 0$ para $j = n, 1 \le i \le m$

izquierdo $DF_{i+(j-1)m,i+(j-1)m} = 3,\ DF_{i+(j-1)m,i+1+(j-1)m} = -4,$
$DF_{i+(j-1)m,i+2+(j-1)m} = 1,$
$b_{i+(j-1)m} = 0$ para $i = 1, 1 < j < n$

derecho $DF_{i+(j-1)m,i+(j-1)m} = 3,\ DF_{i+(j-1)m,i-1+(j-1)m} = -4,$
$DF_{i+(j-1)m,i-2+(j-1)m} = 1,$
$b_{i+(j-1)m} = 0$ para $i = m, 1 < j < n$

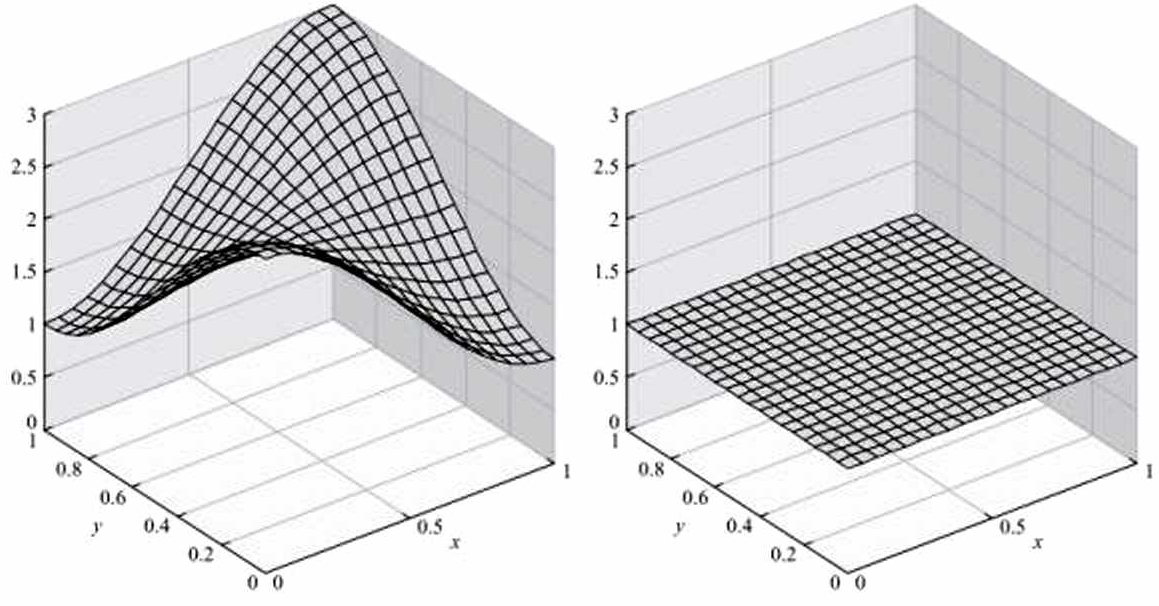


Figura 8.21 Ecuaciones de Fisher con condiciones de frontera de Neumann en un dominio bidimensional. La solución tiende hacia la solución de equilibrio $u(x, y, t) = 1$ a medida que aumenta t. (a) Condición inicial $u(x, y, 0) = 2 + \cos \pi x \cos \pi y$. (b) Solución aproximada después de 5 unidades de tiempo. Tamaños de paso $h = k = \Delta t = 0.05$.

En el siguiente programa, se ejecuta la iteración de Newton. Observe que se ha utilizado el lema 8.11 para dividir las contribuciones a *DF* en términos de primero y segundo grado.

```
% Programa 8.8 Método de la diferencia hacia atrás con la iteración de Newton
%              para la ecuación de Fisher con dominio bidimensional
% entrada: región del espacio [xl xr]x[yb yt], intervalo de tiempo [tb te],
% pasos espaciales M, N en las direcciones x e y, pasos de tiempo tsteps
% salida: malla de solución [x, y, w]
% Uso de ejemplo: [x, y, w] = fisher2d(0,1,0,1,0,5,20,20,100);
function [x,y,w]=fisher2d(xl,xr,yb,yt,tb,te,M,N,tsteps)
f=@(x,y) 2+cos(pi*x).*cos(pi*y)
delt=(te-tb)/tsteps;
D=1;
m=M+1;n=N+1;mn=m*n;
h=(xr-xl)/M;k=(yt-yb)/N;
x=linspace(xl,xr,m);y=linspace(yb,yt,n);
for i=1:m          %Define u inicial
  for j=1:n
      w(i,j)=f(x(i),y(j));
  end
end
for tstep=1:tsteps
 v=[reshape(w,mn,1)];
 wold=w;
 for it=1:3
   b=zeros(mn,1);DF1=zeros(mn,mn);DF2=zeros(mn,mn);
   for i=2:m-1
     for j=2:n-1
       DF1(i+(j-1)*m,i-1+(j-1)*m)=-D/h^2;
       DF1(i+(j-1)*m,i+1+(j-1)*m)=-D/h^2;
       DF1(i+(j-1)*m,i+(j-1)*m)= 2*D/h^2+2*D/k^2-1+1/(1*delt);
       DF1(i+(j-1)*m,i+(j-2)*m)=-D/k^2;DF1(i+(j-1)*m,i+j*m)=-D/k^2;
       b(i+(j-1)*m)=-wold(i,j)/(1*delt);
       DF2(i+(j-1)*m,i+(j-1)*m)=2*w(i,j);
     end
   end
   for i=1:m      % inferior y superior
     j=1; DF1(i+(j-1)*m,i+(j-1)*m)=3;
     DF1(i+(j-1)*m,i+j*m)=-4;DF1(i+(j-1)*m,i+(j+1)*m)=1;
     j=n; DF1(i+(j-1)*m,i+(j-1)*m)=3;
     DF1(i+(j-1)*m,i+(j-2)*m)=-4;DF1(i+(j-1)*m,i+(j-3)*m)=1;
   end
   for j=2:n-1    % izquierda y derecha
     i=1; DF1(i+(j-1)*m,i+(j-1)*m)=3;
     DF1(i+(j-1)*m,i+1+(j-1)*m)=-4;DF1(i+(j-1)*m,i+2+(j-1)*m)=1;
     i=m; DF1(i+(j-1)*m,i+(j-1)*m)=3;
     DF1(i+(j-1)*m,i-1+(j-1)*m)=-4;DF1(i+(j-1)*m,i-2+(j-1)*m)=1;
   end
   DF=DF1+DF2;
   F=(DF1+DF2/2)*v+b;
   v=v-DF\F;
   w=reshape(v(1:mn),m,n);
 end
 mesh(x,y,w');axis([xl xr yb yt tb te]);
 xlabel('x');ylabel('y');drawnow
end
```

El comportamiento dinámico de la ecuación bidimensional de Fisher es similar al de la versión unidimensional de la figura 8.20, donde se vio la convergencia hacia la solución de equilibrio estable en $u(x, t) = 1$. En la figura 8.21(a) se muestran los datos iniciales $f(x, y) = 2 + \cos \pi x \cos \pi y$. En la figura 8.21(b) se muestra la solución después de $t = 5$ unidades de tiempo. La solución se estabiliza rápidamente hacia el equilibrio estable en $u(x, y, t) = 1$. ◄

El matemático Alan Turing (1912-1954), en un artículo de referencia (Turing [1952]), propuso una posible explicación de muchas formas y estructuras que se encuentran en la Biología. Ciertas ecuaciones de reacción-difusión que modelan las concentraciones químicas dieron lugar a interesantes patrones en el espacio, incluyendo franjas y formas hexagonales. Éstas eran vistas como un impresionante ejemplo de orden emergente en la naturaleza, y ahora se conocen como **patrones de Turing**.

Turing descubrió que con la sola adición de un término difusivo a un modelo de una reacción química estable, podía causar que equilibrios estables, constantes en el espacio, como el de la figura 8.21(b), se volvieran inestables. Esta situación llamada **inestabilidad de Turing** provoca una transición en la que los patrones evolucionan hacia una nueva solución de estado estacionario a un estado variable en el espacio. Por supuesto, esto es contrario al efecto de difusión que se ha visto hasta ahora, en el cual las condiciones iniciales se promedian y uniforman con el paso del tiempo.

Un ejemplo interesante de la inestabilidad de Turing se encuentra en el **modelo Brusselator**, propuesto por el químico belga I. Prigogine, a finales de la década de 1960. El modelo consta de dos EDP acopladas, donde cada ecuación representa una de dos especies de reacción química.

► EJEMPLO 8.15 Aplique el método de las diferencias hacia atrás con la iteración de Newton a la ecuación Brusselator con condiciones de frontera homogéneas de Neumann en el cuadrado $[0, 40] \times [0, 40]$:

$$\begin{cases} p_t = D_p \Delta p + p^2 q + C - (K+1)p \\ q_t = D_q \Delta q - p^2 q + Kp \\ \\ p(x, y, 0) = C + 0.1 \quad \text{para } 0 \le x, y \le 40 \\ q(x, y, 0) = K/C + 0.2 \quad \text{para } 0 \le x, y \le 40 \\ u_{\vec{n}}(x, y, t) = 0 \text{ en la frontera del rectángulo, para toda } t \ge 0. \end{cases} \tag{8.76}$$

El sistema de dos ecuaciones acopladas tiene variables p, q, dos coeficientes de difusión D_p, $D_q > 0$ y otros dos parámetros C, $K > 0$. De acuerdo con el ejercicio 5, el modelo Brusselator tiene una solución de equilibrio en $p \equiv C$, $q \equiv K/C$. Se sabe que el equilibrio es estable para valores pequeños del parámetro K, y que cuando

$$K > \left(1 + C\sqrt{\frac{D_p}{D_q}}\right)^2. \tag{8.77}$$

se encuentra una inestabilidad de Turing. Las ecuaciones discretizadas en los puntos interiores de la malla, para $1 < i < m$, $1 < j < n$, son

$$\begin{aligned} &\frac{p_{ij}^t - p_{ij}^{t-\Delta t}}{\Delta t} - \frac{D_p}{h^2}(p_{i+1,j}^t - 2p_{ij}^t + p_{i-1,j}^t) - \frac{D_p}{k^2}(p_{i,j+1}^t - 2p_{ij}^t + p_{i,j-1}^t) \\ &\qquad - (p_{ij}^t)^2 q_{ij}^t - C + (K+1)p_{ij}^t = 0 \\ &\frac{q_{ij}^t - q_{ij}^{t-\Delta t}}{\Delta t} - \frac{D_q}{h^2}(q_{i+1,j}^t - 2q_{ij}^t + q_{i-1,j}^t) - \frac{D_q}{k^2}(q_{i,j+1}^t - 2q_{ij}^t + q_{i,j-1}^t) \\ &\qquad + (p_{ij}^t)^2 q_{ij}^t - Kp_{ij}^t = 0 \end{aligned}$$

Éste es el primer ejemplo que se presenta aquí con dos variables acopladas, p y q. El vector de coordenadas alternativo v tendrá una longitud de $2mn$ y (8.39) se extenderá a

$$v_{i+(j-1)m} = p_{ij} \quad \text{para } 1 \le i \le m, 1 \le j \le n$$
$$v_{mn+i+(j-1)m} = q_{ij} \quad \text{para } 1 \le i \le m, 1 \le j \le n. \tag{8.78}$$

Las condiciones de frontera de Neumann son, en esencia, iguales a las del ejemplo 8.14, ahora para cada variable p y q. Observe que hay términos de primer y tercer grado a diferenciar para el jacobiano DF. Si se usa la tabla 8.1 expandida de una manera directa para cubrir dos variables, y el lema 8.11, se llega al siguiente código de MATLAB:

```
% Programa 8.9 Método de la diferencia hacia atrás con la iteración de Newton
%              para la ecuación Brusselator
% entrada: región del espacio [xl xr]x[yb yt], intervalo de tiempo [tb te],
%    pasos espaciales M, N en las direcciones x y y, pasos de tiempo tsteps
% salida: malla de solución [x, y, w]
% Uso de ejemplo: [x, y, p, q] = brusselator(0,40,0,40,0,20,40,40,20);
function [x,y,p,q]=brusselator(xl,xr,yb,yt,tb,te,M,N,tsteps)
Dp=1;Dq=8;C=4.5;K=9;
fp=@(x,y) C+0.1;
fq=@(x,y) K/C+0.2;
delt=(te-tb)/tsteps;
m=M+1;n=N+1;mn=m*n;mn2=2*mn;
h=(xr-xl)/M;k=(yt-yb)/N;
x=linspace(xl,xr,m);y=linspace(yb,yt,n);
for i=1:m          %Define condiciones iniciales
  for j=1:n
    p(i,j)=fp(x(i),y(j));
    q(i,j)=fq(x(i),y(j));
  end
end
for tstep=1:tsteps
  v=[reshape(p,mn,1);reshape(q,mn,1)];
  pold=p;qold=q;
  for it=1:3
    DF1=zeros(mn2,mn2);DF3=zeros(mn2,mn2);
    b=zeros(mn2,1);
    for i=2:m-1
      for j=2:n-1
        DF1(i+(j-1)*m,i-1+(j-1)*m)=-Dp/h^2;
        DF1(i+(j-1)*m,i+(j-1)*m)= Dp*(2/h^2+2/k^2)+K+1+1/(1*delt);
        DF1(i+(j-1)*m,i+1+(j-1)*m)=-Dp/h^2;
        DF1(i+(j-1)*m,i+(j-2)*m)=-Dp/k^2;
        DF1(i+(j-1)*m,i+j*m)=-Dp/k^2;
        b(i+(j-1)*m)=-pold(i,j)/(1*delt)-C;
        DF1(mn+i+(j-1)*m,mn+i-1+(j-1)*m)=-Dq/h^2;
        DF1(mn+i+(j-1)*m,mn+i+(j-1)*m)= Dq*(2/h^2+2/k^2)+1/(1*delt);
        DF1(mn+i+(j-1)*m,mn+i+1+(j-1)*m)=-Dq/h^2;
        DF1(mn+i+(j-1)*m,mn+i+(j-2)*m)=-Dq/k^2;
        DF1(mn+i+(j-1)*m,mn+i+j*m)=-Dq/k^2;
        DF1(mn+i+(j-1)*m,i+(j-1)*m)=-K;
        DF3(i+(j-1)*m,i+(j-1)*m)=-2*p(i,j)*q(i,j);
        DF3(i+(j-1)*m,mn+i+(j-1)*m)=-p(i,j)^2;
        DF3(mn+i+(j-1)*m,i+(j-1)*m)=2*p(i,j)*q(i,j);
        DF3(mn+i+(j-1)*m,mn+i+(j-1)*m)=p(i,j)^2;
        b(mn+i+(j-1)*m)=-qold(i,j)/(1*delt);
      end
    end
```

```
    for i=1:m   %   condiciones de Neumman inferior y superior
      j=1;DF1(i+(j-1)*m,i+(j-1)*m)=3;
      DF1(i+(j-1)*m,i+j*m)=-4;
      DF1(i+(j-1)*m,i+(j+1)*m)=1;
      j=n;DF1(i+(j-1)*m,i+(j-1)*m)=3;
      DF1(i+(j-1)*m,i+(j-2)*m)=-4;
      DF1(i+(j-1)*m,i+(j-3)*m)=1;
      j=1;DF1(mn+i+(j-1)*m,mn+i+(j-1)*m)=3;
      DF1(mn+i+(j-1)*m,mn+i+j*m)=-4;
      DF1(mn+i+(j-1)*m,mn+i+(j+1)*m)=1;
      j=n;DF1(mn+i+(j-1)*m,mn+i+(j-1)*m)=3;
      DF1(mn+i+(j-1)*m,mn+i+(j-2)*m)=-4;
      DF1(mn+i+(j-1)*m,mn+i+(j-3)*m)=1;
    end
    for j=2:n-1   %condiciones de Neumman izquierda y derecha
      i=1;DF1(i+(j-1)*m,i+(j-1)*m)=3;
      DF1(i+(j-1)*m,i+1+(j-1)*m)=-4;
      DF1(i+(j-1)*m,i+2+(j-1)*m)=1;
      i=m;DF1(i+(j-1)*m,i+(j-1)*m)=3;
      DF1(i+(j-1)*m,i-1+(j-1)*m)=-4;
      DF1(i+(j-1)*m,i-2+(j-1)*m)=1;
      i=1;DF1(mn+i+(j-1)*m,mn+i+(j-1)*m)=3;
      DF1(mn+i+(j-1)*m,mn+i+1+(j-1)*m)=-4;
      DF1(mn+i+(j-1)*m,mn+i+2+(j-1)*m)=1;
      i=m;DF1(mn+i+(j-1)*m,mn+i+(j-1)*m)=3;
      DF1(mn+i+(j-1)*m,mn+i-1+(j-1)*m)=-4;
      DF1(mn+i+(j-1)*m,mn+i-2+(j-1)*m)=1;
    end
    DF=DF1+DF3;
    F=(DF1+DF3/3)*v+b;
    v=v-DF\F;
    p=reshape(v(1:mn),m,n);q=reshape(v(mn+1:mn2),m,n);
  end
  contour(x,y,p');drawnow;
end
```

En la figura 8.22 se muestran las gráficas de contorno de las soluciones de la ecuación Brusselator. En una gráfica de contorno, el trazo de las curvas cerradas nivela los valores de la variable $p(x, y)$. En los modelos, p y q representan las concentraciones químicas que organizan por sí mismas los patrones diferentes que se muestran en las gráficas. ◄

Las ecuaciones de reacción-difusión con una inestabilidad de Turing se utilizan de manera habitual para modelar la formación de patrones en Biología, incluyendo los patrones en las alas de las mariposas, las manchas en la piel de los animales, la pigmentación de los peces y las conchas, entre muchos otros ejemplos. Los patrones de Turing se han encontrado de manera experimental en reacciones químicas tales como la reacción del almidón ACYM (ácido clorito-yoduro-malónico). Los modelos para la glucólisis y las ecuaciones de Gray-Scott para las reacciones químicas están muy relacionados con la ecuación Brusselator.

El uso de ecuaciones de reacción-difusión para estudiar la formación de patrones es una sola dirección entre varias de interés contemporáneo. Las ecuaciones diferenciales parciales no lineales se utilizan para modelar una variedad de fenómenos espaciales y temporales en la ingeniería y las ciencias. Otra clase importante de problemas se describe mediante las ecuaciones de Navier-Stokes, que representan el flujo de fluidos incompresibles. Navier-Stokes se utiliza para fenómenos tan diversos como el modelado de los recubrimientos de película, la lubricación, la dinámica de la sangre en las arterias, el flujo de aire sobre el ala de un avión y la turbulencia del gas estelar. La mejora en las diferencias y los elementos finitos para resolver ecuaciones diferenciales parciales lineales y no lineales constituye una de las áreas de investigación más activas en las ciencias computacionales.

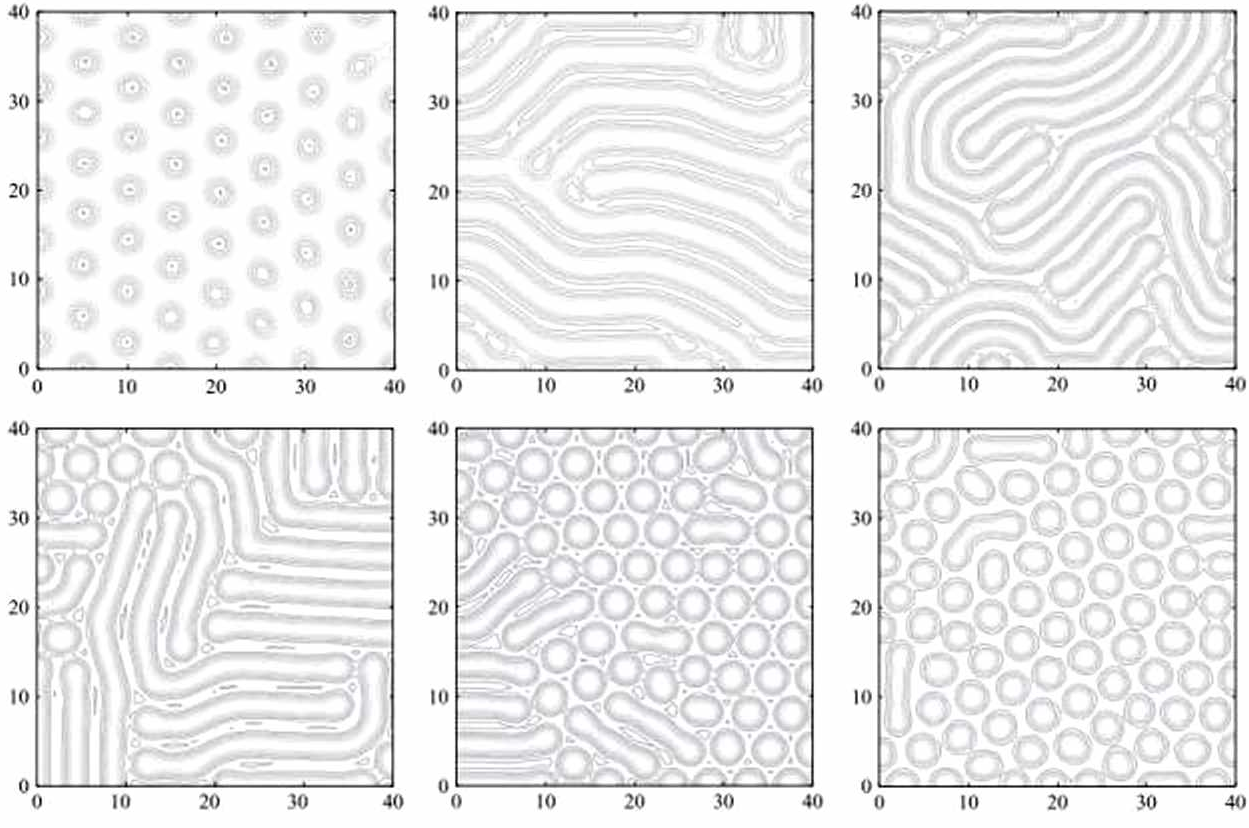


Figura 8.22 Formación de patrones en la ecuación Brusselator. Gráficas de contorno de las soluciones $p(x, y)$ en $t = 2000$ que muestran patrones de Turing. Los parámetros son $D_p = 1$, $D_q = 8$. $C = 4.5$ y (a) $K = 7$ (b) $K = 8$ (c) $K = 9$ (d) $K = 10$ (e) $K = 11$ (f) $K = 12$. Los valores establecidos para las diferencias finitas son $h = k = 0.5$, $\Delta t = 1$.

8.4 Ejercicios

1. Demuestre que para cualquier constante c, la función $u(x, t) = c$ es una solución de equilibrio para la ecuación de Burgers $u_t + uu_x = Du_{xx}$.
2. Demuestre que en un intervalo $[x_l, x_r]$ que no contiene a 0, la función $u(x, t) = x^{-1}$ es una solución invariante en el tiempo de la ecuación de Burgers $u_t + uu_x = -\frac{1}{2}u_{xx}$.
3. Demuestre que la función $u(x, t)$ en (8.68) es una solución de la ecuación de Burgers con las condiciones de frontera de Dirichlet (8.66).
4. Encuentre todas las soluciones de equilibrio estable de la ecuación de Fisher (8.69) cuando $f(u) = u(u - 1)(2 - u)$.
5. Demuestre que la ecuación Brusselator tiene una solución de equilibrio en $p \equiv C$, $q \equiv K/C$.
6. Para la configuración de parámetros $D_p = 1$, $D_q = 8$, $C = 4.5$ de la ecuación Brusselator, ¿con cuáles valores de K es estable la solución de equilibrio $p \equiv C$, $q \equiv K/C$? Consulte los problemas de computadora 5 y 6.

8.4 Problemas de computadora

1. Resuelva la ecuación de Burgers (8.63) en [0, 1] con la condición inicial $f(x) = \text{sen } 2\pi x$ y las condiciones de frontera $l(t) = r(t) = 0$, usando tamaños de paso (a) $h = k = 0.1$ y (b) $h = k = 0.02$.

Grafique las soluciones aproximadas para $0 \le t \le 1$. ¿A qué solución de equilibrio se aproxima la solución a medida que aumenta el tiempo?

2. Resuelva la ecuación de Burgers en el intervalo [0, 1] con condiciones de frontera de Dirichlet homogéneas y la condición inicial dada en (8.66) con parámetros $\alpha = 4$, $\beta = 3$ y $D = 0.2$. Grafique la solución aproximada usando tamaños de paso $h = 0.01$ y $k = 1/16$; asimismo, haga una gráfica log-log del error de la aproximación en $x = 1/2, t = 1$ como una función de k para $k = 2^{-p}$, $p = 4, \ldots, 8$.

3. Resuelva la ecuación de Fisher (8.69) con $f(u) = u(u - 1)(2 - u)$ y condiciones de frontera de Neumann homogéneas, utilizando la condición inicial (a) $f(x) = 1/2 + \cos 2\pi x$ (b) $f(x) = 3/2 - \cos 2\pi x$. Grafique la solución aproximada para $0 \le t \le 2$ para los tamaños de paso $h = k = 0.05$. ¿A qué solución de equilibrio se aproxima la solución a medida que aumenta el tiempo?

4. Resuelva la ecuación de Fisher, con $f(u) = u(u - 1)(2 - u)$ en un dominio espacial de dos dimensiones. Suponga condiciones de frontera de Neumann homogéneas y las condiciones iniciales de (8.73). Grafique la solución aproximada para los tiempos enteros $t = 0, \ldots, 5$ con tamaños de paso $h = k = 0.05$ y $\Delta t = 0.05$. ¿A qué solución de equilibrio se aproxima la solución a medida que aumenta el tiempo?

5. Resuelva las ecuaciones Brusselator para $D_p = 1, D_q = 8, C = 4.5$ y (a) $K = 4$ (b) $K = 5$ (c) $K = 6$ (d) $K = 6.5$. Use condiciones de frontera de Neumann homogéneas y las condiciones iniciales $p(x, y, 0) = 1 + \cos \pi x \cos \pi y$, $q(x, y, 0) = 2 + \cos 2\pi x \cos 2\pi y$, estime el valor mínimo T para el que $|p(x, y, t) - C| < 0.01$ para todo $t > T$.

6. Haga gráficas de contorno de las soluciones $p(x, y, 2000)$ de la ecuación Brusselator para $D_p = 1, D_q = 8, C = 4.5$ y $K = 7.2, 7.4, 7.6$ y 7.8. Utilice tamaños de paso $h = k = 0.5, \Delta t = 1$. Estas gráficas ocupan el rango de la figura 8.22.

Software y lecturas adicionales

Existe una gran variedad de literatura sobre ecuaciones diferenciales parciales y sus aplicaciones en la ciencia y la ingeniería. Entre los libros de texto más recientes con un punto de vista aplicado se encuentran Haberman [2004], Logan [1994], Evans [2002], Strauss [1992] y Gockenbach [2002]. Muchos libros de texto proporcionan información más completa acerca de los métodos numéricos para EDP, como las diferencias y los elementos finitos, incluyendo Strikwerda [1989], Lapidus y Pinder [1982], Hall y Porsching [1990], y Morton y Mayers [1996]. Brenner y Scott [1994], Ames [1992], y Strang y Fix [1973] se enfocan primordialmente en el método del elementos finito.

Las herramientas para EDP en MATLAB son muy recomendables. Se han vuelto muy populares como un complemento de los cursos sobre EDP y matemáticas para ingeniería. Maple tiene un paquete similar llamado PDEtools, aunque también se han desarrollado varios paquetes independientes de software para EDP numéricas de uso general o dirigidos a problemas específicos. ELLPACK (Rice y Boisvert [1984]) y PLTMG (Bank [1998]) son paquetes de uso libre para resolver ecuaciones diferenciales parciales elípticas en regiones generales del plano. Ambos están disponibles en Netlib.

El software que aplica el método del elemento finito incluye los programas gratuitos FEAST (herramientas de solución por elementos finitos), FreeFEM y PETSc (herramientas extensibles portátiles para el cálculo científico), y el software comercial entre los que se encuentran COMSOL, NASTRAN y DIFFPACK. El IMSL contiene la rutina DFPS2H para resolver la ecuación de Poisson en un rectángulo y DFPS3H en una caja tridimensional. Estos métodos se basan en diferencias finitas.

La biblioteca NAG contiene varias rutinas para diferencias y elementos finitos. El programa D03EAF resuelve la ecuación de Laplace en dos dimensiones por medio de un método de ecuaciones integrales; D03EEF utiliza una fórmula de diferencias finitas de siete puntos y maneja muchos tipos de condiciones de frontera. Las rutinas D03PCF y D03PFF manejan ecuaciones parabólicas e hiperbólicas, respectivamente.

CAPÍTULO

9

Números aleatorios y sus aplicaciones

El movimiento browniano es un modelo de comportamiento aleatorio, propuesto por Robert Brown en 1827. Su interés inicial fue entender el movimiento errático de las partículas de polen que flotan en la superficie del agua, impulsadas por las moléculas cercanas. Las aplicaciones del modelo se han extendido más allá del contexto original.

Los analistas financieros actuales conciben de la misma manera los precios de los activos, dada la incertidumbre en el comportamiento de los mercados financieros. En 1973, Fischer Black y Myron Scholes usaron de manera novedosa el movimiento browniano exponencial para proporcionar valoraciones exactas de las opciones bursátiles. Reconocida de inmediato como una innovación importante, la fórmula de Black-Scholes se programó en algunas de las primeras calculadoras portátiles diseñadas para ser usadas en el piso de remates de Wall Street. Este trabajo fue galardonado con el Premio Nobel de Economía en 1997 y sigue estando presente en la teoría y la práctica económicas.

Comprobación en la realidad En la página 464 se explora la famosa fórmula de la simulación de Monte Carlo.

Los tres capítulos previos trataron acerca de los modelos deterministas regidos por ecuaciones diferenciales. Dadas las condiciones iniciales y de frontera adecuadas, la solución es matemáticamente cierta y puede determinarse mediante los métodos numéricos apropiados con la exactitud requerida. Por su parte, un modelo estocástico incluye la incertidumbre como parte de su definición.

La simulación computacional de un sistema estocástico requiere la generación de números aleatorios para simular condiciones óptimas. Este capítulo comienza con algunos hechos fundamentales acerca de los números aleatorios y su uso en la simulación. La segunda sección cubre uno de los usos más importantes de los números aleatorios, la simulación de Monte Carlo, y la tercera sección presenta las caminatas aleatorias y el movimiento browniano. En la última sección se cubren las ideas básicas del cálculo estocástico, incluyendo muchos ejemplos estándar de ecuaciones diferenciales estocásticas (EDE) que han demostrado su utilidad en la física, la biología y las finanzas. Los métodos computacionales para las EDE se basan en los métodos para resolver EDO desarrollados en el capítulo 7, pero extendidos para incluir términos en condiciones óptimas.

En este capítulo se requerirán en ocasiones algunos conceptos básicos de probabilidad. Estos prerrequisitos adicionales, como el valor esperado, la varianza y la independencia de variables aleatorias, son importantes en las secciones 9.2 a 9.4.

9.1 NÚMEROS ALEATORIOS

Cualquier persona tiene una noción sobre lo que son los números aleatorios, pero el concepto es muy difícil de definir con precisión. Tampoco resulta fácil encontrar métodos sencillos y eficaces para producirlos. Por supuesto, con computadoras que funcionen de acuerdo con reglas deterministas prescritas, asignadas por el programador, se puede lograr un programa que produzca números en verdad aleatorios. Aquí se estudiará la forma de producir números pseudoaleatorios, lo cual es tan sólo una manera de decir que se considerarán programas deterministas que funcionen de la misma manera cada vez, y que produzcan series de números que se vean tan aleatorios como sea posible.

El objetivo de un generador de números aleatorios es que los números de salida sean independientes e idénticamente distribuidos. Por "independiente" se entiende que cada nuevo número x_n no debe depender del (ser más o menos probable debido a) número anterior x_{n-1}, o, de hecho, de todos los números anteriores $x_{n-1}, x_{n-2}, \ldots$ Por "idénticamente distribuidos" se entiende que si se traza el histograma de x_n con muchas repeticiones diferentes de la generación de números aleatorios, debe verse igual que el histograma de x_{n-1}. En otras palabras, independiente significa que x_n es independiente de x_{n-1}, x_{n-2}, etcétera e idénticamente distribuido significa que la distribución de x_n es independiente de n. El histograma deseado, o la distribución deseada, puede ser una distribución uniforme de números reales entre 0 y 1, o puede ser tan sofisticada, como una distribución normal.

Por supuesto, la parte de la independencia de la definición de los números aleatorios está en desacuerdo con los métodos prácticos computacionales para generar números aleatorios, que producen series de números completamente predecibles y repetibles. De hecho, la repetición puede ser muy útil para algunos fines de la simulación. El truco es hacer que los números *parezcan* independientes entre sí, a pesar de que el método de generación pueda ser cualquier cosa menos independiente. El término número **pseudoaleatorio** está reservado para los números generados en una situación determinista que se esfuerzan por ser aleatorios en el sentido de independencia y distribución idéntica.

El hecho de que se hayan usado medios demasiado dependientes para producir algo que pretende ser independiente explica por qué no existe un generador de números aleatorios perfecto basado en software de propósito general. Como John von Neumann dijo en 1951: "Cualquiera que considere métodos aritméticos para producir dígitos aleatorios está, por supuesto, en estado de pecado". La esperanza principal es que la hipótesis particular de que el usuario desee probar mediante el uso de números aleatorios sea insensible a las dependencias y deficiencias del generador elegido.

Los números aleatorios son representantes elegidos a partir de una distribución de probabilidad fija. Existen muchas elecciones posibles de la distribución. Para mantener al mínimo los prerrequisitos, en este libro se restringirá la atención a dos posibilidades: la distribución uniforme y la distribución normal.

9.1.1 Números pseudoaleatorios

El conjunto más simple de números aleatorios es la distribución uniforme en el intervalo [0, 1]. Estos números corresponden a ponerse una venda en los ojos y elegir números del intervalo, sin preferencia por ningún área del intervalo en particular. Cada número real en el intervalo tiene la misma probabilidad de ser elegido. ¿Cómo puede producirse una serie de números de ese tipo con un programa de computadora?

A continuación se presenta un primer intento de producir números (pseudo-) aleatorios uniformes en [0, 1]. Elija un entero de inicio $x_0 \neq 0$, llamado la **semilla**. Después produzca la secuencia de números u_i de acuerdo con la iteración

$$x_i = 13x_{i-1} \pmod{31}$$
$$u_i = \frac{x_i}{31}, \tag{9.1}$$

es decir, multiplique la x_{i-1} por 13, evalúe el módulo 31 y divida entre 31 para obtener el siguiente número pseudoaleatorio. La secuencia resultante se repetirá sólo después de ejecutar la iteración a través de los 30 números distintos de cero 1/31, ... , 30/31. En otras palabras, el **periodo** de este generador de números aleatorios es 30. No hay nada que parezca aleatorio en esta secuencia de números. Una vez que la semilla se elige, la secuencia se cicla a través de los 30 posibles números en un orden predeterminado. Los primeros generadores de números aleatorios seguían la misma lógica, aunque con un periodo más amplio.

Con $x_0 = 3$ como semilla aleatoria, aquí están los primeros 10 números generados por el método anterior:

x	u
8	0.2581
11	0.3548
19	0.6129
30	0.9677
18	0.5806
17	0.5484
4	0.1290
21	0.6774
25	0.8065
15	0.4839

Se inicia con $3 * 13 = 39 \rightarrow 8 \pmod{31}$, de manera que el número aleatorio uniforme es $8/31 \approx 0.2581$. El segundo número aleatorio es $8 * 13 = 104 \rightarrow 11 \pmod{31}$, obteniéndose $11/31 \approx 0.3548$, y así sucesivamente, hasta ejecutar la iteración a través de los 30 números aleatorios posibles.

Éste es un ejemplo de un generador de números aleatorios del tipo más básico.

DEFINICIÓN 9.1 Un **generador de congruencia lineal (GCL)** tiene la forma

$$x_i = ax_{i-1} + b \pmod{m}$$
$$u_i = \frac{x_i}{m}, \tag{9.2}$$

para el **multiplicador** a, el **desplazamiento** b y el **módulo** m. ❐

En el generador anterior, $a = 13$, $b = 0$ y $m = 31$. En los dos ejemplos siguientes, se conservará $b = 0$. La sabiduría convencional es que b distinto de cero añade una pequeña complicación adicional al generador de números aleatorios.

Una aplicación de los números aleatorios es aproximar el promedio de una función al sustituir números aleatorios del intervalo de interés. Ésta es la forma más simple de la técnica de Monte Carlo que se discutirá con más detalle en la siguiente sección.

► EJEMPLO 9.1 Aproxime el área bajo la curva $y = x^2$ en $[0, 1]$.

Por definición, el valor medio de una función en $[a, b]$ es

$$\frac{1}{b-a}\int_a^b f(x)\,dx,$$

por lo que el área en cuestión es exactamente el valor medio de $f(x) = x^2$ en [0, 1]. Este valor medio puede aproximarse mediante el promedio de los valores de la función en puntos aleatorios del intervalo, como se muestra en la figura 9.1. El promedio de la función

$$\frac{1}{10}\sum_{i=1}^{10} f(u_i)$$

para los primeros 10 números aleatorios uniformes generados por medio del método actual es 0.350, no demasiado lejos de la respuesta correcta, 1/3. Si se usan los 30 números aleatorios en el promedio, resulta la estimación mejorada 0.328.

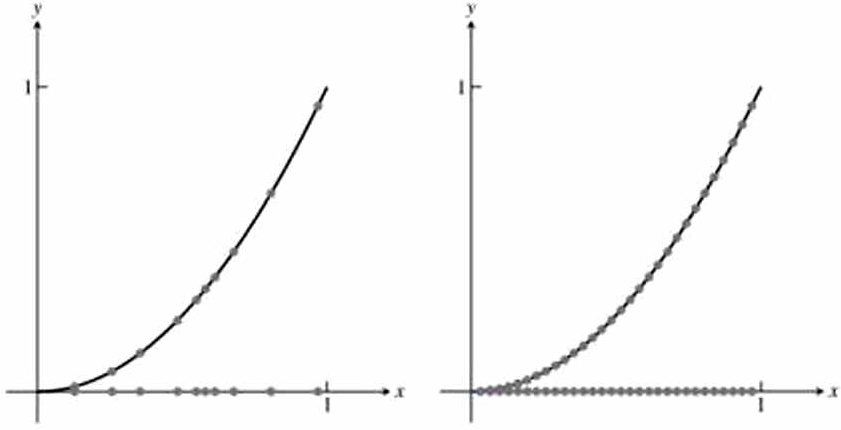


Figura 9.1 Promedio de una función mediante el uso de números aleatorios. (a) Los primeros 10 números aleatorios del generador elemental (9.1) con semilla $x_0 = 3$ dan el promedio 0.350. (b) Al utilizar los 30 números se obtiene un promedio más exacto de 0.328.

◀

La aplicación del ejemplo 9.1 se denominará problema de Monte Carlo tipo 1, puesto que se reduce al promedio de una función. Observe que se han agotado los 30 números al azar que puede proporcionar el generador (9.1). Si se requiere más precisión, se necesitan más números. Es posible quedarse con el modelo GCL, pero es necesario incrementar el multiplicador a y el módulo m.

Park y Miller [1998] propusieron un generador de congruencia lineal que suele denominarse generador "mínimo estándar", porque es casi tan bueno como es posible y con un código muy simple. Este generador de números aleatorios se utilizó en MATLAB versión 4 en la década de 1990.

Generador de números aleatorios mínimo estándar

$$\begin{aligned} x_i &= ax_{i-1} \pmod m \\ u_i &= \frac{x_i}{m}, \end{aligned} \tag{9.3}$$

donde $m = 2^{31} - 1, a = 7^5 = 16807$, y $b = 0$.

Un número entero de la forma $2^p - 1$ que es un número primo, donde p es un entero, se llama **primo de Mersenne**. Euler descubrió este número primo de Mersenne en 1772. El tiempo de repetición del generador de números aleatorios mínimo estándar es el máximo posible $2^{31} - 2$, lo que significa que se necesitan todos los números enteros no nulos por debajo del máximo antes de repetir, siempre y cuando la semilla sea distinta de cero. Esto es aproximadamente 2×10^9 números, tal vez suficiente para el siglo XX, pero no es suficiente ahora que las computadoras ejecutan de manera rutinaria muchos ciclos de reloj por segundo.

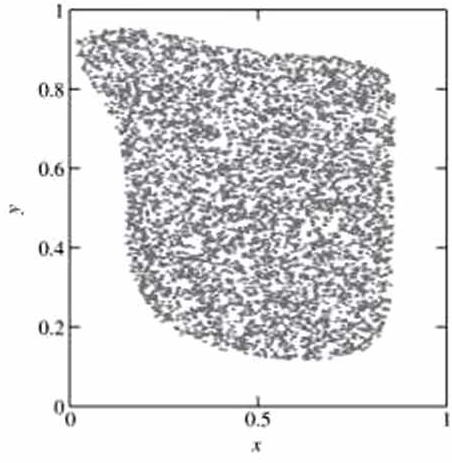


Figura 9.2 Cálculo de área de Monte Carlo. De 10 000 pares aleatorios en [0, 1] × [0, 1], se grafican los que satisfacen la desigualdad del ejemplo 9.2. La proporción de pares aleatorios graficados es una aproximación al área.

► EJEMPLO 9.2 Encuentre el área del conjunto de puntos (x, y) que satisfacen

$$4(2x - 1)^4 + 8(2y - 1)^8 < 1 + 2(2y - 1)^3(3x - 2)^2.$$

Éste se denominará un problema de Monte Carlo tipo 2. No hay una manera clara de describir esta área como el valor promedio de una función de una variable, puesto que no puede resolverse para y. Sin embargo, dada una pareja de puntos (x, y), puede comprobarse con facilidad si pertenece o no al conjunto. Se igualará el área deseada con la probabilidad de que un par aleatorio dado $(x, y) = (u_i, u_{i+1})$ pertenezca al conjunto y se intentará aproximar la probabilidad.

En la figura 9.2 se muestra esta idea puesta en práctica con 10 000 pares aleatorios generados mediante el GCL mínimo estándar. La proporción de pares en la unidad cuadrada $0 \leq x, y \leq 1$ que satisfacen la desigualdad, y se representan en la figura, es 0.547, que se tomará como una aproximación al área. ◄

Aunque se ha hecho una distinción entre los dos tipos de problemas de Monte Carlo, no hay una frontera firme entre ellos. Lo que tienen en común es que ambos calculan el promedio de una función. Esto es explícito en el ejemplo anterior del "tipo 1". En el ejemplo del "tipo 2" se trata de calcular el promedio de la **función característica** del conjunto, la función que toma el valor 1 para los puntos en el interior del conjunto y 0 para los puntos que están fuera de éste. El contraste principal aquí es que a diferencia de la función $f(x) = x^2$ del ejemplo 9.1, la función característica de un conjunto es discontinua (hay una transición abrupta en la frontera del conjunto). También es posible imaginar fácilmente combinaciones de los tipos 1 y 2 (vea el problema de computadora 8).

Uno de los generadores de números aleatorios menos reconocido es el generador `randu`, utilizado en muchas de las primeras computadoras IBM y exportado de ahí a otros ordenadores. Los rastros de este generador pueden encontrarse con facilidad en internet mediante cualquier buscador, por lo que aparentemente sigue en uso.

El generador `randu`

$$\begin{aligned} x_i &= a x_{i-1} \pmod{m} \\ u_i &= \frac{x_i}{m}, \end{aligned} \tag{9.4}$$

donde $a = 65539 = 2^{16} + 3$ y $m = 2^{31}$.

La semilla aleatoria $x_0 \neq 0$ se elige arbitrariamente. El módulo no primo se seleccionó en un principio para hacer que la operación del módulo fuera lo más rápida posible, y el multiplicador se seleccionó primordialmente porque su representación binaria era simple. El problema grave con este generador es que desobedece de manera evidente al postulado de independencia para los números aleatorios. Tenga en cuenta que

$$\begin{aligned} a^2 - 6a &= (2^{16} + 3)^2 - 6(2^{16} + 3) \\ &= 2^{32} + 6 \cdot 2^{16} + 9 - 6 \cdot 2^{16} - 18 \\ &= 2^{32} - 9. \end{aligned}$$

Donde, $a^2 - 6a + 9 = 0 \pmod{m}$, por lo que

$$\begin{aligned} x_{i+2} - 6x_{i+1} + 9x_i &= a^2 x_i - 6ax_i + 9x_i \pmod{m} \\ &= 0 \pmod{m}. \end{aligned}$$

Al dividir entre m se obtiene

$$u_{i+2} = 6u_{i+1} - 9u_i \pmod{1}. \tag{9.5}$$

El problema no es que u_{i+2} sea predecible a partir de los dos números generados antes. Por supuesto, será predecible incluso a partir de un número anterior, debido a que el generador es determinístico. El problema reside en los pequeños coeficientes de la relación (9.5), que hacen que la correlación entre los números aleatorios sea muy notable. En la figura 9.3(a) se muestra una gráfica de 10 000 números aleatorios generados por `randu` y graficados en tripletas (u_i, u_{i+1}, u_{i+2}). Una consecuencia de la relación (9.5) es que todas las tripletas de números aleatorios se encuentran en uno de 15 planos, como puede verse en la figura. De hecho, $u_{i+2} - 6u_{i+1} + 9u_i$ debe ser un entero, y las únicas posibilidades son los números enteros entre -5, en el caso donde u_{i+1} es grande y u_i, u_{i+2} son pequeñas, y $+9$, en el caso contrario. Los planos $u_{i+2} - 6u_{i+1} + 9u_i = k$, para $-5 \leq k \leq 9$, son los 15 planos que se observan en la figura 9.3. En el ejercicio 5 se le pide analizar otro generador de números aleatorios muy conocido con una deficiencia similar.

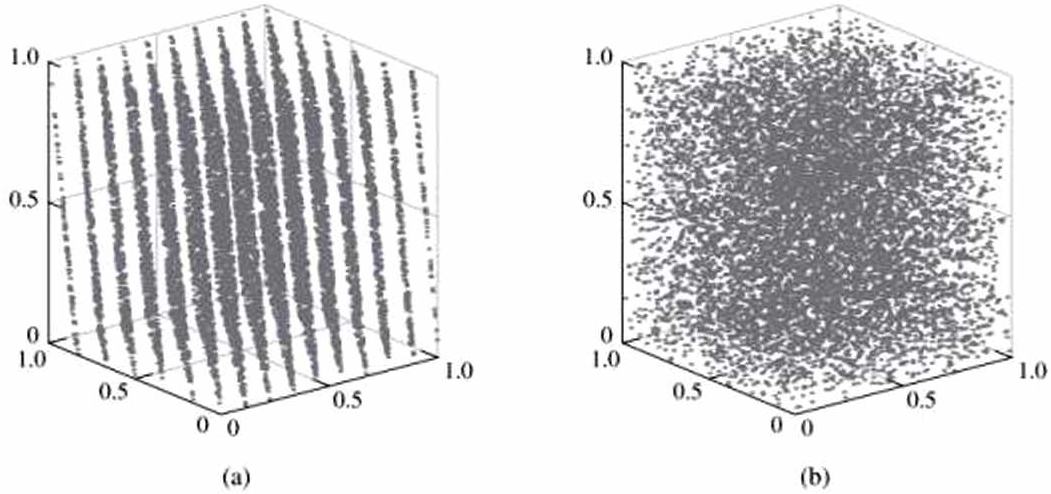


Figura 9.3 Comparación de dos generadores de números aleatorios. Se representan diez mil tripletas (u_i, u_{i+1}, u_{i+2}) generadas mediante (a) `randu` y (b) el generador mínimo estándar.

El GCL mínimo estándar no sufre de este problema, al menos en el mismo grado. Como m y a en (9.3) son relativamente primos, las relaciones entre u_i sucesivas con coeficientes pequeños, como la de (9.5), son mucho más difíciles de conseguir y las correlaciones entre los tres números

aleatorios sucesivos de este generador son mucho más complicadas. Esto puede verse en la figura 9.3(b), que compara una gráfica de 10 000 números aleatorios generados por el generador de números aleatorios mínimo estándar con una gráfica similar obtenida de `randu`.

► **EJEMPLO 9.3** Use `randu` para aproximar el volumen de la pelota de radio 0.4 centrado en (1/3, 1/3, 1/2).

Aunque la pelota tiene un volumen diferente de cero, un intento sencillo para aproximar el volumen con `randu` surge con el valor 0. El método de Monte Carlo sirve para generar puntos de manera aleatoria en un cubo tridimensional unitario y contar la proporción de puntos generados que se encuentran en la pelota como el volumen aproximado.

El punto (1/3, 1/3, 1/2) se encuentra a la mitad del camino entre los planos $9x - 6y + z = 1$ y $9x - 6y + z = 2$, a una distancia de $1/(2\sqrt{118}) \approx 0.046$ desde cada plano. Por lo tanto, la generación del punto tridimensional $(x, y, z) = (u_i, u_{i+1}, u_{i+2})$ con `randu` nunca resultará en un punto contenido en la pelota especificada. Las aproximaciones de Monte Carlo para este sencillo problema serán muy poco exitosas debido a la elección del generador de números aleatorios. De manera sorprendente, las dificultades de este tipo pasaron inadvertidas durante las décadas de 1960 y 1970, cuando este generador era la base de muchas de las simulaciones realizadas por computadora. ◄

Los números aleatorios en las versiones actuales de MATLAB ya no se generan por GCL. Comenzando con MATLAB 5, se ha utilizado un generador de Fibonacci demorado, que desarrolló G. Marsaglia *et al.* [1991], en el comando `rand`. Se usan todos los posibles números de punto flotante entre 0 y 1. MATLAB afirma que el periodo de este método es mayor a 2^{1400}, que es mucho mayor que el número total de pasos ejecutados por todos los programas de MATLAB desde su creación.

Hasta ahora se ha puesto énfasis en la generación de números pseudoaleatorios en el intervalo [0, 1]. Para generar una distribución uniforme de números aleatorios en el intervalo general $[a, b]$, se necesita ampliar en $b - a$ la longitud del intervalo nuevo. Así, cada número aleatorio r generado en [0, 1], debe sustituirse por $(b - a)r + a$.

Esto puede hacerse para cada dimensión en forma independiente. Por ejemplo, para generar un punto aleatorio uniforme en el rectángulo $[1, 3] \times [2, 8]$ en el plano xy, genere el par de números aleatorios uniformes r_1, r_2 y después utilice $(2r_1 + 1, 6r_2 + 2)$ como el punto aleatorio.

9.1.2 Números aleatorios exponenciales y normales

Una variable aleatoria **exponencial** V elige números positivos de acuerdo con la **función de distribución de probabilidad** $p(x) = ae^{-ax}$ para $a > 0$. En otras palabras, un histograma de números aleatorios exponenciales $r_1, \ldots, r_n$ tenderá a $p(x)$ cuando $n \to \infty$.

Si se utiliza un generador de números aleatorios uniformes de la sección anterior, resulta bastante fácil generar números aleatorios exponenciales. La **función de la distribución acumulada** es

$$P(x) = \text{Prob}(V \leq x) = \int_0^x p(x)dx = 1 - e^{-ax}.$$

La idea principal es elegir la variable aleatoria exponencial de modo que la $\text{Prob}(V \leq x)$ sea uniforme entre 0 y 1. Es decir, dado un número aleatorio uniforme u, establezca

$$u = \text{Prob}(V \leq x) = 1 - e^{-ax}$$

y resuelva para x, lo que resulta en

$$x = \frac{-\ln(1 - u)}{a}. \tag{9.6}$$

Por lo tanto, la fórmula (9.6) genera números aleatorios exponenciales, utilizando números aleatorios uniformes u como entradas.

Esta idea funciona en lo general. Sea $P(x)$ la función de distribución acumulada de la variable aleatoria que debe generarse. Sea $Q(x) = P^{-1}(x)$ la función inversa. Si $U[0, 1]$ indica los números aleatorios uniformes en $[0, 1]$, entonces $Q(U[0, 1])$ generará las variables aleatorias requeridas. Todo lo que resta es encontrar maneras de hacer la evaluación de Q lo más eficiente posible.

La **normal** estándar, o variable aleatoria **gaussiana**, $N(0, 1)$ elige números reales de acuerdo con la función de distribución de probabilidad

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}},$$

que tiene la forma de la famosa curva de campana. La variable $N(0, 1)$ tiene media 0 y varianza 1. De manera más general, la variable aleatoria normal $N(\mu, \sigma^2) = \mu + \sigma N(0, 1)$ tiene media μ y varianza σ^2. Dado que esta variable es sólo una versión reducida de la variable aleatoria normal estándar $N(0, 1)$, se estudiarán los métodos para generar esta última.

A pesar de que podría aplicarse directamente la inversa de la función de distribución acumulada como acaba de señalarse, resulta más eficiente generar dos números aleatorios normales a la vez. La distribución normal estándar bidimensional tiene una función de distribución de la probabilidad $p(x, y) = (1/2\pi)e^{-(x^2+y^2)/2}$, o $p(r) = (1/2\pi)e^{-r^2/2}$ en coordenadas polares. Como $p(r)$ tiene simetría polar, sólo es necesario generar la distancia radial r de acuerdo con $p(r)$ y después elegir un ángulo arbitrario θ uniforme en $[0, 2\pi]$. Dado que $p(r)$ es una distribución exponencial para r^2 con el parámetro $a = 1/2$, genera r mediante

$$r^2 = \frac{-\ln(1 - u_1)}{1/2}$$

de la fórmula (9.6), donde u_1 es un número aleatorio uniforme. Entonces

$$\begin{aligned} n_1 &= r\cos 2\pi u_2 = \sqrt{-2\ln(1 - u_1)}\cos 2\pi u_2 \\ n_2 &= r\,\text{sen}\, 2\pi u_2 = \sqrt{-2\ln(1 - u_1)}\,\text{sen}\, 2\pi u_2 \end{aligned} \tag{9.7}$$

es un par de números aleatorios normales independientes, donde u_2 es un segundo número aleatorio uniforme. Tenga en cuenta que $1 - u_1$ puede reemplazarse por u_1 en la fórmula, puesto que la distribución $U[0, 1]$ no experimenta cambios después de restarle 1. Éste es el **método de Box-Muller** (Box y Muller [1958]) para generar números aleatorios normales. Para cada par se requieren evaluaciones de raíz cuadrada, logaritmo, coseno y seno.

Para obtener una versión más eficiente de Box-Muller, u_1 puede generarse de una manera diferente. Elija x_1, x_2 de $U[0, 1]$, y defina $u_1 = x_1^2 + x_2^2$ si la expresión es menor que 1. Si no es así, deseche x_1 y x_2 y empiece de nuevo. Observe que u_1 elegida de esta manera es $U[0, 1]$. La ventaja es que puede utilizarse $u_2 = \arctan x_2/x_1$, el ángulo que forma el segmento de línea que une el origen con el punto (x_1, x_2), porque es evidente que u_2 es uniforme en $[0, 2\pi]$. Como $\cos 2\pi u_2 = x_1/u_1$ y $\text{sen}\, 2\pi u_2 = x_2/u_1$, la fórmula (9.7) se traduce en

$$\begin{aligned} n_1 &= x_1\sqrt{\frac{-2\ln(u_1)}{u_1}} \\ n_2 &= x_2\sqrt{\frac{-2\ln(u_1)}{u_1}}, \end{aligned} \tag{9.8}$$

donde $u_1 = x_1^2 + x_2^2$, que se calcula sin necesidad de evaluar el seno y el coseno como en (9.7).

La versión revisada del método de Box-Muller es un **método de rechazo**, porque algunas entradas no se utilizan. Al comparar el área del cuadrado unitario $[-1, 1] \times [-1, 1]$ con el círculo unitario, el rechazo se producirá $(4 - \pi)/4 \approx 21\%$ de las veces. Éste es un precio aceptable a pagar para evitar las evaluaciones del seno y coseno.

Existen métodos más sofisticados para generar números aleatorios normales. Para mayores detalles consulte Knuth [1997]. Por ejemplo, el comando `randn` de MATLAB utiliza el algoritmo "ziggurat" de Marsaglia y Tsang [2000], que es en esencia una forma muy eficiente de invertir la función de distribución acumulada.

9.1 Ejercicios

1. Encuentre el periodo del generador de congruencia lineal definido por (a) $a = 2, b = 0, m = 5$ (b) $a = 4, b = 1, m = 9$.

2. Encuentre el periodo del GCL definido por $a = 4, b = 0, m = 9$. ¿El periodo depende de la semilla?

3. Aproxime el área bajo la curva $y = x^2$ para $0 \leq x \leq 1$, usando el GCL con (a) $a = 2, b = 0, m = 5$ (b) $a = 4, b = 1, m = 9$.

4. Aproxime el área bajo la curva $y = 1 - x$ para $0 \leq x \leq 1$, utilizando el GCL con (a) $a = 2, b = 0, m = 5$ (b) $a = 4, b = 1, m = 9$.

5. Considere el generador de números aleatorios RANDNUM-CRAY, que se utilizó en la Cray X-MP, una de las primeras supercomputadoras. Este GCL utilizaba $m = 2^{48}$, $a = 2^{24} + 3$ y $b = 0$. Demuestre que $u_{i+2} = 6u_{i+1} - 9u_i \pmod 1$. ¿Esto le parece preocupante? Consulte los problemas de computadora 9 y 10.

9.1 Problemas de computadora

1. Implemente el generador de números aleatorios mínimo estándar y encuentre la aproximación de Monte Carlo al volumen del ejemplo 9.3. Use 10^6 puntos tridimensionales con semilla $x_0 = 1$. ¿Qué tan cerca está su aproximación de la respuesta correcta?

2. Implemente `randu` y encuentre la aproximación de Monte Carlo al volumen del ejemplo 9.3, como en el problema de computadora 1. Compruebe que ningún punto (ui, u_{i+1}, u_{i+2}) entra en la pelota dada.

3. (a) Utilizando el cálculo, encuentre el área delimitada por las dos parábolas $P_1(x) = x^2 - x + 1/2$ y $P_2(x) = -x^2 + x + 1/2$. (b) Estime el área como una simulación de Monte Carlo tipo 1, encontrando el valor promedio de $P_2(x) - P_1(x)$ en $[0, 1]$. Encuentre las estimaciones para $n = 10^i$ para $2 \leq i \leq 6$. (c) Haga lo mismo que en (b), pero estime como en un problema de Monte Carlo tipo 2: Encuentre la proporción de puntos en el cuadrado $[0, 1] \times [0, 1]$ que se encuentra entre las parábolas. Compare la eficiencia de los dos enfoques de Monte Carlo.

4. Realice los pasos del problema de computadora 3 para el subconjunto del primer cuadrante limitado por los polinomios $P_1(x) = x^3$ y $P_2(x) = 2x - x^2$.

5. Use $n = 10^4$ puntos pseudoaleatorios para estimar el área interior de las elipses

   (a) $13x^2 + 34xy + 25y^2 \leq 1$ en $-1 \leq x, y \leq 1$ y

   (b) $40x^2 + 25y^2 + y + 9/4 \leq 52xy + 14x$ en $0 \leq x, y \leq 1$. Compare su estimación con las áreas correctas (a) $\pi/6$ y (b) $\pi/18$, y reporte el error de la estimación. Repita con $n = 10^6$ puntos y compare los resultados.

6. Use $n = 10^4$ puntos pseudoaleatorios para estimar el volumen interior del elipsoide definido por $2 + 4x^2 + 4z^2 + y^2 \leq 4x + 4z + y$, contenido en el cubo unitario $0 \leq x, y, z \leq 1$. Compare su estimación con el volumen correcto $\pi/24$ y registre el error. Repita la estimación con $n = 10^6$ puntos.

7. (a) Utilice el cálculo para evaluar la integral $\int_0^1 \int_{x^2}^{\sqrt{x}} xy\,dy\,dx$. (b) Utilice $n = 10^6$ pares en el cuadrado unitario $[0, 1] \times [0, 1]$ para estimar la integral como un problema de Monte Carlo tipo 1. (Promedio de la función que es igual a xy si (x, y) está en el dominio de integración y 0 si no lo está).

8. Use 10^6 pares aleatorios en el cuadrado unitario para estimar $\int_A xy\,dx\,dy$, donde A es el área descrita en el ejemplo 9.2.

9. Implemente el generador de números aleatorios cuestionado del ejercicio 5, y dibuje una gráfica similar a la figura 9.3.

10. Diseñe un problema de aproximación por Monte Carlo que contraste por completo con el generador RANDNUM-CRAY del ejercicio 5, siguiendo las ideas del ejemplo 9.3.

9.2 SIMULACIÓN DE MONTE CARLO

Ya se han visto ejemplos de dos tipos de simulación de Monte Carlo. En esta sección se explora el rango de problemas que son adecuados para esta técnica y se analizan algunas de las mejoras que hacen que funcione mejor, incluyendo los números cuasialeatorios. En esta sección deberá usarse el lenguaje de las variables aleatorias y los valores esperados.

9.2.1 Leyes de potencia para la estimación de Monte Carlo

En principio es necesario comprender la velocidad de convergencia de la simulación de Monte Carlo. ¿Con qué rapidez disminuye el error de estimación a medida que crece el número n de puntos usados en el cálculo? Esta pregunta es similar a las preguntas de convergencia del capítulo 6 para los métodos de cuadratura y en los capítulos 7, 8 y 9 para los solucionadores de ecuaciones diferenciales. En los casos anteriores, se plantean como preguntas acerca del error frente al tamaño de paso. La reducción del tamaño de paso es análoga a la adición de números aleatorios en las simulaciones de Monte Carlo.

Considere el método de Monte Carlo tipo 1 como el cálculo de la media de una función usando muestras aleatorias, que después se multiplica por el volumen de la región de integración. El cálculo de la media de una función puede verse como el cálculo de la media de una distribución de probabilidad dada por la función. Se utilizará la notación $E(X)$ para el valor esperado de la variable aleatoria X. La **varianza** de una variable aleatoria X es $E[(X - E(X))^2]$, y la **desviación estándar** de X es la raíz cuadrada de su varianza. El error esperado en la estimación de la media disminuirá con el número n de puntos aleatorios, de la siguiente manera:

Monte Carlo tipo 1 o tipo 2 con números pseudoaleatorios.

$$\text{Error} \propto n^{-\frac{1}{2}} \tag{9.9}$$

Para entender esta fórmula, vea la integral como el volumen del dominio multiplicado por el valor medio A de la función sobre el dominio. Tenga en cuenta las mismas variables aleatorias X_i correspondientes a una evaluación de la función en un punto aleatorio. Entonces el valor medio es el valor esperado de la variable aleatoria $Y = (X_1 + \cdots + X_n)/n$, o bien

$$E\left[\frac{X_1 + \cdots + X_n}{n}\right] = nA/n = A,$$

ANOTACIÓN

Convergencia Una estimación de Monte Carlo tipo 1 hace algo muy similar al método compuesto del punto medio del capítulo 5. Se ha encontrado que el error no es proporcional al tamaño de paso h, lo que equivale aproximadamente a $1/n$ cuando se toma en cuenta el número de evaluaciones de la función. Esto es más eficiente que la ley de la potencia de la raíz cuadrada de Monte Carlo.

Sin embargo, Monte Carlo tiene éxito con problemas como el del ejemplo 9.2. Aunque la convergencia al valor correcto todavía es lenta, no está clara la manera de plantear el problema como un problema tipo 1, con el fin de aplicar las técnicas del capítulo.

y la varianza de Y es

$$E\left[\left(\frac{X_1 + \cdots + X_n}{n} - A\right)^2\right] = \frac{1}{n^2}\sum E[(X_i - A)^2] = \frac{1}{n^2} n\sigma^2 = \frac{\sigma^2}{n},$$

donde σ es la varianza original de cada X_i. Por lo tanto, la desviación estándar de Y disminuye con $\sigma/\sqrt{n}$. Este argumento se aplica a las simulaciones de Monte Carlo de los tipos 1 y 2.

► **EJEMPLO 9.4** Encuentre estimaciones de Monte Carlo tipo 1 y 2 utilizando números pseudoaleatorios para el área bajo la curva $y = x^2$ en $[0, 1]$.

Esto es una extensión del Monte Carlo tipo 1 del ejemplo 9.1, donde se prestó atención al error como una función del número n de puntos aleatorios. Para cada intento, se generan n números aleatorios uniformes x en $[0, 1]$, y se encuentra el valor promedio de $y = x^2$. El error es el valor absoluto de la diferencia entre el valor medio y la respuesta correcta 1/3. Se promedia el error de más de 500 intentos para cada n y se grafican los resultados, con lo que se obtiene la curva inferior de la figura 9.4.

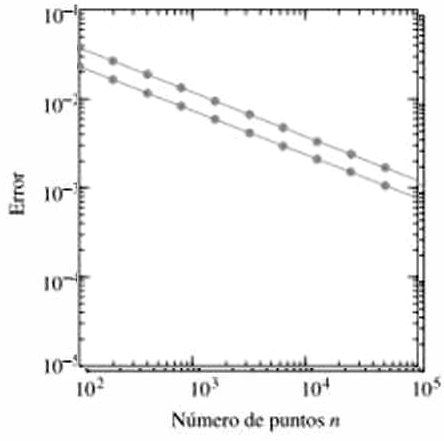


Figura 9.4 Error medio de la estimación de Monte Carlo. Error de estimación en el ejemplo 9.4, como problemas de Monte Carlo tipo 1 (curva inferior) y tipo 2 (curva superior) cuando se usan números pseudoaleatorios. La dependencia de la ley de potencia tiene exponente −1/2 para los dos tipos de problemas.

Para Monte Carlo tipo 2, se generan pares aleatorios uniformes (x, y) en el cuadrado unitario $[0, 1] \times [0, 1]$ y se sigue la proporción que satisface $y < x^2$. Una vez más, el error se promedia para

500 intentos y se representa como la curva superior de la figura 9.4. Aunque el error de tipo 2 es ligeramente mayor que el error de tipo 1, ambos siguen la ley de la potencia de la raíz cuadrada (9.9). ◄

¿La aleatoriedad de las muestras es en verdad necesaria para un problema de Monte Carlo del tipo 2?, ¿por qué no utilizar una malla rectangular y regular de muestras, en lugar de números aleatorios, para resolver un problema como el ejemplo 9.2? Por supuesto, se perdería la capacidad de detenerse después de un número arbitrario de n muestras, a menos que hubiera algún modo del tipo aleatorio para ordenarlos, a fin de evitar un sesgo muy importante en la estimación. Resulta que hay una opción media, que mantiene las ventajas de la malla regular pero ordena los números para que parezcan aleatorios. Éste es el tema de la siguiente sección.

9.2.2 Números cuasialeatorios

La idea de los números cuasialeatorios es sacrificar la propiedad de independencia de los números aleatorios cuando en realidad no es esencial para el problema que debe resolverse. El sacrificio de la independencia implica que los números cuasialeatorios no sólo son no aleatorios, sino que a diferencia de los números pseudoaleatorios, no pretenden ser aleatorios. Este sacrificio se hace con la esperanza de una convergencia más rápida hacia el valor correcto en un entorno de Monte Carlo. Las secuencias de números cuasialeatorios están diseñadas para ser autoexcluyentes en vez de independientes. Es decir, el flujo de números trata de llenar de manera eficiente los huecos dejados por los números anteriores y evitar las agrupaciones. La comparación con los números pseudoaleatorios se ilustra en la figura 9.5.

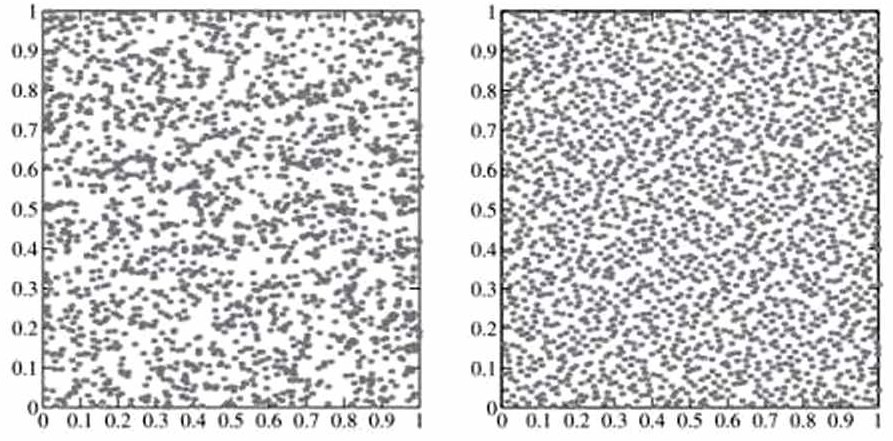


Figura 9.5 Comparativo de números pseudoaleatorios y cuasialeatorios. (a) 2000 pares de números pseudoaleatorios, producidos mediante `rand` de MATLAB. (b) 2000 pares de números cuasialeatorios, producidos mediante secuencias de baja-discrepancias de Halton, con base 2 en la coordenada x y base 3 en la coordenada y.

Existen muchas maneras de producir números cuasialeatorios. Tal vez la forma más popular se remonta a una sugerencia de Van der Corput en 1935, llamada **secuencia de baja-discrepancia con base-p**. Se proporciona la implementación realizada por Halton [1960]. Sea p un número primo, por ejemplo, $p = 2$. Escriba los primeros n enteros en base aritmética p. Suponiendo que el i-ésimo entero tiene una representación $b_k b_{k-1} \ldots b_2 b_1$, se asignará el i-ésimo número aleatorio como $0.b_1 b_2 \cdots b_{k-1} b_k$, de nuevo escrito en la base aritmética p. En otras palabras, escriba el en-

tero i-ésimo en base p, en seguida invierta los dígitos y póngalos después del punto decimal para obtener el i-ésimo número aleatorio uniforme en [0, 1]. Al establecer $p = 2$, se obtiene la siguiente lista de los primeros ocho números aleatorios:

i	$(i)_2$	$(u_i)_2$	u_i
1	1	.1	0.5
2	10	.01	0.25
3	11	.11	0.75
4	100	.001	0.125
5	101	.101	0.625
6	110	.011	0.375
7	111	.111	0.875
8	1000	.0001	0.0625

Si se establece $p = 3$ se obtiene la secuencia de Halton con base 3:

i	$(i)_3$	$(u_i)_3$	u_i
1	1	.1	$0.\overline{3}$
2	2	.2	$0.\overline{6}$
3	10	.01	$0.\overline{1}$
4	11	.11	$0.\overline{4}$
5	12	.21	$0.\overline{7}$
6	20	.02	$0.\overline{2}$
7	21	.12	$0.\overline{5}$
8	22	.22	$0.\overline{8}$

A continuación se muestra el código de MATLAB para la secuencia de Halton. Es una versión sencilla y directa de la idea original de baja-discrepancia. Para una mayor eficacia, puede codificarse al nivel de bits.

```
% Programa 9.1 Generador de números cuasialeatorios
% Secuencia de Halton con base p,
% Entrada: número primo p, números aleatorios requeridos n
% Salida: Arreglo u de números cuasialeatorios en [0,1]
% Uso de ejemplo: halton(2,100)
function u=halton(p,n)
b=zeros(ceil(log(n)/log(p)),1);    % mayor número de dígitos
for j=1:n
  i=1;
  b(1)=b(1)+1;                     % suma uno al entero actual
  while b(i)>p-1+eps               % este ciclo se hace
    b(i)=0;                        %    en base p
    i=i+1;
    b(i)=b(i)+1;
  end
  u(j)=0;
  for k=1:length(b(:))             % agrega dígitos invertidos
    u(j)=u(j)+b(k)*p^(-k);
  end
end
```

Para cualquier número primo, la secuencia de Halton dará un conjunto de número cuasialeatorios. Para generar una secuencia de vectores d-dimensionales, puede utilizarse un número primo diferente para cada coordenada. Es importante recordar que los números cuasialeatorios no son independientes, su utilidad radica en su propiedad de autoexcluyentes. Para los problemas de Monte Carlo, son mucho más eficientes que los números pseudoaleatorios, como se verá a continuación.

La razón para el uso de números cuasialeatorios es que dan lugar a una convergencia más rápida en las estimaciones de Monte Carlo. Eso significa que como una función de n, el número de evaluaciones de la función, el error disminuye a una tasa proporcional a una potencia negativa más grande de n que la correspondiente tasa de los números pseudoaleatorios. Las siguientes fórmulas de error deben compararse con las fórmulas correspondientes (9.9) para los números pseudoaleatorios (con d se indica la dimensión de los números aleatorios generados):

Monte Carlo tipo 1 con números cuasialeatorios

$$\text{Error} \propto (\ln n)^d n^{-1} \tag{9.10}$$

Monte Carlo tipo 2 con números cuasialeatorios

$$\text{Error} \propto n^{-\frac{1}{2}-\frac{1}{2d}} \tag{9.11}$$

El error está dominado por lo que sucede en las discontinuidades. En lugar de una demostración, se describe lo que ocurre en el caso de los ejemplos tipo 2 que se han encontrado, donde la función es característica de un subconjunto en el espacio d-dimensional que tiene una frontera $(d-1)$-dimensional. En este caso, el número de puntos de discontinuidad, a lo largo de la frontera del conjunto, es proporcional a $(n^{1/d})^{d-1}$. Esto se deduce del hecho de que la frontera es $(d-1)$-dimensional, y hay alrededor de $n^{1/d}$ puntos de malla a lo largo de cada una de las d dimensiones. Estos puntos asumen "aleatoriamente" los valores 0 o 1, dependiendo de qué lado de la frontera se encuentren. Como los errores en todos los otros puntos son mucho más pequeños, la varianza de la evaluación de la función es, en promedio,

$$\frac{n^{\frac{d-1}{d}}}{n} = n^{-\frac{1}{d}},$$

y la desviación estándar es la raíz cuadrada $n^{-\frac{1}{2d}}$. Por el mismo argumento que en el caso pseudoaleatorio de Monte Carlo, donde se promedian n puntos, la desviación estándar se reduce en un factor de $\sqrt{n}$, lo que deja a la desviación estándar del método cuasi-Monte Carlo como

$$\frac{n^{-1/2d}}{n^{1/2}} = n^{-\frac{1}{2}-\frac{1}{2d}}.$$

► EJEMPLO 9.5 Encuentre una estimación de Monte Carlo usando cuasialeatorios para el área bajo la curva de $y = x^2$ en $[0, 1]$.

Éste es un problema de Monte Carlo tipo 1, donde pueden generarse coordenadas x en $[0, 1]$ para encontrar el promedio de $f(x) = x^2$ como una aproximación del área. Se usa la secuencia de Halton con el número primo $p = 2$ a fin de generar 10^5 números cuasialeatorios. Los resultados, comparados con la misma estrategia utilizada para los números pseudoaleatorios, se muestran en la figura 9.6. Los números cuasialeatorios son claramente superiores, como se predijo antes. ◄

► EJEMPLO 9.6 Encuentre una estimación de Monte Carlo cuasialeatoria para el área del ejemplo 9.2.

Para diferentes n, se generaron muestras cuasialeatorias en el cuadrado unitario mediante la secuencia de Halton. Para aplicaciones multidimensionales, resulta conveniente utilizar secuencias de Halton con diferentes números primos p para cada coordenada. El área es un subconjunto de un espacio bidimensional con una frontera unidimensional, por lo que $d = 2$. Se determinó la

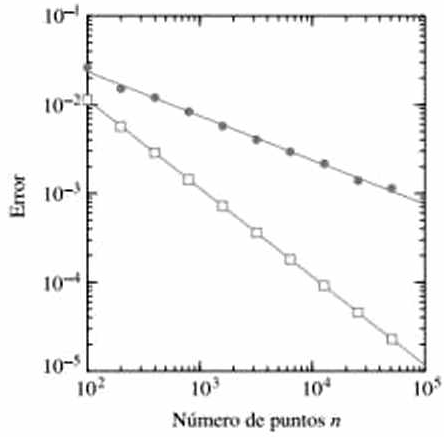


Figura 9.6 Error medio de la estimación de Monte Carlo tipo 1. Estimación de la integral del ejemplo 9.1. Los círculos representan el error cuando se usan números pseudoaleatorios, los cuadrados corresponden al error de los cuasialeatorios. Observe la dependencia de la ley de potencia con exponente $-1/2$ y -1, respectivamente, para los números pseudo y cuasialeatorios.

proporción que satisface la condición definida en el ejemplo 9.2, y se calculó el error. El error se promedió en 50 intentos y se representa en la figura 9.7(a). El exponente de la ley de potencia para un problema de Monte Carlo tipo 2 en 2 dimensiones es $-1/2 - 1/(2d) = -1/2 - 1/4 = -3/4$, que es la pendiente aproximada de la curva inferior. Con fines comparativos, en la figura se muestra el mismo cálculo para los números pseudoaleatorios, con una ley de potencia de raíz cuadrada. ◄

► **EJEMPLO 9.7** Encuentre una estimación de Monte Carlo cuasialeatoria para el volumen de la pelota tridimensional de radio uno en R^3.

Se procede de manera similar a la del ejemplo 9.6. Como el problema de tipo 2 ocurre en tres dimensiones, el exponente de la ley de potencia es $-1/2 - 1/6 = -2/3$, que es aproximadamente la pendiente de la curva inferior de la figura 9.7(b). ◄

9.2 Problemas de computadora

1. Realice la aproximación de Monte Carlo del problema de computadora 3 de la sección 9.1 con $n = 10^k$ números cuasialeatorios de la secuencia de Halton para $k = 2, 3, 4$ y 5. En el inciso (c), use `halton(2,n)` y `halton(3,n)` para las coordenadas x y y, respectivamente.
2. Realice la aproximación de Monte Carlo del problema de computadora 4 de la sección 9.1 con números cuasi-aleatorios.
3. Realice la aproximación de Monte Carlo del problema de computadora 5 de la sección 9.1 con $n = 10^4$ y $n = 10^5$ puntos cuasialeatorios.
4. Realice la aproximación de Monte Carlo del problema de computadora 6 de la sección 9.1 con $n = 10^4$ y $n = 10^5$ puntos cuasialeatorios.
5. Calcule las aproximaciones de Monte Carlo y cuasi-Monte Carlo al volumen de la pelota en cuatro dimensiones de radio 1 con $n = 10^5$ puntos. Compare su respuesta con el volumen exacto $\pi^2/2$.
6. Uno de los problemas más conocidos de Monte Carlo es la aguja de Buffon. Si una aguja se deja caer en un piso pintado con franjas blancas y negras, cada una con la misma anchura que la longitud de la aguja, entonces existe una probabilidad de $2/\pi$ de que la aguja caiga sobre ambos

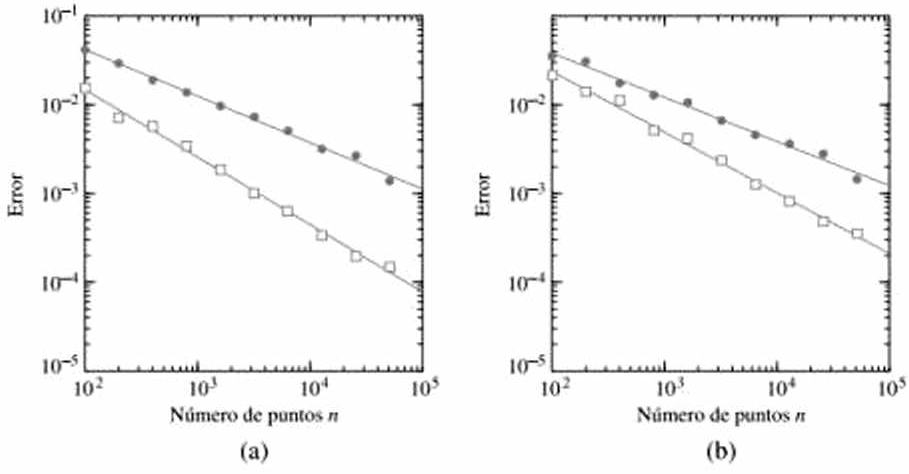


Figura 9.7 Error medio de la estimación de Monte Carlo tipo 2. Los círculos representan el error cuando se utilizan números pseudoaleatorios, los cuadros corresponden al error de los cuasialeatorios. (a) Estimación del área del ejemplo 9.2, un problema de Monte Carlo tipo 2 en la dimensión $d = 2$. Los errores siguen las leyes de potencia con exponentes $-1/2$ y $-3/4$, respectivamente, para los números pseudo- y cuasialeatorios. (b) Estimación del volumen de la pelota tridimensional de diámetro 1, un problema de Monte Carlo tipo 2 en la dimensión $d = 3$. Los errores siguen las leyes de potencia con exponentes $-1/2$ y $-2/3$.

colores. (a) Demuestre de manera analítica este resultado. Considere la distancia d del punto medio de la aguja al borde más cercano, y su ángulo θ respecto a las franjas. Exprese la probabilidad como una integral simple. (b) Diseñe una simulación de Monte Carlo tipo 2 que se aproxime a la probabilidad, y realícela con $n = 10^6$ pares pseudoaleatorios (d, θ).

7. (a) ¿Qué proporción de las matrices de 2×2 con entradas en el intervalo $[0, 1]$ tienen determinante positivo? Encuentre el valor exacto y aproximado con una simulación de Monte Carlo. (b) ¿Qué proporción de matrices simétricas de 2×2 con entradas en $[0, 1]$ tienen determinante positivo? Encuentre el valor exacto y aproximado con una simulación de Monte Carlo.

8. Ejecute una simulación de Monte Carlo para aproximar la proporción de las matrices de 2×2 con entradas en $[-1, 1]$, cuyos valores propios son reales.

9. ¿Qué proporción de las matrices de 4×4 con entradas en $[0, 1]$ no implican intercambios entre renglones al ser sometidas al pivoteo parcial? Utilice una simulación de Monte Carlo que involucre el comando `lu` de MATLAB para estimar esta probabilidad.

9.3 MOVIMIENTO BROWNIANO DISCRETO Y CONTINUO

Aunque los capítulos anteriores de este libro se han centrado primordialmente en los principios que son importantes para las matemáticas de los modelos deterministas, estos modelos son sólo una parte del arsenal de las técnicas modernas. Una de las aplicaciones más importantes de los números aleatorios consiste en hacer posible el modelado estocástico.

Se iniciará con uno de los modelos estocásticos más sencillos, la caminata aleatoria, también llamada movimiento browniano discreto. Los principios básicos detrás de este modelo discreto son en esencia los mismos que para los modelos más sofisticados que se presentarán después, basados en el movimiento browniano continuo.

9.3.1 Caminatas aleatorias

Una **caminata aleatoria** W_t se define en la recta real comenzando en $W_0 = 0$ y avanzando un paso de longitud s_i en cada unidad de tiempo entera i, donde las s_i son variables aleatorias independientes e idénticamente distribuidas. Aquí se asumirá que cada s_i es $+1$ o -1 con la misma probabilidad igual a 1/2. El **movimiento browniano discreto** se define como la caminata aleatoria dada por la secuencia de pasos acumulados

$$W_t = W_0 + s_1 + s_2 + \cdots + s_t,$$

para $t = 0, 1, 2, \ldots$ En la figura 9.8 se ilustra uno de los desarrollos del movimiento browniano discreto.

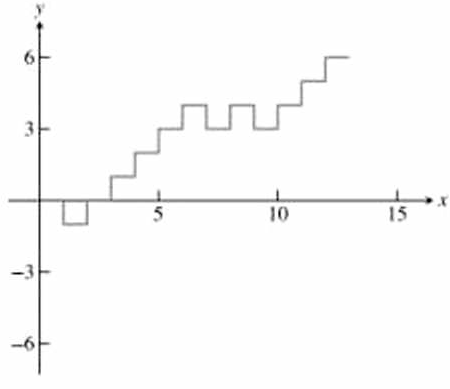


Figura 9.8 Uno de los desarrollos de una caminata aleatoria. La ruta toca la frontera del intervalo (vertical) $[-3, 6]$ en el paso 12. En promedio, las caminatas aleatorias escapan por la parte superior de este intervalo una tercera parte de las veces, en promedio.

El siguiente código de MATLAB realiza una caminata aleatoria de 10 pasos:

```
t=10;
w=0;
for i=1:t
  if rand>1/2
    w=w+1;
  else
    w=w-1;
  end
end
```

Dado que una caminata aleatoria es un dispositivo probabilístico, deberán utilizarse algunos conceptos de probabilidad elemental. Para cada t, el valor de W_t es una variable aleatoria. El encadenamiento de una serie de variables aleatorias $\{W_0, W_1, W_2, \ldots\}$ es, por definición, un **proceso estocástico**.

El valor esperado de un solo paso s_i de la caminata aleatoria W_t, es $(0.5)(1) + (0.5) \times (-1) = 0$, y la varianza de s_i es $E[(s_i - 0)^2] = (0.5)(1)^2 + (0.5)(-1)^2 = 1$. El valor esperado de la caminata aleatoria después de un número entero t de pasos es $E(W_t) = E(s_1 + \cdots + s_t) = E(s_1) + \cdots + E(s_t) = 0$, y la varianza es $V(W_t) = V(s_1 + \cdots + s_t) = V(s_1) + \cdots + V(s_t) = t$, porque la varianza es aditiva sobre las variables aleatorias independientes.

La media y la varianza son cantidades estadísticas que resumen la información acerca de una distribución de probabilidad. El hecho de que la media de W_t sea 0 y su varianza sea t indica que al calcular n diferentes desarrollos de la variable aleatoria W_t, la

$$\textbf{media muestral} = E_{\text{muestral}}(W_t) = \frac{W_t^1 + \cdots + W_t^n}{n}$$

y la

$$\textbf{varianza muestral} = V_{\text{muestral}}(W_t) = \frac{(W_t^1 - E_s)^2 + \cdots + (W_t^n - E_s)^2}{n-1}$$

deberían aproximarse a 0 y t, respectivamente. La **desviación estándar muestral**, que se define como la raíz cuadrada de la varianza muestral, también se conoce como el **error estándar** de la media.

Muchas aplicaciones interesantes de las caminatas aleatorias se basan en los tiempos de escape, también llamados primeros tiempos de paso. Sean a y b enteros positivos, y considere la primera vez que la caminata aleatoria que inicia en 0 alcanza la frontera del intervalo $[-b, a]$. Esto se denomina **tiempo de escape** de la caminata aleatoria. Puede demostrarse (Steele [2001]) que la probabilidad de que el escape ocurra en a (en vez de en $-b$) es exactamente $b/(a+b)$.

► EJEMPLO 9.8 Utilice una simulación de Monte Carlo para aproximar la probabilidad de que la caminata aleatoria salga del intervalo $[-3, 6]$ a través de la frontera superior 6.

Esto debe suceder 1/3 de las veces. Se calculará la media muestral y el error de la probabilidad de escape a través de $a = 6$ como un problema de Monte Carlo tipo 2. Se ejecutan n caminatas aleatorias hasta el escape, y se registra la proporción que llega a 6 antes de -3. Para distintos valores de n, se encuentra la siguiente tabla:

n	salidas superiores	prob	error
100	35	0.3500	0.0167
200	72	0.3600	0.0267
400	135	0.3375	0.0042
800	258	0.3225	0.0108
1600	534	0.3306	0.0027
3200	1096	0.3425	0.0092
6400	2213	0.3458	0.0124

El error es el valor absoluto de la diferencia entre la estimación y la probabilidad correcta de 1/3. El error disminuye poco a poco a medida que se utilizan más caminatas aleatorias, pero de manera irregular, como lo muestra la tabla. En la figura 9.9 se presenta el error promedio de más de 50 intentos. Con este promedio, los errores muestran que la ley de potencia de la raíz cuadrada disminuye, como es característico de la simulación de Monte Carlo. ◄

La longitud esperada del tiempo de escape de $[-b, a]$ se conoce (Steele [2001]) como ab. La misma simulación puede usarse para investigar la eficacia de Monte Carlo en este problema.

► EJEMPLO 9.9 Utilice una simulación de Monte Carlo para estimar el tiempo de escape en el que una caminata aleatoria sale del intervalo $[-3, 6]$.

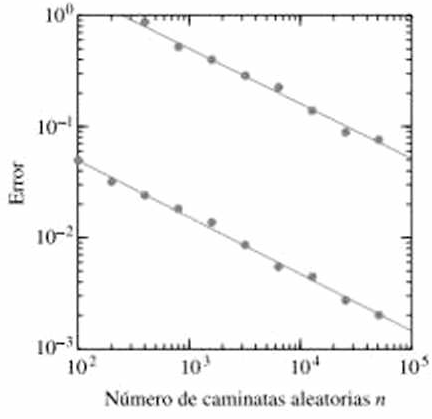


Figura 9.9 Error de la estimación de Monte Carlo para el problema de escape. En la curva inferior se muestra el error de estimación contra el número de caminatas aleatorias para la probabilidad de escapar de $[-3, 6]$ alcanzando 6. El valor esperado de la probabilidad es 1/3. La curva superior muestra el error de estimación para el tiempo de escape del mismo problema. El valor esperado es de 18 pasos de tiempo. Los errores se promediaron para 50 intentos.

El valor esperado del tiempo de escape es $ab = 18$. Un cálculo de muestra resulta en la tabla siguiente:

n	tiempo promedio de escape	error
100	18.84	0.84
200	17.47	0.53
400	19.64	1.64
800	18.53	0.53
1600	18.27	0.27
3200	18.16	0.16
6400	18.05	0.05

Una vez más, el error disminuye gradualmente a una tasa irregular. Para ver la ley de potencia de la raíz cuadrada para el error, deben promediarse varios intentos para cada n. En la figura 9.9 se muestra el resultado de 50 intentos. ◀

9.3.2 Movimiento browniano continuo

En la sección anterior se encontró que para la caminata aleatoria estándar en t pasos de tiempo se espera el valor 0 y la varianza t. Imagine ahora que se realiza el doble de pasos por unidad de tiempo. Si se da un paso cada 1/2 unidad de tiempo, el valor esperado de la caminata aleatoria en el tiempo t sigue siendo 0, pero la varianza cambia a

$$V(W_t) = V(s_1 + \cdots + s_{2t}) = V(s_1) + \cdots + V(s_{2t}) = 2t,$$

puesto que se han realizado $2t$ pasos. Con el fin de representar la incertidumbre en un modelo continuo, por ejemplo en una ecuación diferencial, se requiere una versión continua de la caminata aleatoria. La duplicación del número de pasos por unidad de tiempo es un buen punto de partida, pero para mantener la varianza fija mientras se aumenta el número de pasos, será necesario reducir el tamaño (vertical) de cada paso. Al aumentar el número de pasos por un factor k, debe cambiarse la altura de los pasos en un factor de $1/\sqrt{k}$ para mantener la varianza en el mismo valor previo. Esto se debe a que la multiplicación de una variable aleatoria por una constante cambia la varianza por un factor igual al cuadrado de la constante.

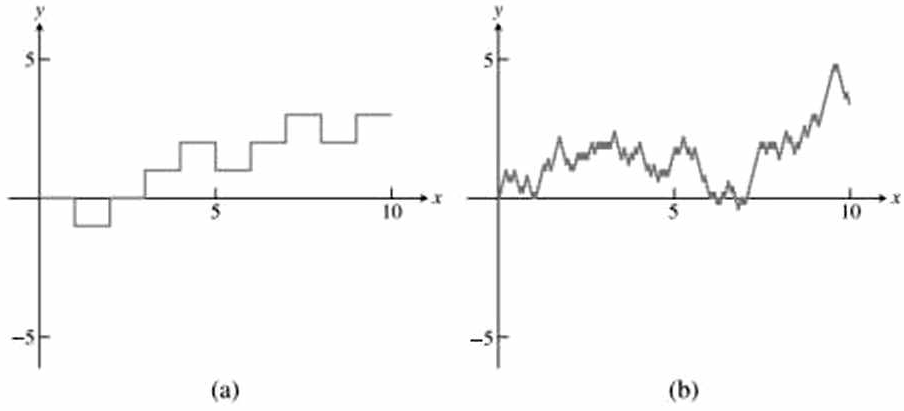


Figura 9.10 Movimiento browniano discreto. (a) Caminata aleatoria W_t de 10 pasos. (b) Caminata aleatoria W_t^{25} usando 25 veces más pasos que (a), pero con una altura de paso de $1/\sqrt{25}$. La media y la varianza de la altura en el momento $t = 10$ son idénticas (0 y 10, respectivamente) para los procesos (a) y (b).

Por lo tanto, W_t^k se define como la caminata aleatoria que realiza pasos s_i^k de longitud horizontal $1/k$, una altura de paso $\pm 1/\sqrt{k}$ y la misma probabilidad. Entonces, el valor esperado en el tiempo t sigue siendo

$$E(W_t^k) = \sum_{i=1}^{kt} E(s_i^k) = \sum_{i=1}^{kt} 0 = 0,$$

y la varianza es

$$V(W_t^k) = \sum_{i=1}^{kt} V(s_i^k) = \sum_{i=1}^{kt} \left[\left(\frac{1}{\sqrt{k}}\right)^2 (.5) + \left(-\frac{1}{\sqrt{k}}\right)^2 (.5) \right] = kt\frac{1}{k} = t. \qquad (9.12)$$

Si se reduce el tamaño de paso y la altura de paso de la caminata aleatoria de esta manera precisa a medida que k aumenta, la varianza y la desviación estándar se mantienen constantes, independientemente del número k de pasos por unidad de tiempo. En la figura 9.10(b) se muestra un desarrollo de W_t^k, donde $k = 25$, de modo que se realizan 250 pasos individuales en 10 unidades de tiempo. La media y la varianza en $t = 10$ son las mismas que en la figura 9.10(a).

El límite W_t^∞ de esta progresión cuando $k \to \infty$ produce un **movimiento browniano continuo**. Ahora el tiempo t es un número real y $B_t \equiv W_t^\infty$ es una variable aleatoria para cada $t \geq 0$. El movimiento browniano continuo B_t tiene tres propiedades importantes:

Propiedad 1 Para cada t, la variable aleatoria B_t está por lo regular distribuida con media 0 y varianza t.

Propiedad 2 Para cada $t_1 < t_2$, la variable aleatoria normal $B_{t_2} - B_{t_1}$ es independiente de la variable aleatoria B_{t_1}, y de hecho es independiente de toda B_s, $0 \leq s \leq t_1$.

Propiedad 3 El movimiento browniano B_t puede representarse mediante trayectorias continuas.

La aparición de la distribución normal es una consecuencia del teorema del límite central, un hecho profundo acerca de la probabilidad.

La simulación por computadora del movimiento browniano se basa en el respeto a estas tres propiedades. Establezca una malla de pasos

$$0 = t_0 \leq t_1 \leq \cdots \leq t_n$$

en el eje t, e inicie con $B_0 = 0$. La propiedad 2 dice que el incremento $B_{t_1} - B_{t_0}$ es una variable aleatoria normal, y su media y su varianza son 0 y t_1. Por lo tanto, puede hacerse un desarrollo de la variable aleatoria B_{t_1} eligiéndolo de la distribución normal $N(0, t_1) = \sqrt{t_1 - t_0}N(0, 1)$; es decir, al multiplicar un número aleatorio normal estándar por $\sqrt{t_1 - t_0}$. Para encontrar B_{t_2}, se procede de manera similar. La distribución de $B_{t_2} - B_{t_1}$ es $N(0, t_2 - t_1) = \sqrt{t_2 - t_1}N(0, 1)$, por lo que se elige un número aleatorio normal estándar, se multiplica por $\sqrt{t_2 - t_1}$, y se suma a B_{t_1} para obtener B_{t_2}. En general, el incremento del movimiento browniano es la raíz cuadrada del paso de tiempo multiplicado por un número aleatorio normal estándar.

En MATLAB, puede escribirse una aproximación al movimiento browniano utilizando el generador integrado de números aleatorios normales `randn`. Aquí se usa el tamaño de paso $\Delta t = 1/25$), como en la figura 9.10(b).

```
k=250;
sqdelt=sqrt(1/25);
b=0;
for i=1:k
  b=b+sqdelt*randn;
end
```

Las estadísticas del tiempo de escape para el movimiento browniano continuo son idénticas a las de las caminatas aleatorias. Sean a y b números positivos (no necesariamente enteros), y considere la primera vez que el movimiento browniano continuo que inicia en 0 alcance la frontera del intervalo $[-b, a]$. Esto se denomina el tiempo de escape del intervalo para el movimiento browniano. Puede demostrarse que la probabilidad de que el escape ocurra en a (en vez de en b) es exactamente $b/(a + b)$. Por otra parte, el valor esperado del tiempo de escape es ab. En el problema de computadora 5 se pide al lector que ilustre estos hechos con simulaciones de Monte Carlo.

9.3 Problemas de computadora

1. Diseñe una simulación de Monte Carlo para estimar la probabilidad de que una caminata aleatoria llegue al límite superior de un intervalo dado $[-b, a]$. Realice $n = 10000$ caminatas aleatorias. Calcule el error comparando los resultados con la respuesta correcta. (a) $[-2, 5]$ (b) $[-5, 3]$ (c) $[-8, 3]$

2. Calcule el tiempo de escape medio para las caminatas aleatorias del problema de computadora 1. Realice $n = 10000$ caminatas aleatorias. Calcule el error comparando los resultados con la respuesta correcta.

3. En una **caminata aleatoria sesgada**, la probabilidad de subir una unidad es $0 < p < 1$, y la probabilidad de descender una unidad es $q = 1 - p$. Diseñe una simulación de Monte Carlo con $n = 10000$ para encontrar la probabilidad de que la caminata aleatoria sesgada con $p = 0.7$, en el intervalo del problema de computadora 1, llegue al límite superior. Calcule el error comparando los resultados con la respuesta correcta $[(q/p)^b - 1]/[(q/p)^{a+b} - 1]$ para $p \neq q$.

4. Resuelva el problema de computadora 3 para el tiempo de escape. El tiempo de escape medio para la caminata aleatoria sesgada con $p \neq q$ es $[b - (a + b)(1 - (q/p)^b)/(1 - (q/p)^{a+b})]/[q - p]$.

5. Diseñe una simulación de Monte Carlo para estimar la probabilidad de que el movimiento browniano escape a través del límite superior del intervalo dado $[-b, a]$. Utilice $n = 1000$ trayectorias de movimiento browniano con tamaño de paso $\Delta t = 0.01$. Calcule el error comparando los resultados con la respuesta correcta $b/(a + b)$. (a) $[-2, 5]$ (b) $[-2, \pi]$ (c) $[-8/3, 3]$.

6. Calcule el tiempo de escape medio del movimiento browniano para los intervalos del problema de computadora 5. Realice $n = 1000$ trayectorias de movimiento browniano con tamaño de paso $\Delta t = 0.01$. Calcule el error comparando los resultados con la respuesta correcta.

7. La **ley del arcoseno** del movimiento browniano sostiene que para $0 \leq t_1 \leq t_2$, la probabilidad de que una trayectoria no pase por cero en el intervalo de tiempo $[t_1, t_2]$ es $(2/\pi)$ arcsen $\sqrt{t_1/t_2}$. Realice una simulación de Monte Carlo de esta probabilidad usando 10 000 trayectorias con $\Delta t = 0.01$, y compárela con la probabilidad correcta, para los intervalos de tiempo: (a) $3 < t < 5$ (b) $2 < t < 10$ (c) $8 < t < 10$.

9.4 ECUACIONES DIFERENCIALES ESTOCÁSTICAS

Las ecuaciones diferenciales ordinarias son modelos deterministas. Dada una EDO y una condición inicial correspondiente, existe una solución única, lo que significa que la solución está determinada por completo. Tal sabiduría no siempre está disponible para el analista. En muchos sistemas, algunas partes pueden modelarse con facilidad, otras partes parecen moverse al azar (aparentemente de manera independiente del estado actual del sistema). En tales situaciones, en vez de abandonar la idea de un modelo, es común añadir un término de incertidumbre a la ecuación diferencial para representar los efectos aleatorios. El resultado se conoce como una ecuación diferencial estocástica (EDE).

En esta sección se analizan algunas ecuaciones diferenciales estocásticas elementales y se explica cómo encontrar soluciones aproximadas en forma numérica. Las soluciones serán procesos estocásticos continuos como el movimiento browniano. Se inicia con algunas definiciones necesarias y una breve introducción al cálculo de Ito. Para más información, el lector puede consultar Klebaner [1998], Oksendal [1998], y Steele [2001].

9.4.1 Incorporación de la incertidumbre a las ecuaciones diferenciales

Las soluciones a las ecuaciones diferenciales ordinarias son funciones. Por otra parte, las soluciones a las ecuaciones diferenciales estocásticas son procesos estocásticos.

DEFINICIÓN 9.2 Un conjunto de variables aleatorias x_t indexadas mediante números reales $t \geq 0$ se denomina un **proceso estocástico de tiempo continuo.** ❐

Cada instancia, o **desarrollo** del proceso estocástico es una elección de la variable aleatoria x_t para cada t, y por lo tanto es una función de t.

El movimiento browniano B_t es un proceso estocástico. Cualquier función (determinista) $f(t)$ también puede considerarse de manera trivial como un proceso estocástico, con varianza $V(f(t)) = 0$. La solución al problema del valor inicial de la EDE

$$\begin{cases} dy = r\,dt + \sigma\,dB_t \\ y(0) = 0 \end{cases}, \tag{9.13}$$

con constantes r y σ, es el proceso estocástico $y(t) = rt + \sigma B_t$, aunque es necesario definir algunos términos.

Observe que la EDE (9.13) se da en forma diferencial, en contraste con la forma derivada de una EDO. Esto se debe a que muchos de los procesos estocásticos interesantes, como el movimiento browniano, son continuos pero no diferenciables. Por lo tanto, el significado de la EDE

$$dy = f(t, y)\,dt + g(t, y)\,dB_t$$

es, por definición, la ecuación integral

$$y(t) = y(0) + \int_0^t f(s, y)\,ds + \int_0^t g(s, y)\,dB_s,$$

donde todavía debe definirse el significado de la última integral, denominado integral de Ito.

Sea $a = t_0 < t_1 < \cdots < t_{n-1} < t_n = b$ una malla de puntos en el intervalo $[a, b]$. La integral de Riemann se define como un límite

$$\int_a^b f(t)\,dt = \lim_{\Delta t \to 0} \sum_{i=1}^{n} f(t_i')\Delta t_i,$$

donde $\Delta t_i = t_i - t_{i-1}$ y $t_{i-1} \le t_i' \le t_i$. De manera similar, la **integral de Ito** es el límite

$$\int_a^b f(t)\,dB_t = \lim_{\Delta t \to 0} \sum_{i=1}^{n} f(t_{i-1})\Delta B_i,$$

donde $\Delta B_i = B_{t_i} - B_{t_{i-1}}$, un paso del movimiento browniano a través del intervalo. Mientras que la t_i' en la integral de Riemann puede elegirse en cualquier punto del intervalo (t_{i-1}, t_i), se requiere que el punto correspondiente de la integral de Ito sea el punto final izquierdo de ese intervalo.

Como f y B_t son variables aleatorias, también lo es la integral de Ito $I = \int_a^b f(t)\,dB_t$. La **diferencial** dI es una conveniencia de notación; así que,

$$I = \int_a^b f\,dB_t$$

es, por definición, equivalente a

$$dI = f\,dB_t.$$

La diferencial dB_t del movimiento browniano B_t se denomina **ruido blanco**.

► EJEMPLO 9.10 Resuelva la ecuación diferencial estocástica $dy(t) = r\,dt + \sigma\,dB_t$ con la condición inicial $y(0) = y_0$.

Se supone que r y σ son números reales constantes. La ecuación diferencial ordinaria (determinista)

$$y'(t) = r \tag{9.14}$$

tiene solución $y(t) = y_0 + rt$, una línea recta como una función del tiempo t. Si r es positiva, la solución se mueve hacia arriba con pendiente constante, si r es negativa, la solución se mueve hacia abajo.

Al sumar el ruido blanco $\sigma\,dBt$ para un número real constante σ al lado derecho, se obtiene la ecuación diferencial estocástica

$$dy(t) = r\,dt + \sigma\,dB_t. \tag{9.15}$$

La integración de ambos lados da

$$y(t) - y(0) = \int_0^t dy = \int_0^t r\,ds + \int_0^t \sigma\,dB_s = rt + \sigma B_t.$$

Esto confirma que la solución es el proceso estocástico

$$y(t) = y_0 + rt + \sigma B_t, \tag{9.16}$$

una combinación del desvío (el término rt) y la difusión del movimiento browniano.

En la figura 9.11 se muestran dos soluciones de la EDE (9.15) junto con la única solución a la EDO (9.14). En sentido estricto, esta última también es una solución de (9.15), que representa

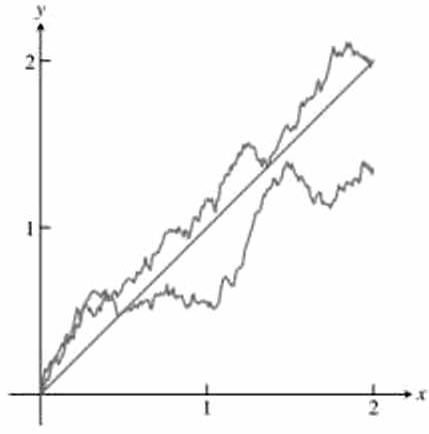


Figura 9.11 Soluciones al ejemplo 9.10. Se muestra una solución $y(t) = rt$ de la EDO $y'(t) = r$, junto con dos desarrollos diferentes del proceso de solución $y'(t) = rt + \sigma B(t)$ para (9.15). Los parámetros son $r = 1$ y $\sigma = 0.3$.

el desarrollo que acompaña a todas las entradas de ruido $z_i = 0$. Éste es un desarrollo particular posible, pero muy poco probable, del proceso estocástico de solución. ◄

Para resolver EDE en forma analítica, es necesario introducir la regla básica para la manipulación de diferenciales estocásticas, llamada fórmula de Ito.

Fórmula de Ito

Si $y = f(t, x)$, entonces

$$dy = \frac{\partial f}{\partial t}(t, x)\, dt + \frac{\partial f}{\partial x}(t, x)\, dx + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(t, x)\, dx\, dx, \tag{9.17}$$

donde el término $dx\, dx$ se interpreta usando las identidades $dt\, dt = 0$, $dt\, dB_t = dB_t\, dt = 0$, y $dB_t\, dB_t = d_t$.

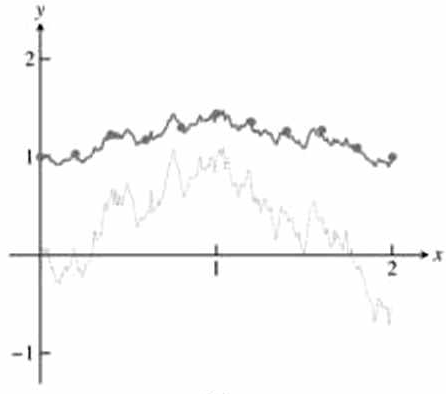
La fórmula de Ito es el análogo estocástico para la regla de la cadena del cálculo convencional. Aunque se expresa de forma diferencial para mayor facilidad de comprensión, su significado es nada más y nada menos que la igualdad de la integral de Ito a ambos lados de la ecuación. Lo anterior se demuestra haciendo referencia de nuevo a la ecuación de la definición de la integral de Ito (Oksendal [1998]).

► EJEMPLO 9.11 Demuestre que $y(t) = B_t^2$ es una solución de la EDE $dy = dt + 2B_t\, dB_t$.

Para utilizar la fórmula de Ito, escriba $y = f(t, x)$, donde $x = B_t$ y $f(t, x) = x^2$. De acuerdo con (9.17),

$$\begin{aligned} dy &= f_t\, dt + f_x\, dx + \frac{1}{2} f_{xx}\, dx\, dx \\ &= 0\, dt + 2x\, dx + \frac{1}{2} 2dx\, dx \\ &= 2B_t\, dB_t + dB_t\, dB_t \\ &= 2B_t\, dB_t + dt. \end{aligned}$$

◄



(a)

Figura 9.12 Solución al movimiento browniano exponencial de la EDE (9.19). La solución (9.18) se representa de manera gráfica como una curva continua junto con la aproximación de Euler-Maruyama, graficada en forma de círculos. La curva punteada es la trayectoria del movimiento browniano para el desarrollo correspondiente. Los parámetros se establecen como $r = 0.1$, $\sigma = 0.3$ y $\Delta t = 0.2$.

► **EJEMPLO 9.12** Demuestre que el movimiento browniano geométrico

$$y(t) = y_0 e^{(r-\frac{1}{2}\sigma^2)t+\sigma B_t} \tag{9.18}$$

satisface la ecuación diferencial estocástica

$$dy = ry\,dt + \sigma y\,dB_t. \tag{9.19}$$

Escriba $y = f(t, x) = y_0 e^x$, donde $x = (r - \frac{1}{2}\sigma^2)t + \sigma B_t$. Por la fórmula de Ito,

$$dy = y_0 e^x\,dx + \frac{1}{2} y_0 e^x\,dx\,dx,$$

donde $dx = (r - 1/2\sigma^2)\,dt + \sigma\,dB_t$. Usando las identidades diferenciales de la fórmula de Ito, se obtiene

$$dx\,dx = \sigma^2\,dt.$$

Por lo tanto,

$$\begin{aligned} dy &= y_0 e^x \left(r - \frac{1}{2}\sigma^2\right) dt + y_0 e^x \sigma\,dB_t + \frac{1}{2} y_0 \sigma^2 e^x\,dt \\ &= y_0 e^x r\,dt + y_0 e^x \sigma\,dB_t \\ &= ry\,dt + \sigma y\,dB_t. \end{aligned}$$

◄

En la figura 9.12 se muestra un desarrollo del movimiento geométrico browniano con **coeficiente de desviación** r constante y **coeficiente de difusión** σ. Este modelo se utiliza en gran medida en los modelos financieros. En particular, el movimiento browniano geométrico es el modelo esencial en las ecuaciones de Black-Scholes que se utilizan para establecer el precio de los derivados financieros.

Los ejemplos 9.11 y 9.12 son excepciones. Así como en el caso de las EDO, relativamente pocas EDE tienen soluciones cerradas. Con frecuencia es necesario utilizar técnicas numéricas de aproximación.

9.4.2 Métodos numéricos para EDE

Es posible aproximar una solución de una EDE de una manera similar al método de Euler del capítulo 6. El método de Euler-Maruyama funciona discretizando el eje del tiempo, al igual que lo hace Euler. Se define la trayectoria de la solución aproximada en una malla de puntos

$$a = t_0 < t_1 < t_2 < \cdots < t_n = b$$

y asignarán valores aproximados de y

$$w_0 < w_1 < w_2 < \cdots < w_n$$

en los puntos t respectivos. Dado el problema de valor inicial de la EDE

$$\begin{cases} dy(t) = f(t, y)dt + g(t, y)dB_t \\ y(a) = y_a \end{cases}, \tag{9.20}$$

se calcula la solución aproximadamente:

Método de Euler-Maruyama

$w_0 = y_0$
for $i = 0, 1, 2, \ldots$
$\quad w_{i+1} = w_i + f(t_i, w_i)(\Delta t_i) + g(t_i, w_i)(\Delta B_i)$ (9.21)
end

donde

$$\Delta t_i = t_{i+1} - t_i$$
$$\Delta B_i = B_{t_{i+1}} - B_{t_i}. \tag{9.22}$$

La parte crucial es cómo modelar el movimiento browniano ΔB_i. Defina $N(0, 1)$ para la variable aleatoria estándar que se distribuye por lo regular con media 0 y desviación estándar 1. Cada número aleatorio ΔB_i se calcula de acuerdo con la descripción de la sección 9.3.2 como

$$\Delta B_i = z_i \sqrt{\Delta t_i}, \tag{9.23}$$

donde z_i se elige de $N(0, 1)$. En MATLAB, la z_i puede generarse mediante el comando `randn`. Una vez más, observe la salida del caso determinístico de las EDO. Cada conjunto de $\{w_0, \ldots, w_n\}$ que se produce es un desarrollo aproximado del proceso estocástico de solución $y(t)$, el cual depende de los números aleatorios z_i que fueron seleccionados. Como B_t es un proceso estocástico, cada desarrollo será diferente, lo mismo que las aproximaciones generadas.

Como primer ejemplo, se muestra cómo aplicar el método de Euler-Maruyama al movimiento browniano exponencial de la EDE (9.19). El método de Euler-Maruyama tiene la forma

$$w_0 = y_0$$
$$w_{i+1} = w_i + r w_i(\Delta t_i) + \sigma w_i(\Delta B_i), \tag{9.24}$$

de acuerdo con (9.21). En la figura 9.12 se muestra un desarrollo correcto (generado a partir de la solución (9.18)) y la aproximación correspondiente de Euler-Maruyama. Por "correspondiente" se entiende que la aproximación utiliza el mismo desarrollo del movimiento browniano (también se muestra en la figura 9.12) como la solución correcta. Observe la concordancia cercana entre la solución correcta y los puntos de la aproximación, graficados como pequeños círculos cada 0.2 unidades de tiempo.

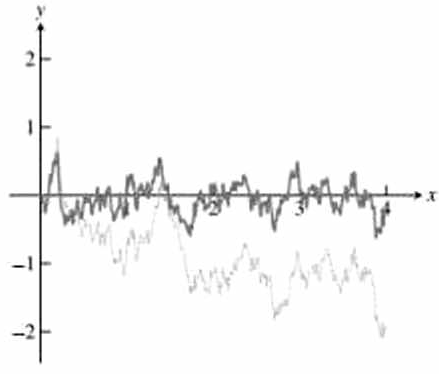


Figura 9.13 Solución a la ecuación de Langevin (9.25). La trayectoria superior es la aproximación a la solución para los parámetros $r = 10$, $\sigma = 1$, calculada mediante el método de Euler-Maruyama. La trayectoria punteada es el desarrollo correspondiente del movimiento browniano.

► EJEMPLO 9.13 Resuelva de manera numérica la **ecuación de Langevin**

$$dy = -ry\,dt + \sigma\,dB_t, \tag{9.25}$$

donde r y σ son constantes positivas.

Contrariamente a los ejemplos anteriores, no es posible obtener en forma analítica la solución a esta ecuación en términos de procesos simples. La solución de la ecuación de Langevin es un proceso estocástico llamado **proceso de Ornstein-Uhlenbeck.** En la figura 9.13 se muestra un desarrollo de la solución aproximada; ésta se generó a partir de una aproximación de Euler-Maruyama usando los pasos

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i - r w_i(\Delta t_i) + \sigma(\Delta B_i) \end{aligned} \tag{9.26}$$

para $i = 1, \ldots, n$.

Esta EDE se utiliza para modelar sistemas que tienden a volver a un estado particular. En este caso, el estado $y = 0$, en la presencia de incertidumbre. Puede pensarse en un recipiente que contiene una pelota de ping-pong, el cual se encuentra en un automóvil que conduce por un camino accidentado. La distancia de la pelota $y(t)$ desde el centro del recipiente puede modelarse mediante la ecuación de Langevin. ◄

A continuación se analiza el concepto del orden para los métodos de EDE. La idea es la misma que para los métodos de ODE, excepto por las diferencias causadas por el hecho de que la solución a una EDE es un proceso estocástico y que cada trayectoria calculada es sólo un desarrollo de ese proceso. Cada desarrollo del movimiento browniano forzará un desarrollo diferente de la solución $y(t)$. Si se establece un punto $T > 0$ en el eje t, cada trayectoria de solución iniciada en $t = 0$ proporciona un valor aleatorio en T —por lo tanto, $y(T)$ es una variable aleatoria. También, cada trayectoria de solución $w(t)$ calculada, utilizando Euler-Maruyama, por ejemplo, proporciona un valor aleatorio en T, de modo que $w(T)$ también es una variable aleatoria. Por todo lo anterior, la diferencia entre los valores en el tiempo T, $e(T) = y(T) - w(T)$, es una variable aleatoria. El concepto de orden cuantifica el valor esperado del error de una manera similar a la de los métodos de EDO.

DEFINICIÓN 9.3 Un método de EDE tiene un **orden** m si el valor esperado del error es de orden m-ésimo en el tamaño del paso; es decir, si para cualquier tiempo T, $E\{|y(T) - w(T)|\} = O((\Delta t)^m)$ como el tamaño del paso $\Delta t \to 0$. ❐

Es sorprendente que a diferencia del caso de las EDO donde el método de Euler es de orden 1, el método de Euler-Maruyama para EDE tiene orden $m = 1/2$. Para construir un método de orden 1 para EDE, debe agregarse otro término al método en la "serie estocástica de Taylor". Sean

$$\begin{cases} dy(t) = f(t, y)\,dt + g(t, y)\,dB_t \\ y(0) = y_0 \end{cases}$$

las EDE.

Método de Milstein

$w_0 = y_0$
for $i = 0, 1, 2, \ldots$

$$w_{i+1} = w_i + f(t_i, w_i)(\Delta t_i) + g(t_i, w_i)(\Delta B_i) + \tfrac{1}{2} g(t_i, w_i)\frac{\partial g}{\partial y}(t_i, w_i)((\Delta B_i)^2 - \Delta t_i) \tag{9.27}$$

end

El método de Milstein tiene orden 1. Observe que el método de Milstein es idéntico al método de Euler-Maruyama si no hay ningún término y en la parte de difusión $g(y, t)$ de la ecuación. En caso de que lo haya, Milstein convergirá al proceso estocástico de solución correcto con mayor rapidez que Euler-Maruyama cuando el tamaño de paso h tienda a cero.

► EJEMPLO 9.14 Aplique el método de Milstein al movimiento browniano geométrico.

La ecuación es

$$dy = ry\,dt + \sigma y\,dB_t \tag{9.28}$$

con el proceso de solución

$$y = y_0 e^{(r - \frac{1}{2}\sigma^2)t + \sigma B_t}. \tag{9.29}$$

La aproximación de Euler-Maruyama se analizó con anterioridad. Si se usa el tamaño de paso Δt, el método de Milstein se convierte en

$$w_0 = y_0$$
$$w_{i+1} = w_i + r w_i \Delta t + \sigma w_i \Delta B_i + \frac{1}{2}\sigma^2 w_i((\Delta B_i)^2 - \Delta t). \tag{9.30}$$

Al aplicar el método de Euler-Maruyama y el método de Milstein con tamaños de paso Δt decrecientes se obtienen aproximaciones sucesivamente mejoradas, como lo muestra la tabla siguiente:

Δt	Euler–Maruyama	Milstein
2^{-1}	0.169369	0.063864
2^{-2}	0.136665	0.035890
2^{-3}	0.086185	0.017960
2^{-4}	0.060615	0.008360
2^{-5}	0.048823	0.004158
2^{-6}	0.035690	0.002058
2^{-7}	0.024277	0.000981
2^{-8}	0.016399	0.000471
2^{-9}	0.011897	0.000242
2^{-10}	0.007913	0.000122

ANOTACIÓN

Convergencia Los órdenes de los métodos presentados aquí para las EDE, 1/2 para Euler-Maruyama y 1 para Milstein, se consideran bajos para los estándares de las EDO. Es posible desarrollar métodos de orden superior para las EDE, pero son mucho más complicados a medida que el orden crece. La necesidad de un método de orden superior en una aplicación dada depende de la forma en que las soluciones aproximadas resultantes van a utilizarse. En el caso de las EDO, el supuesto habitual es que la condición inicial y la ecuación se conocen con gran precisión. Entonces tiene sentido calcular la solución lo más cercanamente posible a la misma precisión, y se requieren métodos de orden superior sencillos. En muchas situaciones, las ventajas de los métodos de EDE de orden superior no son tan evidentes, y si vienen con un gasto computacional agregado, el uso de estos métodos no puede garantizarse.

Las dos columnas representan la media, en más de 100 desarrollos, del error $|w(T) - y(T)|$ en $T = 8$. Observe que los desarrollos de $w(t)$ y $y(t)$ comparten los mismos incrementos ΔB_i del movimiento browniano. Los órdenes de 1/2 para Euler-Maruyama y 1 para Milstein son claramente visibles en la tabla. Si se usa el método de Euler-Maruyama, es necesario reducir el tamaño de paso por un factor de 4 para disminuir el error en un factor de 2. Para el método de Milstein, se logra el mismo resultado al reducir el tamaño de paso por un factor de 2. En la figura 9.14, los datos de la tabla se representan de manera gráfica en una escala log-log. ◀

Una desventaja del método de Milstein es que en el método de aproximación aparece la derivada parcial, la cual debe ser proporcionada por el usuario. Esto es análogo a lo que sucede con los métodos de Taylor en la resolución de ecuaciones diferenciales ordinarias. Por esa razón, se desarrollaron los métodos de Runge-Kutta para EDO, que intercambian estas derivadas parciales adicionales en la expansión de Taylor por más evaluaciones de la función.

En el contexto de las EDE, puede hacerse el mismo intercambio con el método de Milstein, resultando en un método de primer orden que requiere la evaluación de $g(t, y)$ en dos lugares a cada paso. Es posible realizar una deducción heurística haciendo la sustitución

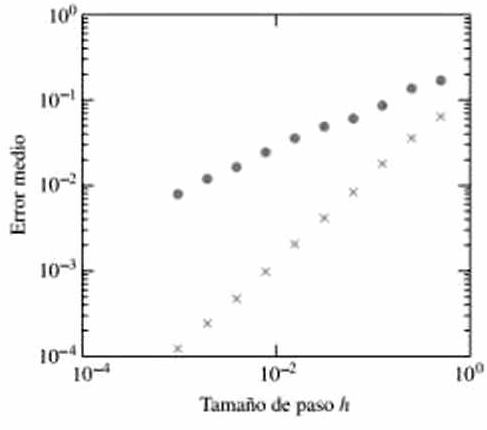


Figura 9.14 Error en los métodos de Euler-Maruyama y Milstein. Se calculan las trayectorias de solución para la ecuación de movimiento geométrico browniano (9.28) y se comparan con la respuesta correcta dada por (9.29). Se grafica la diferencia absoluta contra el tamaño de paso h para los dos métodos. Los errores de Euler-Maruyama se representan con círculos, y los de Milstein con cruces. Observe en la gráfica log-log que las pendientes son 1/2 y 1, respectivamente.

$$\frac{\partial g}{\partial y}(t_i, w_i) \approx \frac{g(t_i, w_i + g(t_i, w_i)\sqrt{\Delta t_i}) - g(t_i, w_i)}{g(t_i, w_i)\sqrt{\Delta t_i}}$$

en la fórmula Milstein, que conduce al siguiente método.

Método estocástico de primer orden de Runge-Kutta

$w_0 = y_0$
for $i = 0, 1, 2, \ldots$
$$w_{i+1} = w_i + f(t_i, w_i)\Delta t_i + g(t_i, w_i)\Delta B_i + \frac{1}{2\sqrt{\Delta t_i}}\left[g(t_i, w_i + g(t_i, w_i)\sqrt{\Delta t_i}) - g(t_i, w_i)\right]\left[(\Delta B_i)^2 - \Delta t_i\right]$$
end

► **EJEMPLO 9.15** Utilice el método de Euler-Maruyama, el método de Milstein y el método estocástico de primer orden de Runge-Kutta para resolver la EDE

$$dy = -2e^{-2y}\,dt + 2e^{-y}\,dB_t. \tag{9.31}$$

Este ejemplo tiene una propiedad precautoria interesante que vale la pena analizar. Es posible encontrar una solución explícita, pero sólo existe para un periodo de tiempo finito. Si se usa la fórmula de Ito (9.17), puede demostrarse que $y(t) = \ln(2B_t + e^{y_0})$ es una solución, siempre y cuando la cantidad dentro del logaritmo sea positiva. En el primer tiempo t cuando el desarrollo del movimiento browniano ocasiona que $2B_t + e^{y_0}$ sea negativa, la solución deja de existir.

El método de Euler-Maruyama para esta ecuación es

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i - 2e^{-2w_i}(\Delta t_i) + 2e^{-w_i}(\Delta B_i). \end{aligned}$$

El método de Milstein es

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i - 2e^{-2w_i}(\Delta t_i) + 2e^{-w_i}(\Delta B_i) - 2e^{-2w_i}\left[(\Delta B_i)^2 - \Delta t_i\right]. \end{aligned}$$

El método estocástico de primer orden de Runge-Kutta es

$$\begin{aligned} w_0 &= y_0 \\ w_{i+1} &= w_i - 2e^{-2w_i}(\Delta t_i) + 2e^{-w_i}(\Delta B_i) \\ &\quad + \frac{1}{2\sqrt{\Delta t_i}}\left[2e^{-(w_i + 2e^{-w_i}\sqrt{\Delta t_i})} - 2e^{-w_i}\right]\left[(\Delta B_i)^2 - \Delta t_i\right]. \end{aligned}$$

En la figura 9.15 se muestra una solución del método de Milstein en el intervalo $0 \le t \le 4$. ◄

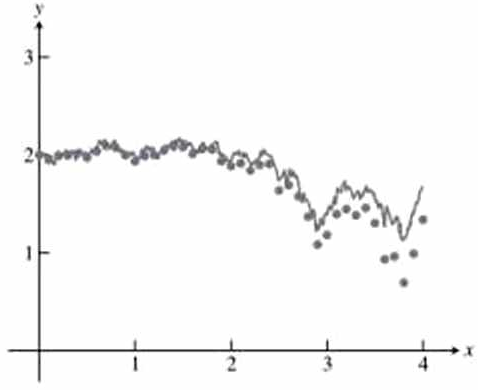


Figura 9.15 Solución a la ecuación (9.31). Se muestra la solución correcta junto con la aproximación de Milstein representada con círculos.

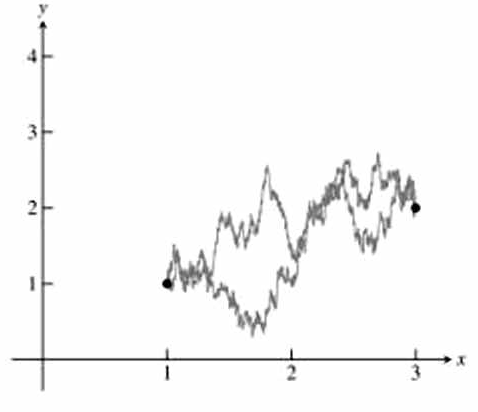
Los procesos estocásticos vistos hasta ahora han tenido variancias que aumentan con t. Por ejemplo, la varianza del movimiento browniano es $V(B_t) = t$. Se concluye esta sección con un ejemplo notable para el cual el final del desarrollo es tan predecible como el principio.

► EJEMPLO 9.16 Resuelva numéricamente el **puente browniano** de la EDE

$$\begin{cases} dy = \dfrac{y_1 - y}{t_1 - t}\,dt + dB_t \\ y(t_0) = y_0 \end{cases} \tag{9.32}$$

donde y_1 y $t_1 > t_0$ están dados.

La solución del puente browniano (9.32) se ilustra en la figura 9.16. Como la pendiente objetivo cambia de manera adaptativa a medida que se crea la trayectoria, todos los desarrollos del proceso de solución terminan en el punto deseado (t_1, y_1). Las trayectorias de solución pueden considerarse como "puentes" estocásticamente generados entre los dos puntos dados (t_0, y_0) y (t_1, y_1). ◄



(a)

Figura 9.16 Puente browniano. Dos métodos en la solución de (9.32). Los puntos considerados son $(t_0, y_0) = (1,1)$ y $(t_1, y_1) = (3, 2)$.

9.4 Ejercicios

1. Use la fórmula de Ito para demostrar que las soluciones a los problemas de valores iniciales de las EDE

$$\text{(a)} \begin{cases} dy = B_t\,dt + t\,dB_t \\ y(0) = c \end{cases} \qquad \text{(b)} \begin{cases} dy = 2B_t\,dB_t \\ y(0) = c \end{cases}$$

son (a) $y(t) = tB_t + c$ (b) $y(t) = B_t^2 - t + c$.

2. Use la fórmula de Ito para demostrar que las soluciones a los problemas de valores iniciales de las EDE

$$\text{(a)} \begin{cases} dy = (1 - B_t^2)e^{-2y}\,dt + 2B_t e^{-y}\,dB_t \\ y(0) = 0 \end{cases} \qquad \text{(b)} \begin{cases} dy = B_t\,dt + \sqrt[3]{9y^2}\,dB_t \\ y(0) = 0 \end{cases}$$

son (a) $y(t) = \ln(1 + B_t^2)$ (b) $y(t) = \frac{1}{3}B_t^3$.

3. Use la fórmula de Ito para demostrar que las soluciones a los problemas de valores iniciales de las EDE

$$\text{(a)} \begin{cases} dy = ty\,dt + e^{t^2/2}\,dB_t \\ y(0) = 1 \end{cases} \qquad \text{(b)} \begin{cases} dy = 3(B_t^2 - t)\,dB_t \\ y(0) = 0 \end{cases}$$

son (a) $y(t) = (1 + B_t)e^{t^2/2}$ (b) $y(t) = B_t^3 - 3tB_t$.

4. Use la fórmula de Ito para demostrar que las soluciones a los problemas de valores iniciales de las EDE

$$\text{(a)} \begin{cases} dy = -\frac{1}{2}y\,dt + \sqrt{1 - y^2}\,dB_t \\ y(0) = 0 \end{cases} \qquad \text{(b)} \begin{cases} dy = y(1 + 2\ln y)\,dt + 2yB_t\,dB_t \\ y(0) = 1 \end{cases}$$

son (a) $y(t) = \operatorname{sen} B_t$ y (b) $y(t) = e^{B_t^2}$.

5. Utilice la fórmula de Ito para demostrar que la solución de la ecuación (9.31) es $\ln(2B_t + e^{y_0})$.

6. (a) Resuelva la EDO análoga al puente browniano:

$$\begin{cases} y' = \dfrac{y_1 - y}{t_1 - t} \\ y(t_0) = y_0 \end{cases} \tag{9.33}$$

¿La solución alcanza el punto (t_1, y_1) cuando lo hace el puente browniano? Responda las mismas preguntas para las variantes

$$\text{(b)} \begin{cases} y' = \dfrac{y_1 - y_0}{t_1 - t_0} \\ y(t_0) = y_0 \end{cases} \qquad \text{(c)} \begin{cases} dy = \dfrac{y_1 - y_0}{t_1 - t_0}\,dt + dB_t \\ y(t_0) = y_0 \end{cases}$$

9.4 Problemas de computadora

1. Use el método de Euler-Maruyama para encontrar soluciones aproximadas a los problemas de valores iniciales de las EDE en el ejercicio 1. Use la condición inicial $y(0) = 0$. Grafique la solución correcta (que se encuentra al realizar un seguimiento del movimiento browniano B_t, con los mismos incrementos aleatorios) junto con la solución aproximada en el intervalo [0, 10], utilizando el tamaño de paso $h = 0.01$. Trace el error en el intervalo en una gráfica semilogarítmica.

2. Use el método de Euler-Maruyama para encontrar soluciones aproximadas a los problemas de valores iniciales de las EDE en el ejercicio 2. Use la condición inicial $y(0) = 0$. Grafique la solución correcta junto con la solución aproximada en el intervalo [0, 1], utilizando el tamaño de paso $h = 0.01$. Trace el error en el intervalo en una gráfica semilogarítmica.

3. Aplique el método de Euler-Maruyama con tamaño de paso $h = 0.01$ para aproximar las soluciones del ejercicio 3 en el intervalo [0, 2]. Grafique dos métodos del proceso estocástico de solución.

4. Aplique el método de Euler-Maruyama con tamaño de paso $h = 0.01$ para aproximar las soluciones del ejercicio 4 en el intervalo [0, 1]. Grafique dos desarrollos del proceso estocástico de solución.

5. Encuentre las soluciones aproximadas de Euler-Maruyama para

$$\begin{cases} dy = B_t\, dt + \sqrt[3]{9y^2}\, dB_t \\ y(0) = 0 \end{cases}$$

en el intervalo [0, 1] para los tamaños de paso $h = 0.1$, 0.01 y 0.001. Para cada tamaño de paso, ejecute 5000 desarrollos de la solución aproximada y encuentre el error promedio en $t = 1$. Haga una tabla con el error promedio en $t = 1$ en contraste con el tamaño de paso. ¿El error promedio se encuentra en una escala acorde con la teoría?

6. Utilice el método de Euler-Maruyama para resolver el problema de valores iniciales de la EDE $dy = y\, dt + y\, dB_t$, $y(0) = 1$. Grafique la solución aproximada y la solución correcta $y(t) = e^{\frac{1}{2}t + B_t}$. Use un tamaño de paso $h = 0.1$ en el intervalo $0 \le t \le 2$.

7. Utilice el método de Milstein para encontrar soluciones aproximadas al problema de valores iniciales de la EDE en el ejercicio 2(b). Grafique la solución correcta junto con la solución aproximada en el intervalo [0, 5]; use un tamaño de paso $h = 0.1$. Trace el error en el intervalo mediante una gráfica semilogarítmica.

8. Utilice el método de Milstein para encontrar soluciones aproximadas al problema de valores iniciales de la EDE en el ejercicio 4(a). Grafique la solución correcta junto con la solución aproximada en el intervalo [0, 5]; use un tamaño de paso $h = 0.1$. Trace el error en el intervalo mediante una gráfica semilogarítmica.

9. Utilice el método estocástico de primer orden de Runge-Kutta para encontrar soluciones aproximadas al problema de valores iniciales de la EDE en el ejercicio 2(b). Grafique la solución correcta junto con la solución aproximada en el intervalo [0, 5]; use un tamaño de paso $h = 0.1$. Trace el error en el intervalo mediante una gráfica semilogarítmica.

10. Utilice el método estocástico de primer orden de Runge-Kutta para encontrar soluciones aproximadas al problema de valores iniciales de la EDE en el ejercicio 4(a). Grafique la solución correcta junto con la solución aproximada en el intervalo [0, 5]; use un tamaño de paso $h = 0.1$. Trace el error en el intervalo mediante una gráfica semilogarítmica.

11. Encuentre las soluciones aproximadas de Milstein para

$$\begin{cases} dy = B_t\, dt + \sqrt[3]{9y^2}\, dB_t \\ y(0) = 0 \end{cases}.$$

en el intervalo [0, 1] con los tamaños de paso $h = 0.1$, 0.01 y 0.001. Para cada tamaño de paso, ejecute 5000 desarrollos de la solución aproximada y encuentre el error promedio en $t = 1$. Haga una tabla con el error promedio en $t = 1$ en contraste con el tamaño de paso. ¿El error promedio se encuentra en una escala acorde con la teoría?

12. Realice una estimación de Monte Carlo de $y(1)$, donde $y(t)$ es la solución de Euler-Maruyama de la ecuación de Langevin

$$\begin{cases} dy = -y dt + dB_t \\ y(0) = e \end{cases}.$$

Promedie $n = 1000$ desarrollos con tamaño de paso $h = 0.01$. Compare la respuesta con el valor esperado de $y(1)$, que es 1.

Comprobación en la realidad 9

La fórmula de Black-Scholes

La simulación de Monte Carlo y los modelos estocásticos de ecuaciones diferenciales son muy utilizados en los cálculos financieros. Un **derivado financiero** es un instrumento financiero cuyo valor se deriva del valor de otro instrumento. En particular, una **opción** es el derecho, pero no la obligación, de realizar una transacción financiera en particular.

Una opción **call** (llamada europea) es el derecho de comprar una acción de un valor a un precio preestablecido, llamado **precio del ejercicio**, en una fecha futura, llamada la **fecha del ejercicio.** Las opciones de compra son comúnmente compradas y vendidas por las empresas para administrar el riesgo, y por los individuos y los fondos de inversión como parte de las estrategias de inversión. El objetivo aquí es calcular el valor de la opción de compra.

Por ejemplo, una opción call de compra de 15 dólares en diciembre para ABC Corp. representa el derecho a comprar una acción a \$15 en diciembre. Suponga que el precio de ABC el 1 de junio es de \$12. ¿Cuál es el valor de ese derecho? En la fecha de ejercicio, el valor de una opción call de \$K está definido. Es máx$(X - K, 0)$, donde X es el precio actual de las acciones en el mercado. Esto es así porque, si $X > K$, el derecho de compra de ABC en \$K vale $\$X - K$; y si $X < K$, el derecho a comprar en K no tiene valor, ya que es posible comprar todo lo que se quiera a un precio aún más bajo. Mientras que el valor de una opción en la fecha del ejercicio es claro, la dificultad es la valoración de la opción de compra en algún momento antes del vencimiento.

En la década de 1960, Fisher Black y Myron Scholes exploraron la hipótesis del movimiento browniano geométrico,

$$dX = mX\,dt + \sigma X dB_t, \tag{9.34}$$

como el modelo de los valores, donde m es la desviación, o tasa de crecimiento, de las acciones y σ es la constante de difusión, o **volatilidad.** Tanto m como σ pueden estimarse a partir de los datos de los precios de las acciones en el pasado. La idea de Black y Scholes fue el desarrollo de una teoría de arbitraje que repite la opción a través del equilibrio juicioso de la tenencia de acciones y préstamos en efectivo al tipo de interés r prevaleciente. El resultado de su argumento fue que el valor de la opción de compra correcta, con fecha de vencimiento T años en el futuro, es el valor presente del valor esperado de la opción en el momento del vencimiento, donde el precio subyacente de la acción $X(t)$ satisface la EDE

$$dX = rX\,dt + \sigma X\,dB_t. \tag{9.35}$$

Es decir, para el precio de una acción $X = X_0$ en el tiempo $t = 0$, el valor de la opción de compra con fecha de vencimiento $t = T$ es el valor esperado

$$C(X, T) = e^{-rT} E(\text{máx}(X(T) - K, 0)) \tag{9.36}$$

donde $X(t)$ está dado por (9.35). La sorpresa en su deducción fue la sustitución de la desviación m en (9.34) por la tasa de interés r en (9.35). De hecho, ¡la tasa de crecimiento proyectada de la acción resulta ser irrelevante para el precio de la opción! Esto se desprende de la hipótesis de no arbitraje, una piedra angular de la teoría de Black-Scholes, que dice que en un mercado eficiente no hay ganancias disponibles que estén libres de riesgo.

La fórmula (9.36) depende del valor esperado de la variable aleatoria $X(T)$, que sólo está disponible a través de la simulación. Así, además de esta idea, Black y Scholes [1973] proporcionaron una expresión de forma cerrada para el precio de la opción de compra, a saber,

$$C(X, T) = XN(d_1) - Ke^{-rT}N(d_2), \tag{9.37}$$

donde $N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-s^2/2} ds$ es la función de distribución normal acumulada y

$$d_1 = \frac{\ln(X/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}, \quad d_2 = \frac{\ln(X/K) + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}.$$

La ecuación (9.37) se conoce como la **fórmula de Black-Scholes.**

Actividades sugeridas:

Suponga que una acción de la compañía ABC tiene un precio de \$12. Considere una opción call europea con precio del ejercicio de \$15 y fecha del ejercicio en seis meses a partir de hoy, así que $T = 0.5$ años. Suponga que hay una tasa de interés fija de $r = 0.05$ y que la volatilidad de la acción es de 0.25 (es decir, 25 por ciento anual).

1. Realice una simulación de Monte Carlo para calcular el valor esperado de (9.36). Utilice el método de Euler-Maruyama para aproximar la solución de (9.35), con un tamaño de paso $h = 0.01$ y el valor inicial $X_0 = 12$. Tenga en cuenta que la EDE (9.34) no es relevante para este cálculo. Ejecute un mínimo de 10000 repeticiones.

2. Compare su aproximación en el paso 1 con el valor correcto de la fórmula de Black-Scholes (9.37). La función $N(x)$ puede calcularse usando la función de error de MATLAB `erf` como $N(x) = (1 + \text{erf}(x/\sqrt{2}))/2$.

3. Reemplace Euler-Maruyama por el método de Milstein y repita el paso 1. Compare los errores de los dos métodos.

4. Una opción **put** (europea) se diferencia de una opción call en que representa el derecho a vender, no a comprar, al precio del ejercicio. El valor de una opción put es

$$P(X, T) = e^{-rT} E(\text{máx}(K - X(T), 0)), \tag{9.38}$$

utilizando $X(T)$ a partir de (9.35). Calcule el valor mediante la simulación de Monte Carlo para los mismos datos del paso 1, utilizando tanto Euler-Maruyama como el método de Milstein.

5. Compare su aproximación en el paso 4 con la fórmula de Black-Scholes para una opción de venta:

$$P(X, T) = Ke^{-rT} N(-d_2) - XN(-d_1). \tag{9.39}$$

6. Una **opción con barrera** down-and-out tiene un pago que se cancela si la acción cruza un nivel determinado. Considere la opción call con barrera que tiene un precio del ejercicio $K = \$15$ y una barrera $L = \$10$. El pago es de máx$(X - K, 0)$ si $X(t) > L$ para $0 < t < T$, y 0 en caso contrario. Diseñe y ejecute una simulación de Monte Carlo, utilizando el movimiento browniano geométrico (9.35) y (9.36) modificado para el pago de una opción con barrera. Compare con el valor correcto

$$V(X, T) = C(X, T) - \left(\frac{X}{L}\right)^{1-2r/\sigma^2} C(L^2/X, T),$$

donde $C(X, T)$ es el valor estándar de la call europea con precio del ejercicio K. Consulte Wilmott *et al.* [1995], McDonald [2005] y Hull [2008] para mayores detalles sobre las opciones exóticas, sus fórmulas de precios y el papel de la simulación de Monte Carlo en las finanzas.

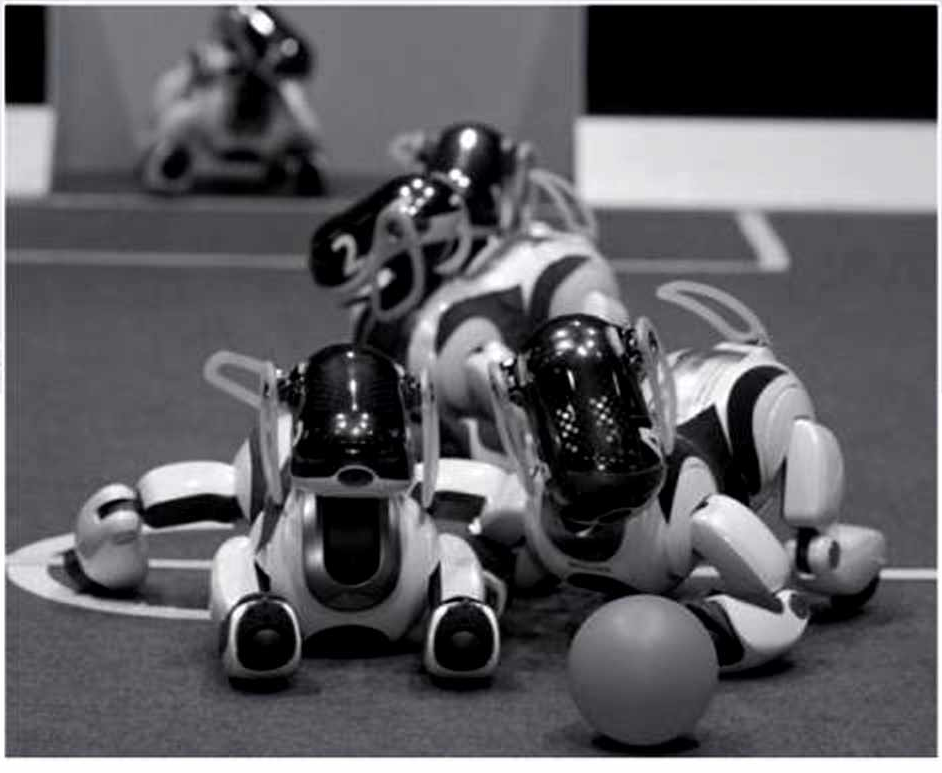
Software y lecturas adicionales

El libro de texto Gentle [2003] es una introducción al problema de la generación de números aleatorios. Otras fuentes clásicas en el campo son Knuth [1997] y Neiderreiter [1992]. En Hellekalek [1998] puede encontrarse una comparación de los métodos para generar números aleatorios y un análisis de los criterios de evaluación más comunes.

El problema `randu` se aborda en Marsaglia [1968]. El generador estándar mínimo se introdujo en Park y Miller [1988]. El generador de números aleatorios de MATLAB se basa en los métodos de resta-con-préstamo descritos por Marsaglia y Zaman [1991]. Entre las fuentes de información más completas sobre Monte Carlo y sus aplicaciones se encuentran Fishman [1996] y Rubenstein [1981].

Los libros de texto modernos sobre ecuaciones diferenciales estocásticas incluyen a Oksendal [1998] y Klebaner [1998]. El estudio adecuado de esta área requiere una sólida formación en probabilidad básica. Los aspectos de cálculo de las EDE se tratan con mayor detalle en Kloeden y Platen [1992], y en el manual orientado a la aplicación manual de Kloeden *et al.* [1994]. El artículo de Higham [2001] es una introducción muy interesante que incluye el software MATLAB para algoritmos básicos. Steele [2001] es una introducción a las ecuaciones diferenciales estocásticas ilustradas mediante numerosas aplicaciones financieras.

CAPÍTULO



10

Interpolación trigonométrica y la TRF

El chip de procesamiento de señales digitales (PSD) es la columna vertebral de los aparatos electrónicos avanzados de consumo. Los teléfonos celulares, los controladores de CD y DVD, los aparatos electrónicos para automóvil, los asistentes personales digitales, los módems, las cámaras digitales y los televisores, todos hacen uso de estos dispositivos siempre presentes. La característica principal del chip para PSD es su capacidad de hacer cálculos digitales rápidos, incluyendo la transformada rápida de Fourier (TRF).

Una de las funciones más básicas del PSD es separar la información de entrada deseada del ruido no deseado mediante su capacidad de filtrado. La posibilidad de extraer señales de un medio desordenado es una parte importante de la búsqueda constante para construir software de reconocimiento de voz confiable. También es un elemento clave de los dispositivos de reconocimiento de patrones, utilizados por los perros robot que juegan al fútbol, los cuales convierten los estímulos sensoriales en datos utilizables.

Comprobación en la realidad En la página 492 se describe el filtro de Wiener, un bloque de construcción fundamental para la reducción de ruido a través del PSD.

Hace medio siglo, ni siquiera el profesor de trigonometría más optimista podría haber previsto el impacto que han tenido los senos y los cosenos en la tecnología moderna. Como se vio en el capítulo 4, las funciones trigonométricas de frecuencias múltiples son funciones de interpolación naturales para los datos periódicos. La transformada de Fourier es casi tan eficaz al llevar a cabo una interpolación y es insustituible en las aplicaciones con datos intensivos para el procesamiento de señales moderno.

La eficacia de la interpolación trigonométrica está ligada con el concepto de ortogonalidad. Se verá que las funciones ortogonales básicas hacen que la interpolación y el ajuste por mínimos

cuadrados de los datos sean mucho más sencillos y precisos. La transformada de Fourier explota esta ortogonalidad y proporciona un medio eficaz para interpolar con senos y cosenos. El descubrimiento computacional de Cooley y Tukey, llamado la transformada rápida de Fourier (TRF), implica que la TDF puede calcularse de una manera muy sencilla.

Este capítulo trata de las ideas básicas de la transformada discreta de Fourier (TDF), incluyendo una introducción breve a los números complejos. El papel de la TDF en la interpolación trigonométrica y la aproximación por mínimos cuadrados se caracteriza y visualiza como un caso especial de la aproximación mediante funciones ortogonales básicas. Ésta es la esencia del filtrado y procesamiento de señales digitales.

10.1 LA TRANSFORMADA DE FOURIER

El matemático francés Jean Baptiste Joseph Fourier, después de escapar de la guillotina durante la Revolución Francesa y de ir a la guerra al lado de Napoleón, encontró tiempo para desarrollar una teoría de la conducción del calor. Para hacer el trabajo teórico necesitaba expandir funciones (no en términos de polinomios, como las series de Taylor, sino de un modo revolucionario desarrollado primero por Euler y Bernoulli) en términos de las funciones seno y coseno. Aunque rechazados por los principales matemáticos de la época debido a una supuesta falta de rigor, en la actualidad los métodos de Fourier están presentes en muchas áreas de las matemáticas aplicadas, la física y la ingeniería. En esta sección se presenta la transformada discreta de Fourier y se describe un algoritmo eficaz para calcularla: la transformada rápida de Fourier.

10.1.1 Aritmética compleja

Los requisitos de conteo para las funciones trigonométricas pueden simplificarse mucho adoptando el lenguaje de los números complejos. Cada número complejo tiene la forma $z = a + bi$, donde $i = \sqrt{-1}$. Cada z se representa geométricamente como un vector bidimensional de tamaño a a lo largo del eje x (horizontal) y tamaño b a lo largo del eje imaginario (vertical), como se muestra en la figura 10.1. La **magnitud compleja** del número $z = a + bi$ se define como $|z| = \sqrt{a^2 + b^2}$ y es justo la distancia del número complejo al origen en el plano complejo. El **conjugado complejo** de un número complejo $z = a + bi$ es $\bar{z} = a - bi$.

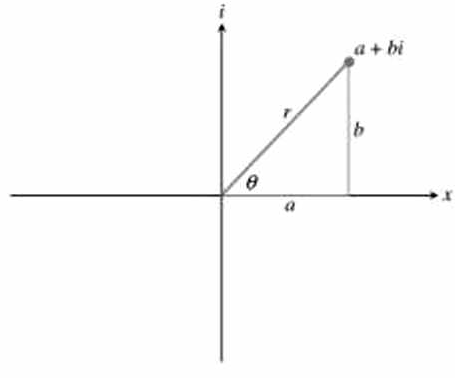


Figura 10.1 Representación de un número complejo. Las partes real e imaginaria son a y bi, respectivamente. La representación polar es $a + bi = re^{i\theta}$.

La célebre **fórmula de Euler** para la aritmética compleja dice que $e^{i\theta} = \cos\theta + i\,\text{sen}\theta$. La magnitud compleja de $z = e^{i\theta}$ es 1, así que los números complejos con esta forma caen sobre el círculo unitario en el plano complejo, como se muestra en la figura 10.2. Cualquier número complejo $a + bi$ puede escribirse en su **representación polar**

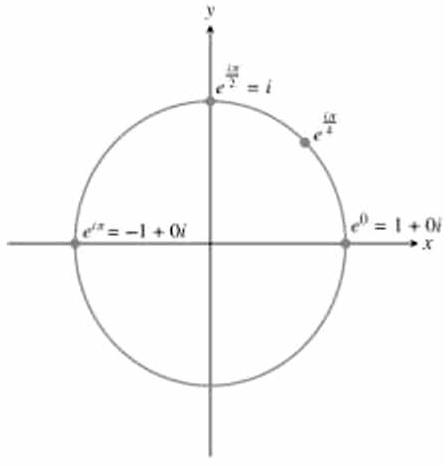


Figura 10.2 Círculo unitario en el plano complejo. Los números complejos de la forma $e^{i\theta}$ para algún ángulo θ tienen magnitud uno y caen sobre el círculo unitario.

$$z = a + bi = re^{i\theta}, \tag{10.1}$$

donde r es la magnitud compleja $|z| = \sqrt{a^2 + b^2}$ y $\theta = \arctan b/a$.

El círculo unitario en el plano complejo corresponde a los números complejos de magnitud $r = 1$. Para multiplicar los dos números $e^{i\theta}$ y $e^{i\gamma}$ en el círculo unitario, podrían convertirse en funciones trigonométricas para después multiplicar:

$$\begin{aligned} e^{i\theta}e^{i\gamma} &= (\cos\theta + i\,\mathrm{sen}\,\theta)(\cos\gamma + i\,\mathrm{sen}\,\gamma) \\ &= \cos\theta\cos\gamma - \mathrm{sen}\,\theta\,\mathrm{sen}\,\gamma + i(\mathrm{sen}\,\theta\cos\gamma + \mathrm{sen}\,\gamma\cos\theta). \end{aligned}$$

Reagrupando y usando la fórmula de la suma de cosenos y la fórmula de la suma de senos, esto puede reescribirse como

$$\cos(\theta + \gamma) + i\,\mathrm{sen}(\theta + \gamma) = e^{i(\theta+\gamma)}.$$

De manera equivalente, sólo se suman los exponentes:

$$e^{i\theta}e^{i\gamma} = e^{i(\theta+\gamma)}. \tag{10.2}$$

La ecuación (10.2) muestra que el producto de dos números en el círculo unitario da un nuevo punto en el círculo unitario, cuyo ángulo es la suma de los dos ángulos anteriores. La fórmula de Euler oculta los detalles de la trigonometría, como las fórmulas de las sumas de senos y cosenos, y hace que el conteo sea mucho más fácil. Ésta es la razón por la que se introduce la aritmética compleja en el estudio de la interpolación trigonométrica. A pesar de que puede hacerse en su totalidad con números reales, la fórmula de Euler tiene un profundo efecto de simplificación.

Se selecciona un subconjunto especial de números complejos con magnitud 1. Un número complejo z es una n-ésima **raíz de unidad** si $z^n = 1$. En el eje x, sólo hay dos raíces de unidad, -1 y 1. Sin embargo, en el plano complejo, hay muchas. Por ejemplo, i es una cuarta raíz de unidad, porque $i^4 = (-1)^2 = 1$.

Una n-ésima raíz de unidad se llama **primitiva** si no es una k-ésima raíz de unidad para cualquier $k < n$. Según esta definición, -1 es una segunda raíz de unidad primitiva y una cuarta raíz de unidad no primitiva. Es fácil comprobar que, para cualquier entero n, el número complejo $w_n = e^{-i2\pi/n}$ es una n-ésima raíz de unidad primitiva. El número $e^{i2\pi/n}$ también es una n-ésima raíz de unidad primitiva, pero se seguirá la convención usual de utilizar el primer número como base de la transformada de Fourier. En la figura 10.3 se muestra una octava raíz de unidad primitiva $w_8 = e^{-i2\pi/8}$ y las otras siete raíces de unidad, que son potencias de w_8.

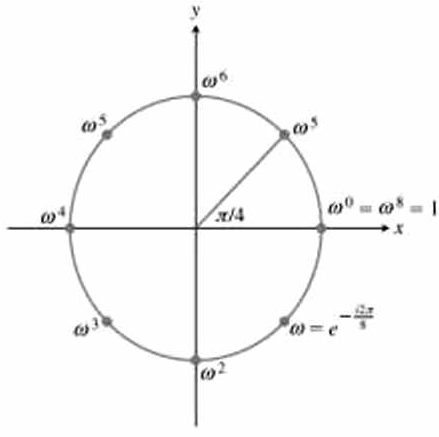


Figura 10.3 Raíces de unidad. Se muestran las ocho octavas raíces de unidad. Éstas se generan mediante $\omega = e^{-i2\pi/8}$, lo que significa que cada una es ω^k para algún entero k. Aunque ω y ω^3 son octavas raíces de unidad primitivas, ω^2 no lo es debido a que también es una cuarta raíz de unidad.

Aquí hay una identidad fundamental que se necesitará más tarde para simplificar los cálculos de la transformada discreta de Fourier. Sea ω la n-ésima raíz de unidad $\omega = e^{-i2\pi/n}$, donde $n > 1$. Entonces,

$$1 + \omega + \omega^2 + \omega^3 + \cdots + \omega^{n-1} = 0. \tag{10.3}$$

La demostración de esta identidad se desprende de la suma telescópica

$$(1 - \omega)(1 + \omega + \omega^2 + \omega^3 + \cdots + \omega^{n-1}) = 1 - \omega^n = 0. \tag{10.4}$$

Como el primer término de la izquierda no es igual a cero, el segundo debe serlo. Un método similar muestra que

$$\begin{aligned} 1 + \omega^2 + \omega^4 + \omega^6 + \cdots + \omega^{2(n-1)} &= 0, \\ 1 + \omega^3 + \omega^6 + \omega^9 + \cdots + \omega^{3(n-1)} &= 0, \\ &\vdots \\ 1 + \omega^{n-1} + \omega^{(n-1)2} + \omega^{(n-1)3} + \cdots + \omega^{(n-1)(n-1)} &= 0. \end{aligned} \tag{10.5}$$

El siguiente es diferente:

$$\begin{aligned} 1 + \omega^n + \omega^{2n} + \omega^{3n} + \cdots + \omega^{n(n-1)} &= 1 + 1 + 1 + 1 + \cdots + 1 \\ &= n. \end{aligned} \tag{10.6}$$

Esta información se concentra en el siguiente lema.

LEMA 10.1 **Raíces de unidad primitivas.** Sean ω una n-ésima raíz de unidad primitiva y k un número entero. Entonces

$$\sum_{j=0}^{n-1} \omega^{jk} = \begin{cases} n & \text{si } k/n \text{ es un entero} \\ 0 & \text{de otra manera} \end{cases}. \qquad \blacksquare$$

El ejercicio 6 le pide al lector que complete los detalles de la demostración.

10.1.2 Transformada discreta de Fourier

Considere que $x = [x_0, \ldots, x_{n-1}]^T$ es un vector n-dimensional (con valores reales) y que $\omega = e^{-i2\pi/n}$. Ésta es la definición fundamental de este capítulo.

DEFINICIÓN 10.2 La **transformada discreta de Fourier** (TDF) de $x = [x_0, \ldots, x_{n-1}]^T$ es el vector n-dimensional $y = [y_0, \ldots, y_{n-1}]$, donde $\omega = e^{-i2\pi/n}$ y

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}. \tag{10.7}$$

Por ejemplo, el lema 10.1 muestra que la TDF de $x = [1, 1, \ldots, 1]$ es $y = [\sqrt{n}, 0, \ldots, 0]$. En términos matriciales, esta definición dice

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 + ib_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ \vdots \\ a_{n-1} + ib_{n-1} \end{bmatrix} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}. \tag{10.8}$$

Cada $y_k = a_k + ib_k$ es un número complejo. La matriz de $n \times n$ en (10.8) es la **matriz de Fourier**

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}. \tag{10.9}$$

A excepción del renglón superior, cada renglón de la matriz de Fourier se suma a cero; esto también se cumple para las columnas, puesto que F_n es una matriz simétrica. La matriz de Fourier tiene una inversa explícita

$$F_n^{-1} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \cdots & \omega^{-3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)^2} \end{bmatrix}, \tag{10.10}$$

y la **transformada discreta de Fourier inversa** del vector y es $x = F_n^{-1} y$. La comprobación de que (10.10) es la inversa de la matriz F_n requiere del lema 11.1 acerca de las n-ésimas raíces de unidad. Vea el ejercicio 8.

Sea $z = e^{i\theta} = \cos\theta + i \operatorname{sen}\theta$ un punto sobre el círculo unitario. Entonces su recíproco $e^{-i\theta} = \cos\theta - i \operatorname{sen}\theta$ es su conjugado complejo. Por lo tanto, la TDF inversa es la matriz de los conjugados complejos de las entradas de F_n:

$$F_n^{-1} = \overline{F}_n. \tag{10.11}$$

DEFINICIÓN 10.3 La magnitud de un vector complejo v es el número real $||v|| = \sqrt{\overline{v}^T v}$. Una matriz compleja cuadrada F es **unitaria** si $\overline{F}^T F = I$. ❐

Una matriz unitaria, como la matriz de Fourier, es la versión compleja de una matriz ortogonal real. Si F es unitaria, entonces $||Fv||^2 = \overline{v}^T \overline{F}^T Fv = \overline{v}^T v = ||v||^2$. Por lo tanto, la magnitud de un vector no se altera al multiplicar en el lado izquierdo por F —o F^{-1} según sea el caso.

La aplicación de la transformada discreta de Fourier implica multiplicar por la matriz F_n de $n \times n$, y por lo tanto requiere $O(n^2)$ operaciones (en específico n^2 multiplicaciones y $n\,(n-1)$ sumas). La inversa de la transformada discreta de Fourier, que se aplica al multiplicar por F_n^{-1}, también es un proceso de $O(n^2)$ operaciones. En la sección 10.1.3 se desarrolla una versión de la TDF que requiere un número de operaciones mucho menor, llamada la transformada rápida de Fourier.

► EJEMPLO 10.1 Encuentre la TDF del vector $x = [1, 0, -1, 0]^T$.

Sea ω la cuarta raíz de unidad, o bien $\omega = e^{-i\pi/2} = \cos(\pi/2) - i\,\text{sen}(\pi/2) = -i$. Al aplicar la TDF, se obtiene

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}. \tag{10.12}$$

◄

El comando `fft` de MATLAB realiza la TDF con una normalización un poco diferente, de modo que $F_n x$ se calcula mediante `fft(x)/sqrt(n)`. El comando inverso `ifft` es la inversa de `fft`. Por lo tanto, $F_n^{-1}\,y$ se calcula mediante el comando `ifft(y)*sqrt(n)` de MATLAB. En otras palabras, los comandos `fft` e `ifft` de MATLAB son inversos entre sí, aunque su normalización difiere de la definición dada aquí, que tiene la ventaja de que F_n y F_n^{-1} son matrices unitarias.

Incluso si el vector x tiene componentes que son números reales, no hay ninguna razón para que las componentes de y sean números reales. Pero si las x_j son reales, los números complejos y_k tienen una propiedad especial:

LEMA 10.4 Sea $\{y_k\}$ la TDF de $\{x_j\}$, donde las x_j son números reales. Entonces (a) y_0 es real y (b) $y_{n-k} = \overline{y}_k$ para $k = 1, \ldots, n-1$. ■

Demostración. La razón de (a) se desprende de (10.7), puesto que y_0 es la suma de las x_j dividida entre $\sqrt{n}$. El inciso (b) se deduce del hecho que

$$\omega^{n-k} = e^{-i2\pi(n-k)/n} = e^{-i2\pi} e^{i2\pi k/n} = \cos(2\pi k/n) + i\,\text{sen}(2\pi k/n)$$

mientras que

$$\omega^k = e^{-i2\pi k/n} = \cos(2\pi k/n) - i\,\text{sen}(2\pi k/n),$$

lo que implica que $\omega^{n-k} = \overline{\omega^k}$. A partir de la definición de la transformada de Fourier,

$$\begin{aligned} y_{n-k} &= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j (\omega^{n-k})^j \\ &= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j (\overline{\omega^k})^j \\ &= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \overline{x_j (\omega^k)^j} = \overline{y_k}. \end{aligned}$$

Aquí se ha utilizado el hecho de que el producto de los conjugados complejos es el conjugado del producto.

El lema 10.4 tiene una consecuencia interesante. Sea n un número par y sean las $x_0, \ldots, x_{n-1}$ números reales. Entonces la TDF las sustituye por exactamente otros n números reales $a_0, a_1, b_1, a_2, b_2, \ldots, a_{n/2}$, las partes real e imaginaria de la transformada de Fourier $y_0, \ldots, y_{n-1}$. Por ejemplo, la TDF con $n = 8$ tiene la forma

$$F_8 \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 + ib_1 \\ a_2 + ib_2 \\ a_3 + ib_3 \\ a_4 \\ a_3 - ib_3 \\ a_2 - ib_2 \\ a_1 - ib_1 \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_{\frac{n}{2}-1} \\ y_{\frac{n}{2}} \\ \overline{y}_{\frac{n}{2}-1} \\ \vdots \\ \overline{y}_1 \end{bmatrix}. \tag{10.13}$$

10.1.3 La transformada rápida de Fourier

Como se mencionó en la sección anterior, la transformada discreta de Fourier aplicada en forma tradicional a un vector de n requiere $O(n^2)$ operaciones. Cooley y Tukey [1965] encontraron una manera de completar la TDF en $O(n \log n)$ operaciones con un algoritmo llamado la **transformada rápida de Fourier** (TRF). La popularidad de la TRF para el análisis de datos se dio casi de inmediato. El campo del procesamiento de señales convertidas principalmente de analógicas a digitales, se debe en gran medida a este algoritmo. A continuación se explicará este método y se demostrará su superioridad sobre la TDF convencional (10.8) a través de un recuento de operaciones.

La TDF $F_n x$ puede escribirse como

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = \frac{1}{\sqrt{n}} M_n \begin{bmatrix} x_0 \\ \vdots \\ x_{n-1} \end{bmatrix},$$

donde

$$M_n = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}.$$

ANOTACIÓN **Complejidad** El logro de Cooley y Tukey consistente en la reducción de la complejidad de la TDF de $O(n^2)$ operaciones a $O(n \log n)$ operaciones abrió un mundo de posibilidades para los métodos que usan transformadas de Fourier. Un método que se escala "casi linealmente" con el tamaño del problema resulta muy valioso. Por ejemplo, existe la posibilidad de usarlo para datos en tiempo real, puesto que el análisis puede ocurrir aproximadamente en el mismo plazo en el que se recopilan los datos. Al desarrollo de la TRF le siguió en poco tiempo después el desarrollo de circuitos especializados, que están presentes ahora en los chips PSD para el procesamiento de señales digitales en los sistemas electrónicos de análisis y control.

Ahora se mostrará cómo calcular $z = M_n x$ en forma recursiva. Para completar la TDF es necesario dividir entre $\sqrt{n}$, o $y = F_n x = z/\sqrt{n}$.

Se inicia mostrando cómo funciona el caso $n = 4$, con lo que se ilustrará la idea principal. Después de esto, el caso general resultará claro. Sea $\omega = e^{-i2\pi/4} = -i$. La transformada discreta de Fourier es

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{10.14}$$

Escriba el producto matricial, pero cambie el orden de los términos, de modo que los términos con número par aparezcan primero:

$$\begin{aligned} z_0 &= \omega^0 x_0 + \omega^0 x_2 + \omega^0(\omega^0 x_1 + \omega^0 x_3) \\ z_1 &= \omega^0 x_0 + \omega^2 x_2 + \omega^1(\omega^0 x_1 + \omega^2 x_3) \\ z_2 &= \omega^0 x_0 + \omega^4 x_2 + \omega^2(\omega^0 x_1 + \omega^4 x_3) \\ z_3 &= \omega^0 x_0 + \omega^6 x_2 + \omega^3(\omega^0 x_1 + \omega^6 x_3) \end{aligned}$$

Si se usa el hecho de que $\omega^4 = 1$, estas ecuaciones pueden reescribirse como

$$\begin{aligned} z_0 &= (\omega^0 x_0 + \omega^0 x_2) + \omega^0(\omega^0 x_1 + \omega^0 x_3) \\ z_1 &= (\omega^0 x_0 + \omega^2 x_2) + \omega^1(\omega^0 x_1 + \omega^2 x_3) \\ z_2 &= (\omega^0 x_0 + \omega^0 x_2) + \omega^2(\omega^0 x_1 + \omega^0 x_3) \\ z_3 &= (\omega^0 x_0 + \omega^2 x_2) + \omega^3(\omega^0 x_1 + \omega^2 x_3) \end{aligned}$$

Observa que cada término entre paréntesis en las dos primeras líneas se repite de manera textual en las dos líneas inferiores. Defina

$$\begin{aligned} u_0 &= \mu^0 x_0 + \mu^0 x_2 \\ u_1 &= \mu^0 x_0 + \mu^1 x_2 \end{aligned}$$

y

$$\begin{aligned} v_0 &= \mu^0 x_1 + \mu^0 x_3 \\ v_1 &= \mu^0 x_1 + \mu^1 x_3, \end{aligned}$$

donde $\mu = \omega^2$ es una segunda raíz de unidad. Tanto $u = (u_0, u_1)^T$ como $v = (v_0, v_1)^T$ son esencialmente TDF con $n = 2$; de manera más precisa,

$$\begin{aligned} u &= M_2 \begin{bmatrix} x_0 \\ x_2 \end{bmatrix} \\ v &= M_2 \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}. \end{aligned}$$

La $M_4 x$ original puede escribirse como

$$\begin{aligned} z_0 &= u_0 + \omega^0 v_0 \\ z_1 &= u_1 + \omega^1 v_1 \\ z_2 &= u_0 + \omega^2 v_0 \\ z_3 &= u_1 + \omega^3 v_1. \end{aligned}$$

En resumen, el cálculo de la TDF(4) se ha reducido a un par de TDF(2) además de algunas multiplicaciones y sumas adicionales.

Si por un momento se hace caso omiso de la $1/\sqrt{n}$, la TDF(n) puede reducirse al cálculo de dos TDF($n/2$) más $2n - 1$ operaciones adicionales ($n - 1$ multiplicaciones y n sumas). Un conteo cuidadoso de las sumas y multiplicaciones necesarias genera el teorema 10.5.

TEOREMA 10.5 **Conteo de operaciones de la TRF.** Sea n una potencia de 2. Entonces la transformada rápida de Fourier de tamaño n puede completarse en $n(2\log_2 n - 1) + 1$ sumas y multiplicaciones, además de una división entre $\sqrt{n}$. ■

Demostración. No tome en cuenta la raíz cuadrada, que se aplica al final. El resultado es equivalente a decir que la TDF(2^m) puede completarse en $2^m(2m - 1) + 1$ sumas y multiplicaciones. De hecho, ya se vio cómo una TDF(n), donde n es par, puede reducirse a un par de TDF($n/2$). Si n es una potencia de dos (por ejemplo, $n = 2^m$) entonces, es posible descomponer en forma recursiva el problema hasta llegar a la TDF(1), que es la multiplicación por la matriz identidad de 1×1, que implica cero operaciones. Si se inicia de abajo hacia arriba, la TDF(1) no implica operaciones, y la TDF(2) requiere dos sumas y una multiplicación: $y_0 = u_0 + 1v_0$, $y_1 = u_0 + \omega v_0$, donde u_0 y v_0 son las TDF(1) (es decir, $u_0 = y_0$ y $v_0 = y_1$).

La TDF(4) requiere dos TDF(2) más $2 * 4 - 1 = 7$ operaciones adicionales, para un total de $2(3) + 7 = 2^m(2m - 1) + 1$ operaciones, donde $m = 2$. Se procede por inducción: suponga que esta fórmula es correcta para una m dada. Entonces la TDF(2^{m+1}) requiere dos TDF(2^m), que implican $2(2^m(2m - 1) + 1)$ operaciones, además de $2 \cdot 2^{m+1} - 1$ adicionales (completando un número de ecuaciones similar a (10.15)), para un total de

$$\begin{aligned} 2(2^m(2m - 1) + 1) + 2^{m+2} - 1 &= 2^{m+1}(2m - 1 + 2) + 2 - 1 \\ &= 2^{m+1}(2(m + 1) - 1) + 1. \end{aligned}$$

Por lo tanto, la fórmula de $2^m(2m - 1) + 1$ operaciones se prueba para la versión rápida de la TDF(2^m), a partir de la cual se obtiene el resultado. ❐

El algoritmo rápido para la TDF puede aprovecharse para hacer un algoritmo rápido de la TDF inversa sin trabajo adicional. La TDF inversa es la matriz compleja conjugada $\overline{F}_n$. Para llevar a cabo la TDF inversa de un vector complejo y, sólo conjugue, aplique la TRF y conjugue el resultado, porque

$$F_n^{-1} y = \overline{F}_n y = \overline{F_n \overline{y}}. \tag{10.15}$$

10.1 Ejercicios

1. Encuentre la TDF de los siguientes vectores: (a) $[0, 1, 0, -1]$ (b) $[1, 1, 1, 1]$ (c) $[0, -1, 0, 1]$ (d) $[0, 1, 0, -1, 0, 1, 0, -1]$

2. Encuentre la TDF de los siguientes vectores: (a) $[3/4, 1/4, -1/4, 1/4]$ (b) $[9/4, 1/4, -3/4, 1/4]$ (c) $[1, 0, -1/2, 0]$ (d) $[1, 0, -1/2, 0, 1, 0, -1/2, 0]$

3. Encuentre la TDF inversa de los siguientes vectores: (a) $[1, 0, 0, 0]$ (b) $[1, 1, -1, 1]$ (c) $[1, -i, 1, i]$ (d) $[1, 0, 0, 0, 3, 0, 0, 0]$

4. Encuentre la TDF inversa de los siguientes vectores: (a) $[0, -i, 0, i]$ (b) $[2, 0, 0, 0]$ (c) $[1/2, 1/2, 0, 1/2]$ (d) $[1, 3/2, 1/2, 3/2]$

5. (a) Escriba todas las cuartas raíces de unidad y todas las cuartas raíces de unidad primitivas. (b) Escriba todas las séptimas raíces de unidad primitivas. (c) ¿Cuántas p-ésimas raíces de unidad primitivas existen para un número primo p?

6. Demuestre el lema 10.1.
7. Encuentre los números reales $a_0, a_1, b_1, a_2, b_2, \ldots, a_{n/2}$ como en (10.13) para las transformadas de Fourier en el ejercicio 1.
8. Demuestre que la matriz en (10.10) es la inversa de la matriz de Fourier F_n.

10.2 INTERPOLACIÓN TRIGONOMÉTRICA

¿Qué hace en realidad la transformada discreta de Fourier? En esta sección se presenta una interpretación del vector de salida y de la transformada de Fourier como los coeficientes de interpolación para datos uniformemente espaciados, a fin de que su funcionamiento sea más comprensible.

10.2.1 Teorema de interpolación de la TDF

Sean $[c, d]$ un intervalo y n un entero positivo. Defina $\Delta t = (d - c)/n$ y $t_j = c + j\,\Delta t$ para $j = 0, \ldots, n - 1$ como puntos uniformemente espaciados en el intervalo. Dado un vector de entrada x para la transformada de Fourier, se interpretará el componente x_j como el j-ésimo componente de una señal medida. Por ejemplo, puede pensarse en los componentes de x como una serie de mediciones, tomadas en tiempos t_j discretos y uniformemente espaciados, como se muestra en la figura 10.4.

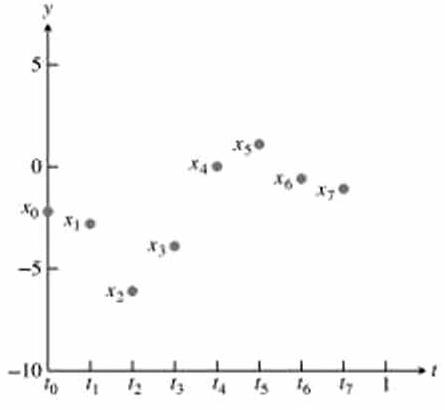


Figura 10.4 Los componentes de x vistos como una serie de tiempo. La transformada de Fourier es una forma de calcular el polinomio trigonométrico que interpola estos datos.

Sea $y = F_n x$ la TDF de x. Como x es la TDF inversa de y, es posible escribir una fórmula explícita para los componentes de x a partir de (10.10), si se recuerda que $\omega = e^{-i2\pi/n}$:

$$x_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k (\omega^{-k})^j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k e^{i2\pi kj/n} = \sum_{k=0}^{n-1} y_k \frac{e^{\frac{i2\pi k(t_j - c)}{d-c}}}{\sqrt{n}}. \tag{10.16}$$

Esto puede verse como la interpolación de los puntos (t_j, x_j) mediante las funciones trigonométricas básicas donde los coeficientes son y_k. El teorema 10.6 es una simple repetición de (10.16), diciendo que los puntos (t_j, x_j) se interpolan mediante las funciones básicas $e^{i2\pi k(t-c)/(d-c)}/\sqrt{n}$ para $k = 0, \ldots, n - 1$, con coeficientes de interpolación dados por $F_n x$.

TEOREMA 10.6 **Teorema de interpolación de la TDF.** Dados un intervalo $[c, d]$ y un entero positivo n, sea $t_j = c + j(d - c)/n$ para $j = 0, \ldots, n - 1$, y sea $x = (x_0, \ldots, x_{n-1})$ un vector de n elementos.

Defina $\vec{a} + \vec{b}i = F_n x$, donde F_n es la matriz de la transformada discreta de Fourier. Entonces, la función compleja

$$Q(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k + ib_k) e^{i2\pi k(t-c)/(d-c)}$$

satisface $Q(t_j) = x_j$ para $j = 0, \ldots, n-1$. Además, si las x_j son reales, la función real

$$P(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(a_k \cos \frac{2\pi k(t-c)}{d-c} - b_k \operatorname{sen} \frac{2\pi k(t-c)}{d-c} \right)$$

satisface $P(t_j) = x_j$ para $j = 0, \ldots, n-1$. ■

En otras palabras, la transformada de Fourier F_n transforma los datos $\{x_j\}$ en coeficientes de interpolación.

La explicación para la última parte del teorema es que, si se utiliza la fórmula de Euler, la función de interpolación en (10.16) puede escribirse como

$$Q(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} (a_k + ib_k) \left(\cos \frac{2\pi k(t-c)}{d-c} + i \operatorname{sen} \frac{2\pi k(t-c)}{d-c} \right).$$

Separando la función de interpolación $Q(t) = P(t) + iI(t)$ en sus partes real e imaginaria. Como las x_j son números reales, sólo se requiere la parte real de $Q(t)$ para interpolar las x_j. La parte real es

$$P(t) = P_n(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(a_k \cos \frac{2\pi k(t-c)}{d-c} - b_k \operatorname{sen} \frac{2\pi k(t-c)}{d-c} \right). \tag{10.17}$$

Un subíndice n identifica el número de términos en el modelo trigonométrico. En ocasiones P_n se llamará **función trigonométrica** de **orden** n. El lema 10.4 y el siguiente lema 10.7 pueden utilizarse para simplificar aún más la función de interpolación $P_n(t)$:

LEMA 10.7 Sea $t = j/n$, donde j y n son números enteros. Sea k un entero. Entonces

$$\cos 2(n-k)\pi t = \cos 2k\pi t \text{ y } \operatorname{sen} 2(n-k)\pi t = -\operatorname{sen} 2k\pi t. \tag{10.18}$$

■

De hecho, la expresión de la suma de cosenos produce $\cos 2(n-k)\pi j/n = \cos(2\pi j - 2jk\pi/n) = \cos(-2jk\pi/n)$ y de manera similar para los senos.

El lema 10.7, junto con el lema 10.4, implica que la segunda mitad de la expansión trigonométrica (10,17) es redundante. Es posible interpolar en las t_j utilizando sólo la primera mitad de los términos (excepto por un cambio de signo en los términos sinusoidales). Por el lema 10.4, los coeficientes de la segunda mitad de la expansión son iguales a los de la primera mitad (excepto por un cambio de signo de los términos sinusoidales). Por lo tanto, los cambios de signo se cancelan mutuamente, y se ha demostrado que la versión simplificada de P_n es

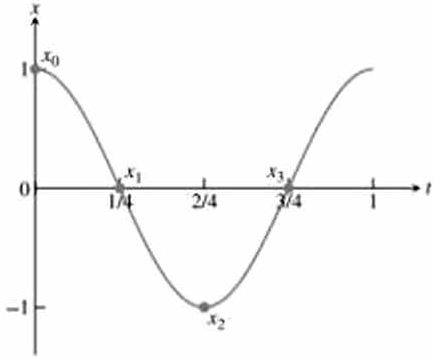


Figura 10.5 Interpolación trigonométrica. El vector de entrada x es $[1, 0, -1, 0]^T$. La fórmula (10.19) proporciona la función de interpolación $P_4(t) = \cos 2\pi t$.

$$P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi(t-c)}{d-c} - b_k \operatorname{sen} \frac{2k\pi(t-c)}{d-c} \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \frac{n\pi(t-c)}{d-c}.$$

Para escribir esta expresión se ha supuesto que n es par. La fórmula es un poco diferente para una n impar. Vea el ejercicio 5.

COROLARIO 10.8 Para un entero par n, sea $t_j = c + j(d-c)/n$ para $j = 0, \ldots, n-1$, y sea $x = (x_0, \ldots, x_{n-1})$ un vector de n números reales. Defina $\vec{a} + \vec{b}i = F_n x$, donde F_n es la transformada discreta de Fourier. Entonces, la función

$$P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi(t-c)}{d-c} - b_k \operatorname{sen} \frac{2k\pi(t-c)}{d-c} \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \frac{n\pi(t-c)}{d-c} \tag{10.19}$$

satisface $P_n(t_j) = x_j$ para $j = 0, \ldots, n-1$. ■

► EJEMPLO 10.2 Encuentre el interpolante trigonométrico para el ejemplo 10.1.

El intervalo es $[c, d] = [0, 1]$. Sea $x = [1, 0, -1, 0]^T$ y calcule su TDF como $y = [0, 1, 0, 1]^T$. Los coeficientes de interpolación son $a_k + ib_k = y_k$. Por lo tanto, $a_0 = a_2 = 0$, $a_1 = a_3 = 1$ y $b_0 = b_1 = b_2 = b_3 = 0$. De acuerdo con (10.19), sólo se requieren a_0, a_1, a_2 y b_1. Una función de interpolación trigonométrica para x está dada por

$$\begin{aligned} P_4(t) &= \frac{a_0}{2} + (a_1 \cos 2\pi t - b_1 \operatorname{sen} 2\pi t) + \frac{a_2}{2} \cos 4\pi t \\ &= \cos 2\pi t. \end{aligned}$$

En la figura 10.5 se muestra la interpolación de los puntos (t, x), donde $t = [0, 1/4, 1/2, 3/4]$ y $x = [1, 0, -1, 0]$. ◄

► EJEMPLO 10.3 Encuentre el interpolante trigonométrico para los datos de la temperatura del ejemplo 4.6: $x = [-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, -1.1]$ en el intervalo $[0, 1]$.

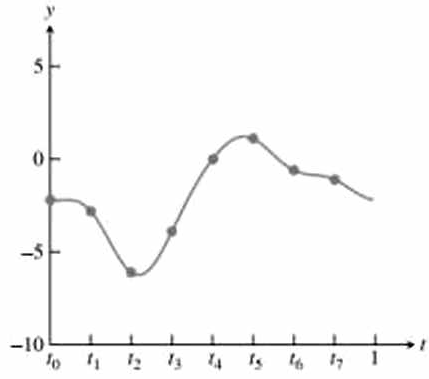


Figura 10.6 Interpolación trigonométrica de los datos del ejemplo 4.6. Se interpolan los datos $t = [0, 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8]$, $x = [-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, -1.1]$ usando la transformada de Fourier con $n = 8$. La gráfica se hizo mediante el programa 10.1 con $p = 100$.

La salida de la transformada de Fourier con una precisión de cuatro decimales, es

$$y = \begin{bmatrix} -5.5154 & & \\ -1.0528 & + & 3.6195i \\ 1.5910 & - & 1.1667i \\ -0.5028 & - & 0.2695i \\ -0.7778 & & \\ -0.5028 & + & 0.2695i \\ 1.5910 & + & 1.1667i \\ -1.0528 & - & 3.6195i \end{bmatrix}.$$

De acuerdo con la fórmula (10.19), la función de interpolación es

$$\begin{aligned} P_8(t) &= \frac{-5.5154}{\sqrt{8}} - \frac{1.0528}{\sqrt{2}}\cos 2\pi t - \frac{3.6195}{\sqrt{2}}\operatorname{sen} 2\pi t \\ &\quad + \frac{1.5910}{\sqrt{2}}\cos 4\pi t + \frac{1.1667}{\sqrt{2}}\operatorname{sen} 4\pi t \\ &\quad - \frac{0.5028}{\sqrt{2}}\cos 6\pi t + \frac{0.2695}{\sqrt{2}}\operatorname{sen} 6\pi t \\ &\quad - \frac{0.7778}{\sqrt{8}}\cos 8\pi t \\ &= -1.95 - 0.7445\cos 2\pi t - 2.5594\operatorname{sen} 2\pi t \\ &\quad + 1.125\cos 4\pi t + 0.825\operatorname{sen} 4\pi t \\ &\quad - 0.3555\cos 6\pi t + 0.1906\operatorname{sen} 6\pi t \\ &\quad - 0.2750\cos 8\pi t. \end{aligned} \tag{10.20}$$

En la figura 10.6 se muestra la gráfica y la función de interpolación trigonométrica. ◀

10.2.2 Evaluación eficiente de funciones trigonométricas

El corolario 10.8 es un poderoso enunciado acerca de la interpolación. Aunque parece complicado al principio, hay otra manera de evaluar y representar gráficamente el polinomio de interpolación

trigonométrica de las figuras 10.5 y 10.6, utilizando la TDF para hacer todo el trabajo en vez de graficar los senos y cosenos de (10.19). Después de todo, se sabe por el teorema 10.6 que al multiplicar el vector x de los datos por F_n se cambian por los coeficientes de interpolación. En forma inversa, es posible convertir los coeficientes de interpolación en los datos. En lugar de evaluar (10.19), sólo invierta la TDF: multiplique el vector de coeficientes de interpolación $\{a_k + ib_k\}$ por F_n^{-1}.

Por supuesto, si la operación F_n está seguida por su inversa, F_n^{-1}, se obtienen los datos originales de nuevo y no se gana nada. En vez de esto, se dejará que $p \geq n$ sea un número más grande. Se considera a (10.19) como una función trigonométrica de orden p y después invertir la transformada de Fourier para evaluar la curva en p puntos igualmente espaciados. Puede tomarse una p suficientemente grande como para obtener una gráfica con apariencia continua.

Para ver los coeficientes de $P_n(t)$ como los coeficientes de un polinomio trigonométrico de orden p, observe que es posible reescribir (10.19) como

$$P_p(t) = \frac{\sqrt{\frac{p}{n}}a_0}{\sqrt{p}} + \frac{2}{\sqrt{p}}\sum_{k=1}^{p/2-1}\left(\sqrt{\frac{p}{n}}a_k \cos\frac{2k\pi(t-c)}{d-c} - \sqrt{\frac{p}{n}}b_k \operatorname{sen}\frac{2k\pi(t-c)}{d-c}\right) + \frac{\sqrt{\frac{p}{n}}a_{n/2}}{\sqrt{p}}\cos n\pi t \tag{10.21}$$

donde se establece $a_k = b_k = 0$ para $k = \frac{n}{2} + 1, \ldots, \frac{p}{2}$. Se concluye a partir de (10.21) que la manera de producir p puntos que caigan sobre la curva (10.19) en $t_j = c + j(d - c)/n$ para $j = 0, \ldots, n - 1$ consiste en multiplicar los coeficientes de Fourier por $\sqrt{p/n}$ y después invertir la TDF.

Ahora se escribirá el código de MATLAB para llevar a cabo esta idea. En términos generales, se desea implementar

$$F_p^{-1}\sqrt{\frac{p}{n}}F_n x$$

usando los comandos `fft` e `ifft` de MATLAB, donde

$$F_p^{-1} = \sqrt{p}\cdot \texttt{ifft} \quad \text{y} \quad F_n = \frac{1}{\sqrt{n}}\cdot \texttt{fft}.$$

Al igualar las ecuaciones, corresponde a las operaciones:

$$\sqrt{p}\cdot \texttt{ifft}_{[p]}\sqrt{\frac{p}{n}}\frac{1}{\sqrt{n}}\cdot \texttt{fft}_{[n]} = \frac{p}{n}\cdot \texttt{ifft}_{[p]}\cdot \texttt{fft}_{[n]}. \tag{10.22}$$

Por supuesto, F_p^{-1} sólo puede aplicarse a un vector de longitud p, por lo que es necesario colocar los coeficientes de Fourier de grado n en un vector de longitud p antes de invertir. El programa corto `dftinterp.m` lleva a cabo estos pasos.

```
%Programa 10.1 Interpolación de Fourier
%Interpola n puntos de datos en [c, d] con función de disparo P(t)
%  y grafica el interpolante en p (>=n) puntos uniformemente espaciados.
%Entrada: intervalo [c, d], puntos de datos x, número par de puntos
%  de datos n, número par p>=n
%Salida: puntos de datos del interpolante xp
function xp=dftinterp(inter,x,n,p)
c=inter(1);d=inter(2);t=c+(d-c)*(0:n-1)/n; tp=c+(d-c)*(0:p-1)/p;
y=fft(x);                    % aplica la TDF
yp=zeros(p,1);               % yp contendrá los coeficientes para ifft
yp(1:n/2+1)=y(1:n/2+1);      % mueve n frecuencias de n a p
yp(p-n/2+2:p)=y(n/2+2:n);    % lo mismo para el nivel superior
xp=real(ifft(yp))*(p/n);     % invierte fft para recuperar datos
plot(t,x,'o',tp,xp)          % grafica los puntos de datos y el interpolante
```

Por ejemplo, la ejecución de la función `dftinterp([0,1],[-2.2 -2.8 -6.1 -3.9 0.0 1.1 -0.6 -1.1],8,100)`, produce los $p = 100$ puntos graficados en la figura 10.6 sin usar en forma explícita senos o cosenos. Algunos comentarios sobre el código son los siguientes. El objetivo es aplicar $\texttt{fft}_{[n]}$ seguida por $\texttt{ifft}_{[p]}$, y luego multiplicar por p/n. Después de aplicar `fft` a los n valores en x, los coeficientes en el vector `y` se mueven de las n frecuencias en $P_n(t)$ a un vector `yp` que contiene p frecuencias, donde $p \geq n$. Hay muchas frecuencias más altas entre las p frecuencias que no son utilizadas por P_n, lo que conduce a coeficientes cero en esas altas frecuencias, en las posiciones de $n/2 + 2$ a $p/2 + 1$. La mitad superior de las entradas en `yp` da una recapitulación de la mitad inferior, con conjugados complejos y en orden inverso, de acuerdo con (10.13). Después de invertir la TDF con el comando `ifft`, aunque en teoría el resultado es real, computacionalmente puede haber una pequeña parte imaginaria debido al redondeo. Esto se elimina mediante la aplicación del comando `real`.

Un caso muy simple y útil es $c = 0$, $d = n$. Los puntos de datos x_j se recogen en los nodos de interpolación enteros $s_j = j$ para $j = 0, \ldots, n-1$. Los puntos (j, x_j) se interpolan mediante la función trigonométrica

$$P_n(s) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} \left(a_k \cos \frac{2k\pi}{n} s - b_k \operatorname{sen} \frac{2k\pi}{n} s \right) + \frac{a_{n/2}}{\sqrt{n}} \cos \pi s. \qquad (10.23)$$

En el capítulo 11 se utilizarán nodos de interpolación sólo enteros, para tener compatibilidad con las convenciones habituales de los algoritmos de compresión de datos de audio e imagen.

10.2 Ejercicios

1. Utilice la TDF y el corolario 10.8 para encontrar la función de interpolación trigonométrica para los siguientes datos:

(a)

t	x
0	0
$\frac{1}{4}$	1
$\frac{1}{2}$	0
$\frac{3}{4}$	-1

(b)

t	x
0	1
$\frac{1}{4}$	1
$\frac{1}{2}$	-1
$\frac{3}{4}$	-1

(c)

t	x
0	-1
$\frac{1}{4}$	1
$\frac{1}{2}$	-1
$\frac{3}{4}$	1

(d)

t	x
0	1
$\frac{1}{4}$	1
$\frac{1}{2}$	1
$\frac{3}{4}$	1

2. Utilice (10.23) para encontrar la función de interpolación trigonométrica para los datos siguientes:

(a)

t	x
0	0
1	1
2	0
3	-1

(b)

t	x
0	1
1	1
2	-1
3	-1

(c)

t	x
0	1
1	2
2	4
3	1

(d)

t	x
0	1
1	0
2	1
3	0

3. Encuentre la función de interpolación trigonométrica para los siguientes datos:

(a)

t	x
0	0
$\frac{1}{8}$	1
$\frac{1}{4}$	0
$\frac{3}{8}$	-1
$\frac{1}{2}$	0
$\frac{5}{8}$	1
$\frac{3}{4}$	0
$\frac{7}{8}$	-1

(b)

t	x
0	1
$\frac{1}{8}$	2
$\frac{1}{4}$	1
$\frac{3}{8}$	0
$\frac{1}{2}$	1
$\frac{5}{8}$	2
$\frac{3}{4}$	1
$\frac{7}{8}$	0

(c)

t	x
0	1
$\frac{1}{8}$	1
$\frac{1}{4}$	1
$\frac{3}{8}$	1
$\frac{1}{2}$	0
$\frac{5}{8}$	0
$\frac{3}{4}$	0
$\frac{7}{8}$	0

(d)

t	x
0	1
$\frac{1}{8}$	-1
$\frac{1}{4}$	1
$\frac{3}{8}$	-1
$\frac{1}{2}$	1
$\frac{5}{8}$	-1
$\frac{3}{4}$	1
$\frac{7}{8}$	-1

4. Encuentre la función de interpolación trigonométrica para los siguientes datos:

(a)

t	x
0	0
1	1
2	0
3	−1
4	0
5	1
6	0
7	−1

(b)

t	x
0	1
1	2
2	1
3	0
4	1
5	2
6	1
7	0

(c)

t	x
0	1
1	0
2	1
3	0
4	1
5	0
6	1
7	0

(d)

t	x
0	−1
1	0
2	0
3	0
4	1
5	0
6	0
7	0

5. Encuentre una versión de (10.19) para la función de interpolación en el caso donde n es impar.

10.2 Problemas de computadora

1. Encuentre la función de interpolación trigonométrica de orden 8 $P_8(t)$ para los siguientes datos:

(a)

t	x
0	0
$\frac{1}{8}$	1
$\frac{1}{4}$	2
$\frac{3}{8}$	3
$\frac{1}{2}$	4
$\frac{5}{8}$	5
$\frac{3}{4}$	6
$\frac{7}{8}$	7

(b)

t	x
0	2
$\frac{1}{8}$	−1
$\frac{1}{4}$	0
$\frac{3}{8}$	1
$\frac{1}{2}$	1
$\frac{5}{8}$	3
$\frac{3}{4}$	−1
$\frac{7}{8}$	−1

(c)

t	x
0	3
1	1
2	4
3	2
4	3
5	1
6	4
7	2

(d)

t	x
1	1
2	−2
3	5
4	3
5	−2
6	−3
7	1
8	2

Grafique los puntos de datos y $P_8(t)$.

2. Encuentre la función de interpolación trigonométrica de orden 8 $P_8(t)$ para los siguientes datos:

(a)

t	x
0	6
$\frac{1}{8}$	5
$\frac{1}{4}$	4
$\frac{3}{8}$	3
$\frac{1}{2}$	2
$\frac{5}{8}$	1
$\frac{3}{4}$	0
$\frac{7}{8}$	−1

(b)

t	x
0	3
$\frac{1}{8}$	1
$\frac{1}{4}$	2
$\frac{3}{8}$	−1
$\frac{1}{2}$	−1
$\frac{5}{8}$	−2
$\frac{3}{4}$	3
$\frac{7}{8}$	0

(c)

t	x
0	1
2	2
4	4
6	−1
8	0
10	1
12	0
14	2

(d)

t	x
−7	2
−5	1
−3	0
−1	5
1	7
3	2
5	1
7	−4

Grafique los puntos de datos y $P_8(t)$.

3. Encuentre la función de interpolación trigonométrica de orden $n = 8$ para $f(t) = e^t$ en los puntos uniformemente espaciados $(j/8, f(j/8))$ para $j = 0, \ldots, 7$. Grafique $f(t)$, los puntos de datos y la función de interpolación.

4. Grafique la función de interpolación $P_n(t)$ en $[0, 1]$ del problema de computadora 3, junto con los puntos de datos y $f(t) = e^t$ para (a) $n = 16$ (b) $n = 32$.

5. Encuentre la función de interpolación trigonométrica de orden 8 para $f(t) = \ln t$ en los puntos uniformemente espaciados $(1 + j/8, f(1 + j/8))$ para $j = 0, \ldots, 7$. Grafique $f(t)$, los puntos de datos y la función de interpolación.
6. Grafique la función de interpolación $P_n(t)$ en $[0, 1]$ del problema de computadora 5, junto con los puntos de datos y $f(t) = \ln t$ para (a) $n = 16$ (b) $n = 32$.

10.3 FFT Y EL PROCESAMIENTO DE SEÑALES

El teorema 10.6 de la TDF de interpolación es sólo una aplicación de la transformada de Fourier. En esta sección se analizará la interpolación desde un punto de vista más general, que mostrará cómo encontrar aproximaciones por mínimos cuadrados usando funciones trigonométricas. Estas ideas forman la base del procesamiento de señales moderno y harán una segunda aparición en el capítulo 11, aplicadas a la transformada discreta del coseno.

10.3.1 Ortogonalidad e interpolación

La interpolación en apariencia simple que resulta del teorema 10.6 fue posible debido a que $F_n^{-1} = \overline{F}_n^T = \overline{F}_n$, lo que hace a F_n una matriz unitaria. La versión real de esta definición se encuentra en el capítulo 4, donde se le llama matriz ortogonal U si $U^{-1} = U^T$. Ahora se estudia una forma particular de una matriz ortogonal que se traducirá de inmediato en un buen interpolante.

TEOREMA 10.9 **Teorema de interpolación de la función ortogonal.** Sean $f_0(t), \ldots, f_{n-1}(t)$ funciones de t y sean $t_0, \ldots, t_{n-1}$ números reales. Suponga que la matriz de $n \times n$

$$A = \begin{bmatrix} f_0(t_0) & f_0(t_1) & \cdots & f_0(t_{n-1}) \\ f_1(t_0) & f_1(t_1) & \cdots & f_1(t_{n-1}) \\ \vdots & \vdots & & \vdots \\ f_{n-1}(t_0) & f_{n-1}(t_1) & \cdots & f_{n-1}(t_{n-1}) \end{bmatrix} \tag{10.24}$$

es una matriz ortogonal real de $n \times n$. Si $y = Ax$, la función

$$F(t) = \sum_{k=0}^{n-1} y_k f_k(t)$$

interpola $[t_0, x_0), \ldots, (t_{n-1}, x_{n-1})$, es decir $F(t_j) = x_j$ para $j = 0, \ldots, n-1$. ■

Demostración. El hecho de que $y = Ax$ implica que

$$x = A^{-1}y = A^T y,$$

y se deduce que

$$x_j = \sum_{k=0}^{n-1} a_{kj} y_k = \sum_{k=0}^{n-1} y_k f_k(t_j)$$

para $j = 0, \ldots, n-1$, lo que completa la demostración. ❐

► EJEMPLO 10.4 Sean $[c, d]$ un intervalo y n un entero positivo. Demuestre que las hipótesis del teorema 10.9 se satisfacen para $t_j = c + j(d - c)/n, j = 0, ..., n - 1$, y

$$f_0(t) = \sqrt{\frac{1}{n}}$$

$$f_1(t) = \sqrt{\frac{2}{n}} \cos \frac{2\pi(t - c)}{d - c}$$

$$f_2(t) = \sqrt{\frac{2}{n}} \operatorname{sen} \frac{2\pi(t - c)}{d - c}$$

$$f_3(t) = \sqrt{\frac{2}{n}} \cos \frac{4\pi(t - c)}{d - c}$$

$$f_4(t) = \sqrt{\frac{2}{n}} \operatorname{sen} \frac{4\pi(t - c)}{d - c}$$

$$\vdots$$

$$f_{n-1}(t) = \frac{1}{\sqrt{n}} \cos \frac{n\pi(t - c)}{d - c}.$$

La matriz es

$$A = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ 1 & \cos\frac{2\pi}{n} & \cdots & \cos\frac{2\pi(n-1)}{n} \\ 0 & \operatorname{sen}\frac{2\pi}{n} & \cdots & \operatorname{sen}\frac{2\pi(n-1)}{n} \\ \vdots & \vdots & & \vdots \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\cos\pi & \cdots & \frac{1}{\sqrt{2}}\cos(n-1)\pi \end{bmatrix}. \tag{10.25}$$

El lema 10.10 muestra que las filas de A son ortogonales por pares. ◄

LEMA 10.10 Sean $n \geq 1$ y k, l números enteros. Entonces

$$\sum_{j=0}^{n-1} \cos\frac{2\pi jk}{n} \cos\frac{2\pi jl}{n} = \begin{cases} n & \text{si } (k-l)/n \text{ y } (k+l)/n \text{ son números enteros} \\ \frac{n}{2} & \text{si exactamente uno de } (k-l)/n \text{ y } (k+l)/n \text{ es un número entero} \\ 0 & \text{si ninguno es un número entero} \end{cases}$$

$$\sum_{j=0}^{n-1} \cos\frac{2\pi jk}{n} \operatorname{sen}\frac{2\pi jl}{n} = 0$$

$$\sum_{j=0}^{n-1} \operatorname{sen}\frac{2\pi jk}{n} \operatorname{sen}\frac{2\pi jl}{n} = \begin{cases} 0 & \text{si ambos } (k-l)/n \text{ y } (k+l)/n \text{ son números enteros} \\ \frac{n}{2} & \text{si } (k-l)/n \text{ es un número entero y } (k+l)/n \text{ no lo es} \\ -\frac{n}{2} & \text{si } (k+l)/n \text{ es un número entero y } (k-l)/n \text{ no lo es} \\ 0 & \text{si ninguno es un número entero} \end{cases}$$

■

La demostración de este lema sigue del lema 10.1. Vea el ejercicio 5.

De regreso al ejemplo 10.4, $y = Ax$. El teorema 10.9 da de inmediato la función de interpolación

$$
\begin{aligned}
F(t) = {} & \frac{1}{\sqrt{n}} y_0 \\
& + \sqrt{\frac{2}{n}} y_1 \cos \frac{2\pi(t-c)}{d-c} + \sqrt{\frac{2}{n}} y_2 \operatorname{sen} \frac{2\pi(t-c)}{d-c} \\
& + \sqrt{\frac{2}{n}} y_3 \cos \frac{4\pi(t-c)}{d-c} + \sqrt{\frac{2}{n}} y_4 \operatorname{sen} \frac{4\pi(t-c)}{d-c} \\
& \vdots \\
& + \frac{1}{\sqrt{n}} y_{n-1} \cos \frac{n\pi(t-c)}{d-c}
\end{aligned} \tag{10.26}
$$

para los puntos (t_j, x_j), de acuerdo con (10.19).

► **EJEMPLO 10.5** Utilice las funciones básicas del ejemplo 10.4 para interpolar los datos $x = [-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, -1.1]$ del ejemplo 10.3.

Al calcular el producto de la matriz A de 8×8 por x se obtiene

$$
Ax = \sqrt{\frac{2}{8}} \begin{bmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\
1 & \cos 2\pi\frac{1}{8} & \cos 2\pi\frac{2}{8} & \cdots & \cos 2\pi\frac{7}{8} \\
0 & \operatorname{sen} 2\pi\frac{1}{8} & \operatorname{sen} 2\pi\frac{2}{8} & \cdots & \operatorname{sen} 2\pi\frac{7}{8} \\
1 & \cos 4\pi\frac{1}{8} & \cos 4\pi\frac{2}{8} & \cdots & \cos 4\pi\frac{7}{8} \\
0 & \operatorname{sen} 4\pi\frac{1}{8} & \operatorname{sen} 4\pi\frac{2}{8} & \cdots & \operatorname{sen} 4\pi\frac{7}{8} \\
1 & \cos 6\pi\frac{1}{8} & \cos 6\pi\frac{2}{8} & \cdots & \cos 6\pi\frac{7}{8} \\
0 & \operatorname{sen} 6\pi\frac{1}{8} & \operatorname{sen} 6\pi\frac{2}{8} & \cdots & \operatorname{sen} 6\pi\frac{7}{8} \\
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\cos\pi & \frac{1}{\sqrt{2}}\cos 2\pi & \cdots & \frac{1}{\sqrt{2}}\cos 7\pi
\end{bmatrix}
\begin{bmatrix} -2.2 \\ -2.8 \\ -6.1 \\ -3.9 \\ 0.0 \\ 1.1 \\ -0.6 \\ -1.1 \end{bmatrix}
= \begin{bmatrix} -5.5154 \\ -1.4889 \\ -5.1188 \\ 2.2500 \\ 1.6500 \\ -0.7111 \\ 0.3812 \\ -0.7778 \end{bmatrix}.
$$

La fórmula (10.26) proporciona la función de interpolación,

$$
\begin{aligned}
P(t) = {} & -1.95 - 0.7445\cos 2\pi t - 2.5594\operatorname{sen} 2\pi t \\
& + 1.125\cos 4\pi t + 0.825\operatorname{sen} 4\pi t \\
& - 0.3555\cos 6\pi t + 0.1906\operatorname{sen} 6\pi t \\
& - 0.2750\cos 8\pi t,
\end{aligned}
$$

de acuerdo con el ejemplo 10.3. ◄

10.3.2 Ajuste por mínimos cuadrados con funciones trigonométricas

El corolario 10.8 mostró la forma en la que la TDF facilita la interpolación de n puntos de datos uniformemente espaciados en $[0, 1]$ mediante una función trigonométrica de la forma

$$
P_n(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{n/2-1} (a_k \cos 2k\pi t - b_k \operatorname{sen} 2k\pi t) + \frac{a_{n/2}}{\sqrt{n}} \cos n\pi t. \tag{10.27}
$$

Observe que el número de términos n es igual al número de datos (como es habitual en este capítulo, se supone que n es par). Entre más datos se tengan, más cosenos y senos se añadirán para ayudar con la interpolación.

ANOTACIÓN **Ortogonalidad** En el capítulo 4 se establecieron las ecuaciones normales $A^T A\bar{x} = A^T b$ para resolver aproximaciones por mínimos cuadrados a datos, mediante funciones básicas. La intención del teorema 10.9 es encontrar casos especiales que hacen triviales a las ecuaciones normales, lo que simplifica en gran medida el procedimiento de mínimos cuadrados. Esto conduce a una teoría muy útil para las llamadas funciones ortogonales. Los principales ejemplos incluyen la transformada de Fourier en este capítulo y la transformada del coseno en el capítulo 11.

Como se encontró en el capítulo 3, cuando el número de datos n es grande, se vuelve menos común ajustar una función modelo de manera exacta. De hecho, una aplicación común de un modelo es olvidar algunos detalles (no representar algunos datos) con el fin de simplificar las cosas. Una segunda razón para alejarse de la interpolación, analizada en el capítulo 4, es el caso en el que se supone que los propios datos son inexactos, de modo que la aplicación estricta de una función de interpolación es inapropiada.

En cualquiera de estas situaciones, existe una motivación para hacer un ajuste por mínimos cuadrados con una función del tipo (10.27). Dado que los coeficientes a_k y b_k son lineales en el modelo, es posible continuar con el mismo programa descrito en el capítulo 4, usando las ecuaciones normales a fin de resolver para los mejores coeficientes. Al intentar esto, se encuentra un resultado sorprendente, que conduce de regreso a la TDF.

Considere de nuevo el teorema 10.9. Sea n el número de datos x_j, que por simplicidad se conciben uniformemente espaciados, $t_j = j/n$ en $[0, 1]$. Se introducirá el número par entero positivo m para indicar el número de funciones básicas que se utilizarán en el ajuste por mínimos cuadrados. Es decir, se ajustará con las primeras m funciones básicas, $f_0(t), \ldots, f_{m-1}(t)$. La función utilizada para ajustar los n puntos de datos será

$$P_m(t) = \sum_{k=0}^{m-1} c_k f_k(t), \tag{10.28}$$

donde las c_k deben determinarse. Cuando $m = n$, el problema es aún de interpolación. Cuando $m < n$, se reduce el problema. En este caso, se espera encontrar coincidencia con los datos mediante P_m con error mínimo cuadrático.

El problema de mínimos cuadrados es encontrar los coeficientes $c_0, \ldots, c_{m-1}$ de modo que la igualdad

$$\sum_{k=0}^{m-1} c_k f_k(t_j) = x_j$$

se cumpla con tan poco error como sea posible. En términos matriciales,

$$A_m^T c = x, \tag{10.29}$$

donde A_m es la matriz de los primeros m renglones de A. Bajo los supuestos del teorema 10.9, A_m^T tiene columnas ortonormales en pares. Cuando se establecen las ecuaciones normales

$$A_m A_m^T c = A_m x$$

para c, $A_m A_m^T$ es la matriz identidad. Por lo tanto, la solución por mínimos cuadrados,

$$c = A_m x, \tag{10.30}$$

puede calcularse con facilidad. Se ha demostrado el siguiente resultado útil, que extiende el teorema 10.9:

TEOREMA 10.11 **Teorema de la aproximación por mínimos cuadrados a una función ortogonal.** Sean $m \leq n$ números enteros, y suponga que se dan los datos $(t_0, x_0), \ldots, (t_{n-1}, x_{n-1})$. Establezca $y = Ax$, donde A es una matriz ortogonal de la forma (10.24). Entonces, el polinomio de interpolación para las funciones básicas $f_0(t), \ldots, f_{n-1}(t)$ es

$$F_n(t) = \sum_{k=0}^{n-1} y_k f_k(t), \tag{10.31}$$

y la mejor aproximación por mínimos cuadrados, utilizando sólo las funciones $f_0, \ldots, f_{m-1}$, es

$$F_m(t) = \sum_{k=0}^{m-1} y_k f_k(t). \tag{10.32}$$

■

Éste es un hecho hermoso y útil. Dice que, dados n puntos de datos, para encontrar la mejor función trigonométrica por mínimos cuadrados con $m < n$ términos que se ajuste a los datos, basta con calcular el interpolante real con n términos y mantener sólo los primeros m términos deseados. En otras palabras, los coeficientes de interpolación Ax para x se degradan con tanta gracia como sea posible al eliminar los términos con las frecuencias más altas. Al mantener los m términos más bajos en la expansión de n términos, se garantiza el mejor ajuste posible con los m términos de la frecuencia más baja. Esto refleja de manera apropiada la "ortogonalidad" de las funciones básicas.

El razonamiento anterior al teorema 10.11 se adapta con facilidad para demostrar algo más general. Se mostró cómo encontrar la solución por mínimos cuadrados para las primeras m funciones básicas, pero en realidad, el orden no fue relevante; podría haberse especificado cualquier subconjunto de las funciones básicas. La solución por mínimos cuadrados se encuentra tan sólo eliminando todos los términos en (10.31) que no estén incluidos en el subconjunto. La versión (10.32) es un filtro "pasa-bajas", asumiendo que las funciones con índices más bajos acompañen a las "frecuencias" menores; pero al cambiar el subconjunto de funciones básicas retenidas, se puede pasar cualquier frecuencia de interés con sólo eliminar los coeficientes no deseados.

De regreso al polinomio trigonométrico (10.27), se demostrará cómo ajustar una versión de orden m a n puntos, donde $m < n$. Las funciones básicas utilizadas son las funciones del ejemplo 10.4, que satisfacen los supuestos del teorema 10.9. El teorema 10.11 muestra que, cualesquiera que sean los coeficientes de interpolación, los coeficientes de la mejor aproximación por mínimos cuadrados de orden m se encuentran al eliminar todos los términos por encima del orden m. Se ha llegado a la siguiente aplicación:

COROLARIO 10.12 Sea $[c, d]$ un intervalo, para $m < n$ de números pares enteros positivos, $x = (x_0, \ldots, x_{n-1})$ un vector de n números reales y $t_j = c + j(d - c)/n$ para $j = 0, \ldots, n - 1$. Sean $\{a_0, a_1, b_1, a_2, b_2, \ldots, a_{n/2-1}, b_{n/2-1}, a_{n/2}\} = F_n x$ los coeficientes de interpolación para x de modo que

$$x_j = P_n(t_j) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{\frac{n}{2}-1} \left(a_k \cos \frac{2k\pi(t_j - c)}{d - c} - b_k \operatorname{sen} \frac{2k\pi(t_j - c)}{d - c} \right) + \frac{a_{\frac{n}{2}}}{\sqrt{n}} \cos \frac{n\pi(t_j - c)}{d - c}$$

para $j = 0, \ldots, n - 1$. Entonces

$$P_m(t) = \frac{a_0}{\sqrt{n}} + \frac{2}{\sqrt{n}} \sum_{k=1}^{\frac{m}{2}-1} \left(a_k \cos \frac{2k\pi(t - c)}{d - c} - b_k \operatorname{sen} \frac{2k\pi(t - c)}{d - c} \right) + \frac{2a_{\frac{m}{2}}}{\sqrt{n}} \cos \frac{n\pi(t - c)}{d - c}$$

es el mejor ajuste por mínimos cuadrados de orden m para los datos (t_j, x_j) para $j = 0, \ldots, n - 1$. ■

Otra forma de apreciar el poder del teorema 10.11 es compararlo con las funciones básicas monomiales que se han utilizado con anterioridad para modelos de mínimos cuadrados. La mejor parábola de mínimos cuadrados que se ajusta a los puntos (0, 3), (1, 3), (2, 5) es $y = x^2 - x + 3$. En otras palabras, los mejores coeficientes del modelo $y = a + bx + cx^2$ para estos datos son $a = 3, b = -1$ y $c = 1$ (en este caso porque el error cuadrático es igual a cero —ésta es la parábola de interpolación). Ahora se ajustará a un subconjunto de funciones básicas (por ejemplo, cambie el modelo a $y = a + bx$). Se calcula el mejor ajuste lineal con $a = 8/3$, $b = 1$. Observe que los coeficientes para el ajuste de primer grado no tienen relación aparente con sus correspondientes coeficientes para el ajuste de segundo grado. Esto es exactamente lo que *no* sucede para funciones básicas trigonométricas. Un ajuste de interpolación, o cualquier ajuste por mínimos cuadrados de la forma (10.28), contiene de manera explícita toda la información sobre los ajustes por mínimos cuadrados de orden inferior.

Debido a la respuesta tan sencilla que la TDF tiene para los mínimos cuadrados, resulta muy fácil escribir un programa de computadora para realizar los pasos. Sean $m < n < p$ números enteros, donde n es el conjunto de datos, m es el orden del modelo trigonométrico de mínimos cuadrados y p rige la resolución de la gráfica del mejor modelo. Puede pensarse en los mínimos cuadrados como el "filtrado" de las contribuciones con mayor frecuencia del interpolante de orden n, que deja sólo los m componentes de frecuencia más baja. Eso explica el nombre de la siguiente función de MATLAB:

```
% Programa 10.2 Ajuste trigonométrico por mínimos cuadrados
% Ajusta por mínimos cuadrados n puntos de datos en [0,1] con una función trigonométrica
%     donde 2 <=m <=n. Grafica el mejor ajuste en p (>=n) puntos.
% Entrada: intervalo [c, d], puntos de datos x, número par m,
%     número par de puntos de datos n, número par p>=n
% Salida: puntos filtrados xp
function xp=dftfilter(inter,x,m,n,p)
c=inter(1); d=inter(2);
t=c+(d-c)*(0:n-1)/n;          % puntos de tiempo para los datos (n)
tp=c+(d-c)*(0:p-1)/p          % puntos de tiempo para el interpolante (p)
y=fft(x);                     % calcular los coeficientes de interpolación
yp=zeros(p,1);                % contendrá los coeficientes para ifft
yp(1:m/2)=y(1:m/2);           % conserva sólo las primeras m frecuencias
yp(m/2+1)=real(y(m/2+1));     % como m es par, mantiene sólo el término cos
if(m<n)                       % al menos que en la frecuencia máxima,
  yp(p-m/2+1)=yp(m/2+1);      %    agregue un conjugado complejo al
end                           %    lugar correspondiente en el nivel superior
yp(p-m/2+2:p)=y(n-m/2+2:n);   % más conjugados para el nivel superior
xp=real(ifft(yp))*(p/n);      % invierte fft para recuperar datos
plot(t,x,'o',tp,xp)           % grafica los datos y la aprox por mínimos cuadrados
```

► EJEMPLO 10.6 Ajuste los datos de temperatura del ejemplo 10.3 mediante funciones trigonométricas de mínimos cuadrados con polinomios de cuarto y sexto orden.

El punto del corolario 10.12 es que sólo es posible interpolar los datos mediante la aplicación de F_n y después eliminar términos cuando sea necesario para conseguir mínimos cuadrados de orden inferior. El resultado del ejemplo 10.3 fue

$$\begin{aligned} P_8(t) = &-1.95 - 0.7445\cos 2\pi t - 2.5594\,\text{sen}\,2\pi t \\ &+ 1.125\cos 4\pi t + 0.825\,\text{sen}\,4\pi t \\ &- 0.3555\cos 6\pi t + 0.1906\,\text{sen}\,6\pi t \\ &- 0.2750\cos 8\pi t. \end{aligned} \tag{10.33}$$

Por lo tanto, los modelos de mínimos cuadrados para polinomios de cuarto y sexto orden son

$$P_4(t) = -1.95 - 0.7445\cos 2\pi t - 2.5594\operatorname{sen} 2\pi t + 1.125\cos 4\pi t$$
$$P_6(t) = -1.95 - 0.7445\cos 2\pi t - 2.5594\operatorname{sen} 2\pi t + 1.125\cos 4\pi t + 0.825\operatorname{sen} 4\pi t - 0.3555\cos 6\pi t.$$

En la figura 10.7 se muestran ambos ajustes por mínimos cuadrados, generados mediante

```
dftfilter([0,1],[-2.2,-2.8,-6.1,-3.9,0.0,1.1,-0.6,-1.1],m,8,200)
```

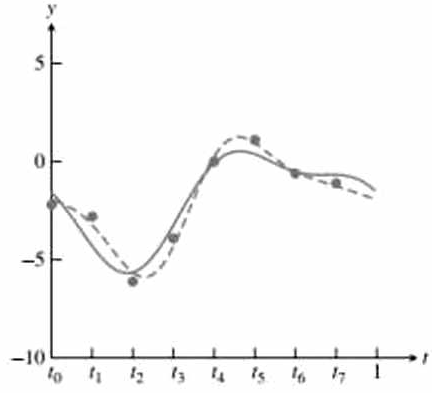


Figura 10.7 Ajustes trigonométricos por mínimos cuadrados para el ejemplo 10.6. Se muestran los ajustes para $m = 4$ (curva sólida) y 6 (curva discontinua). El vector de entrada x es $[-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, -1.1]^T$. El ajuste para $m = 8$ es la interpolación trigonométrica mostrada en la figura 10.6.

para $m = 4$ y 6, respectivamente. El ajuste $m = 4$ coincide con los ajustes explícitos por mínimos cuadrados a las funciones básicas 1, $\cos 2\pi t$, $\operatorname{sen} 2\pi t$, $\cos 4\pi t$ realizados en el ejemplo 4.6 y graficados en la figura 4.5(b). ◀

El programa `dftfilter.m` puede hacerse más eficaz. Calcula el interpolante de orden n y después descarte $n - m$ coeficientes. Por supuesto, un vistazo a la matriz de Fourier F_n muestra que si se desea conocer sólo los primeros m coeficientes de Fourier de los n puntos, es posible multiplicar x sólo por los m renglones superiores de la función F_n y dejarla así. En otras palabras, sería suficiente sustituir la matriz F_n de $n \times n$ por una submatriz de $m \times n$. Una versión mejorada de `dftfilter.m` haría uso de este hecho.

10.3.3 Sonido, ruido y filtrado

El código `dftfilter.m` de la última sección es una introducción a la vasta área del procesamiento de señales digitales. Se está utilizando la transformada de Fourier como una forma de transferir la información de una señal $\{x_0, \ldots, x_{n-1}\}$ en el "dominio del tiempo" al "dominio de la frecuencia", donde es más fácil de manipular. Al terminar de modificar lo que desea cambiarse, se envía la señal de regreso al dominio del tiempo mediante una TRF inversa.

Si x representa una señal de audio, lo anterior es útil debido a la manera en que está construido nuestro sistema auditivo. El oído humano contiene estructuras que responden a frecuencias, por lo que los cimientos en el dominio de la frecuencia son directamente significativos. Esto se ilustrará con la introducción de algunos conceptos básicos del procesamiento de audio y señales, y algunos comandos convenientes de MATLAB.

Una señal de audio se compone de un conjunto de números reales indexados según el tiempo. Cada número real representa una intensidad de sonido. Al reproducir una señal de audio, se hace vibrar el cabezal del altavoz de modo que coincida con la amplitud de la señal, lo que hace que el aire circundante vibre a las mismas frecuencias. Cuando las ondas sonoras alcanzan el oído, se percibe el sonido.

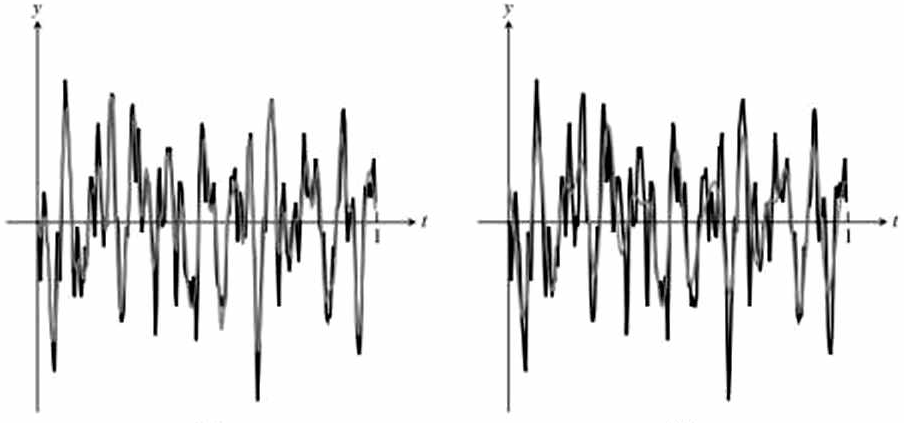


Figura 10.8 Curva de sonido junto con las versiones filtradas. Primer 1/32 de segundo del *Coro Aleluya* (256 puntos en la curva negra) junto con la versión filtrada (curva azul) usando (a) 64 funciones básicas, una razón de compresión de 4:1 y (b) 32 funciones básicas, una razón de compresión 8:1.

MATLAB proporciona una señal de audio de los primeros 9 segundos del *Coro Aleluya* de Handel como muestra. La curva de la figura 10.8 representa los primeros $2^8 = 256$ valores del archivo, que consiste en intensidades de sonido. La razón de muestreo de la música es $2^{13} = 8192$ Hz, lo que significa que las intensidades se representan a razón de 2^{13} por segundo, uniformemente espaciadas. Para acceder a la señal, pulse

```
>> load handel
```

lo cual coloca las variables `Fs` y `y` en el espacio de trabajo. La primera variable es la razón de muestreo $Fs = 8192$. La variable `y` es un vector con longitud de 73113 que contiene la señal de sonido. El comando de MATLAB

```
>> sound (y,Fs)
```

reproduce la señal en los altavoces de la computadora, si está disponible, en la razón de muestreo F^s correcta.

Los datos del *Coro Aleluya* pueden utilizarse para aplicar el filtrado del Corolario 10.12. Si se utiliza `dftfilter.m` con las primeras $n = 256$ muestras de la señal, y $m = 64$ y 32 funciones básicas, resultan las curvas azules de la figura 10.8. Es probable que el lector desee explorar el filtrado con otros archivos de audio.

Un formato común de archivo de audio es el formato `.wav`. Un archivo estéreo `.wav` lleva dos pares de señales que se reproducirán desde dos altavoces diferentes. Por ejemplo, si se usa el comando de MATLAB

```
>> [y,Fs] =wavread('castanets'),
```

se extraerá la señal estéreo del archivo `castanets.wav` y se cargará en MATLAB como una matriz y de $n \times 2$, donde cada columna es una señal de sonido independiente. (`castanets.wav` es un archivo común para probar el audio y puede encontrarse con facilidad mediante una búsqueda en la web). El comando `wavwrite` de MATLAB invierte el proceso, creando un archivo `.wav` de señales acústicas simples.

El filtrado se usa de dos maneras. Puede utilizarse para hacer coincidir, tanto como sea posible, la onda de sonido original con una función más simple. Ésta es una forma de reducción. En vez de

ANOTACIÓN

Resumen El filtrado es una forma de reducción con pérdida. En el caso de una señal de audio, el objetivo es reducir la cantidad de datos requeridos para almacenar o transmitir el sonido sin comprometer el efecto musical o la información hablada que la señal debe representar. Esto se realiza mejor en el dominio de la frecuencia, lo que implica la aplicación de la TDF, la manipulación de los componentes de la frecuencia y después la inversión de la TDF.

utilizar 256 números para almacenar la onda, podrían sólo almacenarse los m componentes de frecuencia más baja y después reconstruir la onda cuando sea necesario mediante el corolario 10.12. En la figura 10.8(a) se utilizaron $m = 64$ números reales en lugar de los originales 256, una relación de reducción de 4:1. Tenga en cuenta que es una reducción con pérdida, donde la onda original no se reproduce de manera exacta.

La segunda aplicación principal del filtrado es la eliminación del ruido. Dado un archivo de música donde la música o voz fue corrompida por un ruido de alta frecuencia (o silbido), la eliminación de las contribuciones con mayores frecuencias puede ser importante para mejorar el sonido. Por supuesto, los llamados filtros de pasa-bajas son martillos romos (una parte del sonido deseado con alta frecuencia, posiblemente en sobretonos que ni siquiera son evidentes para el oyente, pueden ser eliminados también. El tema del filtrado es parte de una vasta literatura sobre el procesamiento de señales, y se le pide al lector que consulte Oppenheim y Schafer [2009] para un estudio más detallado. En la comprobación en la realidad 10 se investiga un filtro de aplicación generalizada llamado filtro de Wiener.

10.3 Ejercicios

1. Encuentre la mejor aproximación por mínimos cuadrados de segundo orden para los datos del ejercicio 10.2.1, utilizando las funciones básicas 1 y $\cos 2\pi t$.
2. Encuentre la mejor aproximación por mínimos cuadrados de tercer orden para los datos del ejercicio 10.2.1, utilizando las funciones básicas 1, $\cos 2\pi t$ y $\operatorname{sen} 2\pi t$.
3. Encuentre la mejor aproximación por mínimos cuadrados de cuarto orden para los datos del ejercicio 10.2.3, utilizando las funciones básicas 1, $\cos 2\pi t$, $\operatorname{sen} 2\pi t$ y $\cos 4\pi t$.
4. Encuentre la mejor aproximación por mínimos cuadrados de cuarto orden para los datos del ejercicio 10.2.4, utilizando las funciones básicas 1, $\cos \frac{\pi}{4} t$, $\operatorname{sen} \frac{\pi}{4} t$ y $\cos \frac{\pi}{2} t$.
5. Demuestre el lema 10.10. (*Sugerencia*: exprese $\cos 2\pi jk/n$ como $(e^{i2\pi jk/n} + e^{-i2\pi jk/n})/2$, y escriba todo en términos de $\omega = e^{-i2\pi/n}$, de modo que pueda aplicarse el lema 10.1).

10.3 Problemas de computadora

1. Encuentre las funciones trigonométricas de aproximación por mínimos cuadrados, con polinomios de orden $m = 2$ y 4, para los siguientes puntos de datos:

(a)

t	y
0	3
$\frac{1}{4}$	0
$\frac{1}{2}$	−3
$\frac{3}{4}$	0

(b)

t	y
0	2
$\frac{1}{4}$	0
$\frac{1}{2}$	5
$\frac{3}{4}$	1

(c)

t	y
0	5
1	2
2	6
3	1

(d)

t	y
1	−1
2	1
3	4
4	3
5	3
6	2

Utilice `dftfilter.m` para graficar los datos y las funciones de aproximación, como en la figura 10.7.

1. Encuentre las funciones trigonométricas de aproximación por mínimos cuadrados, con órdenes de polinomios de 4, 6 y 8, para los siguientes datos:

(a)

t	y
0	3
$\frac{1}{8}$	0
$\frac{1}{4}$	−3
$\frac{3}{8}$	0
$\frac{1}{2}$	3
$\frac{5}{8}$	0
$\frac{3}{4}$	−6
$\frac{7}{8}$	0

(b)

t	y
0	1
$\frac{1}{8}$	0
$\frac{1}{4}$	−2
$\frac{3}{8}$	1
$\frac{1}{2}$	3
$\frac{5}{8}$	0
$\frac{3}{4}$	−2
$\frac{7}{8}$	1

(c)

t	y
0	1
$\frac{1}{8}$	2
$\frac{1}{4}$	3
$\frac{3}{8}$	1
$\frac{1}{2}$	−1
$\frac{5}{8}$	−1
$\frac{3}{4}$	−3
$\frac{7}{8}$	0

(d)

t	y
0	4.2
$\frac{1}{8}$	5.0
$\frac{1}{4}$	3.8
$\frac{3}{8}$	1.6
$\frac{1}{2}$	−2.0
$\frac{5}{8}$	−1.4
$\frac{3}{4}$	0.0
$\frac{7}{8}$	1.0

Grafique los datos y las funciones de aproximación, como en la figura 10.7.

3. Grafique la función trigonométrica de aproximación por mínimos cuadrados, con polinomios de orden $m = n/2$, $n/4$ y $n/8$, junto con el vector x que contiene los primeros 2^{14} valores de intensidad de sonido en el archivo de sonido handel de MATLAB. (Esto cubre alrededor de 2 segundos de audio. Puede utilizarse el código `dftfilter` de MATLAB con $p = n$. Haga tres gráficas independientes). Utilice el comando `sound` de MATLAB para comparar el original con la aproximación. ¿Cuánto se ha perdido?

4. Descargue `castanets.wav` desde un sitio web apropiado y forme un vector que contenga la señal en las primeras 2^{14} muestras de tiempo. Realice los pasos del problema de computadora 3 para cada canal estéreo de manera separada.

5. Reúna 24 lecturas horarias consecutivas de la temperatura en un periódico o página web. Represente los puntos de datos junto con (a) la función de interpolación trigonométrica y las funciones de aproximación por mínimos cuadrados de orden (b) $m = 6$ y (c) $m = 12$.

Comprobación en la realidad **10**

El filtro de Wiener

Sea c una señal de audio limpia y añada un vector r de la misma longitud a c. ¿La señal resultante $x = c + r$ es ruidosa? Si $r = c$, no se consideraría que r es ruido, puesto que el resultado sería una versión más fuerte, pero todavía limpia, de c. Por definición, el ruido no está correlacionado con la señal. En otras palabras, si r es el ruido, el valor esperado del producto interno $c^T r$ es cero. A continuación se aprovechará esta falta de correlación.

En una aplicación típica se presenta una señal x ruidosa y se le pide que encuentre c. La señal c puede ser el valor de una variable importante del sistema, que está siendo monitoreada en un ambiente ruidoso. O, como en el ejemplo siguiente, c podría ser una muestra de audio que es necesario extraer del ruido. A mediados del siglo XX, Norbert Wiener sugirió buscar el filtro óptimo para eliminar el ruido de x, en el sentido del error de mínimos cuadrados. Sugirió encontrar una matriz real, diagonal Φ de tal forma que la norma euclideana de

$$F^{-1}\,\Phi\,Fx - c$$

sea lo más pequeña posible, donde F indica la transformada discreta de Fourier. La idea es limpiar la señal x mediante la aplicación de la transformada de Fourier, operando sobre los componentes de la frecuencia por medio de la multiplicación por Φ, para después invertir la transformada de Fourier. Esto se denomina filtrado en el dominio de la frecuencia, puesto que se está cambiando la versión de x de la transformada de Fourier en vez de x en sí misma.

Para encontrar la mejor matriz diagonal Φ, tenga en cuenta que

$$\begin{aligned} ||F^{-1}\Phi Fx - c||_2 &= ||\Phi Fx - Fc||_2 \\ &= ||\Phi F(c + r) - Fc||_2 \\ &= ||(\Phi - I)C + \Phi R||_2, \end{aligned} \tag{10.34}$$

donde se establecen $C = Fc$ y $R = Fr$ como las transformadas de Fourier. Observe también que la definición de ruido implica

$$\overline{C}^T R = \overline{Fc}^T Fr = c^T \overline{F}^T Fr = c^T r = 0.$$

Se usará esto como motivación para hacer caso omiso de los términos cruzados en la norma, de modo que la magnitud al cuadrado se reduce a

$$\begin{aligned} \left(\overline{(\Phi - I)C + \Phi R}\right)^T ((\Phi - I)C + \Phi R) &= \left(\overline{C}^T (\Phi - I) + \overline{R}^T \Phi\right) ((\Phi - I)C + \Phi R) \\ &\approx \overline{C}^T (\Phi - I)^2 C + \overline{R}^T \Phi^2 R \\ &= \sum_{i=1}^{n} (\phi_i - 1)^2 |C_i|^2 + \phi_i^2 |R_i|^2. \end{aligned} \tag{10.35}$$

Para encontrar las entradas diagonales ϕ_i que disminuyen al mínimo esta expresión, diferencie por separado con respecto a cada ϕ_i para obtener

$$2(\phi_i - 1)|C_i|^2 + 2\phi_i |R_i|^2 = 0$$

para cada i, o bien, si se despeja ϕ_i,

$$\phi_i = \frac{|C_i|^2}{|C_i|^2 + |R_i|^2}. \tag{10.36}$$

Esta fórmula da los valores de Wiener para las entradas de la matriz diagonal Φ, para disminuir al mínimo la diferencia entre la versión filtrada $F^{-1}\,\Phi\, Fx$ y la señal limpia c. El único problema es que en los casos más típicos, no se conoce C o R, y es necesario hacer algunas aproximaciones para aplicar la fórmula.

Su trabajo consiste en investigar las formas de realizar una aproximación. Sea $X = Fx$ la transformada de Fourier. De nuevo, si se usa la no correlación de la señal y el ruido, se aproxima

$$|X_i|^2 \approx |C_i|^2 + |R_i|^2.$$

Entonces es posible escribir la opción óptima como

$$\phi_i \approx \frac{|X_i|^2 - |R_i|^2}{|X_i|^2} \tag{10.37}$$

y use el mejor conocimiento disponible del nivel de ruido. Por ejemplo, si el ruido es gaussiano no correlacionado (se modela al sumar un número aleatorio normal independiente para cada muestra de la señal limpia), es posible reemplazar $|R_i|^2$ en (10.37) por la constante $(p\sigma)^2$, donde σ es la desviación estándar del ruido y p es un parámetro cercano a uno que debe elegirse. Tenga en cuenta que

$$\sum_{i=1}^{n} |R_i|^2 = \overline{R}^T R = r\overline{F}^T Fr = r^T r = \sum_{i=1}^{n} r_i^2.$$

En el siguiente código se agrega 50 por ciento de ruido a la señal de Handel, y se usa $p = 1.3$ desviaciones estándar para aproximar R_i:

```
load handel                          % y es la señal limpia
c=y(1:40000);                        % funciona con las primeras 40K muestras
p=1.3;                               % parámetro de corte
noise=std(c)*.50;                    % 50 por ciento de ruido
n=length(c);                         % n es la longitud de la señal
r=noise*randn(n,1);                  % ruido puro
x=c+r;                               % señal ruidosa
fx=fft(x);sfx=conj(fx).*fx;          % obtiene fft de la señal, y
sfcapprox=max(sfx-n*(p*noise)^2,0); % aplica el corte
phi=sfcapprox./sfx;                  % define phi según su derivación
xout=real(ifft(phi.*fx));            % invierte la fft
%  después compare el sonido (x) el sonido (xout)
```

Actividades sugeridas:

1. Ejecute el código para formar la señal filtrada `yf`, y use el comando `sound` de MATLAB para comparar las señales de entrada y de salida.
2. Calcule el error cuadrático medio (ECM) de la entrada (`ys`) y de la salida (`yf`) comparándolas con la señal limpia (`yc`).
3. Encuentre el mejor valor del parámetro p para el 50 por ciento del ruido. Compare el valor que disminuye al mínimo el ECM con el que mejor se escuche.
4. Cambie el nivel de ruido a 10 por ciento, 25 por ciento, 100 por ciento, 200 por ciento y repita el paso 3. Resuma sus conclusiones.
5. Diseñe una comparación justa entre el filtro de Wiener y el filtro pasa-bajas descrito en la sección 10.2; realice la comparación.
6. Descargue un archivo `.wav` de su elección, añada ruido y realice los pasos indicados anteriormente.

Software y lecturas adicionales

Entre las mejores fuentes de lectura adicional sobre la transformada discreta de Fourier se encuentran Briggs [1995], Brigham [1988], y Briggs y Henson [1995]. El descubrimiento original de Cooley y Tukey apareció en Cooley y Tukey [1965], y su forma de calcular continúan siendo el punto central de la transformada rápida de Fourier en el procesamiento de señales moderno (Winograd [1978], Van Loan [1992] y Chu y George [1999]). La TRF es un algoritmo importante por derecho propio y, además, se utiliza como fundamento para otros algoritmos, debido a su aplicación eficaz. Por ejemplo, la utiliza MATLAB para calcular la transformada discreta del coseno, que se define en el capítulo 11. Curiosamente, la estrategia de divide y vencerás utilizada por Cooley y Tukey se aplicó con éxito a muchos otros problemas de cálculo posteriores.

El comando `fft` de MATLAB se basa en la "transformada de Fourier más rápida del oeste" (FFTW, por sus siglas en inglés), desarrollada en la década de 1990 en el MIT (Frigo y Johnson [1998]). En caso de que el tamaño n no sea una potencia de dos, el programa fracciona el problema, utilizando los factores primos de n, en "piezas de código" más pequeñas optimizadas para determinados tamaños fijos. Para obtener más información sobre la FFTW, incluyendo código descargable, consulte `http://www.fftw.org`. IMSL ofrece la transformada hacia adelante FFTCF y la transformada inversa FFTCB, basadas en FFTPACK de Netlib (Swarztrauber [1982]); éste es un paquete de subprogramas en Fortran para la transformada rápida de Fourier, los cuales están optimizados para su uso en implementaciones paralelas.

CAPÍTULO

11 Compresión

El movimiento cada vez mayor de la información en todo el mundo depende de ingeniosos métodos de representación de datos, que a su vez son posibles por las transformaciones ortogonales. El formato JPEG para la representación de imágenes se basa en la transformada discreta del coseno, la cual se desarrolla en este capítulo. Los formatos MPEG-1 y MPEG-2 para los datos de televisión y video, y el formato H.263 para los videoteléfonos también se basan en la TDC, pero con un énfasis adicional para la compresión en la dimensión de tiempo.

Los archivos de sonido pueden comprimirse en una variedad de formatos, incluyendo MP3, la codificación avanzada de audio (usada por los iTunes de Apple y los radios satelitales XM), Microsoft Windows Media Audio (WMA) y otros métodos modernos. Lo que estos formatos tienen en común es que la compresión principal se realiza mediante una variante de la TDC llamada la transformada discreta del coseno modificada.

Comprobación en la realidad En la página 527 se explora la aplicación de la TDCM en un algoritmo sencillo, que se utiliza para comprimir audio.

En los capítulos 4 y 10 se observó la utilidad de la ortogonalidad para representar y comprimir datos. Aquí se presenta la transformada discreta del coseno (TDC), una variante de la transformada de Fourier que puede calcularse en la aritmética real. En la actualidad, es el método más usado para la compresión de archivos de sonido e imagen.

La simplicidad de la transformada de Fourier se deriva de la ortogonalidad, debido a su representación como una matriz unitaria compleja. La transformada discreta del coseno tiene una representación como una matriz ortogonal real, por lo que las mismas propiedades de ortogonalidad la hacen sencilla de aplicar y fácil de invertir. Su similitud con la transformada discreta de Fourier (TDF) es lo suficientemente cercana para que existan versiones rápidas de la TDC, en analogía con la transformada rápida de Fourier (TRF).

En este capítulo se explican las propiedades básicas de la TDC y se investigan sus vínculos con los formatos de compresión elementales. Por ejemplo, en el conocido formato JPEG se aplica la TDC bidimensional a bloques de píxeles de 8×8 en una imagen, y se almacenan los resultados utilizando la codificación de Huffman. Los detalles de la compresión JPEG se investigan como un estudio de caso en las secciones 11.2 y 11.3.

Una versión modificada de la transformada discreta del coseno, llamada la transformada discreta del coseno modificada (TDCM), es la base de los formatos de compresión de audio más modernos. La TDCM es la norma de oro actual para la compresión de archivos de sonido. Se presentará la TDCM y se estudiará su aplicación para la codificación y decodificación, que proporcionan la tecnología primordial de formatos de archivo como MP3 y AAC (codificación avanzada de audio).

11.1 LA TRANSFORMADA DISCRETA DEL COSENO

En esta sección se presenta la transformada discreta del coseno. Esta transformación interpola los datos utilizando funciones básicas que son todas funciones coseno e implica sólo cálculos reales. Sus características de ortogonalidad simplifican las aproximaciones por mínimos cuadrados, como en el caso de la transformada discreta de Fourier.

11.1.1 TDC unidimensional

Sea n un entero positivo. La transformada discreta del coseno unidimensional de orden n se define mediante la matriz C de $n \times n$ cuyas entradas son

$$C_{ij} = \frac{\sqrt{2}}{\sqrt{n}} a_i \cos \frac{i(2j+1)\pi}{2n} \tag{11.1}$$

para $i, j = 0, \ldots, n-1$, donde

$$a_i \equiv \begin{cases} 1/\sqrt{2} & \text{si } i = 0, \\ 1 & \text{si } i = 1, \ldots, n-1 \end{cases}$$

o

$$C = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{2n} & \cos\frac{3\pi}{2n} & \cdots & \cos\frac{(2n-1)\pi}{2n} \\ \cos\frac{2\pi}{2n} & \cos\frac{6\pi}{2n} & \cdots & \cos\frac{2(2n-1)\pi}{2n} \\ \vdots & \vdots & & \vdots \\ \cos\frac{(n-1)\pi}{2n} & \cos\frac{(n-1)3\pi}{2n} & \cdots & \cos\frac{(n-1)(2n-1)\pi}{2n} \end{bmatrix}. \tag{11.2}$$

Con imágenes bidimensionales la convención es comenzar con 0 en lugar de 1. La notación será más fácil si se extiende esta convención a la numeración de la matriz, como se hizo en (11.1). *En este capítulo los subíndices de las matrices de $n \times n$ irán de* 0 a $n-1$. Por simplicidad, en el siguiente análisis se tratará sólo el caso en el que n es par.

DEFINICIÓN 11.1 Sea C la matriz definida en (11.2). La **transformada discreta del coseno** (TDC) de $x = [x_0, \ldots, x_{n-1}]^T$ es el vector n-dimensional $y = [y_0, \ldots, y_{n-1}]^T$, donde

$$y = Cx. \tag{11.3}$$

❐

Observe que C es una matriz ortogonal real, lo que significa que su transpuesta es su inverso:

$$C^{-1} = C^T = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \cos\frac{\pi}{2n} & \cdots & \cos\frac{(n-1)\pi}{2n} \\ \frac{1}{\sqrt{2}} & \cos\frac{3\pi}{2n} & \cdots & \cos\frac{(n-1)3\pi}{2n} \\ \vdots & \vdots & & \vdots \\ \frac{1}{\sqrt{2}} & \cos\frac{(2n-1)\pi}{2n} & \cdots & \cos\frac{(n-1)(2n-1)\pi}{2n} \end{bmatrix}. \tag{11.4}$$

Los renglones de una matriz ortogonal son vectores unitarios ortogonales por pares. La ortogonalidad de C se sigue del hecho de que las columnas de C^T son los vectores propios unitarios de la matriz real simétrica de $n \times n$

$$\begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 1 & \end{bmatrix}. \tag{11.5}$$

El ejercicio 6 le pide al lector que verifique este hecho.

El hecho de que C sea una matriz ortogonal real hace que la TDC resulte útil. El teorema de la función de interpolación ortogonal 10.9 aplicada a la matriz C implica el teorema 11.2.

TEOREMA 11.2 **Teorema de interpolación de la TDC.** Sea $x = [x_0, \ldots, x_{n-1}]^T$ un vector de n números reales. Defina $y = [y_0, \ldots, y_{n-1}]^T = Cx$, donde C es la matriz de la transformada discreta del coseno con orden n. Entonces la función real

$$P_n(t) = \frac{1}{\sqrt{n}} y_0 + \frac{\sqrt{2}}{\sqrt{n}} \sum_{k=1}^{n-1} y_k \cos \frac{k(2t+1)\pi}{2n}$$

satisface $P_n(j) = x_j$ para $j = 0, \ldots, n-1$. ■

Demostración. Aplicando directamente el teorema 10.9. ❐

El teorema 11.2 muestra que la matriz C de $n \times n$ transforma en los n puntos en n coeficientes de interpolación. Al igual que la transformada discreta de Fourier, la transformada discreta del coseno proporciona los coeficientes de una función de interpolación trigonométrica. A diferencia de la TDF, la TDC sólo utiliza términos de coseno y se define por completo en términos de aritmética real.

► EJEMPLO 11.1 Utilice la TDC para interpolar los puntos $(0, 1), (1, 0), (2, -1), (3, 0)$.

Es importante destacar, utilizando la trigonometría elemental, que la matriz de la TDC de 4×4 puede verse como

$$C = \frac{1}{\sqrt{2}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{8} & \cos\frac{3\pi}{8} & \cos\frac{5\pi}{8} & \cos\frac{7\pi}{8} \\ \cos\frac{2\pi}{8} & \cos\frac{6\pi}{8} & \cos\frac{10\pi}{8} & \cos\frac{14\pi}{8} \\ \cos\frac{3\pi}{8} & \cos\frac{9\pi}{8} & \cos\frac{15\pi}{8} & \cos\frac{21\pi}{8} \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}, \tag{11.6}$$

donde

$$a = \frac{1}{2}, b = \frac{1}{\sqrt{2}} \cos\frac{\pi}{8} = \frac{\sqrt{2+\sqrt{2}}}{2\sqrt{2}}, c = \frac{1}{\sqrt{2}} \cos\frac{3\pi}{8} = \frac{\sqrt{2-\sqrt{2}}}{2\sqrt{2}}. \tag{11.7}$$

La TDC de cuarto orden multiplicada por los datos $x = (1, 0, -1, 0)^T$ es

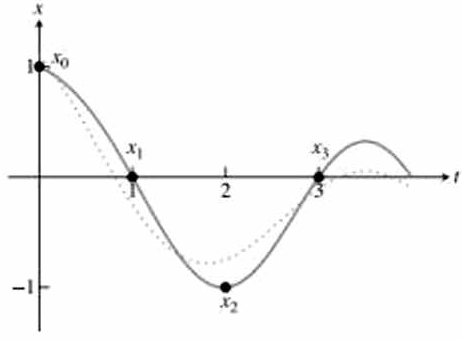


Figura 11.1 Interpolación de la TDC y aproximación por mínimos cuadrados. Los puntos de datos son (j, x_j), donde $x = [1, 0, -1, 0]$. La función de interpolación de la TDC $P_4(t)$ de (11.8) se muestra como una curva sólida (continua). También se muestra como una curva punteada, la función de aproximación por mínimos cuadrados de la TDC $P_3(t)$ de (11.9).

$$\begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ c+b \\ 2a \\ c-b \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\sqrt{2-\sqrt{2}}+\sqrt{2+\sqrt{2}}}{2\sqrt{2}} \\ 1 \\ \frac{\sqrt{2-\sqrt{2}}-\sqrt{2+\sqrt{2}}}{2\sqrt{2}} \end{bmatrix} \approx \begin{bmatrix} 0.0000 \\ 0.9239 \\ 1.0000 \\ -0.3827 \end{bmatrix}.$$

De acuerdo con el teorema 11.2 con $n = 4$, la función

$$P_4(t) = \frac{1}{\sqrt{2}}\left[0.9239\cos\frac{(2t+1)\pi}{8} + \cos\frac{2(2t+1)\pi}{8} - 0.3827\cos\frac{3(2t+1)\pi}{8}\right] \quad (11.8)$$

interpola los cuatro puntos. La función $P_4(t)$ se representa como la curva sólida (continua) de la figura 11.1. ◀

11.1.2 La TDC y la aproximación por mínimos cuadrados

Así como el teorema de interpolación de la TDC 11.2 es una consecuencia inmediata del teorema 10.9, el teorema resultante de mínimos cuadrados 10.11 muestra cómo encontrar una aproximación de los datos con la TDC por mínimos cuadrados, utilizando sólo una parte de las funciones básicas. Debido a la ortogonalidad de las funciones básicas, esto se puede lograr tan sólo descartando los términos con frecuencia mayor.

ANOTACIÓN

Ortogonalidad La idea detrás de la aproximación por mínimos cuadrados es que encontrar la distancia más corta desde un punto hasta un plano (o subespacio en general) implica construir la perpendicular desde el punto hasta el plano. Esta construcción se realiza mediante las ecuaciones normales, como se vio en el capítulo 4. En los capítulos 10 y 11 este concepto se aplica a los datos aproximados tanto como sea posible con un conjunto relativamente pequeño de funciones básicas, lo que resulta en una compresión. El mensaje principal es elegir las funciones básicas de modo que sean ortogonales, como se refleja en los renglones de la matriz de la TDC. Entonces, las ecuaciones normales se vuelven muy simples hablando en términos de cálculo (vea el teorema 10.11).

TEOREMA 11.3 **Teorema de aproximación de la TDC por mínimos cuadrados.** Sea $x = [x_0, \ldots, x_{n-1}]^T$ un vector de n números reales. Defina $y = [y_0, \ldots, y_{n-1}]^T = Cx$, donde C es la matriz de la transformada discreta del coseno. Entonces, para cualquier entero positivo $m \leq n$, la elección de los coeficientes $y_0, \ldots, y_{m-1}$ en

$$P_m(t) = \frac{1}{\sqrt{n}} y_0 + \frac{\sqrt{2}}{\sqrt{n}} \sum_{k=1}^{m-1} y_k \cos \frac{k(2t+1)\pi}{2n}$$

minimizando el error cuadrático de la aproximación $\sum_{j=0}^{n-1} (P_m(j) - x_j)^2$ sobre los n puntos de datos. ■

Demostración. Se desprende directamente del teorema 10.11. ❐

En referencia al ejemplo 11.1, si se requiere la mejor aproximación por mínimos cuadrados a los mismos cuatro puntos, pero usando sólo las tres funciones básicas

$$1, \cos \frac{(2t+1)\pi}{8}, \cos \frac{2(2t+1)\pi}{8}$$

la solución es

$$P_3(t) = \frac{1}{2} \cdot 0 + \frac{1}{\sqrt{2}} \left[0.9239 \cos \frac{(2t+1)\pi}{8} + \cos \frac{2(2t+1)\pi}{8} \right]. \tag{11.9}$$

En la figura 11.1 se compara la solución por mínimos cuadrados P_3 con la función de interpolación P_4.

► EJEMPLO 11.2 Utilice la TDC y el teorema 11.3 para encontrar los ajustes por mínimos cuadrados a los puntos $t = 0, \ldots, 7$ y $x = [-2.2, -2.8, -6.1, -3.9, 0.0, 1.1, -0.6, -1.1]^T$ para $m = 4, 6$ y 8.

Si se establece $n = 8$, se encuentra que la TDC de los datos es

$$y = Cx = \begin{bmatrix} -5.5154 \\ -3.8345 \\ 0.5833 \\ 4.3715 \\ 0.4243 \\ -1.5504 \\ -0.6243 \\ -0.5769 \end{bmatrix}.$$

De acuerdo con el teorema 11.2, el interpolante discreto del coseno para los ocho puntos de datos es

$$\begin{aligned} P_8(t) = \frac{1}{\sqrt{8}}(-5.5154) + \frac{1}{2}\Bigg[& -3.8345 \cos \frac{(2t+1)\pi}{16} + 0.5833 \cos \frac{2(2t+1)\pi}{16} \\ & + 4.3715 \cos \frac{3(2t+1)\pi}{16} + 0.4243 \cos \frac{4(2t+1)\pi}{16} \\ & - 1.5504 \cos \frac{5(2t+1)\pi}{16} - 0.6243 \cos \frac{6(2t+1)\pi}{16} \\ & - 0.5769 \cos \frac{7(2t+1)\pi}{16} \Bigg]. \end{aligned}$$

El interpolante P_8 se grafica en la figura 11.2, junto con los ajustes por mínimos cuadrados P_6 y P_4. Éstos se obtienen, de acuerdo con el teorema 11.3, conservando, respectivamente, los primeros seis o cuatro términos de P_8. ◄

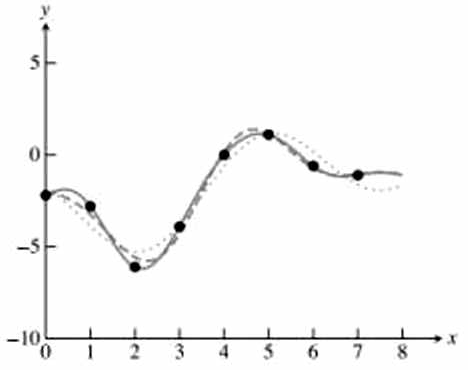


Figura 11.2 Interpolación de la TDC y aproximación por mínimos cuadrados. La curva continua (sólida) es el interpolante de la TDC para los puntos del ejemplo 11.2. La curva discontinua (punteada) es el ajuste por mínimos cuadrados sólo de los primeros seis términos, y la curva punteada representa el ajuste de los primeros cuatro términos.

11.1 Ejercicios

1. Use la matriz de 2×2 de la TDC y el teorema 11.2 para encontrar la función de interpolación de la TDC para los puntos dados.

(a)

t	x
0	3
1	3

(b)

t	x
0	2
1	−2

(c)

t	x
0	3
1	1

(d)

t	x
0	4
1	−1

2. Describa la aproximación de la TDC por mínimos cuadrados con $m = 1$ en términos de los datos de entrada $(0, x_0), (1, x_1)$.

3. Encuentre la TDC de los siguientes vectores de datos x, y encuentre la correspondiente función interpolante $P_n(t)$ para los puntos de datos (i, x_i), $i = 0, \ldots, n-1$ (puede indicar sus respuestas en términos de b y c definidas en (11.7)):

(a)

t	x
0	1
1	0
2	1
3	0

(b)

t	x
0	1
1	1
2	1
3	1

(c)

t	x
0	1
1	0
2	0
3	0

(d)

t	x
0	1
1	2
2	3
3	4

4. Encuentre la aproximación de la TDC por mínimos cuadrados con $m = 2$ términos para los datos del ejercicio 3.

5. Realice las operaciones trigonométricas necesarias para establecer las ecuaciones (11.6) y (11.7).

6. (a) Demuestre la fórmula trigonométrica $\cos(x + y) + \cos(x - y) = 2 \cos x \cos y$ para cualesquiera x, y. (b) Demuestre que las columnas de C^T son vectores propios de la matriz T en (11.5), e identifique los valores propios. (c) Demuestre que las columnas de C^T son vectores unitarios.

7. Extienda el teorema 11.2 de la interpolación de la TDC al intervalo $[c, d]$ de la siguiente manera. Sea n un entero positivo y establezca $\Delta_t = (d - c)/n$. Utilice la TDC para producir un polinomio $P_n(t)$ que satisfaga $P_n(c + j\Delta_t) = x_j$ para $j = 0, \ldots, n-1$.

11.1 Problemas de computadora

1. Grafique los datos del ejercicio 3, junto con el interpolante de la TDC y la aproximación de la TDC por mínimos cuadrados con $m = 2$ términos.

2. Grafique los datos junto con las aproximaciones de la TDC por mínimos cuadrados con $m = 4, 6$ y 8.

(a)

t	x
0	3
1	5
2	−1
3	3
4	1
5	3
6	−2
7	4

(b)

t	x
0	4
1	1
2	−3
3	0
4	0
5	2
6	−4
7	0

(c)

t	x
0	3
1	−1
2	−1
3	3
4	3
5	−1
6	−1
7	3

(d)

t	x
0	4
1	2
2	−4
3	2
4	4
5	2
6	−4
7	2

3. Grafique la función $f(t)$, los puntos $(j, f(j)), j = 0, \ldots, 7$, y la función de interpolación de la TDC. (a) $f(t) = e^{-t/4}$ (b) $f(t) = \cos \frac{\pi}{2} t$.

11.2 TDC BIDIMENSIONAL Y COMPRESIÓN DE IMÁGENES

La transformada discreta del coseno bidimensional se utiliza a menudo para comprimir pequeños bloques de una imagen, hasta de 8×8 pixeles. La compresión es con pérdida, lo que significa que parte de la información del bloque se descarta. La característica clave de la TDC es que ayuda a organizar la información de modo que la porción que no se toma en cuenta es la parte a la que el ojo humano es menos sensible. De manera más precisa, la TDC mostrará cómo interpolar los datos con un conjunto de funciones básicas que guardan un orden de importancia descendente, en lo que al sistema visual humano se refiere. Los términos de menor importancia en la interpolación pueden eliminarse si así se desea, de la misma forma que un editor de periódico reduce una historia larga en una fecha límite.

Posteriormente se aplicará lo aprendido acerca de la TDC en la compresión de imágenes. Usando las herramientas adicionales de cuantificación y codificación de Huffman, cada bloque de la imagen de 8×8 puede reducirse a un flujo de bits que se almacena con los flujos de bits de los otros bloques de la imagen. El flujo completo de bits se decodifica, cuando la imagen debe ser descomprimida y mostrada, mediante la inversión del proceso de codificación. Se describirá este método, llamado línea de base JPEG, el método predeterminado para el almacenamiento de imágenes JPEG.

11.2.1 TDC bidimensional

La transformada discreta del coseno bidimensional es tan sólo la TDC unidimensional aplicada en dos dimensiones, una después de la otra. Se puede utilizar para interpolar o aproximar datos dados en una malla bidimensional, en una analogía directa con el caso unidimensional. En el contexto del procesamiento de imágenes, la malla bidimensional representa un bloque de valores de píxel (por ejemplo, intensidades de la escala de grises o intensidades del color).

Sólo en este capítulo se indicará primero la coordenada vertical y después la coordenada horizontal cuando se haga referencia a un punto bidimensional, como se muestra en la figura 11.3. El objetivo es ser consistente con la convención usual de las matrices, donde el índice i de la entrada x_{ij} cambia a lo largo de la dirección vertical, y j a lo largo de la dirección horizontal. Una aplicación importante de esta sección es a los archivos de pixeles que representan imágenes, los cuales pueden verse de manera natural como matrices de números.

En la figura 11.3 se muestra una malla de puntos (s, t) en el plano bidimensional, con valores x_{ij} asignados a cada punto de la malla rectangular (s_i, t_j). En concreto, se utilizará toda la malla $s_i = \{0, 1, \ldots, n-1\}$ (recuerde, a lo largo del eje vertical) y $t_j = \{0, 1, \ldots, n-1)$ a lo largo del eje horizontal. El propósito de la TDC bidimensional es construir una función de interpolación $F(s, t)$ que se ajuste a los n^2 puntos (s_i, t_j, x_{ij}) para $i, j = 0, \ldots, n-1$. La función TDC-2D logra esto de una manera óptima desde el punto de vista de los mínimos cuadrados, lo que significa que el ajuste se degrada adecuadamente a medida que las funciones básicas se eliminan de la función de interpolación.

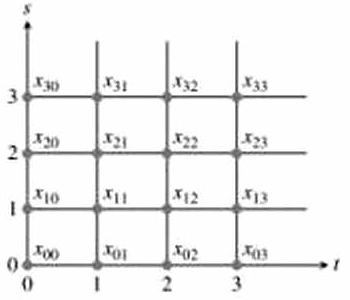


Figura 11.3 Malla bidimensional de puntos. La TDC-2D puede usarse para interpolar los valores de una función en una malla, como los valores de los pixeles de una imagen.

La TDC-2D es la TDC unidimensional aplicada sucesivamente en las direcciones horizontal y vertical. Considere la matriz X que consiste en los valores x_{ij} como en la figura 11.3. Para aplicar la TDC-1D en la dirección horizontal s, primero es necesario transponer X y después multiplicar por C. Las columnas resultantes son las TDC-1D de los renglones de X. Cada columna de CX^T corresponde a una t_i fija. Realizar una TDC-1D en el la dirección t significa moverse a través de los renglones; de modo que, una vez más, al transponer y multiplicar por C se obtiene

$$C(CX^T)^T = CXC^T. \tag{11.10}$$

DEFINICIÓN 11.4 La **transformada discreta del coseno bidimensional** (TDC-2D) de la matriz X de $n \times n$ es la matriz $Y = CXC^T$, donde C se define como en (11.1). ❐

► EJEMPLO 11.3 Encuentre la transformada discreta del coseno 2D para los datos de la figura 11.4(a).

A partir de la definición y de (11.6), la TDC-2D es la matriz

$$Y = CXC^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11.11}$$

◄

La inversa de la TDC-2D se expresa con facilidad en términos de la matriz C de la TDC. Como $Y = CXC^T$ y C es ortogonal, la X se recupera como $X = C^T YC$.

DEFINICIÓN 11.5 La **inversa de la transformada discreta del coseno** bidimensional de la matriz Y de $n \times n$ es la matriz $X = C^T YC$. ❐

Como se ha visto, existe una estrecha relación entre la inversión de una transformada ortogonal (como la TDC-2D) y la interpolación. El objetivo de la interpolación es recuperar los puntos de datos originales a partir de las funciones construidas con los coeficientes de interpolación que surgieron de la transformada. Como C es una matriz ortogonal, $C^{-1} = C^T$. La inversión de la TDC-2D puede escribirse como un hecho sobre la interpolación, $X = C^T YC$, puesto que en esta ecuación las x_{ij} se expresan en términos de productos de cosenos.

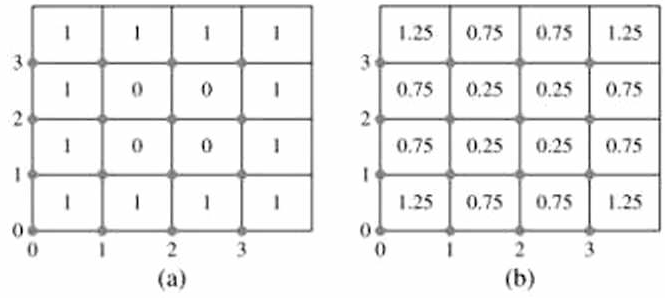


Figura 11.4 Datos bidimensionales para el ejemplo 11.3. (a) Los 16 puntos de datos (i, j, x_{ij}). (b) Valores de la aproximación por mínimos cuadrados (11.4) en los puntos de la malla.

A fin de escribir una expresión útil para la función de interpolación, recuerde la definición de C en (11.1),

$$C_{ij} = \frac{\sqrt{2}}{\sqrt{n}} a_i \cos \frac{i(2j+1)\pi}{2n} \tag{11.12}$$

para $i, j = 0, \ldots, n-1$, donde

$$a_i \equiv \begin{cases} 1/\sqrt{2} & \text{si } i = 0, \\ 1 & \text{si } i = 1, \ldots, n-1 \end{cases}.$$

De acuerdo con las reglas de la multiplicación de matrices, la ecuación $X = C^T YC$ se traduce como

$$\begin{aligned} x_{ij} &= \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} C_{ik}^T y_{kl} C_{lj} \\ &= \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} C_{ki} y_{kl} C_{lj} \\ &= \frac{2}{n} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} y_{kl} a_k a_l \cos \frac{k(2i+1)\pi}{2n} \cos \frac{l(2j+1)\pi}{2n}. \end{aligned} \tag{11.13}$$

Éste es exactamente el enunciado de interpolación que se buscaba.

TEOREMA 11.6 **Teorema de interpolación de la TDC-2D.** Sea $X = (x_{ij})$ una matriz de n^2 números reales. Sea $Y = (y_{kl})$ la transformada discreta del coseno bidimensional de X. Defina $a_0 = 1/\sqrt{2}$ y $a_k = 1$ para $k > 0$. Entonces la función real

$$P_n(s,t) = \frac{2}{n} \sum_{k=0}^{n-1} \sum_{l=0}^{n-1} y_{kl} a_k a_l \cos \frac{k(2s+1)\pi}{2n} \cos \frac{l(2t+1)\pi}{2n}$$

satisface $P_n(i, j) = x_{ij}$ para $i, j = 0, \ldots, n-1$. ■

De regreso al ejemplo 11.3, los únicos coeficientes de interpolación distintos de cero son $y_{00} = 3$, $y_{02} = y_{20} = 1$, y $y_{22} = -1$. Si se escribe la función de interpolación del teorema 11.6, resulta

$$
\begin{aligned}
P_4(s,t) &= \frac{2}{4}\left[\frac{1}{2}y_{00} + \frac{1}{\sqrt{2}}y_{02}\cos\frac{2(2t+1)\pi}{8} + \frac{1}{\sqrt{2}}y_{20}\cos\frac{2(2s+1)\pi}{8}\right.\\
&\quad \left. + y_{22}\cos\frac{2(2s+1)\pi}{8}\cos\frac{2(2t+1)\pi}{8}\right]\\
&= \frac{1}{2}\left[\frac{1}{2}(3) + \frac{1}{\sqrt{2}}(1)\cos\frac{2(2t+1)\pi}{8} + \frac{1}{\sqrt{2}}(1)\cos\frac{2(2s+1)\pi}{8}\right.\\
&\quad \left. + (-1)\cos\frac{2(2s+1)\pi}{8}\cos\frac{2(2t+1)\pi}{8}\right]\\
&= \frac{3}{4} + \frac{1}{2\sqrt{2}}\cos\frac{(2t+1)\pi}{4} + \frac{1}{2\sqrt{2}}\cos\frac{(2s+1)\pi}{4}\\
&\quad - \frac{1}{2}\cos\frac{(2s+1)\pi}{4}\cos\frac{(2t+1)\pi}{4}.
\end{aligned}
$$

Al comprobar la interpolación, se obtiene, por ejemplo,

$$P_4(0,0) = \frac{3}{4} + \frac{1}{4} + \frac{1}{4} - \frac{1}{4} = 1$$

y

$$P_4(1,2) = \frac{3}{4} - \frac{1}{4} - \frac{1}{4} - \frac{1}{4} = 0,$$

de acuerdo con los datos de la figura 11.4. El término constante y_{00}/n de la función de interpolación se llama componente de expansión "DC" (para "corriente directa"). Es simplemente el promedio de los datos; los términos no constantes contienen las fluctuaciones de los datos sobre dicho valor promedio. En este ejemplo, el promedio de los 12 unos y 4 ceros es $y_{00}/4 = 3/4$.

Las aproximaciones por mínimos cuadrados con la TDC-2D se realizan de la misma manera que con la TDC-1D. Por ejemplo, la aplicación de un filtro pasa-bajas significaría sólo borrar los componentes de "alta frecuencia", aquellos cuyos coeficientes tengan mayores índices en la función de interpolación. En el ejemplo 11.3 el mejor ajuste por mínimos cuadrados a las funciones básicas

$$\cos\frac{i(2s+1)\pi}{8}\cos\frac{j(2t+1)\pi}{8}$$

para $i + j \leq 2$ se da al descartar todos los términos que no satisfagan $i + j \leq 2$. En este caso, el único término de "frecuencia alta" distinto de cero es el término $i = j = 2$, lo que deja

$$P_2(s,t) = \frac{3}{4} + \frac{1}{2\sqrt{2}}\cos\frac{(2t+1)\pi}{4} + \frac{1}{2\sqrt{2}}\cos\frac{(2s+1)\pi}{4}. \tag{11.14}$$

Esta aproximación por mínimos cuadrados se muestra en la figura 11.4(b).

La definición de la matriz C de la TDC en MATLAB puede hacerse a través del fragmento de código

```
for i=1:n
  for j=1:n
    C(i,j)=cos((i-1)*(2*j-1)*pi/(2*n));
  end
end
C=sqrt(2/n)*C;
C(1,:)=C(1,:)/sqrt(2);
```

De manera alternativa, si se dispone de la caja de herramientas para el procesamiento de señales en MATLAB, la TDC unidimensional de un vector x puede calcularse como

```
>> y=dct(x);
```

Para llevar a cabo la TDC-2D de una matriz X, se recurre de nuevo a la ecuación (11.10), o

```
>> Y=C*X*C'
```

Si se dispone de `dct` en MATLAB, el comando

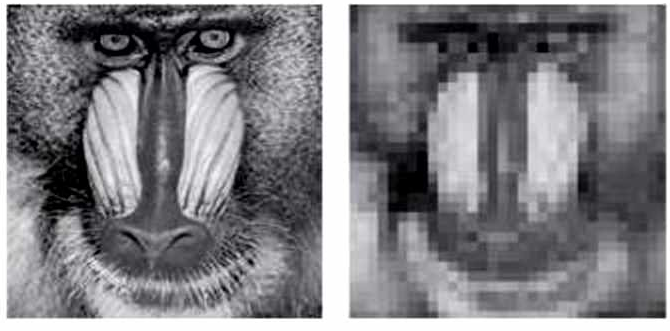
```
>> Y=dct(dct(X')')
```

calcula la TDC-2D mediante dos aplicaciones de la TDC-1D.

11.2.2 Compresión de imágenes

El concepto de ortogonalidad, según se representa en la transformada discreta del coseno, es crucial para realizar la compresión de imágenes. Las imágenes se componen de pixeles, cada uno representado por un número (o tres números, para las imágenes en color). La manera conveniente en que los métodos como la TDC pueden realizar aproximaciones por mínimos cuadrados facilita la reducción del número de bits necesarios para representar los valores de los pixeles, mientras que la imagen sólo se degrada ligeramente, y quizá de manera imperceptible a la vista de los seres humanos.

En la figura 11.5(a) se muestra una representación en escala de grises de un arreglo de pixeles de 256×256. Los tonos de gris de cada píxel está representado por un byte, una cadena de 8 bits que representan desde $0 = 00000000$ (negro) hasta $255 = 11111111$ (blanco). Puede pensarse en la información que se muestra en la figura como una matriz de enteros de 256×256. Representada de esta manera, la imagen mantiene $(256)^2 = 2^{16} = 64$K bytes de información.



(a) (b)

Figura 11.5 Imagen en escala de grises. (a) Cada píxel en la cuadrícula de 256×256 se representa mediante un número entero comprendido entre 0 y 255. (b) Compresión burda (cada cuadro de pixeles de 8×8 se colorea de acuerdo con su valor promedio en la escala de grises).

MATLAB importa valores en escala de grises o en RGB (rojo-verde-azul) de las imágenes en los formatos de imagen estándar. Por ejemplo, dado un archivo de imagen en escala de grises `picture.jpg`, el comando

```
>> x = imread('picture.jpg');
```

pone la matriz de valores en escala de grises en la variable `x` de doble precisión. Si el archivo JPEG es una imagen en color, la variable del arreglo tendrá una tercera dimensión para indexar los tres colores. Aquí se restringirá la atención a la escala de grises como un inicio del análisis: la extensión al color es directa.

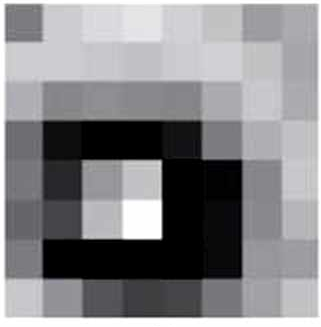
Una matriz de $m \times n$ valores en la escala de grises puede representarse en MATLAB con los comandos

```
>> imagesc(x); colormap(gray)
```

mientras que una matriz RGB a colores de $m \times n \times 3$ se representa sólo con el comando `imagesc(x)`. Una fórmula común para convertir una imagen a colores RGB en escala es grises es

$$X_{\text{gris}} = 0.2126R + 0.7152G + 0.0722B, \tag{11.15}$$

o en código MATLAB,



(a)

110	168	176	182	170	159	134	145
166	168	164	161	165	171	159	141
146	116	124	122	119	145	162	144
102	34	22	25	38	111	146	159
107	49	130	159	2	29	117	164
95	71	153	207	15	30	122	150
112	21	0	19	0	30	132	135
163	129	83	67	69	107	139	159

(b)

−18	40	48	54	42	31	6	17
38	40	36	33	37	43	31	13
18	−10	−4	−6	−9	17	34	16
−26	−94	−106	−103	−90	−17	18	31
−21	−79	2	31	−126	−99	−11	36
−33	−57	25	79	−113	−98	−6	22
−16	107	−128	−109	−128	−98	4	7
35	1	−45	−61	−59	−21	11	31

(c)

Figura 11.6 Ejemplo de un bloque de 8 × 8. (a) Vista en escala de grises (b) Valores de píxel en la escala de grises (c) Valores de píxel en la escala de grises menos 128.

```
>> x=double(x);
>> r=x(:,:,1);g=x(:,:,2);b=x(:,:,3);
>> xgray=0.2126*r+0.7152*g+0.0722*b;
>> xgray=uint8(xgray);
>> imagesc(xgray);colormap(gray)
```

Observe que se ha convertido el tipo de datos predeterminado para MATLAB `uint8`, o enteros sin signo, en números reales de doble precisión antes de hacer el cálculo. Lo mejor es volver a convertir al tipo `uint8` antes de dibujar la imagen con `imagesc`.

En la figura 11.5(b) se muestra un método burdo de compresión, donde se sustituye cada bloque de pixeles de 8×8 por su valor promedio de píxel. El tamaño de la compresión de datos es considerable (sólo hay $(32)^2 = 2^{10}$ bloques, cada uno representado ahora por un único entero) pero la calidad de la imagen resultante es pobre. El objetivo es comprimir de una manera menos burda, mediante la sustitución de cada uno de los bloques de 8×8 por algunos enteros que retengan mejor la información de la imagen original.

Para empezar se simplifica el problema a un solo bloque de pixeles de 8×8, como se muestra en la figura 11.6(a). El bloque se tomó del centro del ojo izquierdo del gorila de la figura 11.5. En la figura 11.6(b) se muestran los números enteros de un byte que representan las intensidades en escala de grises de los 64 pixeles. En la figura 11.6(c) se ha restado 256/2 = 128 del número de pixeles para que estén centrados alrededor de cero. Este paso no es esencial, pero el centrado resultará en un mejor uso de la TDC-2D.

Para comprimir el bloque de pixeles de 8×8 mostrado, se transformará la matriz de valores de píxel en escala de grises

$$X = \begin{bmatrix} -18 & 40 & 48 & 54 & 42 & 31 & 6 & 17 \\ 38 & 40 & 36 & 33 & 37 & 43 & 31 & 13 \\ 18 & -10 & -4 & -6 & -9 & 17 & 34 & 16 \\ -26 & -94 & -106 & -103 & -90 & -17 & 18 & 31 \\ -21 & -79 & 2 & 31 & -126 & -99 & -11 & 36 \\ -33 & -57 & 25 & 79 & -113 & -98 & -6 & 22 \\ -16 & -107 & -128 & -109 & -128 & -98 & 4 & 7 \\ 35 & 1 & -45 & -61 & -59 & -21 & 11 & 31 \end{bmatrix} \tag{11.16}$$

y se confiará en la capacidad de la TDC-2D para ordenar la información de acuerdo con su importancia para el sistema visual humano. Se calcula la TDC-2D de X como

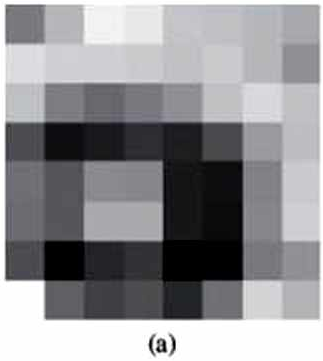
$$Y = C_8 X C_8^T = \begin{bmatrix} -121 & -66 & 127 & -65 & 27 & 98 & 7 & -25 \\ 200 & 22 & -124 & 34 & -36 & -62 & 5 & 6 \\ 113 & 43 & -32 & 55 & -25 & -75 & -21 & 12 \\ -10 & 35 & -69 & -131 & 28 & 54 & -4 & -24 \\ -14 & -18 & 16 & 1 & -5 & -27 & 14 & -6 \\ -124 & -74 & 47 & 60 & -1 & -16 & -8 & 13 \\ 81 & 35 & -57 & -54 & -7 & 6 & 1 & -16 \\ -16 & 11 & 5 & -15 & 11 & 12 & -1 & 9 \end{bmatrix}, \tag{11.17}$$

después de redondear, por simplicidad, al entero más próximo. Este redondeo añade una pequeña cantidad de error adicional y no es estrictamente necesario, pero de nuevo, ayudará a la compresión. Tenga en cuenta que, debido a las amplitudes más grandes, hay una tendencia a almacenar más información en la parte superior izquierda de la matriz de la transformada Y, en comparación con la parte inferior derecha. La parte inferior derecha representa las funciones básicas de más alta frecuencia, que suelen ser menos importantes para el sistema visual. Sin embargo, debido a que la TDC-2D es una transformación invertible, la información en Y puede utilizarse para reconstruir por completo la imagen original, hasta el redondeo.

La primera estrategia de compresión que se intentará será una forma de filtrado de pasa-bajas. Como se analizó en la sección anterior, la aproximación por mínimos cuadrados con la TDC-2D es sólo una cuestión de eliminar términos de la función de interpolación $P_8(s, t)$. Por ejemplo, puede cortarse la contribución de las funciones con frecuencia espacial relativamente alta al establecer que toda $y_{kl} = 0$ para $k + l \geq 7$ (recuerde que se siguen numerando las entradas de la matriz como $0 \leq k, l \leq 7$). Después del filtrado pasa-bajas, los coeficientes de la transformada son

$$Y_{\text{bajas}} = \begin{bmatrix} -121 & -66 & 127 & -65 & 27 & 98 & 7 & 0 \\ 200 & 22 & -124 & 34 & -36 & -62 & 0 & 0 \\ 113 & 43 & -32 & 55 & -25 & 0 & 0 & 0 \\ -10 & 35 & -69 & -131 & 0 & 0 & 0 & 0 \\ -14 & -18 & 16 & 0 & 0 & 0 & 0 & 0 \\ -124 & -74 & 0 & 0 & 0 & 0 & 0 & 0 \\ 81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11.18}$$

Para reconstruir la imagen se aplica la inversa de la TDC-2D como $C_8^T Y_{\text{bajas}} C_8$ y se obtienen los valores de píxel en escala de grises que se muestran en la figura 11.7. La imagen en el inciso (a) es similar al original de la figura 11.6(a), pero a diferente nivel de detalle.



199	151	191	185	162	158	152	141
177	169	170	165	159	164	152	127
160	113	98	110	126	158	174	160
78	34	41	55	43	75	133	156
103	83	123	119	39	35	115	164
100	84	143	141	39	31	120	167
77	18	48	50	-3	28	111	126
206	89	68	76	47	103	173	150

-19	23	63	57	34	30	24	13
49	41	42	37	31	36	24	-1
32	-15	-30	-16	-2	30	46	32
-50	-94	-87	-73	-85	-53	5	28
-25	-45	-5	-9	-89	-93	-13	36
-28	-44	15	13	-89	-97	-8	39
-51	-110	-80	-69	-131	-102	-17	-2
78	-39	-60	-52	-81	-25	45	22

(a) (b) (c)

Figura 11.7 Resultado del filtrado pasa-bajas. (a) Imagen filtrada (b) Valores de píxel en escala de grises, después de transformar y sumar 128 (c) Datos transformados inversos.

¿Cuánto se ha comprimido la información del bloque de 8×8? La imagen original puede reconstruirse (sin pérdidas, excepto por el redondeo al número entero) mediante la transformada inversa de la TDC-2D (11.17) y la suma de 128. Al hacer el filtrado pasa-bajas con la matriz (11.17), se han reducido los requisitos de almacenamiento aproximadamente a la mitad, mientras se conserva la mayoría de los aspectos cualitativos visuales del bloque.

11.2.3 Cuantificación

La idea de cuantificación permitirá que se logren los efectos de filtrado de pasa-bajas de una manera más selectiva. En lugar de ignorar por completo los coeficientes, conservaremos versiones de baja precisión de algunos coeficientes, a un costo inferior de almacenamiento. Esta idea explota los mismos aspectos del sistema visual humano (que es menos sensible a las altas frecuencias espaciales). La idea principal es asignar menos bits para almacenar información sobre la esquina inferior derecha de la matriz Y de la transformada, en lugar de tirarla a la basura.

Módulo q de cuantificación

Cuantificación: $z = \text{redondear}\left(\frac{y}{q}\right)$

Descuantificación: $\bar{y} = qz$ (11.19)

Aquí, "redondear" significa "al entero más cercano". El **error de cuantificación** es la diferencia entre la entrada y y la salida $\bar{y}$ después de la cuantificación y descuantificación. El máximo error del módulo de cuantificación q es $q/2$.

► EJEMPLO 11.4 Cuantifique los números -10, 3 y 65 con el módulo 8.

Los valores cuantificados son -1, 0 y 8. Al descuantificar, los resultados son -8, 0 y 64. Los errores son $|-2|$, $|3|$ y $|1|$, respectivamente, cada uno menor que $q/2 = 4$. ◄

De regreso al ejemplo de la imagen, el número de bits permitidos para cada frecuencia puede elegirse en forma arbitraria. Sea Q una matriz de 8×8 llamada la **matriz de cuantificación**. Las entradas q_{kl}, $0 \leq k, l \leq 7$ regularán la cantidad de bits que se asigna a cada entrada de la matriz Y de la transformada. Cambie Y por la matriz comprimida

$$Y_Q = \left[\text{redondear}\left(\frac{y_{kl}}{q_{kl}}\right)\right], 0 \leq k, l \leq 7. \tag{11.20}$$

La entrada de la matriz Y se divide entre la matriz de cuantificación. El redondeo subsiguiente es donde ocurre la pérdida, y hace que este método sea una forma de compresión con pérdida. Observe que cuanto más grande es la entrada de Q, se pierde mucho más en la cuantificación.

Como primer ejemplo, la **cuantificación lineal** está definida por la matriz

$$q_{kl} = 8p(k + l + 1) \text{ para } 0 \leq k, l \leq 7 \tag{11.21}$$

para alguna p constante, llamada el **parámetro pérdida**. Por lo tanto,

$$Q = p\begin{bmatrix} 8 & 16 & 24 & 32 & 40 & 48 & 56 & 64 \\ 16 & 24 & 32 & 40 & 48 & 56 & 64 & 72 \\ 24 & 32 & 40 & 48 & 56 & 64 & 72 & 80 \\ 32 & 40 & 48 & 56 & 64 & 72 & 80 & 88 \\ 40 & 48 & 56 & 64 & 72 & 80 & 88 & 96 \\ 48 & 56 & 64 & 72 & 80 & 88 & 96 & 104 \\ 56 & 64 & 72 & 80 & 88 & 96 & 104 & 112 \\ 64 & 72 & 80 & 88 & 96 & 104 & 112 & 120 \end{bmatrix}.$$

En MATLAB la matriz lineal de cuantificación puede definirse mediante `Q=p*8./hilb(8);`

El parámetro de pérdida p es un botón que puede activarse para intercambiar bits por precisión visual. Entre menor sea el parámetro de pérdida, mejor será la reconstrucción. El conjunto de números resultante en la matriz Y_Q representa la nueva versión cuantificada de la imagen.

Para descomprimir el archivo la matriz Y_Q se descuantifica invirtiendo el proceso, que es la multiplicación de cada entrada por Q. Ésta es la parte con pérdidas de la codificación de imágenes. Al sustituir las entradas y_{kl} dividiendo entre q_{kl} y redondeando, para después reconstruir multiplicando por q_{kl}, se añade un error de tamaño $q_{kl}/2$ a y_{kl}. Éste es el error de cuantificación. Cuanto mayor sea q_{kl}, mayor será la posibilidad de error en la reconstrucción de la imagen. Por otra parte, cuanto mayor sea q_{kl}, menores serán las entradas enteras de Y_Q, y será necesario almacenar menos bits. Ésta es la compensación entre la precisión de la imagen y el tamaño de archivo.

De hecho, con la cuantificación se logran dos cosas: muchas contribuciones pequeñas con frecuencias altas se igualan de inmediato a cero por (11.20), y las contribuciones que permanecen distintas de cero tienen un tamaño reducido, de modo que pueden transmitirse o almacenarse utilizando menos bits. El conjunto resultante de números se convierte en un flujo de bits con el uso de la codificación de Huffman, que se analiza en la sección siguiente.

A continuación se muestra la serie completa de pasos para la compresión de una matriz de valores de píxel en MATLAB. La salida del comando `imread` de MATLAB es una matriz de $m \times n$ enteros de 8 bits para una fotografía en escala de grises, o tres de esas matrices para una fotografía a color. (Las tres matrices contienen la información para el rojo, verde y azul, respectivamente; más adelante se tratará el color con mayor detalle). Un entero de 8 bits se llama `uint8`, para distinguirlo de un `double`, que requiere 64 bits de almacenamiento, como se estudió en el capítulo 0. El comando `double(x)` convierte el número `x` `uint8` al formato `double`, y el comando `uint8(x)` hace lo contrario al redondear `x` al entero más cercano entre 0 y 255.

Los siguientes cuatro comandos realizan la conversión, el centrado, la transformación y la cuantificación de una matriz cuadrada X de $n \times n$ números `uint8`, como las matrices de píxel de 8×8 consideradas con anterioridad. Se indica por `C` la matriz de la TDC de $n \times n$.

```
>> Xd=double(X);
>> Xc=Xd-128;
>> Y=C*Xc*C';
>> Yq=round(Y./Q);
```

En este punto la `Yq` resultante se almacena o se transmite. Para recuperar la imagen es necesario deshacer los cuatro pasos en orden inverso:

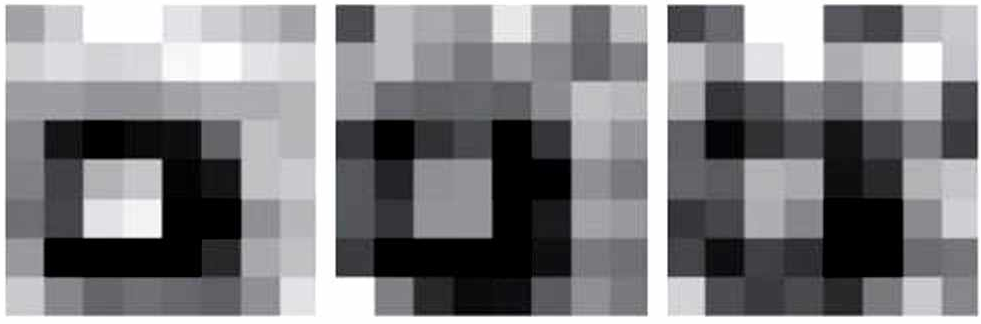
```
>> Ydq=Yq.*Q;
>> Xdq=C'*Ydq*C;
>> Xe=Xdq+128;
>> Xf=uint8(Xe);
```

Después de la descuantificación se aplica la transformada inversa de la TDC, se suma de nuevo el desplazamiento de 128, y el formato `double` se convierte otra vez en una matriz `Xf` de enteros `uint8`.

Cuando se aplica la cuantificación lineal a (11.17) con $p = 1$, los coeficientes resultantes son

$$Y_Q = \begin{bmatrix} -15 & -4 & 5 & -2 & 1 & 2 & 0 & 0 \\ 13 & 1 & -4 & 1 & -1 & -1 & 0 & 0 \\ 5 & 1 & -1 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & -1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11.22}$$

El bloque de imagen reconstruido, que se formó al descuantificar y transformar de manera inversa Y_Q, se muestra en la figura 11.8(a). Pueden verse pequeñas diferencias en comparación con el bloque original, pero es más fiel que la reconstrucción con el filtrado pasa-bajas.



(a) (b) (c)

Figura 11.8 Resultado de la cuantificación lineal. El parámetro de pérdida es (a) $p = 1$ (b) $p = 2$ (c) $p = 4$.

Después de la cuantificación lineal con $p = 2$, los coeficientes de la transformada cuantificada son

$$Y_Q = \begin{bmatrix} -8 & -2 & 3 & -1 & 0 & 1 & 0 & 0 \\ 6 & 0 & -2 & 0 & 0 & -1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{11.23}$$

y después de la cuantificación lineal con $p = 4$, los coeficientes de la transformada cuantificada son

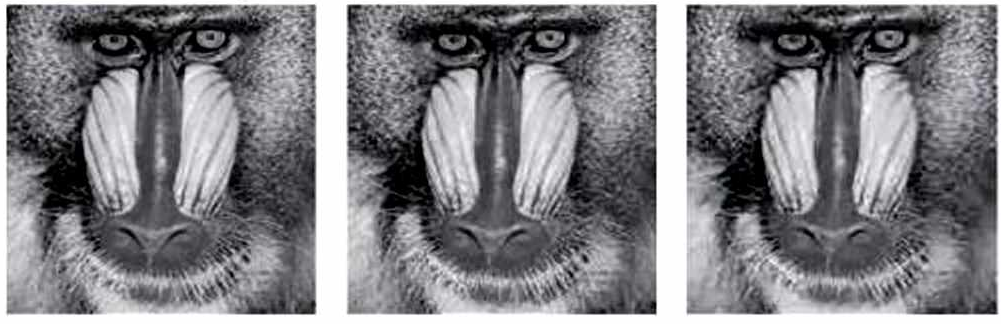
$$Y_Q = \begin{bmatrix} -4 & -1 & 1 & -1 & 0 & 1 & 0 & 0 \\ 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{11.24}$$

En la figura 11.8 se muestra el resultado de la cuantificación lineal para los tres distintos valores del parámetro de pérdida p. Tenga en cuenta que entre mayor sea el valor del parámetro de pérdida p, más entradas de la matriz Y_Q se igualarán a cero por el procedimiento de cuantificación, menores serán los requisitos de datos para representar los pixeles y se reconstruirá con menos fidelidad la imagen original.

A continuación se cuantifican los $32 \times 32 = 1024$ bloques de la imagen de la figura 11.5. Es decir, se llevan a cabo 1024 versiones independientes del ejemplo anterior. En la figura 11.9 se muestran los resultados para el parámetro de pérdida $p = 1, 2$ y 4. La imagen comienza a deteriorarse de manera significativa con $p = 4$.

Puede hacerse un cálculo aproximado para cuantificar la cantidad de la compresión de la imagen debido a la cuantificación. La imagen original utiliza un valor para los pixeles de 0 a 255, que es de un byte, u 8 bits. Para cada bloque de 8×8, el número total de bits necesarios sin compresión es $8(8)^2 = 512$ bits.

Ahora, suponga que se usa la cuantificación lineal con el parámetro de pérdida $p = 1$. Suponga que la entrada máxima de la transformada Y es 255. Entonces, las mayores entradas posibles de Y_Q, después de la cuantificación por Q, son



(a) (b) (c)

Figura 11.9 Resultado de la cuantificación lineal para los 1024 bloques de 8 × 8. Los parámetros de pérdida son (a) $p = 1$ (b) $p = 2$ (c) $p = 4$.

$$\begin{bmatrix} 32 & 16 & 11 & 8 & 6 & 5 & 5 & 4 \\ 16 & 11 & 8 & 6 & 5 & 5 & 4 & 4 \\ 11 & 8 & 6 & 5 & 5 & 4 & 4 & 3 \\ 8 & 6 & 5 & 5 & 4 & 4 & 3 & 3 \\ 6 & 5 & 5 & 4 & 4 & 3 & 3 & 3 \\ 5 & 5 & 4 & 4 & 3 & 3 & 3 & 2 \\ 5 & 4 & 4 & 3 & 3 & 3 & 2 & 2 \\ 4 & 4 & 3 & 3 & 3 & 2 & 2 & 2 \end{bmatrix}.$$

Dado que pueden presentarse tanto entradas positivas como negativas, el número de bits necesario para almacenar cada entrada es

$$\begin{bmatrix} 7 & 6 & 5 & 5 & 4 & 4 & 4 & 4 \\ 6 & 5 & 5 & 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 4 & 4 & 4 & 4 & 4 & 3 \\ 5 & 4 & 4 & 4 & 4 & 4 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix}.$$

La suma de estos 64 números es 249, o $249/64 \approx 3.89$ bits/píxel, que es menos de la mitad del número de bits (512, u 8 bits/píxel) necesario para almacenar los valores de píxel originales de la matriz de 8×8 de la imagen. Las estadísticas correspondientes a los otros valores de p se muestran en la siguiente tabla:

p	bits totales	bits/píxel
1	249	3.89
2	191	2.98
4	147	2.30

Como se ve en la tabla, el número de bits necesarios para representar la imagen se reduce por un factor de 2 cuando $p = 1$, con poco cambio reconocible en la imagen. Esta compresión se debe a la cuantificación. Con el fin de comprimir aún más, puede aprovecharse el hecho de que muchos de los términos con alta frecuencia en la transformada son iguales a cero después de la cuantificación. Esto puede hacerse de manera más eficiente usando a Huffman y la codificación de corrida larga, que se presenta en la siguiente sección.

La cuantificación lineal con $p = 1$ está cerca de la cuantificación predeterminada para JPEG. La matriz de cuantificación que proporciona la mayor compresión con la menor degradación de la

imagen ha sido objeto de mucha investigación y análisis. El estándar JPEG incluye un apéndice llamado "Anexo K: Ejemplos y directrices", que contiene una Q basada en experimentos con el sistema visual humano. La matriz

$$Q_Y = p\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{11.25}$$

se utiliza mucho en los codificadores JPEG que se distribuyen en la actualidad. Al establecer el parámetro de pérdida $p = 1$ debe obtenerse una reconstrucción casi perfecta en lo que al sistema visual humano se refiere, mientras que con $p = 4$ se presentan por lo general defectos notables. Hasta cierto punto, la calidad visual depende del tamaño del píxel: si los pixeles son pequeños, algunos errores pueden pasar inadvertidos.

Hasta ahora se ha realizado el análisis sólo en la escala de grises. Resulta bastante fácil extender la aplicación a imágenes en color, que pueden expresarse en el sistema de color RGB. A cada píxel se le asigna tres enteros, cada uno para la intensidad del rojo, el verde y el azul. Un enfoque para la compresión de imágenes es repetir el procesamiento anterior de forma independiente para cada uno de los tres colores, tratándolos como si estuvieran en la escala de grises, para finalmente reconstituir la imagen a partir de sus tres colores.

Aunque el estándar JPEG no se pronuncia sobre la forma en que debe tratarse el color, el método que con frecuencia se considera el básico de JPEG, utiliza un enfoque más delicado. Defina la **luminancia** $Y = 0.299R + 0.587G + 0.114B$ y las **diferencias de color** $U = B - Y$ y $V = R - Y$. Esto convierte los datos de color RGB al sistema YUV. Lo anterior implica una transformada completamente reversible, puesto que los valores en RGB pueden encontrarse como $B = U + Y$, $R = V + Y$, y $G = (Y - 0.299R - 0.114B)/(0.587)$. Este método básico de JPEG aplica el filtrado de la TDC que se analizó antes de manera independiente para Y, U y V, usando la matriz de cuantificación Q_Y del anexo K para la variable de luminancia Y, y la matriz de cuantificación

$$Q_C = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix} \tag{11.26}$$

para las diferencias de color U y V. Después de reconstruir Y, U y V, se unen de nuevo y vuelven a convertirse a RGB para reconstituir la imagen.

Debido a los papeles menos importantes de U y V en el sistema visual humano, se permite una cuantificación más agresiva en ellas, como se ve en (11.26). Puede obtenerse compresión adicional a partir de una serie de trucos ad hoc adicionales (por ejemplo, promediar las diferencias de color y tratarlas en una malla menos fina).

11.2 Ejercicios

1. Encuentre la TDC-2D de las siguientes matrices de datos X, y encuentre la correspondiente función interpolante $P_2(s, t)$ para los puntos de datos (i, j, x_{ij}), $i, j = 0, 1$:

(a) $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ (d) $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

2. Encuentre la TDC-2D de la matriz de datos X, y encuentre la correspondiente función interpolante $P_n(s, t)$ para los puntos de datos (i, j, x_{ij}), $i, j = 0, \ldots, n-1$.

(a) $\begin{bmatrix} 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

(c) $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ (d) $\begin{bmatrix} 3 & 3 & 3 & 3 \\ 3 & -1 & -1 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & -1 & -1 & 3 \end{bmatrix}$

3. Encuentre la aproximación por mínimos cuadrados, utilizando las funciones básicas 1, $\cos\frac{(2s+1)\pi}{8}$, $\cos\frac{(2t+1)\pi}{8}$ para los datos del ejercicio 2.

4. Utilice la matriz de cuantificación $Q = \begin{bmatrix} 10 & 20 \\ 20 & 100 \end{bmatrix}$ para cuantificar las matrices siguientes. Indique la matriz cuantificada, la matriz (con pérdidas) descuantificada y la matriz de los errores de cuantificación.

(a) $\begin{bmatrix} 24 & 24 \\ 24 & 24 \end{bmatrix}$ (b) $\begin{bmatrix} 32 & 28 \\ 28 & 45 \end{bmatrix}$ (c) $\begin{bmatrix} 54 & 54 \\ 54 & 54 \end{bmatrix}$

11.2 Problemas de computadora

1. Encuentre la TDC-2D de la matriz de datos X.

(a) $\begin{bmatrix} -1 & 1 & -1 & 1 \\ -2 & 2 & -2 & 2 \\ -3 & 3 & -3 & 3 \\ -4 & 4 & -4 & 4 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 2 & -1 & -2 \\ -1 & -2 & 1 & 2 \\ 1 & 2 & -1 & -2 \\ -1 & -2 & 1 & 2 \end{bmatrix}$

(c) $\begin{bmatrix} 1 & 3 & 1 & -1 \\ 2 & 1 & 0 & 1 \\ 1 & -1 & 2 & 3 \\ 3 & 2 & 1 & 0 \end{bmatrix}$ (d) $\begin{bmatrix} -3 & -2 & -1 & 0 \\ -2 & -1 & 0 & 1 \\ -1 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}$

2. Con la TDC-2D del problema de computadora 1 encuentre la aproximación a X por mínimos cuadrados con filtrado pasa-bajas; establezca todos los valores de transformación $Y_{kl} = 0$, para $k + l \leq 4$.

3. Escoja un archivo de imagen en escala de grises y utilice el comando `imread` para importarlo a MATLAB. Corte la matriz resultante de modo que cada dimensión sea un múltiplo de 8. Si es necesario, convierta una imagen en color RGB a la escala de grises; esto puede hacerse mediante la fórmula general (11.15).

   (a) Por ejemplo, extraiga un bloque de 8×8 pixeles usando el comando `xb=x(81:88, 81:88)` de MATLAB. Despliegue el bloque con el comando `imagesc`.
   (b) Aplique la TDC-2D.
   (c) Cuantifique usando la cuantificación lineal con $p = 1, 2$ y 4. Imprima cada Y_Q.

(d) Reconstruya el bloque utilizando la TDC-2D inversa y compárelo con el original. Utilice los comandos `colormap(gray)` y `imagesc(X, [0 255])` de MATLAB.
(e) Realice los pasos (a) a (d) para los 8×8 bloques y reconstruya la imagen en cada caso.

4. Realice los pasos del problema de computadora 3, pero cuantifique mediante la matriz de cuantificación sugerida para JPEG (11.25) con $p = 1$.
5. Elija un archivo de imagen a color y realice los pasos del problema de computadora 3 para los colores R, G y B en forma separada, utilizando la cuantificación lineal, y combine de nuevo como una imagen a color.
6. Utilice una imagen a color y transforme los valores RGB a coordenadas de luminancia/diferencia de color. Realice los pasos del problema de computadora 3 para Y, U y V en forma separada mediante la cuantificación JPEG, y combine de nuevo como una imagen a color.

11.3 CODIFICACIÓN DE HUFFMAN

La compresión con pérdida para imágenes requiere hacer un intercambio entre la precisión y el tamaño del archivo. Si las reducciones de precisión son lo suficientemente pequeñas para ser imperceptibles al propósito pretendido de la imagen, el intercambio puede valer la pena. La pérdida de precisión se produce en el paso de cuantificación, después de la transformación para separar la imagen en sus frecuencias espaciales. La compresión sin pérdida se refiere a la compresión adicional que puede aplicarse sin perder más precisión, tan sólo debido a la codificación eficaz de la imagen cuantificada de la transformada TDC.

En esta sección se analiza la compresión sin pérdida. Como una aplicación importante, existen métodos sencillos y eficaces para convertir la matriz de la transformada TDC cuantificada que se estudió en la sección anterior, en un flujo de bits JPEG. Para encontrar la manera de hacer esto, será necesario realizar un viaje corto por la teoría de la información básica.

11.3.1 Teoría de la información y codificación

Considere un mensaje que consiste en una cadena de símbolos. Los símbolos son arbitrarios; suponga que provienen de un conjunto finito. En esta sección se consideran las formas eficientes para codificar una cadena de dígitos binarios, o bits. Cuanto más corta sea la cadena de bits, más fácil y más sencillo será almacenar o transmitir el mensaje.

► EJEMPLO 11.5 Codifique el mensaje ABAACDAB como una cadena binaria.

Como hay cuatro símbolos, una codificación binaria conveniente podría asociar dos bits con cada letra. Por ejemplo, podría elegirse la correspondencia

A	00
B	01
C	10
D	11

Entonces, el mensaje se codifica como

(00)(01)(00)(00)(10)(11)(00)(01).

Con este código se requiere un total de 16 bits para almacenar o transmitir el mensaje. ◄

Resulta que hay métodos de codificación más eficientes. Para comprenderlos, primero debe presentarse la idea de la información. Suponga que se tienen k símbolos diferentes y se indica mediante p_i la probabilidad de la aparición del símbolo i en cualquier punto de la cadena. La pro-

babilidad puede conocerse a priori, o puede estimarse de manera empírica dividiendo el número de apariciones del símbolo i en la cadena entre la longitud de la cadena.

DEFINICIÓN 11.7 La **información de Shannon,** o **entropía de Shannon** de la cadena es $I = -\sum_{i=1}^{k} p_i \log_2 p_i$.

❐

La definición se llama así en honor de C. Shannon de los Laboratorios Bell, quien hizo un trabajo trascendental sobre la teoría de la información a mediados del siglo XX. La información de Shannon de una cadena se considera un promedio del número de bits por símbolo que se requiere, como mínimo, para codificar el mensaje. La lógica es la siguiente: en promedio, si un símbolo aparece p_i de las veces, entonces se espera necesitar $-\log_2 p_i$ bits para representarlo. Por ejemplo, un símbolo que aparece 1/8 de las veces podría representarse mediante uno de los símbolos de $-\log_2(1/8) = 3$ bits, 000, 001, ... , 111, de los cuales hay 8. A fin de encontrar los bits promedio por símbolo de todos los símbolos, es necesario ponderar los bits por símbolo i de acuerdo con su probabilidad p_i. Esto significa que el número promedio de bits/símbolo para todo el mensaje es la sumatoria I de la definición.

► EJEMPLO 11.6 Encuentre la información de Shannon de la cadena ABAACDAB.

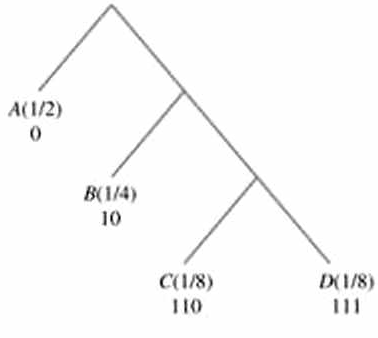
Las probabilidades empíricas de aparición de los símbolos A, B, C, D son $p_1 = 4/8 = 2^{-1}$, $p_2 = 2/8 = 2^{-2}$, $p_3 = 1/8 = 2^{-3}$, $p_4 = 2^{-3}$, respectivamente. La información de Shannon es

$$-\sum_{i=1}^{4} p_i \log_2 p_i = \frac{1}{2}1 + \frac{1}{4}2 + \frac{1}{8}3 + \frac{1}{8}3 = \frac{7}{4}.$$ ◄

Por lo tanto, la información de Shannon estima que se requieren al menos 1.75 bits/símbolo para codificar la cadena. Como la cadena tiene una longitud de 8, el número total óptimo de bits debería ser (1.75)(8) = 14, no 16, como se codificó la cadena anteriormente.

De hecho, puede enviarse el mensaje a través de los 14 bits predichos, utilizando el método conocido como **codificación de Huffman**. El objetivo es asignar un código binario único para cada símbolo que refleje la probabilidad de encontrar dicho símbolo, donde los símbolos más comunes reciben los códigos más cortos.

El algoritmo funciona mediante la construcción de un árbol a partir del cual puede leerse el código binario. Comience con los dos símbolos con la menor probabilidad y considere el símbolo "combinado", asignándole la probabilidad combinada. Los dos símbolos forman una ramificación del árbol. Después repita este paso combinando los símbolos y avanzando hacia arriba por las ramas del árbol, hasta que sólo haya un grupo de símbolos a la izquierda, que corresponde a la parte superior del árbol. Aquí, primero se combinan lo símbolos menos probables C y D en un símbolo CD con probabilidad 1/4. Las probabilidades restantes son A (1/2), B (1/4) y CD (1/4). De nuevo, se combinan los dos símbolos con menor probabilidad para obtener A (1/2), BCD (1/2). Por último, la combinación de los dos símbolos restantes da ABCD (1). Cada combinación forma una rama del árbol de Huffman:



Una vez que el árbol se ha completado, el código de Huffman puede leerse para cada símbolo al recorrer el árbol desde la parte superior, escribiendo un 0 para una rama a la izquierda y un 1 para una rama a la derecha, como se mostró en la figura anterior. Por ejemplo, A está representada por 0 y C está representada por dos derechas y una izquierda, 110. Ahora la cadena de letras ABAACDAB puede traducirse en un flujo de bits de longitud 14:

$$(0)(10)(0)(0)(110)(111)(0)(10).$$

La información de Shannon del mensaje proporciona una cota inferior para los bits/símbolo de la codificación binaria. En este caso, el código de Huffman ha alcanzado una cota de la información de Shannon de 14/8 = 1.75 bits/símbolo. Desafortunadamente, esto no siempre es posible, como lo muestra el siguiente ejemplo.

► **EJEMPLO 11.7** Encuentre la información de Shannon y una codificación de Huffman para el mensaje ABRA CADABRA.

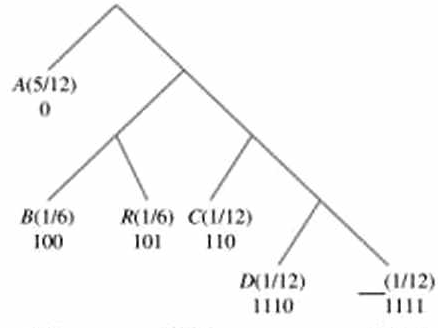
Las probabilidades empíricas de los seis símbolos son

A	5/12
B	2/12
R	2/12
C	1/12
D	1/12
_	1/12

Observe que el espacio se ha tratado como un símbolo. La información de Shannon es

$$-\sum_{i=1}^{6} p_i \log_2 p_i = -\frac{5}{12}\log_2\frac{5}{12} - 2\frac{1}{6}\log_2\frac{1}{6} - 3\frac{1}{12}\log_2\frac{1}{12} \approx 2.28 \text{ bits/símbolo.}$$

Éste es el mínimo teórico del promedio de los bits/símbolo para codificar el mensaje ABRA CADABRA. Para encontrar la codificación de Huffman proceda como se ha descrito. Se inicia haciendo la combinación de los símbolos D y _, aunque podrían haberse escogido cualesquiera dos de los tres con probabilidad de 1/12 como la rama más baja. El símbolo A se toma al final, puesto que tiene mayor probabilidad. En el siguiente diagrama se muestra una codificación de Huffman.



Observe que A tiene un código corto, debido a que es un símbolo reiterado en el mensaje. La secuencia binaria codificada para ABRA CADABRA es

$$(0)(100)(101)(0)(1111)(110)(0)(1110)(0)(100)(101)(0),$$

que tiene una longitud de 28 bits. El promedio para esta codificación es $28/12 = 2\frac{1}{3}$ bits/símbolo, un poco más grande que el mínimo teórico calculado con anterioridad. Los códigos de Huffman no siempre pueden coincidir con la información de Shannon, pero a menudo se acercan mucho. ◄

El secreto de un código de Huffman es el siguiente: dado que cada símbolo se presenta sólo al final de una rama del árbol, ningún código de símbolo completo puede ser el inicio de otro código de símbolo. Por lo tanto, al convertir de nuevo el código en símbolos no se presenta ninguna ambigüedad.

11.3.2 Codificación de Huffman para el formato JPEG

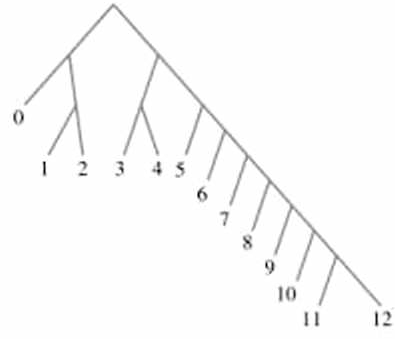
Esta sección se dedica a un extenso ejemplo práctico de codificación de Huffman. El formato de compresión de imágenes JPEG siempre está presente en la fotografía digital moderna y constituye un estudio de caso fascinante debido a la yuxtaposición de las matemáticas teóricas y ciertas consideraciones de la ingeniería.

La codificación binaria de los coeficientes de transformación para un archivo de imagen JPEG utiliza la codificación de Huffman en dos formas diferentes, una para el componente DC (la entrada (0, 0) de la matriz de la transformada) y otro para las otras 63 entradas de la matriz de 8×8, los componentes llamados AC.

DEFINICIÓN 11.8 Sea y un entero. El **tamaño** de y se define como

$$L = \begin{cases} \text{piso}(\log_2 |y|) + 1 & \text{si } y \neq 0 \\ 0 & \text{si } y = 0 \end{cases}. \qquad \square$$

La codificación de Huffman para JPEG tiene tres ingredientes: un árbol de Huffman para los componentes DC, otro árbol de Huffman para los componentes AC y una tabla identificadora de enteros. La primera parte de la codificación para la entrada $y = y_{00}$ es la codificación binaria para el tamaño de y, a partir del siguiente árbol de Huffman para los componentes DC, llamado el **árbol DPCM** (modulación de código del pulso diferencial).



Una vez más, el árbol debe interpretarse mediante la codificación de un 0 o un 1 al bajar por una rama hacia la izquierda o hacia la derecha, respectivamente. La primera parte está seguida por una cadena binaria extraída de la siguiente tabla identificadora de enteros:

L	entrada	binario
0	0	- -
1	−1,1	0,1
2	−3,−2,2,3	00,01,10,11
3	−7,−6,−5,−4,4,5,6,7	000,001,010,011,100,101,110,111
4	−15,−14,...,−8,8,...,14,15	0000,0001,.....0111,1000,.....,1110,1111
5	−31,−30,...,−16,16,...,30,31	00000,00001,.....,01111,10000.....,11110,11111
6	−63,−62,...,−32,32,...,62,63	000000,000001,...,011111,100000,...,111110,111111
⋮	⋮	⋮

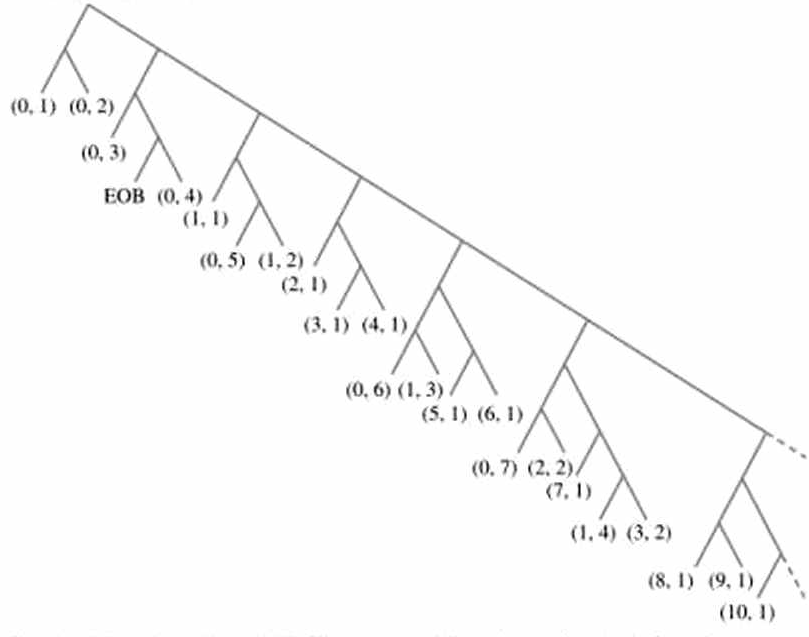
Como ejemplo, la entrada $y_{00} = 13$ tendría un tamaño $L = 4$. De acuerdo con el árbol DPCM, el código de Huffman para 4 es (101). La tabla muestra que los dígitos adicionales para 13 son (1101), por lo que se almacenaría la concatenación de las dos partes, 1011101, para el componente DC.

Dado que a menudo existen correlaciones entre los componentes DC de los bloques de 8×8 cercanos, después del primer bloque sólo se almacenan las diferencias entre los bloques. Estas diferencias se almacenan de izquierda a derecha, utilizando el árbol DPCM.

Para los restantes 63 componentes AC del bloque de 8×8, se usa la **codificación de la longitud de corrida** (RLE, por sus siglas en inglés) como una manera de almacenar eficazmente largas cadenas de ceros. El orden convencional para almacenar los 63 componentes es el patrón de zigzag

$$\begin{bmatrix} 0 & 1 & 5 & 6 & 14 & 15 & 27 & 28 \\ 2 & 4 & 7 & 13 & 16 & 26 & 29 & 42 \\ 3 & 8 & 12 & 17 & 25 & 30 & 41 & 43 \\ 9 & 11 & 18 & 24 & 31 & 40 & 44 & 53 \\ 10 & 19 & 23 & 32 & 39 & 45 & 52 & 54 \\ 20 & 22 & 33 & 38 & 46 & 51 & 55 & 60 \\ 21 & 34 & 37 & 47 & 50 & 56 & 59 & 61 \\ 35 & 36 & 48 & 49 & 57 & 58 & 62 & 63 \end{bmatrix}. \tag{11.27}$$

En vez de codificar los 63 números en sí, se codifica un par (n, L) con longitud de corrida cero, donde n indica la longitud de una corrida de ceros, y L representa el tamaño de la entrada distinta de cero siguiente. En el siguiente árbol de Huffman para los componentes AC se muestran los códigos más comunes encontrados en las imágenes JPEG típicas, así como sus códigos predeterminados de acuerdo con el estándar JPEG.



En el flujo de bits, el código de Huffman en el árbol (que sólo identifica el tamaño de la entrada) está seguido inmediatamente por el código binario que identifica al número entero, con base en la tabla previa. Por ejemplo, la secuencia de entradas $-5, 0, 0, 0, 2$ se representa como $(0, 3)\ -5\ (3, 2)\ 2$, donde $(0, 3)$ significa que no hay ceros seguidos de un número de tamaño 3, y $(3, 2)$ representa 3 ceros seguidos de un número de tamaño 2. A partir del árbol de Huffman se encuentra que $(0, 3)$ se codifica como (100), y $(3, 2)$ como (111110111). El identificador de -5 es (010) y para 2 es (10), con base en la tabla identificadora de enteros. Por lo tanto, el flujo de bits utilizados para codificar $-5, 0, 0, 0, 2$ es (100)(010)(111110111)(10).

El árbol de Huffman anterior muestra sólo los códigos de longitud de corrida que aparecen con más frecuencia en JPEG. Otros códigos útiles son $(11, 1) = 1111111001$, $(12, 1) = 1111111010$ y $(13, 1) = 11111111000$.

► EJEMPLO 11.8 Codifique la matriz de transformación TDC cuantificada en (11.24) para un archivo de imagen JPEG.

La entrada DC $y_{00} = -4$ tiene un tamaño de 3, y se codifica como (100) mediante el árbol DPCM y los bits adicionales (011) a partir de la tabla identificadora de enteros. Después, se considera la cadena de coeficientes AC.

De acuerdo con (11.27), los coeficientes AC se ordenan como $-1, 3, 1, 0, 1, -1, -1$, siete ceros, 1, cuatro ceros, -1, tres ceros, -1 y ceros para el resto. La codificación de longitud de corrida comienza con -1, que tiene un tamaño de 1 y entonces contribuye (0, 1) con base en el código de longitud de corrida. El siguiente número 3 tiene un tamaño de 2 y contribuye (0, 2). Los pares con longitud de corrida cero son

$$(0, 1)\ -1\ (0, 2)\ 3\ (0, 1)\ 1\ (1, 1)\ 1\ (0, 1)\ -1\ (0, 1)\ -1$$
$$(7, 1)\ 1\ (4, 1)\ -1\ (3, 1)\ -1\ \text{EOB}.$$

Aquí, EOB significa end of block o "final del bloque", lo que significa que el resto de las entradas son ceros. Después se leen las representaciones de los bits a partir del árbol de Huffman en la página 518 y la tabla identificadora de enteros. A continuación se presenta el flujo de bits que almacena el bloque de 8×8 en la fotografía de la figura 11.8(c), donde los paréntesis se incluyen sólo para facilitar la lectura humana:

$$(100)(011)$$
$$(00)(0)(01)(11)(00)(1)(1100)(1)(00)(0)(00)(0)$$
$$(11111010)(1)(111011)(0)(111010)(0)(1010)$$

El bloque de pixeles en la figura 11.8(c), que es una aproximación razonable de la figura 11.6(a) original, se representa exactamente mediante estos 54 bits. Sobre la base de los bits por píxel, esto funciona hasta $54/64 \approx 0.84$ bits/píxel. Observe la superioridad de esta codificación sobre los bits/píxel obtenidos mediante el filtrado pasa-bajas y la cuantificación. Dado que los pixeles iniciaron como números enteros de 8 bits, la imagen de 8×8 se ha comprimido en un factor de más de 9:1. ◄

La descompresión de un archivo JPEG consiste en invertir los pasos de la compresión. El lector de JPEG decodifica el flujo de bits como símbolos de longitud de corrida, los cuales forman bloques de la transformada TDC de 8×8 que, a su vez, se convierten en bloques de píxeles mediante el uso de la TDC inversa.

11.3 Ejercicios

1. Encuentre la probabilidad de cada símbolo y la información de Shannon para los mensajes.
   (a) `BABBCABB` (b) `ABCACCAB` (c) `ABABCABA`
2. Dibuje un árbol de Huffman y utilícelo para codificar los mensajes del ejercicio 1. Compare la información de Shannon con el número promedio de bits necesarios por símbolo.
3. Dibuje un árbol de Huffman y convierta el mensaje, incluyendo espacios y signos de puntuación, en un flujo de bits mediante la codificación de Huffman. Compare la información de Shannon con el número promedio de bits necesarios por símbolo (a) `AY CARUMBA!` (b) `COMPRESS THIS MESSAGE` (c) `SHE SELLS SEASHELLS BY THE SEASHORE`
4. Traduzca los componentes transformados y cuantificados de la imagen (a) (11.22) y (b) (11.23) en flujos de bits, utilizando la codificación de Huffman para JPEG.

11.4 TDC MODIFICADA Y COMPRESIÓN DE AUDIO

En esta sección se abordará de nuevo el problema de las señales unidimensionales y el análisis de los métodos modernos para la compresión de audio. Aunque podría pensarse que una dimensión es más fácil de manejar que dos, el reto es que el sistema auditivo humano es muy sensible en el dominio de la frecuencia, y los defectos sonoros no deseados que introduce la compresión y descompresión se detectan aún con mayor facilidad. Por esa razón, es común que los métodos de compresión del sonido usen trucos sofisticados diseñados para ocultar la realización de una compresión.

En primer lugar se introduce la TDC4, una nueva versión de la transformada discreta del coseno y la denominada transformada discreta del coseno modificada (TDCM). La TDCM se representa por medio de una matriz no cuadrada, de modo que, a diferencia de la TDC y la TDC4, no es invertible. Sin embargo, cuando se aplica en la superposición de ventanas, puede utilizarse para reconstruir por completo el flujo de datos original. Más importante aún, puede combinarse con la cuantificación para realizar compresiones con pérdida con una mínima degradación de la calidad del sonido. La TDCM constituye la base de la mayoría de los actuales formatos de compresión de sonido, como MP3, AAC y WMA.

11.4.1 Transformada discreta del coseno modificada

Se iniciará con una forma ligeramente diferente de la TDC que se presentó con anterioridad. Existen cuatro versiones diferentes de la TDC que se utilizan por lo regular (en la sección anterior se usó la versión TDC1 para la compresión de imágenes). La versión TDC4 es la más popular para comprimir sonido.

DEFINICIÓN 11.9 La **transformada discreta del coseno (versión 4)** (TDC4) de $x = (x_0, \ldots, x_{n-1})^T$ es el vector n-dimensional

$$y = Ex,$$

donde E es la matriz de $n \times n$

$$E_{ij} = \sqrt{\frac{2}{n}} \cos \frac{(i + \frac{1}{2})(j + \frac{1}{2})\pi}{n}. \tag{11.28}$$

Al igual que en la TDC1, la matriz E en la TDC4 es una matriz ortogonal real: es cuadrada y sus columnas son vectores unitarios ortogonales por pares. Esto último se desprende del hecho de que las columnas de E son los vectores propios de la matriz real simétrica de $n \times n$

$$\begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 3 & \end{bmatrix}. \tag{11.29}$$

El ejercicio 6 le pide al lector que verifique este hecho.

Enseguida se destacan dos hechos importantes acerca de las columnas de la matriz de la TDC4. Tratar a n como fija y considerar no sólo las n columnas en la TDC4, sino también a los vectores de columna definidos por (11.28) para todos los números enteros positivos y negativos j.

LEMA 11.10 Indique con c_j la j-ésima columna de la matriz (11.28) de la TDC4 (extendida). Entonces (a) $c_j = c_{-1-j}$ para todos los números enteros j (las columnas son simétricas en torno a $j = -\frac{1}{2}$) y (b) $c_j = -c_{2n-1-j}$ para todos los números enteros j (las columnas son antisimétricas en torno a $j = n - \frac{1}{2}$). ■

Demostración. Para demostrar el inciso (a) del lema, escriba $j = -\frac{1}{2} + (j + \frac{1}{2})$ y $-1 - j = -\frac{1}{2} - (j + \frac{1}{2})$. Usando la ecuación (11.28) se obtiene

$$\begin{aligned} c_j = c_{-\frac{1}{2}+(j+\frac{1}{2})} &= \sqrt{\frac{2}{n}} \cos \frac{(i + \frac{1}{2})(j + \frac{1}{2})\pi}{n} = \sqrt{\frac{2}{n}} \cos \frac{(i + \frac{1}{2})(-j - \frac{1}{2})\pi}{n} \\ &= c_{-\frac{1}{2}-(j+\frac{1}{2})} = c_{-1-j} \end{aligned}$$

Para $i = 0, \ldots, n - 1$.

Para la demostración de (b), establezca $r = n - \frac{1}{2} - j$. Entonces, $j = n - \frac{1}{2} - r$ y $2n - 1 - j = n - \frac{1}{2} + r$, y debe demostrarse que $c_{n-\frac{1}{2}-r} + c_{n-\frac{1}{2}+r} = 0$. Por la fórmula de la suma de cosenos,

$$\begin{aligned} c_{n-\frac{1}{2}-r} &= \sqrt{\frac{2}{n}} \cos \frac{(2i+1)(n-r)\pi}{2n} \\ &= \sqrt{\frac{2}{n}} \cos \frac{2i+1}{2}\pi \cos \frac{(2i+1)r\pi}{2n} + \sqrt{\frac{2}{n}} \operatorname{sen} \frac{2i+1}{2}\pi \operatorname{sen} \frac{(2i+1)r\pi}{2n} \\ c_{n-\frac{1}{2}+r} &= \sqrt{\frac{2}{n}} \cos \frac{(2i+1)(n+r)\pi}{2n} \\ &= \sqrt{\frac{2}{n}} \cos \frac{2i+1}{2}\pi \cos \frac{(2i+1)r\pi}{2n} - \sqrt{\frac{2}{n}} \operatorname{sen} \frac{2i+1}{2}\pi \operatorname{sen} \frac{(2i+1)r\pi}{2n} \end{aligned}$$

para $i = 0, \ldots, n-1$. Puesto que $\cos \frac{1}{2}(2i+1)\pi = 0$ para todos los enteros i, la suma $c_{n-\frac{1}{2}-r} + c_{n-\frac{1}{2}+r} = 0$, tal como se solicitó.

Se utilizará la matriz E de la TDC4 para construir la transformada discreta del coseno modificada. Suponga que n es par. Se creará una nueva matriz, utilizando las columnas $c_{\frac{n}{2}}, \ldots, c_{\frac{5}{2}n-1}$. El lema 11.10 demuestra que para cualquier número entero j, la columna c_j puede expresarse como una de las columnas de la TDC4 (es decir, una de las c_j para $0 \le i \le n-1$, como se muestra en la figura 11.10, hasta un posible cambio de signo).

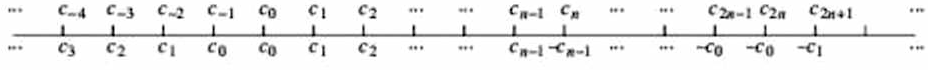


Figura 11.10 Ilustración del lema 11.10. Las columnas $c_0, \ldots, c_{n-1}$ constituyen la matriz de la TDC4 de $n \times n$. Para los números enteros j fuera de este rango, la columna definida por c_j en la ecuación (11.28) aún corresponde a una de las n columnas de la TDC4, mostrada justo debajo de ella en la figura. Esto ilustra el lema 11.10.

DEFINICIÓN 11.11 Sea n un número par entero positivo. La **transformada discreta del coseno modificada** (TDCM) de $x = (x_0, \ldots, x_{2n-1})^T$ es el vector n-dimensional

$$y = Mx, \tag{11.30}$$

donde M es la matriz de $n \times 2n$

$$M_{ij} = \sqrt{\frac{2}{n}} \cos \frac{(i+\frac{1}{2})(j+\frac{n}{2}+\frac{1}{2})\pi}{n} \tag{11.31}$$

para $0 \le i \le n-1$ y $0 \le j \le 2n-1$. ❐

Observe la gran diferencia respecto a las formas anteriores de la TDC: la TDCM de un vector con longitud $2n$ es un vector de longitud n. Por esta razón, la TDCM no es invertible directamente, pero se verá más adelante que puede lograrse el mismo efecto al superponer los vectores de longitud $2n$.

Al comparar con la definición de 11.9, es posible escribir la matriz M de la TDCM en términos de las columnas de la TDC4, para después simplificar utilizando el lema 11.10:

$$\begin{aligned} M &= \left[c_{\frac{n}{2}} \cdots c_{\frac{5}{2}n-1} \right] \\ &= \left[c_{\frac{n}{2}} \cdots c_{n-1} | c_n \cdots c_{\frac{3}{2}n-1} | c_{\frac{3}{2}n} \cdots c_{2n-1} | c_{2n} \cdots c_{\frac{5}{2}n-1} \right] \end{aligned}$$

$$= \left[c_{\frac{n}{2}} \cdots c_{n-1} | - c_{n-1} \cdots - c_{\frac{n}{2}} | - c_{\frac{n}{2}-1} \cdots - c_0 | - c_0 \cdots - c_{\frac{n}{2}-1} \right]. \tag{11.32}$$

Por ejemplo, la matriz de la TDCM con $n = 4$ es

$$M = [c_2 c_3 | c_4 c_5 | c_6 c_7 | c_8 c_9] = [c_2 c_3 | - c_3 - c_2 | - c_1 - c_0 | - c_0 - c_1]$$

Para simplificar la notación, sean A y B las mitades izquierda y derecha de la matriz TDC4, de modo que $E = [A|B]$. Defina la matriz de permutación formada al invertir las columnas de la matriz identidad, izquierda por derecha:

$$R = \begin{bmatrix} & & 1 \\ & \cdot & \\ \cdot & & \\ 1 & & \end{bmatrix}.$$

La matriz de permutación R invierte las columnas de la derecha por la izquierda al multiplicar una matriz desde la derecha. Si se multiplica desde la izquierda, los renglones se invierten de arriba a abajo. Tenga en cuenta que R es una matriz ortogonal simétrica, puesto que $R^{-1} = R^T = R$. Ahora (11.32) puede escribirse de manera más simple como

$$M = (B| - BR| - AR| - A), \tag{11.33}$$

donde AR y BR son versiones de A y B en las que el orden de las columnas ha sido invertido, de izquierda a derecha.

La acción de la TDCM puede expresarse en términos de la TDC4. Sea

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

un vector de $2n$, donde cada x_i es un vector de longitud $n/2$ (recuerde que n es par). Entonces, por la caracterización de M en (11.33),

$$\begin{aligned} Mx &= Bx_1 - BRx_2 - ARx_3 - Ax_4 \\ &= [A|B] \begin{bmatrix} -Rx_3 - x_4 \\ x_1 - Rx_2 \end{bmatrix} = E \begin{bmatrix} -Rx_3 - x_4 \\ x_1 - Rx_2 \end{bmatrix}, \end{aligned} \tag{11.34}$$

donde E es la matriz TDC4 de $n \times n$, y Rx_2 y Rx_3 representan a x_2 y x_3 con las entradas invertidas de arriba a abajo. Esto es muy útil (es posible expresar la salida de M en términos de una matriz ortogonal E).

Como la matriz M de $n \times 2n$ de la TDCM no es una matriz cuadrada, no es invertible. Sin embargo, dos TDCM adyacentes pueden tener un rango total de $2n$, y al unirlas es posible reconstruir perfectamente los valores x de entrada, como se muestra a continuación.

La TDCM "inversa" se representa mediante la matriz $N = M^T$ de $2n \times n$, que tiene las entradas transpuestas

$$N_{ij} = \sqrt{\frac{2}{n}} \cos \frac{(j + \frac{1}{2})(i + \frac{n}{2} + \frac{1}{2})\pi}{n}. \tag{11.35}$$

No es una inversa real, aunque es lo más cercano que puede obtenerse para una matriz rectangular. Al trasponer (11.33), se tiene

$$N = \begin{bmatrix} B^T \\ -RB^T \\ -RA^T \\ -A^T \end{bmatrix}, \tag{11.36}$$

usando la notación anterior $E = [A|B]$ para la transformada discreta del coseno TDC4. Se sabe que, como E es una matriz ortogonal,

$$A^T A = I$$
$$B^T B = I$$
$$A^T B = B^T A = 0,$$

donde I indica la matriz identidad de $n \times n$.

Ahora es posible calcular NM, para ver en qué sentido N invierte a la matriz M de la TDCM. Considere que x se divide en cuatro partes, como antes. De acuerdo con (11.34) y (11.36), la ortogonalidad de A y B, y el hecho de que $R^2 = I$, se tiene

$$NM\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} B^T \\ -RB^T \\ -RA^T \\ -A^T \end{bmatrix} [A(-Rx_3 - x_4) + B(x_1 - Rx_2)]$$
$$= \begin{bmatrix} x_1 - Rx_2 \\ -Rx_1 + x_2 \\ x_3 + Rx_4 \\ Rx_3 + x_4 \end{bmatrix}. \tag{11.37}$$

En los algoritmos de compresión de audio, la TDCM se aplica a los vectores de datos superpuestos. La razón es que los defectos sonoros debidos a los extremos de los vectores se producirán con una frecuencia fija, por la longitud constante del vector. El sistema auditivo es aún más sensible a los errores periódicos que el sistema visual; después de todo, un error de frecuencia fija es un tono de esa frecuencia, el cual puede ser captado por el oído porque está diseñado para ello. Suponga que los datos se presentarán de manera superpuesta. Sean

$$Z_1 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \text{y} \quad Z_2 = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

dos vectores de $2n$ para un entero par n, donde cada x_i es un vector de longitud $n/2$. Los vectores Z_1 y Z_2 se superponen en la mitad de su longitud. Como (11.37) muestra que

$$NMZ_1 = \begin{bmatrix} x_1 - Rx_2 \\ -Rx_1 + x_2 \\ x_3 + Rx_4 \\ Rx_3 + x_4 \end{bmatrix} \quad \text{y} \quad NMZ_2 = \begin{bmatrix} x_3 - Rx_4 \\ -Rx_3 + x_4 \\ x_5 + Rx_6 \\ Rx_5 + x_6 \end{bmatrix}, \tag{11.38}$$

el vector $[x_3, x_4]$ de longitud n puede reconstruirse con exactitud promediando la mitad inferior de NMZ_1 y la mitad superior de NMZ_2:

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \frac{1}{2}(NMZ_1)_{n,\ldots,2n-1} + \frac{1}{2}(NMZ_2)_{0,\ldots,n-1}. \tag{11.39}$$

Esta igualdad indica la forma en que se usa N para decodificar la señal después de haber sido codificada mediante M.

Este resultado se resume en el teorema 11.12.

TEOREMA 11.12 **Inversión de la TDCM a través de la superposición.** Sean M la matriz de la TDCM de $n \times 2n$ y $N = M^T$. Si u_1, u_2, u_3 son vectores de longitud n, y se establece

$$v_1 = M\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \text{y} \quad v_2 = M\begin{bmatrix} u_2 \\ u_3 \end{bmatrix}.$$

Entonces los vectores de longitud n, w_1, w_2, w_3, w_4, definidos por

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = Nv_1 \quad \text{y} \quad \begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = Nv_2$$

satisfacen $u_2 = \frac{1}{2}(w_2 + w_3)$. ■

Ésta es una reconstrucción exacta. El teorema 11.12 se utiliza por lo regular con una señal larga de vectores concatenados de longitud n $[u_1, u_2, \ldots, u_m]$. La TDCM se aplica a pares adyacentes para obtener una señal transformada $(v_1, v_2, \ldots, v_{m-1})$. Ahora, la compresión con pérdida entra en juego. Los v_i son componentes de frecuencia, por lo que puede optarse por mantener ciertas frecuencias y quitarle el énfasis a otras. En la siguiente sección se tomará esta dirección.

Después de reducir el contenido del v_i mediante la cuantificación u otros medios, es posible descomprimir $(u_2, \ldots, u_{m-1})$ a través del teorema 11.12. Tenga en cuenta que u_1 y u_m no pueden recuperarse; deben ser partes poco importantes de la señal o bien información de relleno que se añade de antemano.

► EJEMPLO 11.9 Use la TDCM superpuesta para transformar la señal $x = [1, 2, 3, 4, 5, 6]$. Después invierta la transformada para reconstruir la sección media [3, 4].

Se superpondrán los vectores [1, 2, 3, 4] y [3, 4, 5, 6]. Sea $n = 2$ y establezca

$$E_2 = \begin{bmatrix} \cos\frac{\pi}{8} & \cos\frac{3\pi}{8} \\ \cos\frac{3\pi}{8} & \cos\frac{9\pi}{8} \end{bmatrix} = \begin{bmatrix} b & c \\ c & -b \end{bmatrix}.$$

Observe que las definiciones de b y c han cambiado un poco respecto a (11.7) para que sean compatibles con la TDCM. Al aplicar la TDCM de 2×4 se obtiene

$$v_1 = M\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = E_2\begin{bmatrix} -R(3) - 4 \\ 1 - R(2) \end{bmatrix} = E_2\begin{bmatrix} -7 \\ -1 \end{bmatrix} = \begin{bmatrix} -7b - c \\ b - 7c \end{bmatrix} = \begin{bmatrix} -6.8498 \\ -1.7549 \end{bmatrix}$$

$$v_2 = M\begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} = E_2\begin{bmatrix} -R(5) - 6 \\ 3 - R(4) \end{bmatrix} = E_2\begin{bmatrix} -11 \\ -1 \end{bmatrix} = \begin{bmatrix} -11b - c \\ b - 11c \end{bmatrix} = \begin{bmatrix} -10.5454 \\ -3.2856 \end{bmatrix}.$$

La señal transformada se representa mediante

$$[v_1|v_2] = \begin{bmatrix} -6.8498 & -10.5454 \\ -1.7549 & -3.2856 \end{bmatrix}.$$

Para invertir la TDCM, defina A y B como

$$E_2 = \left[A\middle|B\right] = \left[\begin{array}{c|c} b & c \\ c & -b \end{array}\right]$$

y calcule

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = Nv_1 = \begin{bmatrix} B^T v_1 \\ -RB^T v_1 \\ -RA^T v_1 \\ -A^T v_1 \end{bmatrix} = \begin{bmatrix} c & -b \\ -c & b \\ -b & -c \\ -b & -c \end{bmatrix}\begin{bmatrix} -7b - c \\ b - 7c \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 7 \\ 7 \end{bmatrix}$$

$$\begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = Nv_2 = \begin{bmatrix} B^T v_2 \\ -RB^T v_2 \\ -RA^T v_2 \\ -A^T v_2 \end{bmatrix} = \begin{bmatrix} c & -b \\ -c & b \\ -b & -c \\ -b & -c \end{bmatrix}\begin{bmatrix} -11b - c \\ b - 11c \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 11 \\ 11 \end{bmatrix},$$

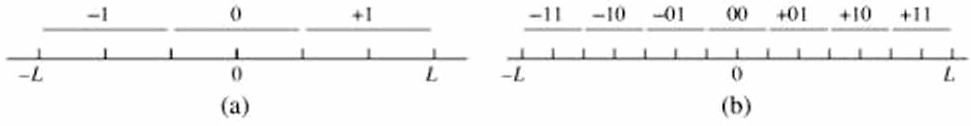


Figura 11.11 Cuantificación de bits. Ilustración de (11.39). (a) 2 bits (b) 3 bits.

donde se ha utilizado el hecho $b^2 + c^2 = 1$. El resultado del teorema 11.12 es que puede recuperarse la superposición [3, 4] mediante

$$u_2 = \frac{1}{2}(w_2 + w_3) = \frac{1}{2}\left(\begin{bmatrix} 7 \\ 7 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 3 \\ 4 \end{bmatrix}. \quad \blacktriangleleft$$

La definición y el uso de la TDCM son menos directos que el uso de la TDC, que se analizó antes en este capítulo. Su ventaja es que permite la superposición de vectores adyacentes de una manera eficiente. El efecto es promediar las contribuciones de dos vectores, reduciendo los defectos sonoros de las transiciones abruptas que se producen en los extremos. Como en el caso de la TDC, es posible filtrar o cuantificar los coeficientes de la transformada antes de reconstruir la señal, a fin de mejorarla o comprimirla. A continuación se muestra cómo puede utilizarse la TDCM en la compresión al añadir un paso de cuantificación.

11.4.2 Cuantificación de bits

La compresión con pérdida de señales de audio se consigue mediante la cuantificación de la salida de la TDCM de una señal. En esta sección se expandirá la cuantificación utilizada para la compresión de imágenes, a fin de tener más control sobre el número de bits utilizados para representar la versión con pérdidas de la señal.

Comience con el intervalo abierto de números reales $(-L, L)$. Suponga que el objetivo es representar un número en $(-L, L)$ por medio de b bits y que se tiene tolerancia a un pequeño error. Se utilizará un bit para el signo y se cuantificará en un entero binario de $b - 1$ bits. La fórmula es:

Cuantificación de $(-L, L)$ con b bits

$$\text{Cuantificación: } z = \text{redondear}\left(\frac{y}{q}\right), \text{ donde } q = \frac{2L}{2^b - 1}$$

$$\text{Descuantificación: } \overline{y} = qz \tag{11.40}$$

Como ejemplo, se muestra la forma de representar los números en el intervalo $(-1, 1)$ mediante 4 bits. Establezca $q = 2(1)/(2^4 - 1) = 2/15$, y cuantifique por medio de q. El número $y = -0.3$ se representa como

$$\frac{-0.3}{2/15} = -\frac{9}{4} \longrightarrow -2 \longrightarrow -010,$$

y el número $y = 0.9$ se representa por

$$\frac{0.9}{2/15} = \frac{27}{4} = 6.75 \longrightarrow 7 \longrightarrow +111.$$

La descuantificación invierte el proceso. La versión cuantificada de -0.3 se descuantifica como

$$(-2)q = (-2)(2/15) = -4/15 \approx -0.2667$$

y la versión cuantificada de 0.9 como

$$(7)q = (7)(2/15) = 14/15 \approx 0.9333.$$

En ambos casos, el error de cuantificación es 1/30.

► **EJEMPLO 11.10** Cuantifique la salida de la TDCM del ejemplo 11.9 en enteros de 4 bits. Después descuantifique, invierta la TDCM y encuentre el error de cuantificación.

Todas las entradas de la transformada se encuentran en el intervalo $(-12, 12)$. Si se usa $L = 12$, la cuantificación de cuatro bits requiere $q = 2(12)/(2^4 - 1) = 1.6$. Entonces

$$v_1 = \begin{bmatrix} -6.8498 \\ -1.7549 \end{bmatrix} \longrightarrow \begin{bmatrix} \text{redondear } (\frac{-6.8948}{1.6}) \\ \text{redondear } (\frac{-1.7549}{1.6}) \end{bmatrix} \longrightarrow \begin{bmatrix} -4 \\ -1 \end{bmatrix} \longrightarrow \begin{matrix} -100 \\ -001 \end{matrix}$$

y

$$v_2 = \begin{bmatrix} -10.5454 \\ -3.2856 \end{bmatrix} \longrightarrow \begin{bmatrix} \text{redondear } (\frac{-10.5454}{1.6}) \\ \text{redondear } (\frac{-3.2856}{1.6}) \end{bmatrix} \longrightarrow \begin{bmatrix} -7 \\ -2 \end{bmatrix} \longrightarrow \begin{matrix} -111 \\ -010 \end{matrix}.$$

Las variables de la transformada v_1, v_2 pueden almacenarse como cuatro números enteros de 4 bits, para un total de 16 bits.

La descuantificación con $q = 1.6$ es

$$\begin{bmatrix} -4 \\ -1 \end{bmatrix} \longrightarrow \begin{bmatrix} -6.4 \\ -1.6 \end{bmatrix} = \bar{v}_1$$

y

$$\begin{bmatrix} -7 \\ -2 \end{bmatrix} \longrightarrow \begin{bmatrix} -11.2 \\ -3.2 \end{bmatrix} = \bar{v}_2.$$

Al aplicar la inversa de la TDCM se obtiene

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = N\bar{v}_1 = \begin{bmatrix} -0.9710 \\ 0.9710 \\ 6.5251 \\ 6.5251 \end{bmatrix},$$

$$\begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = N\bar{v}_2 = \begin{bmatrix} -1.3296 \\ 1.3296 \\ 11.5720 \\ 11.5720 \end{bmatrix},$$

y la señal reconstruida

$$u_2 = \frac{1}{2}(w_2 + w_3) = \frac{1}{2}\left(\begin{bmatrix} 6.5251 \\ 6.5251 \end{bmatrix} + \begin{bmatrix} -1.3296 \\ 1.3296 \end{bmatrix}\right) = \begin{bmatrix} 2.5977 \\ 3.9274 \end{bmatrix}.$$

El error de cuantificación es la diferencia entre las señales original y reconstruida:

$$\left\| \begin{bmatrix} 2.5977 \\ 3.9274 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} \right| = \begin{bmatrix} 0.4023 \\ 0.0726 \end{bmatrix}.$$ ◄

La codificación de archivos de audio se realiza por lo general usando una asignación de bits preestablecida para rangos de frecuencia prescritos. La comprobación en la realidad 11 guía al lector a través de la construcción de un **codec** completo, o protocolo de codificación y decodificación, que utiliza la TDCM junto con la cuantificación de bits.

11.4 Ejercicios

1. Encuentre la TDCM de la entrada. Exprese la respuesta en términos de $b = \cos\pi/8$ y $c = \cos 3\pi/8$. (a) [1, 3, 5, 7] (b) [−2, −1, 1, 2] (c) [4, −1, 3, 5]
2. Encuentre la TDCM de las dos ventanas de longitud 4 superpuestas de la entrada dada, como en el ejemplo 11.9. Después reconstruya la sección media, usando la TDCM inversa. (a) [−3, −2, −1, 1, 2, 3] (b) [1, −2, 2, −1, 3, 0] (c) [4, 1, −2, −3, 0, 3]
3. Cuantifique cada número real en (−1, 1) a 4 bits, y después descuantifique y calcule el error de cuantificación. (a) 2/3 (b) 0.6 (c) 3/7
4. Repita el ejercicio 3, pero cuantifique a 8 bits.
5. Cuantifique cada número real en (−4, 4) a 8 bits, y después descuantifique y calcule el error de cuantificación. (a) 3/2 (b) −7/5 (c) 2.9 (d) π
6. Demuestre que la matriz de la TDC4 de $n \times n$ es una matriz ortogonal para cada n entero par.
7. Reconstruya la sección media de los datos del ejercicio 2 después de cuantificar a 4 bits en (−6, 6). Compare su respuesta con la sección media correcta.
8. Reconstruya la sección media de los datos del ejercicio 2 después de cuantificar a 6 bits en (−6, 6). Compare su respuesta con la sección media correcta.
9. Explique por qué el vector de columna n-dimensional c_k definido por (11.28) para cualquier entero k puede expresarse en términos de una columna $c_{k'}$ para $0 \leq k' \leq n - 1$. Exprese c_{5n} y c_{6n} de esta manera.
10. Encuentre una cota superior para el error de cuantificación (el error causado por la cuantificación, seguida por la descuantificación) al convertir un número real en un entero de b bits en el intervalo $(-L, L)$.

11.4 Problemas de computadora

1. Escriba un programa en MATLAB para aceptar un vector como entrada, aplicar la TDCM para cada una de las ventanas de longitud $2n$ y reconstruir las n secciones de longitud superpuesta, como en el ejemplo 11.9. Demuestre que funciona sobre las siguientes señales de entrada, (a) $n = 4$, x = [1 2 3 4 5 6 7 8 9 10 11 12] (b) $n = 4$, $x_i = \cos(i\pi/6)$ para $i = 0,\ldots, 11$ (c) $n = 8$, $x_i = \cos(i\pi/10)$ para $i = 0, \ldots, 63$
2. Adapte su programa del problema de computadora 1 para aplicar la cuantificación a b bits antes de reconstruir las superposiciones. Después reconstruya los ejemplos de ese problema, y calcule los errores de reconstrucción al comparar cada resultado con la entrada original.

Comprobación en la realidad 11

Un codec de audio simple

La transmisión y el almacenamiento eficaces de archivos de audio es una parte clave de la comunicación moderna, y el papel desempeñado por la compresión es crucial. En esta comprobación en la realidad se presentará un breve protocolo para compresión y descompresión basado en la capacidad de la TDCM para dividir la señal de audio en sus componentes de frecuencia y el método de la cuantificación de bits analizado en la sección 11.4.2.

La TDCM se aplica a una ventana de entrada de $2n$ valores de señal y proporciona una salida de n componentes de frecuencia que se aproximan a los datos (y junto con la ventana siguiente, interpola los últimos n puntos de entrada). La parte de compresión del algoritmo consiste en codificar los componentes de la frecuencia después de la cuantificación para ahorrar espacio, como se demostró en el ejemplo 11.10.

En los formatos comunes de almacenamiento de audio, la forma en que los bits se asignan a los diferentes componentes de la frecuencia durante la cuantificación se basa en la **psicoacústica**, la ciencia de la percepción humana del sonido. Se usan técnicas como el **enmascaramiento de frecuencias**, el hecho empírico de que el oído puede manejar sólo un sonido dominante en cada rango de frecuencia en un momento dado, para decidir qué componentes de la frecuencia son más o menos importantes de preservar. A los componentes más importantes se les asignan más bits de cuantificación. Los métodos más competitivos se basan en la TDCM y difieren en la forma en que se tratan los factores psicoacústicos. En la descripción de este libro se tomará un enfoque simplificado que no toma en cuenta la mayoría de los factores psicoacústicos y se basa sólo en el **filtrado por importancia**, la tendencia de asignar más bits a los componentes de la frecuencia con mayor magnitud.

Se inicia con la reconstrucción de un tono puro. Si se establece $n = 32$, la frecuencia de tono inferior catalogada por la TDCM es de 64 Hz, en el borde inferior de las frecuencias perceptibles para el oído humano. Un tono puro de 64 Hz se representa mediante $x(t) = \cos 2\pi(64)t$, donde t se mide en segundos. Si F_s es el número de muestras por segundo, entonces $1/F_s, 2/F_s, \ldots, F_s/F_s$ representan un segundo valor de los puntos en el tiempo. Los comandos de MATLAB

```
Fs=8192;
x=cos(2*pi*64*(1:Fs)/Fs);
sound(x,Fs)
```

reproducen un segundo de un tono de 64 Hz. La frecuencia de muestreo `Fs` de $8192 = 2^{13}$ bytes/s es bastante común, y corresponde a $2^{16} = 65536$ bits/s, que se refiere como una tasa de muestreo de 64 Kb/s para un archivo de audio. (Los archivos de mayor calidad suelen muestrearse con tasas hasta dos o tres veces superiores a ésta, es decir 128 o 192 Kbs).

Los tonos más altos se obtienen al sustituir 64 por un múltiplo entero $64f$. Al establecer $f = 2$ o 4 se obtienen versiones en octavas superiores. Con $f = 7$, se reproduce un tono de 448 Hz, lo suficientemente alejado del concierto A (440 Hz) que si usted tiene amigos con un oído natural, esto debe distraerlos muy rápido.

El siguiente fragmento de código en MATLAB aplica la TDCM y cuantifica, seguida por una descuantificación inmediata y la TDCM inversa en los segmentos superpuestos, como se describe en la sección 11.4. De esta manera, puede examinarse el efecto del error de cuantificación que acompaña a la compresión con pérdida.

```
n=32;                              % longitud de ventana
nb=127;                            % número de ventanas; debe ser > 1
b=4; L=5;                          % información de cuantificación
q=2*L/(2^b-1);                     % b bits en el intervalo [-L, L]
for i=1:n                          % forma la matriz TDCM
  for j=1:2*n
    M(i,j)= cos((i-1+1/2)*(j-1+1/2+n/2)*pi/n);
  end
end
M=sqrt(2/n)*M;
N=M';                              % TDCM inversa
Fs=8192;f=7;                       % Fs= tasa de muestreo
x=cos((1:4096)*pi*64*f/4096);      % señal de prueba
sound(x,Fs)                        % comando de sonido de Matlab
out=[];
for k=1:nb                         % ciclo en las ventanas
  x0=x(1+(k-1)*n:2*n+(k-1)*n)';
  y0=M*x0;
  y1=round(y0/q);                  % transforma los componentes cuantificados
  y2=y1*q;                         % y descuantificados
  w(:,k)=N*y2;                     % invierte la TDCM
```

```
  if(k>1)
     w2=w(n+1:2*n,k-1);w3=w(1:n,k);
     out=[out;(w2+w3)/2];        % recoger la señal reconstruida
  end
end
pause(1)
sound(out,Fs)                  % reproduce el tono reconstruido
```

El código reproduce el tono original de 1/2 segundo (448 Hz), seguido por el tono reconstruido. Compare el efecto de cambiar el número de bits que representan a los componentes de la transformada, dado por la variable `b` en el código.

Actividades sugeridas:

1. ¿Cuál es la diferencia de la salida de la TDCM si se usan valores enteros impares de f, en comparación con el caso de los valores pares de f? Explique por qué el número de bits necesarios para que el sonido reconstruido sea similar al original difiere para los valores pares e impares de f.

2. Añada una "función de ventana" al código. La función de ventana escala uniformemente la señal de entrada x hasta cero en cada extremo de la ventana, para contrarrestar el problema de que la señal no es con exactitud periódica. Una opción común es reemplazar x_i por $x_i h_i$, donde

$$h_i = \sqrt{2}\,\text{sen}\,\frac{(i-\frac{1}{2})\pi}{2n}$$

   para una ventana de longitud $2n$. Para anular la función de ventana, multiplique las salidas de la TDCM inversa w_2 y w_3 componente a componente por la misma h_i; lo anterior utiliza la ortogonalidad del seno, puesto que la función de ventana está ahora desviada en 1/4 de periodo. Compare el efecto de la función de ventana en el número de bits necesarios para reconstruir el tono de manera correcta.

3. Introduzca el muestreo de importancia. Haga un nuevo tono de prueba que sea una combinación de tonos puros. Modifique el código para que cada uno de los 32 componentes de la frecuencia de y tenga su propio número b_k de bits para la cuantificación. Proponga un método que haga a b_k más grande si las contribuciones $|y_k|$ son más grandes, en promedio. Cuente el número de bits necesarios para mantener la señal y mejore su propuesta.

4. Construya dos subprogramas separados, un codificador y un decodificador. El codificador debe escribir un archivo (o variable de MATLAB) de los bits que representan la salida cuantificada de la MDCT e imprimir el número de bits utilizados. El decodificador debe cargar el archivo escrito por el codificador y reconstruir la señal.

5. Descargue un archivo `.wav` con el comando `wavread` de MATLAB o descargue otro archivo de audio de su elección. (Como alternativa, se puede utilizar `handel`. Si utiliza un archivo estéreo, deberá trabajar con cada canal por separado). Proponga e implemente un método para determinar la mejor asignación de bits, representada por b_k. Utilice el codificador para comprimir el archivo de audio y del decodificador para descomprimir. Compare la calidad del sonido para resultados diferentes, donde se hayan logrado cantidades de compresión distintas.

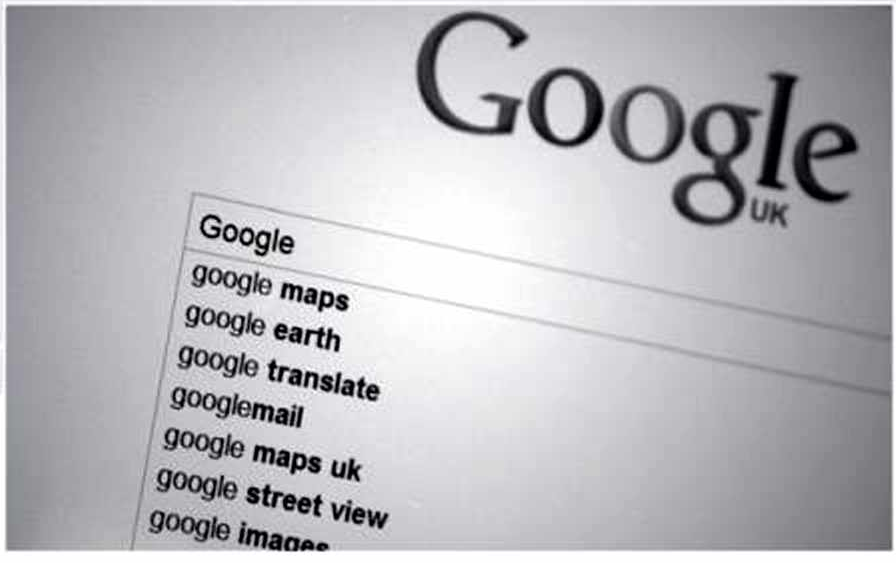
6. Investigue otros trucos que utilice la industria del sonido para hacer una compresión más eficiente. Por ejemplo, en el caso de un archivo de audio estéreo, ¿existe un mejor método que tratar los canales s_1 y s_2 por separado?, ¿por qué podría ser más ventajoso comprimir $(s_1 + s_2)/2$ y $(s_1 - s_2)/2$? ✔

Software y lecturas adicionales

Para leer buenas introducciones prácticas a la compresión de datos, consulte Nelson y Gailly [1995], Storer [1988] y Sayood [1996]. Algunas referencias generales sobre la compresión de imagen y sonido son Bhaskaran y Konstandtinides [1995]. Rao y Yip [1990] es una buena fuente para obtener información sobre la transformada discreta del coseno. El artículo de gran influencia sobre la codificación de Huffman es Huffman [1952].

En este libro se ha presentado el estándar JPEG de referencia (Wallace [1991]) para la compresión de imágenes. El estándar completo está disponible en Pennebaker y Mitchell [1993]. El estándar JPEG-2000 recientemente presentado (Taubman y Marcellin [2002]) permite la compresión de ondoletas en vez de la TDC.

La mayoría de los protocolos para la compresión de sonido se basan en la transformada discreta del coseno modificada (Wang y Vilermo [2003], Malvar [1992]). Puede encontrarse información más específica en formatos individuales como MP3 (abreviatura de MPEG Audio Layer 3, vea Hacker [2000]), AAC (Advanced Audio Coding, que se utiliza en el iTunes de Apple, el video QuickTime y la radio satelital XM) y el formato de audio de fuente abierta Ogg-Vorbis.



CAPÍTULO

12

Valores y vectores característicos y valores singulares

La World Wide Web hace que cualquier usuario pueda tener un fácil acceso a grandes cantidades de información (de hecho, esta información es tan vasta, que resulta esencial navegar con un potente motor de búsqueda). La tecnología también ha proporcionado la miniaturización y sensores de bajo costo, lo que permite que grandes cantidades de datos estén a disposición de las personas. ¿Cómo puede explotarse, de manera eficaz, este acceso a grandes cantidades de información?

Muchos aspectos de la tecnología de búsqueda, y descubrimiento del conocimiento en general, se benefician de su tratamiento como un problema de valor propio (o característico) o valor singular. Los métodos numéricos para resolver estos problemas de alta dimensión generan proyecciones a subespacios dimensionales inferiores. Ésta es exactamente la simplificación que es requerida en la mayoría de los entornos de datos complejos.

Comprobación en la realidad En la página 549 se explora lo que se ha llamado el mayor cálculo en el desarrollo actual de un valor propio (o valor característico) en el mundo, utilizado por uno de los más renombrados proveedores de búsqueda en la web.

Los métodos computacionales para localizar valores propios se basan en la idea fundamental de la iteración de potencia, un tipo de iteración de punto fijo para espacios propios. Una versión sofisticada de la idea, llamada el algoritmo QR, es el algoritmo estándar para determinar todos los valores propios de las matrices típicas.

La descomposición del valor singular revela la estructura básica de una matriz y es muy usada en aplicaciones estadísticas para encontrar las relaciones entre los datos. En este capítulo se analizan los métodos para encontrar los valores y vectores propios (característicos) de una matriz cuadrada, así como los valores y los vectores singulares de una matriz general.

12.1 MÉTODOS DE ITERACIÓN DE POTENCIA

No existe un método directo para calcular valores propios (o característicos). La situación es similar a encontrar raíces, en la que todos los métodos posibles dependen de algún tipo de iteración. Para comenzar en esta sección se estudiará si el problema puede reducirse a la localización de raíces.

En el apéndice A se muestra un método para calcular los valores y los vectores propios (característicos) de una matriz de $m \times m$. Este enfoque, basado en la localización de raíces del polinomio característico de grado m, funciona bien para las matrices de 2×2. Para las matrices grandes, el procedimiento requiere un método para determinar raíces de los estudiados en el capítulo 1.

La dificultad de este método para encontrar valores propios (característicos) se hace evidente al recordar el ejemplo del polinomio de Wilkinson del capítulo 1. Se encontró que cambios muy pequeños en los coeficientes de un polinomio pueden cambiar las raíces del polinomio en cantidades demasiado grandes. En otras palabras, el número de condición del problema de entrada-salida que convierte los coeficientes en raíces puede ser extremadamente grande. Debido a que el cálculo de los coeficientes del polinomio característico estará sujeto a errores del orden del redondeo de la máquina o mayores, el cálculo de los valores propios mediante este método es susceptible a grandes errores. Esta dificultad es muy grave como para justificar la eliminación del método para encontrar raíces del polinomio característico como una vía para el cálculo exacto de valores propios (característicos).

Un ejemplo simple de la poca precisión de este método se desprende de la existencia del polinomio de Wilkinson. Si se desea encontrar los valores propios de la matriz

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 2 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 20 \end{bmatrix}, \tag{12.1}$$

se calcularán los coeficientes del polinomio característico $P(x) = (x-1)(x-2)\cdots(x-20)$ y se utilizará un método para determinar raíces. Sin embargo, como se muestra en el capítulo 1, algunas de las raíces de la versión de máquina de $P(x)$ están lejos de las raíces de la versión verdadera de $P(x)$, que son los valores propios de A.

En esta sección se presentan los métodos basados en la multiplicación reiterada de la matriz por un vector, que al hacerlo n veces se convierte en el vector propio. La idea se perfeccionará más adelante, pero es el fundamento principal de los métodos más sofisticados.

12.1.1 Iteración de potencia

La motivación detrás de la iteración de potencia es que la multiplicación por una matriz tiende a mover los vectores hacia la dirección del vector propio dominante.

ANOTACIÓN

Condicionamiento Los grandes errores a los que está sujeto el "método del polinomio característico" no son culpa del método para encontrar las raíces. Un método perfectamente exacto no tendrá un mejor desempeño. Cuando el polinomio se multiplica para determinar los coeficientes de entrada en el método para localizar las raíces, en general, los coeficientes estarán sujetos a errores en el orden del épsilon de la máquina. Entonces, se le pedirá al método que encuentre las raíces de un polinomio que es un poco erróneo, lo cual, como se ha visto, puede tener consecuencias desastrosas. No hay una solución general a este problema y la única manera de combatirlo sería aumentar el tamaño de la mantisa que representa los números de punto flotante, lo que tendría el efecto de bajar el épsilon de la máquina. Si el épsilon de la máquina pudiera hacerse menor que 1/cond(P), entonces podría asegurarse la precisión para los valores propios. Por supuesto, esto no es en realidad una solución, sino sólo un paso más en una carrera armamentista imposible de ganar. Si se utiliza mayor precisión de cálculo, siempre es posible extender el polinomio de Wilkinson a un mayor grado para encontrar un número de condición incluso más alto.

DEFINICIÓN 12.1 Sea A una matriz de $m \times m$. Un **valor propio** (característico) **dominante** de A es un valor propio λ, cuya magnitud es mayor que todos los otros valores propios de A. Si existe, un vector propio asociado a λ se llama **vector propio** (característico) **dominante**. ❐

La matriz

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}$$

tiene un valor propio dominante de 4 con vector propio $[1, 1]^T$, y un vector propio que es de menor magnitud, -1, con un vector propio asociado $[-3, 2]^T$. Observe los resultados de multiplicar la matriz A por un vector "aleatorio", por ejemplo $[-5, 5]^T$:

$$x_1 = Ax_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} -5 \\ 5 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$
$$x_2 = A^2x_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \end{bmatrix}$$
$$x_3 = A^3x_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 10 \\ 20 \end{bmatrix} = \begin{bmatrix} 70 \\ 60 \end{bmatrix}$$
$$x_4 = A^4x_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 70 \\ 60 \end{bmatrix} = \begin{bmatrix} 250 \\ 260 \end{bmatrix} = 260\begin{bmatrix} \frac{25}{26} \\ 1 \end{bmatrix}$$

La multiplicación reiterada de un vector de inicio aleatorio por la matriz ha dado como resultado el desplazamiento del vector a un punto muy cercano del vector propio dominante de A. Esto no es casual, como puede verse al expresar x_0 como una combinación lineal de los vectores propios

$$x_0 = 1\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$

y al revisar el cálculo bajo este conocimiento:

$$x_1 = Ax_0 = 4\begin{bmatrix} 1 \\ 1 \end{bmatrix} - 2\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$
$$x_2 = A^2x_0 = 4^2\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$
$$x_3 = A^3x_0 = 4^3\begin{bmatrix} 1 \\ 1 \end{bmatrix} - 2\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$
$$x_4 = A^4x_0 = 4^4\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2\begin{bmatrix} -3 \\ 2 \end{bmatrix}$$
$$= 256\begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2\begin{bmatrix} -3 \\ 2 \end{bmatrix}.$$

El punto es que el vector propio correspondiente al valor propio más grande en magnitud dominará el cálculo después de varios pasos. En este caso, el valor propio 4 es el más grande, por lo que el cálculo se desplaza más y más hacia un vector propio en su dirección $[1, 1]^T$.

Para evitar que los números se salgan de control, es necesario normalizar el vector a cada paso. Una forma de hacer esto es dividir el vector actual entre el mayor valor del vector (normalizar el vector resultante) antes de cada paso. Las dos operaciones, la normalización y la multiplicación por A constituyen el método de iteración de potencia.

A medida que los pasos proporcionan mejores vectores propios aproximados, ¿cómo pueden encontrarse los valores propios aproximados? Para plantear la pregunta de manera más general, suponga que se conocen una matriz A y un vector propio aproximado. ¿Cuál es la mejor estimación para el valor propio asociado?

ANOTACIÓN

Convergencia La iteración de potencia es en esencia una iteración de punto fijo con normalización a cada paso. Al igual que la IPF, converge linealmente, lo que significa que durante la convergencia, el error disminuye en un factor constante a cada paso de la iteración. Más adelante en esta sección, se encontrará una variante cuadráticamente convergente de la iteración de potencia llamada iteración del cociente de Rayleigh.

Se recurrirá a los mínimos cuadrados. Considere la ecuación de valor propio $x\lambda = Ax$, donde x es un vector propio aproximado y λ es una incógnita. Vista de esta manera, la matriz de coeficientes es la matriz x de $n \times 1$. Las ecuaciones normales dicen que la respuesta de mínimos cuadrados es la solución de $x^T x\lambda = x^T Ax$, o

$$\lambda = \frac{x^T Ax}{x^T x}, \tag{12.2}$$

conocido como el **cociente de Rayleigh**. Dado un vector propio aproximado, el cociente de Rayleigh es el mejor valor propio aproximado. La aplicación del cociente de Rayleigh al vector propio normalizado añade una aproximación al valor propio para la iteración de potencia.

Iteración de potencia

Dado el vector inicial x_0.

for $j = 1, 2, 3, \ldots$
$u_{j-1} = x_{j-1}/||x_{j-1}||_2$
$x_j = Au_{j-1}$
$\lambda_j = u_{j-1}^T Au_{j-1}$
end
$u_j = x_j/||x_j||_2$

Para encontrar el vector propio dominante de la matriz A, comience con un vector cualquiera. Cada iteración consiste en normalizar el vector actual y multiplicar por A. El cociente de Rayleigh se utiliza para aproximar el valor propio. El comando `norm` de MATLAB facilita la implementación, como se muestra en el siguiente código:

```
% Programa 12.1 Iteración de potencia
% Calcula el vector propio dominante de la matriz cuadrada
% Entrada: matriz A, vector inicial x (distinto de cero), número de pasos k
% Salida: valor propio dominante lam, vector propio u
function [lam,u]=powerit(A,x,k)
for j=1:k
    u=x/norm(x);              % normaliza el vector
    x=A*u;                    % paso de potencia
    lam=u'*x;                 % cociente de Rayleigh
end
u=x/norm(x);
```

12.1.2 Convergencia de la iteración de potencia

Se demostrará la convergencia de la iteración de potencia bajo ciertas condiciones que deben cumplir los valores propios. Aunque estas condiciones no son por completo generales, sirven para demostrar por qué el método tiene éxito en el caso más claro posible. Después, se construirán métodos de valores propios que serán sucesivamente más sofisticados, con base en el concepto de la iteración de potencia, que cubre a las matrices más generales.

TEOREMA 12.2 Sea A una matriz de $m \times m$ con valores propios reales $\lambda_1, \ldots, \lambda_m$ que satisfacen $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_m|$. Suponga que los vectores propios de A abarcan R^m. Para casi cualquier vector inicial,

la iteración de potencia converge linealmente a un vector propio asociado a λ_1, con razón de convergencia constante $S = |\lambda_2/\lambda_1|$. ■

Demostración. Sean $v_1, \ldots, v_n$ los vectores propios que forman una base de R^n, con los valores propios correspondientes $\lambda_1, \ldots, \lambda_n$, respectivamente. Exprese el vector inicial en esta base como $x_0 = c_1v_1 + \cdots + c_nv_n$ para algunos coeficientes c_i. La frase "para casi cualquier vector inicial" significa que puede asumirse que $c_1, c_2 \neq 0$. Al aplicar la iteración de potencia se obtiene

$$\begin{aligned} Ax_0 &= c_1\lambda_1 v_1 + c_2\lambda_2 v_2 + \cdots + c_n\lambda_n v_n \\ A^2x_0 &= c_1\lambda_1^2 v_1 + c_2\lambda_2^2 v_2 + \cdots + c_n\lambda_n^2 v_n \\ A^3x_0 &= c_1\lambda_1^3 v_1 + c_2\lambda_2^3 v_2 + \cdots + c_n\lambda_n^3 v_n \\ &\vdots \end{aligned}$$

con normalización a cada paso. Cuando el número de pasos $k \to \infty$, dominará el primer término del lado derecho, sin importar cómo se realice la normalización, porque

$$\frac{A^k x_0}{\lambda_1^k} = c_1v_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k v_n.$$

El supuesto de que $|\lambda_1| > |\lambda_i|$ para $i > 1$ implica que todos los términos excepto el primero de la derecha convergen a cero con una razón de convergencia $S \leq |\lambda_2/\lambda_1|$, y exactamente a esa razón, mientras $c_2 \neq 0$. Como resultado, el método converge a un múltiplo del vector propio dominante v_1, con valor propio λ_1. ❐

El término "casi todos" en la conclusión del teorema significa que el conjunto de vectores iniciales x_0 para los que la iteración falla es un conjunto de dimensión pequeña en R^m. En específico, la iteración tendrá éxito a la razón especificada si x_0 no está contenido en la unión de los planos de dimensión $m - 1$ abarcados por $\{v_1, v_3, \ldots, v_m\}$ y $\{v_2, v_3, \ldots, v_m\}$.

12.1.3 Iteración de potencia inversa

La iteración de potencia se limita a localizar el valor propio de mayor magnitud (valor absoluto). Si la iteración de potencia se aplica a la inversa de la matriz, pueden encontrarse el valor propio más pequeño.

LEMA 12.3 Sean los valores propios de la matriz A de $m \times m$ indicados por $\lambda_1, \lambda_2, \ldots, \lambda_m$. (a) Los valores propios de la matriz inversa A^{-1} son $\lambda_1^{-1}, \lambda_2^{-1}, \ldots, \lambda_m^{-1}$, asumiendo que la inversa existe. Los vectores propios son iguales a los de A. (b) Los valores propios de la matriz desplazada $A - sI$ son $\lambda_1 - s$, $\lambda_2 - s$, ..., $\lambda_m - s$, y los vectores propios son iguales a los de A. ■

Demostración. (a) $Av = \lambda v$ implica que $v = \lambda A^{-1} v$ y, por lo tanto, $A^{-1} v = (1/\lambda)v$. Observe que el vector propio es invariable. (b) Reste sIv de ambos lados de $Av = \lambda v$. Entonces $(A - sI)$ $v = (\lambda - s)v$ es la definición del valor propio para $(A - sI)$, y de nuevo puede utilizarse el mismo vector propio. ❐

De acuerdo con el lema 12.3, el valor propio con la mayor magnitud en la matriz A^{-1} es el recíproco del valor propio con menor magnitud de A. Al aplicar la iteración de potencia a la matriz inversa, seguida de la inversión del valor propio resultante de A^{-1}, se obtiene el valor propio con menor magnitud de A.

Para evitar el cálculo explícito de la inversa de A, se reescribe la aplicación de la iteración de potencia sobre A^{-1}, a saber,

$$x_{k+1} = A^{-1}x_k \tag{12.3}$$

como el equivalente

$$Ax_{k+1} = x_k, \tag{12.4}$$

que después se resuelve para x_{k+1} por eliminación gaussiana.

Ahora se sabe cómo encontrar el mayor y el menor valor propio de una matriz. En otras palabras, para una matriz de 100×100, se ha realizado el 2 por ciento. ¿Cómo se encuentra el restante 98 por ciento?

El lema 12.3(b) sugiere un posible enfoque. Es posible disminuir cualquiera de los otros valores propios al desplazar A hacia un valor cercano al valor propio. Si se llega a saber que hay un valor propio cercano a 10 (por ejemplo, 10.05), entonces $A - 10I$ tiene un valor propio $\lambda = 0.05$. Si es la menor magnitud de valor propio de $A - 10I$, entonces la iteración de potencia inversa $x_{k+1} = (A - 10I)^{-1} x_k$ la localizará. Esto es, la iteración de potencia inversa converge al recíproco $1/(.05) = 20$, después de lo cual se invierte a .05 y se suma de nuevo el cambio para obtener 10.05. Este truco localizará el valor propio más pequeño después del desplazamiento (que es otra manera de decir el valor propio más cercano al desplazamiento). En resumen, se escribe

Iteración de potencia inversa

Dado el vector inicial x_0 y el desplazamiento s

for $j = 1, 2, 3, \ldots$
$u_{j-1} = x_{j-1}/||x_{j-1}||_2$
Solve $(A - sI)x_j = u_{j-1}$
$\lambda_j = u_{j-1}^T x_j$
end
$u_j = x_j/||x_j||_2$

Para encontrar el valor propio de A más cercano al número real s, aplique la iteración de potencia a $(A - sI)^{-1}$ para obtener el valor propio con mayor magnitud de b de $(A - sI)^{-1}$. Las iteraciones de potencia deben realizarse mediante la eliminación gaussiana sobre $(A - sI)y_{k+1} = x_k$. Entonces $\lambda = b^{-1} + s$ es el valor propio de A más cercano a s. El vector propio asociado a λ se da directamente en el cálculo.

```
% Programa 12.2 Iteración de potencia inversa
% Calcula el valor propio de la matriz cuadrada más cercano a la entrada s
% Entrada: matriz A, (distinta de cero) vector x, desplazamiento s, pasos k
% Salida: valor propio dominante lam, vector propio de inv(A-sI)
function [lam,u]=invpowerit(A,x,s,k)
As=A-s*eye(size(A));
for j=1:k
  u=x/norm(x);                  % normaliza el vector
  x=As\u;                       % paso de potencia
  lam=u'*x;                     % cociente de Rayleigh
end
lam=1/lam+s; u=x/norm(x);
```

► EJEMPLO 12.1 Suponga que A es una matriz de 5×5 con valores propios $-5, -2, 1/2, 3/2, 4$. Encuentre el valor propio y la razón de convergencia esperada cuando se aplica (a) la iteración de potencia (b) la iteración de potencia inversa con desplazamiento $s = 0$ (c) la iteración de potencia inversa con desplazamiento $s = 2$.

(a) La iteración de potencia con un vector inicial aleatorio convergerá al valor propio con mayor magnitud -5, con una razón de convergencia $S = |\lambda_2|/|\lambda_1| = 4/5$. (b) La iteración de potencia

inversa (sin desplazamiento) converge al más pequeño, 1/2, porque su recíproco 2 es mayor que los recíprocos de los demás $-1/5$, $-1/2$, 2/3 y 1/4. La razón de convergencia será la razón de los dos valores propios mayores de la matriz inversa, $S = (2/3)/2 = 1/3$. (c) La iteración de potencia inversa con desplazamiento $s = 2$ localizará el valor propio más cercano a 2, que es 3/2. La razón es que, después de cambiar los valores propios a -7, -4, $-3/2$, $-1/2$ y 2, el más grande de los recíprocos es -2. Después de invertir para obtener $-1/2$ y de sumar de nuevo el desplazamiento $s = 2$, se obtiene 3/2. La razón de convergencia es de nuevo la relación $(2/3)/2 = 1/3$. ◄

12.1.4 Iteración del cociente de Rayleigh

El cociente de Rayleigh puede utilizarse junto con la iteración de potencia inversa. Se sabe que converge al vector propio asociado con el valor propio que guarda la menor distancia con el desplazamiento s, y que la convergencia es rápida si esta distancia es pequeña. Si en cualquier paso del proceso se conoce un valor propio aproximado, éste podría utilizarse como el desplazamiento s, para acelerar la convergencia.

El uso del cociente de Rayleigh como el desplazamiento actualizado en la iteración de potencia inversa conduce a la iteración del cociente de Rayleigh (ICR).

Iteración del cociente de Rayleigh

Dado el vector inicial x_0.

for $j = 1, 2, 3, \ldots$

$u_{j-1} = x_{j-1}/||x_{j-1}||$

$\lambda_{j-1} = u_{j-1}^T A u_{j-1}$

Resuelvo $(A - \lambda_{j-1} I)x_j = u_{j-1}$

end

$u_j = x_j/||x_j||_2$

```
% Programa 12.3 iteración del cociente de Rayleigh
% Entrada: matriz A, vector inicial x (distinto de cero), número de pasos k
% Salida: valor propio lam y vector propio u
function [lam,u]=rqi(A,x,k)
for j=1:k
  u=x/norm(x);                % normaliza
  lam=u'*A*u;                 % cociente de Rayleigh
  x=(A-lam*eye(size(A)))\u;   % iteración de potencia inversa
end
u=x/norm(x);
lam=u'*A*u;                   % cociente de Rayleigh
```

Mientras la iteración de potencia inversa converge linealmente, la iteración del cociente de Rayleigh es cuadráticamente convergente para los valores propios simples (sin repetición) y convergerá cúbicamente si la matriz es simétrica. Esto significa que el método del cociente de Rayleigh requiere muy pocos pasos para converger a la precisión de máquina. Después de la convergencia, la matriz $A - \lambda_{j-1} I$ es singular y no pueden realizarse más pasos. Como resultado, es necesario utilizar ensayo y error con el programa 12.3 para detener la iteración justo antes de que esto ocurra. Observe que la complejidad se ha incrementado con la ICR. La iteración de potencia inversa requiere sólo una factorización LU; pero para la ICR, cada paso requiere una factorización nueva, dado que el desplazamiento ha cambiado. Aun así, la iteración del cociente de Rayleigh es el método convergente más rápido que se ha presentado en esta sección para la búsqueda de un valor propio a la vez. En la siguiente sección se analizan maneras de encontrar todos los valores propios de una matriz en el mismo cálculo. El motor básico seguirá siendo la iteración de potencia (sólo los detalles se volverán más sofisticados).

12.1 Ejercicios

1. Encuentre el polinomio característico y los valores y vectores propios de las siguientes matrices simétricas:

(a) $\begin{bmatrix} 3.5 & -1.5 \\ -1.5 & 3.5 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} -0.2 & -2.4 \\ -2.4 & 1.2 \end{bmatrix}$ (d) $\begin{bmatrix} 136 & -48 \\ -48 & 164 \end{bmatrix}$

2. Encuentre el polinomio característico y los valores y vectores propios de las siguientes matrices:

(a) $\begin{bmatrix} 7 & 9 \\ -6 & -8 \end{bmatrix}$ (b) $\begin{bmatrix} 2 & 6 \\ 1 & 3 \end{bmatrix}$ (c) $\begin{bmatrix} 2.2 & 0.6 \\ -0.4 & 0.8 \end{bmatrix}$ (d) $\begin{bmatrix} 32 & 45 \\ -18 & -25 \end{bmatrix}$

3. Encuentre el polinomio característico y los valores y vectores propios de las siguientes matrices:

(a) $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 3 & -2 \\ 0 & 0 & 2 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & 1 & \frac{2}{3} \\ -1 & 1 & 1 \end{bmatrix}$ (c) $\begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{6} \\ -1 & 0 & \frac{1}{3} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$

4. Demuestre que una matriz cuadrada y su transpuesta tienen el mismo polinomio característico y, por lo tanto, el mismo conjunto de valores propios.

5. Suponga que A es una matriz de 3×3 con los valores propios dados. Decida a qué valor propio de la iteración de potencia converge y determine la razón de convergencia constante S. (a) $\{3, 1, 4\}$ (b) $\{3, 1, -4\}$ (c) $\{-1, 2, 4\}$ (d) $\{1, 9, 10\}$

6. Suponga que A es una matriz de 3×3 con los valores propios dados. Decida a qué valor propio de la iteración de potencia converge y determine la razón de convergencia constante S. (a) $\{1, 2, 7\}$ (b) $\{1, 1, -4\}$ (c) $\{0, -2, 5\}$ (d) $\{8, -9, 10\}$

7. Suponga que A es una matriz de 3×3 con los valores propios dados. Decida a qué valor propio de la iteración de potencia inversa con el desplazamiento dado s convergerá, y determine la razón de convergencia constante S. (a) $\{3, 1, 4\}$, $s = 0$ (b) $\{3, 1, -4\}$, $s = 0$ (c) $\{-1, 2, 4\}$, $s = 0$ (d) $\{1, 9, 10\}$, $s = 6$

8. Suponga que A es una matriz de 3×3 con los valores propios dados. Decida a qué valor propio de la iteración de potencia inversa con el desplazamiento dado s convergerá y determine la razón de convergencia constante S. (a) $\{3, 1, 4\}$, $s = 5$ (b) $\{3, 1, -4\}$, $s = 4$ (c) $\{-1, 2, 4\}$, $s = 1$ (d) $\{1, 9, 10\}$, $s = 8$

9. Sea $A = \begin{bmatrix} 1 & 2 \\ 4 & 3 \end{bmatrix}$. (a) Encuentre todos los valores y vectores propios de A (b) Aplique tres pasos de la iteración de potencia con el vector inicial $x_0 = (1, 0)$. En cada paso, aproxime el valor propio mediante el cociente actual de Rayleigh. (c) Prediga el resultado de aplicar la iteración de potencia inversa con desplazamiento $s = 0$ (d) con desplazamiento $s = 3$.

10. Sea $A = \begin{bmatrix} -2 & 1 \\ 3 & 0 \end{bmatrix}$. Realice los pasos del ejercicio 9 para esta matriz.

11. Si A es una matriz de 6×6 con valores propios $-6, -3, 1, 2, 5, 7$, ¿cuál valor propio de A encontrarán los siguientes algoritmos? (a) Iteración de potencia (b) Iteración de potencia inversa con desplazamiento $s = 4$ (c) Encuentre las razones de convergencia lineal para los dos cálculos. ¿Cuál de ellos converge más rápido?

12.1 Problemas de computadora

1. Use el código proporcionado (o el código de su preferencia) para el método de la iteración de potencia, encuentre el vector propio dominante de A y estime el valor propio dominante calculando un cociente de Rayleigh. Compare sus conclusiones con la parte correspondiente del ejercicio 5.

(a) $\begin{bmatrix} 10 & -12 & -6 \\ 5 & -5 & -4 \\ -1 & 0 & 3 \end{bmatrix}$ (b) $\begin{bmatrix} -14 & 20 & 10 \\ -19 & 27 & 12 \\ 23 & -32 & -13 \end{bmatrix}$

(c) $\begin{bmatrix} 8 & -8 & -4 \\ 12 & -15 & -7 \\ -18 & 26 & 12 \end{bmatrix}$ (d) $\begin{bmatrix} 12 & -4 & -2 \\ 19 & -19 & -10 \\ -35 & 52 & 27 \end{bmatrix}$

2. Use el código proporcionado (o el código de su preferencia) para el método de la iteración de potencia inversa, verifique sus conclusiones del ejercicio 7, utilice la matriz apropiada de problema de computadora 1.

3. Para el método de la iteración de potencia inversa, verifique sus conclusiones del ejercicio 8, utilice la matriz apropiada de problema de computadora 1.

4. Aplique la iteración del cociente de Rayleigh para las matrices del problema de computadora 1. Pruebe diferentes vectores de inicio hasta encontrar los tres valores propios.

12.2 ALGORITMO QR

El objetivo de esta sección es desarrollar métodos para encontrar todos los valores propios a la vez. Se inicia con un método que funciona para matrices simétricas, y posteriormente se complementará a fin de que funcione de manera general. Las matrices simétricas son más fáciles de manejar debido a que sus valores propios son reales y sus vectores propios forman una base ortonormal de R^m (vea el apéndice A). Esto motiva la aplicación de la iteración de potencia con m vectores en paralelo, donde se trabaja de manera activa para mantener los vectores ortogonales entre sí.

12.2.1 Iteración simultánea

Suponga que se comienza con m vectores iniciales ortogonales por pares $v_1, \ldots, v_m$. Después de aplicar un paso de la iteración de potencia a cada vector, $Av_1, \ldots, Av_m$ ya no se garantiza que sean ortogonales entre sí. De hecho, con más multiplicaciones por A, todos tenderían a converger al vector propio dominante, de acuerdo con el teorema 12.2.

Para evitar esto, se vuelve a ortogonalizar el conjunto de m vectores en cada paso. La multiplicación simultánea de los m vectores por A se escribe de manera eficaz como el producto matricial

$$A[v_1|\cdots|v_m].$$

Como se encontró en el capítulo 4, el paso de ortogonalización puede verse como la factorización del producto resultante como QR. Si los vectores básicos elementales se utilizan como vectores iniciales, entonces el primer paso de la iteración de potencia seguido por la reortogonalización es $AI = \overline{Q}_1 R_1$, o

$$\left[\, A\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix} \middle| A\begin{bmatrix}0\\1\\\vdots\\0\end{bmatrix} \middle| \cdots \middle| A\begin{bmatrix}0\\0\\\vdots\\1\end{bmatrix} \right] = \left[\overline{q}_1^1|\cdots|\overline{q}_m^1\right] \begin{bmatrix} r_{11}^1 & r_{12}^1 & \cdots & r_{1m}^1 \\ & r_{22}^1 & & \vdots \\ & & \ddots & \vdots \\ & & & r_{mm}^1 \end{bmatrix}. \tag{12.5}$$

La $\overline{q}_i^1$ para $i = 1, \ldots, m$ son el nuevo conjunto ortogonal de vectores unitarios en el proceso de la iteración de potencia. Enseguida, se repite el paso:

$$\begin{aligned} A\overline{Q}_1 &= \left[A\overline{q}_1^1 | A\overline{q}_2^1 | \cdots | A\overline{q}_m^1 \right] \\ &= \left[\overline{q}_1^2 | \overline{q}_2^2 | \cdots | \overline{q}_m^2 \right] \begin{bmatrix} r_{11}^2 & r_{12}^2 & \cdots & r_{1m}^2 \\ & r_{22}^2 & & \vdots \\ & & \ddots & \vdots \\ & & & r_{mm}^2 \end{bmatrix} \\ &= \overline{Q}_2 R_2. \end{aligned} \tag{12.6}$$

En otras palabras, se ha desarrollado una forma matricial de la iteración de potencia que busca los m vectores propios de una matriz simétrica en forma simultánea.

Iteración simultánea normalizada

Establezca $\overline{Q}_0 = I$
for $j = 1, 2, 3, \ldots$
$A\overline{Q}_j = \overline{Q}_{j+1} R_{j+1}$
end

En el j-ésimo paso las columnas de $\overline{Q}_j$ son aproximaciones a los vectores propios de A y los elementos diagonales, $r_{11}^j, \ldots, r_{mm}^j$, son aproximaciones a los valores propios. Este algoritmo, que se llamará iteración simultánea normalizada (ISN), puede escribirse en código de MATLAB de manera muy compacta.

```
% Programa 12.4 Iteración simultánea normalizada
% Calcula los valores/vectores propios de una matriz simétrica
% Entrada: matriz A, número de pasos k
% Salida: valores propios lam y matriz Q de vectores propios
function [lam,Q]=nsi(A,k)
[m,n]=size(A);
Q=eye(m,m);
for j=1:k
    [Q,R]=qr(A*Q);              % Factorización QR
end
lam=diag(Q'*A*Q);               % Cociente de Rayleigh
```

Existe una forma aún más compacta de implementar la iteración simultánea normalizada. Establezca $\overline{Q}_0 = I$. Después, la ISN procede de la siguiente manera:

$$\begin{aligned} A\overline{Q}_0 &= \overline{Q}_1 R_1 \\ A\overline{Q}_1 &= \overline{Q}_2 R_2 \\ A\overline{Q}_2 &= \overline{Q}_3 R_3 \\ &\vdots \\ &\cdot \end{aligned} \tag{12.7}$$

Considere la iteración similar $Q_0 = I$, y

$$\begin{aligned} A_0 &\equiv AQ_0 = Q_1 R_1' \\ A_1 &\equiv R_1' Q_1 = Q_2 R_2' \\ A_2 &\equiv R_2' Q_2 = Q_3 R_3' \\ &\vdots \end{aligned} \tag{12.8}$$

que se llamará el **algoritmo QR sin desplazamiento**. La única diferencia es que A no es necesaria después del primer paso, sino que se sustituye por la actual R_k. Al comparar (12.7) y (12.8) se observa que podría elegirse $Q_1 = \overline{Q}_1$ y $R_1 = R_1'$ en (12.7). Además, como

$$\overline{Q}_2 R_2 = A\overline{Q}_1 = Q_1 R_1' \overline{Q}_1 = Q_1 R_1' Q_1 = Q_1 Q_2 R_2', \tag{12.9}$$

podría elegirse $\overline{Q}_2 = Q_1 Q_2$ y $R_2 = R_2'$ en (12.7). De hecho, si se ha elegido $\overline{Q}_{k-1} = Q_1 \cdots Q_{k-1}$ y $R_{j-1} = R_{j-1}'$, entonces

$$\begin{aligned} \overline{Q}_j R_j &= A\overline{Q}_{j-1} = AQ_1 \cdots Q_{j-1} \\ &= \overline{Q}_2 R_2 Q_2 \cdots Q_{j-1} \\ &= \overline{Q}_2 Q_3 R_3 Q_3 \cdots Q_{j-1} \\ &= Q_1 Q_2 Q_3 Q_4 R_4 Q_4 \cdots Q_{j-1} \\ &= \cdots = Q_1 \cdots Q_j R_j, \end{aligned} \tag{12.10}$$

y puede definirse $\overline{Q}_j = Q_1 \cdots Q_j$ y $R_j = R_j'$ en (12.7).
Por lo tanto, el algoritmo QR sin desplazamiento hace los mismos cálculos que la iteración simultánea normalizada, con una notación ligeramente diferente. Tenga en cuenta también que

$$A_{j-1} = Q_j R_j = Q_j R_j Q_j Q_j^T = Q_j A_j Q_j^T, \tag{12.11}$$

de modo que todas las matrices A_j son similares y tienen el mismo conjunto de valores propios.

```
% Programa 12.5 Algoritmo QR sin desplazamiento
% Calcula los valores/vectores propios de una matriz simétrica
% Entrada: matriz A, número de pasos k
% Salida: valores propios lam y matriz de vectores propios Qbar
function [lam,Qbar]=unshiftedqr(A,k)
[m,n]=size(A);
Q=eye(m,m);
Qbar=Q; R=A;
for j=1:k
    [Q,R]=qr(R*Q);          % factorización QR
    Qbar=Qbar*Q;            % acumula Q's
end
lam=diag(R*Q);              % diagonal converge a los valores propios
```

TEOREMA 12.4 Suponga que A es una matriz simétrica de $m \times m$ con valores propios λ_i que satisfacen $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_m|$. El algoritmo QR sin desplazamiento converge linealmente a los vectores propios y valores propios de A. Cuando $j \to \infty$, A_j converge a una matriz diagonal que contiene los valores propios en la diagonal principal y $\overline{Q}_j = Q_1 \cdots Q_j$ converge a una matriz ortogonal cuyas columnas son los vectores propios. ■

Puede encontrar una demostración del teorema 12.4 en Golub y Van Loan [1996]. La iteración simultánea normalizada, es en esencia el mismo algoritmo, converge bajo las mismas condiciones. Tenga en cuenta que el algoritmo QR sin desplazamiento puede fallar incluso para matrices simétricas si no se cumplen los supuestos del teorema. Vea el ejercicio 5.

Aunque la QR sin desplazamiento es una versión mejorada de la iteración de potencia, las condiciones exigidas por el teorema 12.4 son estrictas, y se requieren un par de mejoras para que este buscador de valores propios funcione de una manera más general; por ejemplo, en el caso de las matrices no simétricas. Un problema, que también se presenta para las matrices simétricas, es que la QR sin desplazamiento no garantiza su funcionamiento en el caso de simetría en el vector propio dominante. Un ejemplo de esto es

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

que tiene valores propios 1 y -1. Otra forma de simetría" se produce cuando los valores propios son complejos. Los valores propios de la matriz no simétrica

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

son i y $-i$, ambos de magnitud compleja 1. Nada en la definición del algoritmo QR sin desplazamiento permite calcular valores propios complejos. Por otra parte, la QR sin desplazamiento no utiliza el truco de la iteración de potencia inversa. Se encontró que la iteración de potencia podría acelerarse de manera considerable con este truco, y se desea encontrar una manera de aplicar dicha idea a la nueva implementación. Estas opciones se aplicarán a continuación, después de introducir el objetivo del algoritmo QR, que es reducir la matriz A a su forma real de Schur.

12.2.2 Forma real de Schur y el algoritmo QR

La forma en que el algoritmo QR encuentra valores propios de una matriz A consiste en localizar una matriz similar cuyos valores propios sean obvios. Un ejemplo de esto último es la forma real de Schur.

DEFINICIÓN 12.5 Una matriz T tiene la **forma real de Schur** si es triangular superior, con la posible excepción de bloques de 2×2 sobre la diagonal principal. ❐

Por ejemplo, una matriz de la forma

$$\begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & x & x & x \\ & & & & x \end{bmatrix}$$

Tiene una forma real de Schur. De acuerdo con el ejercicio 6, los valores propios de una matriz con esta forma son los valores propios del bloque diagonal (entradas diagonales cuando el bloque es de 1×1, o los valores propios del bloque de 2×2 en ese caso). De cualquier manera, los valores propios de la matriz se calculan rápido.

El valor de la definición es que cualquier matriz cuadrada con elementos reales es similar a una de esta forma. Ésta es la conclusión del siguiente teorema, demostrado en Golub y Van Loan [1996]:

TEOREMA 12.6 Sea A una matriz cuadrada con entradas reales. Entonces existe una matriz ortogonal Q y una matriz T con forma real de Schur, tal que $A = Q^T T Q$. ■

La llamada factorización de Schur de la matriz A es una "factorización reveladora de valores propios", lo que significa que si puede llevarse a cabo, se conocerán los valores propios y los vectores propios.

El algoritmo QR completo mueve iterativamente una matriz arbitraria A hacia su factorización de Schur mediante una serie de transformaciones similares. Se procederá en dos etapas. En primer lugar recordaremos la iteración de potencia inversa con desplazamientos, y en segundo término la deflación para desarrollar el algoritmo QR desplazado. Después se desarrollará una versión mejorada que permita valores propios complejos.

La versión modificada es fácil de escribir. Cada paso consiste en aplicar el desplazamiento, completando una factorización QR, para después tomar el desplazamiento de nuevo. De manera simbólica,

$$\begin{aligned} A_0 - sI &= Q_1 R_1 \\ A_1 &= R_1 Q_1 + sI. \end{aligned} \tag{12.12}$$

Observe que

$$\begin{aligned} A_1 - sI &= R_1 Q_1 \\ &= Q_1^T (A_0 - sI) Q_1 \\ &= Q_1^T A_0 Q_1 - sI \end{aligned}$$

implica que A_1 es similar a A_0 y, por lo tanto, tiene los mismos valores propios. Se repite este paso, generando una secuencia A_k de matrices, todas similares a $A = A_0$.

¿Cuáles son algunas buenas opciones para el desplazamiento s? Esto conduce al concepto de **deflación** para el cálculo de valores propios. Se elegirá el desplazamiento como la entrada inferior derecha de la matriz A_k. Esto hará que la iteración, a medida que converja a la forma real de Schur, mueva el renglón inferior hacia un renglón de ceros, a excepción de la entrada inferior derecha. Después de que esta entrada ha convergido a un valor propio, deflactamos la matriz mediante la eliminación del último renglón y la última columna. Después se procede a encontrar el resto de los valores propios.

Un primer intento en el algoritmo QR desplazado se da en el código de MATLAB que se muestra en el programa 12.6. En cada paso, se aplica un paso de QR desplazado, y después se verifica el renglón inferior. Si todas las entradas son pequeñas, excepto la entrada diagonal a_{nn}, se declara que la entrada es un valor propio y se deflacta al no tomar en cuenta la última fila y la última columna para el resto del cálculo. Este programa tendrá éxito bajo las hipótesis del teorema 12.4. Los valores propios complejos, o valores propios reales con la misma magnitud, pueden causar problemas, que se resolverán más adelante en una versión más sofisticada. El ejercicio 7 ilustra las deficiencias de esta versión preliminar del algoritmo QR.

```
% Programa 12.6 Algoritmo QR desplazado, versión preliminar
% Calcula los valores propios de matrices sin valores propios del mismo tamaño
% Entrada: matriz a
% Salida: valores propios lam
function lam=shiftedqr0(a)
tol=1e-14;
m=size(a,1);lam=zeros(m,1);
n=m;
while n>1
    while max(abs(a(n,1:n-1)))>tol
        mu=a(n,n);      % define el desplazamiento mu
        [q,r]=qr(a-mu*eye(n));
        a=r*q+mu*eye(n);
    end
    lam(n)=a(n,n);      % declara el valor propio
    n=n-1;              % decremento n
    a=a(1:n,1:n);       % deflacta
end
lam(1)=a(1,1);          % permanece la matriz 1x1
```

Finalmente, para calcular los valores propios complejos, debe permitirse la existencia de bloques de 2×2 sobre la diagonal de la forma real de Schur. La versión mejorada del algoritmo QR desplazado dado en el programa 12.7 intenta iterar la matriz hasta un bloque diagonal de 1×1 en la esquina inferior derecha, y si falla (después de un número de intentos especificado por el usuario), declara un bloque de 2×2, encuentra el par de valores propios, y después se deflacta en 2. Esta versión mejorada converge a la forma real de Schur para la mayoría, pero no para todas, las matrices de entrada. Para redondear algunas ideas finales, así como hacer que el algoritmo sea más eficiente, en la siguiente sección se desarrollará una forma superior de Hessenberg.

```
% Programa 12.7 Algoritmo QR desplazado, versión general
% Calcula los valores propios reales y complejos de una matriz cuadrada
% Entrada: matriz a
```

```
% Salida: valores propios lam
function lam=shiftedqr(a)
tol=1e-14;kounttol=500;
m=size(a,1);lam=zeros(m,1);
n=m;
while n>1
    kount=0;
    while max(abs(a(n,1:n-1)))>tol & kount<kounttol
        kount=kount+1;      % da seguimiento al número de qr's
        mu=a(n,n);          % el desplazamiento es mu
        [q,r]=qr(a-mu*eye(n));
        a=r*q+mu*eye(n);
    end
    if kount<kounttol       % ha aislado el bloque de 1x1
        lam(n)=a(n,n);      % declara el valor propio
        n=n-1;
        a=a(1:n,1:n);       % deflacta por 1
    else                    % ha aislado el bloque de 2x2
        disc=(a(n-1,n-1)-a(n,n))^2+4*a(n,n-1)*a(n-1,n);
        lam(n)=(a(n-1,n-1)+a(n,n)+sqrt(disc))/2;
        lam(n-1)=(a(n-1,n-1)+a(n,n)-sqrt(disc))/2;
        n=n-2;
        a=a(1:n,1:n);       % deflactado por 2
    end
end
if n>0;lam(1)=a(1,1);end    % sólo queda un bloque de 1x1
```

Incluso en su forma general, el algoritmo QR desplazado falla para el ejemplo siguiente:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \tag{12.13}$$

Matrices como ésta, con un valor propio complejo repetido, no pueden moverse hacia la forma real de Schur mediante el algoritmo QR desplazado. La ayuda extra necesaria para estos ejemplos más difíciles consiste en reemplazar A por una matriz similar en forma superior de Hessenberg, que es el tema de estudio de la siguiente sección.

12.2.3 Forma superior de Hessenberg

La eficiencia del algoritmo QR aumenta de manera considerable si primero se pone A en la forma superior de Hessenberg. La idea es aplicar transformaciones similares, antes de comenzar la iteración QR, que ponga tantos ceros como sea posible en A, preservando al mismo tiempo todos los valores propios. Además, la forma superior de Hessenberg eliminará la dificultad final de la versión del algoritmo QR que se ha desarrollado (la convergencia a varios valores propios complejos) garantizando que la iteración QR siempre procederá hasta los bloques de 1×1 o 2×2.

DEFINICIÓN 12.7 La matriz A de $m \times n$ está en la **forma superior de Hessenberg** si $a_{ij} = 0$ para $i > j + 1$. ❐

Una matriz de la forma

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \end{bmatrix}$$

es superior de Hessenberg. Existe un algoritmo finito para poner matrices en la forma superior de Hessenberg mediante transformaciones similares de semejanza.

TEOREMA 12.8 Sea A una matriz cuadrada. Existe una matriz ortogonal Q tal que $A = QBQ^T$ y B está en la forma superior de Hessenberg. ■

Se construirá B usando Householder de la sección 4.3.3, donde se emplearon para construir la factorización QR. Sin embargo, hay una diferencia importante: ahora se tiene interés en la multiplicación por el reflector H a la izquierda y la derecha de la matriz, puesto que se quiere terminar con una matriz similar con valores propios idénticos. Debido a esto, es necesario ser menos agresivos en cuanto a los ceros que pueden instalarse en A.

Defina x como el vector de longitud $n - 1$ que consta de todas las entradas excepto la primera entrada de la primera columna de A. Sea $\hat{H}_1$ el reflector de Householder que mueve a x hacia $(\pm\|x\|, 0, \ldots, 0)$. (Como se indicó en el capítulo 4, debe elegirse el signo como $-\text{signo}(x_1)$ para evitar problemas de cancelación en la práctica, pero en la teoría cualquiera de las opciones es válida). Sea H_1 la matriz ortogonal formada al insertar $\hat{H}_1$ en la esquina inferior $(n-1) \times (n-1)$ de la matriz identidad de $n \times n$. Entonces, se tiene

$$H_1 A = \left[\begin{array}{c|cccc} 1 & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & \hat{H}_1 & & \\ 0 & & & & \end{array}\right] \left[\begin{array}{c|cccc} x & x & x & x & x \\ \hline x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{array}\right] = \left[\begin{array}{c|cccc} x & x & x & x & x \\ \hline x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{array}\right].$$

Antes de que se pueda evaluar el éxito al colocar ceros en la matriz, es necesario terminar la transformación de semejanza multiplicando por H_1^{-1} a la derecha. Recuerde que los reflectores de Householder son matrices simétricas ortogonales, por lo que $H_1^{-1} = H_1^T = H_1$. Por lo tanto,

$$H_1 A H_1 = \left[\begin{array}{c|cccc} x & x & x & x & x \\ \hline x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{array}\right] \left[\begin{array}{c|cccc} 1 & 0 & 0 & 0 & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & \hat{H}_1 & & \\ 0 & & & & \end{array}\right] = \left[\begin{array}{c|cccc} x & x & x & x & x \\ \hline x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{array}\right].$$

Los ceros realizados en H_1A no se cambian en la matriz H_1AH_1. Sin embargo, observe que si se hubieran tratado de eliminar todos los elementos menos uno distinto de cero en la primera columna, como se hizo en la factorización QR de la sección previa, se habría fallado en mantener los ceros al multiplicar por la derecha. De hecho, no hay ningún algoritmo finito que calcule una transformación similar entre una matriz arbitraria y una matriz triangular superior. Si lo hubiera, este capítulo sería mucho más corto, ya que podrían haberse leído los valores propios de la matriz arbitraria a partir de la diagonal de la matriz similar, triangular superior.

El siguiente paso para lograr la forma superior de Hessenberg es repetir el paso anterior, utilizando para x el vector $(n-2)$-dimensional que consiste en las $n-2$ entradas inferiores de la segunda columna. Sea $\hat{H}_2$ el reflector de Householder de $(n-2) \times (n-2)$ para la nueva x, y defina H_2 como la matriz identidad con $\hat{H}_2$ en la esquina inferior. Entonces

$$H_2(H_1 A H_1) = \left[\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & & & \\ 0 & 0 & & \hat{H}_2 & \\ 0 & 0 & & & \end{array}\right] \left[\begin{array}{cc|ccc} x & x & x & x & x \\ x & x & x & x & x \\ \hline 0 & x & x & x & x \\ 0 & x & x & x & x \\ 0 & x & x & x & x \end{array}\right] = \left[\begin{array}{cc|ccc} x & x & x & x & x \\ \hline x & x & x & x & x \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & x & x & x \end{array}\right],$$

y, además, compruebe que al igual que H_1, la multiplicación a la derecha por H_2 no afecta de manera negativa a los ceros ya obtenidos. Si $n = 5$, entonces después de un paso más, se obtiene la matriz de 5×5

$$H_3H_2H_1AH_1^TH_2^TH_3^T = H_3H_2H_1A(H_3H_2H_1)^T = QAQ^T$$

en la forma superior de Hessenberg. Dado que la matriz es similar a A, tiene los mismos valores propios y multiplicidades que A. En general, para una matriz A de $n \times n$, se necesitan $n - 2$ pasos de Householder para poner A en la forma superior de Hessenberg.

► EJEMPLO 12.2 Sea $\begin{bmatrix} 2 & 1 & 0 \\ 3 & 5 & -5 \\ 4 & 0 & 0 \end{bmatrix}$ la forma superior de Hessenberg.

Sea $x = [3, 4]$. Anteriormente, se encontró el reflector de Householder

$$\hat{H}_1x = \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & -0.6 \end{bmatrix}\begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}.$$

Por lo tanto,

$$H_1A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.6 & 0.8 \\ 0 & 0.8 & -0.6 \end{bmatrix}\begin{bmatrix} 2 & 1 & 0 \\ 3 & 5 & -5 \\ 4 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 0 \\ 5 & 3 & -3 \\ 0 & 4 & -4 \end{bmatrix}$$

y

$$A' \equiv H_1AH_1 = \begin{bmatrix} 2 & 1 & 0 \\ 5 & 3 & -3 \\ 0 & 4 & -4 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.6 & 0.8 \\ 0 & 0.8 & -0.6 \end{bmatrix} = \begin{bmatrix} 2.0 & 0.6 & 0.8 \\ 5.0 & -0.6 & 4.2 \\ 0.0 & -0.8 & 5.6 \end{bmatrix}.$$

El resultado es una matriz A' que está en la forma superior de Hessenberg y es similar a A. ◄

A continuación se pone en práctica la estrategia anterior y se crea un algoritmo para la búsqueda de Q, usando las reflexiones de Householder:

```
% Programa 12.8 Forma superior de Hessenberg
% Entrada: matriz a
% Salida: matriz a y reflectores v en forma de Hessenberg
% Uso: [a,v] =hessen(a) produce una matriz a similar en
%   la forma de Hessenberg y una matriz v cuyas columnas contienen
%   las v's que definen los reflectores de Householder.
function [a,v]=hessen(a)
[m,n]=size(a);
v=zeros(m,m);
for k=1:m-2
  x=a(k+1:m,k);
  v(1:m-k,k)=-sign(x(1)+eps)*norm(x)*eye(m-k,1)-x;
  v(1:m-k,k)=v(1:m-k,k)/norm(v(1:m-k,k));
  a(k+1:m,k:m)=a(k+1:m,k:m)-2*v(1:m-k,k)*v(1:m-k,k)'*a(k+1:m,k:m);
  a(1:m,k+1:m)=a(1:m,k+1:m)-2*a(:,k+1:m)*v(1:m-k,k)*v(1:m-k,k)';
end
```

Una ventaja de la forma superior de Hessenberg para los cálculos de valores propios es que sólo pueden presentarse bloques de 2×2 a lo largo de la diagonal durante el algoritmo QR, con lo que se elimina la dificultad causada por los valores propios complejos repetidos de la sección anterior.

► EJEMPLO 12.3 Encuentre los valores propios de la matriz (12.13).

Para

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix},$$

la matriz similar con la forma superior de Hessenberg dada por los reflectores de Householder es

$$A' = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

Donde $A' = QAQ^T$ y

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

La matriz A' ya está en la forma real de Schur. Sus valores propios son los valores propios de las dos matrices de 2×2 a lo largo de la diagonal principal, que son pares repetidos de $\{i, -i\}$. ◄

Por lo tanto, finalmente se tiene un método completo para buscar todos los valores propios de una matriz cuadrada arbitraria A. La matriz se pone primero en la forma superior de Hessenberg usando una transformación similares (programa 12.8), y después se aplica el algoritmo QR desplazado (programa 12.7). El comando `eig` de MATLAB proporciona los valores propios precisos con base en esta progresión de cálculos.

Existen muchas técnicas alternativas para acelerar la convergencia del algoritmo QR que no se cubren aquí. El algoritmo QR está diseñado para matrices completas. Para los grandes sistemas dispersos, hay métodos alternativos que suelen ser más eficientes, vea Saad [2003].

12.2 Ejercicios

1. Encuentre las siguientes matrices en la forma superior de Hessenberg:

(a) $\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} 2 & 1 & 0 \\ 4 & 1 & 1 \\ 3 & 0 & 1 \end{bmatrix}$ (d) $\begin{bmatrix} 1 & 1 & 0 \\ 2 & 3 & 1 \\ 2 & 1 & 0 \end{bmatrix}$

2. Encuentre la matriz $\begin{bmatrix} 1 & 0 & 2 & 3 \\ -1 & 0 & 5 & 2 \\ 2 & -2 & 0 & 0 \\ 2 & -1 & 2 & 0 \end{bmatrix}$ en la forma superior de Hessenberg.

3. Demuestre que una matriz simétrica en la forma de Hessenberg es tridiagonal.

4. Una matriz cuadrada se llama **estocástica** si las entradas de cada columna suman uno. Demuestre que una matriz estocástica (a) tiene un valor propio igual a uno y (b) que todos los valores propios son, a lo sumo, uno en valor absoluto.

5. Realice la iteración simultánea normalizada con las siguientes matrices y explique porque falla el procedimiento:

$$\text{(a)} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

6. (a) Demuestre que el determinante de una matriz en la forma real de Schur es el producto de los determinantes de los bloques de 1×1 y 2×2 de la diagonal principal. (b) Demuestre que los valores propios de una matriz en la forma real de Schur son los valores propios de los bloques de 1×1 y 2×2 sobre la diagonal principal.

7. Decida si la versión preliminar del algoritmo QR encuentra los valores propios correctos, tanto antes como después de cambiar a la forma de Hessenberg.

$$\text{(a)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

8. Decida si la versión general del algoritmo QR encuentra los valores propios correctos, tanto antes como después de cambiar a la forma de Hessenberg, para las matrices del ejercicio 7.

12.2 Problemas de computadora

1. Aplique el algoritmo QR desplazado (versión preliminar `shiftedqr0`) con tolerancia 10^{-14} directamente sobre las matrices siguientes:

$$\text{(a)} \begin{bmatrix} -3 & 3 & 5 \\ 1 & -5 & -5 \\ 6 & 6 & 4 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 3 & 1 & 2 \\ 1 & 3 & -2 \\ 2 & 2 & 6 \end{bmatrix}$$

$$\text{(c)} \begin{bmatrix} 17 & 1 & 2 \\ 1 & 17 & -2 \\ 2 & 2 & 20 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} -7 & -8 & 1 \\ 17 & 18 & -1 \\ -8 & -8 & 2 \end{bmatrix}$$

2. Aplique el método del algoritmo QR desplazado directamente para encontrar todos los valores propios de las matrices siguientes:

$$\text{(a)} \begin{bmatrix} 3 & 1 & -2 \\ 4 & 1 & 1 \\ -3 & 0 & 3 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 1 & 5 & 4 \\ 2 & -4 & -3 \\ 0 & -2 & 4 \end{bmatrix}$$

$$\text{(c)} \begin{bmatrix} 1 & 1 & -2 \\ 4 & 2 & -3 \\ 0 & -2 & 2 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} 5 & -1 & 3 \\ 0 & 6 & 1 \\ 3 & 3 & -3 \end{bmatrix}$$

3. Aplique el método del algoritmo QR desplazado directamente para encontrar todos los valores propios de las matrices siguientes:

$$\text{(a)} \begin{bmatrix} -1 & 1 & 3 \\ 3 & 3 & -2 \\ -5 & 2 & 7 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 7 & -33 & -15 \\ 2 & 26 & 7 \\ -4 & -50 & -13 \end{bmatrix}$$

$$\text{(c)} \begin{bmatrix} 8 & 0 & 5 \\ -5 & 3 & -5 \\ 10 & 0 & 13 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} -3 & -1 & 1 \\ 5 & 3 & -1 \\ -2 & -2 & 0 \end{bmatrix}$$

4. Repita el problema de computadora 3, pero antes de la aplicación de la iteración QR reduzca a la forma superior de Hessenberg. Imprima la forma de Hessenberg y los valores propios.

5. Aplique el algoritmo QR directamente para encontrar todos los valores propios reales y complejos de las siguientes matrices:

$$\text{(a)} \begin{bmatrix} 4 & 3 & 1 \\ -5 & -3 & 0 \\ 3 & 2 & 1 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 3 & 2 & 0 \\ -4 & -2 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$\text{(c)} \begin{bmatrix} 7 & 2 & -4 \\ -8 & 0 & 7 \\ 2 & -1 & -2 \end{bmatrix} \quad \text{(d)} \begin{bmatrix} 11 & 4 & -2 \\ -10 & 0 & 5 \\ 4 & 1 & 2 \end{bmatrix}$$

6. Utilice el algoritmo QR para encontrar los valores propios. En cada matriz, todos los valores propios tienen la misma magnitud, de modo que puede ser necesaria la forma de Hessenberg. Compare los resultados del algoritmo QR, antes y después de la reducción a la forma de Hessenberg.

$$\text{(a)} \begin{bmatrix} -5 & -10 & -10 & 5 \\ 4 & 16 & 11 & -8 \\ 12 & 13 & 8 & -4 \\ 22 & 48 & 28 & -19 \end{bmatrix} \quad \text{(b)} \begin{bmatrix} 7 & 6 & 6 & -3 \\ -26 & -20 & -19 & 10 \\ 0 & -1 & 0 & 0 \\ -36 & -28 & -24 & 13 \end{bmatrix}$$

$$\text{(c)} \begin{bmatrix} 13 & 10 & 10 & -5 \\ -20 & -16 & -15 & 8 \\ -12 & -9 & -8 & 4 \\ -30 & -24 & -20 & 11 \end{bmatrix}$$

Comprobación en la realidad ✔

12 Cómo clasifican los motores de búsqueda la calidad de la página

Los motores de búsqueda en la web, como `Google.com` se distinguen por la calidad de sus respuestas a las consultas de búsqueda. Se analizará una aproximación al método de Google para juzgar la calidad de las páginas web usando el conocimiento de la red de vínculos que existen en la web.

Cuando se inicia una búsqueda en internet, hay una serie bastante compleja de tareas que realiza el motor de búsqueda. Una tarea obvia es la búsqueda de palabras coincidentes, para encontrar las páginas que contienen las palabras de la consulta, en el título o el cuerpo de la página. Otra tarea fundamental es clasificar las páginas que se identifican mediante la primera tarea, para ayudar al usuario a navegar por un conjunto grande de opciones. Para consultas muy específicas, puede haber sólo unas pocas coincidencias de texto, todas las cuales pueden presentarse al usuario. (En los primeros días de la web, había un juego para tratar de descubrir las consultas de búsqueda que resultaban exactamente en una coincidencia). En el caso de las consultas muy específicas, la calidad de las páginas obtenidas no es tan importante, puesto que puede no ser necesario clasificarlas. La necesidad de una clasificación de calidad es evidente para las búsquedas más generales. Por ejemplo, la consulta en Google "automóvil nuevo" devuelve varios millones de páginas, comenzando con servicios de compra de automóviles, un resultado bastante útil. ¿Cómo se determina la clasificación?

La respuesta a esta pregunta es que `Google.com` asigna un número real no negativo, llamado **page rank** (evaluación de página), a cada página web que indexa. La clasificación de página es calculada por Google en la que es una de las más grandes iteraciones de potencia en curso en el mundo para la determinación de vectores propios. Considere una gráfica como en la figura 12.1, donde cada uno de los n nodos representa una página web, y un borde dirigido desde el nodo i hasta el nodo j significa que la página i contiene un vínculo a la página j. Sea A la **matriz de adyacencia**,

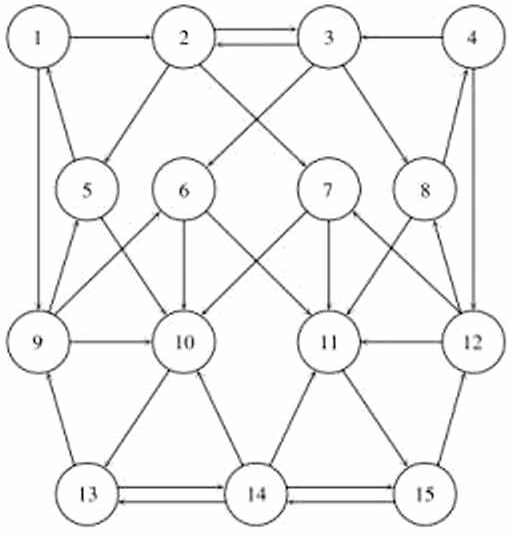


Figura 12.1 Una red de páginas y vínculos web. Cada borde dirigido de una página a otra significa que la primera página contiene al menos un vínculo a la segunda.

una matriz de $n \times n$ cuya ij-ésima entrada es 1 si hay un vínculo desde el nodo i hasta el nodo j, y 0 en caso contrario. Para la gráfica de la figura 12.1, la matriz de adyacencia es

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Los inventores de Google imaginaron a un surfista en una red de n páginas, que se encuentra en la página i con probabilidad p_i. Enseguida, el surfista se traslada ya sea a una página aleatoria (con probabilidad fija q, con frecuencia aproximada a 0.15) o bien, con probabilidad $1 - q$, hace clic de manera aleatoria en un vínculo de la página actual i. La probabilidad de que el surfista se mueva de la página i a la página j después del clic es $q/n + (1 - q)A_{ij}/n_i$, donde A_{ij} es la entrada de la matriz de adyacencia A y n_i es la suma de la i-ésima fila de A (en efecto, el número de vínculos en la página i).

Puesto que el tiempo es arbitrario, la probabilidad de estar en el nodo j es la suma de esta expresión sobre toda i, y es independiente del tiempo; es decir,

$$p_j = \sum_i \left(\frac{q\,p_i}{n} + (1 - q)\frac{p_i}{n_i} A_{ij} \right),$$

lo que equivale en términos matriciales a la ecuación de valor propio

$$p = Gp, \tag{12.14}$$

donde $p = (p_i)$ es el vector de n probabilidades de estar en las n páginas y G es la matriz cuya entrada ij es $q/n + A_{ji}(1 - q)/n_j$. Se llamará a G la **matriz google**. Cada columna de la matriz G suma uno, por lo que es una matriz estocástica y, de acuerdo con el ejercicio 12.2.4, tiene un valor propio mayor igual a uno. El vector propio p correspondiente al valor propio 1 es el conjunto de probabilidades de estado estable de las páginas, que son por definición las clasificaciones de página de las n páginas. (Ésta es la solución de estado estable del proceso de Markov definido por G^T. La idea original de medir la influencia de las probabilidades de estado estable se remonta a Pinski y Narin [1976]. La probabilidad de salto q fue añadida por Brin y Page [1998], los creadores de Google).

Se ilustrará la definición de la clasificación de página con el ejemplo que se muestra en la figura 12.1. Establezca $q = 0.15$. El vector propio principal (correspondiente al valor propio dominante 1) de la matriz google G es

$$p = \begin{bmatrix} 0.0268 \\ 0.0299 \\ 0.0299 \\ 0.0268 \\ 0.0396 \\ 0.0396 \\ 0.0396 \\ 0.0396 \\ 0.0746 \\ 0.1063 \\ 0.1063 \\ 0.0746 \\ 0.1251 \\ 0.1163 \\ 0.1251 \end{bmatrix}.$$

El vector propio se ha normalizado, al dividir entre la suma de todas las entradas, para tener una sumatoria igual a uno, como debe ser con las probabilidades. El vector propio con esta normalización contiene las clasificaciones de página. La clasificación de página es más alta para los nodos 13 y 15, seguidos por el nodo 14 y los nodos 10 y 11. Observe que la clasificación no depende simplemente de la "clasificación interior", o del número de vínculos que apuntan al interior de la página, sino que la asignación de evaluaciones de importancia es más sofisticada. Aunque los nodos 10 y 11 tienen el mayor número de vínculos que apuntan hacia adentro, el hecho de que apunten hacia 13 y 15 transfiere el control al nodo que sigue. Ésta es la idea detrás del "bombardeo de google", la práctica de inflar de manera artificial la importancia de un sitio al convencer a sitios de alto tráfico que se enlacen con él.

Tenga en cuenta que en la definición de la clasificación de página de esta manera se utiliza la palabra "importancia", aunque en realidad nadie sabe lo que eso significa. La clasificación de página es una forma autorreferencial de asignar una importancia que probablemente será suficiente hasta que se encuentre un mejor método.

Actividades sugeridas:

1. Demuestre que la matriz google G es una matriz estocástica.
2. Construya la matriz G para la red mostrada y verifique el vector propio dominante p dado.
3. Cambie la probabilidad de salto q a (a) 0 y (b) 0.5. Describa los cambios resultantes en la clasificación de página. ¿Cuál es el propósito de la probabilidad de salto?
4. Suponga que la página 7 en la red quiere mejorar su clasificación de página, en comparación con su competidor página 6 (por ejemplo, persuadiendo a las páginas 2 y 12 para que muestren de manera más prominente sus vínculos con la 7). Modele esto al sustituir A_{27} y $A_{12,7}$ por 2 en la matriz de adyacencia. ¿Tendrá éxito esta estrategia?, ¿qué otros cambios ve en las clasificaciones de página relativas?

5. Estudie el efecto de remover la página 10 de la red (todos los enlaces hacia y desde esta página se suprimen). ¿Cuáles clasificaciones aumentan y cuáles disminuyen?
6. Diseñe su propia red, calcule las clasificaciones de página y haga un análisis de acuerdo con las preguntas precedentes. ✔

12.3 DESCOMPOSICIÓN DE VALOR SINGULAR

La imagen de la esfera unitaria en R^m bajo una matriz de $m \times m$ es un elipsoide. Este hecho interesante subyace a la descomposición de valor singular, que tiene muchas aplicaciones en el análisis matricial en general y, en especial, para fines de compresión. En la figura 12.2 se muestra una ilustración de la elipse que corresponde a la matriz

$$A = \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \tag{12.15}$$

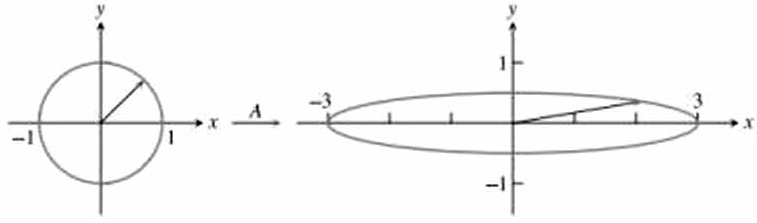


Figura 12.2 Imagen del círculo unitario bajo una matriz de 2×2. El círculo unitario en R^2 se mapea como una elipse de semiejes mayores (3, 0) y (0, 1/2) mediante la matriz A (12.15).

En la figura 12.2, piense en tomar el vector v correspondiente a cada punto sobre el círculo unitario, multiplicando por A, para después graficar el punto final del vector resultante Av. El resultado es la elipse mostrada. Con el fin de describir la elipse, resulta útil emplear un conjunto ortonormal de vectores para definir la base de un sistema de coordenadas.

Se verá en el teorema 12.11 que por cada matriz A de $m \times n$, hay conjuntos ortonormales $\{u_1, \ldots, u_m\}$ y $\{v_1, \ldots, v_n\}$, junto con los números no negativos $s_1 \geq \cdots \geq s_n \geq 0$, que satisfacen

$$\begin{aligned} Av_1 &= s_1 u_1 \\ Av_2 &= s_2 u_2 \\ &\vdots \\ Av_n &= s_n u_n. \end{aligned} \tag{12.16}$$

Los vectores se visualizan en la figura 12.3. Las v_i se llaman los **vectores singulares derechos** de la matriz A, las u_i son los **vectores singulares izquierdos** de A, y las s_i son los **valores singulares** de A. (La terminología de estos vectores es un poco extraña, pero las razones se aclararán en breve).

Este hecho explica de inmediato por qué una matriz de 2×2 mapea el círculo unitario en una elipse. Puede pensarse en las v_i como la base de un sistema de coordenadas rectangular en el que A actúa de una manera sencilla: produce los vectores básicos de un nuevo sistema de coordenadas, las u_i con un poco de extensión cuantificado por los escalares s_i. Los vectores básicos de extensión $s_i u_i$ son los semiejes mayores de la elipse, como se muestra en la figura 12.3.

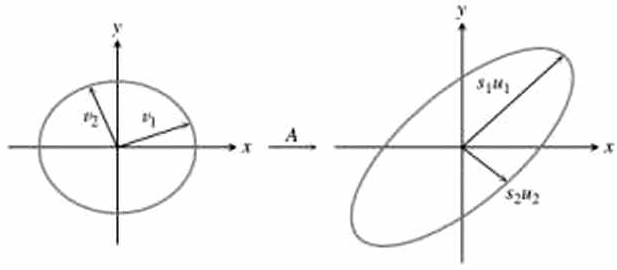


Figura 12.3 Elipse asociada a una matriz. Cada matriz A de 2×2 puede verse de la siguiente manera : existe un sistema de coordenadas $\{v_1, v_2\}$ para el que A envía $\mathbf{v_1} \to s_1\mathbf{u_1}$ y $\mathbf{v_2} \to s_2\mathbf{u_2}$, donde $\{u_1, u_2\}$ es otro sistema de coordenadas y s_1, s_2 son números no negativos. Esta visión se extiende a R^m para una matriz de $m \times m$.

► EJEMPLO 12.4 Encuentre los valores singulares y los vectores singulares de la matriz (12.15) representada en la figura 12.2.

Claramente, la matriz se extiende por 3 en la dirección x se contrae en un factor de 1/2 en la dirección y. Los vectores y valores singulares de A son

$$A\begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$A\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{12.17}$$

Los vectores $3(1, 0)$ y $\frac{1}{2}(0, 1)$ forman los semiejes mayores de la elipse. Los vectores singulares derechos son $[1, 0], [0, 1]$ y los vectores singulares izquierdos son $[1, 0], [0, 1]$. Los valores singulares son 3 y 1/2. ◄

► EJEMPLO 12.5 Encuentre los valores singulares y los vectores singulares de

$$A = \begin{bmatrix} 0 & -\frac{1}{2} \\ 3 & 0 \\ 0 & 0 \end{bmatrix}. \tag{12.18}$$

Ésta es una pequeña variación del ejemplo 12.4. La matriz intercambia los ejes x y y, con algunos cambios en la escala, y añade un eje z, a lo largo del cual no pasa nada. Los vectores y valores singulares de A son

$$Av_1 = A\begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = s_1u_1$$
$$Av_2 = A\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = s_2u_2. \tag{12.19}$$

Los vectores singulares derechos son $[1, 0], [0, 1]$ y los vectores singulares izquierdos son $[0, 1, 0]$, $[-1, 0, 0]$. Los valores singulares son 3, 1/2. Observe que siempre se requiere que la s_i sea un número no negativo, y cualesquiera signos negativos necesarios son absorbidos en la u_i y la v_i. ◄

Existe una manera estándar de dar seguimiento a esta información, en una factorización matricial de la matriz A de $m \times n$. Forme una matriz U de $m \times m$ cuyas columnas sean los vectores

singulares izquierdos u_i, una matriz V de $n \times n$ cuyas columnas sean los vectores singulares derechos v_i, y una matriz diagonal S de $m \times n$ cuyas entradas diagonales sean los valores singulares s_i. Entonces, la **descomposición de valor singular** (DVS) de la matriz A de $m \times n$ es

$$A = USV^T. \tag{12.20}$$

El ejemplo 12.5 tiene la representación de la DVS

$$\begin{bmatrix} 0 & -\frac{1}{2} \\ 3 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{12.21}$$

Como U y V son matrices cuadradas con columnas ortonormales, son matrices ortogonales. Tenga en cuenta que fue necesario añadir una tercera columna u_3 a U para completar la base de R^3. Finalmente, la terminología puede explicarse. Las u_i (v_i) son los vectores singulares izquierdos (derechos) porque aparecen en ese lado de la representación matricial (12.20).

12.3.1 Localización de la DVS en general

Se han mostrado dos ejemplos simples de la DVS. Para demostrar que la DVS existe para una matriz A general, es necesario el siguiente lema:

LEMA 12.10 Sea A una matriz de $m \times n$. Los valores propios de $A^T A$ son no negativos. ■

Demostración. Sea v un vector propio unitario de $A^T A$, y $A^T Av = \lambda v$. Entonces

$$0 \leq ||Av||^2 = v^T A^T Av = \lambda v^T v = \lambda. \qquad \square$$

Para una matriz A de $m \times n$, la matriz $A^T A$ de $n \times n$ es simétrica, por lo que sus vectores propios son ortogonales y sus valores propios son reales. El lema 12.10 muestra que los valores propios son números reales no negativos y por lo tanto deben expresarse como $s_1^2 \geq \cdots \geq s_n^2$, donde el correspondiente conjunto ortonormal de vectores propios es $\{v_1, \ldots, v_n\}$. Esto ya proporciona dos terceras partes de la DVS. Use las siguientes instrucciones a fin de encontrar las u_i para $1 \leq i \leq m$:

Si $s_i \neq 0$, defina u_i mediante la ecuación $s_i u_i = Av_i$.
Si $s_i = 0$, elija u_i como un vector unitario arbitrario sujeto a ser ortogonal a $u_1, \ldots, u_{i-1}$.

El lector debe comprobar que esta opción implica que $u_1, \ldots, u_m$ son vectores unitarios ortogonales por pares y, por lo tanto, otra base ortonormal de R^m. De hecho, $u_1, \ldots, u_m$ forma un conjunto ortonormal de vectores propios de AA^T (vea el ejercicio 4). En resumen, se ha demostrado el siguiente teorema:

TEOREMA 12.11 Sea A una matriz de $m \times n$. Entonces existen dos bases ortonormales $\{v_1, \ldots, v_n\}$ de R^n, y $\{u_1, \ldots, u_m\}$ de R^m y los números reales, $s_1 \geq \cdots \geq s_n \geq 0$ tales que $Av_i = s_i u_i$ para $1 \leq i \leq \text{mín}\{m, n\}$. Las columnas de $V = [v_1|\ldots|v_n]$, los vectores singulares derechos, son el conjunto de vectores propios ortonormales de $A^T A$, y las columnas de $U = [u_1|\ldots|u_m]$, los vectores singulares izquierdos, son el conjunto de vectores propios ortonormales de AA^T. ■

La DVS no es única para una matriz A dada. Por ejemplo, en la ecuación $Av_1 = s_1 u_1$, si se sustituye v_1 por $-v_1$ y u_1 por $-u_1$, no cambia la igualdad, pero sí cambian las matrices U y V.

A partir de este teorema se llega a la conclusión de que la imagen de la esfera unitaria de vectores es un elipsoide de vectores, centrado en el origen, con semiejes mayores $s_i u_i$. La figura

12.3 muestra que el círculo unitario de vectores se mapea en una elipse con ejes $\{s_1u_1, s_2u_2\}$. Para encontrar dónde A_x tiende a un vector x, puede escribirse $x = a_1v_1 + a_2v_2$ (donde $a_1v_1(a_2v_2)$ es la proyección de x sobre la dirección $v_1(v_2)$), y entonces $Ax = a_1s_1u_1 + a_2s_2u_2$.

La representación matricial (12.20) se desprende directamente del teorema 12.11. Defina S como una matriz diagonal de $m \times n$ cuyas entradas son $s_1 \geq \cdots \geq s_{\min\{m,n\}} \geq 0$. Defina U como la matriz cuyas columnas son $u_1, \ldots, u_m$, y V como la matriz cuyas columnas son $v_1, \ldots, v_n$. Observe que $USV^T v_i = s_i u_i$ para $i = 1, \ldots, m$. Como las matrices A y USV^T concuerdan en la base $v_1, \ldots, v_n$, son matrices idénticas de $m \times n$.

► EJEMPLO 12.6 Encuentre los valores y los vectores singulares de la matriz de 2×2

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}. \tag{12.22}$$

Los valores propios de

$$A^T A = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix},$$

ordenadas en tamaño decreciente, son $v_1 = [0, 1], s_1^2 = 2$; y $v_2 = [1, 0], s_2^2 = 0$. Los valores singulares son $\sqrt{2}$ y 0. De acuerdo con las indicaciones anteriores, u_1 se define mediante

$$\sqrt{2}u_1 = Av_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$u_1 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix},$$

y $u_2 = [1/\sqrt{2}, 1/\sqrt{2}]$ se elige para ser ortogonal a u_1. La DVS es

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{12.23}$$

De acuerdo con el comentario de no unicidad que se desprende del teorema 12.11, otra DVS perfectamente adecuada para esta matriz es

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -\sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \tag{12.24}$$

La imagen del círculo unitario bajo A es el segmento de línea $y[1, -1]$, donde y va de -1 a 1. Así, la acción de A es extender el círculo unitario hacia una elipse unidimensional con semiejes mayores $\sqrt{2}[\sqrt{2}/2, -\sqrt{2}/2]$ y 0. ◄

El comando de MATLAB para la descomposición de valor singular es `svd`, y

```
>> [u,s,v] =svd(a)
```

devolverá las tres matrices de la factorización.

12.3.2 Caso especial: matrices simétricas

La localización de la DVS de una matriz simétrica de $m \times m$ es tan sólo una cuestión de encontrar los valores y los vectores propios. El teorema A.5 del apéndice A garantiza que existe un conjunto ortonormal de vectores propios. Como los vectores propios se mapean a sí mismos (con una escala

λ, que es el valor propio), satisfacer la ecuación (12.16) es fácil: sólo ordene los valores propios en magnitud decreciente

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_m|, \tag{12.25}$$

y usándolos para los valores singulares $s_1 \geq s_2 \geq \ldots$. Para las v_1, use los vectores propios unitarios en el orden correspondiente a los valores propios de (12.25), y use

$$u_i = \begin{cases} +v_i \text{ si } \lambda_i \geq 0 \\ -v_i \text{ si } \lambda_i < 0 \end{cases}. \tag{12.26}$$

El cambio de signo en (12.26) compensa cualquier signo negativo perdido al tomar los valores absolutos (12.25).

► **EJEMPLO 12.7** Encuentre los valores singulares y los vectores singulares de

$$A = \begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix}. \tag{12.27}$$

Los pares de valor propio/vector propio son $2, [1, 2]^T$ y $-\frac{1}{2}, [-2, 1]^T$. Se define v_i a partir de los vectores propios unitarios y las u_i a partir de (12.26):

$$\begin{aligned} Av_1 &= A\begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = 2\begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = s_1 u_1 \\ Av_2 &= A\begin{bmatrix} \frac{2}{\sqrt{5}} \\ -\frac{1}{\sqrt{5}} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix} = s_2 u_2. \end{aligned} \tag{12.28}$$

La DVS es

$$\begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}. \tag{12.29}$$

Observe que era necesario cambiar el signo para definir u_2, según lo prescrito en (12.26). ◄

12.3 Ejercicios

1. Encuentre la DVS de las siguientes matrices simétricas mediante un cálculo manual, y describa geométricamente la acción de la matriz sobre el círculo unitario:

(a) $\begin{bmatrix} -3 & 0 \\ 0 & 2 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}$ (c) $\begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{bmatrix}$

(d) $\begin{bmatrix} -\frac{3}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{3}{2} \end{bmatrix}$ (e) $\begin{bmatrix} 0.75 & 1.25 \\ 1.25 & 0.75 \end{bmatrix}$

2. Encuentre la DVS de las siguientes matrices mediante un cálculo manual:

(a) $\begin{bmatrix} 3 & 0 \\ 4 & 0 \end{bmatrix}$ (b) $\begin{bmatrix} 6 & -2 \\ 8 & \frac{3}{2} \end{bmatrix}$ (c) $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

(d) $\begin{bmatrix} -4 & -12 \\ 12 & 11 \end{bmatrix}$ (e) $\begin{bmatrix} 0 & -2 \\ -1 & 0 \end{bmatrix}$

3. La DVS no es única. ¿Cuántas DVS diferentes existen para el ejemplo 12.4? Haga una lista.
4. (a) Demuestre que las u_i definidas como en el teorema 12.11 son vectores propios de $A\,A^T$. (b) Demuestre que las u_i son vectores unitarios. (c) Pruebe que forman una base ortonormal de R^m.

12.4 APLICACIONES DE LA DVS

En esta sección se reúnen algunas propiedades útiles de la DVS y se señalan algunos de sus usos generalizados. Por ejemplo, la DVS resulta ser la mejor manera de encontrar el rango de una matriz. El determinante y la inversa de una matriz cuadrada, si existen, pueden encontrarse a partir de la DVS. Tal vez las aplicaciones más útiles de la DVS se derivan de la propiedad de aproximación de bajo rango.

12.4.1 Propiedades de la DVS

Suponga en lo sucesivo que $A = USV^T$ es la descomposición de valor singular. El **rango** de una matriz A de $m \times n$ es el número de renglones (o en forma equivalente, de columnas) linealmente independientes.

Propiedad 1 El rango de la matriz $A = USV^T$ es el número de entradas distintas de cero en S.

Demostración. Como U y V^T son matrices invertibles, rango(A) = rango(S), y la segunda es el número de entradas diagonales distintas de cero. ❐

Propiedad 2 Si A es una matriz de $n \times n$, $|\det(A)| = s_1 \ldots s_n$.

Demostración. Como $U^T U = I$ y $V^T V = I$, los determinantes de U y V^T son 1 o -1, debido a que el determinante de un producto es igual al producto de los determinantes, la propiedad 2 se desprende de la factorización $A = USV^T$. ❐

Propiedad 3 Si A es una matriz invertible de $m \times m$, entonces $A^{-1} = V\,S^{-1}\,U^T$.

Demostración. Por la propiedad 1, S es invertible, lo que significa que toda $s_i > 0$. Ahora la propiedad 3 se deduce del hecho de que si A_1, A_2 y A_3 son matrices invertibles, entonces $(A_1 A_2 A_3)^{-1} = A_3^{-1} A_2^{-1} A_1^{-1}$.

Por ejemplo, la DVS

$$\begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

a partir de (12.29) muestra que la matriz inversa es

$$\begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ -\frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} = \begin{bmatrix} -\frac{3}{2} & 1 \\ 1 & 0 \end{bmatrix}. \quad (12.30)$$

Propiedad 4 La matriz A de $m \times n$ puede escribirse como la suma de matrices con rango uno

$$A = \sum_{i=1}^{r} s_i u_i v_i^T, \tag{12.31}$$

donde r es el rango de A, y u_i y v_i son las i-ésimas columnas de U y V, respectivamente.

Demostración.

$$A = USV^T = U \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_r \end{bmatrix} V^T$$

$$= U \left(\begin{bmatrix} s_1 & & \\ & & \\ & & \end{bmatrix} + \begin{bmatrix} & & \\ & s_2 & \\ & & \end{bmatrix} + \cdots + \begin{bmatrix} & & \\ & & \\ & & s_r \end{bmatrix} \right) V^T$$

$$= s_1 u_1 v_1^T + s_2 u_2 v_2^T + \cdots + s_r u_r v_r^T$$

La propiedad 4 es la propiedad de aproximación de bajo rango de la DVS. La mejor aproximación a A por mínimos cuadrados del rango $p \leq r$ se obtiene al retener los primeras p términos de (12.31).

► **EJEMPLO 12.8** Encuentre la mejor aproximación de rango uno a la matriz $\begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix}$.

Al escribir (12.31) se obtiene

$$\begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \left(\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \right) \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

$$= 2 \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -\frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{8}{5} \end{bmatrix} + \begin{bmatrix} -\frac{2}{5} & \frac{1}{5} \\ \frac{1}{5} & -\frac{1}{10} \end{bmatrix}. \tag{12.32}$$

Observe cómo la matriz original se separa en una contribución mayor más una contribución más pequeña, debido a los diferentes tamaños de los valores singulares. La mejor aproximación de rango uno a la matriz está dada por la primera matriz de rango uno

$$\begin{bmatrix} \frac{2}{5} & \frac{4}{5} \\ \frac{4}{5} & \frac{8}{5} \end{bmatrix},$$

mientras que la segunda matriz proporciona pequeñas correcciones. Ésta es la idea principal detrás de la reducción de dimensión y las aplicaciones de compresión de la DVS. ◄

En las dos secciones siguientes se presentan dos usos muy relacionados para la DVS. En la reducción de dimensión, la atención se centra en la aproximación de una gran colección de vectores multidimensionales mediante un conjunto de vectores que abarcan menos dimensiones. La otra aplicación es la compresión con pérdidas, reduciendo la cantidad de información necesaria para representar en forma aproximada una matriz. Ambas aplicaciones se basan en la propiedad 4 concerniente a la aproximación de bajo rango.

12.4.2 Reducción de dimensión

La idea es proyectar datos en una dimensión inferior. Suponga que $a_1, \ldots, a_n$ comprenden una colección de vectores m-dimensionales. En aplicaciones ricas en datos, m es bastante menor que n. El objetivo de la reducción de dimensión es reemplazar $a_1, \ldots, a_n$ con n vectores que abarquen $p < m$ dimensiones, mientras se minimice el error asociado con ello. Por lo general, se inicia con un conjunto de vectores con media cero. Si no es así, puede restarse la media para alcanzar esta condición y sumarla de nuevo más tarde.

La DVS proporciona una forma sencilla de llevar a cabo la reducción de dimensión. Considere los vectores como columnas de una matriz de $m \times n$ $A = [a_1] \cdots [a_n]$, y calcule la descomposición de valor singular $A = USV^T$. Sea e_j el j-ésimo vector básico elemental (sólo ceros a excepción de la j-ésima entrada 1). Entonces $Ae_j = a_j$. Utilizando la aproximación de rango p

$$A \approx A_p = \sum_{i=1}^{p} s_i u_i v_i^T$$

a partir de la propiedad 4, es posible proyectar a_j en el espacio p-dimensional generado por las columnas $u_1, \ldots, u_p$ de U mediante

$$a_j = Ae_j \approx A_p e_j. \tag{12.33}$$

Puesto que la multiplicación de una matriz por e_j sólo recoge la j-ésima columna, se puede describir de manera más eficiente el hallazgo realizado de la siguiente manera:

El espacio $\langle u_1, \ldots, u_p \rangle$ abarcado por los vectores singulares izquierdos $u_1, \ldots, u_p$ es el mejor subespacio de aproximación en la dimensión p a $a_1, \ldots, a_n$ en el sentido de los mínimos cuadrados, y las proyecciones ortogonales de las columnas a_i de A en este espacio son las columnas de A_p. En otras palabras, la proyección de una colección de vectores $a_1, \ldots, a_n$ a su mejor subespacio p-dimensional por mínimos cuadrados es precisamente la mejor matriz de aproximación A_p de rango p.

► EJEMPLO 12.9 Encuentre el mejor subespacio unidimensional que ajuste los vectores [3, 2], [2, 4], [−2, −1], [−3, −5].

Los cuatro vectores, que se muestran en la figura 12.4(a), apuntan aproximadamente a lo largo del mismo subespacio unidimensional. Se desea encontrar este subespacio, el que minimiza la suma de los errores cuadráticos de la proyección de los vectores en ese subespacio, y luego encontrar los vectores proyectados.

Use los vectores s como columnas de la matriz

$$A = \begin{bmatrix} 3 & 2 & -2 & -3 \\ 2 & 4 & -1 & -5 \end{bmatrix},$$

y encuentre su DVS, que es

$$\begin{bmatrix} 0.5886 & -0.8084 \\ 0.8084 & 0.5886 \end{bmatrix} \begin{bmatrix} 8.2809 & 0 & 0 & 0 \\ 0 & 1.8512 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.4085 & 0.5327 & -0.2398 & -0.7014 \\ -0.6741 & 0.3985 & 0.5554 & -0.2798 \\ 0.5743 & -0.1892 & 0.7924 & -0.0801 \\ 0.2212 & 0.7223 & 0.0780 & 0.6507 \end{bmatrix},$$

con cuatro decimales. El mejor subespacio unidimensional, que se muestra como una línea punteada en la figura 12.4(b), está dado por $u_1 = [0.5886, 0.8084]$. La reducción a un subespacio de

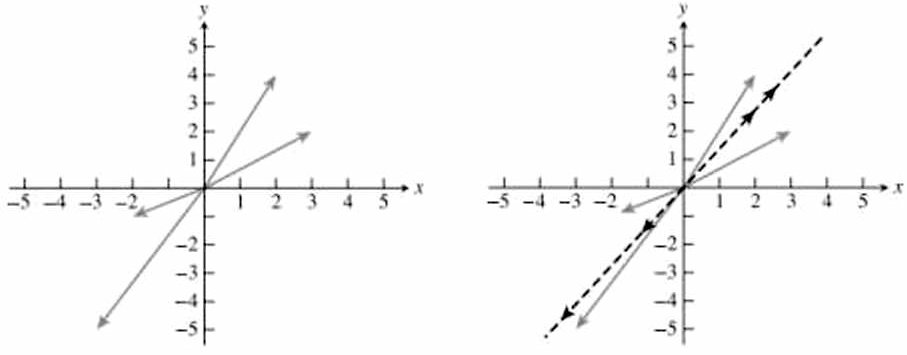


Figura 12.4 Reducción de dimensión mediante la DVS. (a) Cuatro vectores que deben proyectarse al mejor subespacio unidimensional. (b) La línea punteada representa el mejor subespacio. Las puntas de flecha muestran las proyecciones ortogonales hacia el subespacio.

dimensión $p = 1$ significa establecer $s_2 = 0$ y reconstruir la matriz. En otras palabras, $A_1 = US_1V^T$, donde

$$S_1 = \begin{bmatrix} 8.2809 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Así, las columnas de

$$A_1 = \begin{bmatrix} 1.9912 & 2.5964 & -1.1689 & -3.4188 \\ 2.7346 & 3.5657 & -1.6052 & -4.6951 \end{bmatrix} \tag{12.34}$$

son los cuatro vectores proyectados correspondientes a los cuatro vectores de datos originales. Éstos se muestran en la figura 12.4(b). ◄

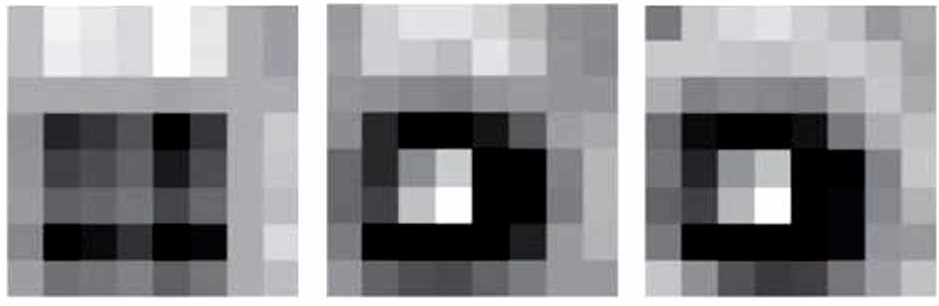
12.4.3 Compresión

La propiedad 4 también puede utilizarse para comprimir la información en una matriz. Tenga en cuenta que cada término en la expansión de rango uno de la propiedad 4 se especifica mediante dos vectores u_i, v_i y un número más s_i. Si A es una matriz de $n \times n$, es posible intentar la compresión con pérdida de A descartando los términos al final de la suma en la propiedad 4, los que tienen menor s_i. Cada término en la expansión requiere almacenar o transmitir $2n + 1$ números.

Por ejemplo, si $n = 8$, la matriz se especifica mediante 64 números, pero podría transmitir o almacenar el primer término de la expansión usando sólo $2n + 1 = 17$ números. Si la mayoría de la información está capturada por el primer término (por ejemplo, si el primer valor singular es mucho mayor que el resto), puede ahorrarse 75 por ciento del espacio al trabajar de esta manera.

Como ejemplo, regrese al bloque de 8×8 pixeles mostrado en la figura 11.6. Después de restar 128 para centrar los valores de los pixeles alrededor de 0, la matriz se da en la ecuación (11.16). Los valores singulares de esta matriz de 8×8 son los siguientes:

387.78
216.74
83.77
62.69
34.75
21.47
10.50
4.35



(a) (b) (c)

Figura 12.5 Resultado de la compresión y la descompresión mediante la DVS. Número de valores singulares retenidos: (a) $p = 1$ (b) $p = 2$ (c) todos.

El bloque original se muestra en la figura 12.5(c), junto con las versiones comprimidas en (a) y (b). La figura 12.5(a) corresponde a sustituir la matriz con el primer término de la expansión de la propiedad 4, la mejor aproximación de rango uno del valor del pixel en la matriz. Como se ha comentado antes, éste alcanza una compresión aproximada de 4:1. En la figura 12.5(b) se utilizan dos términos para una relación de compresión aproximada de 2:1. (Por supuesto, aquí se simplifica el análisis al no incluir trucos de cuantificación. Sería de ayuda llevar los coeficientes correspondientes a valores singulares más pequeños con menos precisión, como se hizo en el capítulo 11).

La imagen en escala de grises de la figura 11.5 es una imagen de 256×256 pixeles. También puede aplicarse la propiedad 4 a toda la matriz, después de restar 128 en cada entrada de píxel. Los 256 valores singulares de la matriz varían en tamaño desde 8108 hasta 0.46. La figura 12.6 muestra la imagen reconstruida que resulta de mantener la p de los términos de la expansión de rango uno en la propiedad 4. Para $p = 8$, sólo deben almacenarse $8(2(256) + 1) = 4104$ números, en comparación con $(256)^2 = 65536$ valores de píxel originales, una relación de compresión aproximada de 16:1. En la figura 12.6(c), donde se mantienen 32 términos, la relación de compresión es de aproximadamente 4:1.

12.4.4 Cálculo de la DVS

Si A es una matriz simétrica real, la DVS se reduce al cálculo de los valores propios que se analizó con anterioridad en el presente capítulo. En este caso, los vectores propios forman una base ortogonal. Si se define una matriz V que contiene los vectores propios como columnas, entonces $AV = US$ expresa la ecuación del vector propio, en donde S es una matriz diagonal que contiene los

(a) (b) (c)

Figura 12.6 Resultado de la compresión y la descompresión mediante la DVS. Número de valores singulares retenidos: (a) $p = 8$ (b) $p = 16$ (c) $p = 32$.

valores absolutos de los valores propios y U es igual a V, pero con el signo de la columna cambiado si el valor propio es negativo, según se analizó en (12.26). Como U y V son matrices ortogonales,

$$A = USV^T$$

es una descomposición de valor singular de A.

Para una matriz A general, no simétrica de $m \times n$, hay dos enfoques distintos de cálculo para determinar la DVS. El primero y más evidente es el de formar A^TA y encontrar sus valores propios. De acuerdo con el teorema 12.11, esto revela las columnas v_i de V, y al normalizar los vectores $Av_i = s_iu_i$, se obtienen tanto los valores singulares como las columnas de U.

Sin embargo, este método no se recomienda para todos los ejemplos, sólo para los sencillos. Si el número de condición A es grande, entonces el número de condición de A^TA, a menudo con una magnitud del cuadrado del número de condición de A, puede llegar a ser demasiado grande, y es posible que se pierdan dígitos de precisión.

Por fortuna, existe un método alternativo para encontrar los vectores propios de A^TA que evita la formación del producto matricial. Considere la matriz

$$B = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}. \tag{12.35}$$

Tenga en cuenta que B es una matriz simétrica de $(m + n) \times (m + n)$ (verifique su transpuesta). Por lo tanto, tiene valores propios reales y una base de vectores propios. Sea $[v, w]$ un vector con longitud $(m + n)$ que es un vector propio de B. Entonces

$$\begin{bmatrix} A^Tw \\ Av \end{bmatrix} = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} v \\ w \end{bmatrix} = \lambda\begin{bmatrix} v \\ w \end{bmatrix},$$

o $Av = \lambda w$. Al multiplicar a la izquierda por A^T se obtiene

$$A^TAv = \lambda A^Tw = \lambda^2 v, \tag{12.36}$$

lo que muestra que w es un vector propio de A^TA con el correspondiente valor propio λ^2. Observe que es posible determinar los valores propios y los vectores propios de A^TA de este modo, sin formar la matriz A^TA.

Por lo tanto, el segundo método preferido para calcular los valores y vectores singulares comienza con poner la matriz simétrica B en la forma superior de Hessenberg. Debido a la simetría, la forma superior de Hessenberg es equivalente a la forma tridiagonal. Entonces, pueden aplicarse métodos como el algoritmo QR desplazado para encontrar los valores propios, que son los cuadrados de los valores singulares, y los vectores propios, cuyas n entradas superiores son los vectores singulares v_i. Aunque este enfoque parece duplicar el tamaño de la matriz, evita aumentar el número de condición innecesariamente, y hay maneras más eficientes de poner en práctica esta idea (las cuales no se estudiarán aquí) que evitan un almacenamiento adicional.

12.4 Problemas de computadora

1. Utilice el comando svd de MATLAB para encontrar la mejor aproximación de rango uno a las matrices siguientes:

(a) $\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$ (c) $\begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 3 \\ 0 & 0 & 1 \end{bmatrix}$ (d) $\begin{bmatrix} 1 & 5 & 3 \\ 2 & -3 & 2 \\ -3 & 1 & 1 \end{bmatrix}$

2. Encuentre la mejor aproximación de rango dos a las matrices siguientes:

$$\text{(a)}\ \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 3 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 2 & -2 & 4 \\ 1 & -1 & 2 \\ -3 & 3 & -6 \end{bmatrix} \quad \text{(c)}\ \begin{bmatrix} 1 & 5 & 3 \\ 2 & -3 & 2 \\ -3 & 1 & 1 \end{bmatrix}$$

3. Encuentre la mejor línea aproximada por mínimos cuadrados para los siguientes vectores, y las proyecciones de los vectores sobre el subespacio unidimensional:

$$\text{(a)}\ \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 5 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \text{(c)}\ \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix}$$

4. Encuentre el mejor plano aproximado por mínimos cuadrados para los siguientes vectores tridimensionales, y las proyecciones de los vectores sobre el subespacio:

$$\text{(a)}\ \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 7 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

5. Escriba un programa en MATLAB que utilice la matriz de (12.35) para calcular los valores singulares de una matriz. Utilice el código superior de Hessenberg dado con anterioridad y utilice el QR desplazado para resolver el problema de valor propio resultante. Aplique su método para encontrar los valores singulares de las matrices siguientes:

$$\text{(a)}\ \begin{bmatrix} 3 & 0 \\ 4 & 0 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 6 & -2 \\ 8 & \frac{3}{2} \end{bmatrix} \quad \text{(c)}\ \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{(d)}\ \begin{bmatrix} -4 & -12 \\ 12 & 11 \end{bmatrix} \quad \text{(e)}\ \begin{bmatrix} 0 & -2 \\ -1 & 0 \end{bmatrix}$$

6. Continuando con el problema de computadora 5, agregue código para encontrar la DVS completa de las matrices.

7. Utilice el código desarrollado en el problema de computadora 6 para encontrar la DVS completa de las siguientes matrices, y compare sus resultados con el comando `svd` de MATLAB (su respuesta debe coincidir con la elección de los signos negativos en u_i, v_i):

$$\text{(a)}\ \begin{bmatrix} 1 & 3 & 0 \\ 4 & 5 & 0 \\ 2 & 5 & 3 \end{bmatrix} \quad \text{(b)}\ \begin{bmatrix} 1 & 0 & 2 & 4 \\ 1 & 1 & 1 & 3 \end{bmatrix} \quad \text{(c)}\ \begin{bmatrix} 0 & 1 & 3 \\ 1 & 3 & 1 \\ 2 & -1 & 3 \\ 0 & 1 & -1 \end{bmatrix} \quad \text{(d)}\ \begin{bmatrix} 0 & 1 & 3 & 1 \\ -1 & 1 & 1 & 0 \\ 0 & 1 & 3 & -1 \\ 2 & -1 & -1 & 2 \end{bmatrix}$$

8. Importe una fotografía usando el comando `imread` de MATLAB. Utilice la DVS para crear versiones comprimidas de la fotografía a 8:1, 4:1 y 2:1. Si la fotografía es en color, comprima cada uno de los colores RGB por separado.

Software y lecturas adicionales

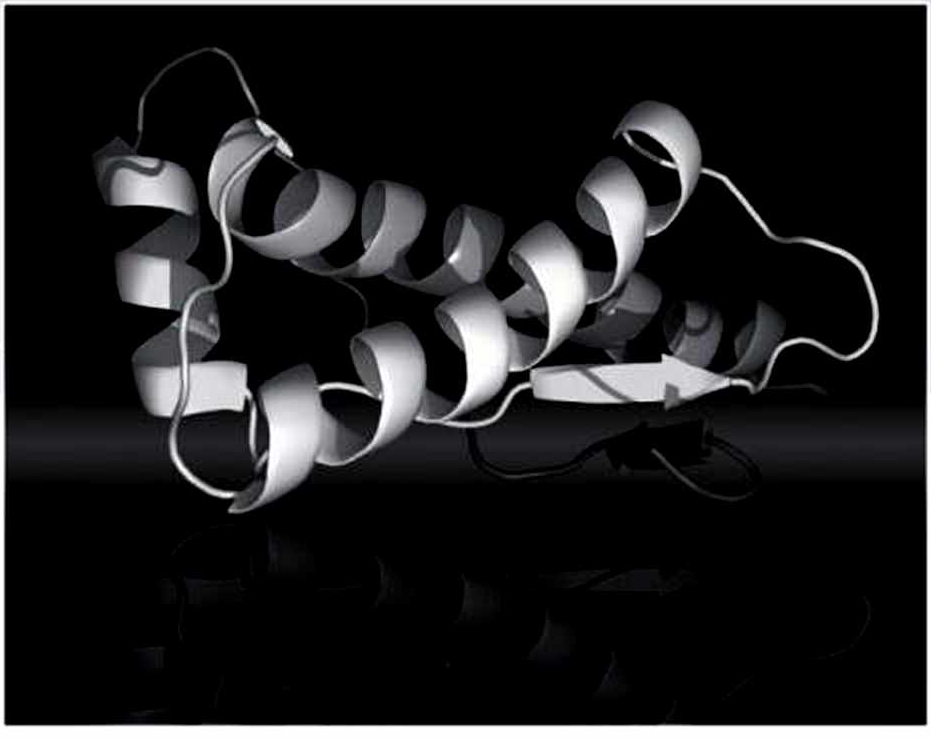
La era moderna del cálculo de valores propios se inició con Wilkinson [1965], y el algoritmo QR y la forma superior de Hessenberg ya estaban presentes en Wilkinson y Reinsch [1971]. Otras referencias importantes en el cálculo de valores propios son Stewart [1973], Parlett [1998], Golub y Van Loan [1996], y los artículos reveladores de Parlett [2000] y Watkins [1982].

Lapack (Anderson *et al.* [1990]) proporciona rutinas para las reducciones a la forma superior de Hessenberg y para el problema del valor propio simétrico y no simétrico. Estas rutinas son

descendientes del paquete Eispack (Smith *et al.* [1970]), desarrollado en la década de 1960. DGEHRD de Netlib reduce una matriz real a la forma superior de Hessenberg usando los reflectores de Householder, y DHSEQR implementa el algoritmo QR para calcular los valores propios y la forma de Schur de una matriz real superior de Hessenberg. NAG proporciona F08NEF y F08PEF, respectivamente, para las mismas dos operaciones. Hay programas análogos para las matrices complejas.

Saad [2003] y Bai *et al.* [2000] consideran los métodos más modernos para los problemas de valores propios de gran tamaño. Cuppen [1981] introdujo el método de divide y vencerás para el problema de valores propios tridiagonales simétricos. Arpack es una utilería para la iteración de Arnoldi en problemas grandes dispersos, y Parpack es una extensión para los procesadores en paralelo.

Entre los algoritmos para la descomposición de valor singular se encuentran el DGESVD, original de Lapack, y el método de divide y vencerás DGESDD que es aconsejable para las matrices de gran tamaño. Asimismo, también se cuenta con versiones complejas.



CAPÍTULO

13

Optimización

El descubrimiento de la estructura en doble hélice del ADN en 1953 ha llevado, medio siglo más tarde, a una secuencia casi completa del genoma humano. La secuencia contiene instrucciones para el plegado de cadenas de aminoácidos en proteínas individuales que realizan las actividades de la vida, pero escritas en un lenguaje codificado. Esta información está a la espera de su traducción, de modo que pueda ser dirigida hacia una comprensión detallada de la función fisiológica. Una gran cantidad de aplicaciones potenciales, incluyendo la terapia genómica y el diseño racional de medicamentos, pueden promover la prevención temprana, el diagnóstico y la cura de enfermedades.

El plegado de aminoácidos en proteínas funcionales depende en gran medida de las fuerzas de Van der Waals, la atracción y repulsión microscópica entre átomos no enlazados. Los modelos de conjuntos atómicos, donde estas fuerzas se modelan mediante el potencial de Lennard-Jones, se estudian para las configuraciones mínimas de energía, con lo que el problema recae en el ámbito de la optimización.

Comprobación en la realidad En la página 580 se aplica las técnicas de optimización del capítulo para resolver este problema de minimizar la energía.

La optimización es encontrar el máximo o mínimo de una función real, llamada **función objetivo**. Puesto que la localización del máximo de una función $f(x)$ es equivalente a localizar el mínimo de $-f(x)$, para el desarrollo de métodos computacionales basta considerar sólo el problema de la minimización.

Algunos problemas de optimización requieren un mínimo de la función objetivo sujeto a varias restricciones de igualdad y desigualdad. Por ejemplo, aunque x_1 es el mínimo global de la función de la figura 13.1, x_2 sería el mínimo sujeto a la restricción $x \geq 0$. En particular, el campo de la programación lineal considera los problemas en los que la función objetivo y las restricciones son lineales. En este capítulo, se mantendrán las cosas simples y se considerará sólo la optimización no restringida.

Los métodos para la optimización no restringida se clasifican en dos grupos, dependiendo de si usan o no las derivadas de la función objetivo $f(x)$. Si una función algebraica se conoce para $f(x)$,

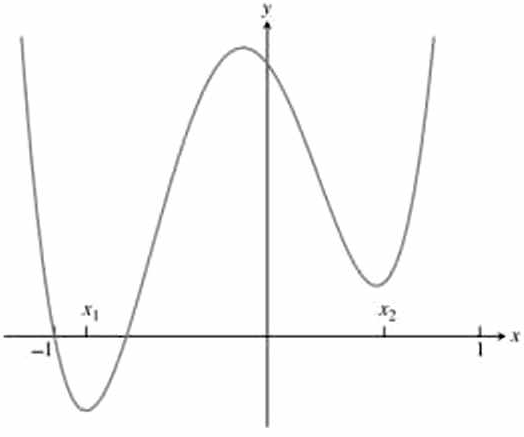


Figura 13.1 Problema de minimización para $f(x) = 5x^4 + 3x^3 - 4x^2 - x + 2$. La solución al problema de minimizar no restringido $\min_x f(x)$ es x_1.

en la mayoría de los casos, las derivadas pueden determinarse con facilidad usando el álgebra en papel o en computadora. La información de la derivada se debe utilizar siempre que sea posible, pero hay varias razones por las que podría no estar disponible. En particular, la función objetivo puede ser demasiado complicada, tener una dimensión demasiado grande o no contar con una forma conocida que sea diferenciable.

13.1 OPTIMIZACIÓN NO RESTRINGIDA SIN DERIVADAS

En esta sección se adopta el supuesto de que la función objetivo $f(x)$ puede evaluarse para cualquier entrada x, pero que la derivada $f'(x)$ (o las derivadas parciales si f es una función de varias variables) no está disponible. Se analizarán tres métodos para optimizar sin derivadas: la búsqueda de la sección dorada, la interpolación parabólica sucesiva y el método de Nelder-Mead. Los dos primeros se aplican sólo a las funciones $f(x)$ de una variable escalar, mientras que Nelder-Mead puede buscar a través de varias dimensiones.

13.1.1 Búsqueda de la sección dorada

La búsqueda de la sección dorada es un método eficiente para la búsqueda de un mínimo de una función $f(x)$ de una variable, una vez que se conoce un intervalo de confinamiento.

DEFINICIÓN 13.1 La función continua $f(x)$ se llama **unimodal** en el intervalo $[a, b]$ si hay justo un mínimo o un máximo relativo en $[a, b]$ y f es estrictamente decreciente o creciente en todos los demás puntos. ❐

Una función unimodal se incrementa hasta un máximo relativo en $[a, b]$ y después disminuye a medida que x se mueve desde a hasta b, o bien se reduce hasta un mínimo relativo y después se incrementa.

Suponga que f es unimodal y tiene un mínimo relativo en $[a, b]$. Elija dos puntos x_1 y x_2 al interior del intervalo, tal que $a < x_1 < x_2 < b$, como se muestra en la figura 13.2 para el caso $[a, b] = [0, 1]$. Se reemplazará el intervalo original por uno nuevo más pequeño que sigue confinando un mínimo relativo, de acuerdo con la siguiente regla: si $f(x_1) \leq f(x_2)$, entonces conserve el intervalo $[a, x_2]$ en el paso siguiente. Si $f(x_1) > f(x_2)$, retenga el intervalo $[x_1, b]$.

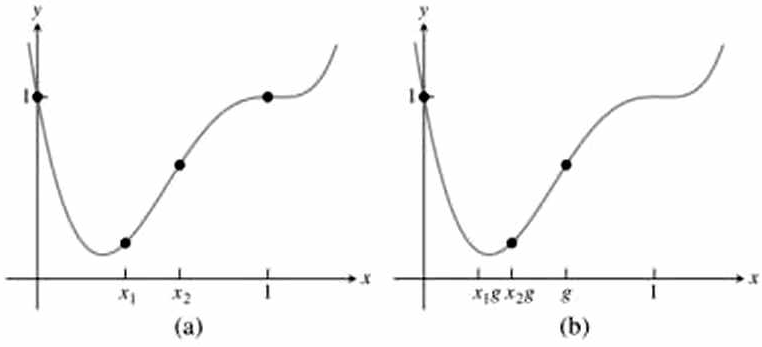


Figura 13.2 Búsqueda de la sección dorada. (a) Evalúe la función objetivo en dos puntos x_1, x_2 en el intervalo actual [0, 1]. Si $f(x_1) \leq f(x_2)$, entonces el nuevo intervalo será $[0, x_2]$. (b) En el siguiente paso, establezca $g = x_2$ y repita la misma comparación con x_1g y x_2g.

Observe que en cualquier caso el nuevo intervalo contiene un mínimo relativo de la función unimodal f. Por ejemplo, si $f(x_1) < f(x_2)$, como se muestra en la figura 13.2, esto debido a la suposición unimodal, el mínimo debe estar a la izquierda de x_2. Lo anterior se debe a que f debe disminuir a la izquierda del mínimo, por lo que $f(x_1) < f(x_2)$ significa que x_2 debe estar a la derecha del mínimo. Del mismo modo, $f(x_1) > f(x_2)$ implica que $[x_1, b]$ contiene el mínimo. Como el nuevo intervalo es menor que el intervalo anterior $[a, b]$, se ha tenido un progreso hacia la localización del mínimo. Este paso básico se repite hasta que el intervalo que contiene al mínimo sea tan pequeño como se desea. El método es una reminiscencia del método de bisección para la localización de raíces.

A continuación se discute cómo deben colocarse x_1 y x_2 en el intervalo $[a, b]$. En cada paso, se desea reducir la longitud del intervalo tanto como sea posible realizando el menor esfuerzo. La forma de hacer esto se muestra en la figura 13.3 para el intervalo $[a, b] = [0, 1]$. Acepte dos criterios para la elección de x_1 y x_2: (a) Haga que sean simétricas con respecto al intervalo (puesto que no se tiene ninguna información sobre el lado del intervalo donde se encuentra el mínimo), y (b) elíjalos sin importar cuál nuevo intervalo se seleccione, x_1 y x_2 se utilicen en el paso siguiente. Es decir, requiera (a) $x_1 = 1 - x_2$ y (b) $x_1 = x_2^2$. Como se muestra en la figura 13.3, si el nuevo intervalo es $[0, x_2]$, el criterio (b) garantiza que la x_1 original será la "x_2" para el intervalo siguiente; por lo tanto, sólo será necesaria una nueva evaluación de la función, a saber, $f(x_1g)$. Del mismo modo, si el nuevo intervalo es $[x_1, 1]$, entonces x_2 se convertirá en la nueva "x_1". Esta capacidad de reutilizar las evaluaciones de la función implica que después de la primera etapa, sólo se necesita una sola evaluación de la función objetivo por paso.

Los criterios (a) y (b) en conjunto implican que $x_2^2 + x_2 - 1 = 0$. La solución positiva de esta ecuación cuadrática es $x_2 = g = (\sqrt{5} - 1)/2$. Para iniciar el método, debe saberse que la función objetivo f es unimodal sobre $[a, b]$, y después f se evalúa en los puntos interiores x_1 y x_2, donde $a < x_1 = a + (1 - g)(b - a) < x_2 = a + g(b - a) < b$. Tenga en cuenta que x_1 y x_2 se establece exactamente en $1 - g$ y g entre a y b. El nuevo intervalo se escoge como se ha mostrado y este paso básico se repite. El nuevo intervalo tiene una longitud de g veces el intervalo anterior, por lo que después de k pasos el intervalo actual tiene una longitud $g^k(b - a)$. El punto medio del intervalo final es correcto dentro de una incertidumbre de la mitad de la longitud del intervalo final, $g^k(b - a)/2$. Se ha demostrado el siguiente teorema:

TEOREMA 13.2 Después de k pasos de la búsqueda de la sección dorada con intervalo inicial $[a, b]$, el punto medio del intervalo final está a una distancia máxima $g^k(b - a)/2$ del mínimo, donde $g = (\sqrt{5} - 1)/2 \approx 0.618$. ■

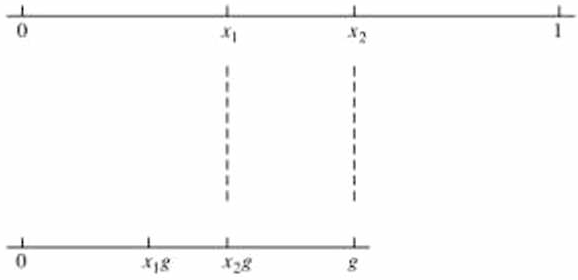


Figura 13.3 Selección de proporciones en la búsqueda de la sección dorada. La relación del segmento superior sobre el segmento inferior es $1/g = (1+\sqrt{5})/2$, la **sección dorada**. Los puntos x_1 y x_2 se eligen exactamente, de modo que sin importar si el nuevo intervalo es $[0, x_2]$ o $[x_1, 1]$, uno de los puntos puede reutilizarse como un nuevo punto interior, lo que reduce el número de nuevas evaluaciones de la función objetivo a una por cada paso.

Búsqueda de la sección dorada

Dada una f unimodal con un mínimo en $[a, b]$

for $i = 1, 2, 3, \ldots$
  $g = (\sqrt{5} - 1)/2$
  if $f(a + (1 - g)(b - a)) < f(a + g(b - a))$
    $b = a + g(b - a)$
  else
    $a = a + (1 - g)(b - a)$
  end
end

El intervalo final $[a, b]$ contiene un mínimo.

El código de MATLAB para la búsqueda de la sección dorada requiere una evaluación de la función por cada paso después del primero de ellos, como se mencionó antes.

```
% Programa 13.1 Búsqueda de la sección dorada para un mínimo de f(x)
% Inicio con f(x) unimodal y el mínimo en [a,b]
% Entrada: función f, intervalo [a,b], número de pasos k
% Salida: mínimo aproximado y
function y=gss(f,a,b,k)
g=(sqrt(5)-1)/2;
x1 = a+(1-g)*(b-a);
x2 = a+g*(b-a);
f1=f(x1);f2=f(x2);
for i=1:k
  if f1 < f2          % si f(x1) < f(x2), reemplaza b por x2
    b=x2; x2=x1; x1=a+(1-g)*(b-a);
    f2=f1; f1=f(x1);  % una sola evaluación de la función
  else                % en caso contrario, reemplaza a por x1
    a=x1; x1=x2; x2=a+g*(b-a);
    f1=f2; f2=f(x2);  % una sola evaluación de la función
  end
end
y=(a+b)/2;
```

ANOTACIÓN

Convergencia De acuerdo con el teorema 13.2, la búsqueda de la sección dorada converge linealmente al mínimo con una razón de convergencia lineal $g \approx 0.618$. Es interesante destacar las grandes similitudes de este método con el método de bisección del capítulo 1 para la búsqueda de raíces. A pesar de que resuelven problemas diferentes, ambos son globalmente convergentes, lo que significa que si se inicia con las condiciones adecuadas (unimodalidad en $[a, b]$ para la búsqueda de la sección dorada, y $f(a)\, f(b) < 0$ para la bisección), ambos garantizan su convergencia a una solución. Ninguno de los dos requiere información de las derivadas. Ambos requieren una evaluación de la función por paso y ambos son linealmente convergentes. La bisección es un poco más rápida, con la razón de convergencia lineal $K = 0.5 < g = 0.618$. Ambos pertenecen a la categoría valiosa de los métodos "lentos, pero seguros".

► **EJEMPLO 13.1** Utilice la búsqueda de la sección dorada para encontrar el mínimo de $f(x) = x^6 - 11x^3 + 17x^2 - 7x + 1$ en el intervalo $[0, 1]$.

En la figura 13.2 se muestran los dos primeros pasos del método. En el primer paso $x_1 = 1 - g$ y $x_2 = g$, donde $g = (\sqrt{5} - 1)/2$. Como $f(x_1) < f(x_2)$, el intervalo $[0, 1]$ se sustituye por $[0, g]$. Las nuevas x_1, x_2 son las anteriores, $x_1 g$, $x_2 g$, respectivamente. En el segundo paso, de nuevo $f(x_1) < f(x_2)$, por lo que el intervalo $[0, g]$ se sustituye con $[0, x_2]$. En la siguiente tabla se muestran las primeras 15 iteraciones:

iteración	a	x_1	x_2	b
0	0.0000	0.3820	0.6180	1.0000
1	0.0000	0.2361	0.3820	0.6180
2	0.0000	0.1459	0.2361	0.3820
3	0.1459	0.2361	0.2918	0.3820
4	0.2361	0.2918	0.3262	0.3820
5	0.2361	0.2705	0.2918	0.3262
6	0.2705	0.2918	0.3050	0.3262
7	0.2705	0.2837	0.2918	0.3050
8	0.2705	0.2786	0.2837	0.2918
9	0.2786	0.2837	0.2868	0.2918
10	0.2786	0.2817	0.2837	0.2868
11	0.2817	0.2837	0.2849	0.2868
12	0.2817	0.2829	0.2837	0.2849
13	0.2829	0.2837	0.2841	0.2849
14	0.2829	0.2834	0.2837	0.2841
15	0.2834	0.2837	0.2838	0.2841

Después de 15 iteraciones, puede decirse que el mínimo está entre 0.2834 y 0.2838. ◄

13.1.2 Interpolación parabólica sucesiva

En la búsqueda de la sección dorada, las evaluaciones de la función $f(x_1)$ y $f(x_2)$ sólo se usan con fines de comparación. Se toma una decisión sobre cómo proceder, sin importar que una evaluación sea más grande que la otra. En esta sección, se describe un nuevo método que economiza la evaluación de los valores de la función; los utiliza para construir un modelo local de la función f.

El modelo local elegido es una parábola que, como se vio en el capítulo 3, puede determinarse sólo con tres puntos. Se inicia con tres puntos r, s y t en las proximidades del mínimo, como se muestra en la figura 13.4. Evalúe la función objetivo f en los tres puntos y dibuje la parábola que pasa a través de ellos. Las diferencias divididas son

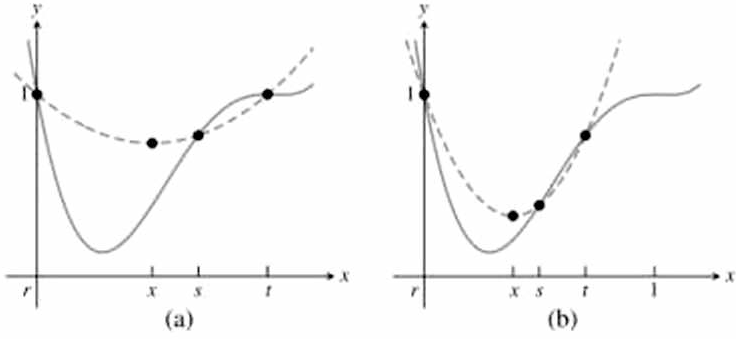


Figura 13.4 Interpolación parabólica sucesiva. (a) Se dibuja una parábola a través de los tres puntos actuales *r, s, t*, y la *x* mínima de la parábola se usa para sustituir la *s* actual. (b) El paso se repite con las nuevas *r, s, t*.

$$\begin{array}{c|cccc} r & f(r) & & & \\ & & d_1 & & \\ s & f(s) & & d_3 & \\ & & d_2 & & \\ t & f(t) & & & \end{array}$$

donde $d_1 = (f(s) - f(r))/(s - r)$, $d_2 = (f(t) - f(s))/(t - s)$, y $d_3 = (d_2 - d_1)/(t - r)$. Por lo tanto, la parábola puede expresarse como

$$P(x) = f(r) + d_1(x - r) + d_3(x - r)(x - s). \tag{13.1}$$

Si se establece la derivada de $P(x) = 0$ para encontrar el mínimo de la parábola se obtiene la fórmula

$$x = \frac{r + s}{2} - \frac{(f(s) - f(r))(t - r)(t - s)}{2[(s - r)(f(t) - f(s)) - (f(s) - f(r))(t - s)]} \tag{13.2}$$

En la nueva aproximación para el mínimo. En la IPS, la nueva x puede sustituir al punto menos reciente o menos óptimo entre r, s y t, y el paso se repite según sea necesario. No hay ninguna garantía de convergencia de la IPS, a diferencia de la búsqueda de la sección dorada. Sin embargo, cuando converge suele hacerlo más rápido, puesto que utiliza la información de la evaluación de la función de manera más eficiente.

Interpolación parabólica sucesiva

Inicie con los mínimos aproximados r, s, t

for $i = 1, 2, 3, \ldots$

$$x = \frac{r + s}{2} - \frac{(f(s) - f(r))(t - r)(t - s)}{2[(s - r)(f(t) - f(s)) - (f(s) - f(r))(t - s)]}$$

$t = s$

$s = r$

$r = x$

end

En el siguiente código de MATLAB, el mínimo de la parábola sustituye al menos reciente de los tres puntos actuales:

```
% Programa 13.2 Interpolación parabólica sucesiva
% Entrada: función f, estimaciones iniciales r, s, t, pasos k
% Salida: mínimo aproximado x
```

```
function x=spi(f,r,s,t,k)
x(1)=r;x(2)=s;x(3)=t;
fr=f(r);fs=f(s);ft=f(t);
for i=4:k+3
 x(i)=(r+s)/2-(fs-fr)*(t-r)*(t-s)/(2*((s-r)*(ft-fs)
      -(fs-fr)*(t-s)));
 t=s;s=r;r=x(i);
 ft=fs;fs=fr;fr=f(r);            % una sola evaluación de la función
end
```

► EJEMPLO 13.2 Use la interpolación parabólica sucesiva para encontrar el mínimo de $f(x) = x^6 - 11x^3 + 17x^2 - 7x + 1$ en el intervalo [0, 1].

Usando los puntos de inicio $r = 0, s = 0.7, t = 1$, se calculan las siguientes aproximaciones:

iteración	x	$f(x)$
0	1.00000000000000	1.00000000000000
0	0.70000000000000	0.77464900000000
0	0.00000000000000	1.00000000000000
1	0.50000000000000	0.39062500000000
2	0.38589683548538	0.20147287814500
3	0.33175129602524	0.14844165724673
4	0.23735573316721	0.14933737764402
5	0.28526617269372	0.13172660338164
6	0.28516942161639	0.13172426136234
7	0.28374069464218	0.13170646451792
8	0.28364647631123	0.13170639859035
9	0.28364826437569	0.13170639856301
10	0.28364835832962	0.13170639856295
11	0.28364835808377	0.13170639856295
12	0.28364833218729	0.13170639856295

Se concluye que el mínimo está cerca de $x_{\text{mín}} = 0.2836483$. Tenga en cuenta que después de 12 iteraciones se ha superado mucho la precisión de la búsqueda de la sección dorada con un menor número de evaluaciones de la función. No se ha utilizado ninguna información sobre las derivadas de la función objetivo, aunque se ha utilizado el conocimiento de los valores exactos de f, mientras que para la BSD sólo era necesario conocer las comparaciones entre los valores.

Observe también en la tabla, una curiosidad cercana al final. Como se analizó en el capítulo 1, las funciones son muy planas cerca de los máximos y mínimos relativos. Los valores calculados con una aproximación de 10^{-7} de $x_{\text{mín}}$ dan el mismo valor en la función, no es posible una mayor precisión cuando se usa la IEEE de doble precisión, sin importar cuántos pasos puedan ejecutarse. Ya que los mínimos ocurren por lo regular en los puntos donde las derivadas de la función son iguales a cero, esta dificultad no se atribuye al método de optimización, sino que es endémica del cálculo en punto flotante.

La secuencia de la BSD a la IPS es similar a la del método de bisección, al método de la secante y a la interpolación cuadrática inversa. La construcción de un modelo local de la función y su consideración como la función objetivo ayudan a acelerar la convergencia. ◄

13.1.3 Búsqueda de Nelder-Mead

Para una función de más de una variable, los métodos se vuelven más sofisticados. La búsqueda de Nelder-Mead intenta rodar un poliedro cuesta abajo hasta el nivel más bajo posible. Por esta razón, también se le llama el **método simplex cuesta abajo**. No utiliza información de la derivada sobre la función objetivo.

Suponga que la función a minimizar es una función f de n variables. El método comienza con $n + 1$ vectores estimados iniciales $x_1, \ldots, x_{n+1}$ que pertenecen a R^n y juntos forman los vértices de un simplex n-dimensional. Por ejemplo, si $n = 2$, las tres aproximaciones iniciales forman los vértices de un triángulo en el plano.

Los vértices del simplex se prueban y se colocan en orden ascendente de acuerdo con sus valores de la función $y_1 < y_2 < \cdots < y_{n+1} = y_h$. El vector simplex $x_h = x_{n+1}$ que es el menos óptimo se sustituye de acuerdo con el diagrama de flujo mostrado en la figura 13.5. En primer lugar se define el centroide $\overline{x}$ de la cara del simplex que omite x_h. Después se prueba el valor de la función $y_r = f(x_r)$ del punto de reflexión $x_r = 2\overline{x} - x_h$, tal como se muestra en la figura 13.5(a). Si el nuevo valor y_r está en el rango $y_1 < y_r < y_n$, se reemplaza el peor punto x_n con x_r, se ordenan los vértices de acuerdo con sus valores de la función y se repite el paso.

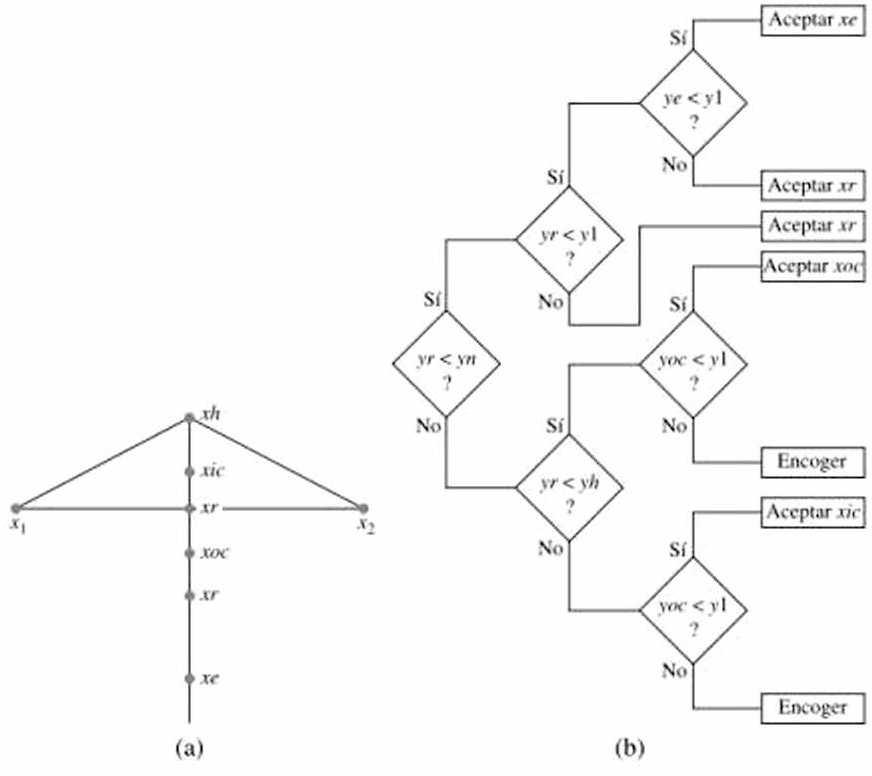


Figura 13.5 Búsqueda de Nelder-Mead. (a) Se prueban los puntos a lo largo de la línea que conecta al punto más alto de la función x_h y el centroide $\overline{x}$. (b) Diagrama de flujo que describe un paso del método.

En caso de que y_r sea inferior al mínimo actual y_1, se hace un intento de extrapolación, utilizando $x_e = 3\overline{x} - 2x_h$, para ver si debe avanzar aún más en esta dirección. Se acepta el mejor entre x_e y x_r para el paso. Por otro lado, en caso de que y_r sea mayor que y_n (el máximo actual una vez que x_{n+1} se descarta), se realiza una prueba adicional, ya sea en el punto de contracción exterior $x_{oc} = 1.5\overline{x} - 0.5x_h$ o en el punto de contracción interior $x_{ic} = 0.5\overline{x} + 0.5x_h$, como se muestra en la figura. Si no se presenta mejoría en cualquiera de estos puntos, entonces no hay avance al ramificar hacia afuera y el método debe mirar más a nivel local para encontrar el óptimo. Esto se logra mediante la reducción del simplex por un factor de 2 en la dirección del mínimo actual x_1 antes de ir al paso siguiente. A continuación se presenta el código de MATLAB. La función f debe definirse en las variables `x(1), x(2), ..., x(n)`.

```
% Programa 13.3 Búsqueda de Nelder-Mead
% Entrada: función f, mejor estimación xbar (vector columna),
%    radio de búsqueda inicial rad y número de pasos k
% Salida: matriz x cuyas columnas son los vértices de un simplex,
%    valores de la función y de esos vértices
function [x,y]=neldermead(f,xbar,rad,k)
n=length(xbar);
x(:,1)=xbar;          % cada columna de x es un vértice simplex
x(:,2:n+1)=xbar*ones(1,n)+rad*eye(n,n);
for j=1:n+1
  y(j)=f(x(:,j));     % evalúa la función obj f en cada vértice
end
[y,r]=sort(y);        % ordena los valores de la función en forma ascendente
x=x(:,r);             %      y clasifica los vértices de la misma forma
for i=1:k
  xbar=mean(x(:,1:n)')';  % xbar es el centroide de la cara
  xh=x(:,n+1);            %   omitiendo el peor vértice xh
  xr = 2*xbar - xh; yr = f(xr);
  if yr < y(n)
    if yr < y(1)      % intenta la expansión xe
      xe = 3*xbar - 2*xh; ye = f(xe);
      if ye < yr      % acepta la expansión
        x(:,n+1) = xe; y(n+1) = f(xe);
      else            % acepta la reflexión
        x(:,n+1) = xr; y(n+1) = f(xr);
      end
    else              % xr es la mitad del paquete, acepta la reflexión
      x(:,n+1) = xr; y(n+1) = f(xr);
    end
  else                % xr sigue siendo el peor vértice, contraer
    if yr < y(n+1)    % intenta contracción externa xoc
      xoc = 1.5*xbar - 0.5*xh; yoc = f(xoc);
      if yoc < yr     % acepta la contracción externa
        x(:,n+1) = xoc; y(n+1) = f(xoc);
      else            % encoge el simplex hacia el mejor punto
        for j=2:n+1
          x(:,j) = 0.5*x(:,1)+0.5*x(:,j); y(j) = f(x(:,j));
        end
      end
    else              % xr es aún peor que el peor anterior
      xic = 0.5*xbar+0.5*xh; yic = f(xic);
      if yic < y(n+1) % acepta la contracción interna
        x(:,n+1) = xic; y(n+1) = f(xic);
      else            % encoge el simplex hacia el mejor punto
        for j=2:n+1
          x(:,j) = 0.5*x(:,1)+0.5*x(:,j); y(j) = f(x(:,j));
        end
      end
    end
  end
  [y,r] = sort(y);    % reordena los valores de la función obj
  x=x(:,r);           %     y clasifica los vértices de la misma manera
end
```

El código representa al diagrama de flujo de la figura 13.5(b). Se requiere como entrada, el número de pasos de iteración. El problema de computadora 8 le pide al lector que vuelva a escribir el código con un criterio de terminación basado en la tolerancia al error dada por el usuario. Un

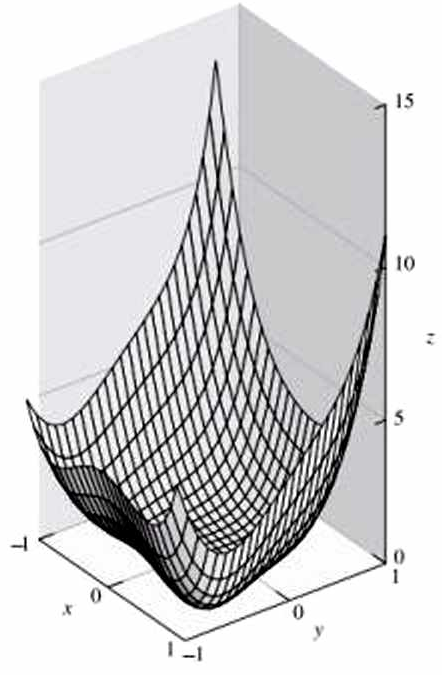


Figura 13.6 Gráfica de superficie de una función bidimensional. Gráfica de $z = 5x^4 + 4x^2y - xy^3 + 4y^4 - x$. Mediante el método de Nelder-Mead, se encuentra que el mínimo se produce en $\approx (0.4923, -0.3643)$.

criterio de terminación común consiste en requerir que el tamaño del simplex se haya reducido dentro de una pequeña cantidad de cifras, y que la extensión máxima de los valores de la función en los vértices esté dentro de una tolerancia pequeña. MATLAB implementa el método de Nelder-Mead mediante su comando `fminsearch`.

► **EJEMPLO 13.3** Localice el mínimo de la función $f(x, y) = 5x^4 + 4x^2y - xy^3 + 4y^4 - x$, usando el método de Nelder-Mead.

La función se muestra en la figura 13,6. La función f de dos variables se define mediante

```
>> f=@(x) 5*x(1)^4+4*x(1)^2*x(2)-x(1)*x(2)^3+4*x(2)^4-x(1)
```

y se ejecutan 60 iteraciones del método de Nelder-Mead en el programa 13.3 con el comando

```
>> [x,y]=neldermead(f,[1;1],1,60)

x =

   0.492307778751573   0.492307773822840   0.492307807617628
  -0.364285558245531  -0.364285542189284  -0.364285562179872

y =

  -0.457521622634071  -0.457521622634070  -0.457521622634069
```

Se utilizó el vector $[x, y] = [1, 1]$ como la estimación inicial y un radio inicial de 1, pero hay una amplia variedad de opciones que pueden funcionar. Después de 60 iteraciones, el simplex se ha reducido a un triángulo cuyos vértices son las tres columnas del vector de salida x. Hasta cuatro cifras decimales correctas, el mínimo de -0.4575 ocurre en el punto $[x, y] = [0.4923, -0.3643]$. ◄

13.1 Ejercicios

1. Demuestre que las funciones son unimodales en algunos intervalos, encuentre el mínimo absoluto y dónde se produce. (a) $f(x) = e^x + e^{-x}$ (b) $f(x) = x^6$ (c) $f(x) = 2x^4 + x$ (d) $f(x) = x - \ln x$.

2. Encuentre el mínimo absoluto en los intervalos dados y en qué x se produce.
   (a) $f(x) = \cos x, [3, 4]$ (b) $f(x) = 2x^3 + 3x^2 - 12x + 3, [0, 2]$
   (c) $f(x) = x^3 + 6x^2 + 5, [-5, 5]$ (d) $f(x) = 2x + e^{-x}, [-5, 5]$

13.1 Problemas de computadora

1. Grafique la función $y = f(x)$ y encuentre un intervalo de inicio de longitud uno sobre el cual f sea unimodal en torno a cada mínimo relativo. Después aplique la búsqueda de la sección dorada para localizar cada uno de los mínimos relativos de la función con cinco dígitos correctos.
   (a) $f(x) = 2x^4 + 3x^2 - 4x + 5$ (b) $f(x) = 3x^4 + 4x^3 - 12x^2 + 5$
   (c) $f(x) = x^6 + 3x^4 - 2x^3 + x^2 - x - 7$ (d) $f(x) = x^6 + 3x^4 - 12x^3 + x^2 - x - 7$

2. Aplique la interpolación parabólica sucesiva a las funciones del problema de computadora 1. Localice los mínimos con cinco dígitos correctos.

3. Encuentre el punto sobre la hipérbola $y = 1/x$ más cercano al punto (2, 3) de dos maneras diferentes: (a) por medio del método de Newton aplicado para encontrar un punto crítico, (b) mediante la búsqueda de la sección dorada sobre el cuadrado de la distancia entre un punto del cono y (2, 3).

4. Encuentre el punto sobre la elipse $4x^2 + 9y^2 = 4$ más alejado de (1, 5), utilizando los métodos (a) y (b) del problema de computadora 3.

5. Use el método de Nelder-Mead para encontrar el mínimo de
   $f(x, y) = e^{-x^2y^2} + (x - 1)^2 + (y - 1)^2$. Pruebe diferentes condiciones iniciales y compare las respuestas. ¿Cuántos dígitos correctos pueden obtenerse mediante el uso de este método?

6. Aplique el método de Nelder-Mead para encontrar los mínimos de las siguientes funciones con seis posiciones decimales correctas (cada función tiene dos mínimos):
   (a) $f(x, y) = x^4 + y^4 + 2x^2y^2 + 6xy - 4x - 4y + 1$
   (b) $f(x, y) = x^6 + y^6 + 3x^2y^2 - x^2 - y^2 - 2xy$

7. Aplique el método de Nelder-Mead para encontrar el mínimo de la función de Rosenbrock
   $f(x, y) = 100(y - x^2)^2 + (x - 1)^2$.

8. Reescriba el programa 13.3 e incorpore un criterio de terminación para Nelder-Mead basado en una tolerancia al error especificada por el usuario. Pruebe su programa localizando los mínimos de las funciones objetivo del problema de computadora 6 hasta seis decimales correctos.

13.2 OPTIMIZACIÓN NO RESTRINGIDA CON DERIVADAS

Las derivadas contienen información acerca de la rapidez de crecimiento y disminución de una función, y en el caso de las derivadas parciales, también acerca de las direcciones de más rápido crecimiento y disminución. Si se cuenta con dicha información sobre la función objetivo, entonces ésta puede aprovecharse para encontrar el óptimo de una manera más eficiente.

13.2.1 Método de Newton

Si la función es continuamente diferenciable y es posible evaluar la derivada, entonces el problema de optimización puede expresarse como un problema de localización de raíces. Se iniciará en una dimensión, donde la interpretación es más sencilla.

En un mínimo x^* de una función continua diferenciable $f(x)$, la primera derivada debe ser cero. La ecuación resultante $f'(x) = 0$ puede resolverse empleando los métodos del capítulo 1. Si la función objetivo es unimodal y tiene un mínimo en un intervalo, entonces, empezando la aplicación del método de Newton con una estimación inicial cercana al mínimo x^* tendrá convergencia a x^*. El método de Newton aplicado a $f'(x) = 0$ se convierte en la iteración

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}. \tag{13.3}$$

Aunque el método de Newton (13.3) encontrará puntos en los que $f'(x) = 0$, en general, dichos puntos no tienen que ser mínimos. Es importante tener un valor inicial razonablemente cercano al óptimo y comprobar que una vez localizados sean los óptimos.

Para la optimización de una función $f(x_1, \ldots, x_n)$ se utiliza el método de Newton con varias variables. Como en el caso unidimensional, se desea igualar la derivada a cero y resolver. Así que se tiene

$$\nabla f = 0, \tag{13.4}$$

donde

$$\nabla f = \left[\frac{\partial f}{\partial x_1}(x_1, \ldots, x_n), \ldots, \frac{\partial f}{\partial x_n}(x_1, \ldots, x_n) \right]$$

indica el gradiente de f.

El método de Newton para las funciones con valores vectoriales del capítulo 2 permite resolver (13.4). Si se establece $F(x) = \nabla f(x)$, el paso iterativo del método de Newton se establece como $x_{k+1} = x_k + v$, donde v es la solución de $DF(x_k)v = -F(x_k)$. La matriz jacobiana DF del gradiente es

$$H_f = DF = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}, \tag{13.5}$$

que es la matriz hessiana de f. Por lo tanto, el paso de Newton es

$$\begin{cases} H_f(x_k)v = -\nabla f(x_k) \\ x_{k+1} = x_k + v \end{cases}. \tag{13.6}$$

► **EJEMPLO 13.4** Localice el mínimo de la función $f(x, y) = 5x^4 + 4x^2y - xy^3 + 4y^4 - x$, usando el método de Newton.

La función se muestra en la figura 13.6. El gradiente es $\nabla f = (20x^3 + 8xy - y^3 - 1, 4x^2 - 3xy^2 + 16y^3)$, y la hessiana es

$$H_f(x, y) = \begin{bmatrix} 60x^2 + 8y & 8x - 3y^2 \\ 8x - 3y^2 & -6xy + 48y^2 \end{bmatrix}.$$

En 10 iteraciones del método de Newton (13.6) resulta:

iteración	x	y	$f(x, y)$
0	1.00000000000000	1.00000000000000	11.00000000000000
1	0.64429530201342	0.63758389261745	1.77001867827422
2	0.43064034542956	0.39233298702231	0.10112006537534
3	0.33877971433352	0.19857714160717	−0.17818585977225
4	0.50009733696780	−0.44771929519763	−0.42964065053918
5	0.49737350571430	−0.37972645728644	−0.45673719664708
6	0.49255000651877	−0.36497753746514	−0.45752009007757
7	0.49230831759106	−0.36428704569173	−0.45752162262701
8	0.49230778672681	−0.36428555993321	−0.45752162263407
9	0.49230778672434	−0.36428555992634	−0.45752162263407
10	0.49230778672434	−0.36428555992634	−0.45752162263407

El método de Newton converge dentro de la precisión de máquina a un valor mínimo cercano a −0.4575. Observe otra característica de la disminución del error usando el método de Newton: se ha logrado la precisión de máquina en la solución, a diferencia del caso unidimensional de la interpolación parabólica sucesiva. La razón es que ya no se trabaja con la función objetivo, sino que se ha replanteado el problema sólo como un problema de localización de raíces que involucra al gradiente. Como ∇f tiene una raíz simple en el óptimo, no hay dificultad en conseguir un error hacia adelante cercano al épsilon máquina. ◀

Con frecuencia, el método de Newton se elige si es posible calcular la hessiana. En problemas bidimensionales la hessiana casi siempre está disponible. En una dimensión n alta puede ser factible sólo calcular el gradiente (un vector n-dimensional) en cada punto, pero no factible construir la hessiana de $n \times n$. Los siguientes dos métodos suelen ser más lentos que el de Newton, pero sólo requieren calcular el gradiente en varios puntos.

13.2.2 Gradiente descendente

La idea fundamental detrás del **gradiente descendente**, también llamado **búsqueda del gradiente**, es la localización de un mínimo de la función al moverse en la dirección de la máxima inclinación a partir del punto actual. Como los puntos del gradiente ∇f en la dirección de la máxima inclinación de f, la dirección opuesta $-\nabla f$ es la línea de declinación. ¿Hasta dónde debe llegarse a lo largo de esta dirección? Ahora que se ha reducido el problema a una minimización a lo largo de una línea, uno de los métodos unidimensionales debe decidir hasta dónde llegar. Después de localizar el nuevo mínimo a lo largo de la línea de declinación, repita el proceso, a partir de ese punto. Es decir, encuentre el gradiente en el nuevo punto, y haga una minimización unidimensional en la nueva dirección.

El algoritmo del gradiente descendente es un ciclo iterativo.

Gradiente descendente

for $i = 0, 1, 2, \ldots$
$v = \nabla f(x_i)$
Minimizar $f(x - sv)$ para el escalar $s = s^*$
$x_{i+1} = x_i - s^* v$
end

Se aplicará el gradiente descendente a la función objetivo del ejemplo 13.3.

► EJEMPLO 13.5 Localice el mínimo de la función $f(x, y) = 5x^4 + 4x^2y - xy^3 + 4y^4 - x$ usando el gradiente descendente.

Se siguen los pasos anteriores, utilizando la interpolación parabólica sucesiva como método unidimensional. Los resultados en 25 iteraciones son los siguientes:

iteración	x	y	$f(x, y)$
0	1.00000000000000	−1.00000000000000	11.00000000000000
5	0.40314579518113	−0.27992088271756	−0.41964888830651
10	0.49196895085112	−0.36216404374206	−0.45750680523754
15	0.49228284433776	−0.36426635686172	−0.45752161934016
20	0.49230786417532	−0.36428539567277	−0.45752162263389
25	0.49230778262142	−0.36428556578033	−0.45752162263407

La convergencia es más lenta comparada con el método de Newton, por una buena razón. El método de Newton consiste en resolver una ecuación y utiliza la primera y segunda derivadas (incluyendo el hessiano). El gradiente descendente es en realidad una minimización siguiendo la dirección de descenso y sólo utiliza la información de la primera derivada. ◄

13.2.3 Búsqueda del gradiente conjugado

En el capítulo 2 el método del gradiente conjugado se usó para resolver ecuaciones matriciales simétricas definidas positivas. Ahora se empleará de nuevo el método, visto desde una diferente perspectiva.

La resolución de $Ax = w$ cuando A es simétrica y definida positiva es equivalente a la localización del mínimo de un paraboloide. Por ejemplo, en dos dimensiones la solución del sistema lineal

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix} \tag{13.7}$$

es el mínimo del paraboloide

$$f(x_1, x_2) = \frac{1}{2}ax_1^2 + bx_1x_2 + \frac{1}{2}cx_2^2 - ex_1 - fx_2. \tag{13.8}$$

La razón es que el gradiente de f es

$$\nabla f = [ax_1 + bx_2 - e, bx_1 + cx_2 - f].$$

El gradiente es cero en el mínimo, lo que da la ecuación matricial anterior. El signo positivo significa que el paraboloide es cóncavo hacia arriba.

La observación clave es que el residuo $r = w - Ax$ del sistema lineal (13,7) es $-\nabla f(x)$, la dirección de la declinación de la función f en el punto x. Suponga que se ha elegido una dirección de búsqueda, indicada por el vector d. La minimización de f en (13.8) a lo largo de esa dirección consiste en encontrar la α que minimice la función $h(\alpha) = f(x + \alpha d)$. La derivada se igualará a cero para encontrar el mínimo:

$$\begin{aligned} 0 &= \nabla f \cdot d \\ &= (A(x + \alpha d) - (e, f)^T) \cdot d \\ &= (\alpha Ad - r)^T d. \end{aligned}$$

Esto implica que

$$\alpha = \frac{r^T d}{d^T Ad} = \frac{r^T r}{d^T Ad},$$

donde la última igualdad se desprende del teorema 2.16 referente al método del gradiente conjugado.

De este cálculo se concluye que podría resolverse de manera alternativa para el mínimo de un paraboloide mediante el método del gradiente conjugado, pero reemplazando

$$r_i = -\nabla f$$

y

$$\alpha_i = \alpha \text{ que disminuye mínimo } f(x_{i-1} + \alpha d_{i-1}).$$

De hecho, al verlo de esta manera, observe que se ha expresado el gradiente conjugado por completo en términos de f. Sigue sin hacerse mención de la matriz A. Es posible ejecutar el algoritmo en esta forma para una f general. Cerca de las regiones donde f tiene una forma parabólica, el método se mueve muy rápido hacia la parte inferior. El nuevo algoritmo tiene los siguientes pasos:

Búsqueda del gradiente conjugado

Sea x_0 la estimación inicial y establezca $d_0 = r_0 = -\nabla f$.

for $i = 1, 2, 3, \ldots$
$\alpha_i = \alpha$ tal que minimice a $f(x_{i-1} + \alpha d_{i-1})$
$x_i = x_{i-1} + \alpha_i d_{i-1}$
$r_i = -\nabla f(x_i)$
$\beta_i = \dfrac{r_i^T r_i}{r_{i-1}^T r_{i-1}}$
$d_i = r_i + \beta_i d_{i-1}$
end

Se probará el nuevo método en un ejemplo muy conocido.

► EJEMPLO 13.6 Localice el mínimo de la función $f(x, y) = 5x^4 + 4x^2y - xy^3 + 4y^4 - x$, usando la búsqueda del gradiente conjugado.

Se siguen los pasos anteriores, utilizando la interpolación parabólica sucesiva como el método unidimensional. Los resultados para 20 iteraciones son los siguientes:

iteración	x	y	$f(x, y)$
0	1.00000000000000	-1.00000000000000	.11.00000000000000
5	0.46038657599935	−0.38316114029860	−0.44849953420621
10	0.49048892807181	−0.36106561127830	−0.45748477171484
15	0.49243714956128	−0.36421661473526	−0.45752147604312
20	0.49231477751583	−0.36429817275371	−0.45752162206984

El tema de la optimización no restringida es muy amplia, y los métodos de este capítulo representan sólo la punta del iceberg. Los **métodos de la región de confianza** forman modelos locales, como lo hacen la interpolación parabólica sucesiva o la búsqueda del gradiente conjugado, pero permiten su uso sólo dentro de una región determinada que se estrecha a medida que avanza la búsqueda. La rutina `fminunc` de la caja de herramientas de optimización en MATLAB es un ejemplo de un método de la región de confianza. El **templado simulado** es un método estocástico que trata de avanzar más abajo en la función objetivo, pero aceptará un paso hacia arriba con una probabilidad pequeña, positiva, con el fin de evitar la convergencia a un mínimo local no óptimo. Los **algoritmos genéticos** y la computación evolutiva, en general, proponen enfoques muy novedosos para la optimización y todavía se siguen estudiando de manera activa.

La optimización restringida toma como meta la minimización de una función objetivo sujeta a un conjunto de restricciones. El subconjunto más común de estos problemas, la programación lineal, se ha resuelto mediante el método simplex desde su desarrollo a mediados del siglo XX;

aunque recientemente se han desarrollado nuevos algoritmos más rápidos y que se basan en los métodos de punto interior. Los problemas de programación no lineal y cuadráticos requieren métodos más sofisticados. Consulte las referencias donde encontrará los puntos de entrada a la literatura relacionada con estos métodos.

13.2 Problemas de computadora

1. Utilice el método de Newton para encontrar el mínimo de $f(x, y) = e^{-x^2y^2} + (x - 1)^2 + (y - 1)^2$. Pruebe diferentes condiciones iniciales y compare las respuestas. ¿Cuántos dígitos correctos pueden obtenerse con este método?

2. Aplique el método de Newton para encontrar los mínimos de las siguientes funciones con seis posiciones decimales correctas (cada función tiene dos mínimos):
   (a) $f(x, y) = x^4 + y^4 + 2x^2y^2 + 6xy - 4x - 4y + 1$
   (b) $f(x, y) = x^6 + y^6 + 3x^2y^2 - x^2 - y^2 - 2xy$

3. Encuentre el mínimo de la función de Rosenbrock $f(x, y) = 100(y - x^2)^2 + (x - 1)^2$ mediante (a) el método de Newton y (b) el gradiente descendente. Use la estimación inicial (2, 2). ¿Después de cuántos pasos la solución deja de mejorar? Explique la diferencia en la precisión que se logra.

4. Utilice el gradiente descendente para encontrar los mínimos de las funciones del problema de computadora 2.

5. Use la búsqueda del gradiente conjugado para encontrar los mínimos de las funciones del problema de computadora 2.

6. Encuentre los mínimos con cinco dígitos correctos mediante la búsqueda del gradiente conjugado:
   (a) $f(x, y) = x^4 + 2y^4 + 3x^2y^2 + 6x^2y - 3xy^2 + 4x - 2y$
   (b) $f(x, y) = x^6 + x^2y^4 + y^6 + 3x + 2y$

Comprobación en la realidad 13

Conformación molecular y optimización numérica

La función de una proteína sigue su forma: los nudos y los pliegues de las formas moleculares permiten los enlaces y bloqueos que son parte integral de sus funciones. Las fuerzas que gobiernan la **conformación**, o el doblamiento, de los aminoácidos en proteínas se deben a los enlaces entre los átomos individuales y a las interacciones intermoleculares débiles entre átomos no unidos como las fuerzas electrostáticas y las fuerzas de Van der Waals. Para las moléculas densamente agrupadas, como las proteínas, éstas últimas son muy importantes.

Un enfoque actual para predecir las conformaciones de las proteínas consiste en encontrar la energía potencial mínima de la configuración total de los aminoácidos. Las fuerzas de Van der Waals se modelan mediante el potencial de Lennard-Jones

$$U(r) = \frac{1}{r^{12}} - \frac{2}{r^6},$$

donde r indica la distancia entre dos átomos. En la figura 13.7 se muestra el pozo de energía que se define mediante el potencial. La fuerza es atractiva para las distancias $r > 1$, pero se vuelve fuertemente repulsiva cuando los átomos tratan de acercarse a más de $r = 1$. Para un grupo de átomos con posiciones $(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)$, la función objetivo a minimizar es la suma de los potenciales de Lennard-Jones por pares

$$U = \sum_{i<j} \left(\frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^6} \right)$$

en todos los pares de átomos, donde

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$

indica la distancia entre los átomos i y j. Las variables en el problema de optimización son las coordenadas rectangulares de los átomos.

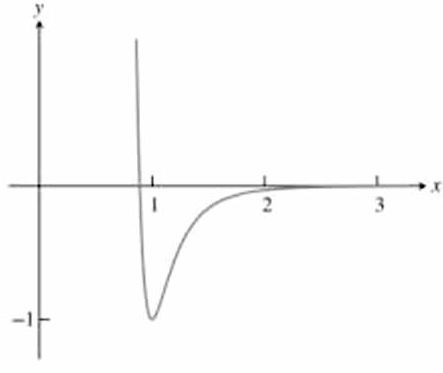


Figura 13.7 El potencial de Lennard-Jones $U(r) = r^{-12} - 2r^{-6}$. La energía mínima es -1, que se alcanza en $r = 1$.

Existen simetrías de traslación y de rotación que deben considerarse: la energía total no cambia si el grupo de átomos se mueve en una línea recta o se gira. Para hacer frente a las simetrías, se limitarán las posibles configuraciones al fijar el primer átomo en el origen $v_1 = (0, 0, 0)$ y al requerir que el segundo átomo se encuentre en el eje z en $v_2 = (0, 0, z_2)$. Las restantes variables de posición $(x_3, y_3, z_3), \ldots, (x_n, y_n, z_n)$ se disponen en una configuración tal que minimice la energía potencial U.

Con la ayuda de la figura 13.7, resulta fácil ordenar cuatro o menos átomos con el más bajo nivel posible de energía de Lennard-Jones. Tenga en cuenta que el mínimo potencial individual tiene el valor -1 en $r = 1$. Así, dos átomos pueden colocarse justo a una unidad del otro, de modo que la energía esté exactamente en el punto mínimo baja del valle. Tres átomos pueden colocarse en un triángulo cuyo lado es la misma distancia común, y un cuarto átomo puede colocarse a la misma distancia de los tres vértices, por ejemplo, por encima del triángulo, creando un tetraedro equilátero. La energía total U para los casos en que $n = 2, 3$ y 4 es -1 veces el número de interacciones o -1, -3 y -6, respectivamente.

Sin embargo, la colocación de un quinto átomo no es tan evidente. No hay un punto equidistante de los vértices del tetraedro en el caso de $n = 4$, y se requiere una nueva técnica, optimización numérica.

Actividades sugeridas:

1. Escriba el archivo de una función que devuelva la energía potencial. Aplique el método Nelder-Mead para encontrar la energía mínima para $n = 5$. Pruebe varias aproximaciones iniciales hasta que esté convencido de tener el mínimo absoluto. ¿Cuántos pasos se necesitan?

2. Utilice el comando `plot3` de MATLAB para graficar cinco átomos en la configuración de energía mínima como círculos, y conecte todos los círculos con segmentos de línea para visualizar la molécula conformada.

3. Extienda la función del paso 1 para que devuelva f y el vector gradiente ∇f. Aplique la búsqueda del gradiente para el caso en que $n = 5$. Encuentre la energía mínima como lo hizo anteriormente.

4. Si la caja de herramientas de optimización de MATLAB está disponible, aplique el comando `fminunc`, usando sólo la función objetivo f.
5. Aplique `fminunc`, con f y ∇f.
6. Aplique los métodos anteriores para $n = 6$. Clasifique los métodos de acuerdo con su confiabilidad y eficacia.
7. Determine y grafique conformaciones de energía mínima para una n más grande. Es posible encontrar información sobre la energía mínima de Lennard-Jones en grupos para n de hasta varios cientos, en algunos sitios de internet, por lo que sus respuestas pueden comprobarse con facilidad.

El problema del plegamiento de proteínas se ha convertido en un semillero de investigación en optimización multidisciplinaria. Los eficaces métodos del templado simulado y cuasi-Newton suelen utilizarse para predecir la conformación de moléculas complicadas, generando modelos cada vez más realistas de las fuerzas intermoleculares. El banco de datos de las proteínas `http://www.rcsb.org/pdb` es un útil archivo internacional de datos estructurales sobre las macromoléculas biológicas. En este sitio se dispone de amplias listas de posiciones atómicas medidas experimentalmente para su uso en pruebas y para la validación de hipótesis sobre la minimización de las fuerzas y la energía. ✔

Software y lecturas adicionales

Entre los textos de introducción a la optimización se encuentran Dennis y Schnabel [1987], Nocedal y Wright [1999], y Griva *et al.* [2008]. La útil guía de Moré y Wright [1987] contiene referencias a muchos paquetes de software diseñados en particular para la optimización. En Floudas *et al.* [1999] puede encontrarse un gran conjunto de problemas de prueba de diversos tipos. El Optimization Technology Center (Centro de Tecnología de la Optimización) dirigido por la Northwestern University y el Argonne National Lab `http://www.ece.northwestern.edu/OTC` tiene muchos enlaces al software existente.

El directorio `opt` de Netlib contiene un número de rutinas de optimización gratuitas, incluyendo: `hooke` (optimización no restringida sin derivadas, a través del método de Hooke y Jeeves), `praxis` (optimización no restringida, sin necesidad de derivadas), y `tn` (método de Newton para la optimización no restringida o simplemente enlazada). WNLIB de Chapman y Naylor incluye rutinas para la optimización no lineal restringida y no restringida basadas en los algoritmos del gradiente conjugado y de las direcciones conjugadas (así como una rutina general del templado simulado).

La caja de herramientas de optimización de MATLAB incluye rutinas para una variedad de problemas de optimización no lineal restringida y no restringida. El entorno de optimización TOMLAB ofrece una amplia variedad de herramientas de optimización no lineal basadas en las cajas de herramientas de MATLAB. Contiene un formato unificado de entrada-salida, una GUI opcional y un método para derivar. Las listas de optimización en `mathtools.net` incluyen muchos solucionadores escritos en MATLAB y otros lenguajes.

Apéndice A: Álgebra matricial

Se iniciará con un breve repaso de las definiciones básicas del álgebra matricial.

A.1 FUNDAMENTOS DE LAS MATRICES

Un **vector** es un arreglo de números

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}.$$

Si la lista contiene n números, se denomina un vector n-dimensional. Con frecuencia se hace una distinción entre el arreglo anterior dispuesto verticalmente, o **vector columna**, y un arreglo dispuesto horizontalmente

$$u = [u_1, \ldots, u_n]$$

denominado **vector renglón**. Una **matriz** de $m \times n$ es un arreglo de $m \times n$ números que tiene la forma

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}.$$

Cada renglón (horizontal) de A puede considerarse como un vector renglón de A, y cada columna (vertical) como un vector columna.

La multiplicación matriz-vector forma un vector a partir de una matriz y un vector. El producto matriz-vector se define como

$$Au = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} a_{11}u_1 + a_{12}u_2 + \cdots + a_{1n}u_n \\ \vdots \\ a_{m1}u_1 + a_{m2}u_2 + \cdots + a_{mn}u_n \end{bmatrix}. \quad \text{(A.1)}$$

Observe que para multiplicar una matriz de $m \times n$ por un vector d-dimensional, se requiere que $n = d$.

En la multiplicación matriz-matriz, una matriz de $m \times n$ se multiplica por otra de $n \times p$ para generar una matriz de $m \times p$ como producto. La multiplicación de matrices puede expresarse en términos de la multiplicación matriz-vector. Sea C una matriz de $n \times p$ escrita en términos de sus vectores columna

$$C = \begin{bmatrix} c_1 | & \cdots & | c_p \end{bmatrix}.$$

Entones, el producto matriz-matriz de A y C es

$$AC = A\begin{bmatrix} c_1 | & \cdots & | c_p \end{bmatrix} = \begin{bmatrix} Ac_1 | & \cdots & | Ac_p \end{bmatrix}.$$

Un sistema de m ecuaciones lineales con n incógnitas puede escribirse en forma matricial como

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

que se llama una **ecuación matricial**.

La **matriz identidad** I_n de $n \times n$ es la matriz con $I_{ii} = 1$ para $1 \leq i \leq n$ e $I_{ij} = 0$ para $i \neq j$. La matriz identidad sirve como identidad para la operación de multiplicación matricial, como $AI_n = I_n A = A$ para cada matriz A de $n \times n$. Para una matriz A de $n \times n$, la **inversa** A^{-1} de A es una matriz de $n \times n$ que satisface $AA^{-1} = A^{-1}A = I_n$. Si A tiene una inversa, se llama **invertible**; una matriz que no es invertible se llama **singular**.

La **transpuesta** de una matriz A de $m \times n$ es la matriz A^T cuyas entradas son $A^T_{ij} = A_{ji}$. La regla para la transpuesta de un producto es $(AB)^T = B^T A^T$.

Hay dos formas importantes de multiplicar dos vectores. Sean

$$u = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} \quad \text{y} \quad v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}.$$

El **producto interior** $u^T v$ transpone u como un vector renglón, después la multiplicación ordinaria de matrices da

$$u^T v = u_1 v_1 + \cdots + u_n v_n.$$

Así, el producto de $1 \times n$ por $n \times 1$ resulta en una matriz de 1×1, o un número real. Dos vectores columna son **ortogonales** si $u^T v = 0$. El **producto exterior** uv^T multiplica una columna de $n \times 1$ por un renglón de $1 \times n$. La multiplicación matricial ordinaria da como resultado una matriz de $n \times n$

$$uv^T = \begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_n \\ u_2 v_1 & u_2 v_2 & \cdots & u_2 v_n \\ \vdots & & & \vdots \\ u_n v_1 & \cdots & \cdots & u_n v_n \end{bmatrix}.$$

Un producto exterior es una matriz de rango uno.

Cada producto matricial AB puede representarse como la suma de los productos externos de las columnas de A con los renglones de B. De manera más precisa,

Regla de la suma para el producto exterior

Sean A y B matrices de $m \times p$ y $p \times n$, respectivamente. Entonces

$$AB = \sum_{i=1}^{p} a_i b_i^T$$

donde a_i es la i-ésima columna de A y b_i^T es el i-ésimo renglón de B.

El caso en el que $n = 1$ se llama a veces la "forma alternativa de la multiplicación matriz-vector". Por ejemplo,

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} -3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 7 \end{bmatrix} \begin{bmatrix} -3 \end{bmatrix} + \begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix} \begin{bmatrix} 2 \end{bmatrix}$$

ilustra por qué, cuando se ve como una transformación lineal, el rango de una matriz es equivalente a su espacio de columnas.

Debido a la gran complejidad del cálculo de la inversa de una matriz, éste se evita o minimiza siempre que sea posible. Un truco útil es la fórmula de Sherman-Morrison. Suponga que ya se conoce la inversa de una matriz A de $n \times n$, y que se requiere la inversa de la matriz modificada $A + uv^T$, donde u y v son vectores de tamaño n.

TEOREMA A.1 (Fórmula de Sherman-Morrison) Si $v^T A^{-1} u \neq -1$, entonces $A + uv^T$ es invertible y

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1} u}. \qquad \blacksquare$$

La fórmula de Sherman-Morrison se prueba multiplicando $A + uv^T$ por la expresión en la fórmula. La matriz $A + uv^T$ se denomina **actualización de rango uno** de A, puesto que uv^T es una matriz de rango uno. (Vea el análisis del método de Broyden en el capítulo 2 donde puede encontrar una importante aplicación de la fórmula de Sherman-Morrison. Los hechos elementales sobre las matrices pueden consultarse en los textos de álgebra lineal como Strang [2005] y Lay [2005]).

A.2 MULTIPLICACIÓN EN BLOQUE

La multiplicación de matrices puede hacerse en bloques, lo que será muy útil en el capítulo 12. Si dos matrices se dividen en bloques cuyos tamaños son compatibles para la multiplicación de matrices, entonces, el producto matricial puede llevarse a cabo mediante la multiplicación matricial de los bloques. Por ejemplo, el producto de dos matrices de 3×3 puede realizarse en los siguientes bloques:

$$AB = \left[\begin{array}{c|cc} x & x & x \\ \hline x & x & x \\ x & x & x \end{array}\right]\left[\begin{array}{c|cc} x & x & x \\ \hline x & x & x \\ x & x & x \end{array}\right] = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right]\left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right]$$
$$= \left[\begin{array}{c|c} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} \\ \hline A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} \end{array}\right]$$

Aquí A_{11} y B_{11} son matrices de 1×1, A_{12} y B_{12} son matrices de 1×2, y así sucesivamente. Por ejemplo,

$$\left[\begin{array}{c|cc} 1 & 2 & 3 \\ \hline 0 & 1 & 3 \\ 2 & 2 & 4 \end{array}\right]\left[\begin{array}{c|cc} 2 & 4 & 1 \\ \hline 1 & 0 & 1 \\ 3 & 1 & 2 \end{array}\right] = \left[\begin{array}{cc} 1 \cdot 2 + \begin{bmatrix} 2 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} & 1\begin{bmatrix} 4 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 3 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 2 \end{bmatrix}2 + \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} & \begin{bmatrix} 0 \\ 2 \end{bmatrix}\begin{bmatrix} 4 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \end{array}\right]$$
$$= \left[\begin{array}{c|cc} 13 & 7 & 9 \\ \hline 10 & 3 & 7 \\ 18 & 12 & 12 \end{array}\right].$$

Si se realiza la multiplicación por bloques, se obtiene el mismo resultado que si se hace sin ellos. Esta forma alternativa de mirar la multiplicación matricial no está destinada a reducir los cálculos, sino a ayudar con el conteo, en especial en los cálculos de valores propios del capítulo 12.

La única compatibilidad necesaria para los bloques es que los grupos de columnas de A deben coincidir exactamente con los grupos de renglones de B. En el ejemplo anterior, la primera columna de A está en un grupo y las últimas dos columnas están en otro. Para la matriz B, el primer

renglón está en un grupo y los dos últimos *renglones* están en otro. Como un ejemplo adicional, es posible multiplicar la matriz A de 3×5 y la matriz B de 5×2 en los siguientes bloques:

$$\left[\begin{array}{c:ccc} x & x & x & x \\ \hdashline x & x & x & x \\ x & x & x & x \end{array}\right] \left[\begin{array}{c:c} x & x \\ \hdashline x & x \\ x & x \\ x & x \\ \hdashline x & x \end{array}\right]$$

$$= \left[\begin{array}{c:c:c} A_{11} & A_{12} & A_{13} \\ \hdashline A_{21} & A_{22} & A_{23} \end{array}\right] \left[\begin{array}{c:c} B_{11} & B_{12} \\ \hdashline B_{21} & B_{22} \\ \hdashline B_{31} & B_{32} \end{array}\right]$$

$$= \left[\begin{array}{c:c} A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31} & A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{32} \\ \hdashline A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31} & A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{32} \end{array}\right]$$

En este caso, los tres grupos de columnas de A coinciden con los tres grupos de renglones de B. Por otro lado, no es necesario que los grupos de renglones de A y de columnas de B coincidan; éstos pueden crearse de manera arbitraria.

A.3 VALORES Y VECTORES PROPIOS

Se iniciará con un breve repaso de los conceptos básicos de los valores y vectores propios.

DEFINICIÓN A.2 Sea A una matriz de $m \times m$ y x un vector real o complejo distinto de cero y m-dimensional. Si $Ax = \lambda x$ para algún número real o complejo λ, entonces λ se denomina un **valor propio** de A y x es el **vector propio** correspondiente. ❐

Por ejemplo, la matriz $A = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}$ tiene un vector propio $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, y el correspondiente valor propio 4.

Los valores propios son las raíces λ del **polinomio característico** $\det(A - \lambda I)$. Si λ es un valor propio de A, entonces cualquier vector no nulo en el espacio nulo de $A - \lambda I$ es un vector propio correspondiente a λ. Para este ejemplo,

$$\det(A - \lambda I) = \det\begin{bmatrix} 1-\lambda & 3 \\ 2 & 2-\lambda \end{bmatrix} = (\lambda - 1)(\lambda - 2) - 6 = (\lambda - 4)(\lambda + 1), \quad \text{(A.2)}$$

por lo que los valores propios son $\lambda = 4$, y -1. Los vectores propios que corresponden a $\lambda = 4$ se encuentran en el espacio nulo de

$$A - 4I = \begin{bmatrix} -3 & 3 \\ 2 & -2 \end{bmatrix} \quad \text{(A.3)}$$

y por lo tanto constará de todos los múltiplos de $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ distintos de cero. De manera similar, los vectores propios correspondientes a $\lambda = -1$ son todos los múltiplos de $\begin{bmatrix} 3 \\ -2 \end{bmatrix}$ distintos de cero.

DEFINICIÓN A.3 Las matrices A_1 y A_2 de $m \times m$ son **semejantes**, lo cual se indica mediante $A_1 \sim A_2$, si existe una matriz invertible S de $m \times m$ tal que $A_1 = SA_2S^{-1}$. ❐

Las matrices semejantes tienen valores propios idénticos, debido a que sus polinomios característicos son idénticos:

$$A_1 - \lambda I = SA_2S^{-1} - \lambda I = S(A_2 - \lambda I)S^{-1} \tag{A.4}$$

implica que

$$\det(A_1 - \lambda I) = (\det S)\det(A_2 - \lambda I)\det S^{-1} = \det(A_2 - \lambda I). \tag{A.5}$$

Si una matriz A tiene vectores propios que forman una base para R^m, entonces A es semejante a una matriz diagonal, y A es **diagonalizable**. De hecho, suponga que $Ax_i = \lambda_i x_i$ para $i = 1, \ldots, m$, y defina la matriz

$$S = [\ x_1 \quad \cdots \quad x_m\ ].$$

Se puede verificar que la ecuación matricial

$$AS = S\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} \tag{A.6}$$

se cumpla. La matriz S es invertible porque sus columnas abarcan R^m. Por lo tanto, A es semejante a la matriz diagonal que contiene sus valores propios.

No todas las matrices son diagonalizables, incluso en el caso de 2×2. De hecho, todas las matrices de 2×2 son semejantes a uno de los tres tipos siguientes:

$$A_1 = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$
$$A_2 = \begin{bmatrix} a & 1 \\ 0 & a \end{bmatrix}$$
$$A_3 = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}.$$

Recuerde que los valores propios son idénticos para las matrices semejantes. Una matriz es semejante a una matriz de la forma A_1 si hay dos vectores propios que abarquen R^2; una matriz es semejante a una matriz de la forma A_2, si hay un valor propio repetido con sólo un espacio dimensional de vectores propios; y es semejante a A_3 si tiene un par de valores propios complejos.

A.4 MATRICES SIMÉTRICAS

Para una matriz simétrica, todos los vectores propios son ortogonales entre sí, y juntos abarcan el espacio subyacente. En otras palabras, las matrices simétricas siempre tienen una base ortonormal de vectores propios.

DEFINICIÓN A.4 Un conjunto de vectores es **ortonormal** si los elementos del conjunto son vectores unitarios que son ortogonales por pares. ❐

En términos de productos punto, la ortonormalidad del conjunto $\{w_1, \ldots, w_m\}$ significa que $w_i^T w_j = 0$ si $i \neq j$, y $w_i^T w_i = 1$, para $1 \leq i, j \leq m$. Por ejemplo, los conjuntos $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$ y $\{(\sqrt{2}/2, \sqrt{2}/2), (\sqrt{2}/2, -\sqrt{2}/2)\}$ son conjuntos ortonormales.

TEOREMA A.5 Suponga que A es una matriz simétrica de $m \times m$ con entradas reales. Entonces los valores propios son números reales y el conjunto de vectores propios unitarios de A es un conjunto ortonormal $\{w_1, \ldots, w_m\}$ que forma una base de R^m. ■

► EJEMPLO A.1 Encuentre los valores y vectores propios de

$$A = \begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix}. \tag{A.7}$$

Si se calcula como antes, los pares de valores/vectores propios son $2, (1, 2)^T$ y $-1/2, (-2, 1)^T$. Observe que como lo afirma el teorema, los vectores propios son ortogonales. La base ortonormal correspondiente de vectores propios unitarios es

$$\left\{ \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix}, \begin{bmatrix} -\frac{2}{\sqrt{5}} \\ \frac{1}{\sqrt{5}} \end{bmatrix} \right\}.$$

◄

El siguiente teorema será útil para el estudio de los métodos iterativos en el capítulo 2:

DEFINICIÓN A.6 El **radio espectral** $\rho(A)$ de una matriz cuadrada A es la magnitud máxima de sus valores propios. ❐

TEOREMA A.7 Si la matriz A de $n \times n$ tiene un radio espectral $\rho(A) < 1$, y b es arbitraria, entonces, para cualquier vector x_0, la iteración $x_{k+1} = Ax_k + b$ converge. De hecho, existe una única x_*, de tal manera que $\lim_{k\to\infty} x_k = x_*$ y $x_* = Ax_* + b$. ■

Por otra parte, si $b = 0$, entonces x_* es el vector cero o un vector propio de A con valor propio 1. Esto último se descarta debido al radio espectral, lo que conduce al siguiente corolario que es útil en el capítulo 8:

COROLARIO A.8 Si la matriz A de $n \times n$ tiene un radio espectral $\rho(A) < 1$, entonces, para cualquier vector inicial x_0, la iteración $x_{k+1} = Ax_k$ converge a 0. ■

A.5 CÁLCULO VECTORIAL

En esta sección se definen las derivadas de las funciones con valores escalares y vectoriales, asimismo se reúnen las reglas de los productos que las involucran para su uso posterior.

Sea $f(x_1, \ldots, x_n)$ una función con valores escalares de n variables. El **gradiente** de f es la función vectorial

$$\nabla f(x_1, \ldots, x_n) = \left[f_{x_1}, \ldots, f_{x_n} \right],$$

donde los subíndices indican las derivadas parciales de f con respecto a esa variable.

Sea

$$F(x_1, \ldots, x_n) = \begin{bmatrix} f_1(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{bmatrix}$$

una función vectorial de n variables. El **jacobiano** de F es la matriz

$$DF(x_1, \ldots, x_n) = \begin{bmatrix} \nabla f_1 \\ \vdots \\ \nabla f_n \end{bmatrix}.$$

Ahora pueden establecerse las reglas del producto para dos productos típicos del álgebra matricial. Ambas tienen demostraciones directas cuando se escriben en componentes y se aplica la regla del producto de una sola variable. Sean $u(x_1, \ldots, x_n)$ y $v(x_1, \ldots, x_n)$ funciones vectoriales, y sea $A(x_1, \ldots, x_n)$ una función matricial de $n \times n$. El producto escalar $u^T v$ es una función escalar. La primera fórmula muestra cómo obtener su gradiente. El producto matriz-vector Av es un vector cuyo jacobiano se expresa en la segunda regla.

Regla del producto punto (o escalar) vectorial

$$\nabla(u^T v) = v^T Du + u^T Dv$$

Regla del producto matriz/vector

$$D(Av) = A \cdot Dv + \sum_{i=1}^{n} v_i Da_i,$$

donde a_i indica la i-ésima columna de A.

Apéndice B: Introducción a MATLAB

MATLAB es un entorno de computación de propósito general ideal para la aplicación de métodos matemáticos y numéricos. Se utiliza como una calculadora de gran potencia para los problemas pequeños y como un lenguaje de programación completo para los problemas grandes. Una característica útil de MATLAB es su larga lista de funciones de biblioteca de alta calidad que pueden hacer que los cálculos complicados sean cortos, precisos y fáciles de escribir en un código de nivel superior.

Esta sección contiene una breve introducción a los comandos y características de MATLAB. Es posible encontrar descripciones mucho más detalladas en los centros de ayuda de MATLAB, en la *guía del usuario de* MATLAB, en libros como Sigmon [2002], Hahn [2002], y en los sitios web dedicados especialmente a este paquete (donde encontrará las versiones más recientes de este paquete).

B.1 INICIO DE MATLAB

En los sistemas basados en PC, MATLAB se inicia haciendo clic en el icono correspondiente y se cierra haciendo clic en Archivo/Salir. En los sistemas basados en Unix, pulse MATLAB en el indicador del sistema:

```
$ matlab
```

Después escriba

```
>> exit
```

para salir.

Escriba el comando

```
>> a=5
```

seguido de la tecla de retorno. MATLAB le devolverá la información. Escriba los comandos adicionales

```
>> b=3
>> c=a+b
>> c=a*b
>> d=log(c)
>> who
```

para tener una idea de cómo funciona MATLAB. Puede incluir un punto y coma después de una declaración para suprimir la repetición del valor. El comando `who` contiene una lista de todas las variables que ha definido.

MATLAB tiene una herramienta extensa de ayuda en línea. Escriba `help log` para obtener información sobre el comando `log`. La versión para PC de MATLAB tiene un menú de ayuda que contiene descripciones y sugerencias de uso en todos los comandos.

Para borrar el valor de la variable *a*, pulse `clear a`. Al escribir `clear`, se borrarán todas las variables definidas con anterioridad. Para recuperar un comando anterior, utilice la tecla de cursor hacia arriba. Si se acaba el espacio en la línea de comandos actual, termine la línea con tres puntos y un retorno; después siga escribiendo en la línea siguiente.

Para guardar los valores de las variables para el siguiente inicio de sesión, escriba `save` y luego `load` en su próximo inicio de sesión de MATLAB. Para transcribir una parte o la totalidad de la sesión de MATLAB, escriba `diary filename` para iniciar el registro, y `diary off` para finalizarlo. Utilice un nombre de archivo de su elección para `filename`. Esto es útil para presentar su trabajo en una tarea. El comando `diary` produce un archivo que puede verse o imprimirse una vez que haya terminado su sesión en MATLAB.

Por lo general, MATLAB realiza todos los cálculos en IEEE de doble precisión, aproximadamente con 16 dígitos decimales de precisión. El formato de visualización numérica puede cam-

biarse con la instrucción `format`. Al escribir `format long` cambiará la forma en que se muestran los números hasta un nuevo aviso. Por ejemplo, el número 1/3 se mostrará de forma diferente dependiendo del formato actual:

```
format short      0.3333
format short e    3.3333E-001
format long       0.33333333333333
format long e     3.333333333333333E-001
format bank       0.33
format hex        3fd5555555555555
```

El comando `fprintf` proporciona más control sobre el formato de salida. Los comandos

```
>> x=0:0.1:1;
>> y=x.^2;
>> fprintf('%8.5f %8.5f \n',[x;y])
```

imprimen la tabla

```
0.00000  0.00000
0.10000  0.01000
0.20000  0.04000
0.30000  0.09000
0.40000  0.16000
0.50000  0.25000
0.60000  0.36000
0.70000  0.49000
0.80000  0.64000
0.90000  0.81000
1.00000  1.00000
```

B.2 GRÁFICAS

Para graficar los datos, expréselos como vectores en las direcciones X y Y. Por ejemplo, los comandos

```
>> a=[0.0 0.4 0.8 1.2 1.6 2.0];
>> b=sin(a);
>> plot(a,b)
```

dibujará una aproximación lineal por partes de la gráfica de $y = \text{sen}\, x$ en $0 \leq x \leq 2$, como se muestra en la figura B.1(a). En este caso, `a` y `b` son vectores de 6 dimensiones, o arreglos de 6 elementos. Por ejemplo, la fuente de los números del eje x puede ajustarse a 16 puntos mediante el comando `set(gca,'FontSize',16)`. Una manera más corta de definir el vector `a` es el comando

```
>> a=0:0.4:2;
```

Este comando define a `a` como un vector cuyas entradas comienzan en 0, tienen incrementos de 0.4 y terminan en 2, de manera idéntica a la definición anterior más larga. Una versión más precisa de un ciclo completo de la curva sinusoidal resulta de

```
>> a=0:0.02:2*pi;
>> b=sin(a);
>> plot(a,b)
```

y se muestra en la figura B.1(b).

Para dibujar la gráfica de $y = x^2$ en $0 \leq x \leq 2$, podría utilizarse

```
>> a=0:0.02:2;
>> b=a.^2;
>> plot(a,b)
```

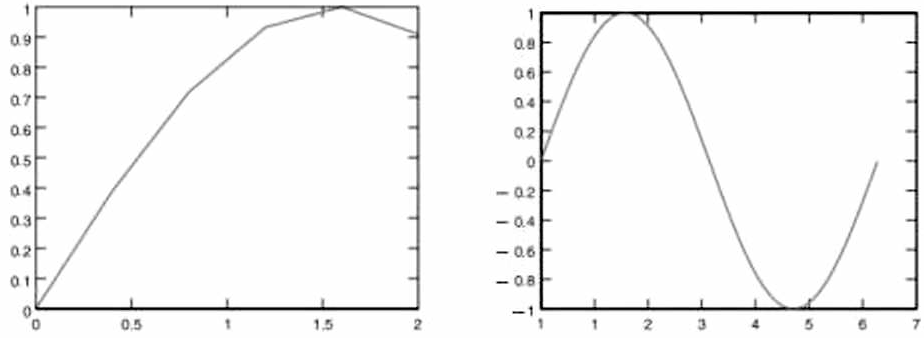


Figura B.1 Figuras en MATLAB. (a) Gráfica lineal por partes de $f(x) = \text{sen } x$, con un incremento x de 0.4. (b) Otra gráfica por partes que se ve más suavizada debido a que el incremento x es de 0.02.

El carácter "." que precede al operador de potencia puede ser omitido, y ocasiona que el operador de potencia se vectorice, es decir, eleva al cuadrado cada entrada del vector `a`. Como se verá en la siguiente sección, MATLAB trata a cada variable como una matriz. Si se omite el punto en este caso significaría multiplicar la matriz `a` de 101×1 por sí misma, bajo las reglas de la multiplicación matricial, lo cual es imposible. Si se le pide a MATLAB hacer esto, marcará error. En general, MATLAB interpreta una operación precedida por un punto como que la operación debe aplicarse a cada entrada, no como una multiplicación matricial.

Hay técnicas más avanzadas para realizar gráficas. MATLAB elegirá la escala de los ejes de forma automática si ésta no se especifica, como en la figura B.1. Para elegir la escala de los ejes manualmente, utilice el comando `axis`. Por ejemplo, al solicitar una gráfica con el comando

```
>> v=[-1 1 0 10]; axis(v)
```

se establece la ventana grafica como $[-1, 1] \times [0, 10]$. El comando `grid` dibuja una cuadrícula detrás de la gráfica.

Utilice el comando `plot(x1, y1, x2, y2, x3, y3)` para graficar tres curvas en la misma ventana gráfica, donde `xi, yi` son pares de vectores con las mismas longitudes. Escriba `help plot` para ver las opciones de tipos de línea continuas, punteadas y discontinuas, y varios tipos de símbolos (círculos, puntos, triángulos, cuadrados, etcétera) en la gráfica. Las gráficas semilogarítmicas están disponibles a través de los comandos `semilogy` y `semilogx`.

El comando subplot divide la ventana gráfica en varias partes. La instrucción `subplot(abc)` parte la ventana en una cuadrícula de $a \times b$ y utiliza el cuadro c para la gráfica. Por ejemplo,

```
>> subplot(121),plot(x,y)
>> subplot(122),plot(x,z)
```

dibuja la primera gráfica en la parte izquierda de la pantalla y la segunda en la parte derecha. El comando `figure` abre nuevas ventanas gráficas y se mueve entre ellas si necesita ver varias gráficas diferentes a la vez.

Las gráficas de superficie tridimensionales se dibujan con el comando `mesh`. Por ejemplo, la función $z = \text{sen}(x^2 + y^2)$ en el dominio $[-1, 1] \times [-2, 2]$ puede graficarse mediante

```
>> [x,y]=meshgrid(-1:0.1:1,-2:0.1:2);
>> z=sin(x.^2+y.^2);
>> mesh(x,y,z)
```

El vector `x` creado mediante `meshgrid` tiene 41 renglones del vector `-1:0.1:1` de tamaño 21 y, de manera similar, `y` tiene 21 columnas del vector columna `-2:0.1:2`. La gráfica generada por este código se muestra en la figura B.2. Al sustituir `mesh` por `surf` se grafica una superficie de colores sobre la malla.

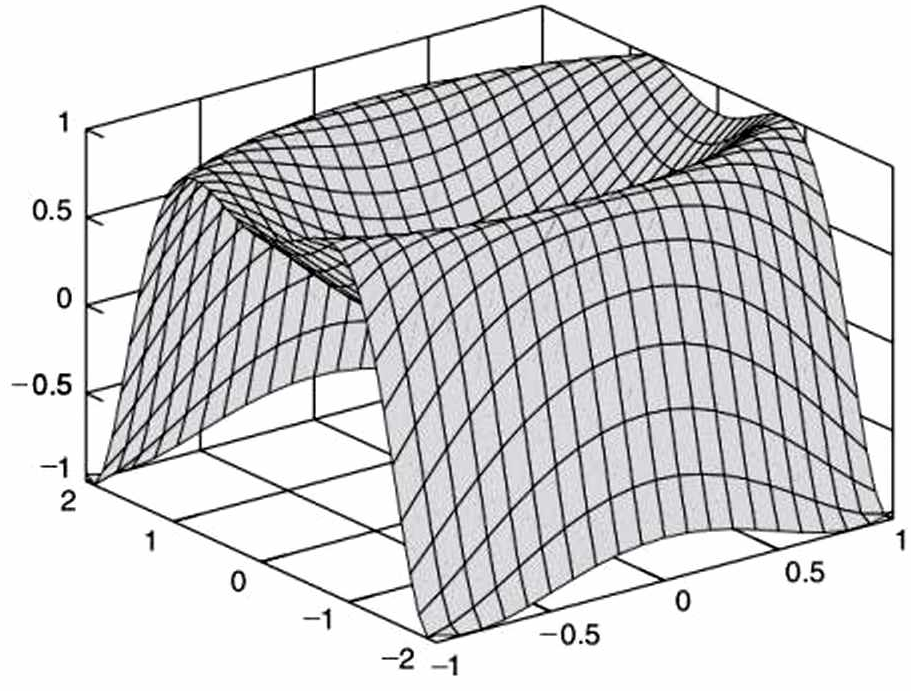


Figura B.2 Gráfica tridimensional en MATLAB. El comando `mesh` se utiliza para graficar superficies.

B.3 PROGRAMACIÓN EN MATLAB

La escritura de programas en el lenguaje MATLAB permite obtener resultados más sofisticados. Un **archivo script** es un archivo que contiene una lista de comandos de MATLAB. El nombre del archivo script tiene un sufijo `.m`, por lo que este tipo de archivos suelen denominarse **archivos m**. Por ejemplo, puede utilizar su editor favorito, o el editor de MATLAB si está disponible, para crear el archivo `cubrt.m`, que contiene las siguientes líneas:

```
% El programa cubrt.m encuentra una raíz cúbica por iteración
y=1;
n=15;
z=input('Enter z:');
for i = 1:n
  y = 2*y/3 + z/(3*y^2)
end
```

Para ejecutar el programa, escriba `cubrt` en el indicador de MATLAB. La razón por la que este código converge a la raíz cúbica se hará evidente a partir del estudio del método de Newton del capítulo 1. Observe que el punto y coma fue eliminado de la línea que define el valor de la nueva y iteración. Esto le permite ver el progreso de las aproximaciones cuando se acercan a la raíz cúbica.

Con la capacidad gráfica de MATLAB es posible analizar los datos del algoritmo de la raíz cúbica. Considere el programa `cubrt1.m`:

```
% El programa cubrt1.m encuentra raíces cúbicas y muestra su progreso
y(1)=1;
n=15;
z=input('Introduzca z:');
for i = 1:n-1
  y(i+1) = 2*y(i)/3 + z/(3*y(i)^2);
end
```

```
plot(1:n,y)
title('Método iterativo para raíces cúbicas')
xlabel('Número de iteración')
ylabel('Raíz cúbica aproximada')
```

Ejecute el programa anterior con $z = 64$. Cuando haya terminado, escriba los comandos

```
>> e=y-4;
>> plot(1:n,e)
>> semilogy(1:n,e)
```

El primer comando resta la raíz cúbica correcta 4 de cada entrada del vector `y`. Este residuo es el error `e` en cada paso de la iteración. El segundo comando grafica el error, y el tercero traza el error en una gráfica semilogarítmica, utilizando unidades logarítmicas en la dirección *y*.

Se recomienda la creación de un archivo script para guardar el código de MATLAB en caso de que el cálculo requiera algunas líneas más. Un archivo script puede invocar a otros archivos de secuencias de comandos, incluido él mismo. (Al pulsar ⟨cntl⟩-C por lo general se abortan los procesos fuera del control de MATLAB).

B.4 CONTROL DE FLUJO

El ciclo `for` se introdujo en el programa de la raíz cúbica presentado con anterioridad. MATLAB tiene una serie de comandos para controlar el flujo de un programa. Varios de ellos, incluyendo los ciclos `while` y las instrucciones `if` y `break`, resultarán familiares para cualquiera que tenga conocimiento de un lenguaje de programación de alto nivel. Por ejemplo,

```
n=5;
 for i=1:n
  for j=1:n
    a(i,j)=1/(i+j-1);
 end
end
a
```

crea y muestra la matriz de Hilbert de 5×5. El punto y coma evita la impresión repetida de resultados parciales, y la `a` final muestra el último resultado. Tenga en cuenta que cada `for` debe estar acompañado de un `end`. Resulta una buena idea, aunque no es requerido por MATLAB, insertar sangrías en los ciclos para facilitar la lectura del programa.

El comando `while` funciona de manera similar:

```
n=5;i=1;
while i<=n
  j=1;
  while j<=n
    a(i,j)=1/(i+j-1);
    j=j+1;
  end
  i=i+1;
end
a
```

Esto produce el mismo resultado que el doble ciclo `for`.

La instrucción `if` se utiliza para tomar decisiones en el programa, y el comando `break` proporciona una salida hacia el final del programa, como se ilustra a continuación :

```
% Para calcular la n-ésima derivada de sen(x) en x = 0
n=input('Enter n, negative number to quit:)
```

```
if n<=0,break,end
r=rem(n,4)  % rem es la función residuo
if r==0
  y=0
elseif r==1
    y=1
elseif r==2
    y=0
else
    y=-1
end
y
```

Los operadores lógicos & y | significan Y y O, respectivamente. El comando `error` detiene la ejecución del archivo m y reporta la información al usuario.

B.5 FUNCIONES

Además de las funciones incorporadas a la biblioteca como `sin` y `exp`, MATLAB permite la creación de funciones definidas por el usuario. El comando

```
>> f=@(x) exp(sin (2*x))
```

crea una función con entrada x y salida $f(x) = e^{\text{sen } 2x}$. Después de definir `f` como se hizo ya, el comando

```
>> f(0)
```

devuelve el resultado correcto $e^{\text{sen}2(0)} = 1$. Además, la definición con @ asigna un **identificador de función** a `f` que puede pasarse a otra función. Si se crea otra función

```
>> firstderiv=@(f,x,h) (f(x+h)-f(x-h))/(2*h)
```

con tres entradas `f, x, h`, el comando

```
>> firstderiv(f,0,0.0001)
```

devuelve una aproximación a la derivada en 0. Aquí, se ha utilizado el identificador `f` de la función definida por el usuario como una entrada a la función `firstderiv` de MATLAB también definida por el usuario.

Una función de MATLAB puede tener varias entradas y varias salidas. Un ejemplo de una función vectorial de varias variables que tiene tres entradas y tres salidas es la siguiente función que convierte coordenadas rectangulares en esféricas:

```
>> rec2sph=@(x,y,z) [sqrt(x^2+y^2+z^2) acos(z/sqrt(x^2+y^2+z^2))...
   atan2(y,x)]
```

Este método para definir funciones es útil cuando la función puede definirse en una línea. Para ejemplos más complicados, MATLAB permite una segunda manera de definir una función, a través de un archivo m especial. La sintaxis de la primera línea debe mantenerse, como en el siguiente ejemplo, donde el nombre de archivo es `cubrtf.m`:

```
function y=cubrtf(x)
% Aproxima la raíz cúbica de x
% Entrada: número real x, salida: su raíz cúbica
y=1;
n=15;
for i = 1:n
  y = 2*y/3 + x/(3*y^2)
end
```

Aquí, se ha transferido la versión del archivo script del método de aproximaciones a la raíz cúbica a una función de MATLAB. La función puede evaluarse mediante

```
>> c=cubrtf(8)
```

Observe que una función de MATLAB difiere de un archivo m de script en la primera línea. El nombre del archivo, con la .m omitida, debe concordar con el nombre de la función en la primera línea. Las variables en un archivo de función son locales de manera predeterminada, pero pueden hacerse globales mediante el comando `global`.

Si se combinan los dos enfoques anteriores, una función de MATLAB previamente definida, como una función de archivo m, puede recibir la asignación de un identificador de función anteponiendo el signo @. El identificador de la función puede pasarse a otra función. Por ejemplo,

```
>> firstderiv(@cubrtf,1,0.0001)
```

devuelve la aproximación 0.3333 para la derivada de $x^{1/3}$ en $x = 1$.

Una función más complicada puede utilizar varias variables como entradas y varias como salidas. Por ejemplo, a continuación se presenta una función que invoca las funciones existentes de MATLAB `mean` y `std` y las une en un arreglo:

```
function [m,sigma]=stat(x)
% Devuelve la media y la desviación estándar muestrales del vector de entrada x
m=mean(x);
sigma=std(x);
```

Si este archivo `stat.m` reside en su ruta de MATLAB, al escribir `stat(x)`, donde `x` es un vector, obtendrá la media y la desviación estándar de las entradas del vector.

El comando `nargin` proporciona el número de argumentos de entrada a una función. Con este comando, el trabajo de una función puede cambiar, dependiendo de cuántos argumentos se le presentan a ésta. Un ejemplo de `nargin` se da en el programa 0.1 sobre la multiplicación anidada.

Un ejemplo de una función definida por partes es

$$h(x) = \begin{cases} x + 2 & \text{para } x \leq -1 \\ 1 & \text{para } -1 < x \leq 0 \\ \cos x & \text{para } x > 0. \end{cases}$$

La función $h(x)$ puede representarse al crear el archivo de función `h.m` en MATLAB que contiene

```
function y=h(x)
p1=(x<=-1);
p2=(x>-1).*(x<=0);
p3=(x>0);
y=p1.*(x+2)+p2.*1+p3.*cos(x);
```

Aquí se hace uso de la evaluación booleana de las expresiones condicionales como 1 si es verdadero y 0 si es falso. También se utiliza el punto que precede a las operaciones aritméticas para vectorizarlas, lo que permite que la entrada `x` sea un vector de números. Ahora `h` puede pasarse a otras funciones de MATLAB a través de su identificador de función `@h`. Por ejemplo,

```
>> ezplot(@h, [-3 3])
```

grafica la función h por partes, y

```
>> fzero(@h,1)
```

encuentra una raíz de $h(x)$ cercana a 1. ¿El resultado de

```
>> firstderiv(@h,-1,0.0001)
```

es confiable?

B.6 OPERACIONES CON MATRICES

La clave de la potencia y la versatilidad de MATLAB es la sofisticación de la estructura de datos de sus variables. Cada variable en MATLAB es una matriz de $m \times n$ números de doble precisión en punto flotante. Un escalar es simplemente el caso especial de una matriz de 1×1. La sintaxis

```
>> A=[1 2 3
4 5 6]
```

o

```
>> A=[1 2 3;4 5 6]
```

define una matriz A de 2×3. El comando `B=A'` crea una matriz `B` de 3×2 que es la transpuesta de A. Las matrices del mismo tamaño pueden sumarse y restarse con los operadores + y −. El comando `size(A)` devuelve las dimensiones de la matriz `A`, y `length(A)` devuelve el máximo de las dos dimensiones.

MATLAB proporciona muchos comandos que permiten que las matrices sean fáciles de construir. Por ejemplo, `zeros(m,n)` produce una matriz de ceros de tamaño $m \times n$. Si `A` es una matriz, entonces `zeros(size(A))` produce una matriz de ceros del mismo tamaño que `A`. Los comandos `ones(m,n)` y `eye(m,n)` (para la matriz identidad) trabajan en esencia de la misma manera. Por ejemplo,

```
>> A=[eye(2) zeros(2,2);zeros(2,2) eye(2)]
```

es una forma complicada pero precisa de construir la matriz identidad de 4×4.

El operador dos puntos puede utilizarse para extraer una submatriz de una matriz. Por ejemplo,

```
>> b=A(1:3,2)
```

asigna a `b` las tres primeras entradas de la segunda columna de `A`. El comando

```
>> b=A(:,2)
```

asigna a `b` la segunda columna completa de `A`, y

```
>> B=A(:,1:3)
```

asigna a `B` la submatriz consistente en las tres primeras columnas de `A`.

La matriz `A` de $m \times n$ y la matriz `B` de $n \times p$ puede multiplicarse mediante el comando `C=A*B`. Si las matrices tienen tamaños inadecuados, MATLAB se niega a hacer la operación y devuelve un mensaje de error.

B.7 ANIMACIÓN Y PELÍCULAS

El campo de las ecuaciones diferenciales incluye el estudio de los sistemas dinámicos, o "las cosas que se mueven". MATLAB hace que la animación sea fácil, y estos aspectos se explotan en el capítulo 6 para seguir soluciones que cambian con el tiempo.

El programa de ejemplo `bounce.m` de MATLAB que se presenta a continuación muestra una pelota de tenis rebotando de pared a pared en un cuadrado unitario. El primer comando `set` establece los parámetros de la figura actual `(gca)`, incluyendo los límites del eje $0 \leq x, y \leq 1$. El comando `cla` borra la ventana de la figura, y `axis square` iguala las unidades en las direcciones x y y.

Enseguida, se usa el comando `line` para definir un objeto lineal llamado `ball`, junto con sus propiedades. El parámetro `erase` junto con `xor` significa que cada vez que se dibuje la pelota, su posición anterior se borrará. Las cuatro instrucciones `if` en el ciclo `while` hacen que la pelota revierta su velocidad cuando golpea una de las cuatro paredes. El ciclo también contiene un comando `set` que actualiza las coordenadas x y y del objeto lineal `ball`, al establecer su atributos `xdata`

y ydata, respectivamente. El comando drawnow dibuja todos los objetos definidos en la ventana actual de la figura. La velocidad de la pelota en movimiento puede ajustarse con el comando pause y a través de los tamaños de paso hx0 y hy0. El ciclo while es infinito y puede interrumpirse mediante ⟨cntl⟩-C. Enseguida se presenta el programa completo:

```
%bounce.m
% Ilustra la animación de Matlab usando el comando drawnow
% Uso: guarde este archivo en bounce.m, después escriba "bounce"
set(gca,'XLim',[0 1],'YLim',[0 1],'Drawmode','fast', ...
     'Visible','on');
cla
axis square
ball = line('color','r','Marker','o','MarkerSize',10, ...
       'LineWidth',2,'erase','xor','xdata',[],'ydata',[]);
hx0=.005;hy0=.0039;hx=hx0;hy=hy0;
xl=.02;xr=.98;yb=xl;yt=xr;x=.1;y=.1;
while 1 == 1
    if x < xl
        hx= hx0;
    end
    if x > xr
        hx = -hx0;
    end
    if y < yb
        hy = hy0;
    end
    if y > yt
        hy = -hy0;
    end
    x=x+hx;y=y+hy;
    set(ball,'xdata',x,'ydata',y);drawnow;pause(0.01)
end
```

El uso del archivo MakeQTMovie.m, es un modo fácil de hacer películas QuickTime en MATLAB. Cada fotograma de la película será una sola figura de MATLAB. Para comenzar el proceso de hacer una película, obtenga el archivo MakeQTMovie.m de internet. Este archivo lo escribió Malcolm Slaney de Interval Research y su descarga y distribución es gratuita. Coloque el archivo de manera que MATLAB pueda encontrarlo, ya sea en su directorio de trabajo actual o en su ruta de búsqueda. Después, el segmento de código de ejemplo

```
MakeQTMovie('start','filename.mov')
for i=1:n
  (plot a figure)
  MakeQTMovie('addfigure')
end
MakeQTMovie('finish')
```

capturará las *n* figuras fijas y las colocará en un archivo de película QuickTime llamado filename.mov.

Respuestas a los ejercicios seleccionados

CAPÍTULO 0

0.1 Ejercicios

1. (a) $P(x) = 1 + x(1 + x(5 + x(1 + x(6)))), P(1/3) = 2.$
   (b) $P(x) = 1 + x(-5 + x(5 + x(4 + x(-3)))), P(1/3) = 0$
   (c) $P(x) = 1 + x(0 + x(-1 + x(1 + x(2)))), P(1/3) = 77/81$
3. $P(x) = 1 + x^2(2 + x^2(-4 + x^2(1))), P(1/2) = 81/64$
5. (a) 5 (b) 41/4
7. n multiplicaciones y $2n$ sumas.

0.1 Problemas de computadora

1. La respuesta correcta de Q es 51.01275208275, error $= 4.76 \times 10^{-12}$

0.2 Ejercicios

1. (a) 1000000 (b) 10001 (c) 1001111 (d) 11100011
3. (a) 1010.1 (b) $0.\overline{01}$ (c) $0.\overline{101}$ (d) $1100.\overline{1100}$ (e) $110111.\overline{0110}$ (f) $0.\overline{00011}$
5. 11.0010010000111
7. (a) 85 (b) 93/8 (c) 70/3 (d) 20/3 (e) 20/7 (f) 48/7 (g) 283/120 (h) 8

0.3 Ejercicios

1. (a) $1.0000\ldots0000 \times 2^{-2}$ (b) $1.0101\ldots0101 \times 2^{-2}$
   (c) $1.0101\ldots0101 \times 2^{-1}$ (d) $1.11001100\ldots11001101 \times 2^{-1}$
3. $1 \leq k \leq 50$
5. (a) $2\epsilon_{máq}$ (b) $4\epsilon_{máq}$
7. (a) 4020000000000000 (b) 4035000000000000 (c) 3fc0000000000000 (d) 3fd5555555555555
   (e) 3fe5555555555555 (f) 3fb999999999999a (g) bfb999999999999a (h) bfc999999999999a
9. (a) Observe que $(7/3 - 4/3) - 1 = \epsilon_{máq}$ en precisión doble. (b) No, $(4/3 - 1/3) - 1 = 0$.
11. No, la ley asociativa falla.
13. (a) 2, representado por 010...0 (b) 2^{-511}, representado por 0010...0 (c) 0, representado por 10...0
15. (a) 2^{-50} (b) 0 (c) 2^{-50}

0.4 Ejercicios

1. (a) Pérdida de significancia cercana a $x = 2\pi n$, n entero. Reescrita como $-1/(1 + \sec x)$ (b) Pérdida de significancia cercana a $x = 0$. Reescrita como $3 - 3x + x^2$ (c) Pérdida de significancia cercana a $x = 0$. Reescrita como $2x/(x^2 - 1)$
3. $x_1 = -(b + \sqrt{b^2 + 4 \times 10^{-12}})/2,\ x_2 = (2 \times 10^{-12})/(b + \sqrt{b^2 + 4 \times 10^{-12}})$

0.4 Problemas de computadora

1. (a)

x	original	modificado
0.100000000000000	−0.49874791371143	−0.49874791371143
0.010000000000000	−0.49998749979096	−0.49998749979166
0.001000000000000	−0.49999987501429	−0.49999987499998
0.000100000000000	−0.49999999362793	−0.49999999875000
0.000010000000000	−0.50000004133685	−0.49999999998750
0.000001000000000	−0.50004445029084	−0.49999999999987
0.000000100000000	−0.51070259132757	−0.50000000000000
0.000000010000000	0	−0.50000000000000
0.000000001000000	0	−0.50000000000000
0.000000000100000	0	−0.50000000000000
0.000000000010000	0	−0.50000000000000
0.000000000001000	0	−0.50000000000000
0.000000000000100	0	−0.50000000000000
0.000000000000010	0	−0.50000000000000
0.000000000000001	0	−0.50000000000000

(b)

x	original	modificado
0.100000000000000	2.71000000000000	2.71000000000000
0.010000000000000	2.97010000000001	2.97010000000000
0.001000000000000	2.99700100000000	2.99700100000000
0.000100000000000	2.99970000999905	2.99970001000000
0.000010000000000	2.99997000008379	2.99997000010000
0.000001000000000	2.99999700015263	2.99999700000100
0.000000100000000	2.99999969866072	2.99999970000001
0.000000010000000	2.99999998176759	2.99999997000000
0.000000001000000	2.99999991515421	2.99999999700000
0.000000000100000	3.00000024822111	2.99999999970000

x	original	modificado
0.000000000001000	3.00000024822111	2.99999999997000
0.000000000000100	2.99993363483964	2.99999999999700
0.000000000000010	3.00093283556180	2.99999999999970
0.000000000000001	2.99760216648792	2.99999999999997

3. 6.127×10^{-13}
5. 2.23322×10^{-10}

0.5 Ejercicios

1. (a) $f(0)f(1) = -2 < 0$ implica que $f(c) = 0$ para alguna c en $(0, 1)$ por el teorema del valor medio.
   (b) $f(0)f(1) = -9 < 0$ implica que $f(c) = 0$ para alguna c en $(0, 1)$ (c) $f(0)f(1/2) = -1/2 < 0$ implica que $f(c) = 0$ para alguna c en $(0, 1/2)$.
3. (a) $c = 2/3$ (b) $c = 1/\sqrt{2}$ (c) $c = 1/(e-1)$
5. (a) $P(x) = 1 + x^2 + 1/2x^4$ (b) $P(x) = 1 - 2x^2 + 2/3x^4$ (c) $P(x) = x - x^2/2 + x^3/3 - x^4/4 + x^5/5$
   (d) $P(x) = x^2 - x^4/3$
7. (a) $P(x) = (x-1) - (x-1)^2/2 + (x-1)^3/3 - (x-1)^4/4$ (b) $P(0.9) = -0.1053583$, $P(1.1) = 0.0953083$
   (c) límite de error = 0.000003387 para $x = 0.9$, 0.000002 para $x = 1.1$ (d) Error real ≈ 0.00000218 en $x = 0.9$, 0.00000185 en $x = 1.1$
9. $\sqrt{1+x} = 1 + x/2 \pm x^2/8$. Para $x = 1.02$, $\sqrt{1.02} \approx 1.01 \pm 0.00005$. El valor real es $\sqrt{1.02} = 1.0099505$, error = 0.0000495

CAPÍTULO 1

1.1 Ejercicios

1. (a) [2, 3] (b) [1, 2] (c) [6, 7]
3. (a) 2.125 (b) 1.125 (c) 6.875
5. (a) [2, 3] (b) 33 iteraciones

1.1 Problemas de computadora

1. (a) 2.080084 (b) 1.169726 (c) 6.776092
3. (a) Intervalos $[-2, -1]$, $[-1, 0]$, $[1, 2]$, raíces -1.641784, -0.168254, 1.810038
   (b) Intervalos $[-2, -1]$, $[-0.5, 0.5]$, $[0.5, 1.5]$, raíces -1.023482, 0.163822, 0.788941
   (c) Intervalos $[-1.7, -0.7]$, $[-0.7, 0.3]$, $[0.3, 1.3]$, raíces -0.818094, 0, 0.506308
5. (a) [1, 2], 27 iteraciones, 1.25992105 (b) [1, 2], 27 iteraciones, 1.44224957 (c) [1, 2], iteraciones, 1.70997595
7. Primera raíz -17.188498, determinante correcto hasta 2 cifras decimales; segunda raíz 9.708299, determinante correcto hasta 3 cifras decimales.
9. $H = 635.5$ mm

1.2 Ejercicios

1. (a) $-\sqrt{3}, \sqrt{3}$ (b) $1, 2$ (c) $(5 \pm \sqrt{17})/2$
3. Se demuestra por sustitución.
5. B, D
7. (a) convergente localmente (b) divergente (c) divergente
9. (a) 0 es localmente convergente, 1 es divergente (b) 1/2 es localmente convergente, 3/4 es divergente
11. (a) Por ejemplo, $x = x^3 + e^x$, $x = (x - e^x)^{1/3}$ y $x = \ln(x - x^3)$; (b) Por ejemplo, $x = 9x^2 + 3/x^3$, $x = 1/9 - 1/(3x^4)$ y $x = (x^5 - 9x^6)/3$
13. (a) 0.3, -1.3 (b) 0.3 (c) más lento
15. Todas convergen a $\sqrt{5}$. De la más rápida a la más lenta: (B), (C), (A).
17. $g(x) = \sqrt{(1 - x)/2}$ es localmente convergente en 1/2, y $g(x) = -\sqrt{(1 - x)/2}$ es localmente convergente en -1.
19. $g(x) = (x + A/x^2)/2$ converge a $A^{1/3}$.
21. (a) Sustituya y verifique (b) $|g'(r)| > 1$ para los tres puntos fijos r.
23. $g'(r^2) > 1$
27. (a) $x = x - x^3$ implica que $x = 0$ (b) Si $0 < x_i < 1$, entonces $x_{i+1} = x_i - x_i^3 = x_i(1 - x_i^2) < x_i$, y $0 < x_{i+1} < x_i < 1$.
    (c) La secuencia monotónica acotada x_i converge a un límite L, que debe ser un punto fijo. Por lo tanto $L = 0$.
29. (a) $c < -2$ (b) $c = -4$
31. El intervalo abierto $(-5/4, 5/4)$ de estimaciones iniciales convergen al punto fijo 1/4; las dos estimaciones iniciales $-5/4$, $5/4$ convergen a $-5/4$.
33. (a) Elija $a = 0$ y $|b| < 1$, c arbitraria. (b) Elija $a = 0$ y $|b| > 1$, c arbitraria.

1.2 Problemas de computadora

1. (a) 1.76929235 (b) 1.67282170 (c) 1.12998050
3. (a) 1.73205081 (b) 2.23606798
5. El punto fijo es $r = 0.641714$ y $S = |g'(r)| \approx 0.959$
7. (a) $0 < x_0 < 1$ (b) $1 < x_0 < 2$ (c) $x_0 > 2.2$, por ejemplo

1.3 Ejercicios

1. (a) EAD = 0.01, EAT = 0.04 (b) EAD = 0.01 EAT = 0.0016 (c) EAD = 0.01, EAT = 0.000064 (d) EAD = 0.01, EAT = 0.342
3. (a) 2 (b) EAD = 0.0001, EAT = 5×10^{-9}
5. EAT = $|a|$ EAD
7. (b) $(-1)^j (j-1)!(20-j)!$

1.3 Problemas de computadora

1. (a) $m = 3$ (b) $x_a = -2.0735 \times 10^{-8}$, EAD = 2.0735×10^{-8}, EAT = 0
3. (a) x_a = EAD = 0.000169, EAT = 0 (b) Termina después de 13 iteraciones, $x_a = -0.00006103$
5. Raíz predicha = $r + \Delta r = 4 + 4^6 10^{-6}/6 = 4.0006826$, raíz real = 4.0006825

1.4 Ejercicios

1. (a) $x_1 = 2, x_2 = 18/13$ (b) $x_1 = 1, x_2 = 1$ (c) $x_1 = -1, x_2 = -2/3$
3. (a) $r = -1, e_{i+1} = \frac{5}{2}e_i^2$; $r = 0, e_{i+1} = 2e_i^2$; $r = 1, e_{i+1} = \frac{2}{3}e_i$ (b) $r = -1/2, e_{i+1} = 2e_i^2$; $r = 1, e_{i+1} = 2/3e_i$
5. $r = 0$, método de Newton; $r = 1/2$, método de bisección
7. No, 2/3
9. $x_{i+1} = (x_i + A/x_i)/2$
11. $x_{i+1} = (n-1)x_i/n + A/(nx_i^{n-1})$
13. (a) 0.75×10^{-12} (b) 0.5×10^{-18}

1.4 Problemas de computadora

1. (a) 1.76929235 (b) 1.67282170 (c) 1.12998050
3. (a) $r = -2/3, m = 3$ (b) $r = 1/6, m = 2$
5. $r = 3.2362$ m
7. −1.197624, convergencia cuadrática; 0, convergencia lineal, $m = 4$; 1.530134, convergencia cuadrática
9. 0.857143, convergencia cuadrática, $M = 2.414$; 2, convergencia lineal, $m = 3$, $S = 2/3$
11. Estimación inicial = 1.75, solución $V = 1.70$ L
13. (a) 3/4 (c) $f(x)$ no es diferenciable en $x = 3/4$.

1.5 Ejercicios

1. (a) $x_2 = 8/5, x_3 = 1.742268$ (b) $x_2 = 1.578707, x_3 = 1.66016$ (c) $x_2 = 1.092907, x_3 = 1.119357$
3. (a) $x_3 = -1/5, x_4 = -0.11996018$ (b) $x_3 = 1.757713, x_4 = 1.662531$ (c) $x_3 = 1.139481, x_4 = 1.129272$
7. Del más rápido al más lento, (B), (D), (A) y (C) que no converge (b) El método de Newton converge más rápidamente.

1.5 Problemas de computadora

1. (a) 1.76929235 (b) 1.67282170 (c) 1.12998050
3. (a) 1.76929235 (b) 1.67282170 (c) 1.12998050
5. `fzero` converge al cero que no es una raíz, igual que el método de bisección

CAPÍTULO 2

2.1 Ejercicios

1. (a) $[4, 2]$ (b) $[5, -3]$ (c) $[1, 3]$
3. (a) $[1/3, 1, 1]$ (b) $[2, -1/2, -1]$
5. Aproximadamente 27 veces más tiempo.
7. Aproximadamente 61 segundos.

2.1 Problemas de computadora

1. (a) $[1, 1, 2]$ (b) $[1, 1, 1]$ (c) $[-1, 3, 2]$

2.2 Ejercicios

1. (a) $\begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ 0 & -4 \end{bmatrix}$ (c) $\begin{bmatrix} 1 & 0 \\ -5/3 & 1 \end{bmatrix}\begin{bmatrix} 3 & -4 \\ 0 & -14/3 \end{bmatrix}$
3. (a) $[-2, 1]$ (b) $[-1, 1]$
5. $[1, -1, 1, -1]$
7. 5 min, 33 s
9. 300

2.3 Ejercicios

1. (a) 7 (b) 8
3. (a) EAD = 2, EAT = 0.0002, FME = 20001 (b) EAD = 1, EAT = 0.0001, FME = 20001 (c) EAD = 1, EAT = 2.0001, FME = 1 (d) EAD = 3, EAT = 0.0003, FME = 20001 (d) EAD = 3.0001, EAT = 0.0002, FME = 30002.5
5. (a) ERAD = 3, ERAT = 3/7, FME = 7 (b) ERAD = 3, ERAT = 1/7, FME = 21 (c) ERAD = 1, ERAT = 1/7, FME = 7 (d) ERAD = 2, ERAT = 6/7, FME = 7/3 (e) 21
7. 137/60

13. (a) $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ (b) $\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$

15. $LU = \begin{bmatrix} 1 & 0 & 0 \\ 0.1 & 1 & 0 \\ 0 & -5000 & 1 \end{bmatrix}\begin{bmatrix} 10 & 20 & 1 \\ 0 & -0.01 & 5.9 \\ 0 & 0 & 29501 \end{bmatrix}$, mayor multiplicador = 5000

2.3 Problemas de computadora

Las respuestas dadas a los problemas de computadora en esta sección son sólo ilustrativas; los resultados pueden variar ligeramente debido a los detalles de implementación.

1.

	n	EAD	FME	cond(A)
(a)	6	5.35×10^{-10}	3.69×10^{6}	7.03×10^{7}
(b)	10	1.10×10^{-3}	9.05×10^{12}	1.31×10^{14}

3.

n	EAD	FME	cond(A)
100	4.62×10^{-12}	3590	9900
200	4.21×10^{-11}	23010	39800
300	7.37×10^{-11}	50447	89700
400	1.20×10^{-10}	55019	159600
500	2.56×10^{-10}	91495	249500

5. $n \geq 13$

2.4 Ejercicios

1. (a) $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 3 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{bmatrix}\begin{bmatrix} 2 & 3 \\ 0 & \frac{3}{2} \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{bmatrix}\begin{bmatrix} 2 & 4 \\ 0 & 1 \end{bmatrix}$

   (c) $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 5 \\ 5 & 12 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{5} & 1 \end{bmatrix}\begin{bmatrix} 5 & 12 \\ 0 & \frac{13}{5} \end{bmatrix}$ (d) $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

3. (a) $[-2, 1]$ (b) $[-1, 1, 1]$

5. $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

7. $\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

9. (a) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 8 \end{bmatrix}$

   (b) $P = I$, L es triangular inferior con todas las entradas no diagonales iguales a -1, las entradas de U distintas de cero son $u_{ii} = 1$ para $1 \leq i \leq n-1$ y $u_{i_n} = 2^{i-1}$ para $1 \leq i \leq n$.

2.5 Ejercicios

1. (a) Jacobi $[u_2, v_2] = [7/3, 17/6]$ Gauss-Seidel $[u_2, v_2] = [47/18, 119/36]$ (b) Jacobi $[u_2, v_2, w_2] = [1/2, 1, 1/2]$ Gauss-Seidel $[u_2, v_2, w_2] = [1/2, 3/2, 3/4]$ (c) Jacobi $[u_2, v_2, w_2] = [10/9, -2/9, 2/3]$ Gauss-Seidel $[u_2, v_2, w_2] = [43/27, 14/81, 262/243]$

3. (a) $[u_2, v_2] = [59/16, 213/64]$ (b) $[u_2, v_2, w_2] = [9/8, 39/16, 81/64]$ (c) $[u_2, v_2, w_2] = [1, 1/2, 5/4]$

2.5 Problemas de computadora

1. $n = 100$, 36 iteraciones, EAT $= 4.58 \times 10^{-7}$; $n = 100000$, 48 iteraciones, EAT $= 2.70 \times 10^{-6}$
5. (a) 21 iteraciones, EAT $= 4.78 \times 10^{-7}$ (b) 16 iteraciones, EAT $= 1.55 \times 10^{-6}$

2.6 Ejercicios

1. (a) $x^T Ax = x_1^2 + 3x_2^2 > 0$ para $x \neq 0$
   (b) $x^T Ax = (x_1 + 3x_2)^2 + x_2^2 > 0$ para $x \neq 0$
   (c) $x_1^2 + 2x_2^2 + 3x_3^2 > 0$ para $x \neq 0$

3. (a) $R = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \end{bmatrix}$ (b) $R = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}$ (c) $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{2} & 0 \\ 0 & 0 & \sqrt{3} \end{bmatrix}$

5. (a) $R = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$ (b) $R = \begin{bmatrix} 2 & -1 \\ 0 & 1/2 \end{bmatrix}$ (c) $R = \begin{bmatrix} 5 & 1 \\ 0 & 5 \end{bmatrix}$ (d) $R = \begin{bmatrix} 1 & -2 \\ 0 & 1 \end{bmatrix}$

7. (a) $[2, -1]$ (b) $[3, 1]$

9. $x^T Ax = (x_1 + 2x_2)^2 + (d-4)x_2^2$. Si $d > 4$, las expresiones pueden ser 0 sólo si $0 = x_2 = x_1 + 2x_2$, lo cual implica que $x_1 = x_2 = 0$.

11. $d > 1$

13. (a) $[3, -1]$ (b) $[-1, 1]$

15. $\alpha_1 = 1/A, x_1 = b/A, r_1 = b - Ab/A = 0$

2.6 Problemas de computadora

1. (a) $[2, 2]$ (b) $[3, -1]$

3. (a) $[-4, 60, -180, 140]$ (b) $[-8, 504, -7560, 46200, -138600, 216216, -168168, 51480]$

2.7 Ejercicios

1. (a) $\begin{bmatrix} 3u^2 & 0 \\ v^3 & 3uv^2 \end{bmatrix}$ (b) $\begin{bmatrix} v\cos uv & u\cos uv \\ ve^{uv} & ue^{uv} \end{bmatrix}$ (c) $\begin{bmatrix} 2u & 2v \\ 2(u-1) & 2v \end{bmatrix}$

   (d) $\begin{bmatrix} 2u & 1 & -2w \\ vw\cos uvw & uw\cos uvw & uv\cos uvw \\ vw^4 & uw^4 & 4uvw^3 \end{bmatrix}$

3. (a) $(1/2, \pm\sqrt{3}/2)$ (b) $(\pm 2/\sqrt{5}, \pm 2/\sqrt{5})$ (c) $(4(1+\sqrt{6})/5, \pm\sqrt{3+8\sqrt{6}}/5)$

5. (a) $x_1 = [0, 1], x_2 = [0, 0]$ (b) $x_1 = [0, 0], x_2 = [0.8, 0.8]$ (c) $x_1 = [8, 4], x_2 = [9.0892, -12.6103]$

2.7 Problemas de computadora

1. (a) $(1/2, \pm\sqrt{3}/2)$ (b) $(\pm 2/\sqrt{5}, \pm 2/\sqrt{5})$ (c) $(4(1+\sqrt{6})/5, \pm\sqrt{3+8\sqrt{6}}/5)$

3. $\pm[0.50799200040795, 0.86136178666199]$

5. (a) $[1, 1, 1], [1/3, 1/3, 1/3]$ (b) $[1, 2, 3], [17/9, 22/9, 19/9]$

7. (a) 11 iteraciones dan la raíz $(1/2, \sqrt{3}/2)$ hasta 15 decimales (b) 13 iteraciones dan la raíz $(2/\sqrt{5}, 2/\sqrt{5})$ hasta 15 decimales (c) 14 iteraciones dan la raíz $(4(1+\sqrt{6})/5, \sqrt{3+8\sqrt{6}}/5)$ hasta 15 decimales

9. Mismas respuestas que en el problema de computadora 5.

11. Mismas respuestas que en el problema de computadora 5.

CAPÍTULO 3

3.1 Ejercicios

1. (a) $P(x) = \dfrac{(x-2)(x-3)}{(0-2)(0-3)} + 3\dfrac{x(x-3)}{(2-0)(2-3)}$

   (b) $P(x) = \dfrac{(x+1)(x-3)(x-5)}{(2+1)(2-3)(2-5)} + \dfrac{(x+1)(x-2)(x-5)}{(3+1)(3-2)(3-5)} + 2\dfrac{(x+1)(x-2)(x-3)}{(5+1)(5-2)(5-3)}$

   (c) $P(x) = -2\dfrac{(x-2)(x-4)}{(0-2)(0-4)} + \dfrac{x(x-4)}{(2-0)(2-4)} + 4\dfrac{x(x-2)}{4(4-2)}$

3. (a) Uno, $P(x) = 3 + (x+1)(x-2)$ (b) Ninguno (c) Un número infinito, por ejemplo $P(x) = 3 + (x+1)(x-2) + C(x+1)(x-1)(x-2)(x-3)^3$, donde C es una constante distinta de cero
5. (a) $P(x) = 4 - 2x$ (b) $P(x) = 4 - 2x + A(x+2)x(x-1)(x-3)$ para $A \neq 0$
7. 4
9. (a) $P(x) = 10(x-1)\cdots(x-6)/6!$ (b) Igual que (a)
11. Ninguno
13. 4/2
15. $P(x) = -x - (x-1)(x-2)\cdots(x-25)/24!$
17. (a) 316 (b) 465

3.1 Problemas de computadora

1. (a) 4494564854 (b) 4454831984 (c) 4472888288

3.2 Ejercicios

1. (a) $P_2(x) = \frac{2}{\pi}x - \frac{4}{\pi^2}x(x-\pi/2)$ (b) $P_2(\pi/4) = 3/4$ (c) $\pi^3/128 \approx 0.242$ (d) $|\sqrt{2}/2 - 3/4| \approx 0.043$
3. (a) 7.06×10^{-11} (b) al menos 9 posiciones decimales, puesto que $7.06 \times 10^{-11} < 0.5 \times 10^{-9}$
5. Se espera que los errores en $x = 0.35$ sean menores; aproximadamente 5/21 del tamaño del error en $x = 0.55$.

3.2 Problemas de computadora

1. (a) $P_4(x) = 1.433329 + (x-0.6)(1.98987 + (x-0.7)(3.2589 + (x-0.8)(3.680667 + (x-0.9)(4.000417))))$ (b) $P_4(0.82) = 1.95891$, $P_4(0.98) = 2.612848$ (c) El límite superior del error en $x = 0.82$ es 0.0000537, el error real es 0.0000234. El límite superior del error en $x = 0.98$ es 0.000217, el error real es 0.000107.
3. -1.952×10^{12} brls/día. La estimación no tiene mucho sentido, debido al fenómeno de Runge.

3.3 Ejercicios

1. (a) $\cos\pi/12, \cos\pi/4, \cos 5\pi/12, \cos 7\pi/12, \cos 3\pi/4, \cos 11\pi/12$
   (b) $2\cos\pi/8, 2\cos 3\pi/8, 2\cos 5\pi/8, 2\cos 7\pi/8$
   (c) $8 + 4\cos\pi/12, 8 + 4\cos\pi/4, 8 + 4\cos 5\pi/12, 8 + 4\cos 7\pi/12, 8 + 4\cos 3\pi/4, 8 + 4\cos 11\pi/12$
   (d) $1/5 + 1/2\cos\pi/10, 1/5 + 1/2\cos 3\pi/10, 1/5, 1/5 + 1/2\cos 7\pi/10, 1/5 + 1/2\cos 9\pi/10$
3. 0.000118, 3 dígitos correctos
5. 0.00521
7. $d = 14$
9. (a) -1 (b) 1 (c) 0 (d) 1 (e) 1 (f) $-1/2$

3.4 Ejercicios

1. (a) No es una spline cúbica (b) spline cúbica
3. (a) $c = 9/4$, natural (b) $c = 4$, terminada parabólicamente y sin nudo (c) $c = 5/2$, sin nudo
5. Una, $S_1(x) = S_2(x) = x$
7. (a) $\begin{cases} \frac{1}{2}x + \frac{1}{2}x^3 & \text{en } [0,1] \\ 1 + 2(x-1) + \frac{3}{2}(x-1)^2 - \frac{1}{2}(x-1)^3 & \text{en } [1,2] \end{cases}$

   (b) $\begin{cases} 1 - (x+1) + \frac{1}{4}(x+1)^3 & \text{en } [-1,1] \\ 1 + 2(x-1) + \frac{3}{2}(x-1)^2 - \frac{1}{2}(x-1)^3 & \text{en } [1,2] \end{cases}$

9. $-3, -12$

11. (a) Una, $S_1(x) = S_2(x) = 2 - 4x + 2x^2$ (b) Un número infinito, $S_1(x) = S_2(x) = 2 - 4x + 2x^2 + cx(x-1)(x-2)$, para una c arbitraria.

13. (a) $b_1 = 1, c_3 = -8/9$ (b) No. (c) Las restricciones son $S'(0) = 1$ y $S'(3) = -1/3$.

15. Sí. Las secciones a la izquierda y la derecha de la spline deben ser lineales.

17. $S_2(x) = 1 + dx^3$ para una d arbitraria.

19. Existe un número infinito de parábolas que pasan a través de dos puntos arbitrarios con $x_1 \neq x_2$; cada una es una spline cúbica terminada parabólicamente.

21. (a) Un número infinito (b) $S_1(x) = S_2(x) = x^2 + dx(x-1)(x-2)$ donde $d \neq 0$.

3.4 Problemas de computadora

1. (a) $$S(x) = \begin{cases} 3 + \frac{8}{3}x - \frac{2}{3}x^3 & \text{en } [0,1] \\ 5 + \frac{2}{3}(x-1) - 2(x-1)^2 + \frac{1}{3}(x-1)^3 & \text{en } [1,2] \\ 4 - \frac{7}{3}(x-2) - (x-2)^2 + \frac{1}{3}(x-2)^3 & \text{en } [2,3] \end{cases}$$

(b) $$S(x) = \begin{cases} 3 + 2.5629(x+1) - 0.5629(x+1)^3 & \text{en } [-1,0] \\ 5 + 0.8742x - 1.6887x^2 + 0.3176x^3 & \text{en } [0,3] \\ 1 - 0.6824(x-3) + 1.1698(x-3)^2 - 0.4874(x-3)^3 & \text{en } [3,4] \\ 1 + 0.1950(x-4) - 0.2925(x-4)^2 + 0.0975(x-4)^3 & \text{en } [4,5] \end{cases}$$

3. $$S(x) = \begin{cases} 1 + \frac{149}{56}x - \frac{37}{56}x^3 & \text{en } [0,1] \\ 3 + \frac{19}{28}(x-1) - \frac{111}{56}(x-1)^2 + \frac{73}{56}(x-1)^3 & \text{en } [1,2] \\ 3 + \frac{5}{8}(x-2) + \frac{27}{14}(x-2)^2 - \frac{87}{56}(x-2)^3 & \text{en } [2,3] \\ 4 - \frac{5}{28}(x-3) - \frac{153}{56}(x-3)^2 + \frac{51}{56}(x-3)^3 & \text{en } [4,5] \end{cases}$$

1. $$S(x) = \begin{cases} 1 + 1.8006x + \frac{3}{2}x^2 - 1.3006x^3 & \text{en } [0,1] \\ 3 + 0.8988(x-1) - 2.4018(x-1)^2 + 1.5030(x-1)^3 & \text{en } [1,2] \\ 3 + 0.6042(x-2) + 2.1071(x-2)^2 - 1.7113(x-2)^3 & \text{en } [2,3] \\ 4 - 0.3155(x-3) - 3.0268(x-3)^2 + 1.3423(x-3)^3 & \text{en } [4,5] \end{cases}$$

3. $$S(x) = \begin{cases} 1 - 2x + \frac{57}{7}x^2 - \frac{29}{7}x^3 & \text{en } [0,1] \\ 3 + \frac{13}{7}(x-1) - \frac{30}{7}(x-1)^2 + \frac{17}{7}(x-1)^3 & \text{en } [1,2] \\ 3 + \frac{4}{7}(x-2) + 3(x-2)^2 - \frac{18}{7}(x-2)^3 & \text{en } [2,3] \\ 4 - \frac{8}{7}(x-3) - \frac{33}{7}(x-3)^2 + \frac{27}{7}(x-3)^3 & \text{en } [4,5] \end{cases}$$

5. $$S(x) = \begin{cases} x - 0.0006x^2 - 0.1639x^3 & \text{en } [0, \frac{\pi}{8}] \\ \operatorname{sen}\frac{\pi}{8} + 0.9237(x - \frac{\pi}{8}) - 0.1937(x - \frac{\pi}{8})^2 - 0.1396(x - \frac{\pi}{8})^3 & \text{en } [\frac{\pi}{8}, \frac{\pi}{4}] \\ \frac{\sqrt{2}}{2} + 0.7070(x - \frac{\pi}{4}) - 0.3582(x - \frac{\pi}{4})^2 - 0.0931(x - \frac{\pi}{4})^3 & \text{en } [\frac{\pi}{4}, \frac{3\pi}{8}] \\ \operatorname{sen}\frac{3\pi}{8} + 0.3826(x - \frac{3\pi}{8}) - 0.4679(x - \frac{3\pi}{8})^2 - 0.0327(x - \frac{3\pi}{8})^3 & \text{en } [\frac{3\pi}{8}, \frac{\pi}{2}] \end{cases}$$

7. $n = 48$

9. (a) 322.6 (b) 318.8 (c) una spline sin nudo idéntica a la solución del ejercicio 3.1.13

3.5 Ejercicios

1. (a) $\begin{cases} x(t) = 6t^2 - 5t^3 \\ y(t) = 6t - 12t^2 + 6t^3 \end{cases}$ (b) $\begin{cases} x(t) = 1 - 3t - 3t^2 + 3t^3 \\ y(t) = 1 - 3t + 3t^2 \end{cases}$ (c) $\begin{cases} x(t) = 1 + 3t^2 - 2t^3 \\ y(t) = 2 + 3t - 3t^2 \end{cases}$

3. $\begin{cases} x(t) = 1 + 6t^2 - 4t^3 \\ y(t) = 2 + 6t^2 - 4t^3 \end{cases}$ $\begin{cases} x(t) = 3 + 6t^2 - 4t^3 \\ y(t) = 4 - 9t^2 + 6t^3 \end{cases}$ $\begin{cases} x(t) = 5 - 12t^2 + 8t^3 \\ y(t) = 1 + 3t^2 - 2t^3 \end{cases}$

5. El número 3.

7. $\begin{cases} x(t) = -1 + 6t^2 - 4t^3 \\ y(t) = 4t - 4t^2 \end{cases}$

9. (a) $\begin{cases} x(t) = 1 + 3t - 9t^2 + 5t^3 \\ y(t) = 6t^2 - 5t^3 \\ z(t) = 3t^2 - 3t^3 \end{cases}$ (b) $\begin{cases} x(t) = 1 - 6t^2 + 6t^3 \\ y(t) = 1 + 3t - 9t^2 + 6t^3 \\ z(t) = 2 + 3t - 12t^2 + 8t^3 \end{cases}$

   (c) $\begin{cases} x(t) = 2 + 3t - 12t^2 + 10t^3 \\ y(t) = 1 \\ z(t) = 1 + 6t^2 - 4t^3 \end{cases}$

CAPÍTULO 4

4.1 Ejercicios

1. (a) $\bar{x} = [-1/7, 10/7], ||e||_2 = \sqrt{14}/7$ (b) $\bar{x} = [-1/2, 2], ||e||_2 = \sqrt{6}/2$
   (c) $\bar{x} = [16/19, 16/19], ||e||_2 = 2.013$

3. $\bar{x} = [4, x_2]$ para una x_2 arbitraria

7. (a) $y = 1/5 - 6/5t$, RMEC $= \sqrt{2/5} \approx 0.6325$ (b) $y = 6/5 + 1/2t$, RMEC $= \sqrt{26}/10 \approx 0.5099$

9. (a) $y = 0.3481 + 1.9475t - 0.1657t^2$, RMEC $= 0.5519$ (b) $y = 2.9615 - 1.0128t + 0.1667t^2$, RMEC $= 0.4160$ (c) $y = 4.8 - 1.2t$, RMEC $= 0.4472$

11. $h(t) = 0.475 + 141.525t - 4.905t^2$, altura máx $= 1021.3m$, tiempo de aterrizaje $= 28.86$ s.

4.1 Problemas de computadora

1. (a) $\bar{x} = [2.5246, 0.6616, 2.0934], ||e||_2 = 2.4135$ (b) $\bar{x} = [1.2739, 0.6885, 1.2124, 1.7497], ||e||_2 = 0.8256$

3. (a) $2,996,236,899 + 76,542,140(t - 1960)$ RMEC $= 36,751,088$
   (b) $3,028,751,748 + 67,871,514(t - 1960)$ $1\,216,766(t - 1960)^2$, RMEC $= 17,129,714$;
   estimaciones de 1980: (a) 4,527,079,702 (b) 4,472,888,288; La parábola da una mejor estimación.

5. (a) $c_1 = 9510.1, c_2 = -8314.36$, RMEC $= 518.3$ (b) precio de venta $= 68.7$ centavos maximiza la utilidad.

7. (a) $y = 0.0769$, RMEC $= 0.265$ (b) $y = 0.1748 - 0.0297t^2$, RMEC $= 0.2519$

9. (a) 4 posiciones decimales correctas $P_5(t) = 1.000009 + 0.999983t + 1.000012t^2 + 0.999996t^3 + 1.000000t^4 + 1.000000t^5$; $\text{cond}(A^T A) = 2.72 \times 10^{13}$ (b) 1 posición decimal correcta $P_6(t) = 0.99 + 1.02t + 0.98t^2 + 1.01t^3 + t^4 + t^5 + t^6$; $\text{cond}(A^T A) = 2.55 \times 10^{16}$ (c) $P_8(t)$ no tiene posiciones decimales correctas, $\text{cond}(A^T A) = 1.41 \times 10^{19}$

4.2 Ejercicios

1. (a) $y = 3/2 - 1/2\cos 2\pi t + 3/2 \text{ sen } 2\pi t$, $\|e\|_2 = 0$, RMEC $= 0$ (b) $y = 7/4 - 1/2\cos 2\pi t + \text{sen } 2\pi t$, $\|e\|_2 = 1/2$, RMEC $= 1/4$ (c) $y = 9/4 + 3/4 \cos 2\pi t$, $\|e\|_2 = 1/\sqrt{2}$, RMEC $= 1/(2\sqrt{2})$

3. (a) $y = 1.932e^{0.3615t}$, $\|e\|_2 = 1.2825$, (b) $y = 2^{t-1/4}$, $\|e\|_2 = 0.9982$

5. (a) $y = 5.5618t^{-1.3778}$, RMEC $= 0.2707$ (b) $y = 2.8256t^{0.7614}$, RMEC $= 0.7099$

4.2 Problemas de computadora

1. $y = 5.5837 + 0.7541\cos 2\pi t + 0.1220\,\text{sen}\,2\,\pi t + 0.1935\cos 4\pi t$ M brls/día, RMEC $= 0.1836$
3. $P(t) = 3,079,440,361e^{0.0174(t-1960)}$, la estimación de 1980 es $P(20) = 4{,}361{,}485{,}000$, error de estimación ≈ 91 millones
5. (a) $t_{\text{máx}} = -1/c_2$ (b) vida media ≈ 7.81 hrs.

4.3 Ejercicios

1. (a) $\begin{bmatrix} 0.8 & -0.6 \\ 0.6 & 0.8 \end{bmatrix}\begin{bmatrix} 5 & 0.6 \\ 0 & 0.8 \end{bmatrix}$ (b) $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} \sqrt{2} & \frac{3\sqrt{2}}{2} \\ 0 & \frac{\sqrt{2}}{2} \end{bmatrix}$

   (c) $\begin{bmatrix} \frac{2}{3} & \frac{\sqrt{2}}{6} & \frac{\sqrt{2}}{2} \\ \frac{1}{3} & -\frac{2\sqrt{2}}{3} & 0 \\ \frac{2}{3} & \frac{\sqrt{2}}{6} & -\frac{\sqrt{2}}{2} \end{bmatrix}\begin{bmatrix} 3 & 1 \\ 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix}$ (d) $\begin{bmatrix} \frac{4}{5} & 0 & -\frac{3}{5} \\ 0 & 1 & 0 \\ \frac{3}{5} & 0 & \frac{4}{5} \end{bmatrix}\begin{bmatrix} 5 & 10 & 5 \\ 0 & 2 & -2 \\ 0 & 0 & 5 \end{bmatrix}$
3. (a) – (d) igual que en el ejercicio 1.
5. (a) – (d) igual que en el ejercicio 1.
7. (a) $\bar{x} = [4, -1]$ (b) $\bar{x} = [-11/18, 4/9]$

4.3 Problemas de computadora

5. (a) $\bar{x} = [1.6154, 1.6615]$, $||e||_2 = 0.3038$ (b) $\bar{x} = [2.0588, 2.3725, 1.5784]$, $||e||_2 = 0.2214$
7. (a) $\bar{x} = [1, \ldots, 1]$ hasta 10 posiciones decimales correctas (b) $\bar{x} = [1, \ldots, 1]$ hasta 6 posiciones decimales correctas

4.4 Ejercicios

1. (a) $x_1 = [0.5834, -0.0050, -0.5812]$, $x_2 = [1.0753, -0.1039, -0.9417]$, $x_3 = [1, 0, -1]$
   (b) $x_1 = [0.3896, 0.1674, 0.3045]$, $x_2 = [0.7650, 0.2107, 0.2502]$, $x_3 = [1/2, 1/2, 0]$
   (c) $x_1 = [0.0332, 0.8505, 0.9668]$, $x_2 = [0.0672, 0.8479, 0.9696]$, $x_3 = [0, 0, 1]$

4.5 Ejercicios

1. (a) $(x_1, y_1) = (2 - \sqrt{2}, 0)$ (b) $(x_1, y_1) = (1 - \sqrt{2}/2, 0)$
5. (a) $\begin{bmatrix} t_1^{c_2} & c_1 t_1^{c_2} \ln t_1 \\ t_2^{c_2} & c_1 t_2^{c_2} \ln t_2 \\ t_3^{c_2} & c_1 t_3^{c_2} \ln t_3 \end{bmatrix}$ (b) $\begin{bmatrix} t_1 e^{c_2 t_1} & c_1 t_1^2 e^{c_2 t_1} \\ t_2 e^{c_2 t_2} & c_1 t_2^2 e^{c_2 t_2} \\ t_3 e^{c_2 t_3} & c_1 t_3^2 e^{c_2 t_3} \end{bmatrix}$

4.5 Problemas de computadora

1. (a) $(\bar{x}, \bar{y}) = (0.410623, 0.055501)$ (b) $(\bar{x}, \bar{y}) = (0.275549, 0)$
3. (a) $(x, y) = (0, -0.586187)$, $K = 0.329572$ (b) $(x, y) = (0.556853, 0)$, $K = 1.288037$
5. $c_1 = 15.9, c_2 = 2.53$, RMEC $= 0$
7. Igual que el problema de computadora 5.
9. (a) $c_1 = 11.993468, c_2 = 0.279608, c_3 = 1.802342$, RMEC $= 0$
   (b) $c_1 = 12.702778, c_2 = 0.159591, c_3 = 5.682764$, RMEC $= 0$
11. (a) $c_1 = 8.670956, c_2 = 0.274184, c_3 = 0.981070, c_4 = 1.232813$, RMEC $= 0$
    (b) $c_1 = 8.683823, c_2 = 0.131945, c_3 = 0.620292, c_4 = -1.921257$, RMEC $= 0$

CAPÍTULO 5

5.1 Ejercicios

1. (a) 0.9531, error = 0.0469 (b) 0.9950, error = 0.0050 (c) 0.9995, error = 0.0005
3. (a) 0.455902, error = 0.044098; el error debe satisfacer $0.0433 \leq$ error ≤ 0.0456 (b) 0.495662, error = 0.004338; el error debe satisfacer $0.004330 \leq$ error ≤ 0.004355 (c) 0.499567, error = 0.000433; el error debe satisfacer $0.000433 \leq$ error ≤ 0.0004333
5. (a) 2.02020202, error = 0.02020202 (b) 2.00020002, error = 0.00020002 (c) 2.00000200, error = 0.00000200
7. $f'(x) = [(f(x) - f(x-h)]/h + hf''(c)/2$
9. $f'(x) = [3f(x) - 4f(x-h) + f(x-2h)]/(2h) + O(h^2)$
11. $f'(x) \approx [4f(x+h/2) - 3f(x) - f(x+h)]/h$
13. $f'(x) = [f(x+3h) + 8f(x) - 9f(x-h)]/(12h) - h^2 f'''(c)/2$, donde $x - h < c < x + 3h$
15. $f''(x) = [f(x+3h) - 4f(x) + 3f(x-h)]/(6h^2) - 2hf'''(c)/3$, donde $x - h < c < x + 3h$
17. $f'(x) = [4f(x+3h) + 5f(x) - 9f(x-2h)]/(30h) - h^2 f'''(c)$, donde $x - 2h < c < x + 3h$

5.1 Problemas de computadora

1. error mínimo en $h = 10^{-5} \approx \epsilon_{\text{máq}}^{1/3}$
3. error mínimo en $h = 10^{-8} \approx \epsilon_{\text{máq}}^{1/2}$
5. (a) error mínimo en $h = 10^{-4} \approx \epsilon_{\text{máq}}^{1/4}$ (b) igual que (a)

5.2 Ejercicios

1. (a) $m = 1 : 0.500000$, err = 0.166667; $m = 2 : 0.375000$, err = 0.041667; $m = 4 : 0.343750$, err = 0.010417
   (b) $m = 1 : 0.785398$, err = 0.214602; $m = 2 : 0.948059$, err = 0.051941; $m = 4 : 0.987116$, err = 0.012884
   (c) $m = 1 : 1.859141$, err = 0.140859; $m = 2 : 1.753931$, err = 0.035649; $m = 4 : 1.727222$, err = 0.008940
3. (a) $m = 1 : 1/3$, err = 0; $m = 2 : 1/3$, err = 0; $m = 4 : 1/3$, err = 0 (b) $m = 1 : 1.002280$, err = 0.002280; $m = 2 : 1.000135$, err = 0.000135; $m = 4 : 1.000008$, err = 0.000008 (c) $m = 1 : 1.718861$, err = 0.000579; $m = 2 : 1.718319$, err = 0.000037; $m = 4 : 1.718284$, err = 0.000002
5. (a) $m = 1 : 1.414214$, err = 0.585786; $m = 2 : 1.577350$, err = 0.422650; $m = 4 : 1.698844$; err = 0.301156
   (b) $m = 1 : 1.259921$, err = 0.240079; $m = 2 : 1.344022$, err = 0.155978; $m = 4 : 1.400461$, err = 0.099539
   (c) $m = 1 : 2.000000$, err = 0.828427; $m = 2 : 2.230710$, err = 0.597717; $m = 4 : 2.402528$, err = 0.425899
7. (a) 1.631729, err = 0.368271
   (b) 1.372055, err = 0.127945
   (c) 2.307614, err = 0.520814
11. (a) 1 (b) 1 (c) 3
13. $$\frac{4h}{3}\sum_{i=1}^{m}[2f(u_i) + 2f(v_i) - f(w_i)] + \frac{7(b-a)h^4}{90}f^{(iv)}(c)$$
15. 5

5.2 Problemas de computadora

1. (a) exacta = 2; $m = 16$ aprox = 1.998638, err = 1.36×10^{-3}; $m = 32$ aprox = 1.999660, err = 3.40×10^{-4} (b) exacta = $1/2(1 - \ln 2)$; $m = 16$ aprox = 0.153752, err = 3.26×10^{-4}; $m = 32$ aprox = 0.153508, err = 8.14×10^{-5} (c) exacta = 1; $m = 16$ aprox = 1.001444, err = 1.44×10^{-3}; $m = 32$ aprox = 1.000361, err = 3.61×10^{-4} (d) exacta = $9 \ln 3 - 26/9$; $m = 16$ aprox = 7.009809, err = 1.12×10^{-2}; $m = 32$ aprox = 7.001419, err = 2.80×10^{-3} (e) exacta = $\pi^2 - 4$; $m = 16$ aprox = 5.837900, err = 3.17×10^{-2}; $m = 32$ aprox = 5.861678, err = 7.93×10^{-3} (f) exacta = $2\sqrt{5} - \sqrt{15}/2$; $m = 16$ aprox = 2.535672, err = 2.80×10^{-5};

$m = 32$ aprox $= 2.535651$, err $= 7.00 \times 10^{-6}$ (g) exacta $= \ln(\sqrt{3} + 2)$; $m = 16$ aprox $= 1.316746$, err $= 2.11 \times 10^{-4}$; $m = 32$ aprox $= 1.316905$, err $= 5.29 \times 10^{-5}$ (h) exacta $= \ln(\sqrt{2} + 1)/2$; $m = 16$ aprox $= 0.440361$, err $= 3.26 \times 10^{-4}$; $m = 32$ aprox $= 0.440605$, err $= 8.14 \times 10^{-5}$

3. (a) $m = 16$ aprox $= 1.464420$; $m = 32$ aprox $= 1.463094$ (b) $m = 16$ aprox $= 0.891197$; $m = 32$ aprox $= 0.893925$ (c) $m = 16$ aprox $= 3.977463$; $m = 32$ aprox $= 3.977463$ (d) $m = 16$ aprox $= 0.264269$; $m = 32$ aprox $= 0.264025$ (e) $m = 16$ aprox $= 0.160686$; $m = 32$ aprox $= 0.160936$ (f) $m = 16$ aprox $= -0.278013$; $m = 32$ aprox $= -0.356790$ (g) $m = 16$ aprox $= 0.785276$; $m = 32$ aprox $= 0.783951$ (h) $m = 16$ aprox $= 0.369964$; $m = 32$ aprox $= 0.371168$

5. (a) $m = 10 : 1.808922$, err $= 0.191078$; $m = 100 : 1.939512$, err $= 0.060488$; $m = 1000 : 1.980871$, err $= 0.019129$
   (b) $m = 10 : 1.445632$, err $= 0.054368$; $m = 100 : 1.488258$, err $= 0.011742$; $m = 1000 : 1.497470$, err $= 0.002530$
   (c) $m = 10 : 2.558203$, err $= 0.270225$; $m = 100 : 2.742884$, err $= 0.085543$; $m = 1000 : 2.801375$, err $= 0.027052$

7. (a) $m = 16$ aprox $= 1.8315299$; $m = 32$ aprox $= 1.83183081$. (b) $m = 16$ aprox $= 2.99986658$; $m = 32$ aprox $= 3.00116293$ (c) $m = 16$ aprox $= 0.91601205$; $m = 32$ aprox $= 0.91597721$

5.3 Ejercicios

1. (a) 1/3 (b) 0.99999157 (c) 1.71828269

5.3 Problemas de computadora

1. (a) correcta $= 2$, aprox $= 2.00000010$, err $= 1.0 \times 10^{-7}$ (b) correcta $1/2(1 - \ln 2)$ aprox $= 0.15342640$, err $= 1.23 \times 10^{-8}$ (c) correcta 1, aprox $= 1.00000000$, err $= 3.5 \times 10^{-13}$ (d) correcta $9\ln 3 - 26/9$, aprox $= 6.99862171$, err $= 3.00 \times 10^{-9}$ (e) correcta $\pi^2 - 4$, aprox $= 5.86960486$, err $= 4.56 \times 10^{-7}$ (f) correcta $2\sqrt{5} - \sqrt{15}/2$, aprox $= 2.53564428$, err $= 1.21 \times 10^{-10}$ (g) correcta $\ln(\sqrt{3} + 2)$, aprox $= 1.31695765$, err $= 2.46 \times 10^{-7}$ (h) correcta $\ln(\sqrt{2} + 1)/2$, aprox $= 0.44068686$, err $= 6.98 \times 10^{-8}$

5.4 Ejercicios

1. (a) 0.3750, error $= 0.0417$ (b) 0.9871, error $= 0.0129$ (c) 1.7539, error $= 0.0356$

3. Use la misma tolerancia, prueba la cuadratura adaptativa con la regla del trapecio; reemplace la regla del trapecio por la regla del punto medio.

5.4 Problemas de computadora

1. (a) 2.00000000, 12606 subintervalos (b) 0.15342641, 6204 subintervalos (c) 1.00000000, 12424 subintervalos (d) 6.99862171, 32768 subintervalos (e) 5.86960440, 73322 subintervalos (f) 2.53564428, 1568 subintervalos (g) 1.31695790, 7146 subintervalos (h) 0.44068679, 5308 subintervalos

3. Los primeros ocho decimales son idénticos al problema de computadora 1 (a) 56 subintervalos (b) 46 subintervalos (c) 40 subintervalos (d) 56 subintervalos (e) 206 subintervalos (f) 22 subintervalos (g) 54 subintervalos (h) 52 subintervalos

5. Los primeros ocho decimales son idénticos al problema de computadora 1 (a) 50 subintervalos (b) 44 subintervalos (c) 36 subintervalos (d) 54 subintervalos (e) 198 subintervalos (f) 22 subintervalos (g) 50 subintervalos (h) 52 subintervalos

7. Igual que en el problema de computadora 6.

9. $\text{erf}(1) = 0.84270079$, $\text{erf}(3) = 0.99997791$

5.5 Ejercicios

1. (a) 0, error $= 0$ (b) 0.222222, error $= 0.1777778$ (c) 2.342696, error $= 0.007706$ (d) -0.481237, error $= 0.481237$

3. (a) 0, error $= 0$ (b) 0.4, error $= 0$ (c) 2.350402, error $= 2.95 \times 10^{-7}$ (d) -0.002136, error $= 0.002136$

5. (a) 1.999825 (b) 0.15340700 (c) 0.99999463 (d) 6.99867782

CAPÍTULO 6

6.1 Ejercicios

3. (a) $y(t) = 1 + t^2/2$ (b) $y(t) = e^{t^3/3}$ (c) $y(t) = e^{t^2+2t}$ (d) $y = e^{t^5}$ (e) $y(t) = (3t+1)^{1/3}$ (f) $y(t) = (3t^4/4 + 1)^{1/3}$
5. (a) $w = [1.0000, 1.0000, 1.0625, 1.1875, 1.3750]$, error $= 0.1250$
   (b) $w = [1.0000, 1.0000, 1.0156, 1.0791, 1.2309]$, error $= 0.1648$
   (c) $w = [1.0000, 1.5000, 2.4375, 4.2656, 7.9980]$, error $= 12.0875$
   (d) $w = [1.0000, 1.0000, 1.0049, 1.0834, 1.5119]$, error $= 1.2064$
   (e) $w = [1.0000, 1.2500, 1.4100, 1.5357, 1.6417]$, error $= 0.0543$
   (f) $w = [1.0000, 1.0000, 1.0039, 1.0349, 1.1334]$, error $= 0.0717$
7. (b) $c = \arctan y_0$
9. (a) $L = 0$, tiene solución única (b) $L = 1$, tiene solución única (c) $L = 1$, tiene solución única (d) Sin constante de Lipschitz
11. (a) Las soluciones son $Y(t) = t^2/2$ y $Z(t) = t^2/2 + 1$. $|Y(t) - Z(t)| = 1 \le e^0|1| = 1$ (b) Las soluciones son $Y(t) = 0$ y $Z(t) = e^t$. $|Y(t) - Z(t)| = e^t \le e^{1(t-0)}|1|$ (c) Las soluciones son $Y(t) = 0$ y $Z(t) = e^{-t}$. $|Y(t) - Z(t)| = e^{-t} \le e^{1(t-0)}|1| = 1$ (d) La condición de Lipschitz no se satisface
13. $y(t) = 1/(1-t)$
15. (a) $[a, b]$

6.1 Problemas de computadora

1. (a)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0050
0.2	1.0100	0.0100
0.3	1.0300	0.0150
0.4	1.0600	0.0200
0.5	1.1000	0.0250
0.6	1.1500	0.0300
0.7	1.2100	0.0350
0.8	1.2800	0.0400
0.9	1.3600	0.0450
1.0	1.4500	0.0500

(b)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0003
0.2	1.0010	0.0017
0.3	1.0050	0.0040
0.4	1.0140	0.0075
0.5	1.0303	0.0123
0.6	1.0560	0.0186
0.7	1.0940	0.0271
0.8	1.1477	0.0384
0.9	1.2211	0.0540
1.0	1.3200	0.0756

(c)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.2000	0.0337
0.2	1.4640	0.0887
0.3	1.8154	0.1784
0.4	2.2874	0.3243
0.5	2.9278	0.5625
0.6	3.8062	0.9527
0.7	5.0241	1.5952
0.8	6.7323	2.6610
0.9	9.1560	4.4431
1.0	12.6352	7.4503

(d)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0001	0.0003
0.3	1.0009	0.0016
0.4	1.0049	0.0054
0.5	1.0178	0.0140
0.6	1.0496	0.0313
0.7	1.1176	0.0654
0.8	1.2517	0.1360
0.9	1.5081	0.2968
1.0	2.0028	0.7154

(e)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.1000	0.0086
0.2	1.1826	0.0130
0.3	1.2541	0.0156
0.4	1.3177	0.0171
0.5	1.3753	0.0181
0.6	1.4282	0.0187
0.7	1.4772	0.0191
0.8	1.5230	0.0193
0.9	1.5661	0.0195
1.0	1.6069	0.0195

(f)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0001	0.0003
0.3	1.0009	0.0011
0.4	1.0036	0.0028
0.5	1.0099	0.0054
0.6	1.0222	0.0092
0.7	1.0429	0.0139
0.8	1.0744	0.0190
0.9	1.1188	0.0239
1.0	1.1770	0.0281

6.2 Ejercicios

1. (a) $w = [1.0000, 1.0313, 1.1250, 1.2813, 1.5000]$, error $= 0$ (b) $w = [1.0000, 1.0078, 1.0477, 1.1587, 1.4054]$, error $= 0.0097$ (c) $w = [1.0000, 1.7188, 3.3032, 7.0710, 16.7935]$, error $= 3.2920$
   (d) $w = [1.0000, 1.0024, 1.0442, 1.3077, 2.7068]$, error $= 0.0115$
   (e) $w = [1.0000, 1.2050, 1.3570, 1.4810, 1.5871]$, error $= 0.0003$
   (f) $w = [1.0000, 1.0020, 1.0193, 1.0823, 1.2182]$, error $= 0.0132$
3. (a) $w_{i+1} = w_i + ht_i w_i + 1/2h^2(w_i + t_i^2 w_i)$
   (b) $w_{i+1} = w_i + h(t_i w_i^2 + w_i^3) + 1/2h^2(w_i^2 + (2t_i w_i + 3w_i^2)(t_i w_i^2 + w_i^3))$
   (c) $w_{i+1} = w_i + hw_i \operatorname{sen} w_i + 1/2h^2(\operatorname{sen} w_i + w_i \cos w_i) w_i \operatorname{sen} w_i$
   (d) $w_{i+1} = w_i + he^{w_i t_i^2} + 1/2h^2 e^{w_i t_i^2}(2t_i w_i + t_i^2 e^{w_i t_i^2})$

6.2 Problemas de computadora

1. (a)

t_i	w_i	error
0.0	1.0000	0
0.1	1.0050	0
0.2	1.0200	0
0.3	1.0450	0
0.4	1.0800	0
0.5	1.1250	0
0.6	1.1800	0
0.7	1.2450	0
0.8	1.3200	0
0.9	1.4050	0
1.0	1.5000	0

(b)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0005	0.0002
0.2	1.0030	0.0003
0.3	1.0095	0.0005
0.4	1.0222	0.0007
0.5	1.0434	0.0008
0.6	1.0757	0.0010
0.7	1.1224	0.0012
0.8	1.1875	0.0014
0.9	1.2767	0.0016
1.0	1.3974	0.0018

(c)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.2320	0.0017
0.2	1.5479	0.0048
0.3	1.9832	0.0106
0.4	2.5908	0.0209
0.5	3.4509	0.0394
0.6	4.6864	0.0725
0.7	6.4878	0.1316
0.8	9.1556	0.2378
0.9	13.1694	0.4297
1.0	19.3063	0.7792

(d)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0005	0.0001
0.3	1.0029	0.0004
0.4	1.0114	0.0011
0.5	1.0338	0.0021
0.6	1.0845	0.0037
0.7	1.1890	0.0060
0.8	1.3967	0.0090
0.9	1.8158	0.0109
1.0	2.7164	0.0018

(e)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0913	0.0001
0.2	1.1695	0.0001
0.3	1.2384	0.0001
0.4	1.3005	0.0001
0.5	1.3571	0.0001
0.6	1.4093	0.0001
0.7	1.4580	0.0001
0.8	1.5036	0.0001
0.9	1.5466	0.0001
1.0	1.5873	0.0001

(f)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0001	0.0000
0.2	1.0005	0.0001
0.3	1.0022	0.0002
0.4	1.0068	0.0004
0.5	1.0160	0.0006
0.6	1.0323	0.0009
0.7	1.0579	0.0011
0.8	1.0948	0.0014
0.9	1.1443	0.0017
1.0	1.2069	0.0018

6.3 Ejercicios

1. (a) $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 & 1.25 & 1.5 & 1.7188 & 1.8594 \\ 0 & -0.25 & -0.625 & -1.1563 & -1.875 \end{bmatrix}$ error $= \begin{bmatrix} 0.3907 \\ 0.4124 \end{bmatrix}$

   (b) $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.7500 & 0.5000 & 0.2813 & 0.1094 \\ 0 & 0.2500 & 0.3750 & 0.4063 & 0.3750 \end{bmatrix}$ error $= \begin{bmatrix} 0.0894 \\ 0.0654 \end{bmatrix}$

   (c) $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 1 & 1.0000 & 0.9375 & 0.8125 & 0.6289 \\ 0 & 0.2500 & 0.5000 & 0.7344 & 0.9375 \end{bmatrix}$ error $= \begin{bmatrix} 0.0886 \\ 0.0960 \end{bmatrix}$

   (d) $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 5 & 6.2500 & 9.6875 & 17.2656 & 32.9492 \\ 0 & 2.5000 & 6.8750 & 15.1563 & 31.3672 \end{bmatrix}$ error $= \begin{bmatrix} 77.3507 \\ 77.0934 \end{bmatrix}$

1. (a) $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1.2500 & 1.4648 & 1.5869 & 1.5354 \\ 0 & -0.3125 & -0.7813 & -1.4343 & -2.2888 \end{bmatrix}$ error $= \begin{bmatrix} 0.0667 \\ 0.0015 \end{bmatrix}$

(b) $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.7500 & 0.5273 & 0.3428 & 0.1990 \\ 0 & 0.1875 & 0.2813 & 0.3098 & 0.2966 \end{bmatrix}$ error $= \begin{bmatrix} 0.0002 \\ 0.0129 \end{bmatrix}$

(c) $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.9688 & 0.8760 & 0.7275 & 0.5327 \\ 0 & 0.2500 & 0.4844 & 0.6882 & 0.8486 \end{bmatrix}$ error $= \begin{bmatrix} 0.0076 \\ 0.0071 \end{bmatrix}$

(d) $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 5 & 7.3438 & 14.3311 & 32.6805 & 79.2426 \\ 0 & 3.4375 & 11.2793 & 30.2963 & 77.3799 \end{bmatrix}$ error $= \begin{bmatrix} 31.0574 \\ 31.0806 \end{bmatrix}$

3. (a) $y_1 = [1.0000, 1.2500, 1.5195, 1.8364, 2.2388]$ (b) $[1, 1.1875, 1.2378, 1.1229, 0.7832]$
(c) $[1, 1.2813, 1.6617, 2.1999, 2.9933]$

6.3 Problemas de computadora

1. errores en $[y_1, y_2]$: (a) $[0.1973, 0.1592]$ para $h = 0.1$ $[0.0226, 0.0149]$ para $h = 0.01$ (b) $[0.0328, 0.0219]$ para $h = 0.1$, $[0.0031, 0.0020]$ para $h = 0.01$ (c) $[0.0305, 0.0410]$ para $h = 0.1$, $[0.0027, 0.0042]$ para $h = 0.01$ (d) $[51.4030, 51.3070]$ para $h = 0.1$, $[8.1919, 8.1827]$ para $h = 0.01$. Observe que, para un método de primer orden, los errores disminuyen aproximadamente por un factor de 10.
5. (a) Hablando en forma aproximada, la trayectoria periódica consiste en $3\frac{1}{2}$ revoluciones en sentido horario, $2\frac{1}{2}$ revoluciones en sentido antihorario, $3\frac{1}{2}$ revoluciones en sentido horario, $2\frac{1}{2}$ revoluciones en sentido antihorario. La otra trayectoria periódica es la misma con los sentidos horarios cambiados por sentidos antihorarios.

6.4 Ejercicios

1. (a) $w = [1.0000, 1.0313, 1.1250, 1.2813, 1.5000]$, error $= 0$ (b) $w = [1.0000, 1.0039, 1.0395, 1.1442, 1.3786]$, error $= 0.0171$ (c) $w = [1.0000, 1.7031, 3.2399, 6.8595, 16.1038]$, error $= 3.9817$
(d) $w = [1.0000, 1.0003, 1.0251, 1.2283, 2.3062]$, error $= 0.4121$
(e) $w = [1.0000, 1.1975, 1.3490, 1.4734, 1.5801]$, error $= 0.0073$
(f) $w = [1.0000, 1.0005, 1.0136, 1.0713, 1.2055]$, error $= 0.0004$
3. (a) $w = [1, 1.0313, 1.1250, 1.2813, 1.5000]$, error $= 0$ (b) $w = [1, 1.0052, 1.0425, 1.1510, 1.3956]$, error $= 1.2476 \times 10^{-5}$ (c) $w = [1, 1.7545, 3.4865, 7.8448, 19.975]$, error $= 0.11007$
(d) $w = [1, 1.001, 1.0318, 1.2678, 2.7103]$, error $= 7.9505 \times 10^{-3}$
(e) $w = [1, 1.2051, 1.3573, 1.4813, 1.5874]$, error $= 4.1996 \times 10^{-5}$
(f) $w = [1, 1.0010, 1.0154, 1.0736, 1.2051]$, error $= 6.0464 \times 10^{-5}$

6.4 Problemas de computadora

1. (a)

t_i	w_i	error
0.0	1.0000	0
0.1	1.0050	0
0.2	1.0200	0
0.3	1.0450	0
0.4	1.0800	0
0.5	1.1250	0
0.6	1.1800	0
0.7	1.2450	0
0.8	1.3200	0
0.9	1.4050	0
1.0	1.5000	0

(b)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0003	0.0001
0.2	1.0025	0.0002
0.3	1.0088	0.0003
0.4	1.0212	0.0004
0.5	1.0420	0.0005
0.6	1.0740	0.0007
0.7	1.1201	0.0010
0.8	1.1847	0.0014
0.9	1.2730	0.0020
1.0	1.3926	0.0030

(c)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.2310	0.0027
0.2	1.5453	0.0074
0.3	1.9780	0.0158
0.4	2.5814	0.0303
0.5	3.4348	0.0555
0.6	4.6594	0.0995
0.7	6.4430	0.1764
0.8	9.0814	0.3120
0.9	13.0463	0.5528
1.0	19.1011	0.9845

(d)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0003	0.0001
0.3	1.0022	0.0002
0.4	1.0097	0.0005
0.5	1.0306	0.0012
0.6	1.0785	0.0024
0.7	1.1778	0.0052
0.8	1.3754	0.0124
0.9	1.7711	0.0338
1.0	2.6107	0.1076

(e)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0907	0.0007
0.2	1.1686	0.0010
0.3	1.2375	0.0011
0.4	1.2995	0.0011
0.5	1.3561	0.0011
0.6	1.4083	0.0011
0.7	1.4570	0.0011
0.8	1.5026	0.0011
0.9	1.5456	0.0010
1.0	1.5864	0.0010

(f)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0003	0.0000
0.3	1.0019	0.0001
0.4	1.0062	0.0002
0.5	1.0151	0.0003
0.6	1.0311	0.0003
0.7	1.0564	0.0003
0.8	1.0931	0.0003
0.9	1.1426	0.0001
1.0	1.2051	0.0001

6.6 Ejercicios

1. (a) $w = [0, 0.0833, 0.2778, 0.6204, 1.1605]$, error $= 0.4422$
   (b) $w = [0, 0.0500, 0.1400, 0.2620, 0.4096]$, error $= 0.0417$
   (c) $w = [0, 0.1667, 0.4444, 0.7963, 1.1975]$, error $= 0.0622$

6.6 Problemas de computadora

1. (a) $y = 1$, tamaño de paso de Euler ≤ 1.8 (b) $y = 1$, tamaño de paso de Euler $\leq 1/3$

6.7 Ejercicios

1. (a) $w = [1.0000, 1.0313, 1.1250, 1.2813, 1.5000]$, error $= 0$
   (b) $w = [1.0000, 1.0078, 1.0314, 1.1203, 1.3243]$, error $= 0.0713$
   (c) $w = [1.0000, 1.7188, 3.0801, 6.0081, 12.7386]$, error $= 7.3469$
   (d) $w = [1.0000, 1.0024, 1.0098, 1.1257, 1.7540]$, error $= 0.9642$
   (e) $w = [1.0000, 1.2050, 1.3383, 1.4616, 1.5673]$, error $= 0.0201$
   (f) $w = [1.0000, 1.0020, 1.0078, 1.0520, 1.1796]$, error $= 0.0255$
3. $w_{i+1} = -4w_i + 5w_{i-1} + h[4f_i + 2f_{i-1}]$; No.
7. (a) $0 < a_1 < 2$ (b) $a_1 = 0$
9. (a) segundo orden inestable (b) segundo orden muy estable (c) tercer orden muy estable (d) tercer orden inestable (e) tercer orden inestable
11. Por ejemplo, $a_1 = 0, a_2 = 1, b_1 = 2 - 2b_0, b_2 = b_0$, donde $b_0 \neq 0$ es arbitraria.
13. (a) $a_1 + a_2 + a_3 = 1, -a_2 - 2a_3 + b_1 + b_2 + b_3 = 1, a_2 + 4a_3 - 2b_2 - 4b_3 = 1, -a_2 - 8a_3 + 3b_2 + 12b_3 = 1$ (c) $P(x) = x^3 - x^2$ tiene doble raíz en 0, una sola raíz en 1.
    (d) $w_{i+1} = w_{i-1} + h[\frac{7}{3}f_i - \frac{2}{3}f_{i-1} + \frac{1}{3}f_{i-2}]$
15. (a) $a_1 + a_2 + a_3 = 1, -a_2 - 2a_3 + b_0 + b_1 + b_2 + b_3 = 1, a_2 + 4a_3 + 2b_0 - 2b_2 - 4b_3 = 1,$ $-a_2 - 8a_3 + 3b_0 + 3b_2 + 12b_3 = 1, a_2 + 16a_3 + 4b_0 - 4b_2 - 32b_3 = 1$ (c) $P(x) = x^3 - x^2 = x^2(x - 1)$ tiene una sola raíz en 1.

6.7 Problemas de computadora

1. (a)

t_i	w_i	error
0.0	1.0000	0
0.1	1.0050	0
0.2	1.0200	0
0.3	1.0450	0
0.4	1.0800	0
0.5	1.1250	0
0.6	1.1800	0
0.7	1.2450	0
0.8	1.3200	0
0.9	1.4050	0
1.0	1.5000	0

(b)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0005	0.0002
0.2	1.0020	0.0007
0.3	1.0075	0.0015
0.4	1.0191	0.0025
0.5	1.0390	0.0035
0.6	1.0698	0.0048
0.7	1.1146	0.0065
0.8	1.1773	0.0088
0.9	1.2630	0.0121
1.0	1.3788	0.0168

(c)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.2320	0.0017
0.2	1.5386	0.0141
0.3	1.9569	0.0368
0.4	2.5355	0.0762
0.5	3.3460	0.1443
0.6	4.4967	0.2621
0.7	6.1533	0.4661
0.8	8.5720	0.8214
0.9	12.1548	1.4443
1.0	17.5400	2.5455

(d)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0000	0.0000
0.2	1.0001	0.0002
0.3	1.0013	0.0012
0.4	1.0070	0.0033
0.5	1.0243	0.0075
0.6	1.0658	0.0150
0.7	1.1534	0.0296
0.8	1.3266	0.0611
0.9	1.6649	0.1400
1.0	2.3483	0.3700

(e)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0913	0.0001
0.2	1.1673	0.0023
0.3	1.2354	0.0032
0.4	1.2970	0.0036
0.5	1.3534	0.0038
0.6	1.4055	0.0039
0.7	1.4542	0.0039
0.8	1.4998	0.0039
0.9	1.5428	0.0038
1.0	1.5836	0.0038

(f)

t_i	w_i	error
0.0	1.0000	0.0000
0.1	1.0001	0.0000
0.2	1.0002	0.0002
0.3	1.0013	0.0007
0.4	1.0050	0.0014
0.5	1.0131	0.0022
0.6	1.0282	0.0032
0.7	1.0528	0.0039
0.8	1.0890	0.0044
0.9	1.1383	0.0044
1.0	1.2011	0.0040

3. (a)

t_i	w_i	error
0.0	0.0000	0.0000
0.1	0.0050	0.0002
0.2	0.0213	0.0002
0.3	0.0493	0.0005
0.4	0.0916	0.0002
0.5	0.1474	0.0013
0.6	0.2222	0.0001
0.7	0.3105	0.0032
0.8	0.4276	0.0020
0.9	0.5510	0.0086
1.0	0.7283	0.0100

(b)

t_i	w_i	error
0.0	0.0000	0.0000
0.1	0.0050	0.0002
0.2	0.0187	0.0000
0.3	0.0413	0.0005
0.4	0.0699	0.0004
0.5	0.1082	0.0016
0.6	0.1462	0.0027
0.7	0.2032	0.0066
0.8	0.2360	0.0134
0.9	0.3363	0.0297
1.0	0.3048	0.0631

(c)

t_i	w_i	error
0.0	0.0000	0.0000
0.1	0.0200	0.0013
0.2	0.0700	0.0003
0.3	0.1530	0.0042
0.4	0.2435	0.0058
0.5	0.3855	0.0176
0.6	0.4645	0.0367
0.7	0.7356	0.0890
0.8	0.5990	0.2029
0.9	1.4392	0.4739
1.0	0.0394	1.0959

CAPÍTULO 7

7.1 Ejercicios

3. (a) $\operatorname{sen} 2t, \cos 2t$ (b) $y_a - y_b = 0$ (c) $y_a + y_b = 0$ (d) sin condición, la solución siempre existe

5. $y(t) = \dfrac{y_1 - e^{-\sqrt{k}}y_0}{e^{\sqrt{k}} - e^{-\sqrt{k}}}e^{\sqrt{k}t} + \dfrac{e^{\sqrt{k}}y_0 - y_1}{e^{\sqrt{k}} - e^{-\sqrt{k}}}e^{-\sqrt{k}t}$

7.1 Problemas de computadora

1. (a) $y(t) = 1/3te^t$ (b) $y(t) = e^{t^2}$
3. (a) $y(t) = 1/(3t^2)$ (b) $y(t) = \ln(t^2 + 1)$
5. (a) $s = y_2(0) = 1$, la solución exacta es $y_1(t) = \arctan t$, $y_2 = t^2 + 1$ (b) $s = y_2(0) = 1/3$, la solución exacta es $y_1(t) = e^{t^3}$, $y_2(t) = 1/3 - t^2$

7.2 Problemas de computadora

5. (a) $y(t) = \dfrac{e^{1+t} - e^{1-t}}{e^2 - 1}$

(c)

n	h	error
3	1/4	0.00026473
7	1/8	0.00006657
15	1/16	0.00001667
31	1/32	0.00000417
63	1/64	0.00000104
127	1/128	0.00000026

7. Extrapole mediante $N_2(h) = (4N(h/2) - N(h))/3$ y $N_3(h) = (16N_2(h/2) - N_2(h))/15$ para llegar a la estimación $y(1/2) \approx 0.443409442296$, error $\approx 3.11 \times 10^{-10}$.
11. 11.786

CAPÍTULO 8

8.1 Problemas de computadora

1. Solución aproximada en puntos representativos:

(a)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$t = 0.2$	3.0432	3.3640	3.9901
$t = 0.5$	5.5451	6.1296	7.2705
$t = 0.8$	10.1039	11.1688	13.2477

(b)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$t = 0.2$	1.8219	2.4593	3.3199
$t = 0.5$	3.3198	4.4811	6.0492
$t = 0.8$	6.0490	8.1651	11.0224

El método de las diferencias hacia adelante es inestable en ambas partes para $h = 0.1$, $K > 0.003$.

3.

(a)

h	k	$u(0.5, 1)$	$w(0.5, 1)$	error
0.02	0.02	16.6642	16.7023	0.0381
0.02	0.01	16.6642	16.6834	0.0192
0.02	0.005	16.6642	16.6738	0.0097

(b)

h	k	$u(0.5, 1)$	$w(0.5, 1)$	error
0.02	0.02	12.1825	12.2104	0.0279
0.02	0.01	12.1825	12.1965	0.0140
0.02	0.005	12.1825	12.1896	0.0071

5.

(a)

h	k	$u(0.5, 1)$	$w(0.5, 1)$	error
0.02	0.02	16.664183	16.664504	0.000321
0.01	0.01	16.664183	16.664263	0.000080
0.005	0.005	16.664183	16.664203	0.000020

(b)

h	k	$u(0.5, 1)$	$w(0.5, 1)$	error
0.02	0.02	12.182494	12.182728	0.000235
0.01	0.01	12.182494	12.182553	0.000059
0.005	0.005	12.182494	12.182509	0.000015

7. $C = \pi^2/100$

8.2 Problemas de computadora

1. Solución aproximada en puntos representativos:

(a)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$t = 0.2$	−0.4755	−0.8090	−0.4755
$t = 0.5$	0.5878	1.0000	0.5878
$t = 0.8$	−0.4755	−0.8090	−0.4755

(b)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$t = 0.2$	0.5489	0.4067	0.3012
$t = 0.5$	0.3012	0.2231	0.1653
$t = 0.8$	0.1652	0.1224	0.0907

(c)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$t = 0.2$	0.3364	0.5306	0.6931
$t = 0.5$	0.5306	0.6930	0.8329
$t = 0.8$	0.6931	0.8329	0.9554

3.

(a)

h	k	$w(1/4, 3/4)$	error
2^{-4}	2^{-6}	−0.70710678	0.0
2^{-5}	2^{-7}	−0.70710678	0.0
2^{-6}	2^{-8}	−0.70710678	0.0
2^{-7}	2^{-9}	−0.70710678	0.0
2^{-8}	2^{-10}	−0.70710678	0.0

(b)

h	k	$w(1/4, 3/4)$	error
2^{-4}	2^{-5}	0.17367424	0.00009971
2^{-5}	2^{-6}	0.17374901	0.00002493
2^{-6}	2^{-7}	0.17376771	0.00000623
2^{-7}	2^{-8}	0.17377238	0.00000156
2^{-8}	2^{-9}	0.17377355	0.00000039

(c)

h	k	$w(1/4, 3/4)$	error
2^{-4}	2^{-4}	0.69308400	0.00006318
2^{-5}	2^{-5}	0.69313136	0.00001582
2^{-6}	2^{-6}	0.69314323	0.00000396
2^{-7}	2^{-7}	0.69314619	0.00000099
2^{-8}	2^{-8}	0.69314693	0.00000025

8.3 Problemas de computadora

1. Solución aproximada en puntos representativos:

(a)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$y = 0.2$	0.3151	0.5362	0.3151
$y = 0.5$	0.1236	0.2103	0.1236
$y = 0.8$	0.0482	0.0821	0.0482

(b)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$y = 0.2$	0.4006	1.3686	3.6222
$y = 0.5$	0.6816	2.3284	6.1624
$y = 0.8$	0.4006	1.3686	3.6222

3. Solución aproximada en puntos representativos:

(a)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$y = 0.2$	0.0347	0.0590	0.0347
$y = 0.5$	0.1185	0.2016	0.1185
$y = 0.8$	0.3136	0.5336	0.3136

(b)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$y = 0.2$	0.4579	0.6752	0.8417
$y = 0.5$	0.6752	0.6708	0.6752
$y = 0.8$	0.8417	0.6752	0.4579

5. 11.4 metros

7.

(a)

h	k	$w(1/4, 3/4)$	error
2^{-2}	2^{-2}	0.072692	0.005672
2^{-3}	2^{-3}	0.068477	0.001457
2^{-4}	2^{-4}	0.067387	0.000367
2^{-5}	2^{-5}	0.067112	0.000092

(b)

h	k	$w(1/4, 3/4)$	error
2^{-2}	2^{-2}	0.673903	0.059660
2^{-3}	2^{-3}	0.629543	0.015300
2^{-4}	2^{-4}	0.618094	0.003851
2^{-5}	2^{-5}	0.615207	0.000964

11. Solución aproximada en puntos representativos:

(a)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$y = 0.2$	0.0631	0.1571	0.2493
$y = 0.5$	0.1571	0.3839	0.5887
$y = 0.8$	0.2493	0.5887	0.8448

(b)

	$x = 0.2$	$x = 0.5$	$x = 0.8$
$y = 0.2$	1.0405	1.1046	1.1731
$y = 0.5$	1.1046	1.2830	1.4910
$y = 0.8$	1.1731	1.4910	1.8956

13. Solución aproximada en puntos representativos:

(a)

	$x = 1.25$	$x = 1.50$	$x = 1.75$
$y = 1.25$	3.1250	3.8125	4.6250
$y = 1.50$	3.8125	4.5000	5.3125
$y = 1.75$	4.6250	5.3125	6.1250

(b)

	$x = 1.25$	$x = 1.50$	$x = 1.75$
$y = 0.50$	0.1999	0.1666	0.1428
$y = 1.00$	0.7999	0.6666	0.5714
$y = 1.50$	1.7999	1.4999	1.2857

15.

(a)

h	k	$w(1/4, 3/4)$	error
2^{-2}	2^{-2}	0.294813	0.004528
2^{-3}	2^{-3}	0.291504	0.001219
2^{-4}	2^{-4}	0.290596	0.000311
2^{-5}	2^{-5}	0.290363	0.000078

(b)

h	k	$w(1/4, 3/4)$	error
2^{-2}	2^{-2}	1.202628	0.003602
2^{-3}	2^{-3}	1.205310	0.000920
2^{-4}	2^{-4}	1.205999	0.000231
2^{-5}	2^{-5}	1.206172	0.000058

8.4 Problemas de computadora

1. La solución se aproxima a $u = 0$.
3. (a) La solución se aproxima a $u = 0$. (b) La solución se aproxima a $u = 2$.

CAPÍTULO 9

9.1 Ejercicios

1. (a) 4 (b) 9
3. (a) 0.3 (b) 0.28

9.1 Problemas de computadora

1. 0.000273, comparado con el volumen correcto ≈ 0.000268.
2. (En las siguientes respuestas se usó el GCL mínimo estándar con semilla 1:)

(a) 1/3 (b)

n	Estimación tipo 1	error
10^2	0.327290	0.006043
10^3	0.342494	0.009161
10^4	0.332705	0.000628
10^5	0.333610	0.000277
10^6	0.333505	0.000172

(c)

n	Estimación tipo 2	error
10^2	0.28	0.053333
10^3	0.354	0.020667
10^4	0.3406	0.007267
10^5	0.33382	0.000487
10^6	0.333989	0.000656

5. (a) $n = 10^4$: 0.5128, error = 0.010799; $n = 10^6$: 0.524980, error = 0.001381 (b) $n = 10^4$: 0.1744, error = 0.000133; $n = 10^6$: 0.174851, error = 0.000318
7. (a) 1/12 (b) 0.083566, error = 0.000232

9.2 Problemas de computadora

1. (a) 1/3 (b)

n	Estimación tipo 1	error
10^2	0.335414	0.002080
10^3	0.333514	0.000181
10^4	0.333339	0.000006
10^5	0.333334	0.000001

(c)

n	Estimación tipo 2	error
10^2	0.35	0.016667
10^3	0.333	0.000333
10^4	0.3339	0.000567
10^5	0.33338	0.000047

3. (a) $n = 10^4$: 0.5232, error = 0.000399; $n = 10^5$: 0.52396, error = 0.000361 (b) $n = 10^4$: 0.1743, error = 0.000233; $n = 10^5$: 0.17455, error = 0.000017
5. Resultados típicos: estimación de Monte Carlo 4.9656, error = 0.030798; estimación cuasi-Monte Carlo 4.92928, error = 0.005522.
7. (a) valor exacto = 1/2; $n = 10^6$, estimación de Monte Carlo 0.500313 (b) valor exacto = 4/9; $n = 10^6$, estimación de Monte Carlo 0.444486
9. $1/24 \approx 4.167\%$

9.3 Problemas de computadora

Las respuestas de esta sección usan el GCL mínimo estándar.

1. (a) Monte Carlo = 0.2907, error = 0.0050 (b) 0.6323, error 0.0073. (c) 0.7322, error 0.0049.
3. (a) 0.8199, error = 0.0014 (b) 0.9871, error = 0.0004 (c) 0.9984, error = 0.0006
5. (a) 0.2969, error = 0.0112 (b) 0.3939, error = 0.0049 (c) 0.4600, error = 0.0106
7. (a) 0.5848, error = 0.0207 (b) 0.3106, error = 0.0154 (c) 0.7155, error = 0.0107

9.4 Problemas de computadora

5. Resultados típicos

Δt	error promedio
10^{-1}	0.2657
10^{-2}	0.0925
10^{-3}	0.0256

Los resultados muestran un orden aproximado de 1/2.

11.

Δt	error promedio
10^{-1}	0.1394
10^{-2}	0.0202
10^{-3}	0.0026

Los resultados muestran un aproximado de primer orden.

CAPÍTULO 10

10.1 Ejercicios

1. (a) $[0, -i, 0, i]$ (b) $[2, 0, 0, 0]$ (c) $[0, i, 0, -i]$ (d) $[0, 0, -\sqrt{2}i, 0, 0, 0, \sqrt{2}i, 0]$
3. (a) $[1/2, 1/2, 1/2, 1/2]$ (b) $[1, 1, -1, 1]$ (c) $[1, 1, 1, -1]$ (d) $[2, -1, 2, -1, 2, -1, 2, -1]/\sqrt{2}$
5. (a) Las cuatro raíces unitarias: $-i, -1, i, 1$; primitivas: $-i, i$ (b) $\omega, \omega^2, \omega^3, \omega^4, \omega^5, \omega^6$ donde $\omega = e^{-2\pi i/7}$ (c) $p - 1$
7. (a) $a_0 = a_1 = a_2 = 0, b_1 = -1$ (b) $a_0 = 2, a_1 = a_2 = 0, b_1 = 0$ (c) $a_0 = a_1 = a_2 = 0, b_1 = 1$
   (d) $b_2 = -\sqrt{2}, a_0 = a_1 = a_2 = a_3 = a_4 = b_1 = b_3 = 0$

10.2 Ejercicios

1. (a) $P_4(t) = \operatorname{sen} 2\pi t$ (b) $P_4(t) = \cos 2\pi t + \operatorname{sen} 2\pi t$ (c) $P_4(t) = -\cos 4\pi t$ (d) $P_4(t) = 1$
3. (a) $P_8(t) = \operatorname{sen} 4\pi t$ (b) $P_8(t) = 1 + \operatorname{sen} 4\pi t$ (c) $P_8(t) = \frac{1}{2} + \frac{1}{4}\cos 2\pi t + \frac{\sqrt{2}+1}{4}\operatorname{sen} 2\pi t + \frac{1}{4}\cos 6\pi t + \frac{\sqrt{2}-1}{4}\operatorname{sen} 6\pi t$ (d) $P_8(t) = \cos 8\pi t$

10.2 Problemas de computadora

1. (a) $P_8(t) = \frac{7}{2} - \cos 2\pi t - (1+\sqrt{2})\operatorname{sen} 2\pi t - \cos 4\pi t - \operatorname{sen} 4\pi t - \cos 6\pi t + (1-\sqrt{2})\operatorname{sen} 6\pi t - \frac{1}{2}\cos 8\pi t$ (b) $P_8(t) = \frac{1}{2} - 0.8107\cos 2\pi t - 0.1036\operatorname{sen} 2\pi t + \cos 4\pi t + \frac{1}{2}\operatorname{sen} 4\pi t + 1.3107\cos 6\pi t - 0.6036\operatorname{sen} 6\pi t$ (c) $P_8(t) = \frac{5}{2} - \frac{1}{2}\cos\frac{\pi}{2}t - \frac{1}{2}\operatorname{sen}\frac{\pi}{2}t + \cos\pi t$ (d) $P_8(t) = \frac{5}{8} + \frac{3}{4}\cos\frac{\pi}{4}(t-1) + 1.3536\operatorname{sen}\frac{\pi}{4}(t-1) - \frac{7}{4}\cos\frac{\pi}{2}(t-1) - \frac{5}{2}\operatorname{sen}\frac{\pi}{2}(t-1) + \frac{3}{4}\cos\frac{3\pi}{4}(t-1) - 0.6464\operatorname{sen}\frac{3\pi}{4}(t-1) + \frac{5}{8}\cos\pi(t-1)$
3. $P_8(t) = 1.6131 - 0.1253\cos 2\pi t - 0.5050\operatorname{sen} 2\pi t - 0.1881\cos 4\pi t - 0.2131\operatorname{sen} 4\pi t - 0.1991\cos 6\pi t - 0.0886\operatorname{sen} 6\pi t - 0.1007\cos 8\pi t$
5. $P_8(t) = 0.3423 - 0.1115\cos 2\pi(t-1) - 0.2040\operatorname{sen} 2\pi(t-1) - 0.0943\cos 4\pi(t-1) - 0.0859\operatorname{sen} 4\pi(t-1) - 0.0912\cos 6\pi(t-1) - 0.0357\operatorname{sen} 6\pi(t-1) - 0.0453\cos 8\pi(t-1)$

10.3 Ejercicios

1. (a) $F_2(t) = 0$ (b) $F_2(t) = \cos 2\pi t$ (c) $F_2(t) = 0$ (d) $F_2(t) = 1$
3. (a) $F_4(t) = 0$ (b) $F_4(t) = 1$ (c) $F_4(t) = \frac{1}{2} + \frac{1}{4}\cos 2\pi t + \frac{\sqrt{2}+1}{4}\operatorname{sen} 2\pi t$ (d) $F_4(t) = 0$

10.3 Problemas de computadora

1. (a) $F_2(t) = F_4(t) = 3\cos 2\pi t$
   (b) $F_2(t) = 2 - \frac{3}{2}\cos 2\pi t$, $F_4(t) = 2 - \frac{3}{2}\cos 2\pi t - \frac{1}{2}\operatorname{sen} 2\pi t + \frac{3}{2}\cos 4\pi t$
   (c) $F_2(t) = \frac{7}{2} - \frac{1}{2}\cos\frac{\pi}{2}t$, $F_4(t) = \frac{7}{2} - \frac{1}{2}\cos\frac{\pi}{2}t + \frac{1}{2}\operatorname{sen}\frac{\pi}{2}t + 2\cos\pi t$
   (d) $F_2(t) = 2 - 2\cos\frac{\pi}{3}(t-1)$, $F_4(t) = 2 - 2\cos\frac{\pi}{3}(t-1) - \cos\frac{2\pi}{3}(t-1)$

CAPÍTULO 11

11.1 Ejercicios

1. La matriz de la TDC es $C = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, y $P_2(t) = \frac{1}{\sqrt{2}}y_0 + y_1\cos\frac{(2t+1)\pi}{4}$

   (a) $y = [3\sqrt{2}, 0]$, $P_2(t) = 3$ (b) $y = [0, 2\sqrt{2}]$, $P_2(t) = 2\sqrt{2}\cos\frac{(2t+1)\pi}{4}$
   (c) $y = [2\sqrt{2}, \sqrt{2}]$, $P_2(t) = 2 + \sqrt{2}\cos\frac{(2t+1)\pi}{4}$ (d) $y = [3\sqrt{2}/2, 5\sqrt{2}/2]$, $P_2(t) = 3/2 + (5\sqrt{2}/2)\cos\frac{(2t+1)\pi}{4}$.
3. (a) $y = [1, b-c, 0, b+c]$, $P_4(t) = \frac{1}{2} + \left((b-c)/\sqrt{2}\right)\cos\frac{(2t+1)\pi}{8} + \left((b+c)/\sqrt{2}\right)\cos\frac{3(2t+1)\pi}{8}$
   (b) $y = [2, 0, 0, 0]$, $P_4(t) = 1$ (c) $y = [1/2, b, 1/2, c]$, $P_4(t) = 1/2 + \left(b/\sqrt{2}\right)\cos\frac{(2t+1)\pi}{8} + (1/2\sqrt{2})\cos\frac{2(2t+1)\pi}{8} + \left(c/\sqrt{2}\right)\cos\frac{3(2t+1)\pi}{8}$ (d) $y = [5, -(c+3b), 0, (b-3c)]$, $P_4(t) = \frac{5}{2} - \left((c+3b)/\sqrt{2}\right)\cos\frac{(2t+1)\pi}{8} + \left((b-3c)/\sqrt{2}\right)\cos\frac{3(2t+1)\pi}{8}$

11.2 Ejercicios

1. (a) $Y = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$, $P_2(s,t) = \frac{1}{4} + \frac{1}{2\sqrt{2}}\cos\frac{(2s+1)\pi}{4} + \frac{1}{2\sqrt{2}}\cos\frac{(2t+1)\pi}{4} +$
$\frac{1}{2}\cos\frac{(2s+1)\pi}{4}\cos\frac{(2t+1)\pi}{4}$ (b) $Y = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$, $P_2(s,t) = \frac{1}{2} + \frac{1}{\sqrt{2}}\cos\frac{(2t+1)\pi}{4}$
(c) $Y = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$, $P_2(s,t) = 1$. (d) $Y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $P_2(s,t) = \frac{1}{2} + \cos\frac{(2s+1)\pi}{4}\cos\frac{(2t+1)\pi}{4}$

3. (a) $P(t) = \left((b+c)/\sqrt{2}\right)\cos\frac{(2t+1)\pi}{8}$ (b) $P(t) = 1/4$ (c) $P(t) = 1/4$
(d) $P(t) = 2 + \sqrt{2}(b-c)\cos\frac{(2s+1)\pi}{8}$

11.2 Problemas de computadora

1. (a) $\begin{bmatrix} 0 & -3.8268 & 0 & -9.2388 \\ 0 & 1.7071 & 0 & 4.1213 \\ 0 & 0 & 0 & 0 \\ 0 & 0.1213 & 0 & 0.2929 \end{bmatrix}$ (b) $\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2.1213 & -0.7654 & -0.8787 \\ 0 & 0 & 0 & 0 \\ 0 & 5.1213 & -1.8478 & -2.1213 \end{bmatrix}$
(c) $\begin{bmatrix} 4.7500 & 1.4419 & 0.2500 & 0.2146 \\ -0.7886 & 0.5732 & -1.4419 & -1.0910 \\ 0.2500 & 2.6363 & -2.2500 & -0.8214 \\ 0.0560 & -2.0910 & -0.2146 & 0.9268 \end{bmatrix}$ (d) $\begin{bmatrix} 0 & -4.4609 & 0 & -0.3170 \\ -4.4609 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.3170 & 0 & 0 & 0 \end{bmatrix}$

11.3 Ejercicios

1. (a) $P(A) = 1/4, P(B) = 5/8, P(C) = 1/8$, 1.30 (b) $P(A) = 3/8, P(B) = 1/4, P(C) = 3/8$, 1.56
(c) $P(A) = 1/2, P(B) = 3/8, P(C) = 1/8$, 1.41

3. (a) Se requieren 34 bits, 34/11 = 3.09 bits/símbolo > 3.03 = inf. de Shannon. (b) Se requieren 73 bits, 73/21 = 3.48 bits/símbolo > 3.42 = inf. de Shannon. (c) Se requieren 108 bits, 108/35 = 3.09 bits/símbolo > 3.04 = inf. de Shannon.

11.4 Ejercicios

1. (a) $[-12b - 2c, 2b - 12c]$ (b) $[-3b - c, b - 3c]$ (c) $[-8b + 5c, -5b - 8c]$
3. (a) +101., error = 0 (b) +101., error = 1/15 (c) +011., error = 1/35
5. (a) +0110000., error = 1/170 (b) −0101101., error = 1/85 (c) +1011100., error = 7/510
(d) +1100100., error ≈ 0.0043
7. (a) $\frac{1}{2}(w_2 + w_3) = [-1.2246, 0.9184] \approx [-1, 1]$ (b) $\frac{1}{2}(w_2 + w_3) = [2.1539, -0.9293] \approx [2, -1]$
(c) $\frac{1}{2}(w_2 + w_3) = [-1.7844, -3.0832] \approx [-2, -3]$
9. $c_{5n} = -c_{n-1}, c_{6n} = -c_0$

CAPÍTULO 12

12.1 Ejercicios

1. (a) $P(\lambda) = (\lambda - 5)(\lambda - 2)$, 2 y [1, 1], 5 y [1, −1] (b) $P(\lambda) = (\lambda + 2)(\lambda - 2)$, −2 y [1, −1], 2 y [1, 1]
(c) $P(\lambda) = (\lambda - 3)(\lambda + 2)$, 3 y [−3, 4], −2 y [4, 3] (d) $P(\lambda) = (\lambda - 100)(\lambda - 200)$, 200 y [−3, 4], 100 y [4, 3]

3. (a) $P(\lambda) = -(\lambda-1)(\lambda-2)(\lambda-3)$, 3 y $[0, 1, 0]$, 2 y $[1, 2, 1]$, 1 y $[1, 0, 0]$
   (b) $P(\lambda) = -\lambda(\lambda-1)(\lambda-2)$, 2 y $[-1, 2, 3]$, 1 y $[1, 1, 0]$, 0 y $[1, -2, 3]$
   (c) $P(\lambda) = -\lambda(\lambda-1)(\lambda+1)$, 1 y $[1, -2, -3]$, 0 y $[1, -2, 3]$, -1 y $[1, 1, 0]$
5. (a) $\lambda = 4, S = 3/4$ (b) $\lambda = -4, S = 3/4$ (c) $\lambda = 4, S = 1/2$ (d) $\lambda = 10, S = 9/10$
7. (a) $\lambda = 1, S = 1/3$ (b) $\lambda = 1, S = 1/3$ (c) $\lambda = -1, S = 1/2$ (d) $\lambda = 9, S = 3/4$
9. (a) 5 y $[1, 2]$, -1 y $[-1, 1]$ (b) $u_1 = \left[1/\sqrt{17}, 4/\sqrt{17}\right]$, RQ $= 1$; $u_2 = [0.4903, 0.8716]$, RQ $= 4.29$; $u_3 = [0.4386, 0.8987]$, CR $= 5.08$ (c) la IPI converge a $1 = -1$. (d) La IPI converge a $\lambda = 5$.
11. (a) 7 (b) 5 (c) $S = 6/7$, $S = 1/2$; la IPI con $s = 4$ es más rápida.

12.1 Problemas de computadora

1. (a) converge a 4 y $[1, 1, -1]$ (b) converge a -4 y $[1, 1, -1]$ (c) converge a 4 y $[1, 1, -1]$, (d) converge a 10 y $[1, 1, -1]$
   (d) converge a 10 y $[1, 1, -1]$
3. (a) $\lambda = 4$ (b) $\lambda = 3$ (c) $\lambda = 2$ (d) $\lambda = 9$

12.2 Ejercicios

1. (a) $\begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\sqrt{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ (c) $\begin{bmatrix} 2 & -\frac{4}{5} & -\frac{3}{5} \\ -5 & \frac{37}{25} & -\frac{16}{25} \\ 0 & \frac{9}{25} & \frac{13}{25} \end{bmatrix}$
   (d) $\begin{bmatrix} 1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\sqrt{8} & \frac{5}{2} & \frac{3}{2} \\ 0 & \frac{3}{2} & \frac{1}{2} \end{bmatrix}$
5. (a) La ISN falla: $\overline{Q}_k$ no converge, alterna con periodo de 2. (b) La ISN falla: $\overline{Q}_k$ no converge, alterna con periodo de 2.
7. (a) antes: no converge; después: igual (ya está en la forma de Hessenberg) (b) antes: no converge; después: no converge

12.2 Problemas de computadora

1. (a) $\{-6, 4, -2\}$ (b) $\{6, 4, 2\}$ (c) $\{20, 18, 16\}$ (d) $\{10, 2, 1\}$
3. (a) $\{3, 3, 3\}$ (b) $\{1, 9, 10\}$ (c) $\{3, 3, 18\}$ (d) $\{-2, 2, 0\}$
5. (a) $\{2, i, -i\}$ (b) $\{1, i, -i\}$ (c) $\{2+3i, 2-3i, 1\}$ (d) $\{5, 4+3i, 4-3i\}$

12.3 Ejercicios

1. (a) $\begin{bmatrix} -3 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

   Se expande por un factor de 3 y se invierte a lo largo del eje x, se expande por un factor de 2 a lo largo del eje y.

   (b) $\begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

   Se proyecta sobre el eje y y se expande por 3 en la dirección y.

   (c) $\begin{bmatrix} \frac{3}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

Se expande en una elipse con un eje mayor de longitud 4 a lo largo de la línea $y = -x$.

(d) $\begin{bmatrix} -\frac{3}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$ igual que (c) pero con una rotación de 180°.

(e) $\begin{bmatrix} \frac{3}{4} & \frac{5}{4} \\ \frac{5}{4} & \frac{3}{4} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

Se expande por un factor de 2 a lo largo de la línea $y = x$, se contrae por un factor de 2 a lo largo de la línea $y = -x$, y los puntos sobre el círculo se invierten.

3. Cuatro: $\begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$

$= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

$= \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$

12.4 Problemas de computadora

1. (a) $\begin{bmatrix} 1.1708 & 1.8944 \\ 1.8944 & 3.0652 \end{bmatrix}$ (b) $\begin{bmatrix} 1.5607 & 3.7678 \\ 1.3536 & 3.2678 \end{bmatrix}$ (c) $\begin{bmatrix} 1.0107 & 2.5125 & 3.6436 \\ 0.9552 & 2.3746 & 3.4436 \\ 0.1787 & 0.4442 & 0.6441 \end{bmatrix}$

(d) $\begin{bmatrix} -0.5141 & 5.2343 & 1.9952 \\ 0.2070 & -2.1076 & -0.8033 \\ -0.1425 & 1.4510 & 0.5531 \end{bmatrix}$

3. (a) La mejor recta es $y = 3.3028x$; las proyecciones son $\begin{bmatrix} 1.1934 \\ 3.9415 \end{bmatrix}, \begin{bmatrix} 1.4707 \\ 4.8575 \end{bmatrix}, \begin{bmatrix} 1.2774 \\ 4.2188 \end{bmatrix}$.

(b) La mejor recta es $y = 0.3620x$; las proyecciones son $\begin{bmatrix} 1.1934 \\ 3.9415 \end{bmatrix}, \begin{bmatrix} 1.4707 \\ 4.8575 \end{bmatrix}, \begin{bmatrix} 1.2774 \\ 4.2188 \end{bmatrix}$.

(c) La mejor recta es $(x(t), y(t), z(t)) = [0.3015, 0.3416, 0.8902]t$; las proyecciones son $\begin{bmatrix} 1.3702 \\ 1.5527 \\ 4.0463 \end{bmatrix}, \begin{bmatrix} 1.8325 \\ 2.0764 \\ 5.4111 \end{bmatrix}$, $\begin{bmatrix} 1.8949 \\ 2.1471 \\ 5.5954 \end{bmatrix} \begin{bmatrix} 0.9989 \\ 1.1319 \\ 2.9498 \end{bmatrix}$.

5. Vea las respuestas al ejercicio 12.3.2.

CAPÍTULO 13

13.1 Ejercicios

1. (a) $(0, 1)$ (b) $(0, 0)$ (c) $(-1/2, -3/8)$ (d) $(1, 1)$

13.1 Problemas de computadora

1. (a) $1/2$ (b) $-2, 1$ (c) 0.47033 (d) 1.43791
3. (a), (b): $(0.358555, 2.788973)$

5. (1.20881759, 1.20881759), aproximadamente 8 decimales correctos.
7. (1, 1)

13.2 Problemas de computadora

1. El mínimo es (1.2088176, 1.2088176). Con condiciones iniciales diferentes se obtendrán respuestas que difieren entre sí en alrededor de $\epsilon^{1/2}$.
3. (1, 1). El método de Newton será exacto hasta la precisión de máquina, pues trata de encontrar una sola raíz. El gradiente descendiente tendrá un error de tamaño $\approx \epsilon^{1/2}$.
5. Igual que el problema de computadora 2.

Bibliografía

Y. Achdou y O. Pironneau [2005] *Computational Methods for Options Pricing*. SIAM, Philadelphia, PA.

A. Ackleh, E. J. Allen, R. B. Kearfott y P. Seshaiyer [2009] *Classical and Modern Numerical Analysis: Theory, Methods, and Practice*. Chapman and Hall, Nueva York.

M. Agoston [2005] *Computer Graphics and Geometric Modeling*. Springer, Nueva York.

K. Alligood, T. Sauer y J. A. Yorke [1996] *Chaos: An Introduction to Dynamical Systems*. Springer, Nueva York.

W. F. Ames [1992] *Numerical Methods for Partial Differential Equations*, 3ra. ed. Academic Press, Boston.

E. Anderson, Z. Bai, C. Bischof, J. W. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney y D. Sorensen [1990] "LAPACK: A Portable Linear Algebra Library for High-performance Computers", Reporte técnico del Departamento de Ciencias Computacionales CS-90-105, University of Tennessee, Knoxville.

U. M. Ascher, R. M. Mattheij y R. B. Russell [1995] *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM, Philadelphia, PA.

U. M. Ascher y L. Petzold [1998] *Computer Methods for Ordinary Differential Equations and Differential-algebraic Equations*. SIAM, Philadelphia, PA.

R. Ashino, M. Nagase y R. Vaillancourt [2000] "Behind and Beyond the MATLAB ODE Suite". *Computers and Mathematics with Application* **40**, 491-572.

R. Aster, B. Borchers y C. Thurber [2005] *Parameter Estimation and Inverse Problems*. Academic Press, Nueva York.

O. Axelsson [1994] *Iterative Solution Methods*. Cambridge University Press, Nueva York.

O. Axelsson y V. A. Barker [1984] *Finite Element Solution of Boundary Value Problems for Ordinary Differential Equations*. Academic Press, Orlando, FL.

Z. Bai, J. Demmel, J. Dongarra, A. Ruhe y H. Van der Vorst [2000] *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. SIAM, Philadelphia, PA.

P. B. Bailey, L. F. Shampine y P. E. Waltman [1968] *Nonlinear Two-Point Boundary-Value Problems*. Academic Press, Nueva York.

R. Bank [1998] "PLTMG, A Software Package for Solving Elliptic Partial Differential Equations". Users' *Guide 8.0*. SIAM, Philadelphia, PA.

R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine y H. van der Vorst [1987] *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, PA.

V. Bhaskaran y K. Konstandtinides [1995] *Image and Video Compression Standards: Algorithms and Architectures*. Kluwer Academic Publishers, Boston, MA.

G. Birkhoff y R. Lynch [1984] *Numerical Solution of Elliptic Problems*. SIAM, Philadelphia, PA.

G. Birkhoff y G. Rota [1989] *Ordinary Differential Equations*, 4a. ed. John Wiley & Sons, Nueva York.

F. Black y M. Scholes [1973] "The Pricing of Options and Corporate Liabilities". *Journal of Political Economy* **81**, 637-654.

P. Blanchard, R. Devaney y G. R. Hall [2002] *Differential Equations*, 2da. ed. Brooks-Cole, Pacific Grove, CA.

F. Bornemann, D. Laurie, S. Wagon y J. Waldvogel [2004] *The SIAM 100-Digit Challenge: A Study in High-Accuracy Numerical Computing*. SIAM, Philadelphia.

W. E. Boyce y R. C. DiPrima [2008] *Elementary Differential Equations and Boundary Value Problems*, 9na. ed. John Wiley & Sons, Nueva York.

G. E. P. Box y M. Muller [1958] "A Note on the Generation of Random Normal Deviates". *The Annals Mathematical Statistics* **29**, 610-611.

R. Bracewell [2000] *The Fourier Transform and its Application*, 3ra. ed. McGraw-Hill, Nueva York.

J. H. Bramble [1993] *Multigrid Methods*. John Wiley & Sons, Nueva York.

K. Brandenburg y M. Bosi [1997] "Overview of MPEG Audio: Current and Future Standards for Low Bit Rate Audio Coding". *Journal of the Audio Engineering Society* **45**, 4-21.

M. Braun [1993] *Differential Equations and Their Applications*, 4a. ed. Springer-Verlag, Nueva York.

S. Brenner y L. R. Scott [2002] *The Mathematical Theory of Finite Element Methods*, 2da. ed. Springer-Verlag, Nueva York.

R. P. Brent [1973] *Algorithms for Minimization without Derivatives*. Prentice Hall, Englewood Cliffs, NJ.

W. Briggs [1987] *A Multigrid Tutorial*. SIAM, Philadelphia, PA.

W. Briggs y V. E. Henson [1995] *The DFT: An Owner's Manual for the Discrete Fourier Transform*. SIAM, Philadelphia, PA.

E. O. Brigham [1988] *The Fast Fourier Transform and Its Applications*. Prentice-Hall, Englewood Cliffs, NJ.

S. Brin y L. Page [1998] "The Anatomy of a Large-scale Hypertextual Web Search Engine". *Computer Networks and ISDN systems* **30**, 107-117.

C. G. Broyden [1965] "A Class of Methods for Solving Nonlinear Simultaneous Equations". *Mathematics of Computation* **19**, 577-593.

C. G. Broyden, J. E. Dennis, Jr. y J. J. Moré [1973] "On the Local and Superlinear Convergence of Quasi-Newton Methods". *IMA Journal of Applied Mathematics* **12**, 223-245.

K. Burrage [1995] *Parallel and Sequential Methods for Ordinary Differential Equations*. Oxford University Press, Nueva York.

J. C. Butcher [1987] *Numerical Analysis of Ordinary Differential Equations*. Wiley, Londres.

E. Cheney [1966] *Introduction to Approximation Theory*. McGraw-Hill, Nueva York.

E. Chu y A. George [1999] *Inside the FFT Black Box*. CRC Press, Boca Ratón, FL.

P. G. Ciarlet [1978] *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam.

CODEE [1999] *ODE Architect Companion*. John Wiley & Sons, Nueva York.

T. F. Coleman y C. van Loan [1988] *Handbook for Matrix Computations*. SIAM, Philadelphia, PA.

R. D. Cook [1995] *Finite Element Modeling for Stress Analysis*. Wiley, Nueva York.

J. W. Cooley y J. W. Tukey [1965] "An Algorithm for the Machine Calculation of Complex Fourier Series". *Mathematics of Computation* **19**, 297-301.

T. Cormen, C. Leiserson, R. Rivest y C. Stein [2009] *Introduction to Algorithms*, 3ra. ed. MIT Press, Cambridge, MA.

R. Courant, K. O. Friedrichs y H. Lewy [1928] "Über die Partiellen Differenzengleichungen der Mathematischen Physik". *Mathematischen Annalen* **100**, 32-74.

J. Crank y P. Nicolson [1947] "A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat Conduction Type". *Proceedings of the Cambridge Philosophical Society* **43**, 1-67.

J. Cuppen [1981] "A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem". *Numerische Mathematik* **36**, 177-195.

B. Datta [2010] *Numerical Linear Algebra and Applications*, 2da. ed. SIAM, Philadelphia.

A. Davies y P. Samuels [1996] *An Introduction to Computational Geometry for Curves and Surfaces*. Oxford University Press, Oxford.

P. J. Davis [1975] *Interpolation and Approximation*. Dover, Nueva York.

P. Davis y P. Rabinowitz [1984] *Methods of Numerical Integration*, 2da. ed. Academic Press, Nueva York.

T. Davis [2006] *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA.

C. de Boor [2001] *A Practical Guide to Splines*, 2da. ed. Springer-Verlag, Nueva York.

J. W. Demmel [1997] *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

J. E. Dennis y Jr., R. B. Schnabel [1987] *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM Publications, Philadelphia, PA.

C. S. Desai y T. Kundu [2001] *Introductory Finite Element Method*. CRC Press. Boca Ratón, FL.

P. Dierckx [1995] *Curve and Surface Fitting with Splines*. Oxford University Press, Nueva York.

J. R. Dormand [1996] *Numerical Methods for Differential Equations*. CRC Press: Boca Ratón, FL.

N. Draper y H. Smith [2001] *Applied Regression Analysis*, 3ra. ed. John Wiley and Sons, Nueva York.

T. Driscoll [2009] *Learning MATLAB*. SIAM, Philadelphia, PA.

P. Duhamel y M. Vetterli [1990] "Fast Fourier Transforms: A Tutorial Review and a State of the Art". *Signal Processing* **19**, 259-299.

C. Edwards y D. Penny [2004] *Differential Equations and Boundary Value Problems*, 5a. ed. Prentice Hall, Upper Saddle River, NJ.

H. Elman, D. J. Silvester y A. Wathen [2004] *Finite Elements and Fast Iterative Solvers*. Oxford University Press, Oxford, UK.

H. Engels [1980] *Numerical Quadrature and Cubature*. Academic Press, Nueva York.

G. Evans [1993] *Practical Numerical Integration*. John Wiley and Sons, Nueva York.

L. C. Evans [2010] *Partial Differential Equations*, 2da. ed. AMS Publications, Providence, RI.

G. Farin [1990] *Curves and Surfaces for Computer-aided Geometric Design*, 2da. ed. Academic Press, Nueva York.

G. S. Fishman [1996] *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, Nueva York.

C. A. Floudas, P. M. Pardalos, C. Adjiman, W. R. Esposito, Z. H. Gms, S. T. Harding, J. L. Klepeis, C. A. Meyer y C. A. Schweiger [1999] *Handbook of Test Problems in Local and Global Optimization*, Vol. 33, Serie titulada Nonconvex Optimization and its Applications, Springer, Berlín, Alemania.

B. Fornberg [1998] *A Practical Guide to Pseudospectral Methods*. Cambridge University Press, Cambridge, UK.

J. Fox [1997] *Applied Regression Analysis, Linear Models, and Related Methods*. Sage Publishing, Nueva York.

M. Frigo y S. G. Johnson [1998] "FFTW: An Adaptive Software Architecture for the FFT". *Proceedings* ICASSP **3**, 1381-1384.

C. W. Gear [1971] *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs. NJ.

J. E. Gentle [2003] *Random Number Generation and Monte Carlo Methods*, 2da. ed. Springer-Verlag, Nueva York.

A. George y J. W. Liu [1981] *Computer Solution of Large Sparse Positive Definite Systems*. Prentice Hall, Englewood Cliff, NJ.

M. Gockenbach [2006] *Understanding and Implementing the Finite Element Method*. SIAM, Philadelphia, PA.

M. Gockenbach [2010] *Partial Differential Equations: Analytical and Numerical Methods*, 2da. ed. SIAM, Philadelphia, PA.

D. Goldberg [1991] "What Every Computer Scientist Should Know about Floating Point Arithmetic". *ACM Computing Surveys* **23**, 5-48.

G. H. Golub y C. F. Van Loan [1996] *Matrix Computations*, 3ra. ed. Johns Hopkins University Press, Baltimore.

D. Gottlieb y S. Orszag [1977] *Numerical Analysis of Spectral Methods: Theory and Applications*. SIAM, Philadelphia, PA.

T. Gowers, J. Barrow-Green e I. Leader [2008] *The Princeton Companion to Mathematics*. Princeton University Press, Princeton, NJ.

I. Griva, S. Nash y A. Sofer [2008] *Linear and Nonlinear Programming*, 2da. ed. SIAM, Philadelphia.

C. Grossmann, H. Roos y M. Stynes [2007] *Numerical Treatment of Partial Differential Equations*. Springer, Berlín, Alemania.

B. Guenter y R. Parent [1990] "Motion Control: Computing the Arc Length of Parametric Curves". *IEEE Computer Graphics and Applications* **10**, 72-78.

S. Haber [1970] "Numerical Evaluation of Multiple Integrals". *SIAM Review* **12**, 481-526.

R. Haberman [2004] *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*. Prentice Hall, Upper Saddle River, NJ.

W. Hackbush [1994] *Iterative Solution of Large Sparse Systems of Equations*. Springer-Verlag, Nueva York.

S. Hacker [2000] *MP3: The Definitive Guide*. O'Reilly Publishing, Sebastopol, CA.

B. Hahn [2002] *Essential MATLAB for Scientists and Engineers*, 3ra. ed. Elsevier, Amsterdam.

E. Hairer, S. P. Norsett y G. Wanner [1993] *Solving Ordinary Differential Equations 1: Nonstiff Problems*, 2da. ed., Springer-Verlag, Berlín.

E. Hairer y G. Wanner [1996] *Solving Ordinary Differential Equations II: Stiff and Differential-algebraic Problems*, 2da. ed., Springer-Verlag, Berlín.

C. Hall y T. Porsching [1990] *Numerical Analysis of Partial Differential Equations*. Prentice Hall, Englewood Cliffs, NJ.

J. H. Halton [1960] "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals". *Numerische Mathematik* **2**, 84-90.

M. Heath [2002] *Scientific Computing*, 2da. ed. McGraw-Hill, Nueva York.

P. Hellekalek [1998] "Good Random Number Generators Are (Not So) Easy to Find". *Mathematics and Computers in Simulation* **46**, 485-505.

P. Henrici [1962] *Discrete Variable Methods in Ordinary Differential Equations*. New York, John Wiley & Sons, Nueva York.

M. R. Hestenes y E. Steifel [1952] "Methods of Conjugate Gradients for Solving Linear Systems". *Journal of Research National Bureau of Standards* **49**, 409-436.

R. C. Hibbeler [2008] *Structural Analysis*, 7ma. ed. Prentice Hall, Englewood Cliffs, NJ.

D. J. Higham [2001] "An Algorithmic Introduction to Numerical Simulation of Stochastic Differential Equations". *SIAM Review* **43**, 525-546.

D. J. Higham y N. J. Higham [2005] *MATLAB Guide*, 2da. ed. SIAM, Philadelphia, PA.

N. J. Higham [2002] *Accuracy and Stability of Numerical Algorithms*. 2da. ed. SIAM Publishing, Philadelphia, PA.

B. Hoffmann-Wellenhof, H. Lichtenegger y J. Collins [2001] *Global Positioning System*: Theory and Practice, 5a. ed. Springer-Verlag, Nueva York.

J. Hoffman [2001] *Numerical Methods for Engineers and Scientists*, 2da. ed. CRC Press, Nueva York.

K. Höllig [2003] *Finite Element Methods with B-Splines*. SIAM, Philadelphia, PA.

M. Holmes [2006] *Introduction to Numerical Methods in Differential Equations*. Springer, Nueva York.

M. Holmes [2009] *Introduction to the Foundations of Applied Mathematics*. Springer, Nueva York.

A. S. Householder [1970] *The Numerical Treatment of a Single Nonlinear Equation*. McGraw-Hill, Nueva York.

J. V. Huddleston [2000] *Extensibility and Compressibility in One-dimensional Structures*, 2da. ed. ECS Publishing, Buffalo, NY.

D. A. Huffman [1952] "A Method for the Construction of Minimum-Redundancy Codes". *Proceedings of the IRE* **40**, 1098-1101.

J. C. Hull [2008] *Options, Futures, and Other Derivatives*, 7ma. ed. Prentice Hall, Upper Saddle River, NJ.

IEEE [1985] Norma para la aritmética del punto flotante binario, IEEE Std. 754-1985, IEEE, Nueva York.

I. Ipsen [2009] *Numerical Matrix Analysis: Linear Systems and Least Squares*. SIAM, Philadelphia, PA.

A. Iserles [1996] *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press, Cambridge, UK.

C. Johnson [2009] *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover Publications, Nueva York.

P. Kattan [2007] *MATLAB Guide to Finite Elements*, 2da. ed. Springer, Nueva York.

H. B. Keller [1968] *Numerical Methods of Two-Point Boundary-Value Problems*. Blaisdell, Waltham, MA.

C. T. Kelley [1995] *Iterative Methods for Linear and Nonlinear Problems*. SIAM Publications, Philadelphia, PA.

J. Kepner [2009] *Parallel MATLAB for Multicare and Multinode Computers*. SIAM, Philadelphia, PA.

E. Klebaner [1998] *Introduction to Stochastic Calculus with Applications*. Imperial College Press, Londres.

P. Kloeden y E. Platen [1992] *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, Berlín, Alemania.

P. Kloeden, E. Platen y H. Schurz [1994] *Numerical Solution of SDE through Computer Experiments*. Springer-Verlag, Berlín, Alemania.

P. Knaber y L. Angerman [2003] *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*. Springer, Berlín, Alemania.

D. Knuth [1981] *The Art of Computer Programming*. Addison-Wesley, Reading, MA.

D. Knuth [1997] *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 3ra. ed. Addison-Wesley, Reading, MA.

E. Kostelich y D. Armbruster [1997] *Introductory Differential Equations: From Linearity to Chaos*. Addison Wesley, Boston, MA.

A. Krommer y C. Ueberhuber [1998] *Computational Integration*. SIAM, Philadelphia, PA.

M. Kutner, C. Nachtsheim, J. Neter y W. Li [2004] *Applied Linear Statistical Models*, 5a. ed. McGraw-Hill, Nueva York.

J. C. Lagarias, J. A. Reeds, M. H. Wright y P. E. Wright [1998] "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions". *SIAM Journal of Optimization* **9**, 112-147.

J. D. Lambert [1991] *Numerical Methods for Ordinary Differential Systems*, John Wiley & Sons, Nueva York.

L. Lapidus y G. F. Pinder [1982] *Numerical Solution of Partial Differential Equations in Science and Engineering*. Wiley-Interscience, Nueva York.

S. Larsson y V. Thomee [2008] *Partial Differential Equations with Numerical Methods*. Springer, Berlín, Alemania.

C. L. Lawson y R. J. Hanson [1995] *Solving Least Squares Problems*. SIAM Publications, Philadelphia, PA.

D. Lay [2011] *Linear Algebra and Its Applications*, 4a. ed. Pearson Education, Boston, MA.

K. Levenberg [1944] "A Method for the Solution of Certain Nonlinear Problems in Least Squares". *The Quarterly of Applied Mathematics* **2**, 164-168.

R. Leveque [2007] *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, Philadelphia, PA.

J. D. Logan [2004] *Applied Partial Differential Equations*, 2da. ed. Springer, Nueva York.

D. L. Logan [2011] *A First Course in the Finite Element Method*, 5a. ed. CL-Engineering, Nueva York.

H. S. Malvar [1992] *Signal Processing with Lapped Transforms*. Artech House, Norwood, MA.

D. Marquardt [1963] "An Algorithm for Least-Squares Estimation of Nonlinear Parameters". *SIAM J. on Applied Mathematics* **11**, 431-441.

G. Marsaglia [1968] "Random Numbers Fall Mainly in the Planes". *Proceedings of the National Academy of Sciences* **61**, 25.

G. Marsaglia y A. Zaman [1991] "A New Class of Random Number Generators". *Annals of Applied Probability* **1**, 462-480.

G. Marsaglia y W. W. Tsang [2000] "The Ziggurat Method for Generating Random Variables", *Journal of Statistical Software* **5**, 1-7.

R. McDonald [2006] *Derivatives Markets*, 2da. ed. Pearson Education, Boston, MA.

P. J. McKenna y C. Tuama [2001] "Large Torsional Oscillations in Suspension Bridges Visited Again: Vertical Forcing Creates Torsional Response". *American Mathematical Monthly* **108**, 738-745.

J.-P. Merlet [2000] *Parallel Robots*. Kluwer Academic Publishers, Londres.

A. R. Mitchell y D. F. Griffiths [1980] *The Finite Difference Method in Partial Differential Equations*. Wiley, Nueva York.

C. Moler [2004] *Numerical Computing with* MATLAB. SIAM, Philadelphia, PA.

J. Moré y S. Wright [1987] *Optimization Software Guide*. SIAM, Philadelphia, PA.

K. W. Morton y D. F. Mayers [1996] *Numerical Solution of Partial Differential Equations*. Cambridge University Press, Cambridge, UK.

J. A. Nelder y R. Mead [1965] "A Simplex Method for Function Minimization". *Computer Journal* **7**, 308-313.

M. Nelson y J. Gailly [1995] *The Data Compression Book*, 2da. ed. M&T Books, Redwood City, CA.

H. Niederreiter [1992] *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM Publications, Philadelphia, PA.

J. Nocedal y S. Wright [1999] *Numerical Optimization*, Serie Springer de Investigación de operaciones. Springer, Nueva York.

B. Oksendal [1998] *Stochastic Differential Equations: An Introduction with Applications*, 5a. ed. Springer-Verlag, Berlín, Alemania.

A. Oppenheim y R. Schafer [2009] *Discrete-time Signal Processing*, 3ra. ed. Prentice Hall, Upper Saddle River, NJ.

J. M. Ortega [1972] *Numerical Analysis: A Second Course*. Academic Press, Nueva York.

A. M. Ostrowski [1966] *Solution of Equations and Systems of Equations*, 2da. ed. Academic Press, Nueva York.

M. Overton [2001] *Numerical Computing with IEEE Floating Point Arithmetic*. SIAM Publishing, Philadelphia, PA.

S. Park y K. Miller [1988] "Random Number Generators: Good Ones Are Hard to Find". *Communications of the ACM* **31**, 1192-1201.

B. Parlett [1998] *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia, PA.

B. Parlett [2000] "The QR Algorithm". *Computing in Science and Engineering* **2**, 38-42.

W. Pennebaker y J. Mitchell [1993] *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, Nueva York.

R. Piessens, E. de Doncker-Kapenga, C. Ueberhuber y D. Kahaner [1983] *QUADPACK: A Subroutine Package for Automatic Integration*, Springer, Nueva York.

G. Pinski y F. Narin [1976] "Citation Influence for Journal Aggregates of Scientific Publications: Theory, with Application to the Literature of Physics". *Information Processing and Management* **12**, 297-312.

J. Polking [1999] *Ordinary Differential Equations Using MATLAB*. Prentice Hall, Upper Saddle River, NJ.

H. Prautzsch, W. Boehm y M. Paluszny [2002] *Bézier and B-Spline Techniques*. Springer, Berlín, Alemania.

A. Quarteroni, R. Sacco y F. Saleri [2000] *Numerical Mathematics*. Springer, Berlín, Alemania.

K. R. Rao y J. J. Hwang [1996] *Techniques and Standards for Image, Video, and Audio Coding*. Prentice Hall, Upper Saddle River, NJ.

K. R. Rao y P. Yip [1990] *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Boston, MA.

J. R. Rice y R. F. Boisvert [1984] *Solving Elliptic Problems Using ELLPACK*. Springer-Verlag, Nueva York.

T. J. Rivlin [1981] *An Introduction to the Approximation of Functions*, 2da. ed. Dover, Nueva York.

T. J. Rivlin [1990] *Chebyshev Polynomials*, 2da. ed. John Wiley and Sons, Nueva York.

S. Roberts y J. Shipman [1972] *Two-Point Boundary Value Problems: Shooting Methods*. Elsevier, Nueva York.

R. Y. Rubinstein [1981] *Simulation and the Monte Carlo Method*. John Wiley, Nueva York.

T. Ryan [1997] *Modern Regression Methods*. John Wiley and Sons.

Y. Saad [2003] *Iterative Methods for Sparse Linear Systems*, 2da. ed. SIAM Publishing, Philadelphia, PA.

D. Salomon [2005] *Curves and Surfaces for Computer Graphics*. Springer, Nueva York.

K. Sayood [1996] *Introduction to Data Compression*. Morgan Kaufmann Publishers, San Francisco.

M. H. Schultz [1973] *Spline Analysis*. Prentice Hall, Englewood Cliffs, NJ.

L. L. Schumaker [1981] *Spline Functions: Basic Theory*. John Wiley, Nueva York.

L. F. Shampine [1994] *Numerical Solution of Ordinary Differential Equations*. Chapman & Hall, Nueva York.

L. F. Shampine, I. Gladwell y S. Thompson [2003] *Solving ODEs with MATLAB*. Cambridge University Press, Cambridge, UK.

L. F. Shampine y M. W. Reichelt [1997] "The MATLAB ODE Suite". *SIAM Journal on Scientific Computing* **18**, 1-22.

K. Sigmon y T. Davis [2002] *MATLAB Primer*, 6a. ed. CRC Press, Boca Ratón, FL.

S. Skiena [2008] *The Algoritlim Design Manual*, 2da. ed. Springer, Nueva York.

I. Smith y D. Griffiths [2004] *Programming the Finite Element Method*, John Wiley, Nueva York.

B. T. Smith, J. M. Boyle, Y. Ikebe, V. Klema y C. B. Moler [1970] *Matrix Eigensystem Routines: EISPACK Guide*, 2da. ed. Springer-Verlag, Nueva York.

W. Stallings [2003] *Computer Organization and Architecture*. 6a. ed. Prentice Hall, Upper Saddle River. NJ.

J. M. Steele [2001] *Stochastic Calculus and Financial Applications*. Springer-Verlag, Nueva York.

G. W. Stewart [1973] *Introduction to Matrix Computations*. Academic Press, Nueva York.

G. W. Stewart [1998] *Afternotes on Numerical Analysis: Afternotes Goes to Graduate School*. SIAM, Philadelphia, PA.

J. Stoer y R. Bulirsch [2002] *Introduction to Numerical Analysis*, 3ra. ed. Springer-Verlag, Nueva York.

J. A. Storer [1988] *Data Compression: Methods and Theory*. Computer Science Press, Rockville, MD.

G. Strang [1988] *Linear Algebra and Its Applications*, 3ra. ed. Saunders, Philadelphia.

G. Strang [2007] *Computational Science and Engineering*. Wellesley-Cambridge Press, Cambridge, MA.

G. Strang y K. Borre [1997] *Linear Algebra, Geodesy, and GPS*. Wellesley Cambridge Press, Cambridge, MA.

G. Strang y G. J. Fix [1973] *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ.

J. C. Strikwerda [1989] *Finite Difference Schemes and Partial Differential Equations*. Wadsworth and Brooks-Cole, Pacific Grove, CA.

W. A. Strauss [1992] *Partial Differential Equations: An Introduction*. John Wiley and Sons, Nueva York.

A. Stroud y D. Secrest [1966] *Gaussian Quadrature Formulas*. Prentice Hall, Englewood Cliffs, NJ.

P. N. Swarztrauber [1982] "Vectorizing the FFTs". En: *Parallel Computations*, ed. G. Rodrigue, pp. 51-83. Academic Press, Nueva York.

D. S. Taubman y M. W. Marcellin [2002] *JPEG 2000: Image Compresion Fundamentals, Standards and Practice*. Kluwer, Boston, MA.

J. Traub [1964] *Iterative Methods for the Solution of Equations*. Prentice-Hall, Englewood Cliffs, NJ.

N. Trefethen [2000] *Spectral Methods in MATLAB*. SIAM, Philadelphia.

N. Trefethen y D. Bau [1997] *Numerical Linear Algebra*. SIAM, Philadelphia, PA.

A. Turing [1952] "The Chemical Basis of Morphogenesis". *Philosophical Transactions Royal of the Society Lond*. B **237**, 3772.

C. Van Loan [1992] *Computational Frameworks for the Fast Fourier Transform*. SIAM Publications, Philadelphia, PA.

C. Van Loan y K. Fan [2010] *Insight Through Computing: A MATLAB Introduction to Computational Science and Engineering*. SIAM, Philadelphia, PA.

R. S. Varga [2000] *Matrix Iterative Analysis*, 2da. ed. Springer-Verlag, Nueva York.

J. Volder [1959] "The CORDIC Trigonometric Computing Technique". *IRE Transactions on Electronic Computing* **8**, 330-334.

G. K. Wallace [1991] "The JPEG Still Picture Compression Standard". *Communications of the ACM* **34**, 30-44.

H. Wang, J. Kearney y K. Atkinson [2003] "Arc-length Parameterized Spline Curves for Real-time Simulation". En: *Curve and Surface Design*: Saint Malo 2002, Eds. T. Lyche, M. Mazure y L. Schumaker. Nashboro Press, Brentwood, TN.

Y. Wang y M. Vilermo [2003] "The Modified Discrete Cosine Transform: Its Implications for Audio Coding and Error Concealment". *Journal of the Audio Engineering Society* **51**, 52-62.

D. S. Watkins [1982] "Understanding the QR Algorithm". *SIAM Review* **24**, 427-440.

D. S. Watkins [2007] *The Matrix Eigenvalue Problem: GR and Krylow Subspace Methods*. SIAM, Philadelphia.

J. Wilkinson [1965] *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford.

J. Wilkinson [1984] "The Perfidious Polynomial". En: Studies in Numerical Analysis. Ed: G. Golub. MAA, Washington, DC.

J. Wilkinson [1994] *Rounding Errors in Algebraic Processes*. Dover, Nueva York.

J. Wilkinson y C. Reinsch [1971] *Handbook for Automatic Computation, Vol. 2: Linear Algebra*. Springer-Verlag, Nueva York.

P. Wilmott, S. Howison y J. Dewynne [1995] *The Mathematics of Financial Derivatives*. Cambridge University Press, Oxford y Nueva York.

S. Winograd [1978] "On Computing the Discrete Fourier Transform". *Mathematics of Computation* **32**, 175-199.

F. Yamaguchi [1988] *Curves and Surfaces in Computer-aided Geometric Design*. Springer-Verlag, Nueva York.

D. M. Young [1971] *Iterative Solution of Large Linear Systems*. Academic Press, Nueva York.

Índice

actualización polinomio de interpolación, 144
acumulación, 453
Adobe Corp., 138
aerogeneradores, 211
aguja de Buffon, 445
ajuste de datos, 188
aleta de enfriamiento, 403
algoritmo
  estable, 50
  ziggurat, 439
algoritmo QR, 544
  desplazado, 543-544
  no desplazado, 541
    convergencia, 541
animación por computadora, 243
Apple Corp., 138
árbol
  de Huffman, 517
  DPCM, 517
archivo
  de audio
    aac, 495
    `mp3`, 496
    `wav`, 490, 529
  de imagen
    escala de grises, 505
    JPEG, 495, 512
    línea de base JPEG, 512
  PDF, 183
aritmética computacional, 45
atasco, 91
atractor caótico, 320
axón del calamar, 318

baricentro, 409
base, 60, 39
  ortonormal, 539, 554
Bézier, P., 138, 179
bien condicionado, **50**
bifurcación
  pandeo, 356
bit, 6
Black F., 431, 464
bombeo de google, 551
búsqueda
  de la sección dorada, 566
  de Nelder-Mead, 571, 581
  del gradiente, 577
  en la web, 549
byte, 11

cálculo vectorial, 588
caminata aleatoria, **447**
  sesgada, 451
campo
  de direcciones, 282
  de pendientes, 282
`castanets.wav`, 490, 492
Casteljau, P., 138, 179
chip DSP, 473
clasificación de página, 549
clima subterráneo, 210
cociente de Rayleigh, **534**
codec, 526
codificación
  de corrida larga, 518
  de Huffman, 501, 515
    en JPEG, 517
coeficiente de difusión, 375
completar el cuadro, 117
componente
  AC, 517
  DC, 504, 517
compresibilidad, 355
compresión, 194
  con pérdidas, 508, 514, 559
  de datos, 138
  de imagen, 505, 508, 561
condición
  CFL, 396
  inicial, 282
condicionamiento
  de ecuaciones normales, 197
condiciones
  de frontera
    convectivas, 405
    de Dirichlet, 383, **398**
    de Neumann, 383, **398**
    de Robin, 405
    homogéneas, 383
  iniciales y de frontera, 375

conducción, 403
conductividad térmica, 404
confinamiento, 25
conjugado
de un número complejo, **468**
conjunto convexo, **288**
constante de Lipschitz, **288**
continua de Lipschitz, **288**
convección, 403
de Rayleigh-Bénard, 319
convergencia, 33
cuadrática, **53**, 57
lineal, **35**, 37, 40, 55
local, **36**, 53, 56-57
superlineal, 61, 135
conversión
de binario a decimal, 7
de decimal a binario, 6
Cooley, J., 473
CORDIC, 165
criterio
de Dahlquist, 341
de detención, 40, 47, 65, 575
cuadratura, 254
adaptativa, 269-270
gaussiana, 276
cuantificación, 508, 561
estándar JPEG, 512
lineal, 508
cuneiforme, 39
curva
Bézier, 179, 279
en archivo PDF, 183
de demanda, 199
de la campana, 438
de progreso, 280

datos
altura *vs* peso, 207
consumo de petróleo de Japón, 210
CPU Intel, 205
oferta de automóviles, 204
temperatura, 201
deflación, 543
dependencia sensible
a las condiciones iniciales, **311**, 320
derivada, 244
parcial, 334
simbólica, 250
derivado financiero, 464
descomposición de valor singular, 554
cálculo de, 562
no unicidad, 554
desigualdad
de Cauchy-Schwarz, 198
de Gronwall, 289
determinante, 30, 557
diagrama de telaraña, 34, **34**, 42
diferencia hacia adelante, 244
diferenciación
numérica, 244
diferencias divididas, 141
difusión, 453
difusividad térmica, 375
dígitos significativos, 43
pérdida de, 248
dinámica caótica, 43, 60
dióxido de carbono, 150, 178, 211
discretización, 71, 102, 357, 375
disipador de calor, 403
división sintética, 3
doble
hélice, 565
precisión, 8, 43-44, 92, 197
dominio fundamental, 151

ecuación
de Burgers, 417, 419
de difusión, 375
de Langevin, 457
de Laplace, 398, 414
de Maxwell, 399
de onda, 393
de Poisson, 398
de reacción-difusión, 390, 421
de Van der Waals, 60
del calor, 375, 385
diferencial, 281
autónoma, **282**
estocástica, 452
lineal de primer orden, 291
ordinaria, 282, 349
rígida, **333**
diferencial parcial, 374
elíptica, 398, 404
hiperbólica, 393
parabólica, 375
elíptica
forma débil, 407
logística, 282

ecuaciones
de Lorenz, 319
de Navier-Stokes, 428
inconsistentes, **189**
normales, 191, 498
ejemplo de Runge, 155
elipsoide, 554
épsilon máquina, 9, 12-13, 46, 248, 532
equipartición, 278
error
absoluto, **10**, 40
cuadrado, **192**
cuadrático medio de la raíz, **192**
cuantificación, **508**
entrada, 88
estándar, 448
hacia atrás, **45**, 50, 86, 93
hacia delante, **45**, 50, 86, 93, 197
interpolación, **151**, 155, 159
local de truncamiento, **293**, 327, 376
redondeo, 10, 248
relativo, **10**, 40
relativo hacia adelante, 87
relativo hacia atrás, 87
tolerancia, 326
truncamiento, 248
truncamiento global, **293**
esfuerzo, 71
estabilidad
condicional, 380, 395
de Von Neumann, 379
incondicional, 382
estándar JPEG, 495
Anexo K, 512
estrictamente diagonal dominante, 107, 171
etapa del solucionador de EDO, 315
exponente, 8
extrapolación, 249, 254, 265, 360, 364
de Richardson, 249
local, 327
factor
de integración, 290
de magnificación del error, 49, 88, 241
factorización
de Cholesky, 119, 121
LU, 79
PA = LU, **98**
QR, 215, 539
conteo de operaciones, 223
reducida, 213
reveladora del valor propio, 542
fenómeno de Runge, 155, 157-158, 367
filtración pasa bajas, 507
flujo
de agua subterránea, 416
incompresible, 399
fl(x), 10
forma
de Hessenberg superior, 544, 562
de Schur real, 542
de tabla, 92
fórmula
cuadrática, 17
de Black-Scholes, 431, 464
de Euler, 468, 477
de la diferencia
centrada, 246, 358, 376
hacia adelante, 245, 376
de sensibilidad para raíces, 48
de Sherman-Morrison, 585
de Taylor, 53
del error de interpolación, 152
Fourier, J., 468
primera ley de, 404
FSAL, **327**, 329
fuerza de Van der Waals, 565, 580
función
armónica, 398
característica, 435
de distribución
acumulada, 437
de probabilidad, 437
de ventana, 529
integrable de Riemann, 409
objetivo, 565
ortogonal, 483
trigonométrica
de orden n, **477**
graficado de, 480
unimodal, **566**

Gauss, C.F., 188
gaussiana
cuadratura, 276
eliminación, 72, 92, 358
conteo de operaciones, 75-77
forma matricial, 79
forma tabular, de 73
simple, 72, 95
Gauss-Seidel, Método de, **109**

generador
de congruencia lineal, 433
de números aleatorios
estándar mínimo, **434**, 437
periodo, 433
RANDNUM, 439
randu, 435
uniforme, 432
GIS, 240
GMRES, 226
precondicionado, 228
reiniciado, 228
Google.com, 549
Gough, E., 24
GPS, 188, 233, 238
condicionamiento de, 241
gradiente, 230, 576
descendente, 577
grado de precisión, **258**, 273
Gram-Schmidt modificado, 218

helado, 60
Herón de Alejandría, 39
hessiana, 231
hipotenusa, 19
Hodgkin, A., 317
Huxley, A., 317

ICI, *vea* Interpolación cuadrática inversa
IEEE, 8, 23, 92
imagen a color
RGB, 505
YUV, **512**
IMSL, 23
información
de Shannon, 515
ingeniería estructural, 71, 83
Instituto Scripps, 211
integración
de Romberg, 266-267
numérica, 254
compuesta, 259
integral
de Ito, 453
de longitud de arco, 243
longitud de arco, 265
impropia, 263, 265
de Riemann, 453
Intel Corp., 374
intercambio de filas, 95
interpolación, **139**
cuadrática inversa, 64-65, 69
de Chebyshev, 159, 162
de la diferencia dividida de Newton, 142, 153
de Lagrange, 64, **140**, 255
parabólica sucesiva, 569
polinomial, 254
de Chebyshev, 159
por funciones ortogonales, 497
trigonométrica, 467, 476
iteración
de potencia, 532, 549
convergencia, 534
desplazada, 536
inversa, 535
de punto fijo, 31, 334
divergencia, 34
geometría, 33
del cociente de Rayleigh, 537
normalizada simultánea, 540
IPF, *vea* iteración de punto fijo

jacobiana, *vea* matriz jacobiana, 361
junta prismática, 67
justificado a la izquierda, 8

Keeling, C., 211

lápiz, 44
Laplaciano, 398
Legendre, A., 188
Ley
de Fick, 375
de Hooke, 322
de los gases ideales, 60
de Moore, 206, 374
de potencia, 206, 445
del arcoseno, 452
línea de mínimos cuadrados, 193
Lorenz, E., 319
luminancia, **512**

mal condicionado, **50**, 90, 367
magnitud
de un número complejo, **468**
de un vector complejo, 471
mantisa, **8**
manufactura asistida por computadora, 243
Maple, 23
matemáticas babilónicas, 39

Mathematica, 23
MATLAB
animación en, 279
cuadro de herramientas simbólicas, 241
MATLAB, código
ab2step.m, 337, 343
adapquad.m, 271
am1step.m, 343
bezierdraw.m, 181
bisect.m, 28,353
broyden2.m, 135
brusselator.m, 427
burgers.m, 419
bvpfem.m, 372
clickinterp.m, 147
crank.m, 387
cubrt.m, 593
dftfilter.m, 488,492
dftinterp.m, 480
euler.m, 286
euler2.m, 303
eulerstep.m, 286
exmultistep.m, 337
fisher2d.m, 425
fpi.m, 32
gss.m, 568
halton.m, 443
heatbdn.m, 384
heatfd.m, 378, 381
hessen.m, 546
hh.m, 318
invpowerit.m, 536
jacobi.m, 115
nest.m, 3, 146, 148, 165
newtdd.m, 146, 148
nlbvpfd.m, 362
nsi.m, 540
orbit.m, 310
pend.m, 307
poisson.m, 402, 406
poissonfem.m, 412
powerit.m, 534
predcorr.m, 343
rk4step.m, 319
romberg.m, 267
rqi.m, 537
shiftedqr.m, 543
shiftedqr0.m, 543
sin2.m, 165
sparsesetup.m, 115
spi.m, 570
splinecoeff.m, 172
splineplot.m, 173
tacoma.m, 324
trapstep.m, 308, 324, 337
unshiftedqr.m, 541
unstable2step.m, 337
weaklystab2step.m, 337
wilkpoly.m, 47
MATLAB, comandos
axis, 592, 597
backslash, 89, 94, 412
break, 594
button, 147
cla, 597
clear, 590
cond, 89
conj, 494
dct, 504
det, 30
diag, 115, 378
diary, 590
diff, 251
double, 505
drawnow, 307, 598
eig, 30, 547
erf, 273
error, 75, 595
fft, 472, 480, 494
figure, 592
fminunc, 582
for, 594
format, 591
format hex, 7, 11
fprintf, 591
fzero, 44, 47, 51, 65, 69
ginput, 147, 181
global, 319, 596
grid, 592
handel, 490
hilb, 30, 90
ifft, 472, 480, 494
imagesc, 505
imread, 505, 513
int, 251
interpl, 187
length, 115, 597
line, 280, 324
load, 590
log, 590

loglog, 265
lu, 101, 115, 446
max, 30, 534
mean, 596
mesh, 392, 402, 406, 592
nargin, 596
ode23s, 331, 335
ode45, 329, 331, 353
odeset, 329
ones, 90, 115,597
pause, 598
pi, 30
plot, 30, 591
plot3, 581
polyfit, 187, 196
polyval, 187, 196
pretty, 251
qr, 540, 541, 543
rand, 437
randn, 439, 456, 494
rem, 594
round, 286, 529
semilogy, 592
set, 280, 307
simple, 251
size, 597
solve, 241
sound, 490, 492, 529
spdiags, 115, 371
spline, 175, 187
std, 494, 596
subplot, 319, 592
subs, 241
surf, 413, 592
svd, 555, 562
syms, 241, 251
wavread, 490, 529
wavwrite, 490
while, 594
xdata, 598
ydata, 598
zeros, 115, 597
matriz
adyacencia, 550
bandas, 104
coeficiente, 79
completa, 113
cuantificación, 508
de Fourier, 471
de google, 551
de Hilbert, **30**, 79, 94, 130, 200, 225, 594
de rango uno, 558, 584
de Van der Monde, **197**
definida positiva, **117**, 578
diagonalizable, 587
dispersa, 71, 113
estocástica, **547**
estructura de, 83
hessiana, 576
identidad, 584
inversa, 557
invertible, **584**
jacobiana, 131, 576
no simétrica, 541
número de condición, 88, **88**
ortogonal, 215, 483, 495, 520, 542, 554
permutación, **97**, 98
proyección, **220**
semejante, 542, 587
simétrica, **117**, 539
singular, **584**
superior de Hessenberg, 544
transpuesta, 190
triangular
inferior, 79
superior, 79, 215, 542
tridiagonal, 171, 359, 379
unitaria, 471
Mauna Loa, 150
media muestral, **448**
método
Bogacki-Shampine, 327
de Adams-Bashforth, 336, 339, 341
de Adams-Moulton, 342, 345
de bisección, 25, 44, 46, 51, 65, 69, 352, 354, 364
criterio de detención, 29
eficiencia, 28
de Box-Muller, 438
de Brent, 64, 69
de Broyden, 134, 357, 585
de colocación
para PVF, 365
de Crank-Nicolson, 254, 385
estabilidad, 387
de disparo, 352, 357
de Dormand-Prince, 328
de Euler, 284, 333
convergencia, 296
error de truncamiento global, 296
error de truncamiento local, 294

mejorado, 298
orden, 296
de Euler-Maruyama, 456
de Galerkin, 367, 407
de Gauss-Newton, 231, 236, 241
de Heun, 298
de Horner, 3
de Jacobi, 106
de la diferencia finita, 358, 375
explícito, 395
inestable, 378
de la diferencia hacia adelante
análisis de estabilidad, 379
condicionalmente estable, **380**
explícito, **376**
de la diferencia hacia atrás, 380
de la posición falsa, de 63
convergencia lenta, 63
de la secante, 61, 64, 65
convergencia, 61
convergencia lenta, 63
de Levenberg-Marquardt, 236
de Milne-Simpson, 344
de Milstein, 458
de Muller, 63
de Newton-Raphson, *vea* Newton, método
de rechazo, 439
de Runge-Kutta, 314
cuarto orden, 316, 339
de primer orden estocástico, 460
error de truncamiento global, 317
orden 2/3, 327
par integrado, 326
de Runge-Kutta-Fehlberg, 328
de Taylor, 300
del elemento finito, 367
del gradiente conjugado, 122, 127
precondicionado, 127
del punto medio, 314, 336
del residuo mínimo generalizado, 226, 228
del trapezoide
explícito, **297**, 336
implícito, 342
directo, 106
espectral, 367
iterativo, 106
predictor-corrector, 342
simplex cuesta abajo, 571
métodos de varios pasos, 336
consistentes, **341**
convergentes, **341**
débilmente estable, **340**
error local de truncamiento, 339
estables, **340**, 341
fuertemente estables, **340**
métodos Krylov, 226
mínimos cuadrados, 558
de la TDC, 499
no lineal, 203
parábola, 488
por factorización QR, 217
trigonométricos, 485
módulo de Young, 71, 102
momento de inercia, 102
Moore, G. C., 206
movimiento del proyectil, 349, 354
muestreo de importancia, 529
multiplicación
anidada, 2, 139
matricial por bloques, 585
multiplicidad, **46**, 50
modelación asistida por computadora, 278
modelo
Brusselator, 426
de concentración del fármaco, 208
de población, 282
exponencial, 203
ley de potencia, 206
linealización de, 204
Monte Carlo
convergencia, 445
cuasialeatorio, 444
pseudoaleatorio, 440
Tipo 1, 434
Tipo 2, 435
movimiento browniano, 456
continuo, 450
discreto, 446
geométrico, 464

NAG, 23
Napoleón, 468
neurociencia computacional, 317
neurona de Hodgkin-Huxley, 317
Newton
ley de enfriamiento, 404
método, 52, 69, 334, 576
convergencia, 53
modificado el 57

multivariado, **131**, 231, 233, 360
periodicidad, 58
segunda ley del movimiento, 282, 305, 309, 322, 349
Newton-Cotes, fórmula, 255
abierta, 262
cerrada, 259
norma
de infinitud, 86
euclidiana, 212
matricial, 88, **90**
máxima, 86
vectorial, **90**
norma-2, 192, 198
nudo
spline cúbica, 167
número
aleatorio
cuasi-, 442
exponencial, 437
normal, 438
pseudo-, 432
uniforme, 432
binario, 5
infinitamente repetitivo, 7
complejo, 468
representación polar, **468**
de condición, 50, 88, 197, 289, 532
de Prandtl, 320
de punto flotante, 8
cero, 13
normalizado, 8
subnormal, 12
de Reynolds, 320
decimal, 5
hexadecimal, 7

opción
con barrera, 465
de compra, 464
de venta, 465
optimización no restringida, 566
orden
de aproximación, 244
de un solucionador de EDO, **296**
de una ecuación diferencial, 303
ortogonal
función, **368**
matriz, 215
ortogonalización, 539
de Gram-Schmidt, 212, 214, 218
modificada, 218
ortonormal, 552, 587
conteo de operaciones, 215

palabra de computadora, 8
pandeo
de anillo circular, 348, 355
panel, 259
parábola, 64
interpolación, 139
mínimos cuadrados, 194
parámetro
de pérdida, 508
de relajación, 110
patrones de Turing, 426
péndulo, 305
amortiguado, 308
doble, 309
pérdida de significancia, 16, 248
pivote, 75, 101
pivoteo parcial, **95**, 100
placa caliente, 416
plantilla, **376**
plataforma Stewart, **24**, 67
plana, 67
población mundial, 151, 178
Poincaré, H., 311
polinomio
característico, 532
de Chebyshev, **159**, 367
de Legendre, 275
de Taylor, 21, 48
de Wilkinson, 47, 50-51, 88, 532
evaluación de, 1
mónico, **161**
ortogonal, **274**
posiciones decimales
correcto hasta, **28**
PostScript, 138
potencial, 398
de Lennard-Jones, 565, 580
electrostático, 415
precio del ejercicio, 464
precisión
doble largo, *vea* precisión extendida
extendida, 8
simple, 8
precondicionador, 126
de Gauss-Seidel, 127
de Jacobi, 126
SRSS, 127

precondicionamiento, 125
Prigogine, I., 426
primo de Mersenne, **434**
problema
  de tres cuerpos, 311
  de un cuerpo, 309
  de valor de frontera, 348
    existencia y unicidad de soluciones, 350
    no lineal, 360
    para los sistemas 353,
  de valor inicial, 282
    existencia y unicidad, 288
  directo de cinemática, 24, 67
  inverso de cinemática, de 67
proceso
  de Markov, 551
  de Ornstein-Uhlenbeck, 457
  estocástico, 447
    de tiempo continuo, 452
producción mundial de petróleo, 157
producto
  exterior, 584
  interno, 584
  punto, 190
proyección ortogonal, 559
psicoacústica, 528
puente browniano, 461
Puente Tacoma Narrows, 281, 322
pulido, 113
puntal, 67
punto
  de inflexión, 169
  fijo, 31
  medio, 26-27, 62
puntos básicos, 143
PVF, *vea* problema de valor de frontera

radio espectral, 111, 382, 588
raíz, 6, **25**
  cuadrada, 30, 38, 54
  cúbica, 30
  de unidad, 469
    primitiva, **469**
  doble, 46
  múltiple, **46**, 56, 59
  simple, **46**
  triple, 46
rango, 557
razón de muestreo, 490
recorte, 9
redondeo, 9
  al más cercano, 9, 14, 15
reducción de dimensiones, 559
reflector de Householder, 220, **220**, 545-546
regla
  de Boole, 264
  de Simpson, 257, 327, 344
    adaptativa, 272
    compuesta, **261**
  del producto punto, 230
  del producto, matriz/vector, 589
  del punto medio, 262
    bidimensional, 410
    compuesta, **263**
  del trapezoide, 257, 298
    adaptativa, 269
    compuesta, **260**
Regula Falsi, *vea* método de la posición falsa
relación recursiva, polinomios de Chebyshev, 160
rellenado, 113, 115
  de ceros, 524
reloj atómico, 239
residuo, 86, 125, 234, 368
  de Taylor, 21
resolución hacia atrás, *vea* sustitución hacia atrás
rigidez, 71
RKF45, *vea* método de Runge-Kutta-Fehlberg
RMEC, **192**
robot, 24
ruido, 492
  gaussiano, 493

Scholes, M., 431, 464
sección cónica, 311
secuencia de baja discrepancia, 442
secuencia de Halton, 443
secuencia de Van der Corput, 443
semilla aleatoria, 432
sensibilidad, **48**
separación de variables, 287
series de tiempo, 476
sesgo exponente, 11
Shannon, C., 515
signo, 8
simulador de vuelo, 24
sinusoidales
  mínimos cuadrados, 201

Sobre-Relajación Sucesiva, 109
solución
de equilibrio, 334
por mínimos cuadrados, 189
solucionador de EDO
con tamaño de paso variable, 325
convergencia, 296
de varios pasos, 336
explícito, **332**
implícito, **333**
soporte, 38, 62
spline
B, 408
lineal por partes, **369**
cúbica, 167
condiciones en los extremos, 169
de curvatura ajustada, 173
natural, 169
predeterminado en MATLAB, 175
restringida, 174
sin nudo, 175
terminada parabólicamente, 174
de Bézier, 138, 179
lineal, 166
SRS, *vea* Sobre-Relajación Sucesiva
submatriz principal, 118
sustitución hacia atrás, 73, 76-77, 83

tamaño
de paso, 284, 376, 417
en código JPEG, 517
temperatura
de congelación, 24
de masa, 404
Teorema
de Green, 407
de Poincaré-Bendixson, 308
de Rolle, 20
de Taylor, 21, 244, 338
del límite central, 450
del valor intermedio, 20, 25, 29
generalizado, **245**
del valor medio, 20, 35
para integrales, **22**, 256, 262
fundamental del álgebra, 141
teoría del arbitraje, 464
tiempo
de escape, 448
de primer paso, 448
transferencia de calor por convección, 404
transformada
discreta de Fourier, 471
inversa, **471**
rápida de Fourier, 473
conteo de operaciones, 475
transformada discreta del coseno, 495
bidimensional, **502**
inversa, **502**
modificada, 496, 521
inversa, **497**
unidimensional, **496**
versión 4, 520
transpuesta de una matriz, 584
tridiagonal, 562
Tukey, J., 473
Turing, A., 426

Unidades MKS, 102

Valor
propio, 30, 531, 586
complejo, 542
dominante, 539, 551
singular, 552
variable aleatoria
desviación estándar, 440
normal estándar, 438, 456
varianza, 440
varianza muestral, **448**
vector
columna, 583
de dirección, 309
del lado derecho, 79
fila, 583
ortogonal, 190
propio, 532
principal, 551
residual, 86
singular, 552
velocidad de onda, 393
vida media, 207
viga
Euler-Bernoulli, 71, 102
Timoshenko, 105
voladizo, 71
volatilidad, 465
Von Neumann, J., 432

Wiener, N., 492
Wilkinson, J., 47

ANÁLISIS NUMÉRICO

SEGUNDA EDICIÓN

La disciplina del análisis numérico está repleta de ideas útiles. Para obtener una comprensión profunda del tema, los lectores deben estudiar los grandes principios que lo permean y aprender a integrarlos a sus intereses con precisión y eficiencia.

Análisis numérico es un libro para estudiantes de ingeniería, ciencias, matemáticas e informática con conocimientos de cálculo elemental y álgebra matricial. El objetivo de este texto es enseñar a construir y explorar algoritmos para resolver problemas científicos y de ingeniería, así como para ayudar a localizar estos algoritmos en un escenario de principios poderosos y de gran alcance que, en conjunto, constituyen un campo dinámico de la investigación en la ciencia moderna numérica y computacional.

Entre las novedades que presenta esta segunda edición destacan las siguientes:

- Una importante expansión de los métodos para resolver sistemas de ecuaciones.
- La factorización de Cholesky para solucionar ecuaciones matriciales simétricas definidas y positivas.
- Un análisis del enfoque de Krylov para los grandes sistemas lineales, incluyendo el método GMRES, así como material nuevo sobre el uso de precondicionadores para problemas simétricos y no simétricos.
- La ortogonalización modificada de Gram-Schmidt y el método Levenberg-Marquardt.
- El estudio de las EDP se ha extendido a las EDP no lineales, incluyendo las ecuaciones de reacción-difusión y de formación de patrones.

El libro utiliza MATLAB para la exposición de los algoritmos y como plataforma sugerida en la realización de tareas y proyectos.

Visite el sitio web de este libro donde encontrará material adicional.

www.pearsonenespañol.com/sauer

ISBN 978-607-32-2059-0

Visítenos en:
www.pearsonenespañol.com