



A 3D rendering of a surface composed of numerous blue, glossy, rectangular blocks arranged in a grid pattern. Two large, smooth, reflective spheres, one red and one orange, are positioned on the surface. The spheres reflect the surrounding environment, creating highlights and shadows that emphasize their rounded form against the flat, geometric blocks.

Métodos numéricos y computación

Sexta Edición

Ward Cheney • David Kincaid

Fórmulas de álgebra

$$1 + r + r^2 + \cdots + r^{n-1} = \frac{r^n - 1}{r - 1}$$

$$\log_a x = (\log_a b)(\log_b x)$$

$$1 + 2 + 3 + \cdots + n = \frac{1}{2}n(n + 1)$$

$$|x| - |y| \leq |x \pm y| \leq |x| + |y|$$

$$1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{1}{6}n(n + 1)(2n + 1)$$

Desigualdad de Cauchy-Schwarz

$$\left(\sum_{i=1}^n x_i y_i \right)^2 \leq \left(\sum_{i=1}^n x_i^2 \right) \left(\sum_{i=1}^n y_i^2 \right)$$

Fórmulas de geometría

$$\text{Área del círculo: } A = \pi r^2 \quad (r = \text{radio}) \quad \text{Circunferencia del círculo: } C = 2\pi r$$

$$\text{Área del trapecio: } A = \frac{1}{2}h(a + b) \quad (h = \text{altura}; a \text{ y } b \text{ son bases paralelas})$$

$$\text{Área del triángulo: } A = \frac{1}{2}bh \quad (b = \text{base}, h = \text{altura})$$

Fórmulas de trigonometría

$$\sin^2 x + \cos^2 x = 1$$

$$\sin\left(\frac{\pi}{2} - x\right) = \cos x$$

$$1 + \tan^2 x = \sec^2 x$$

$$\cos\left(\frac{\pi}{2} - x\right) = \sin x$$

$$\sin x = 1/\csc x$$

$$\sin(x + y) = \sin x \cos y + \cos x \sin y$$

$$\cos x = 1/\sec x$$

$$\cos(x + y) = \cos x \cos y - \sin x \sin y$$

$$\tan x = 1/\cot x$$

$$\sin x + \sin y = 2 \sin\left[\frac{1}{2}(x + y)\right] \cos\left[\frac{1}{2}(x - y)\right]$$

$$\tan x = \sin x / \cos x$$

$$\cos x + \cos y = 2 \cos\left[\frac{1}{2}(x + y)\right] \cos\left[\frac{1}{2}(x - y)\right]$$

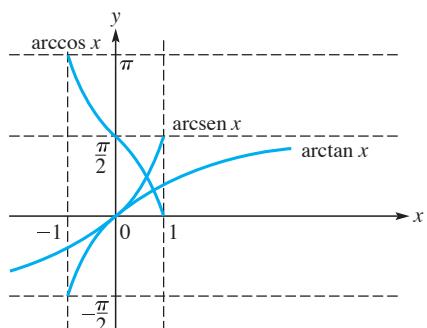
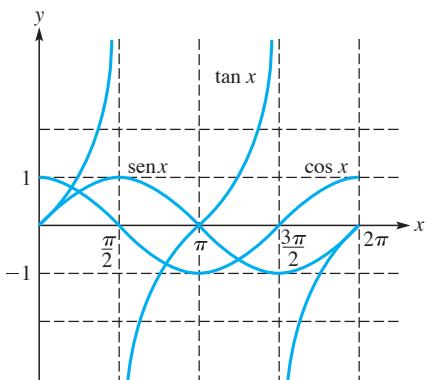
$$\sin x = -\sin(-x)$$

$$\sinh x = \frac{1}{2}(e^x - e^{-x})$$

$$\cos x = \cos(-x)$$

$$\cosh x = \frac{1}{2}(e^x + e^{-x})$$

Gráficas



Fórmulas de geometría analítica

Pendiente de una recta: $m = \frac{y_2 - y_1}{x_2 - x_1}$ (dos puntos (x_1, y_1) y (x_2, y_2))

Ecuación de una recta: $y - y_1 = m(x - x_1)$

Fórmula de una distancia: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Círculo: $(x - x_0)^2 + (y - y_0)^2 = r^2$ (r = radio, (x_0, y_0) es el centro)

Elipse: $\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$ (a y b semiejes)

Definiciones de cálculo

El enunciado de **Límite** $\lim_{x \rightarrow a} f(x) = L$ significa que para cualquier $\varepsilon > 0$, existe una $\delta > 0$ tal que $|f(x) - L| < \varepsilon$ siempre que $0 < |x - a| < \delta$.

Una función f es **continua** en x si $\lim_{h \rightarrow 0} f(x + h) = f(x)$.

Si $\lim_{h \rightarrow 0} \frac{1}{h}[f(x + h) - f(x)]$ existe, se denota por $f'(x)$ o por $\frac{d}{dx} f(x)$ y se llama la **derivada** de f en x .

Fórmulas de cálculo diferencial

$$(f \pm g)' = f' \pm g'$$

$$\frac{d}{dx} \log_a x = x^{-1} \log_a e$$

$$\frac{d}{dx} \arccot x = \frac{-1}{1 + x^2}$$

$$(fg)' = fg' + f'g$$

$$\frac{d}{dx} \sin x = \cos x$$

$$\frac{d}{dx} \operatorname{arcsec} x = \frac{1}{x\sqrt{x^2 - 1}}$$

$$\left(\frac{f}{g}\right)' = \frac{gf' - fg'}{g^2}$$

$$\frac{d}{dx} \cos x = -\sin x$$

$$\frac{d}{dx} \operatorname{arccsc} x = \frac{-1}{x\sqrt{x^2 - 1}}$$

$$(f \circ g)' = (f' \circ g)g'$$

$$\frac{d}{dx} \tan x = \sec^2 x$$

$$\frac{d}{dx} \operatorname{senh} x = \cosh x$$

$$\frac{d}{dx} x^a = a x^{a-1}$$

$$\frac{d}{dx} \cot x = -\csc^2 x$$

$$\frac{d}{dx} \cosh x = \operatorname{senh} x$$

$$\frac{d}{dx} e^x = e^x$$

$$\frac{d}{dx} \sec x = \tan x \sec x$$

$$\frac{d}{dx} \tanh x = \operatorname{sech}^2 x$$

$$\frac{d}{dx} e^{ax} = ae^{ax}$$

$$\frac{d}{dx} \csc x = -\cot x \csc x$$

$$\frac{d}{dx} \coth x = -\operatorname{csch}^2 x$$

$$\frac{d}{dx} a^x = a^x \ln a$$

$$\frac{d}{dx} \arcsen x = \frac{1}{\sqrt{1 - x^2}}$$

$$\frac{d}{dx} \operatorname{sech} x = -\operatorname{sech} x \tanh x$$

$$\frac{d}{dx} x^x = x^x(1 - \ln x)$$

$$\frac{d}{dx} \arccos x = \frac{-1}{\sqrt{1 - x^2}}$$

$$\frac{d}{dx} \operatorname{csch} x = -\operatorname{csch} x \coth x$$

$$\frac{d}{dx} \ln x = x^{-1}$$

$$\frac{d}{dx} \arctan x = \frac{1}{1 + x^2}$$



SEXTA EDICIÓN

MÉTODOS NUMÉRICOS Y COMPUTACIÓN

Ward Cheney

Universidad de Texas en Austin

David Kincaid

Universidad de Texas en Austin

Traductora

Dra. Ana Elizabeth García Hernández

Universidad La Salle Morelia

Revisor Técnico

Ing. Jesús Javier Cortés Rosas

Profesor de carrera

Jefe del Departamento de Matemáticas Avanzadas

Análisis numérico y Dibujo

Facultad de Ingeniería

Universidad Nacional Autónoma de México





**Métodos numéricos
y computación. Sexta edición**
Ward Cheney y David Kincaid

**Presidente de Cengage Learning
Latinoamérica**
Javier Arellano Gutiérrez

**Director general México
y Centroamérica**
Pedro Turbay Garrido

**Director editorial y de producción
Latinoamérica**
Raúl D. Zendejas Espejel

Coordinadora editorial
María Rosas López

Editor
Sergio R. Cervantes González

**Coordinadora de producción
editorial**
Abril Vega Orozco

Editora de producción
Gloria Luz Olguín Sarmiento

Coordinador de producción
Rafael Pérez González

Diseño de portada
Amate Diseñadores

Imagen de portada
Shutter Stock

Composición tipográfica
Amate Diseñadores

© D.R. 2011 por Cengage Learning Editores, S.A. de C.V., una Compañía de Cengage Learning, Inc.
Corporativo Santa Fe
Av. Santa Fe núm. 505, piso 12
Col. Cruz Manca, Santa Fe
C.P. 05349, México, D.F.
Cengage Learning™ es una marca registrada usada bajo permiso.

DERECHOS RESERVADOS. Ninguna parte de este trabajo amparado por la Ley Federal del Derecho de Autor, podrá ser reproducida, transmitida, almacenada o utilizada en cualquier forma o por cualquier medio, ya sea gráfico, electrónico o mecánico, incluido, pero sin limitarse a lo siguiente: fotocopiado, reproducción, escaneo, digitalización, grabación en audio, distribución en Internet, distribución en redes de información o almacenamiento y recopilación en sistemas de información a excepción de lo permitido en el Capítulo III, Artículo 27 de la Ley Federal del Derecho de Autor, sin el consentimiento por escrito de la Editorial.

Traducido del libro: Numerical Mathematics and Computing
Sixth edition
Ward Cheney and David Kincaid
Publicado en inglés por:
Brooks/Cole/Cengage Learning
ISBN-13: 978-0-495-11475-8
ISBN-10: 495-11475-8

Datos para catalogación bibliográfica:
Cheney Ward y David Kincaid
*Métodos numéricos
y computación, Sexta ed.*
ISBN-13: 978-607-481-759-1
ISBN-10: 607-481-759-6

Visite nuestro sitio en:
<http://latinoamerica.cengage.com>

Prefacio

En la preparación de la sexta edición de este libro, nos hemos ceñido al objetivo básico de las ediciones anteriores, a saber, familiarizar a los estudiantes de ciencias e ingeniería con las potencialidades de las computadoras modernas para resolver problemas numéricos que se les puedan presentar en sus profesiones. Un objetivo secundario es dar a los estudiantes la oportunidad de perfeccionar sus habilidades en programación y resolución de problemas. El objetivo final es ayudar a los estudiantes a comprender el importante tema de los *errores* que inevitablemente acompañan a la informática científica y darles métodos para la detección, predicción y control de esos errores.

Gran parte de la ciencia de hoy implica complejos cálculos que utilizan sistemas de software matemático. Los usuarios pueden tener poco conocimiento de los algoritmos numéricos utilizados en que se basan estos entornos de resolución de problemas. Mediante el estudio de métodos numéricos uno se puede volver un usuario más informado y mejor preparado para evaluar y juzgar la exactitud de los resultados. Ello implica que los estudiantes deberían estudiar los algoritmos para aprender no sólo cómo funcionan sino también en qué pueden fallar. El pensamiento crítico y el constante escepticismo son actitudes que queremos que los estudiantes adquieran. Cualquier cálculo numérico extenso, incluso cuando se realice con software de última generación, si es posible, se debe someter a una verificación independiente.

Puesto que este libro es accesible a estudiantes que no necesariamente están avanzados en su estudio formal de las matemáticas y de la informática, hemos tratado de obtener un estilo elemental de presentación. Con este fin, proporcionamos muchos ejemplos y figuras ilustrativas con fragmentos de pseudocódigo, que son descripciones informales de algoritmos de computadora.

Convencidos de que la mayoría de los estudiantes en este nivel necesitan un *repaso* de matemáticas numéricas y de computación, presentamos una gran variedad de temas, incluidos algunos más avanzados que juegan un importante papel en la computación científica actual. Recomendamos que el lector tenga al menos un año de estudio de cálculo como requisito para nuestro texto. Es útil tener conocimientos de matrices, vectores y ecuaciones diferenciales.

Características de la sexta edición

Siguiendo las sugerencias y comentarios de una docena de revisores, hemos analizado todas las secciones del libro hasta cierto punto, y se han agregado las siguientes características más nuevas e importantes:

- Hemos movido algunas secciones (especialmente los códigos de computadora) del texto a la página web de manera que estén fácilmente accesibles sin escritura tediosa. Este esfuerzo incluye todo de los códigos informáticos de Matlab, Mathematica y Maple, así como un apéndice con una visión general del software matemático disponible en el mundo de internet.
- Hemos agregado más cifras y ejemplos numéricos en todo el libro, en la creencia de que los códigos en sí, así como las ayudas visuales son útiles para todos los lectores.

- Se han agregado nuevas secciones y materiales a muchos temas, tal como el método de la falsa posición, el método del gradiente conjugado, el método de Simpson y algunos otros.
- En todas partes se presentan más ejercicios que implican aplicaciones.
- Hay citas adicionales a las referencias recientes y se han sustituido algunas referencias más antiguas.
- Hemos reorganizado los apéndices, agregamos algunos nuevos y omitimos algunos anteriores.

Sugerencias de uso

Métodos numéricos y computación, sexta edición, se puede utilizar de muchas maneras, dependiendo de la importancia que el instructor prefiera y de la inevitable limitación de tiempo. Se suministran abundantes problemas para dar más versatilidad al libro. Se dividen en dos categorías: *Problemas y Problemas de cómputo*. En la primera categoría, hay más de 800 ejercicios de análisis que requieren lápiz, papel y quizás una calculadora. En la segunda categoría, hay aproximadamente 500 problemas que implican escribir un programa y probarlo en una computadora. Se les puede pedir a los estudiantes que resuelvan algunos problemas con el uso de sistemas avanzados de software, como Matlab, Mathematica o Maple. O se les puede pedir que escriban su propio código. Con frecuencia pueden seguir un modelo o ejemplo en el libro para ayudarse en la solución de ejercicios, pero en otros casos deben proceder por su propia cuenta a partir de una descripción matemática dada en el libro o en los problemas.

En algunos de los problemas de cómputo hay algo que aprender más allá de simplemente escribir un código –una *ética*, si quiere. Puede suceder que el problema que se está resolviendo y el código dado para hacerlo de alguna manera no coincidan. Algunos de los problemas de cómputo están diseñados para ganar experiencia en el uso de cualquiera de los sistemas de software matemático, de programas precodificados o de bibliotecas de códigos como *cajas negras*.

Se vende como una publicación separada un *Student's Solution Manual*. Además, los profesores que adopten el libro pueden obtener del editor el *Instructor's Solution Manual*. Ejemplos de programas basados en el seudocódigo presentado en este libro han sido codificados en varios lenguajes de programación. Estos códigos y material adicional se encuentran disponibles en los sitios web del libro:

www.thomsonedu.com/math/cheney
www.ma.utexas.edu/CNA/NMC6/

La disposición de los capítulos refleja nuestro punto de vista de cómo se presentaría mejor el material a un nuevo estudiante del tema. Sin embargo, hay muy poca dependencia mutua entre los capítulos, por lo que el instructor puede ordenar la secuencia de la presentación de diversas maneras. La mayoría de los cursos tendrá sin duda que omitir algunas secciones y capítulos por falta de tiempo.

Nuestras propias recomendaciones para los cursos basados en este libro son las siguientes:

- Un curso de un semestre cubriría cuidadosamente los capítulos 1 a 11 (posiblemente omitiría los capítulos 5 y 8 y las secciones 4.2, 9.3, 10.3 y 11.3), seguido por una selección de material de los capítulos restantes, conforme el tiempo lo permita.
- Un repaso de un semestre rápidamente revisaría superficialmente la mayoría de los capítulos en el libro y omitiría algunas de las secciones más difíciles.
- Un curso de dos semestres cubriría cuidadosamente todos los capítulos.

Proyectos de investigación estudiantil

A lo largo del libro hay algunos problemas de cómputo designados como *Proyectos de investigación estudiantil*, que brindan a los estudiantes la oportunidad de explorar temas más allá del alcance del libro. Muchos de esos proyectos implican áreas de aplicación de los métodos numéricos. Los proyectos deben incluir programación y experimentos numéricos. Un aspecto favorable de estos trabajos es que permiten que los estudiantes elijan un tema de interés para ellos, algo que posiblemente pueda surgir en su futura profesión o en su principal área de estudio. Por ejemplo, cualquier tema sugerido en los capítulos y secciones del libro se puede tratar con mayor profundidad al consultar otros libros y referencias acerca del tema. En la preparación de este proyecto, los estudiantes deben aprender acerca del tema, buscar referencias importantes (libros y artículos de investigación), hacer los cálculos y escribir un informe que explique todo esto de una manera coherente. Los estudiantes pueden hacer uso de sistemas de software matemático como Matlab, Maple o Mathematica, o hacer su propio programa en el lenguaje que prefieran.

Reconocimientos

En la preparación de la sexta edición, nos hemos beneficiado por los consejos y sugerencias que nos han ofrecido amablemente un gran número de colegas, estudiantes y usuarios de la edición anterior.

Deseamos agradecer a los revisores que nos han proporcionado críticas detalladas de esta nueva edición: Krishan Agrawal, Thomas Boger, Charles Collins, Gentil A. Estévez, Terry Feagin, Mahadevan Ganesh, William Gearhart, Juan Gil, Xiaofan Li, Vania Mascioni, Bernard Maxum, Amar Raheja, Daniel Reynolds, Asok Sen, Ching-Kuang Shene, William Slough, Thiaf Taha, Jin Wang, Quiang Ye, Tjalling Ypma y Shangyou Zhan. En particular, Jose Flores fue de gran ayuda en la revisión del manuscrito.

Revisores de ediciones anteriores fueron Neil Berger, Jose E. Castillo, Charles Cullen, Elias Y. Deeba, F. Emad, Terry Feagin, Leslie Foster, Bob Funderlic, John Gregory, Bruce P. Hillam, Patrick Lang, Ren Chi Li, Wu Li, Edward Neuman, Roy Nicolaides, J. N. Reddy, Ralph Smart, Stephen Wirkus y Marcus Wright.

Damos las gracias a quienes nos han ayudado con sus diferentes aptitudes. Muchas personas se tomaron la molestia de escribirnos sus sugerencias y críticas de las ediciones anteriores de este libro: A. Aawwal, Nabeel S. Abo-Ghander, Krishan Agrawal, Roger Alexander, Husain Ali Al-Mohssen, Kistone Anand, Keven Anderson, Vladimir Andrijevik, Jon Ashland, Hassan Basir, Steve Batterson, Neil Berger, Adarsh Beohar, Bernard Bialecki, Jason Brazile, Keith M. Briggs, Carl de Boor, Jose E. Castillo, Ellen Chen, Edmond Chow, John Cook, Roger Crawfis, Charles Cullen, Antonella Cupillari, Jonathan Dautrich, James Arthur Davis, Tim Davis, Elias Y. Deeba, Suhrit Dey, Alan Donoho, Jason Durheim, Wayne Dymacek, Fawzi P. Emad, Paul Enigenbury, Terry Feagin, Leslie Foster, Peter Fraser, Richard Gardner, John Gregory, Katherine Hua Guo, Scott Hagerup, Kent Harris, Bruce P. Hillam, Tom Hogan, Jackie Johnson, Christopher M. Hoss, Kwang-il In, Victoria Interrante, Sadegh Jokar, Erni Jusuf, Jason Karns, Grant Keady, Jacek Kierzenka, S. A. (Seppo) Korpela, Andrew Knyazev, Gary Krenz, Jihoon Kwak, Kim Kyungjin, Minghorng Lai, Patrick Lang, Wu Li, Grace Liu, Wenguo Liu, Mark C. Malburg, P. W. Manual, Juan Meza, F. Milianazzo, Milan Miklavcic, Sue Minkoff, George Minty, Baharen Momken, Justin Montgomery, Ramon E. Moore, Aaron Naiman, Asha Nallana, Edward Neuman, Durene Ngo, Roy Nicolaides, Jeff Nunemacher, Valia Guerra Ones, Tony Praseuth, Rolfe G. Petschek, Mihaela Quirk, Helia Niroomand Rad, Jeremy Rahe, Frank Roberts, Frank Rogers, Simen Rokaas, Robert S. Raposo,

Chris C. Seib, Granville Sewell, Keh-Ming Shyue, Daniel Somerville, Nathan Smith, Mandayam Srinivas, Alexander Stromberger, Xingping Sun, Thiab Taha, Hidajaty Thajeb, Joseph Traub, Phuoc Truong, Vincent Tsao, Bi Roubolo Vona, David Wallace, Charles Walters, Kegnag Wang, Layne T. Watson, Andre Weideman, Perry Wong, Yuan Xu y Rick Zaccone.

Valiosos comentarios y sugerencias fueron hechos por nuestros colegas y amigos. En particular, David Young fue muy generoso con sugerencias para mejorar la exactitud y claridad de la exposición en ediciones anteriores. Algunas partes de las ediciones anteriores fueron escritas con gran cuidado y atención al detalle por Katy Burrell, Kata Carbone y Belinda Trevino. Aaron Naiman en el Jerusalem College of Technology ha sido especialmente útil en la preparación de la presentación gráfica para un curso basado en este libro.

Es un placer dar las gracias a quienes ayudaron con la tarea de preparar la nueva edición. El personal de Brooks/Cole y personas asociadas han sido muy comprensivos y pacientes para llevar este libro a buen término. En particular, damos las gracias a Bob Pirtle, Stacy Green, Elizabeth Rodio y Cheryll Linthicum por sus esfuerzos en favor de este proyecto. Algunas personas relacionadas con las ediciones anteriores son Seema Atwal, Craig Barth, Carol Benedict, Gary Ostedt, Jeremy Hayhurst, Janet Hill, Ragú Raghavan, Anne Seitz, Marlene Thom y Elizabeth Rammel. También damos las gracias a Merrill Peterson y Sara Planck en Matrix Productions Inc. por suministrarnos las macros de L^AT_EX y ayudarnos a dar al libro su forma final.

Agradeceríamos a los lectores que puedan comunicarse con nosotros cualquier comentario, preguntas, críticas o correcciones. Para esto, el correo electrónico es especialmente eficiente.

Ward Cheney

Departamento de Matemáticas
cheney@math.utexas.edu

David Kincaid

Departamento de Ciencias Computacionales
kincaid@cs.utexas.edu

Contenido

1 Introducción 1

1.1 Observaciones preliminares 1

- Dígitos significativos de precisión: ejemplos 3
- Errores: absoluto y relativo 5
- Exactitud y precisión 5
- Redondeo y truncamiento 6
- Multiplicación anidada 7
- Parejas de problemas fácil/difícil 9
- Primer experimento de programación 9
- Software matemático 10
- Resumen 11
- Referencias adicionales 11
- Problemas 1.1 12
- Problemas de cómputo 1.1 14

1.2 Repaso de series de Taylor 20

- Series de Taylor 20
- Algoritmo completo de Horner 23
- Teorema de Taylor en términos de $(x - c)$ 24
- Teorema del valor medio 26
- Teorema de Taylor en términos de h 26
- Series alternantes 28
- Resumen 30
- Referencias adicionales 31
- Problemas 1.2 31
- Problemas de cómputo 1.2 36

2 Representación de punto flotante y errores 43

2.1 Representación de punto flotante 43

- Representación de punto flotante normalizada 44
- Representación de punto flotante 46
- Forma de punto flotante de precisión simple 46

Forma de punto flotante de doble precisión	48
Errores de cómputo en la representación de números	50
Notación $fl(x)$ y análisis de error hacia atrás	51
Notas históricas	54
Resumen	54
Problemas 2.1	55
Problemas de cómputo 2.1	59

2.2 Pérdida de significancia 61

Dígitos significativos	61
Pérdida de significancia causada por la computación	62
Teorema de pérdida de precisión	63
Cómo evitar la pérdida de significancia en la resta	64
Reducción de rango	67
Resumen	68
Referencias adicionales	68
Problemas 2.2	68
Problemas de cómputo 2.2	71

3 Localización de raíces de ecuaciones 76

3.1 Método de bisección 76

Introducción	76
Algoritmo y seudocódigo de la bisección	78
Ejemplos	79
Análisis de convergencia	81
Método de falsa posición (<i>regula falsi</i>) y modificaciones	83
Resumen	85
Problemas 3.1	85
Problemas de cómputo 3.1	87

3.2 Método de Newton 89

Interpretaciones del método de Newton	90
Seudocódigo	92
Ilustración	92
Análisis de convergencia	93
Sistemas de ecuaciones no lineales	96
Cuencas de atracción de fractales	99
Resumen	100
Referencias adicionales	100
Problemas 3.2	101
Problemas de cómputo 3.2	105

3.3 Método de la secante 111

Algoritmo de la secante	112
Análisis de convergencia	114
Comparación de métodos	117

Esquemas híbridos	117
Iteración de punto fijo	117
Resumen	118
Referencias adicionales	119
Problemas 3.3	119
Problemas de cómputo 3.3	121

4 Interpolación y diferenciación numérica 124

4.1 Interpolación polinomial 124

Observaciones preliminares	124
Interpolación polinomial	125
Polinomio de interpolación: forma de Lagrange	126
Existencia de la interpolación de polinomios	128
Interpolación polinomial: forma de Newton	128
Forma anidada	130
Cálculo de coeficientes a_i usando diferencias divididas	131
Algoritmos y seudocódigo	136
Matriz de Vandermonde	139
Interpolación inversa	141
Interpolación polinomial con el algoritmo de Neville	142
Interpolación de funciones de dos variables	144
Resumen	145
Problemas 4.1	146
Problemas de cómputo 4.1	152

4.2 Errores en la interpolación polinomial 153

Función de Dirichlet	154
Función de Runge	154
Teoremas de errores de interpolación	156
Resumen	160
Problemas 4.2	161
Problemas de cómputo 4.2	163

4.3 Cálculo de derivadas y extrapolación de Richardson 164

Fórmulas de primera derivada mediante series de Taylor	164
Extrapolación de Richardson	166
Fórmulas de primera derivada mediante interpolación de polinomios	170
Fórmulas de segunda derivada mediante series de Taylor	173
Ruido en cálculos	174
Resumen	174
Referencias adicionales del capítulo 4	175
Problemas 4.3	175
Problemas de cómputo 4.3	178

5 Integración numérica 180**5.1 Sumas inferior y superior 180**

Integrales definidas e indefinidas 180
Sumas inferior y superior 181
Funciones integrables de Riemann 183
Ejemplos y seudocódigo 184
Resumen 187
Problemas 5.1 187
Problemas de cómputo 5.1 188

5.2 Regla del trapecio 190

Espaciado uniforme 191
Análisis de error 192
Aplicación de la fórmula de error 195
Fórmula recursiva del trapecio para subintervalos iguales 196
Integración multidimensional 198
Resumen 199
Problemas 5.2 200
Problemas de cómputo 5.2 203

5.3 Algoritmo de Romberg 204

Descripción 204
Pseudocódigo 205
Fórmula de Euler-Maclaurin 206
Extrapolación general 209
Resumen 211
Referencias adicionales 211
Problemas 5.3 212
Problemas de cómputo 5.3 214

6 Temas adicionales de integración numérica 216**6.1 Regla de Simpson y adaptable de Simpson 216**

Regla básica de Simpson 216
Regla de Simpson 219
Regla compuesta de Simpson 220
Un esquema adaptable de Simpson 221
Ejemplo del uso del procedimiento adaptable de Simpson 224
Reglas de Newton-Cotes 225
Resumen 226
Problemas 6.1 227
Problemas de cómputo 6.1 229

6.2 Fórmulas de cuadratura gaussiana 230

- Descripción 230
 - Cambio de intervalos 231
 - Nodos gaussianos y pesos 232
 - Polinomios de Legendre 234
 - Integrales con singularidades 237
 - Resumen 237
 - Referencias adicionales 239
 - Problemas 6.2 239
 - Problemas de cómputo 6.2 241
-

7 Sistemas de ecuaciones lineales 245**7.1 Eliminación gaussiana simple 245**

- Un gran ejemplo numérico 247
- Algoritmo 248
- Seudocódigo 250
- Prueba del seudocódigo 253
- Vectores residual y de error 254
- Resumen 255
- Problemas 7.1 255
- Problemas de cómputo 7.1 257

7.2 Eliminación gaussiana con pivoteo escalado parcial 259

- La eliminación gaussiana simple puede fallar 259
- Pivoteo parcial y pivoteo completo parcial 261
- Eliminación gaussiana con pivoteo escalado parcial 262
- Un gran ejemplo numérico 265
- Seudocódigo 266
- Conteo de operaciones largas 269
- Estabilidad numérica 271
- Escalamiento 271
- Resumen 271
- Problemas 7.2 272
- Problemas de cómputo 7.2 276

7.3 Sistemas tridiagonales y en banda 280

- Sistemas tridiagonales 281
- Dominio estrictamente diagonal 282
- Sistemas pentadiagonales 283
- Sistemas pentadiagonales de bloque 285
- Resumen 286
- Referencias adicionales 287
- Problemas 7.3 287
- Problemas de cómputo 7.3 288

8 Temas adicionales referentes a sistemas de ecuaciones lineales 293

8.1 Factorizaciones matriciales 293

- Ejemplo numérico 294
- Deducción formal 296
- Seudocódigo 300
- Resolución de sistemas lineales usando factorización LU 300
- Factorización LDL^T 302
- Factorización de Cholesky 305
- Múltiples lados derechos 306
- Cálculo A^{-1} 307
- Ejemplo con uso de paquetes de software 307
- Resumen 309
- Problemas 8.1 311
- Problemas de cómputo 8.1 316

8.2 Soluciones iterativas de sistemas lineales 319

- Normas de vector y matriz 319
- Número de condición y mal condicionado 321
- Métodos iterativos básicos 322
- Seudocódigo 327
- Teoremas de convergencia 328
- Formulación matricial 331
- Otra visión de la sobrerelajación 332
- Método del gradiente conjugado 332
- Resumen 335
- Problemas 8.2 337
- Problemas de cómputo 8.2 339

8.3 Valores propios y vectores propios 342

- Cálculo de valores propios y vectores propios 343
- Software matemático 344
- Propiedades de los valores propios 345
- Teorema de Gershgorin 347
- Descomposición en valor singular 348
- Ejemplos numéricos de descomposición en valor singular 351
- Aplicación: ecuaciones diferenciales lineales 353
- Aplicación: un problema de vibración 354
- Resumen 355
- Problemas 8.3 356
- Problemas de cómputo 8.3 358

8.4 Método de potencias 360

- Algoritmos del método de potencias 361

Aceleración de Aitken	363
Método de potencias inverso	364
Ejemplos con software: método de potencias inverso	365
Método de potencias (inverso) desplazado	365
Ejemplo: método de potencias inverso desplazado	366
Resumen	366
Referencias adicionales	367
Problemas 8.4	367
Problemas de cómputo 8.4	368

9 Aproximación por funciones spline

371

9.1 Splines de primer y segundo grado 371

Spline de primer grado	372
Módulo de continuidad	374
Splines de segundo grado	376
Interpolación del spline cuadrático $Q(x)$	376
Spline cuadrático de Subbotin	378
Resumen	380
Problemas 9.1	381
Problemas de cómputo 9.1	384

9.2 Splines cúbicos naturales 385

Introducción	385
Spline cúbico natural	386
Algoritmo para el spline cúbico natural	388
Seudocódigo para splines cúbicos naturales	392
Uso de seudocódigo para interpolar y ajustar curvas	393
Curvas espaciales	394
Propiedad de suavidad	396
Resumen	398
Problemas 9.2	399
Problemas de cómputo 9.2	403

9.3 Splines B: interpolación y aproximación 404

Interpolación y aproximación con splines B	410
Seudocódigo y ejemplo de un ajuste de curva	412
Proceso de Schoenberg	414
Seudocódigo	414
Curvas de Bézier	416
Resumen	418
Referencias adicionales	419
Problemas 9.3	420
Problemas de cómputo 9.3	423

10 Ecuaciones diferenciales ordinarias 426

10.1 Métodos de series de Taylor 426

- Problema con valor inicial: solución analítica contra numérica 426
- Ejemplo de un problema práctico 428
- Resolución de ecuaciones diferenciales e integración 428
- Campos vectoriales 429
- Métodos de series de Taylor 431
- Seudocódigo del método de Euler 432
- Método de la serie de Taylor de orden superior 433
- Tipos de errores 435
- Método de la serie de Taylor usando cálculos simbólicos 435
- Resumen 435
- Problemas 10.1 436
- Problemas de cómputo 10.1 438

10.2 Métodos de Runge-Kutta 439

- Serie de Taylor para $f(x, y)$ 440
- Método de Runge-Kutta de orden 2 441
- Método de Runge-Kutta de orden 4 442
- Seudocódigo 443
- Resumen 444
- Problemas 10.2 445
- Problemas de cómputo 10.2 447

10.3 Estabilidad y adaptación de los métodos de Runge-Kutta y de multipaso 450

- Un método adaptado de Runge-Kutta-Fehlberg 450
- Un ejemplo industrial 454
- Fórmulas de Adams-Bashforth-Moulton 455
- Análisis de estabilidad 456
- Resumen 459
- Referencias adicionales 460
- Problemas 10.3 460
- Problemas de cómputo 10.3 461

11 Sistemas de ecuaciones diferenciales ordinarias 465

11.1 Métodos para sistemas de primer orden 465

- Sistemas desacoplados y acoplados 465
- Método de series de Taylor 466
- Notación vectorial 467
- Sistemas de EDO 468

Método de series de Taylor: notación vectorial	468
Método de Runge-Kutta	469
EDO autónoma	471
Resumen	473
Problemas 11.1	474
Problemas de cómputo 11.1	475

11.2 Ecuaciones de orden superior y sistemas 477

Ecuaciones diferenciales de orden superior	477
Sistemas de ecuaciones diferenciales de orden superior	479
Sistemas de EDO autónomas	479
Resumen	480
Problemas 11.2	480
Problemas de cómputo 11.2	482

11.3 Métodos de Adams–Bashforth–Moulton 483

Un esquema predictor–corrector	483
Seudocódigo	484
Un esquema adaptado	488
Un ejemplo de ingeniería	488
Algunas observaciones acerca de las ecuaciones rígidas	489
Resumen	491
Referencias adicionales	492
Problemas 11.3	492
Problemas de cómputo 11.3	492

12 Suavizado de datos y el método de mínimos cuadrados

495

12.1 Método de mínimos cuadrados 495

Recta de mínimos cuadrados	495
Ejemplo lineal	498
Ejemplo no polinomial	499
Funciones base $\{g_0, g_1, \dots, g_n\}$	500
Resumen	501
Problemas 12.1	502
Problemas de cómputo 12.1	517

12.2 Sistemas ortogonales y polinomios de Chebyshev 505

Funciones base ortonormales $\{g_0, g_1, \dots, g_n\}$	505
Diseño de algoritmo	508
Suavizado de datos: regresión polinomial	510
Resumen	515
Problemas 12.2	516
Problemas de cómputo 12.2	517

12.3 Otros ejemplos del principio de mínimos cuadrados 518

Uso de una función de peso $w(x)$	519
-----------------------------------	-----

Ejemplo no lineal	520
Ejemplo lineal y no lineal	521
Detalles adicionales en SVD	522
Uso de la descomposición de valor singular	524
Resumen	527
Referencias adicionales	527
Problemas 12.3	527
Problemas de cómputo 12.3	530

13 Métodos de Monte Carlo y simulación 532

13.1 Números aleatorios 532

Algoritmos y generadores de números aleatorios	533
Ejemplos	535
Uso del seudocódigo <i>Aleatorio</i>	537
Resumen	541
Problemas 13.1	541
Problemas de cómputo 13.1	542

13.2 Cálculo de áreas y volúmenes mediante técnicas de Monte Carlo 544

Integración numérica	544
Ejemplo y seudocódigo	545
Cálculo de volúmenes	547
Ejemplo del barquillo de helado	548
Resumen	549
Problemas 13.2	549
Problemas de cómputo 13.2	549

13.3 Simulación 552

Problema del dado cargado	552
Problema del cumpleaños	553
Problema de la aguja de Buffon	555
Problema de dos dados	556
Escudo de neutrones	557
Resumen	558
Referencias adicionales	558
Problemas de cómputo 13.3	559

14 Problemas con valores en la frontera para ecuaciones diferenciales ordinarias 563

14.1 Método de disparo 563

Algoritmo del método de disparo	565
Modificaciones y refinamientos	567

Resumen	567
Problemas 14.1	568
Problemas de cómputo 14.1	570

14.2 Un método de discretización 570

Aproximaciones por diferencias finitas	570
El caso lineal	571
Seudocódigo y ejemplo numérico	572
Método de disparo en el caso lineal	574
Seudocódigo y ejemplo numérico	575
Resumen	577
Referencias adicionales	578
Problemas 14.2	578
Problemas de cómputo 14.2	580

15 Ecuaciones diferenciales parciales 582

15.1 Problemas parabólicos 582

Algunas ecuaciones diferenciales parciales de problemas de aplicación	582
Problema modelo de la ecuación de calor	585
Método de diferencias finitas	585
Seudocódigo para el método explícito	587
Método de Crank-Nicolson	588
Seudocódigo para el método de Crank-Nicolson	589
Versión alternativa del método de Crank-Nicolson	590
Estabilidad	591
Resumen	593
Problemas 15.1	594
Problemas de cómputo 15.1	596

15.2 Problemas hiperbólicos 596

Problema modelo de la ecuación de onda	596
Solución analítica	597
Solución numérica	598
Seudocódigo	600
Ecuación de advección	601
Método de Lax	602
Método contra el viento	602
Método de Lax-Wendroff	602
Resumen	603
Problemas 15.2	604
Problemas de cómputo 15.2	604

15.3 Problemas elípticos 605

Problema modelo de la ecuación de Helmholtz	605
Método de diferencias finitas	606
Método iterativo de Gauss-Seidel	610

Ejemplo numérico y seudocódigo	610
Métodos de elemento finito	613
Más de elementos finitos	617
Resumen	619
Referencias adicionales	620
Problemas 15.3	620
Problemas de cómputo 15.3	622

16 Minimización de funciones 625

16.1 Caso de una variable 625

Problemas de minimización con y sin restricciones	625
Caso de una variable	626
Funciones unimodales F	627
Algoritmo de búsqueda de Fibonacci	628
Algoritmo de búsqueda de la sección áurea	631
Algoritmo de interpolación cuadrática	633
Resumen	635
Problemas 16.1	635
Problemas de cómputo 16.1	637

16.2 Caso de variables múltiples 639

Series de Taylor para F : vector gradiente y matriz hessiana	640
Forma alternativa de la serie de Taylor	641
Procedimiento de máxima pendiente	643
Diagramas de contorno	644
Algoritmos más avanzados	644
Mínimo, máximo y puntos silla	646
Matriz positiva definida	647
Métodos de quasiNewton	647
Algoritmo de Nelder-Mead	647
Método de recocido simulado	648
Resumen	650
Referencias adicionales	651
Problemas 16.2	651
Problemas de cómputo 16.2	654

17 Programación lineal 657

17.1 Formas estándar y dualidad 657

Primera forma primal	657
Ejemplo numérico	658

Transformación de problemas en la primera forma primal	660
Problema dual	661
Segunda forma primal	663
Resumen	664
Problemas 17.1	665
Problemas de cómputo 17.1	669

17.2 Método simplex 670

Vértices en K y columnas de A linealmente independientes	671
Método simplex	672
Resumen	674
Problemas 17.2	674
Problemas de cómputo 17.2	675

17.3 Solución aproximada de sistemas lineales inconsistentes 675

Problema ℓ_1	676
Problema ℓ_∞	678
Resumen	680
Referencias adicionales	682
Problemas 17.3	682
Problemas de cómputo 17.3	682

Apéndice A Asesoramiento de buenas prácticas en programación 684

A.1 Sugerencias de programación	684
Casos prácticos	687
Desarrollo de software matemático	691

Apéndice B Representación de números en diferentes bases 692

B.1 Representación de números en diferentes bases	692
Números de base β	693
Conversión de partes enteras	693
Conversión de partes fraccionarias	695
Base de conversión $10 \leftrightarrow 8 \leftrightarrow 2$	696
Base 16	698
Más ejemplos	698
Resumen	699
Problemas B.1	699
Problemas de cómputo B.1	701

Apéndice C Detalles adicionales de la aritmética de punto flotante del IEEE 703

C.1 Más de la aritmética estándar de punto flotante del IEEE	703
--	-----

Apéndice D Álgebra lineal: conceptos y notación 706

D.1 Conceptos elementales	706
Vectores	706
Matrices	708

Producto matriz-vector 711
Producto matricial 711
Otros conceptos 713
Regla de Cramer 715

D.2 Espacios vectoriales abstractos 716

Subespacios 717
Independencia lineal 717
Bases 718
Transformaciones lineales 718
Valores propios y vectores propios 719
Cambio de base y similaridad 719
Matrices ortogonales y teorema espectral
Normas 721
Proceso de Gram-Schmidt 722

Resuestas a los problemas seleccionados 724

Bibliografía 745

Índice 754

Introducción

La serie de Taylor para el logaritmo natural $\ln(1 + x)$ es

$$\ln 2 = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \frac{1}{6} + \frac{1}{7} - \frac{1}{8} + \dots$$

Sumando todos los ocho términos que se muestran, obtenemos $\ln 2 \approx 0.63452^*$, que es una pobre aproximación a $\ln 2 = 0.69315\dots$. Por otra parte, la serie de Taylor para $\ln[(1 + x)/(1 - x)]$ nos da (con $x = \frac{1}{3}$)

$$\ln 2 = 2 \left(3^{-1} + \frac{3^{-3}}{3} + \frac{3^{-5}}{5} + \frac{3^{-7}}{7} + \dots \right)$$

Al sumar los cuatro términos que se muestran entre los paréntesis y luego multiplicar el resultado por 2, obtenemos $\ln 2 \approx 0.69313$. Esto muestra el hecho de que la rápida convergencia de una serie de Taylor se puede esperar cerca del punto de expansión pero no en puntos lejanos. Evaluar la serie $\ln[(1 + x) / (1 - x)]$ en $x = \frac{1}{3}$ es un mecanismo para evaluar $\ln 2$ cerca del punto de expansión. También es un ejemplo en el que las propiedades de una función se pueden aprovechar para obtener una serie que converge más rápidamente. Ejemplos como éste serán claros después de que el lector haya estudiado la sección 1.2. Las series de Taylor y el teorema de Taylor son dos de los temas principales que analizaremos en este capítulo, pues se presentan a menudo en gran parte del análisis numérico.

1.1 Observaciones preliminares

El objetivo de este libro es ayudar al lector a entender algunos de los muchos métodos que hay para resolver problemas científicos en una computadora moderna. A propósito nos hemos limitado a los típicos problemas que surgen en ciencia, ingeniería y tecnología. Así, no consideramos los problemas de contabilidad, modelado en ciencias sociales, recuperación de información, inteligencia artificial, etc.

*El símbolo \approx significa “aproximadamente igual a”.

Por lo general, nuestro tratamiento de los problemas no se iniciará en la fuente, lo que nos llevaría muy dentro de áreas como la física, la ingeniería y la química. En su lugar, consideraremos los problemas después de que se han moldeado en algunas formas matemáticas comunes. Por tanto, se pide al lector que dé por cierta la afirmación de que los temas elegidos son verdaderamente importantes en computación científica.

Para analizar muchos temas, debemos tratar a algunos de una manera superficial. Pero esperamos que el lector, tendrá una buena visión general del tema y que, por tanto, estará mejor preparado para un estudio más profundo del análisis numérico.

Por cada tema principal, listamos buenas fuentes actuales para obtener más información. En cualquier situación real de computación, se debe analizar con mucho cuidado la elección del método que se empleará. Aunque la mayoría de los procedimientos que aquí se presentan son útiles e importantes, pueden no ser los óptimos para un problema particular. Para elegir entre los métodos disponibles para resolver un problema, el analista o el programador debe consultar referencias recientes.

Familiarizarse con los métodos numéricos básicos, sin darse cuenta de sus limitaciones, sería una locura. Los cálculos numéricos están casi invariablemente contaminados con errores y es importante entender la fuente, propagación, magnitud y tasa de crecimiento de estos errores. Los métodos numéricos que hacen aproximaciones y estimaciones de errores son más valiosos que los que sólo dan respuestas aproximadas. Aunque no podemos sino estar impresionados con la velocidad y la precisión de la computadora moderna, debemos moderar nuestra admiración con medidas generosas de escepticismo. Como el eminentе analista numérico Carl-Erik Fröberg una vez comentó:

¡Nunca en la historia de la humanidad había sido posible producir tantas respuestas incorrectas y tan rápidamente!

Por ello, uno de nuestros objetivos es ayudar al lector a llegar a este estado de escepticismo, armados con métodos para detectar, estimar y controlar los errores.

Se espera que el lector se familiarice con los fundamentos de la programación. Los algoritmos se presentan como seudocódigo, y no se adopta un lenguaje de programación particular.

Algunos de los principales problemas relacionados con los métodos numéricos son el tipo de errores numéricos, la propagación de errores y la eficiencia de los cálculos implicados, así como el número de operaciones y su posible reducción.

Muchos estudiantes tienen calculadoras graficadoras y acceso a software de sistemas matemáticos que pueden dar soluciones a complicados problemas numéricos con una dificultad mínima. El propósito de un curso de métodos numéricos es examinar las técnicas algorítmicas subyacentes para que los estudiantes aprendan cómo el software o la calculadora encuentran la respuesta. De esta manera, tendrían una mejor comprensión de los límites inherentes de la exactitud que se debe prever al trabajar con dichos sistemas.

Una de las estrategias fundamentales detrás de muchos métodos numéricos es el remplazo de un problema difícil por una serie de otros más simples. Al realizar un proceso iterativo, las soluciones de los problemas más simples se pueden juntar para obtener la solución del difícil problema original. Esta estrategia tiene éxito en la búsqueda de raíces de las funciones (capítulo 3), interpolación (capítulo 4), integración numérica (capítulos 5 y 6) y solución de sistemas lineales (capítulos 7 y 8).

Los estudiantes que se especializan en matemáticas y ciencias computacionales, así como los de ingeniería y otras ciencias están generalmente muy conscientes de que se necesitan métodos numéricos para resolver los problemas que con frecuencia enfrentan. Puede que no se reconozca que la computación científica es muy importante para resolver problemas que provienen de otros campos distintos a la ingeniería y a la ciencia, como los económicos. Por ejemplo, el encontrar raíces de funciones puede surgir en problemas que usan fórmulas para calcular préstamos, intereses y calendario de pagos. También, problemas en áreas tales como las relacionadas con el mercado de

valores pueden requerir soluciones de mínimos cuadrados (Capítulo 12). De hecho, el campo de finanzas computacionales requiere de la solución de problemas matemáticos muy complejos que utilizan una gran cantidad de poder de cómputo. Los modelos económicos requieren normalmente del análisis de sistemas de ecuaciones lineales con miles de incógnitas.

Dígitos significativos de precisión: ejemplos

Los **dígitos significativos** son dígitos que empiezan con el dígito *distinto de cero* del extremo izquierdo y terminan con el dígito *correcto* del extremo derecho, incluyendo los ceros finales que son exactos.

- EJEMPLO 1** En una sala de máquinas, un técnico corta una lámina metálica rectangular de 2 por 3 metros en dos piezas triangulares iguales. ¿Cuánto mide la diagonal de cada triángulo? ¿Estas piezas se pueden modificar un poco para que las diagonales midan exactamente 3.6 metros?

- Solución** Puesto que la pieza es rectangular, se puede utilizar el teorema de Pitágoras. Así, para calcular la diagonal, escribimos $2^2 + 3^2 = d^2$, donde d es la diagonal. Se tiene que

$$d = \sqrt{4 + 9} = \sqrt{13} = 3.60555\ 1275$$

Esta última cifra se obtiene usando una calculadora manual. La exactitud de d como está dada se puede verificar calculando $(3.60555\ 1275) * (3.60555\ 1275) = 13$. ¿Se debe tomar en serio este valor para la diagonal, d ? En realidad no. Para comenzar, no se puede esperar que las dimensiones dadas del rectángulo sean precisamente 2 y 3. Si las dimensiones son exactas a un milímetro, pueden ser de 2.001 y 3.001. Usando de nuevo el teorema de Pitágoras, se encuentra que la diagonal puede medir

$$d = \sqrt{2.001^2 + 3.001^2} = \sqrt{4.00400\ 1 + 9.00600\ 1} = \sqrt{13.01002} \approx 3.6069$$

Un razonamiento similar indica que d puede ser tan pequeña como 3.6042. Ambos casos *están mal*. Podemos concluir que

$$3.6042 \leq d \leq 3.6069$$

No se puede pedir mayor exactitud para la diagonal, d .

Si queremos que la diagonal sea exactamente 3.6, es preciso que

$$(3 - c)^2 + (2 - c)^2 = 3.6^2$$

Por simplicidad, le restamos a cada lado la misma cantidad, lo que nos conduce a

$$c^2 - 5c + 0.02 = 0$$

Usando la fórmula cuadrática, obtenemos la raíz más pequeña

$$c = 2.5 - \sqrt{6.23} \approx 0.00400$$

Cortando 4 milímetros de los dos lados perpendiculares quedan piezas triangulares de lados 1.996 por 2.996 metros. Comprobando, obtenemos $(1.996)^2 + (2.996)^2 \approx 3.6^2$. ■

Para mostrar el efecto del número de dígitos significativos usados en un cálculo, considere el problema de resolver un sistema de ecuaciones lineales.

EJEMPLO 2 Vamos a concentrarnos en resolver este sistema de ecuaciones lineales con dos variables para la variable y

$$\begin{cases} 0.1036x + 0.2122y = 0.7381 \\ 0.2081x + 0.4247y = 0.9327 \end{cases} \quad (1)$$

Primero, hacemos los cálculos con sólo tres dígitos significativos de exactitud. Segundo, repetimos todo con cuatro dígitos significativos. Por último, usamos diez dígitos significativos.

Solución En la primera parte, redondeamos todos los números del problema original a tres dígitos y redondeamos todos los cálculos, conservando sólo tres dígitos significativos. Multiplicando por α a la primera ecuación y restándola de la segunda ecuación para eliminar el término en x en la segunda ecuación. El multiplicador es $\alpha \approx 0.208/0.104 \approx 2.00$. Por tanto, en la segunda ecuación, el nuevo coeficiente del término x es $0.208 - (2.00)(0.104) \approx 0.208 - 0.208 = 0$ y el nuevo coeficiente del término y es $0.425 - (2.00)(0.212) \approx 0.425 - 0.424 = 0.001$. El miembro derecho queda de esta manera: $0.933 - (2.00)(0.738) = 0.933 - 1.48 = -0.547$. Por tanto, encontramos que $y = -0.547/(0.001) \approx -547$.

Decidimos conservar cuatro dígitos significativos en todo y repetir los cálculos. Ahora el multiplicador es $\alpha = 0.2081/0.1036 \approx 2.009$. En la segunda ecuación, el nuevo coeficiente del término x es $0.2081 - (2.009)(0.1036) \approx 0.2081 - 0.2081 = 0$; a la vez, el nuevo coeficiente del término y es $0.4247 - (2.009)(0.2122) \approx 0.4247 - 0.4263 = -0.001600$, y por consiguiente el nuevo miembro derecho es $0.9327 - (2.009)(0.7381) \approx 0.9327 - 1.483 \approx -0.5503$. Por tanto, encontramos que $y = -0.5503/(-0.001600) \approx 343.9$. ¡Estamos impactados al encontrar que la respuesta ha cambiado de -547 a 343.9 , que es una enorme diferencia!

De hecho, si repetimos este proceso y lo realizamos con diez dígitos decimales significativos, encontramos que aún 343.9 no es exacto, ya que obtenemos 356.2907199 . La lección aprendida en este ejemplo es pensar que los datos para ser exactos se deben manejar con toda precisión y *no* se debe redondear antes de realizar los cálculos. ■

En la mayoría de las computadoras, las operaciones aritméticas se realizan en un acumulador de doble longitud que tiene el doble de precisión de las cantidades almacenadas. Sin embargo, aún éste no puede evitar una pérdida de exactitud. La pérdida de exactitud puede ocurrir de muchas formas tal como de los errores de redondeo y de la resta de números casi iguales. Analizaremos la pérdida de precisión en el capítulo 2, y la solución de sistemas lineales en el capítulo 7.

En la figura 1.1 se muestra una situación geométrica que puede suceder al resolver dos ecuaciones con dos incógnitas. El punto de intersección de las dos rectas es la solución exacta. Como se muestra con las rectas punteadas, puede haber un grado de incertidumbre de los errores en las mediciones o errores de redondeo. Así, en lugar de tener un solo punto definido, puede haber una pequeña área trapezoidal que contiene varias soluciones posibles. Sin embargo, si las dos rectas son casi paralelas, entonces

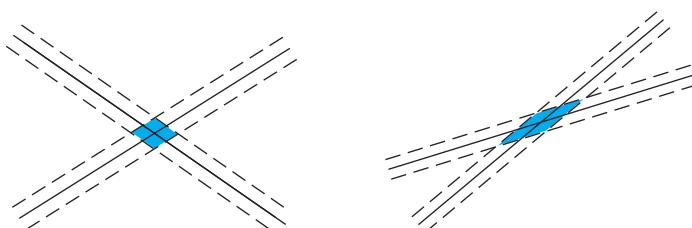


FIGURA 1.1
En dos dimensiones, sistemas lineales bien condicionados y mal condicionados

esta área de posibles soluciones puede aumentar considerablemente! Esto está relacionado con los sistemas de ecuaciones lineales biencondicionados y malcondicionados, que se analizan con más detalle en el capítulo 8.

Errores: absoluto y relativo

Suponga que α y β son dos números, de los cuales uno se considera una aproximación del otro. El **error** de β como una aproximación a α es $|\alpha - \beta|$; lo que significa que el error es igual al valor exacto menos el valor aproximado. El **error absoluto** de β como aproximación a α es $|\alpha - \beta|$. El **error relativo** de β como aproximación a α es $|\alpha - \beta|/|\alpha|$. Observe que al calcular el error absoluto, los papeles de α y β son los mismos, mientras que en el cálculo del error relativo, es fundamental distinguir uno de los dos números como correcto. (Observe que el error relativo está indefinido en el caso $\alpha = 0$.) Por razones prácticas, el error relativo con frecuencia es más importante que el error absoluto. Por ejemplo, si $\alpha_1 = 1.333$, $\beta_1 = 1.334$ y $\alpha_2 = 0.001$, $\beta_2 = 0.002$, entonces el error absoluto de β_i como una aproximación a α_i es el mismo en ambos casos —a saber, 10^{-3} . Sin embargo, los errores relativos son $\frac{3}{4} \times 10^{-3}$ y 1, respectivamente. El error relativo indica claramente que β_1 es una buena aproximación a α_1 , pero que β_2 es una aproximación pobre a α_2 . En resumen, tenemos que

$$\text{error absoluto} = |\text{valor exacto} - \text{valor aproximado}|$$

$$\text{error relativo} = \frac{|\text{valor exacto} - \text{valor aproximado}|}{|\text{valor exacto}|}$$

Aquí el valor exacto es el valor verdadero. Una forma útil de expresar el error absoluto y el error relativo es quitar los signos de valor absoluto y escribir

$$\begin{aligned} (\text{error relativo}) (\text{valor exacto}) &= \text{valor exacto} - \text{valor aproximado} \\ \text{valor aproximado} &= (\text{valor exacto}) [1 + (\text{error relativo})] \end{aligned}$$

Así, el error relativo está relacionado con el valor aproximado más que con el valor exacto, ya que el valor verdadero puede no ser conocido.

EJEMPLO 3 Considere $x = 0.00347$ redondeado a $\tilde{x} = 0.0035$ y $y = 30.158$ redondeado a $\tilde{y} = 30.16$. En cada caso, ¿cuáles son el número de dígitos significativos, errores absolutos y errores relativos? Interprete los resultados.

Solución Caso 1. $\tilde{x} = 0.35 \times 10^{-2}$ tiene dos dígitos significativos, error absoluto 0.3×10^{-4} , y error relativo 0.865×10^{-2} . Caso 2. $\tilde{y} = 0.3016 \times 10^{-2}$ tiene cuatro dígitos significativos, error absoluto 0.2×10^{-2} , y error relativo 0.66×10^{-4} . Claramente, el error relativo es una mejor indicación del número de dígitos significativos que el error absoluto. ■

Exactitud y precisión

Exactitud a n cifras decimales significa que puede confiar en n dígitos a la derecha del lugar decimal. **Exactitud a n dígitos significativos** significa que puede confiar en un total de n dígitos que sean importantes empezando con el dígito distinto de cero del extremo izquierdo.

Suponga que usa una regla graduada en milímetros para medir longitudes. Las medidas serán exactas a un milímetro, o 0.001 m, que tiene tres cifras decimales escrita en metros. Una medida como 12.345 m tendrá una exactitud de tres cifras decimales. Una medida como 12.3456789 m no tendría sentido, ya que la regla sólo tiene tres cifras decimales y ésta será de 12.345 m o 12.346 m.

Si la medida 12.345 m tiene cinco dígitos confiables, entonces tiene una exactitud de cinco números significativos. Por otra parte, una medida como 0.076 m tiene sólo dos números significativos.

Cuando se usa una calculadora o computadora en un experimento de laboratorio, se puede tener la falsa sensación de tener mayor precisión que la garantizada por los datos. Por ejemplo, el resultado

$$(1.2) + (3.45) = 4.65$$

en realidad tiene una exactitud de sólo dos dígitos significativos, ya que el segundo dígito en 1.2 puede ser el efecto de redondear hacia abajo 1.24 o redondear hacia arriba 1.16 con dos números significativos. Entonces el lado izquierdo podría ser tan grande como

$$(1.249) + (3.454) = (4.703)$$

o tan pequeño como

$$(1.16) + (3.449) = (4.609)$$

¡Realmente sólo hay dos cifras decimales significativas en la respuesta! Al sumar y restar números, la exactitud del resultado es igual a la del número más pequeño de dígitos significativos usado en cualquier paso del cálculo. En el ejemplo anterior, el término 1.2 tiene dos dígitos significativos; por tanto, el cálculo final tiene una incertidumbre en el tercer dígito.

En la multiplicación y división de números, los resultados pueden ser aún más engañosos. Por ejemplo, realice estas operaciones con una calculadora: $(1.23)(4.5) = 5.535$ y $(1.23)/(4.5) = 0.27333\ 3333$. Piensa que hay cuatro y nueve dígitos significativos en los resultados, pero ¡realmente sólo hay dos! Como una regla práctica, se deberán conservar tantos dígitos significativos en una secuencia de cálculos como los que hay en el número de menor exactitud implicado en ellos.

Redondeo y truncamiento

El **redondeo** reduce el número de dígitos significativos en un número. El resultado del redondeo es un número de magnitud similar que es un número más *corto* porque tiene menos dígitos diferentes de cero. Hay varias reglas ligeramente diferentes para redondear. El método de **redondeo parejo** también se conoce como el *redondeo estadístico* o *redondeo del banquero*. Lo estudiaremos a continuación. Para un gran conjunto de datos, la regla de redondeo parejo tiende a reducir el error total de redondeo con (en promedio) una parte igual de números redondeados hacia arriba como de redondeados hacia abajo.

Decimos que un número x está **truncado a n dígitos** o números cuando todos los dígitos que siguen al enésimo dígito son descartados y ninguno de los n dígitos restantes se cambia. Por el contrario, x está **redondeado a n dígitos** o números cuando x se remplaza por un n -dígiro número que se aproxima a x con un error mínimo. La pregunta de redondear hacia arriba o hacia abajo un $(n+1)$ -dígito decimal que termina con un 5 es mejor manejado al seleccionar siempre el redondeo de un número con n -dígitos con un enésimo dígito *par*. En principio esto puede parecer extraño, pero excepcionalmente, en esencia las computadoras redondean a decimales cuando usan aritmética de punto flotante. (Este es un tema que se analiza en el capítulo 2.)

Por ejemplo, los resultados de redondear algunos números de tres decimales a dos dígitos son $0.217 \approx 0.22$, $0.365 \approx 0.36$, $0.475 \approx 0.48$ y $0.592 \approx 0.59$, mientras que el truncamiento de ellos da $0.217 \approx 0.21$, $0.365 \approx 0.36$, $0.475 \approx 0.47$ y $0.592 \approx 0.59$. En la computadora, el usuario algunas veces tiene la opción de tener todas las operaciones aritméticas hechas ya sea por truncamiento o redondeo. Por supuesto, se prefiere generalmente el último.

Multiplicación anidada

Comenzaremos con algunas observaciones de la evaluación de un polinomio eficientemente y del redondeo y truncamiento de números reales. Para evaluar el polinomio

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n \quad (2)$$

agrupamos los términos en una **multiplicación anidada**:

$$p(x) = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + x(a_n)) \cdots))$$

El seudocódigo[‡] que evalúa a $p(x)$ inicia con el paréntesis más interno y trabaja hacia afuera. Esto puede escribirse como

```
integer i, n;  real p, x;  real array (ai)0:n
p ← an
for i = n - 1 to 0 do
    p ← ai + xp
end for
```

Aquí suponemos que se han asignado valores numéricos a la variable entera n , la variable real x , así como los coeficientes a_1, a_2, \dots, a_n , que se almacenan en un arreglo lineal real. (Usamos puntos y comas entre estos enunciados de declaración para ahorrar espacio). La flecha que apunta hacia la izquierda (\leftarrow) significa que el valor de la derecha está almacenado en la posición indicada en la izquierda (es decir, “sobreescribe” de derecha a izquierda). El ciclo for con índice i corre hacia atrás, tomando los valores $n - 1, n - 2, \dots, 0$. El valor final de p es el valor del polinomio en x . Este procedimiento de multiplicación anidada también se conoce como **algoritmo de Horner o división sintética**.

En el seudocódigo anterior hay exactamente una suma y una multiplicación cada vez que se recorre el ciclo. Por consiguiente, el algoritmo de Horner puede evaluar un polinomio con sólo n sumas y n multiplicaciones. Este es el mínimo número de operaciones posible. Un método simple de evaluar un polinomio requeriría muchas más operaciones. Por ejemplo, $p(x) = 5 + 3x - 7x^2 + 2x^3$ se calcularía como $p(x) = 5 + x(3 + x(-7 + x(2)))$ para un valor dado de x . ¡Hemos evitado todas las operaciones de elevar un número a una potencia usando la multiplicación anidada!

El polinomio en la ecuación (2) se puede reescribir en una forma alternativa utilizando los símbolos matemáticos para la suma \sum y el producto \prod , a saber,

$$p(x) = \sum_{i=0}^n a_i x^i = \sum_{i=0}^n \left(a_i \prod_{j=1}^i x \right)$$

[‡] Un seudocódigo es una descripción compacta e informal de un algoritmo que usa las convecciones de un lenguaje de programación pero omite la sintaxis detallada. Cuando sea conveniente, se puede ampliar con lenguaje natural.

Recuerde que si $n \leq m$, escribimos

$$\sum_{k=n}^m x_k = x_n + x_{n+1} + \cdots + x_m$$

y

$$\prod_{k=n}^m x_k = x_n x_{n+1} \cdots x_m$$

Por convención, siempre que $m < n$, definimos

$$\sum_{k=n}^m x_k = 0 \quad \text{y} \quad \prod_{k=n}^m x_k = 1$$

El algoritmo de Horner se puede utilizar en la **deflaxión** de un polinomio. Éste es el proceso de eliminar un factor lineal de un polinomio. Si r es una raíz del polinomio p , entonces $x - r$ es un factor de p . Las raíces restantes de p son las $n - 1$ raíces de un polinomio q de grado 1 menos que el grado de p tal que

$$p(x) = (x - r)q(x) + p(r) \tag{3}$$

donde

$$q(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{n-1}x^{n-1} \tag{4}$$

El seudocódigo para el algoritmo de Horner se puede escribir como se muestra a continuación:

```
integer i, n;  real p, r;  real array (ai)0:n, (bi)0:n-1
bn-1 ← an
for i = n - 1 to 0 do
    bi-1 ← ai + r bi
end for
```

Note que $b_{-1} = p(r)$ en este seudocódigo. Si r es una raíz exacta, entonces el cálculo de $b_{-1} = p(r) = 0.1$. Si el cálculo del algoritmo de Horner se realiza con lápiz y papel, con frecuencia se usa el siguiente arreglo:

$$\begin{array}{c|cccccc} & a_n & a_{n-1} & a_{n-2} & \dots & a_1 & a_0 \\ r & & rb_{n-1} & rb_{n-2} & \dots & rb_1 & rb_0 \\ \hline b_{n-1} & b_{n-2} & b_{n-3} & \dots & b_0 & b_{-1} \end{array}$$

EJEMPLO 4 Use el algoritmo de Horner para evaluar $p(3)$, donde p es el polinomio

$$p(x) = x^4 - 4x^3 + 7x^2 - 5x - 2$$

Solución Arreglamos el cálculo como se sugiere líneas arriba:

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 3 & & 3 & -3 & 12 & 21 \\ \hline 1 & -1 & 4 & 7 & 19 \end{array}$$

Por lo que obtenemos $p(3) = 19$ y podemos escribir

$$p(x) = (x - 3)(x^3 - x^2 + 4x + 7) + 19$$

En el proceso de deflaxión, si r es una raíz del polinomio p , entonces $x - r$ es un factor de p y al contrario. Las raíces restantes de p son las $n - 1$ raíces de $q(x)$.

EJEMPLO 5 Realice la deflaxión del polinomio p del ejemplo anterior, usando el hecho de que 2 es una de sus raíces.

Solución Usamos el mismo arreglo de cálculos como el que se acaba de explicar:

$$\begin{array}{c|ccccc} & 1 & -4 & 7 & -5 & -2 \\ 2 & & 2 & -4 & 6 & 2 \\ \hline & 1 & -2 & 3 & 1 & 0 \end{array}$$

Así, tenemos que $p(2) = 0$, y

$$x^4 - 4x^3 + 7x^2 - 5x - 2 = (x - 2)(x^3 - 2x^2 + 3x + 1)$$

Parejas de problemas fácil/difícil

En computación científica, con frecuencia encontramos parejas de problemas, uno de los cuales es fácil y el otro difícil y son inversos uno del otro. Ésta es la idea principal en criptología, en la que multiplicar dos números juntos es trivial pero el problema inverso (factorizar un número enorme) es casi imposible.

El mismo fenómeno sucede con los polinomios. Con raíces dadas, podemos fácilmente encontrar la forma de potencias del polinomio como el de la ecuación (2). Dada la forma de potencias, puede ser un problema difícil calcular las raíces (y puede ser un problema mal condicionado). El problema de cómputo 1.1.24 pide que escriba un código para calcular los coeficientes de la forma de potencias de un polinomio a partir de sus raíces. Éste es un ciclo con fórmulas simples. Cada vez se agrega un factor $(x - r)$. Éste tema surge nuevamente en álgebra lineal, donde calcular $\mathbf{b} = \mathbf{Ax}$ es trivial pero determinar \mathbf{x} a partir \mathbf{A} y \mathbf{b} (el problema inverso) es difícil (véase la sección 7.1).

Los problemas fácil/difícil surgen nuevamente en problemas de valores a la frontera de dos puntos. Determinar Df y $f(0)$ y $f(1)$ cuando f está dada y D es un operador diferencial es fácil, pero determinar f a partir de conocer $Df, f(0)$ y $f(1)$ es difícil (véase la sección 14.1).

Del mismo modo, calcular los valores característicos de una matriz es un problema difícil. Dados los valores característicos $\lambda_1, \lambda_2, \dots, \lambda_n$ de una matriz $n \times n$ y los correspondientes vectores propios $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ de una matriz de $n \times n$, podemos obtener \mathbf{A} al poner los valores característicos en la diagonal de una matriz diagonal \mathbf{D} y los vectores característicos como columnas en una matriz \mathbf{V} . Entonces $\mathbf{AV} = \mathbf{VD}$, y podemos obtener \mathbf{A} a partir de ésta al resolver la ecuación para \mathbf{A} . Pero determinar λ_i y \mathbf{v}_i a partir de \mathbf{A} misma es difícil (véase la sección 8.3).

El lector puede pensar otros ejemplos.

Primer experimento de programación

Concluimos esta sección con un breve experimento de programación que implica cálculos numéricos. Aquí consideraremos, desde el punto de vista computacional, una operación familiar en cálculo: obtener la derivada de una función. Recuerde que la derivada de una función

f en un punto x está definida por la ecuación

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

Una computadora tiene la capacidad de imitar la operación límite al utilizar una sucesión de números h como

$$h = 4^{-1}, 4^{-2}, 4^{-3}, \dots, 4^{-n}, \dots$$

para que seguramente tiendan a cero rápidamente. Por supuesto, son posibles muchas otras sucesiones simples, como $1/n$, $1/n^2$ y $1/10n$. La sucesión $1/4^n$ que consta de números de máquina en una computadora binaria y, para este experimento, en una computadora de 32 bits, estará suficientemente cerca de cero cuando n es 10.

El siguiente es seudocódigo para calcular $f'(x)$ en el punto $x = 0.5$, con $f(x) = \sin x$:

```
program First
integer i, imax, n ← 30
real error, y, x ← 0.5, h ← 1, emax ← 0
for i = 1 to n do
    h ← 0.25h
    y ← [sin(x + h) - sin(x)]/h
    error ← |cos(x) - y|;  output i, h, y, error
    if error > emax then emax ← error;  imax ← i end if
end for
output imax, emax
end program First
```

No hemos explicado el propósito del experimento ni se ha mostrado la salida de éste seudocódigo. Invitamos al lector a descubrir esto al codificarlo y ejecutarlo (o uno parecido) en una computadora. (Véase los problemas de cómputo 1.1.1 a 1.1.3.)

Software matemático

Los algoritmos y problemas de programación de este libro se han codificado y probado de muchas formas, y están disponibles en el sitio web que se indica en el prefacio. Algunos quedan mejor al usar un lenguaje de programación científico como C, C++, Fortran o cualquier otro que permita realizar cálculos con una precisión adecuada. Algunas veces es instructivo utilizar software de sistemas matemáticos como Matlab, Maple, Mathematica u Octave, ya que tienen procedimientos incorporados para la solución de problemas. Como alternativa, se podría usar una biblioteca de programas matemáticos como IMSL, NAG u otras cuando estén disponibles localmente. Algunas bibliotecas numéricas se han expresamente optimizado para procesadores como los de Intel y AMD. Los sistemas son particularmente útiles para obtener resultados gráficos, así como para experimentar con diferentes métodos numéricos para resolver un problema difícil. Los paquetes de software matemático contienen capacidades de manejo simbólico, como sucede en Maple, Mathematica y Macsyma, son particularmente útiles para obtener soluciones exactas, así como soluciones numéricas. Para resolver los problemas de cómputo, los estudiantes deben centrarse en ganar intuición y una mejor comprensión de los métodos numéricos implicados. El apéndice A ofrece asesoramiento sobre la

programación para cálculos científicos. Las sugerencias son independientes del lenguaje que se utilice.

Con el desarrollo de la amplia red mundial y de internet, se ha vuelto fácil localizar y transferir de una computadora a otra buen software matemático. Se pueden utilizar exploradores, buscadores y direcciones URL para encontrar software que sea aplicable a un área de interés particular. Existen colecciones de software matemático que van desde grandes y amplias bibliotecas a versiones pequeñas de estas bibliotecas para PC; algunas de ellas son interactivas. También, referencias a programas de computadora y colecciones de rutinas se pueden encontrar en libros y reportes técnicos. La URL del sitio web para este libro, dada en el prefacio, presenta un panorama del software matemático disponible, así como de otro material de apoyo.

Resumen

(1) Se usa **multiplicación anidada** para evaluar un polinomio eficientemente:

$$\begin{aligned} p(x) &= a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n \\ &= a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + x(a_n)) \cdots)) \end{aligned}$$

Una parte de seudocódigo para hacer esto es

```

 $p \leftarrow a_n$ 
for  $k = 1$  to  $n$  do
     $p \leftarrow xp + a_{n-k}$ 
end for
```

(2) La deflaxión del polinomio $p(x)$ es la eliminación de un factor lineal:

$$p(x) = (x - r)q(x) + p(r)$$

donde

$$q(x) = b_0 + b_1x + b_2x^2 + \cdots + b_{n-1}x^{n-1}$$

El seudocódigo para el algoritmo de Horner para la deflaxión de un polinomio es

```

 $b_{n-1} \leftarrow a_n$ 
for  $i = n - 1$  to  $0$  do
     $b_{i-1} \leftarrow a_i + rb_i$ 
end for
```

Aquí, $b_{-1} = p(r)$.

Referencias adicionales

Dos interesantes artículos que tienen muchos ejemplos de por qué los métodos numéricos son críticamente importantes son Forsythe [1970] y McCartin [1998]. Véase Briggs [2004] y Friedman y Littman [1994] para conocer varios problemas de la industria y del mundo real.

Problemas 1.1*

- 1.** En la secundaria, algunos estudiantes se han confundido al creer que $22/7$ es ya sea el valor real de π o una aproximación aceptable de π . Muestre que $355/113$ es una mejor aproximación en términos de los errores absoluto y relativo. Encuentre algunas otras fracciones racionales simples n/m que se aproximen a π . Por ejemplo, una para la que $|\pi - n/m| < 10^{-9}$. *Sugerencia:* véase el problema 1.1.4.

- “2.** Un número real x se representa aproximadamente por 0.6032 y decimos que el error relativo está a lo más a 0.1% . ¿Cuánto vale x ?

- “3.** ¿Cuál es el error relativo implicado al redondear 4.9997 a 5.000 ?

- “4.** El valor de π se puede generar con la computadora acercándose a la precisión completa de la máquina con el enunciado de asignación

$$pi \leftarrow 4.0 \arctan(1.0)$$

Sugiera al menos otras cuatro formas de calcular π usando funciones básicas del sistema de su computadora.

- 5.** Se puede agregar un arreglo doblemente subindizado $(a_{ij})_{n \times n}$ en cualquier orden. Escriba partes de seudocódigo para cada uno de los incisos siguientes. ¿Cuál es el mejor?

a. $\sum_{i=1}^n \sum_{j=1}^n a_{ij}$

b. $\sum_{j=1}^n \sum_{i=1}^n a_{ij}$

c. $\sum_{i=1}^n \left(\sum_{j=1}^i a_{ij} + \sum_{j=1}^{i-1} a_{ji} \right)$

d. $\sum_{k=0}^{n-1} \sum_{|i-j|=k} a_{ij}$

e. $\sum_{k=2}^{2n} \sum_{i+j=k} a_{ij}$

- “6.** Cuente el número de operaciones implicadas al evaluar un polinomio usando multiplicación anidada. No cuente los cálculos de subíndices.

- 7.** Para x pequeña, muestre que $(1+x)^2$ puede calcularse algunas veces con más exactitud a partir de $(x+2)x+1$. Explique. ¿Qué otras expresiones se pueden utilizar para calcularlo?

- 8.** Muestre cómo se pueden evaluar eficientemente estos polinomios:

a. $p(x) = x^{32}$

b. $p(x) = 3(x-1)^5 + 7(x-1)^9$

c. $p(x) = 6(x+2)^3 + 9(x+2)^7 + 3(x+2)^{15} - (x+2)^{31}$

d. $p(x) = x^{127} - 5x^{37} + 10x^{17} - 3x^7$

- 9.** Usando la función exponencial $\exp(x)$, escriba una parte de seudocódigo eficiente para el enunciado $y = 5e^{3x} + 7e^{2x} + 9e^x + 11$.

- “10.** Escriba una parte de seudocódigo para evaluar la expresión

$$z = \sum_{i=1}^n b_i^{-1} \prod_{j=1}^i a_j$$

donde (a_1, a_2, \dots, a_n) y (b_1, b_2, \dots, b_n) son arreglos lineales que tienen valores dados.

* Los problemas marcados con “ tienen las respuestas en la parte final del libro.

11. Escriba partes de seudocódigo para evaluar las siguientes expresiones eficientemente:

a. $p(x) = \sum_{k=0}^{n-1} kx^k$

^ab. $z = \sum_{i=1}^n \prod_{j=1}^i x^{n-j+1}$

c. $z = \prod_{i=1}^n \sum_{j=1}^i x_j$

d. $p(t) = \sum_{i=1}^n a_i \prod_{j=1}^{i-1} (t - x_j)$

12. Usando notación de sumatoria y de producto, escriba expresiones matemáticas para las siguientes partes de seudocódigo:

a. **integer** i, n ; **real** v, x ; **real array** $(a_i)_{0:n}$

```

 $v \leftarrow a_0$ 
for  $i = 1$  to  $n$  do
     $v \leftarrow v + xa_i$ 
end for
```

^ab. **integer** i, n ; **real** v, x ; **real array** $(a_i)_{0:n}$

```

 $v \leftarrow a_n$ 
for  $i = 1$  to  $n$  do
     $v \leftarrow vx + a_{n-i}$ 
end for
```

c. **integer** i, n ; **real** v, x ; **real array** $(a_i)_{0:n}$

```

 $v \leftarrow a_0$ 
for  $i = 1$  to  $n$  do
     $v \leftarrow vx + a_i$ 
end for
```

d. **integer** i, n ; **real** v, x, z ; **real array** $(a_i)_{0:n}$

```

 $v \leftarrow a_0$ 
 $z \leftarrow x$ 
for  $i = 1$  to  $n$  do
     $v \leftarrow v + za_i$ 
     $z \leftarrow xz$ 
end for
```

^ae. **integer** i, n ; **real** v ; **real array** $(a_i)_{0:n}$

```

 $v \leftarrow a_n$ 
for  $i = 1$  to  $n$  do
     $v \leftarrow (v + a_{n-i})x$ 
end for
```

13. Exprese en notación matemática sin paréntesis el valor final de z en la siguiente parte de seudocódigo:

integer k, n ; **real** z ; **real array** $(b_i)_{0:n}$

```

 $z \leftarrow b_n + 1$ 
for  $k = 1$  to  $n - 2$  do
     $z \leftarrow z b_{n-k} + 1$ 
end for
```

"14. ¿Cuántas multiplicaciones ocurren en la ejecución de la siguiente parte de seudocódigo?

```

integer  $i, j, n;$  real  $x;$  real array  $(a_{ij})_{0:n \times 0:n}, (b_{ij})_{0:n \times 0:n}$ 
 $x \leftarrow 0.0$ 
for  $j = 1$  to  $n$  do
    for  $i = 1$  to  $j$  do
         $x \leftarrow x + a_{ij}b_{ij}$ 
    end for
end for

```

15. Critique las siguientes partes de seudocódigo y escriba versiones mejoradas:

- a. **integer** $i, n;$ **real** $x, z;$ **real array** $(a_i)_{0:n}$

```

for  $i = 1$  to  $n$  do
     $x \leftarrow z^2 + 5.7$ 
     $a_i \leftarrow x/i$ 
end for

```
- "b.** **integer** $i, j, n;$ **real array** $(a_{ij})_{0:n \times 0:n}$

```

for  $i = 1$  to  $n$  do
    for  $j = 1$  to  $n$  do
         $a_{ij} \leftarrow 1/(i + j - 1)$ 
    end for
end for

```
- c. **integer** $i, j, n;$ **real array** $(a_{ij})_{0:n \times 0:n}$

```

for  $j = 1$  to  $n$  do
    for  $i = 1$  to  $n$  do
         $a_{ij} \leftarrow 1/(i + j - 1)$ 
    end for
end for

```

16. La matriz aumentada $\begin{bmatrix} 3.5713 & 2.1426 & | & 7.2158 \\ 10.714 & 6.4280 & | & 1.3379 \end{bmatrix}$ es para un sistema de dos ecuaciones con dos incógnitas x y y . Repita el ejemplo 2 para este sistema. ¿Pueden pequeños cambios en los datos conducir a un cambio enorme en la solución?

17. Una aproximación de base 60 alrededor de 1750 A.C. es

$$\sqrt{2} \approx 1 + \frac{24}{60} + \frac{51}{60^2} + \frac{10}{60^3}$$

Determine cuán exacta es. Consulte Sauer [2006] para más detalles.

Problemas de cómputo 1.1

- Escriba y corra un programa de computadora que corresponda al seudocódigo del programa *First* descrito en el libro (pág. 10) e interprete los resultados.
- (Continuación) Seleccione una función f y un punto x y realice un experimento computacional como el dado en el libro. Interprete los resultados. No seleccione una función muy simple. Por ejemplo, podría considerar $1/x, \log x, e^x, \tan x, \cosh x$ o $x^3 - 23x$.

3. Como vimos en el primer experimento computacional, la exactitud de una fórmula para derivación numérica puede deteriorarse conforme disminuye el paso h . Estudie la siguiente **fórmula de diferencia central**:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

cuando $h \rightarrow 0$. Aprenderemos en el capítulo 4 que el **error de truncamiento** para esta fórmula es $-\frac{1}{6}h^2 f'''(\xi)$ para alguna ξ en el intervalo $(x-h, x+h)$. Modifique y corra el código para el experimento *First* para que se calculen valores aproximados para el **error de redondeo** y error de truncamiento. En la misma gráfica, trace el error de redondeo, el error de truncamiento y el error total (suma de estos dos errores) usando una escala logarítmica; es decir, los ejes en la gráfica deben ser $-\log_{10}|\text{error}|$ contra $\log_{10} h$. Analice estos resultados.

- 4.** El límite $e = \lim_{n \rightarrow \infty} (1 + 1/n)^n$ define al número e en cálculo. Calcule e tomando el valor de esta expresión para $n = 8, 8^2, 8^3, \dots, 8^{10}$. Compare con e obtenido de $e \leftarrow \exp(1.0)$. Interprete los resultados.
- 5.** No es difícil ver que los números $p_n = \int_0^1 x^n e^x dx$ satisface las desigualdades $p_1 > p_2 > p_3 > \dots > 0$. Establezca este hecho. Despues, use integración por partes para mostrar que $p_{n+1} = e - (n+1)p_n$ y que $p_1 = 1$. En la computadora, use la relación de recurrencia para generar los primeros 20 valores de p_n y explique por qué se violan las desigualdades anteriores. No use variables subindexadas. (Véase Dorn y McCracken [1972], págs. 120–129.)
- 6.** (Continuación) Sea $p_{20} = \frac{1}{8}$ y use la fórmula del problema de cómputo anterior para calcular $p_{19}, p_{18}, \dots, p_2$ y p_1 . ¿Los números generados obedecen las desigualdades $1 = p_1 > p_2 > p_3 > \dots > 0$? Explique la diferencia de los dos procedimientos. Repita con $p_{20} = 20$ o $p_{20} = 100$. Explique lo que pasa.
- 7.** Escriba una rutina eficiente que acepte como entrada una lista de números reales a_1, a_2, \dots, a_n y después calcule lo siguiente:

Media aritmética	$m = \frac{1}{n} \sum_{k=1}^n a_k$
Varianza	$v = \frac{1}{n-1} \sum_{k=1}^n (a_k - m)^2$
Desviación estándar	$\sigma = \sqrt{v}$

Pruebe la rutina con un conjunto de datos de su elección.

- 8.** (Continuación) Muestre que otra fórmula es

$$\text{Varianza } v = \frac{1}{n-1} \left[\sum_{k=1}^n a_k^2 - nm^2 \right]$$

De las dos fórmulas dadas para v , ¿cuál es más exacta en la computadora? Compruébelo en la computadora con un conjunto de datos. *Sugerencia:* use un gran conjunto de números reales que varíen en magnitud desde muy pequeña a muy grande.

- 9.** Sea a_1 dada. Escriba un programa para calcular para $1 \leq n \leq 1000$ los números $b_n = na_{n-1}$ y $a_n = b_n/n$. Escriba los números $a_{100}, a_{200}, \dots, a_{1000}$. No use variables subindexadas. ¿Cuál sería a_n ? Explique cómo se desvía el hecho de la teoría. Determine cuatro valores para a_1 para que el cálculo se desvíe de la teoría en su computadora. *Sugerencia:* considere números extremadamente pequeños y grandes e imprima con precisión completa de la máquina.
- 10.** En una computadora, puede suceder que $a + x = a$ cuando $x \neq 0$. Explique por qué. Describa el conjunto de n para los cuales $1 + 2^{-n} = 1$ en su computadora. Escriba y corra programas adecuados para mostrar el fenómeno.
- 11.** Escriba un programa para probar la sugerencia de programación concerniente al error de redondeo en el cálculo de $t \leftarrow t + h$ contra $t \leftarrow t_0 + ih$. Por ejemplo, use $h = \frac{1}{10}$ y calcule $t \leftarrow t + h$ en doble precisión para el valor correcto de t de precisión simple; imprima los valores absolutos de las diferencias entre este cálculo y los valores de los dos procedimientos. ¿Cuál es el resultado de la prueba cuando h es un número de máquina, como $h = \frac{1}{128}$, en una computadora binaria (con más de siete bits por palabra)?
- 12.** El matemático ruso P. L. Chebyshev (1821–1894) escribía su nombre como Чебышев. Son posibles muchas transliteraciones del alfabeto cirílico al latino. *Cheb* puede alternativamente cambiarse por *Ceb*, *Tscheb* o *Tcheb*. La *y* se puede interpretar como *i*. *Shev* que también se puede interpretar como *schef*, *cev*, *cheff* o *scheff*. Tomando todas las combinaciones de estas variantes, escriba un programa de computadora para imprimir todas las formas posibles de escribir ese nombre.
- 13.** Calcule $n!$ usando logaritmos, aritmética para enteros y aritmética de doble precisión y de punto flotante. Para cada parte, escriba una tabla de valores para $0 \leq n \leq 30$ y determine el valor correcto más grande.
- 14.** Dados dos arreglos, un arreglo real $v = (v_1, v_2, \dots, v_n)$ y un arreglo de permutación de enteros $p = (p_1, p_2, \dots, p_n)$ de los enteros $1, 2, \dots, n$, ¿podemos formar un nuevo arreglo permutado; $v = (v_{p_1}, v_{p_2}, \dots, v_{p_n})$ al sobreescribir v y no implicar otro arreglo en memoria? Si es así, escriba y pruebe el código para hacerlo. Si no, use otro arreglo y pruebe.
- Caso 1.** $v = (6.3, 4.2, 9.3, 6.7, 7.8, 2.4, 3.8, 9.7)$, $p = (2, 3, 8, 7, 1, 4, 6, 5)$
- Caso 2.** $v = (0.7, 0.6, 0.1, 0.3, 0.2, 0.5, 0.4)$, $p = (3, 5, 4, 7, 6, 2, 1)$
- 15.** Usando un sistema algebraico computarizado (por ejemplo, Maple, Derive, Mathematica), imprima 200 dígitos decimales de $\sqrt{10}$.
- 16. a.** Repita el ejemplo (1) acerca de la pérdida de dígitos significativos de exactitud pero realice los cálculos con doble precisión antes de redondearlos. ¿Esto ayuda?
- b.** Use Maple o algún otro software de matemáticas en los que pueda establecer el número de dígitos de precisión. *Sugerencia:* en Maple, use `Digits`.
- 17.** En 1706, Machin usó la fórmula

$$\pi = 16 \arctan\left(\frac{1}{5}\right) - 4 \arctan\left(\frac{1}{239}\right)$$

para calcular 100 dígitos de π . Deduzca esta fórmula. Reproduzca los cálculos de Machin usando un software adecuado. *Sugerencia:* sea $\theta = \frac{1}{5}$ y use identidades trigonométricas comunes.

- 18.** Usando un programa que maneje símbolos como Maple, Mathematica o Macsyma, realice las siguientes tareas. Registre su trabajo de alguna forma, por ejemplo, usando un comando `diary` o `script`.
- Encuentre la serie de Taylor, arriba e incluyendo el término x^{10} , para la función $(\tan x)^2$, usando 0 como el punto x_0 .
 - Encuentre la integral indefinida de $(\cos x)^4$.
 - Encuentre la integral definida $\int_0^1 \log |\log x| dx$.
 - Encuentre el primer número primo más grande que 27448.
 - Obtenga el valor numérico de $\int_0^1 \sqrt{1 + \sin^3 x} dx$.
 - Encuentre la solución de la ecuación diferencial $y' + y = (1 + e^x)^{-1}$.
 - Defina la función $f(x, y) = 9x^4 - y^4 + 2y^2 - 1$. Quiere conocer el valor de $f(40545, 70226)$. Calcúlelo en la forma directa por sustitución directa de $x = 40545$ y $y = 70226$ en la definición de $f(x, y)$, usando primero seis decimales, después siete, ocho y así hasta una exactitud de 24 dígitos decimales. Despues, pruebe mediante álgebra elemental que

$$f(x, y) = (3x^2 - y^2 + 1)(3x^2 + y^2 - 1)$$

Use esta fórmula para calcular el mismo valor de $f(x, y)$, nuevamente usando precisiones diferentes, de seis decimales a 24 decimales. Describa que ha aprendido. Para forzar el programa a hacer operaciones de punto flotante en lugar de aritmética con enteros, escriba sus números en la forma 9.0, 40545.0 y así sucesivamente.

- 19.** Considere las siguientes partes de seudocódigo:

```

a. integer i; real x, y, z
for i = 1 to 20 do
    x  $\leftarrow$  2 + 1.0/8i
    y  $\leftarrow$  arctan(x) – arctan(2)
    z  $\leftarrow$  8iy
    output x, y, z
end for
b. real epsi  $\leftarrow$  1
while 1 < 1 + epsi do
    epsi  $\leftarrow$  epsi/2
    output epsi
end while

```

¿Cuál es el propósito de cada programa? ¿Se logra? Explique. Codifique y ejecute cada uno para comprobar sus conclusiones.

- 20.** Considere algunos descuidos implicados en los enunciados de asignación.
- ¿Cuál es la diferencia entre los siguientes dos enunciados de asignación? Escriba un código que los tenga y muestre con ejemplos específicos que algunas veces $x = y$ y algunas veces $x \neq y$.

```
integer m, n; real x, y
x  $\leftarrow$  real(m/n)
y  $\leftarrow$  real(m)/real(n)
output x, y
```

- b. ¿Qué valor recibirá n ?

```
integer n; real x, y
x  $\leftarrow$  7.4
y  $\leftarrow$  3.8
n  $\leftarrow$  x + y
output n
```

¿Qué pasa cuando el último enunciado se remplaza con el siguiente?

$$n \leftarrow \text{integer}(x) + \text{integer}(y)$$

21. Escriba un código de computadora que tenga los siguientes enunciados de asignación exactamente como se muestran. Analice los resultados.

- a. Primero imprima estos valores usando el formato predeterminado y después con un campo de formato extremadamente largo:

```
real p, q, u, v, w, x, y, z
x  $\leftarrow$  0.1
y  $\leftarrow$  0.01
z  $\leftarrow$  x - y
p  $\leftarrow$  1.0/3.0
q  $\leftarrow$  3.0p
u  $\leftarrow$  7.6
v  $\leftarrow$  2.9
w  $\leftarrow$  u - v
output x, y, z, p, q, u, v, w
```

- b. ¿Qué valores se calcularían para x , y y z si se usa este código?

```
integer n; real x, y, z
for n = 1 to 10 do
    x  $\leftarrow$  (n - 1)/2
    y  $\leftarrow$  n2/3.0
    z  $\leftarrow$  1.0 + 1/n
    output x, y, z
end for
```

- c. ¿Qué valores producirían los siguientes enunciados de asignación?

```
integer i, j; real c, f, x, half
x  $\leftarrow$  10/3
i  $\leftarrow$  integer(x + 1/2)
half  $\leftarrow$  1/2
j  $\leftarrow$  integer(half)
```

```

 $c \leftarrow (5/9)(f - 32)$ 
 $f \leftarrow 9/5c + 32$ 
output  $x, i, half, j, c, f$ 

```

- d. Analice por qué está equivocada la siguiente parte de seudocódigo:

```

real area, circum, radius
radius  $\leftarrow 1$ 
area  $\leftarrow (22/7)(radius)^2$ 
circum  $\leftarrow 2(3.1416)radius$ 
output area, circum

```

22. Critique el siguiente seudocódigo para evaluar $\lim_{x \rightarrow 0} \arctan(|x|)/x$. Codifíquelo y ejecútelo para ver qué pasa.

```

integer i; real x, y
x  $\leftarrow 1$ 
for i = 1 to 24 do
    x  $\leftarrow x/2.0$ 
    y  $\leftarrow \arctan(|x|)/x$ 
    output x, y
end for

```

23. Realice algunos experimentos computacionales para mostrar o probar las sugerencias de programación del apéndice A. Los temas específicos por incluir son: (a) cuándo evitar arreglos, (b) cuándo limitar iteraciones, (c) comprobación de igualdad de punto flotante, (d) formas para tomar pasos iguales de punto flotante y (e) varias formas para evaluar funciones. *Sugerencia:* puede ser útil comparar resultados de simple y doble precisión.

24. (**Parejas de problemas fácil/difícil**) Escriba un programa de computadora para obtener la forma de potencias de un polinomio a partir de sus raíces. Sean las raíces r_1, r_2, \dots, r_n . Entonces (excepto para un factor escalar) el polinomio es el producto

$$p(x) = (x - r_1)(x - r_2) \cdots (x - r_n).$$

Encuentre los coeficientes en la expresión $p(x) = \sum_{j=0}^n a_j x^j$. Pruebe su código con los polinomios de Wilkinson en los problemas de cómputo 3.1.10 y 3.3.9. Explique por qué esta tarea de obtener la forma de potencias del polinomio es *trivial*, mientras que el problema inverso de determinar las raíces a partir de la forma de potencias es bastante difícil.

25. Un número primo es un entero positivo que no tiene otros factores enteros que no sean él mismo y 1. ¿Cuántos números primos hay en cada uno de estos intervalos abiertos: (1, 40), (1, 80), (1, 160), y (1, 2000)? Haga una conjetura acerca de cuál es el porcentaje de números primos entre todos los números.
26. El software matemático como Maple y Mathematica hace cálculos numéricos y manejos simbólicos. Compruebe simbólicamente que una multiplicación anidada es correcta para un polinomio general de grado 10.

1.2 Repaso de series de Taylor

La mayoría de los alumnos encontraron series infinitas (particularmente series de Taylor) en su estudio del cálculo sin necesariamente haber adquirido un buen entendimiento de este tema. Por consiguiente, esta sección es particularmente importante para el análisis numérico y merece un particular estudio.

Una vez que los alumnos están bien fundamentados con un entendimiento básico de series de Taylor, el teorema del valor medio y las series alternantes (todos los temas de esta sección), así como de la representación numérica en computadora (sección 2.2), pueden continuar estudiando las bases de los métodos numéricos con mejor comprensión.

Series de Taylor

Ejemplos familiares (y útiles) de series de Taylor son los siguientes:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad (|x| < \infty) \quad (1)$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \quad (|x| < \infty) \quad (2)$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} \quad (|x| < \infty) \quad (3)$$

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots = \sum_{k=0}^{\infty} x^k \quad (|x| < 1) \quad (4)$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^k}{k} \quad (-1 < x \leq 1) \quad (5)$$

En cada caso, la serie representa la función dada y converge en el intervalo especificado. Las series (1) a (5) son series de Taylor desarrolladas alrededor de $c = 0$. Una serie de Taylor desarrollada alrededor de $c = 1$ es

$$\ln(x) = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \cdots = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{(x-1)^k}{k}$$

donde $0 < x \leq 2$. El lector debe recordar la notación **factorial**

$$n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdots \cdot n$$

para $n \geq 1$ y la definición especial de $0! = 1$.

Series de este tipo se usan con frecuencia para calcular buenos valores aproximados de complicadas funciones en puntos específicos.

EJEMPLO 1 Use cinco términos de la serie (5) para aproximar $\ln(1.1)$.

Solución Tomando $x = 0.1$ en los primeros cinco términos de la serie para $\ln(1 + x)$ se obtiene

$$\ln(1.1) \approx 0.1 - \frac{0.01}{2} + \frac{0.001}{3} - \frac{0.0001}{4} + \frac{0.00001}{5} = 0.09531\ 03333\dots$$

donde \approx significa “aproximadamente igual.” Este valor es correcto con seis cifras decimales de exactitud. ■

Por otra parte, no siempre se obtienen buenos resultados usando series.

EJEMPLO 2 Intente calcular e^8 usando la serie (1).

Solución El resultado es

$$e^8 = 1 + 8 + \frac{64}{2} + \frac{512}{6} + \frac{4096}{24} + \frac{32768}{120} + \dots$$

Es obvio que se necesitarán muchos términos para calcular e^8 con razonable precisión. Elevando al cuadrado en forma repetida, encontramos $e^2 = 7.38905\ 6$, $e^4 = 54.59815\ 00$ y $e^8 = 2980.95798\ 7$. Con los primeros seis términos dados se obtiene 570.06666 5. ■

Estos ejemplos ilustran una regla general:

Una serie de Taylor converge rápidamente cerca del punto de expansión y lentamente (o no lo hace) en puntos más lejanos.

Una descripción visual del fenómeno se puede obtener al graficar algunas sumas parciales de una serie de Taylor. En la figura 1.2 mostramos la función

$$1 = \sin x$$

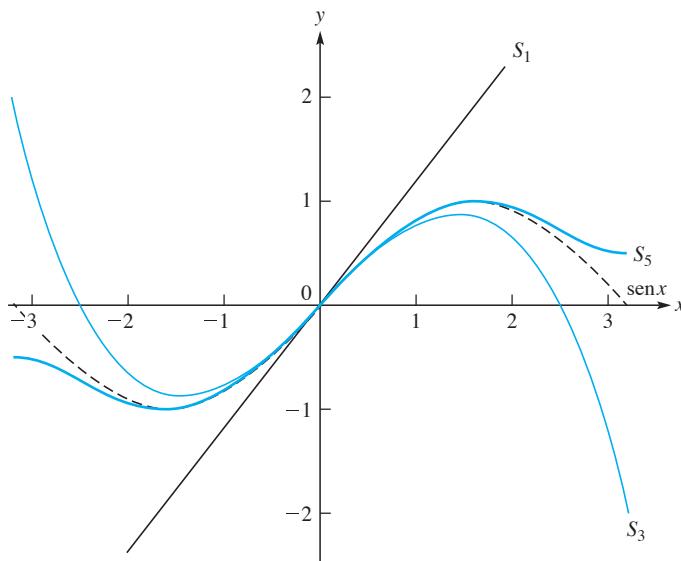


FIGURA 1.2
Aproximaciones de $\sin x$

y la suma parcial de funciones

$$\begin{aligned}S_1 &= x \\S_3 &= x - \frac{x^3}{6} \\S_5 &= x - \frac{x^3}{6} + \frac{x^5}{120}\end{aligned}$$

que provienen de la serie (2). Mientras que S_1 puede ser una aproximación aceptable de $\sin x$ cuando $x \approx 0$, las gráficas para S_3 y S_4 se acoplan a la de $\sin x$ para grandes intervalos con respecto al origen.

Todas las series ilustradas antes son ejemplos de la siguiente serie general:

■ TEOREMA 1

Serie de Taylor formal para f con respecto a c

$$\begin{aligned}f(x) &\sim f(c) + f'(c)(x - c) + \frac{f''(c)}{2!}(x - c)^2 + \frac{f'''(c)}{3!}(x - c)^3 + \dots \\f(x) &\sim \sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!}(x - c)^k\end{aligned}\tag{6}$$

Aquí, en vez de usar $=$ hemos escrito \sim para indicar que no se nos permite suponer que $f(x)$ es igual a la serie de la derecha. Todo lo que tenemos por el momento es que una serie formal se puede escribir suponiendo que las derivadas sucesivas f', f'', f''', \dots , existen en el punto c . La serie (6) se llama **serie de Taylor de f en el punto c** .

En el caso especial $c = 0$, la serie (6) también se llama una **serie de Maclaurin**:

$$\begin{aligned}f(x) &\sim f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \dots \\f(x) &\sim \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!}x^k\end{aligned}\tag{7}$$

El primer término es $f(0)$ cuando $k = 0$.

EJEMPLO 3

¿Cuál es la serie de Taylor de la función

$$f(x) = 3x^5 - 2x^4 + 15x^3 + 13x^2 - 12x - 5$$

en el punto $c = 2$?

Solución Para calcular los coeficientes en la serie necesitamos los valores numéricos de $f^{(k)}(2)$ para $k \geq 0$. A continuación se muestran los detalles del cálculo:

$$\begin{array}{lll}f(x) &= 3x^5 - 2x^4 + 15x^3 + 13x^2 - 12x - 5 & f(2) &= 207 \\f'(x) &= 15x^4 - 8x^3 + 45x^2 + 26x - 12 & f'(2) &= 396 \\f''(x) &= 60x^3 - 24x^2 + 90x + 26 & f''(2) &= 590 \\f'''(x) &= 180x^2 - 48x + 90 & f'''(2) &= 714 \\f^{(4)}(x) &= 360x - 48 & f^{(4)}(2) &= 672 \\f^{(5)}(x) &= 360 & f^{(5)}(2) &= 360 \\f^{(k)}(x) &= 0 & f^{(k)}(2) &= 0\end{array}$$

para $k \geq 6$. Por tanto, tenemos

$$\begin{aligned} f(x) &\sim 207 + 396(x - 2) + 295(x - 2)^2 \\ &\quad + 119(x - 2)^3 + 28(x - 2)^4 + 3(x - 2)^5 \end{aligned}$$

En este ejemplo, no es difícil ver que \sim se puede remplazar por $=$. Simplemente desarrollamos todos los términos en la serie de Taylor y los reunimos para obtener la forma original para f . El teorema de Taylor, que pronto analizaremos, nos permitirá llegar a esta conclusión ¡sin realizar ningún trabajo! ■

Algoritmo completo de Horner

Una aplicación del algoritmo de Horner es la de hallar el desarrollo de Taylor de un polinomio con respecto a cualquier punto. Sea $p(x)$ un polinomio dado de grado n con coeficientes a_k como en la ecuación (2) de la sección 1.1 y suponga que queremos los coeficientes c_k , en la ecuación

$$\begin{aligned} p(x) &= a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0 \\ &= c_n (x - r)^n + c_{n-1} (x - r)^{n-1} + \cdots + c_1 (x - r) + c_0 \end{aligned}$$

Por supuesto, el teorema de Taylor asegura que $c_k = p^{(k)}(r)/k!$, pero buscamos un algoritmo más eficiente. Observe que $p(r) = c_0$, así que este coeficiente se obtiene al aplicar el algoritmo de Horner al polinomio p con el punto r . El algoritmo también produce el polinomio

$$q(x) = \frac{p(x) - p(r)}{x - r} = c_n (x - r)^{n-1} + c_{n-1} (x - r)^{n-2} + \cdots + c_1$$

Esto muestra que el segundo coeficiente, c_1 , se puede obtener al aplicar el algoritmo de Horner al polinomio q con punto r , ya que $c_1 = q(r)$. (Observe que la primera aplicación del algoritmo de Horner no produce q en la forma que se muestra sino más bien como una suma de potencias de x . (Véanse las ecuaciones (3)–(4) de la sección 1.1.) Este proceso se repite hasta que se encuentran todos los coeficientes c_k .

Al algoritmo que acabamos de describir le llamamos **algoritmo completo de Horner**. El pseudocódigo para ejecutarlo se arregla así para que los coeficientes c_k sobreesciban los coeficientes de entrada a_k .

```
integer n, k, j;  real r;  real array (ai)0:n
for k = 0 to n - 1 do
    for j = n - 1 to k do
        aj ← aj + r aj+1
    end for
end for
```

Este procedimiento se puede utilizar para aplicar el método de Newton a fin de determinar las raíces de un polinomio, lo que analizaremos en el capítulo 3. Además, esto se puede hacer con aritmética compleja para manejar polinomios con raíces o coeficientes complejos.

EJEMPLO 4 Usando el algoritmo completo de Horner, encuentre el desarrollo de Taylor del polinomio

$$p(x) = x^4 - 4x^3 + 7x^2 - 5x + 2$$

alrededor del punto $r = 3$.

Solución El trabajo se puede arreglar como se muestra a continuación:

$$\begin{array}{r|ccccc} & 1 & -4 & 7 & -5 & 2 \\ 3 & & 3 & -3 & 12 & 21 \\ \hline & 1 & -1 & 4 & 7 & 23 \\ & & 3 & 6 & 30 & \\ \hline & 1 & 2 & 10 & 37 & \\ & & 3 & 15 & & \\ \hline & 1 & 5 & 25 & & \\ & & 3 & & & \\ \hline & 1 & & 8 & & \end{array}$$

El cálculo muestra que

$$p(x) = (x - 3)^4 + 8(x - 3)^3 + 25(x - 3)^2 + 37(x - 3) + 23$$



Teorema de Taylor en términos de $(x - c)$

TEOREMA 2

Teorema de Taylor para $f(x)$

Si la función f tiene derivadas continuas de órdenes $0, 1, 2, \dots, (n+1)$ en un intervalo cerrado $I = [a, b]$, entonces para cualquier c y x en I ,

$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(c)}{k!}(x - c)^k + E_{n+1} \quad (8)$$

donde el término de error E_{n+1} puede estar dado en la forma

$$E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - c)^{n+1}$$

Aquí ξ es un punto que se encuentra entre c y x y depende de ambos.

En cálculos prácticos con series de Taylor, con frecuencia es necesario *truncar* las series porque no es posible realizar un número infinito de sumas. Se dice que una serie está **truncada** si despreciamos todos los términos después de un punto dado. Por ende, si truncamos la serie exponencial (1) después de siete términos, el resultado es

$$e^x \approx 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!}$$

Esto ya no representa e^x excepto cuando $x = 0$. Pero la serie truncada debe *aproximar* a e^x . Aquí es donde necesitamos el teorema de Taylor. Con esta ayuda podemos evaluar la diferencia entre una función f y su serie de Taylor truncada.

La suposición explícita en este teorema es que $f(x), f'(x), f''(x), \dots, f^{(n+1)}(x)$, son todas funciones continuas en el intervalo $I = [a, b]$. El término final E_{n+1} en la ecuación (8) es el **residuo o término de error**. La fórmula dada para E_{n+1} es válida cuando suponemos sólo que $f^{(n+1)}$ existe en cada punto del intervalo abierto (a, b) . El término de error es similar a los términos que lo preceden, pero observe que $f^{(n+1)}$ se debe evaluar en un punto diferente de c . Este punto ξ

depende de x y está en el intervalo abierto (c, x) o (x, c) . Son posibles otras formas del residuo; la dada aquí es la **forma de Lagrange**. (Aquí no se dé el Teorema de Taylor.)

EJEMPLO 5 Deduzca la serie de Taylor para e^x en $c = 0$ y pruebe que ésta converge a e^x usando el teorema de Taylor.

Solución Si $f(x) = e^x$, entonces $f^{(k)}(x) = e^x$ para $k \geq 0$. Por tanto, $f^{(k)}(c) = f^{(k)}(0) = e^0 = 1$ para toda k . De la ecuación (8), tenemos

$$e^x = \sum_{k=0}^n \frac{x^k}{k!} + \frac{e^\xi}{(n+1)!} x^{n+1} \quad (9)$$

Ahora vamos a considerar todos los valores de x en algún intervalo simétrico alrededor del origen; por ejemplo, $-s \leq x \leq s$. Entonces $|x| \leq s$, $|\xi| \leq s$ y $e^\xi \leq e^s$. Por tanto, el término residuo satisface esta desigualdad:

$$\lim_{n \rightarrow \infty} \left| \frac{e^\xi}{(n+1)!} x^{n+1} \right| \leq \lim_{n \rightarrow \infty} \frac{e^s}{(n+1)!} s^{n+1} = 0$$

Por ende, si tomamos el límite cuando $n \rightarrow \infty$ en ambos miembros de la ecuación (9), obtenemos

$$e^x = \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{x^k}{k!} = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

■

Este ejemplo muestra cómo podemos establecer, en casos específicos, que una serie formal de Taylor (6) en realidad representa la función. Permítanos examinar otro ejemplo para ver cómo la serie formal puede *fallar* al representar la función.

EJEMPLO 6 Deduzca la serie formal de Taylor para $f(x) = \ln(1+x)$ en $c = 0$ y determine el rango de x positivas para las que la serie representa la función.

Solución Necesitamos $f^{(k)}(x)$ y $f^{(k)}(0)$ para $k \geq 1$. Aquí está el trabajo:

$$\begin{aligned} f(x) &= \ln(1+x) & f(0) &= 0 \\ f'(x) &= (1+x)^{-1} & f'(0) &= 1 \\ f''(x) &= -(1+x)^{-2} & f''(0) &= -1 \\ f'''(x) &= 2(1+x)^{-3} & f'''(0) &= 2 \\ f^{(4)}(x) &= -6(1+x)^{-4} & f^{(4)}(0) &= -6 \\ &\vdots & &\vdots \\ f^{(k)}(x) &= (-1)^{k-1}(k-1)!(1+x)^{-k} & f^{(k)}(0) &= (-1)^{k-1}(k-1)! \end{aligned}$$

Por tanto, por el teorema de Taylor obtenemos

$$\begin{aligned} \ln(1+x) &= \sum_{k=1}^n (-1)^{k-1} \frac{(k-1)!}{k!} x^k + \frac{(-1)^n n! (1+\xi)^{-n-1}}{(n+1)!} x^{n+1} \\ &= \sum_{k=1}^n (-1)^{k-1} \frac{x^k}{k} + \frac{(-1)^n}{n+1} (1+\xi)^{-n-1} x^{n+1} \end{aligned} \quad (10)$$

Para que la serie *infinita* represente $\ln(1 + x)$ es necesario y suficiente que el término de error converja a cero cuando $n \rightarrow \infty$. Suponga que $0 \leq x \leq 1$. Entonces $0 \leq \xi \leq x$ (ya que cero es el punto de desarrollo); por ende, $0 \leq x/(1 + \xi) \leq 1$. Por tanto, el término de error converge a cero en este caso. Si $x > 1$, los términos en la serie no tienden a cero y la serie no converge. Luego, la serie representa $\ln(1 + x)$ si $0 \leq x \leq 1$ pero *no* si $x > 1$. (La serie también representa $\ln(1 + x)$ para $-1 < x < 0$ pero *no* si $x \leq -1$.)

Teorema del valor medio

El caso especial $n = 0$ en el teorema de Taylor se conoce como **teorema del valor medio**. Este se enuncia normalmente, sin embargo, en una forma un tanto más precisa.

■ TEOREMA 3

Teorema del valor medio

Si f es una función continua en el intervalo cerrado $[a, b]$ y tiene una derivada en cada punto del intervalo abierto (a, b) , entonces

$$f(b) = f(a) + (b - a)f'(\xi)$$

para alguna ξ en (a, b) .

Por tanto, el cociente $[f(b) - f(a)]/(b - a)$ es igual a la derivada de f en algún punto ξ entre a y b ; esto es, para algún $\xi \in (a, b)$

$$f'(\xi) = \frac{f(b) - f(a)}{b - a}$$

El lado derecho se podría utilizar como una *aproximación* para $f'(x)$ en cualquier x dentro del intervalo (a, b) . La aproximación de derivadas se analiza con más detalle en la sección 4.3.

Teorema de Taylor en términos de h

Con frecuencia son útiles otras formas del teorema de Taylor. Éstas se pueden obtener de la fórmula (8) al cambiar las variables.

■ COROLARIO 1

Teorema del valor medio para $f(x + h)$

Si la función f tiene derivadas continuas de orden $0, 1, 2, \dots, (n + 1)$ en un intervalo cerrado $I = [a, b]$, entonces para cualquier x en I

$$f(x + h) = \sum_{k=0}^n \frac{f^{(k)}(x)}{k!} h^k + E_{n+1} \quad (11)$$

donde h es cualquier valor tal que $x + h$ está en I y donde

$$E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n + 1)!} h^{n+1}$$

para alguna ξ entre x y $x + h$.

La forma (11) se obtiene de la ecuación (8) al remplazar x por $x + h$ y remplazar c por x . Observe que como h puede ser positiva o negativa, el requisito sobre ξ significa $x < \xi < x + h$ si $h > 0$ o bien, $x + h < \xi < x$ si $h < 0$.

El **término de error** E_{n+1} depende de h en dos formas. Primero, h^{n+1} está explícitamente presente; segundo, el punto ξ generalmente depende de h . Puesto que h converge a cero, E_{n+1} converge a cero esencialmente con la misma rapidez con la que h^{n+1} converge a cero. Para n grande, converge muy rápido. Para expresar este hecho cualitativo escribimos

$$E_{n+1} = \mathcal{O}(h^{n+1})$$

cuando $h \rightarrow 0$. Esto se llama **notación O grande** y es la notación abreviada para la desigualdad

$$|E_{n+1}| \leq C|h|^{n+1}$$

donde C es una constante. En las presentes circunstancias, esta constante podría ser cualquier número para el que $|f^{(n+1)}(t)|/(n+1)! \leq C$, para toda t en el intervalo dado inicialmente, I . Más o menos, $E_{n+1} = \mathcal{O}(h^{n+1})$ significa que el comportamiento de E_{n+1} es similar al de la expresión mucho más simple h^{n+1} .

Es importante darse cuenta que la ecuación (11) corresponde a una secuencia completa de teoremas, uno para cada valor de n . Por ejemplo, podemos escribir los casos $n = 0, 1, 2$ como se muestra a continuación:

$$\begin{aligned} f(x+h) &= f(x) + f'(\xi_1)h \\ &= f(x) + \mathcal{O}(h) \\ f(x+h) &= f(x) + f'(x)h + \frac{1}{2!} f''(\xi_2)h^2 \\ &= f(x) + f'(x)h + \mathcal{O}(h^2) \\ f(x+h) &= f(x) + f'(x)h + \frac{1}{2!} f''(x)h^2 + \frac{1}{3!} f'''(\xi_3)h^3 \\ &= f(x) + f'(x)h + \frac{1}{2!} f''(x)h^2 + \mathcal{O}(h^3) \end{aligned}$$

No se puede insistir demasiado en la importancia del término de error en el teorema de Taylor. En capítulos posteriores, muchas situaciones requieren un cálculo de errores en un proceso numéricico usando el teorema de Taylor. A continuación se presentan algunos ejemplos elementales.

EJEMPLO 7 Desarrolle $\sqrt{1+h}$ en potencias de h . Después calcule $\sqrt{1.00001}$ y $\sqrt{0.99999}$.

Solución Sea $f(x) = x^{1/2}$. Entonces $f'(x) = \frac{1}{2}x^{-1/2}$, $f''(x) = -\frac{1}{4}x^{-3/2}$, $f'''(x) = \frac{3}{8}x^{-5/2}$ y así sucesivamente. Ahora use la ecuación (11) con $x = 1$. Tomando $n = 2$ como ejemplo, tenemos que

$$\sqrt{1+h} = 1 + \frac{1}{2}h - \frac{1}{8}h^2 + \frac{1}{16}h^3\xi^{-5/2}$$

donde ξ es un número desconocido que satisface $1 < \xi < 1+h$, si $h > 0$. Es importante observar que la función $f(x) = \sqrt{x}$ tiene derivadas de todos órdenes en cualquier punto $x > 0$.

En la ecuación (12), sea $h = 10^{-5}$. Entonces

$$\sqrt{1.00001} \approx 1 + 0.5 \times 10^{-5} - 0.125 \times 10^{-10} = 1.000004999987500$$

Sustituyendo $-h$ por h en la serie, obtenemos

$$\sqrt{1-h} = 1 - \frac{1}{2}h - \frac{1}{8}h^2 - \frac{1}{16}h^3\xi^{-5/2}$$

Por tanto, tenemos

$$\sqrt{0.99999} \approx 0.99999\ 49999\ 87500$$

Puesto que $1 < \xi < 1 + h$, el error absoluto no excede

$$\frac{1}{16}h^3\xi^{-5/2} < \frac{1}{16}10^{-15} = 0.00000\ 00000\ 00000\ 0625$$

y ambos valores numéricos son correctos para todas las 15 cifras decimales que se muestran. ■

Series alternantes

Otro teorema del cálculo es con frecuencia útil para establecer la convergencia de una serie y calcular el error implicado en el truncamiento. A partir de esto, tenemos el siguiente principio importante para series alternantes:

Si las magnitudes de los términos en una serie alternante convergen monótonicamente a cero, entonces el error en el truncamiento de la serie no es más grande que la magnitud del primer término omitido.

Este teorema se aplica sólo a **series alternantes**, es decir, series en los que términos sucesivos se alternan en positivos y negativos.

TEOREMA 4

Teorema de series alternantes

Si $a_1 \geq a_2 \geq \dots \geq a_n \geq \dots 0$ para toda n y $\lim_{n \rightarrow \infty} a_n = 0$, entonces la serie alternante converge

$$a_1 - a_2 + a_3 - a_4 + \dots$$

converge; es decir,

$$\sum_{k=1}^{\infty} (-1)^{k-1} a_k = \lim_{n \rightarrow \infty} \sum_{k=1}^n (-1)^{k-1} a_k = \lim_{n \rightarrow \infty} S_n = S$$

donde S es su suma y S_n es la n -ésima suma parcial. Además, para toda n ,

$$|S - S_n| \leq a_{n+1}$$

EJEMPLO 8 Si la serie seno se usa en el cálculo de $\sin 1$ con un error menor que $\frac{1}{2} \times 10^{-6}$, ¿cuántos términos se necesitan?

Solución De la serie (2), tenemos

$$\sin 1 = 1 - \frac{1}{3!} + \frac{1}{5!} - \frac{1}{7!} + \dots$$

Si nos detenemos en $1/(2n - 1)!$, el error no excede el primer término despreciado, que es $1/(2n + 1)!$. Por ello, debemos seleccionar n tal que

$$\frac{1}{(2n + 1)!} < \frac{1}{2} \times 10^{-6}$$

Usando logaritmos de base 10, obtenemos $\log(2n + 1)! > \log 2 + 6 = 6.3$. Con una calculadora, calculamos una tabla de valores para $\log n!$ y encontramos que $\log 10! \approx 6.6$. Por tanto, si $n \geq 5$, el error será aceptable. ■

EJEMPLO 9 Si la serie logarítmica (5) se usara para calcular $\ln 2$ con un error menor que $\frac{1}{2} \times 10^{-6}$, ¿cuántos términos se requerirían?

Solución Para calcular $\ln 2$, tomamos $x = 1$ en la serie y usando \approx que significa aproximadamente igual, tenemos

$$S = \ln 2 \approx 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots + \frac{(-1)^{n-1}}{n} = S_n$$

Por el teorema de series alternantes, el error implicado cuando se trunca la serie con n términos es

$$|S - S_n| \leq \frac{1}{n+1}$$

Seleccionamos n tal que

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

Por tanto, ¡se necesitarían más de dos millones de términos! Concluimos que este método para calcular $\ln 2$ no es práctico. (Véanse los problemas del 1.2.10 al 1.2.12 para algunas buenas opciones). ■

Se necesita una nota de precaución con respecto a esta técnica de cálculo del número de términos que se usarán en una serie para hacer exactamente el $(n + 1)$ -ésimo término menor que cierta tolerancia. Este procedimiento es válido sólo para series alternantes en las que los términos disminuyen en magnitud hasta cero, aunque en otros casos esto ocasionalmente se usa para obtener una estimación burda. Por ejemplo, se puede usar para identificar una serie no alterante como una que converge lentamente. Cuando no se puede usar esta técnica, hay que establecer un límite en los términos restantes de la serie. Determinar dicho límite puede ser algo difícil.

EJEMPLO 10 Se sabe que

$$\frac{\pi^4}{90} = 1^{-4} + 2^{-4} + 3^{-4} + \cdots$$

¿Cuántos términos se deben tomar para calcular $\pi^4/90$ con un error a lo más de $\frac{1}{2} \times 10^{-6}$?

Solución Se sigue un método simple

$$1^{-4} + 2^{-4} + 3^{-4} + \cdots + n^{-4}$$

donde n se elige de tal forma que el término siguiente, $(n + 1)^{-4}$, es menor que 37, pero ésta es respuesta errónea, ya que la suma parcial

$$S_{37} = \sum_{k=1}^{37} k^{-4}$$

difiere de $\pi^4/90$ en aproximadamente 6×10^{-6} . Lo que debemos hacer, por supuesto, es seleccionar n para que *todos* los términos omitidos se sumen a menor que $\frac{1}{2} \times 10^{-6}$; es decir,

$$\sum_{k=n+1}^{\infty} k^{-4} < \frac{1}{2} \times 10^{-6}$$

Por una técnica familiar del cálculo (figura 1.3), tenemos

$$\sum_{k=n+1}^{\infty} k^{-4} < \int_n^{\infty} x^{-4} dx = \left. \frac{x^{-3}}{-3} \right|_n^{\infty} = \frac{1}{3n^3}$$

Por ello, basta seleccionar n tal que $(3n^3)^{-1} < \frac{1}{2} \times 10^{-6}$, o $n \geq 88$. (Un análisis más complejo mejorará esto considerablemente.)

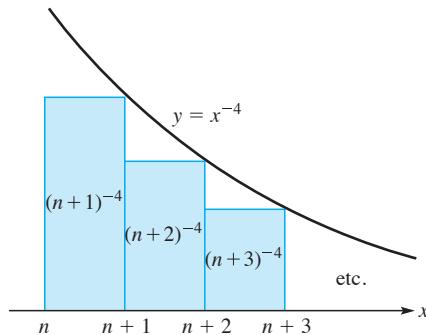


FIGURA 1.3
Ilustración
del ejemplo 10



Resumen

(1) El desarrollo de la serie de Taylor alrededor de c para $f(x)$ es

$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(c)}{k!} (x - c)^k + E_{n+1}$$

con término de error

$$E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - c)^{n+1}$$

Una forma más útil para nosotros es el **desarrollo de la serie de Taylor** para $f(x + h)$, que es

$$f(x + h) = \sum_{k=0}^n \frac{f^{(k)}(x)}{k!} h^k + E_{n+1}$$

con término de error

$$E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1} = \mathcal{O}(h^{n+1})$$

(2) Una serie alternante

$$S = \sum_{k=1}^{\infty} (-1)^{k-1} a_k$$

converge cuando los términos a_k convergen hacia abajo a cero. Además, la suma parcial S_n difiere de S por una cantidad que está limitada por

$$|S - S_n| \leq a_{n+1}$$

Referencias adicionales

Para un estudio adicional, consulte las referencias siguientes que se encuentran en la bibliografía: Atkinson [1988, 1993], Burden y Faires [2001], Conte y De Boor [1980], Dahlquist y Björck [1974], Forsythe, Malcolm y Moler [1977], Fröberg [1969], Gautschi [1997], Gerald y Wheatley [1999], Golub y Ortega [1993], Golub y Van Loan [1996], Häammerlin y Hoffmann [1991], Heath [2002], Higham y Higham [2006], Hildebrand [1974], Isaacson y Keller [1966], Kahaner, Moler y Nash [1989], Kincaid y Cheney [2002], Maron [1991], Moler [2004], Nievergelt, Farra y Reinhold [1974], Oliveira y Stewart [2006], Ortega [1990a], Phillips y Taylor [1973], Ralston [1965], Ralston y Rabinowitz [2001], Rice [1983], Scheid [1968], Skeel y Keiper [1992], Van Loan [1997, 2000] Wood [1999] y Young y Gregory [1988].

Algunos otros libros de métodos numéricos con énfasis en un software matemático o lenguaje de computación específico son Chapman [2000], Devitt [1993], Ellis y Lodi [1991], Ellis, Johnson, Lodi y Schwalbe [1997], Garvan [2002], Knight [2000], Lindfield y Penny [2000], Press, Teukolsky, Vetterling y Flannery [2002], Recktenwald [2000], Schilling y Harris [2000] y Szabo [2002].

Problemas 1.2

1. La serie de Maclaurin para $(1+x)^n$ también se conoce como la **serie binomial**. Establece que

$$(1+x)^n = 1 + nx + \frac{n(n-1)}{2!}x^2 + \frac{n(n-1)(n-2)}{3!}x^3 + \dots \quad (x^2 < 1)$$

Deduzca esta serie. Despues escriba sus formas particulares en notación de sumatoria al hacer $n=2$, $n=3$ y $n=\frac{1}{2}$. Despues use la última forma para calcular $\sqrt{1.0001}$ en forma correcta con 15 cifras decimales (redondeada).

2. (Continuación) Use la serie del problema anterior para obtener la serie (4). ¿Cómo podría usarse esta serie en una máquina de cálculo para producir x/y si sólo están incorporadas las operaciones suma y multiplicación?
3. (Continuación) Use el problema anterior para obtener una serie para $(1+x^2)^{-1}$.

- 4.** ¿Por qué las siguientes funciones no tienen desarrollos de serie de Taylor en $x = 0$?
- a.** $f(x) = \sqrt{x}$ **b.** $f(x) = |x|$ **c.** $f(x) = \arcsen(x - 1)$
d. $f(x) = \cot x$ **e.** $f(x) = \log x$ **f.** $f(x) = x^\pi$
- 5.** Determine la serie de Taylor para $\cosh x$ con respecto a cero. Evalúe $\cosh(0.7)$ al sumar cuatro términos. Compare con el valor real.
- 6.** Determine los primeros dos términos distintos de cero del desarrollo de la serie alrededor de cero para las siguientes funciones:
- a.** $e^{\cos x}$ **b.** $\sin(\cos x)$ **c.** $(\cos x)^2(\sin x)$
- 7.** Encuentre el entero no negativo más pequeño m tal que la serie de Taylor alrededor de m para $(x - 1)^{1/2}$ exista. Determine los coeficientes en la serie.
- 8.** Determine cuántos términos se necesitan para calcular e correctamente con 15 cifras decimales (redondeado) mediante la serie (1) para e^x .
- 9.** (Continuación) Si $x < 0$ en el problema anterior, ¿cuáles son los signos de los términos de la serie? La pérdida de dígitos significativos puede ser un serio problema cuando se usa la serie. ¿La fórmula $e^{-x} = 1/e^x$ es útil para reducir el error? Explique. (Véase la sección 2.3 para un análisis adicional.) Trate con aritmética de alta precisión para ver cuán malos pueden ser los errores de punto flotante.
- 10.** Muestre cómo la simple ecuación $\ln 2 = \ln[e(2/e)]$ se puede usar para acelerar el cálculo de $\ln 2$ en la serie (10).
- 11.** ¿Cuál es la serie para $\ln(1 - x)$? ¿Cuál es la serie para $\ln[(1 + x)/(1 - x)]$?
- 12.** (Continuación) En la serie para $\ln[(1 + x)/(1 - x)]$, determine qué valor de x debemos usar si deseamos calcular $\ln 2$. Calcule el número de términos necesarios para diez dígitos (redondeado) de exactitud. ¿Es este método práctico?
- 13.** Use el teorema de series alternantes para determinar el número de términos en la serie (5) que se necesitan para calcular $\ln 1.1$ con error menor que $\frac{1}{2} \times 10^{-8}$.
- 14.** Escriba la serie de Taylor para la función $f(x) = x^3 - 2x^2 + 4x - 1$, usando $x = 2$ como el punto de desarrollo; es decir, escriba una fórmula para $f(2 + h)$.
- 15.** Determine los primeros cuatro términos distintos de cero en el desarrollo de la serie alrededor de cero para:
- a.** $f(x) = (\sin x) + (\cos x)$ y encuentre un valor aproximado para $f(0.001)$.
b. $g(x) = (\sin x)(\cos x)$ y encuentre un valor aproximado para $g(0.0006)$.
 Compare la exactitud de estas aproximaciones con las obtenidas de las tablas o mediante calculadora.
- 16.** Compruebe esta serie de Taylor y demuestre que converge en el intervalo $-e < x \leq e$.

$$\ln(e + x) = 1 + \frac{x}{e} - \frac{x^2}{2e^2} + \frac{x^3}{3e^3} - \frac{x^4}{4e^4} + \cdots = 1 + \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} \left(\frac{x}{e}\right)^k$$

- 17.** Cuántos términos se necesitan en la serie (3) para calcular $\cos x$ para $|x| < \frac{1}{2}$ con una exactitud de 12 cifras decimales (redondeado)?

“18. Una función f está definida por la serie

$$f(x) = \sum_{k=1}^{\infty} (-1)^k \left(\frac{x^k}{k^4} \right)$$

Determine el número mínimo de términos necesarios para calcular $f(1)$ con un error menor de 10^{-8} .

- 19.** Compruebe que las sumas parciales $s_k = \sum_{i=0}^k x^i / i!$ en la serie para e^x , serie (1), se puede rescribir recursivamente como $s_k = s_{k-1} + t_k$, donde $s_0 = 1$, $t_1 = x$ y $t_k = (x/k)t_{k-1}$.
- “20.** ¿Cuál es el quinto término en la serie de Taylor de $(1 - 2h)^{1/2}$?
- 21.** Muestre que si $E = O(h^n)$, entonces $E = O(h^m)$ para cualquier entero no negativo $m \leq n$. Aquí $h \rightarrow 0$.
- 22.** Muestre cómo $p(x) = 6(x+3) + 9(x+3)^5 - 5(x+3)^8 - (x+3)^{11}$ puede ser eficientemente evaluado.
- “23.** ¿Cuál es el segundo término en la serie de Taylor de $\sqrt[4]{4x-1}$ alrededor de 4.25?
- “24.** ¿Cómo podría calcular una tabla de $\log n!$ para $1 \leq n \leq 1000$?
- 25.** Para x pequeña, la aproximación $\sin x \approx x$ se utiliza con frecuencia. ¿Para qué rango de x es bueno esto para una exactitud relativa de $\frac{1}{2} \times 10^{-14}$?
- 26.** En la serie de Taylor para la función $3x^2 - 7 + \cos x$ (desarrollada en potencias de x), ¿cuál es el coeficiente de x^2 ?
- 27.** En la serie de Taylor (alrededor de $\pi/4$) para la función $\sin x + \cos x$, encuentre el tercer término distinto de cero.
- “28.** Usando el teorema de Taylor, uno puede estar seguro de que para toda x que satisface $|x| < \frac{1}{2}$, $|\cos x - (1 - x^2/2)|$ es menor que o igual a ¿cuál valor numérico?
- 29.** Encuentre el valor de ξ que sirve en el teorema de Taylor cuando $f(x) = \sin x$, con $x = \pi/4$, $c = 0$ y $n = 4$.
- 30.** Use el teorema de Taylor para encontrar una función lineal que aproxime mejor $\cos x$ en la vecindad de $x = 5\pi/6$.
- 31.** Para la serie alternante $S_n = \sum_{k=0}^n (-1)^k a_k$, con, $a_0 > a_1 > \dots > 0$, muestre por inducción que $S_0 > S_2 > S_4 > \dots$, que $S_1 < S_3 < S_5 < \dots$, y que $0 < S_{2n} - S_{2n+1} = a_{2n+1}$.
- “32.** ¿Cuál es la serie de Maclaurin para la función $f(x) = 3 + 7x - 1.33x^2 + 19.2x^4$? ¿Cuál es la serie de Taylor para esta función alrededor de $c = 2$?
- 33.** En el libro se afirmó que $\sum_{k=0}^6 x^k / k$ representa e^x sólo en el punto $x = 0$. Demuéstrelo.
- 34.** Determine los primeros tres términos de la serie de Taylor en términos de h para e^{x-h} . Usando tres términos se obtiene $e^{0.999} \approx Ce$, donde C es una constante. Determine C .

- “35.** ¿Cuál es el menor número de términos requerido para calcular π como 3.14 (redondeado) usando la serie

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \dots ?$$

- 36.** Usando el desarrollo de una serie de Taylor en términos de h , determine los primeros tres términos de la serie para $e^{\sin(x+h)}$. Evalúe $e^{\sin 90.01^\circ}$ con exactitud de diez cifras decimales como Ce para la constante C .

- 37.** Desarrolle los primeros dos términos y el error en la serie de Taylor en términos de h para $\ln(3-2h)$.

- “38.** Determine una serie de Taylor para representar $\cos(\pi/3 + h)$. Evalúe $\cos(60.001^\circ)$ con ocho cifras decimales (redondeado). *Sugerencia:* π radianes es igual 180 grados.

- “39.** Determine una serie de Taylor para representar $\sin(\pi/4 + h)$. Evalúe $\sin(45.0005^\circ)$ a nueve cifras decimales (redondeado).

- 40.** Establezca los primeros tres términos en la serie de Taylor para $\csc(\pi/6 + h)$. Calcule $\csc(30.00001^\circ)$ con la misma exactitud que los datos dados.

- 41.** Establezca la serie de Taylor en términos de h para las siguientes:

a. e^{x+2h} b. $\sin(x-3h)$ c. $\ln[(x-h^2)/(x+h^2)]$

- “42.** Determine los primeros tres términos en la serie de Taylor en términos de h para $(x-h)^m$, donde m es un entero constante.

- 43.** Dada la serie

$$-1 + 2^{-4} - 3^{-4} + 4^{-4} - \dots$$

¿cuántos términos se necesitan para obtener cuatro cifras decimales de exactitud (truncado)?

- 44.** Cuántos términos se necesitan en la serie para calcular $\operatorname{arccot} x$ para $x^2 < 1$ con exactitud de 12 cifras decimales (redondeado)?

$$\operatorname{arccot} x = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} + \frac{x^7}{7} - \dots$$

- 45.** Determine los primeros tres términos en la serie de Taylor para representar $\operatorname{senh}(x+h)$. Evalúe $\operatorname{senh}(0.0001)$ con 20 cifras decimales (redondeado) usando esta serie.

- 46.** Determine una serie de Taylor para representar C^{x+h} para C constante. Use la serie para un valor aproximado de $10^{1.0001}$ a cinco cifras decimales (redondeado).

- “47.** La **fórmula de Stirling** establece que $n!$ es mayor que y muy cercano a $\sqrt{2\pi n} n^n e^{-n}$. Use esto para encontrar una n para la que $1/n! < \frac{1}{2} \times 10^{-14}$.

- 48.** Desarrolle los primeros dos términos distintos de cero y el término de error en la serie de Taylor en términos de h para $\ln[1-(h/2)]$. Aproxime $\ln(0.9998)$ usando estos dos términos.

- 49.** La **regla de L'Hôpital** establece que bajo condiciones adecuadas,

$$\lim_{x \rightarrow a} \frac{f(x)}{g(x)} = \frac{f'(a)}{g'(a)}$$

Ésta es verdad, por ejemplo, cuando f y g tienen derivadas continuas en un intervalo abierto que tiene a y $f(a) = g(a) = 0 \neq g'(a)$. Establezca la regla de L'Hôpital usando el teorema del valor medio.

- 50.** (Continuación) Evalúe los siguientes límites numéricamente y use el problema anterior para mostrar que

$$\text{a. } \lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \quad \text{b. } \lim_{x \rightarrow 0} \frac{\arctan x}{x} = 1 \quad \text{c. } \lim_{x \rightarrow \pi} \frac{\cos x + 1}{\sin x} = 0$$

- 51.** Compruebe que si sólo toma los términos de arriba e incluyendo $x^{2n-1}/(2n-1)!$ en la serie (2) para $\sin x$ y si $|x| < \sqrt{6}$, entonces el error implicado no excede a $|x|^{2n+1}/(2n+1)!$. ¿Cuántos términos se necesitan para calcular $\sin(23)$ con un error de a lo más 10^{-8} ? ¿Cuáles problemas prevé al usar la serie para calcular $\sin(23)$? Muestre cómo usar la periodicidad para calcular $\sin(23)$. Demuestre que cada término en la serie se puede obtener del anterior con una simple operación aritmética.

- 52.** Desarrolle la función de error

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

en una serie usando la serie exponencial e integrando. Obtenga directamente la serie de Taylor de $\operatorname{erf}(x)$ alrededor de cero. ¿Son las dos series iguales? Evalúe $\operatorname{erf}(1)$ sumando los cuatro términos de la serie y compare con el valor $\operatorname{erf}(1) \approx 0.8427$, el cual es correcto con cuatro cifras decimales. *Sugerencia:* recuerde del **teorema fundamental del cálculo** que

$$\frac{d}{dx} \int_0^x f(t) dt = f(x)$$

- a53.** Establezca la validez de la serie de Taylor

$$\arctan x = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k-1}}{2k-1} \quad (-1 \leq x \leq 1)$$

¿Es práctico usar directamente esta serie para calcular $\arctan(1)$ si se requieren diez cifras decimales (redondeado) de exactitud? ¿Cuántos términos de la serie se necesitarán? ¿Ocurrirá pérdida de significancia? *Sugerencia:* comience con la serie para $1/(1+x^2)$ e integre término por término. Observe que este procedimiento es sólo formal; la convergencia de la serie resultante se puede demostrar apelando a algunos teoremas de cálculo avanzado.

- a54.** Se sabe que

$$\pi = 4 - 8 \sum_{k=1}^{\infty} (16k^2 - 1)^{-1}$$

Analice los aspectos numéricos de calcular π por medio de esta fórmula. ¿Cuántos términos se necesitarían para producir diez cifras decimales de exactitud (redondeado)?

- 55.** El teorema de Taylor para $f(x)$ desarrollado alrededor de c se expresa con esta ecuación:

$$\begin{aligned} f(x) &= f(c) + (x - c)f'(c) + \frac{1}{2}(x - c)^2 f''(c) + \cdots \\ &\quad + \frac{1}{(n-1)!}(x - c)^{n-1} f^{(n-1)}(c) + \frac{1}{n!}(x - c)^n f^{(n)}(\xi) \end{aligned}$$

Use esta ecuación para determinar cuántos términos en la serie para e^x se necesitan para calcular e con un error a lo más de 10^{-10} . *Sugerencia:* use estos valores aproximados de $n!$: $9! = 3.6 \times 10^5$, $11! = 4.0 \times 10^7$, $12! = 4.8 \times 10^8$, $13! = 6.2 \times 10^9$, $14! = 8.7 \times 10^{10}$ y $15! = 1.3 \times 10^{12}$.

- 56.** a. Repita el ejemplo 3 usando el algoritmo completo de Horner.
 b. Repita el ejemplo 4 usando la serie de Taylor del polinomio $p(x)$.

Problemas de cómputo 1.2

- 1.** Todo el mundo conoce la fórmula cuadrática $(-b \pm \sqrt{b^2 - 4ac})/(2a)$ para obtener las raíces de la ecuación cuadrática $ax^2 + bx + c = 0$. Usando esta fórmula, a mano y con computadora resuelva la ecuación $x^2 + 10^8x + c = 0$ cuando $c = 1$ y 10^8 . Interprete los resultados
- 2.** Use un sistema algebraico computacional para obtener gráficas de las primeras cinco sumas parciales de la serie

$$\arctan x = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^{2k-1}}{2k-1}$$

- 3.** Use un paquete de trazo de gráficas con computadora para reproducir las gráficas de la figura 1.2, así como las siguientes dos sumas parciales, es decir, S_4 y S_5 . Analice los resultados.
- 4.** Use un sistema algebraico computacional para obtener la serie de Taylor dada en las ecuaciones (1)–(5) para obtener la forma final de una vez sin desplegar todas las derivadas.
- 5.** Use dos o más sistemas algebraicos computacionales para hacer el ejemplo 6 con 50 cifras decimales. ¿Son sus respuestas iguales y correctas para todos los dígitos obtenidos? Repita usando \sqrt{x} desarrollando alrededor de $x_0 = 1$.
- 6.** Use un sistema algebraico computacional para comprobar los resultados de los ejemplos 7 y 9.
- 7.** Diseñe y realice un experimento para comprobar el cálculo de x^y en su computadora. *Sugerencia:* compare los cálculos de algunos ejemplos, como $32^{2.5}$ y $81^{1.25}$, con sus valores correctos. Se puede hacer una prueba más elaborada comparando los resultados de simple y doble precisión para diferentes casos.
- 8.** Compruebe que $x^y = e^{y \ln x}$. Trate de encontrar valores de x y y para los cuales estas dos expresiones difieran en su computadora. Interprete los resultados.
- 9.** (Continuación) Para $\cos(x - y) = (\cos x)(\cos y) + (\operatorname{sen} x)(\operatorname{sen} y)$, repita el problema anterior de cómputo.
- 10.** El número de combinaciones de n distintos elementos tomando m en un tiempo está dado por el **coeficiente binomial**

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

para enteros m y n , con $0 \leq m \leq n$. Recuerde que $\binom{n}{0} = \binom{n}{n} = 1$.

a. Escriba

integer function $ibin(n, m)$

que usa la definición anterior para calcular $\binom{n}{m}$.

b. Compruebe la fórmula

$$\binom{n}{m} = \prod_{k=1}^{\min(m, n-m)} \left[\frac{n-k+1}{k} \right]$$

calculando los coeficientes binomiales. Escriba

integer function $jbin(n, m)$

que está basada en esta fórmula.

c. Compruebe las fórmulas (**triángulo de Pascal**)

$$\begin{cases} a_{i0} = a_{ii} = 1 & (0 \leq i \leq n) \\ a_{ij} = a_{i-1, j-1} + a_{i-1, j} & (2 \leq i \leq n, 1 \leq j \leq i-1) \end{cases}$$

Usando el triángulo de Pascal, calcule los coeficientes binomiales

$$\binom{i}{j} = a_{i,j} \quad (0 \leq i, j \leq n)$$

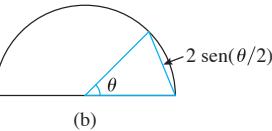
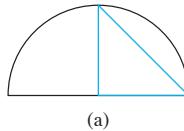
y almacénelos en la parte triangular inferior del arreglo $(a_{ij})_{n \times n}$. Escriba

integer function $kbin(n, m)$

que hace que un arreglo busque después de la primera asignación y calcule las entradas del arreglo.

11. La longitud de la parte curva de un semicírculo unitario es π . Podemos aproximar a π usando triángulos y matemática elemental. Considere el semicírculo con el arco bisecado como se muestra en la figura (a). La hipotenusa del triángulo rectángulo es $\sqrt{2}$. Por tanto, una burda aproximación a π está dada por $2\sqrt{2} \approx 2.8284$. En la figura (b), consideraremos un ángulo θ que es una fracción l/k del semicírculo. La secante que se muestra tiene longitud $2 \operatorname{sen}(\theta/2)$ y así una aproximación de π es $2k \operatorname{sen}(\theta/2)$. De trigonometría, tenemos

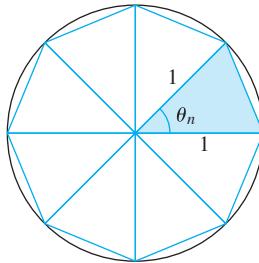
$$\operatorname{sen}^2 \frac{1}{2}\theta = \frac{1}{2}(1 - \cos \theta) = \frac{1}{2} \left(1 - \sqrt{1 - \operatorname{sen}^2 \theta} \right) = \frac{\operatorname{sen}^2 \theta}{2 + 2\sqrt{1 - \operatorname{sen}^2 \theta}}$$



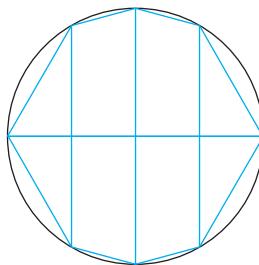
Ahora sea θ_n el ángulo que resulta de la división del arco semicircular en 2^{n-1} partes. Después sea $S_n = \operatorname{sen}^2 \theta_n$ y $P_n = 2^n \sqrt{S_{n+1}}$. Muestre que $S_{n+1} = S_n / (2 + 2\sqrt{1 - S_n})$ y P_n es una aproximación a π . Comenzando con $S_2 = 1$ y $P_1 = 2$, calcule S_{n+1} y P_n recursivamente para $2 \leq n \leq 20$.

12. El número irracional π se puede calcular approximando el área de un círculo unitario como el límite de una sucesión p_1, p_2, \dots , descrita en la forma siguiente. Divida el círculo unitario en 2^n sectores. (La figura muestra el caso $n = 3$.) Aproxime el área del sector por el

área del triángulo isósceles. El ángulo θ_n es $2\pi/2^n$. El área del triángulo es $\frac{1}{2} \sin \theta_n$. (Compruebe.) La enésima aproximación a π es entonces $p_n = 2^{n-1} \sin \theta_n$. Pruebe que $\sin \theta_n = \sin \theta_{n-1} / \{2[1 + (1 - \sin^2 \theta_{n-1})^{1/2}]\}^{1/2}$ por medio de identidades trigonométricas bien conocidas. Use esta relación de recurrencia para generar las sucesiones $\sin \theta_n$ y p_n ($3 \leq n \leq 20$) iniciando con $\sin \theta_2 = 1$. Compare con el cálculo de $4.0 \arctan(1.0)$.



- 13.** (Continuación) Calcule π por un método similar al del problema de cómputo anterior, donde el área del círculo unitario se approxima con una sucesión de trapecios como se muestra en la figura.



- 14.** Escriba una rutina con precisión doble o extendida para implementar los siguientes algoritmos para calcular π .

```

integer k; real a, b, c, d, e, f, g
a ← 0
b ← 1
c ← 1/√2
d ← 0.25
e ← 1
for k = 1 to 5 do
    a ← b
    b ← (b + c)/2
    c ← √ca
    d ← d - e(b - a)2
    e ← 2e
    f ← b2/d
    g ← (b + c)2/(4d)
    output k, f, |f - π|, g, |g - π|
end for
```

¿Cuál converge más rápidamente, f o g ? ¿Qué exactitud tienen los valores finales? También compare con los cálculos de precisión doble o extendida de $4.0 \arctan(1.0)$. *Sugerencia:* el valor correcto de π a 36 dígitos es

$$3.14159\ 26535\ 89793\ 23846\ 26433\ 83279\ 50288$$

Nota: una nueva fórmula para calcular π fue descubierta a principios de la década de 1970. Este algoritmo se basa en dicha fórmula, que es una consecuencia directa de un método desarrollado por Gauss para calcular integrales elípticas y de la relación de las integrales elípticas de Legendre, ¡ambas conocidas por más de 150 años! El análisis de error muestra que la convergencia rápida ocurre en el cálculo de π y el número de dígitos significativos se duplica después de cada paso. (El lector interesado debe consultar Brent [1976], Borwein y Borwein [1987] y Salamin [1976].)

- 15.** Otro esquema cuadráticamente convergente para calcular π fue descubierto por Borwein y Borwein [1984] y se puede reescribir como

```
integer k; real a, b, t, x
a ← √2
b ← 0
x ← 2 + √2
for k = 1 to 5 do
    t ← √a
    b ← t(1 + b)/(a + b)
    a ← ½(t + 1/t)
    x ← xb(1 + a)/(1 + b)
    output k, x, |x - π|
end for
```

Numéricamente compruebe que $|x - \pi| \leq 10^{-2k}$. *Nota:* Ludolf van Ceulen (1540–1610) fue capaz de calcular π con 36 dígitos. Con modernos paquetes de software matemático como Matlab, Maple y Mathematica, ¡cualquiera puede fácilmente calcular π con decenas de miles de dígitos en segundos!

- 16.** La sucesión de Fibonacci $1, 1, 2, 3, 5, 8, 13, 21, \dots$ está definida por la relación de recurrencia lineal

$$\begin{cases} \lambda_1 = 1 & \lambda_2 = 1 \\ \lambda_n = \lambda_{n-1} + \lambda_{n-2} & (n \geq 3) \end{cases}$$

Una fórmula para obtener el enésimo **número de Fibonacci** es

$$\lambda_n = \frac{1}{\sqrt{5}} \left\{ \left[\frac{1}{2}(1 + \sqrt{5}) \right]^n - \left[\frac{1}{2}(1 - \sqrt{5}) \right]^n \right\}$$

Calcule $\ln(1 \leq n \leq 50)$ usando tanto la relación de recurrencia como la fórmula. Escriba tres programas que usen aritmética entera, de precisión simple y de doble precisión, respectivamente. Para cada n , imprima los resultados usando formatos de entero, de precisión simple y de doble precisión, respectivamente.

17. (Continuación) Repita el experimento usando la sucesión dada por la relación de recurrencia

$$\begin{cases} \alpha_1 = 1 & \alpha_2 = \frac{1}{2}(1 + \sqrt{5}) \\ \alpha_n = \alpha_{n-1} + \alpha_{n-2} & (n \geq 3) \end{cases}$$

Una fórmula de forma cerrada es

$$\alpha_n = \left[\frac{1}{2}(1 + \sqrt{5}) \right]^n$$

18. (Continuación) Cambie $+\sqrt{5}$ por $-\sqrt{5}$ y repita el cálculo de α_n . Explique los resultados.

19. Las **funciones de Bessel** J_n están definidas por

$$J_n(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin \theta - n\theta) d\theta$$

Establezca que $|J_n(x)| \leq 1$.

a. Se sabe que $J_{n+1}(x) = 2nx^{-1}J_n(x) - J_{n-1}(x)$. Use esta ecuación para calcular $J_0(1), J_1(1), \dots, J_{20}(1)$, comenzando con los valores conocidos $J_0(1) \approx 0.7651976865$ y $J_1(1) \approx 0.4400505857$. Explique el hecho de que se viola la desigualdad $|J_n(x)| \leq 1$.

b. Otra relación de recurrencia es $J_{n-1}(x) = 2nx^{-1}J_n(x) - J_{n+1}(x)$. Comenzando con los valores conocidos $J_{20}(1) \approx 3.873503009 \times 10^{-25}$ y $J_{19}(1) \approx 1.548478441 \times 10^{-23}$ use esta ecuación para calcular $J_{18}(1), J_{17}(1), \dots, J_1(1), J_0(1)$. Analice los resultados.

20. A un estudiante de cálculo se le pide que determine $\lim_{n \rightarrow \infty} (100^n/n!)$ y escriba un programa para evaluar $x_0, x_1, x_2, \dots, x_n$ como se muestra a continuación:

```
integer parameter n ← 100
integer i; real x; x ← 1
for i = 1 to n do
    x ← 100x/i
    output i, x
end for
```

Los números impresos serán siempre más grandes y el estudiante concluye que $\lim_{n \rightarrow \infty} x_n = \infty$. ¿Cuál es la moraleja aquí?

21. (Aproximaciones de funciones con series de Maclaurin) Usando la serie de Maclaurin truncada, una función $f(x)$ con n derivadas continuas se puede aproximar con un polinomio de enésimo grado

$$f(x) \approx p_n(x) = \sum_{i=0}^n c_i x^i$$

donde $c_i = f^{(i)}(0)/i!$.

- a.** Genere y compare gráficas de computadora para $f(x) = e^x$ y los polinomios $p_2(x)$, $p_3(x)$, $p_4(x)$, $p_5(x)$. ¿Los polinomios de orden superior aproximan la función exponencial e^x satisfactoriamente al aumentar los intervalos alrededor de cero?
- b.** Repita para $g(x) = \ln(1 + x)$.
- 22.** (Continuación, Aproximaciones racionales de Padé) La **aproximación racional de Padé** es la *mejor* aproximación de un función con una función racional de un orden dado. Con frecuencia ésta da una mejor aproximación de la función que el truncamiento de su serie de Taylor y ¡puede funcionar aún cuando la serie de Taylor no converja! Por consiguiente, las aproximaciones racionales de Padé con frecuencia se usan en cálculos con computadora tales como para la función básica $\sin x$ como se analizó en el problema de cómputo 2.2.17. Más que usar polinomios de orden superior, usamos cocientes de polinomios de orden bajo, que son llamados **aproximaciones racionales**. Sea

$$f(x) \approx \frac{p_m(x)}{q_k(x)} = \frac{\sum_{i=0}^m a_i x^i}{\sum_{j=0}^k b_j x^j} = R_{m,k}(x)$$

donde $b_0 = 1$. Aquí hemos normalizado con respecto a $b_0 \neq 0$ y los valores de m y k son modestos. Elegimos los k coeficientes b_j y los $m + 1$ coeficientes a_i en $R_{m,k}$ para acoplar f y un número dado de sus derivadas en el punto fijo $x = 0$. Primero, construimos la serie truncada de Maclaurin $\sum_{i=0}^n c_i x^i$ en la que $c_i = f^{(i)}(0)/i!$ y $c_i = 0$ para $i < 0$. Después, acoplamos las primeras $m + k + 1$ derivadas de $R_{m,k}$ con respecto a x en $x = 0$ en los primeros $m + k + 1$ coeficientes c_i . Ello nos conduce a las siguientes ecuaciones desplegadas. Puesto que $b_0 = 1$, resolvemos este sistema de ecuaciones $k \times k$ para b_1, b_2, \dots, b_k

$$\begin{bmatrix} c_m & c_{m-1} & \cdots & c_{m-(k-2)} & c_{m-(k-1)} \\ c_{m+1} & c_m & \cdots & c_{m-(k-3)} & c_{m-(k-2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m+(k-2)} & c_{m+(k-3)} & \cdots & c_m & c_{m-1} \\ c_{m+(k-1)} & c_{m+(k-2)} & \cdots & c_{m+1} & c_m \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{k-1} \\ b_k \end{bmatrix} = \begin{bmatrix} -c_{m+1} \\ -c_{m+2} \\ \vdots \\ -c_{m+(k-1)} \\ -c_{m+k} \end{bmatrix}$$

(La solución de sistemas de ecuaciones lineales numéricamente se analiza en los capítulos 7 y 8.) Por último, evaluamos estas $m + 1$ ecuaciones para a_0, a_1, \dots, a_m :

$$a_j = \sum_{\ell=0}^j c_{j-\ell} b_\ell \quad (j = 0, 1, \dots, m)$$

Observe que $a_j = 0$ para $j > m$ y $b_j = 0$ para $j > k$. También, si $k = 0$, entonces $R_{m,0}$ es una serie truncada de Maclaurin para f . Además, las aproximaciones de Padé pueden tener singularidades.

- a.** Determine las funciones racionales $R_{1,1}(x)$ y $R_{2,2}(x)$. Genere y compare gráficas de computadora para $f(x) = e^x$, $R_{1,1}$ y $R_{2,2}$. ¿Estas funciones racionales de bajo orden aproximan la función exponencial e^x satisfactoriamente dentro de $[-1, 1]$? ¿Cómo se comparan con los polinomios truncados de Maclaurin del problema anterior?
- b.** Repita usando $R_{2,2}(x)$ y $R_{3,1}(x)$ para la función $g(x) = \ln(1 + x)$.

Información acerca de la vida y del trabajo del matemático francés Herni Eugéne Padé (1863-1953) se puede encontrar en Wood [1999]. Esta referencia también tiene ejemplos y ejercicios similares a estos. Se pueden ver más ejemplos de la aproximación de Padé.

- 23.** (Continuación) Repita para la función de Bessel $J_0(2x)$, cuya serie de Maclaurin es

$$1 - x^2 + \frac{x^4}{4} - \frac{x^6}{36} + \cdots = \sum_{i=0}^{\infty} (-1)^i \left(\frac{x^i}{i!}\right)^2$$

Después determine $R_{2,2}(x)$, $R_{4,3}(x)$ y $R_{2,4}(x)$ y también compare sus gráficas.

- 24.** Lleve a cabo los detalles del ejemplo de introducción de este capítulo primero desarrollando la serie de Taylor para $\ln(1+x)$ y calculando $\ln 2 \approx 0.63452$ usando los primeros ocho términos. Después establezca la serie $\ln[(1+x)/(1-x)]$ y calcule $\ln 2 \approx 0.69313$ usando los términos que se muestran. Determine el error absoluto y el error relativo para estas respuestas.
- 25.** Reproduzca la figura 1.2 usando su computadora, así como sumando la curva para S_4 .
- 26.** Use un software de matemáticas que realice manejos simbólicos como Maple o Mathematica para realizar
- a.** Ejemplo 3 **b.** Ejemplo 6

- 27.** ¿Puede obtener los siguientes resultados numéricos?

$$\begin{aligned}\sqrt{1.00001} &= 1.00000\ 49999\ 87500\ 06249\ 96093\ 77734\ 37500\ 0000 \\ \sqrt{0.99999} &= 0.99999\ 49999\ 87499\ 93749\ 96093\ 72265\ 62500\ 00000\end{aligned}$$

¿Son estas respuestas exactas a todos los dígitos que se muestran?

Representación de punto flotante y errores

Las computadoras normalmente no usan aritmética de base 10 para almacenar o calcular. Los números que tienen una expresión finita en un sistema numérico pueden tener una expresión infinita en otro. Este fenómeno se presenta cuando el conocido número decimal 1/10 se convierte en el sistema binario:

$$(0.1)_{10} = (0.00011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ \dots)_2$$

En este capítulo explicamos el sistema de punto flotante y desarrollamos conceptos básicos acerca de los errores de redondeo. Otro tema es la pérdida de significancia, que ocurre cuando se restan números casi iguales. Se estudia y se muestran varias técnicas de programación para evitarla.

2.1 Representación de punto flotante

La forma normal de representar un número real no negativo en forma decimal es con una parte entera y una parte fraccionaria, y un punto decimal entre ellas, por ejemplo, 37.21829, 0.00227 1828 y 30 00527.11059. Otra forma común, con frecuencia llamada **notación científica normalizada**, se obtiene corriendo el punto decimal y proporcionando adecuadas potencias de 10. Por tanto, los números anteriores tienen como representaciones alternativas

$$\begin{aligned} 37.21829 &= 0.3721829 \times 10^2 \\ 0.00227\ 1828 &= 0.2271828 \times 10^{-2} \\ 30\ 00527.11059 &= 0.30005\ 2711059 \times 10^7 \end{aligned}$$

En notación científica normalizada, el número se representa con una fracción multiplicada por 10^n , y el dígito principal en la fracción no es igual a cero (excepto cuando el número implicado es cero). Así, escribimos 79325 como 0.79325×10^5 , no 0.079325×10^6 o 7.9325×10^4 o de alguna otra forma.

Representación de punto flotante normalizada

En el contexto de la ciencia computacional, la notación científica normalizada también se llama **representación de punto flotante normalizada**. En el sistema decimal, cualquier número real x (diferente de cero) se puede representar en la forma de punto flotante normalizada como

$$x = \pm 0.d_1 d_2 d_3 \dots \times 10^n$$

donde $d_1 \neq 0$ y n es un entero (positivo, negativo o cero). Los números d_1, d_2, \dots son los dígitos decimales 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9.

Dicho de otra forma, el número real x , si es diferente de cero, se puede representar en la forma decimal de punto flotante normalizada como

$$x = \pm r \times 10^n \quad \left(\frac{1}{10} \leq r < 1 \right)$$

Esta representación consta de tres partes: un signo ya sea + o -, un número r en el intervalo $\left[\frac{1}{10}, 1 \right)$, y una potencia entera de 10. El número r se llama la **mantisa normalizada** y n el **exponente**.

La representación de punto flotante en el sistema binario es similar a la del sistema decimal en diferentes formas. Si $x \neq 0$, se puede escribir como

$$x = \pm q \times 2^m \quad \left(\frac{1}{2} \leq q < 1 \right)$$

La mantisa q se podría expresar como una sucesión de ceros o unos en el forma $q = (0.b_1 b_2 b_3 \dots)_2$ donde $b_1 \neq 0$. Por tanto, $b_1 = 1$ y entonces necesariamente $q \geq \frac{1}{2}$.

Un sistema numérico de punto flotante dentro de una computadora es similar al que acabamos de describir, con una diferencia importante: cada computadora tiene sólo una longitud de palabra finita y una capacidad total finita, por lo que sólo se pueden representar números con un número finito de dígitos. A un número se le asigna sólo una palabra en el modo de precisión simple (dos o más palabras en precisión doble o extendida). En cualquier caso, el grado de precisión está estrictamente limitado. Obviamente, los números irracionales no se pueden representar, ni se pueden representar tampoco los números racionales que no se ajusten al formato finito impuesto por la computadora. Además, los números pueden ser demasiado grandes o demasiado pequeños para poderse representar. Los números reales que se pueden representar en una computadora se llaman sus **números de máquina**.

Puesto que cualquier número usado en cálculos con un sistema computacional debe adaptarse al formato de números en el sistema, éstos deben tener una **expansión finita**. Los números que tienen una expansión que no termina no se pueden alojar exactamente. Además, un número que tiene una expansión que termina en una base puede tener una expansión que no termina en otra. Un buen ejemplo de esto es la siguiente fracción simple que se presentó en la introducción a este capítulo.

$$\begin{aligned} \frac{1}{10} &= (0.1)_{10} = (0.06314631463146314 \dots)_8 \\ &= (0.000110011001100110011001100110011 \dots)_2 \end{aligned}$$

El punto importante aquí es que la mayoría de los números reales no se pueden representar exactamente con una computadora. (Véase el apéndice B para un análisis de la representación de números en diferentes bases.)

El sistema numérico real de una computadora *no* es un continuo sino más bien un conjunto discreto peculiar. Para mostrar esto, consideremos un ejemplo extremo, en el que los números de punto flotante deben ser de la forma $x = \pm(0.b_1 b_2 b_3)_2 \times 2^{ \pm k }$, donde b_1, b_2, b_3 y m sólo pueden adoptar los valor 0 o 1.

EJEMPLO 1 Liste todos los números de punto flotante que se pueden expresar en la forma

$$x = \pm(0.b_1b_2b_3)_2 \times 2^{\pm k} \quad (k, b_i \in \{0, 1\})$$

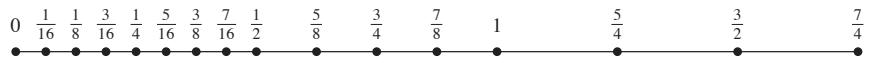
Solución Hay dos opciones para el \pm , dos opciones de b_1 , dos opciones para b_2 , dos elecciones para b_3 y tres opciones para el exponente. Así, en principio, se esperarían $2 \times 2 \times 2 \times 2 \times 3 = 48$ números. Sin embargo, hay algunos duplicados. Por ejemplo, los números no negativos en este sistema son los siguientes:

0.000 $\times 2^0 = 0$	0.000 $\times 2^1 = 0$	0.000 $\times 2^{-1} = 0$
$0.001 \times 2^0 = \frac{1}{8}$	$0.001 \times 2^1 = \frac{1}{4}$	$0.001 \times 2^{-1} = \frac{1}{16}$
$0.010 \times 2^0 = \frac{2}{8}$	$0.010 \times 2^1 = \frac{2}{4}$	$0.010 \times 2^{-1} = \frac{2}{16}$
$0.011 \times 2^0 = \frac{3}{8}$	$0.011 \times 2^1 = \frac{3}{4}$	$0.011 \times 2^{-1} = \frac{3}{16}$
$0.100 \times 2^0 = \frac{4}{8}$	$0.100 \times 2^1 = \frac{4}{4}$	$0.100 \times 2^{-1} = \frac{4}{16}$
$0.101 \times 2^0 = \frac{5}{8}$	$0.101 \times 2^1 = \frac{5}{4}$	$0.101 \times 2^{-1} = \frac{5}{16}$
$0.110 \times 2^0 = \frac{6}{8}$	$0.110 \times 2^1 = \frac{6}{4}$	$0.110 \times 2^{-1} = \frac{6}{16}$
$0.111 \times 2^0 = \frac{7}{8}$	$0.111 \times 2^1 = \frac{7}{4}$	$0.111 \times 2^{-1} = \frac{7}{16}$

En total hay 31 números en el sistema. En la figura 2.1 se muestran en una recta los números positivos obtenidos. Observe que los números están distribuidos simétricamente pero en forma desigual alrededor de cero.

FIGURA 2.1

Números de máquina positivos del ejemplo 1



Si, en el curso de un cálculo, se produce un número x de la forma $\pm q \times 2^m$, donde m está fuera del rango permisible de la computadora, entonces decimos que ha ocurrido un **sobreflujo** o un **subflujo** o que x está **fuera del rango de la computadora**. En general, un sobreflujo da como resultado un error fatal (*o excepción*) y la ejecución normal del programa se detiene. Sin embargo, un subflujo normalmente se trata automáticamente al hacer x igual a cero sin ninguna interrupción del programa pero con un mensaje de advertencia en la mayoría de las computadoras.

En una computadora cuyos números de punto flotante están restringidos a la forma del ejemplo 1, cualquier número cercano a cero como $\frac{1}{16}$ podría hacer un *subflujo* a cero y cualquier número fuera del rango $-1.75 < x < +1.75$ podría hacer un *sobreflujo* a un infinito de máquina.

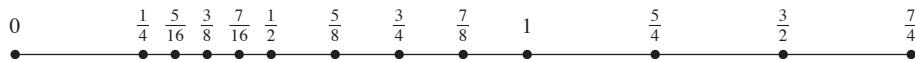
Si, en el ejemplo 1, se nos permiten sólo números *normalizados* de punto flotante, entonces todos nuestros números (con excepción del cero) tienen la forma

$$x = \pm(0.1b_2b_3)_2 \times 2^{\pm k}$$

Esto crea un fenómeno conocido como el **hoyo en cero**. Nuestros números no negativos de máquina están ahora distribuidos como se muestra en la figura 2.2. Hay una separación relativamente amplia entre cero y el más pequeño número de máquina positivo, que es $(0.100)_2 \times 2^{-1} = \frac{1}{4}$.

FIGURA 2.2

Números de máquina normalizados del ejemplo 1



Representación de punto flotante

Una computadora que funciona en modo de punto flotante representa números como se describió antes excepto por las limitaciones impuestas por la palabra de longitud finita. Muchas computadoras binarias tienen una longitud de palabra de 32 bits (dígitos binarios). Describiremos una máquina de este tipo cuyas características imitan muchas estaciones de trabajo y computadoras personales de uso generalizado. La representación interna de números y su almacenamiento es en la **forma de punto flotante estándar** para casi todas las computadoras. Por simplicidad, hemos omitido un análisis de algunos de los detalles y características. Afortunadamente, no se necesitan conocer todos los detalles del sistema aritmético de punto flotante que se usa en una computadora para emplearlo inteligentemente. No obstante, generalmente al depurar un programa resulta provechoso tener un entendimiento básico de la representación de números en su computadora.

Por **números de precisión simple de punto flotante** entendemos todos los números aceptables en una computadora que usan el formato aritmético de punto flotante de precisión simple estándar. (En este análisis, hemos supuesto que dicha computadora almacena estos números en palabras de 32 bits). Este conjunto es un subconjunto finito de los números reales. Consta de ± 0 , $\pm \infty$, números de punto flotante de precisión simple normal y subnormal, pero no valores numéricos (NaN). (En el apéndice B y en las referencias se presentan más detalles acerca de estos temas.) Recuerde que la mayoría de los números reales *no se pueden* representar exactamente como números de punto flotante, ya que tienen una expansión decimal infinita o binaria (todos los números irracionales y algunos números racionales); por ejemplo, π , e , $\frac{1}{3}$, 0.1, etcétera.

Debido a la longitud de palabra de 32 bits, tanto como sea posible el número de punto flotante normalizado

$$\pm q \times 2^m$$

debe estar contenido en esos 32 bits. Una forma de asignar espacio a los 32 bits es la siguiente

signo de q	1 bit
entero $ m $	8 bits
número q	23 bits

La información del signo de m está contenida en los ocho bits asignados al entero $|m|$. En este esquema, podemos representar números reales con $|m|$ tan grande como $2^7 - 1 = 127$. El exponente representa números de -127 a 128 .

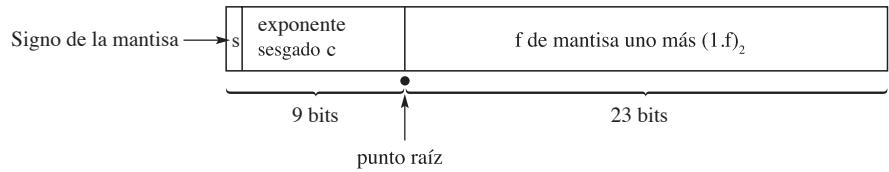
Forma de punto flotante de precisión simple

Ahora describimos un número de máquina de la siguiente forma en la representación de **punto flotante de precisión simple estándar**:

$$(-1)^s \times 2^{c-127} \times (1.f)_2$$

El bit que está más a la izquierda se utiliza para el signo de la mantisa, donde $s = 0$ corresponde a $+$ y $s = 1$ a $-$. Los siguientes ocho bits se utilizan para representar al número c en el

FIGURA 2.3
Palabra de computadora de punto flotante y precisión simple particionada



exponente de 2^{c-127} , que se interpreta como un *código de exceso 127*. Por último, los 23 bits finales representan f de la parte fraccionaria de la mantisa en la forma uno más: (1.f)₂. Cada palabra de punto flotante y precisión simple se partitiona como se muestra en la figura 2.3.

En la representación normalizada de un número de punto flotante distinto de cero, el primer bit en la mantisa es *siempre 1*, por lo que este bit no se tiene que almacenar. Esto se puede lograr corriendo el punto binario a la forma “uno más” (1.f)₂. La mantisa son los 23 bits que están más a la derecha y contiene f con un punto binario sobreentendido como se muestra en la figura 2.3. Por ello, la mantisa (**significando**) *realmente* corresponde a 24 dígitos binarios, ya que hay un **bit escondido**. (Una excepción importante es el número ± 0 .)

Ahora bosquejamos el procedimiento para determinar la representación de un número real x. Si x es cero, se representa por medio de una palabra completa de cero bits con la posible excepción del bit del signo. Para una x distinta de cero, primero asignamos el bit del signo para x y consideramos |x|. Despues convertimos las partes entera y fraccionaria de |x| de decimal a binaria. Despues uno más normaliza (|x|)₂ al correr el punto binario, por lo que el primer bit a la izquierda del punto binario es un 1 y todos los bits a la izquierda de este 1 son 0. Para compensar este corrimiento del punto binario, ajustamos el exponente de 2; es decir, multiplicamos por la adecuada potencia de 2. Así se encuentra la mantisa del binario de 24 bits uno más normalizada. Ahora el exponente actual de 2 deben igualarse a c - 127 para determinar c, que se convierte entonces de decimal a binario. El bit del signo de la mantisa se combina con (c)₂ y (f)₂. Por último escribimos la representación de 32 bits de x como ocho dígitos hexadecimales.

El valor de c en la representación de un número de punto flotante con precisión simple está restringido por la desigualdad

$$0 < c < (11111111)_2 = 255$$

Los valores 0 y 255 están reservados para casos especiales, que incluyen ± 0 y $\pm\infty$, respectivamente. Por tanto, el exponente real del número está restringido por la desigualdad

$$-126 \leq c - 127 \leq 127$$

Asimismo, encontramos que la mantisa de cada número diferente de cero está restringida por la desigualdad

$$1 \leq (1.f)_2 \leq (1.1111111111111111111111)_2 = 2 - 2^{-23}$$

El número más grande que se puede representar es, por tanto, $(2 - 2^{-23})2^{127} \approx 2^{128} \approx 3.4 \times 10^{38}$. El número positivo más pequeño es $2^{-126} \approx 1.2 \times 10^{-38}$.

El número de máquina binario de punto flotante $\epsilon = 2^{-23}$ se llama **épsilon de máquina** cuando se usa precisión simple. Es el número de máquina positivo más pequeño ϵ tal que $1 + \epsilon \neq 1$. Ya que $2^{-23} \approx 1.2 \times 10^{-7}$, inferimos que en un cálculo simple, aproximadamente seis dígitos decimales significativos de exactitud se pueden obtener con precisión simple. Recuerde que 23 bits están asignados para la mantisa.

Forma de punto flotante de doble precisión

Cuando se necesita más precisión se puede usar **doble precisión** y en este caso cada número de punto flotante de doble precisión se almacena en la memoria en dos palabras de computadora. En la doble precisión, hay 52 bits asignados para la mantisa. El **épsilon de máquina** de doble precisión es $2^{-52} \approx 2.2 \times 10^{-16}$, por lo que aproximadamente están disponibles 15 dígitos significativos decimales de precisión. Hay 11 bits permitidos para el exponente, que está sesgado por 1023. El exponente representa números de -1022 a 1023. Un número de máquina en forma de punto **flotante de doble precisión estándar** corresponde a

$$(-1)^s \times 2^{c-1023} \times (1.f)_2$$

El bit del extremo izquierdo se usa para el signo de la mantisa con $s = 0$ para + y $s = 1$ para -. Los siguientes once bits se utilizan para representar el exponente c correspondiente a 2^{c-1023} . Por último, 52 bits representan f de la parte fraccionaria de la mantisa en la forma uno más: $(1.f)_2$.

El valor de c en la representación de un número de punto flotante de doble precisión está restringido por la desigualdad

$$0 < c < (1\ 111\ 111\ 111)_2 = 2047$$

Como con precisión simple, los valores en los extremos de este intervalo están reservados para casos especiales. Por tanto, el exponente real del número está restringido por la desigualdad

$$-1022 \leq c - 1023 \leq 1023$$

Encontramos que la mantisa de cada número distinto de cero está restringida por la desigualdad

$$1 \leq (1.f)_2 \leq (1.111\ 111\ 111 \cdots 111\ 111\ 111\ 1)_2 = 2 - 2^{-52}$$

Como $2^{-52} \approx 1.2 \times 10^{-16}$, inferimos que en un cálculo simple se pueden obtener aproximadamente 15 dígitos significativos decimales de exactitud en doble precisión. Recuerde que 52 bits están asignados para la mantisa. El número de máquina de doble precisión más grande es $(2 - 2^{-52})2^{1023} \approx 2^{1024} \approx 1.8 \times 10^{308}$. El número de máquina positivo de doble precisión más pequeño es $2^{-1022} \approx 2.2 \times 10^{-308}$.

La precisión simple en una computadora de 64 bits es comparable a la doble precisión de una computadora de 32 bits, mientras que la doble precisión en una computadora de 64 bits da cuatro veces la precisión disponible en una computadora de 32 bits.

Con precisión simple, están disponibles 31 bits para un entero, ya que sólo se necesita 1 bit para el signo. Así, el rango para enteros es de $-(2^{31} - 1)$ a $(2^{31} - 1) = 21474\ 83647$. Con doble precisión, se utilizan 63 bits para enteros, lo que da enteros en el rango de $-(2^{63} - 1)$ a $(2^{63} - 1)$. Al usar aritmética para enteros, los cálculos exactos pueden dar como resultado sólo aproximaciones de nueve dígitos con precisión simple y de 18 dígitos con doble precisión! Para una exactitud alta, la mayoría de los cálculos se deben hacer usando aritmética de punto flotante con doble precisión.

EJEMPLO 2 Determine la precisión simple de la representación de máquina del número decimal -52.23437 5 tanto con precisión simple como doble.

Solución Convirtiendo la parte entera a binaria, tenemos $(52.)_{10} = (64.)_8 = (110\ 100.)_2$. Después, al convertir la parte fraccionaria, tenemos $(.23437\ 5)_{10} = (.17)_8 = (.001\ 111)_2$. Ahora

$$(52.23437\ 5)_{10} = (110\ 100.001\ 111)_2 = (1.101\ 000\ 011\ 110)_2 \times 2^5$$

es la forma uno más correspondiente de base 2, y $(.101\ 000\ 011\ 110)_2$ es la mantisa almacenada. Después el exponente es $(5)_{10}$ y puesto que $c - 127 = 5$, vemos inmediatamente que $(132)_{10} = (204)_8 = (10\ 000\ 100)_2$, es el exponente almacenado. Así, la representación de máquina de precisión simple de -52.234375 es

$$\begin{aligned}[1\ 10\ 000\ 100\ 101\ 000\ 011\ 110\ 000\ 000\ 000\ 000\ 00]_2 &= \\ [1100\ 0010\ 0101\ 0000\ 1111\ 0000\ 0000\ 0000]_2 &= [C250F000]_{16}\end{aligned}$$

Con doble precisión, para el exponente $(5)_{10}$ hacemos $c - 1023 = 5$ y tenemos $(1028)_{10} = (2004)_8 = (10\ 000\ 000\ 100)_2$, que es el exponente almacenado. Así, la representación de máquina de doble precisión de -52.234375 es

$$\begin{aligned}[1\ 10\ 000\ 000\ 100\ 101\ 000\ 011\ 110\ 000\ \cdots\ 00]_2 &= \\ [1100\ 0000\ 0100\ 1010\ 0001\ 1110\ 0000\ \cdots\ 0000]_2 &= [C04A1E0000000000]_{16}\end{aligned}$$

Aquí $[...]_k$ es el patrón de bits de la(s) palabra(s) de la máquina que representa números de punto flotante, que se presentan con base k . ■

EJEMPLO 3 Determine los números decimales que corresponden a estas palabras de máquina:

$$[45DE4000]_{16} \quad [BA390000]_{16}$$

Solución El primer número en binario es

$$[0100\ 0101\ 1101\ 1110\ 0100\ 0000\ 0000\ 0000]_2$$

El exponente almacenado es $(10\ 001\ 011)_2 = (213)_8 = (139)_{10}$, así $139 - 127 = 12$. La mantisa es positiva y representa el número

$$\begin{aligned}(1.101\ 111\ 001)_2 \times 2^{12} &= (1\ 101\ 111\ 001\ 000.)_2 \\ &= (15710.)_8 \\ &= 0 \times 1 + 1 \times 8 + 7 \times 8^2 + 5 \times 8^3 + 1 \times 8^4 \\ &= 8(1 + 8(7 + 8(5 + 8(1)))) \\ &= 7112\end{aligned}$$

De forma similar, la segunda palabra en binario es

$$[1011\ 1010\ 0011\ 1001\ 0000\ 0000\ 0000\ 0000]_2$$

La parte exponencial de la palabra es $(01\ 110\ 100)_2 = (164)_8 = 116$, de manera que el exponente es $116 - 127 = -11$. La mantisa es negativa y corresponde al siguiente número de punto flotante:

$$\begin{aligned}- (1.011\ 100\ 100)_2 \times 2^{-11} &= -(0.000\ 000\ 000\ 010\ 111\ 001)_2 \\ &= -(0.000271)_8 \\ &= -2 \times 8^{-4} - 7 \times 8^{-5} - 1 \times 8^{-6} \\ &= -8^{-6}(1 + 8(7 + 8(2))) \\ &= -\frac{185}{262144} \approx -7.0571899 \times 10^{-4}\end{aligned}$$
■

Errores de cómputo en la representación de números

Ahora regresamos a los errores que pueden ocurrir cuando tratamos de representar un número real dado x en la computadora. Usamos una computadora modelo con una longitud de palabra de 32 bits. Suponemos primero que hacemos $x = 2^{53.21697}$ o $x = 2^{-32591}$. Los exponentes de estos números exceden por mucho las limitaciones de la máquina (como se acaba de describir). Estos números podrían cursar sobreflujo y subflujo, respectivamente, y el error relativo al remplazar x por el número de máquina más cercano será muy grande. Estos números están *fueras de rango* de una computadora de longitud de palabra de 32 bits.

Consideraremos a continuación un número positivo real x en la forma normalizada de punto flotante

$$x = q \times 2^m \quad \left(\frac{1}{2} \leq q < 1, -126 \leq m \leq 127 \right)$$

El proceso de remplazar x por su número de máquina más cercano se llama **redondeo correcto** y el **error implicado** se denomina **error de redondeo**. Queremos saber cuán grande puede ser. Suponemos que q se expresa en notación binaria normalizada, así

$$x = (0.1b_2b_3b_4 \dots b_{24}b_{25}b_{26} \dots)_2 \times 2^m$$

Se puede obtener un número de máquina cercano al *redondear hacia abajo* o simplemente desechar los bits excedentes b_{25}, b_{26}, \dots , ya que sólo 23 bits se han asignado a la mantisa almacenada. Este número de máquina es

$$x_- = (0.1b_2b_3b_4 \dots b_{24})_2 \times 2^m$$

Se encuentra a la izquierda de x en el eje de los números reales. Otro número de máquina, x_+ , está exactamente a la derecha de x en el eje real y se obtiene al *redondear hacia arriba*. Se encuentra al sumar una unidad a b_{24} en la expresión para x_- . Así,

$$x_+ = [(0.1b_2b_3b_4 \dots b_{24})_2 + 2^{-24}] \times 2^m$$

El número más cercano de estos números de máquina es el que se elige para representar x .

Las dos situaciones se ejemplifican con diagramas simples en la figura 2.4. Si x está más cerca de x_- que de x_+ , entonces

$$|x - x_-| \leq \frac{1}{2}|x_+ - x_-| = 2^{-25+m}$$

En este caso, el error relativo está acotado como se indica a continuación:

$$\left| \frac{x - x_-}{x} \right| \leq \frac{2^{-25+m}}{(0.1b_2b_3b_4 \dots)_2 \times 2^m} \leq \frac{2^{-25}}{\frac{1}{2}} = 2^{-24} = u$$

donde $u = 2^{-24}$ es el **error unitario de redondeo** para una computadora binaria de 32 bits con aritmética estándar de punto flotante. Recuerde que el épsilon de máquina es $\epsilon = 2^{-23}$, así $u = \frac{1}{2}\epsilon$. Además, $u = 2^{-k}$, donde k es el número de dígitos binarios usados en la mantisa, incluido el bit escondido ($k = 24$ con precisión simple y $k = 53$ con doble precisión). Por otra parte, si x está más cerca de x_+ que de x_- , entonces

$$|x - x_+| \leq \frac{1}{2}|x_+ - x_-|$$

y el mismo análisis muestra que el error relativo no es más grande que $2^{-24} = u$. Por lo que en el caso de redondeo al número de máquina más cercano el error relativo está acotado por u .

FIGURA 2.4
Una posible
relación entre
 x_- , x_+ y x .



Observamos que cuando se descartan todos los dígitos excedentes, el proceso se llama truncamiento. Si una computadora de longitud de palabra de 32 bits se ha diseñado para truncar números, el límite para el error relativo sería del doble que el anterior, o sea $2u = 2^{-23} = \epsilon$.

Notación $\text{fl}(x)$ y análisis de error hacia atrás

Ahora permítanos regresar a los errores que se producen en el curso de operaciones aritméticas elementales. Para mostrar los principios, suponga que estamos trabajando con una máquina con cinco cifras decimales y que deseamos sumar números. Dos números típicos de máquina en forma de punto flotante normalizada son

$$x = 0.37218 \times 10^4 \quad y = 0.71422 \times 10^{-1}$$

Muchas computadoras realizan operaciones aritméticas en un área de trabajo de doble longitud, por lo que permítanos suponer que nuestra computadora tendrá un acumulador de diez lugares. Primero, el exponente del número más pequeño se ajusta para que los dos exponentes sean iguales. Despues se suman los números en el acumulador y el redondeo resultante se coloca en una palabra de computadora:

$$\begin{array}{r} x = 0.37218\ 00000 \times 10^4 \\ y = 0.00000\ 71422 \times 10^4 \\ \hline x + y = 0.37218\ 71422 \times 10^4 \end{array}$$

El número de máquina más cercano es $z = 0.37219 \times 10^4$ y el error relativo implicado en esta suma de la máquina es

$$\frac{|x + y - z|}{|x + y|} = \frac{0.00000\ 28578 \times 10^4}{0.37218\ 71422 \times 10^4} \approx 0.77 \times 10^{-5}$$

Este error relativo se consideraría aceptable en una máquina de baja precisión.

Para facilitar el análisis de estos errores es conveniente introducir la notación $\text{fl}(x)$ para denotar el **número de máquina de punto flotante** que corresponde al número real x . Por supuesto, la función fl depende de la computadora implicada. La supuesta máquina de dígitos de cinco decimales que se usó antes daría

$$\text{fl}(0.37218\ 71422 \times 10^4) = 0.37219 \times 10^4$$

Para una computadora de longitud de palabra de 32 bits, previamente establecimos que si x es cualquier número real dentro del rango de la computadora, entonces

$$\frac{|x - \text{fl}(x)|}{|x|} \leq u \quad (u = 2^{-24}) \quad (1)$$

De aquí en adelante, suponemos que se ha usado el redondeo correcto. Esta desigualdad también se puede expresar en la forma más útil

$$\text{fl}(x) = x(1 + \delta) \quad (|\delta| \leq 2^{-24})$$

Para ver que estas dos desigualdades son equivalentes, simplemente haga $\delta = [\text{fl}(x) - x]/x$. Entonces, por la desigualdad (1), tenemos $|\delta| \leq 2^{-24}$ y despejando $\text{fl}(x)$ se obtiene $\text{fl}(x) = x(1 + \delta)$.

Considerando los detalles en la suma $1 + \varepsilon$, vemos que si $\varepsilon \geq 2^{-23}$, entonces $\text{fl}(1 + \varepsilon) > 1$, mientras que si $\varepsilon < 2^{-23}$, entonces $\text{fl}(1 + \varepsilon) = 1$. Por lo tanto, si la **épsilon de máquina** es el número de máquina positivo más pequeño ε tal que

$$\text{fl}(1 + \varepsilon) > 1$$

entonces $\varepsilon = 2^{-23}$. A veces es necesario proporcionar la épsilon de máquina a un programa. Puesto que es una constante que depende de la máquina, se puede encontrar ya sea llamando una rutina del sistema o escribiendo un simple programa que encuentre el número positivo más pequeño $x = 2^m$ tal que $1 + x > 1$ en la máquina.

Ahora sea que el símbolo \odot denote cualquiera de las operaciones aritméticas $+, -, \times$ o \div . Suponga una computadora de longitud de palabra de 32 bits que se ha diseñado para que siempre que *dos números de máquina* x y y se combinan aritméticamente produzca $\text{fl}(x \odot y)$ en lugar de $x \odot y$. Podemos imaginar que $x \odot y$ se forma primero *correctamente*, después se normaliza y por último se redondea para convertirse en un número de máquina. Bajo esta suposición, el error relativo no excede 2^{-24} por el análisis anterior:

$$\text{fl}(x \odot y) = (x \odot y)(1 + \delta) \quad (| \delta | \leq 2^{-24})$$

Casos especiales de esto son, por supuesto,

$$\begin{aligned} \text{fl}(x \pm y) &= (x \pm y)(1 + \delta) \\ \text{fl}(xy) &= xy(1 + \delta) \\ \text{fl}\left(\frac{x}{y}\right) &= \left(\frac{x}{y}\right)(1 + \delta) \end{aligned}$$

En estas ecuaciones, δ es la variable que satisface $-2^{-24} \leq \delta \leq 2^{-24}$. Las suposiciones que hemos hecho acerca de un modelo de computadora de longitud de palabra de 32 bits no son muy ciertas para una computadora real. Por ejemplo, es posible que x y y sean números de máquina y que $x \odot y$ hagan un sobreflujo o un subflujo. No obstante, las suposiciones deben ser objetivas para la mayoría de las computadoras.

Las ecuaciones anteriores se pueden escribir de muchas formas, algunas de las cuales sugieren interpretaciones alternativas de redondeo. Por ejemplo,

$$\text{fl}(x + y) = x(1 + \delta) + y(1 + \delta)$$

Esta nos dice que el resultado de sumar números de máquina x y y no es en general $x + y$ sino la suma verdadera de $x(1 + \delta)$ y $y(1 + \delta)$. Podemos pensar que $x(1 + \delta)$ es el resultado de perturbar ligeramente a x . Así, la versión de máquina de $x + y$, que es $\text{fl}(x + y)$, es la suma *exacta* de una x ligeramente perturbada y de una y ligeramente perturbada. El lector puede proporcionar interpretaciones similares en los ejemplos dados en los problemas.

Esta interpretación es un ejemplo de **análisis de error hacia atrás**. Éste trata de determinar qué perturbación de los datos originales causaría que los *resultados de la computadora* fueran los resultados exactos de un problema perturbado. Al contrario, un **análisis de error directo** trata de determinar cómo difieren las respuestas calculadas de las respuestas exactas apartir de los mismos datos. En este aspecto de la computación científica, las computadoras han estimulado una nueva forma de observar los errores de cómputo.

EJEMPLO 4 Si x , y , y z son números de máquina en una computadora de longitud de palabra de 32 bits, ¿qué límite superior se le puede dar al error relativo de redondeo al calcular $z(x + y)$?

Solución En la computadora, se hará primero el cálculo de $x + y$. Esta operación aritmética da como resultado el número de máquina $\text{fl}(x + y)$, que difiere de $x + y$ debido al redondeo. Por los principios que ya establecimos, hay una δ_1 tal que

$$\text{fl}(x + y) = (x + y)(1 + \delta_1) \quad (|\delta_1| \leq 2^{-24})$$

Ahora z es ya un número de máquina. Cuando se multiplica el número de máquina $\text{fl}(x + y)$, el resultado es el número de máquina $\text{fl}[z \text{ fl}(x + y)]$. Éste, también, difiere de su contraparte exacta y se tiene, para alguna δ_2 ,

$$\text{fl}[z \text{ fl}(x + y)] = z \text{ fl}(x + y)(1 + \delta_2) \quad (|\delta_2| \leq 2^{-24})$$

Poniendo nuestras dos ecuaciones juntas, tenemos

$$\begin{aligned} \text{fl}[z \text{ fl}(x + y)] &= z(x + y)(1 + \delta_1)(1 + \delta_2) \\ &= z(x + y)(1 + \delta_1 + \delta_2 + \delta_1\delta_2) \\ &\approx z(x + y)(1 + \delta_1 + \delta_2) \\ &= z(x + y)(1 + \delta) \quad (|\delta| \leq 2^{-23}) \end{aligned}$$

En este cálculo, $|\delta_1\delta_2| \leq 2^{-48}$, por lo que lo ignoramos. También, hacemos $\delta = \delta_1 + \delta_2$ y entonces se concluye que $|\delta| = |\delta_1 + \delta_2| \leq |\delta_1| + |\delta_2| \leq 2^{-24} + 2^{-24} = 2^{-23}$. ■

EJEMPLO 5 Juzgue el siguiente intento para estimar el error relativo de redondeo al calcular la suma de dos números reales, x y y . En una computadora de longitud de palabra de 32 bits, el cálculo da como resultado

$$\begin{aligned} z &= \text{fl}[\text{fl}(x) + \text{fl}(y)] \\ &= [x(1 + \delta) + y(1 + \delta)](1 + \delta) \\ &= (x + y)(1 + \delta)^2 \\ &\approx (x + y)(1 + 2\delta) \end{aligned}$$

Por tanto, el error relativo está limitado de esta manera:

$$\left| \frac{(x + y) - z}{(x + y)} \right| = \left| \frac{2\delta(x + y)}{(x + y)} \right| = |2\delta| \leq 2^{-23}$$

¿Por qué este cálculo *no* es correcto?

Solución Las cantidades δ que ocurren en dichos cálculos no son, en general, iguales entre sí. El cálculo correcto es

$$\begin{aligned} z &= \text{fl}[\text{fl}(x) + \text{fl}(y)] \\ &= [x(1 + \delta_1) + y(1 + \delta_2)](1 + \delta_3) \\ &= [(x + y) + \delta_1x + \delta_2y](1 + \delta_3) \\ &= (x + y) + \delta_1x + \delta_2y + \delta_3x + \delta_3y + \delta_1\delta_3x + \delta_2\delta_3y \\ &\approx (x + y) + x(\delta_1 + \delta_3) + y(\delta_2 + \delta_3) \end{aligned}$$

Por tanto, el error relativo de redondeo es

$$\begin{aligned} \left| \frac{(x + y) - z}{(x + y)} \right| &= \left| \frac{x(\delta_1 + \delta_3) + y(\delta_2 + \delta_3)}{(x + y)} \right| \\ &= \left| \frac{(x + y)\delta_3 + x\delta_1 + y\delta_2}{(x + y)} \right| \\ &= \left| \delta_3 + \frac{x\delta_1 + y\delta_2}{(x + y)} \right| \end{aligned}$$

Éste no se puede acotar, ya que el segundo término tiene un denominador que puede ser cero o casi cero. Observe que si x y y son números de máquina, entonces δ_1 y δ_2 son cero y se obtiene un límite útil, a saber, δ_3 . Pero no necesitamos hacer este cálculo, ya que se ha supuesto que cuando se combinan números de máquina con cualquiera de las cuatro operaciones aritméticas el error relativo de redondeo no excederá a 2^{-24} en magnitud (en una computadora de longitud de palabra de 32 bits). ■

Notas históricas

En la Guerra del Golfo en 1991, una falla del sistema de misiles de defensa Patriot fue resultado de un error de conversión de software. El sistema del reloj midió el tiempo en décimas de segundo, que se almacenó como un número de punto flotante de 24 bits, dando como resultado errores de redondeo. Los datos de campo demostraron que el sistema fallaría en rastrear e interceptar un misil entrante después de funcionar durante 20 horas seguidas y necesitaría reiniciarse. Después de que había funcionado durante 100 horas, el sistema falló dando como resultado la muerte de 28 soldados estadounidenses en unas trincheras en Dhahran, Arabia Saudita, ya que éste erró al interceptar un misil iraquí Scud. Ya que el número 0.1 tiene una expansión binaria infinita, el valor en el registro de 24 bits tuvo un error de $(1.1001100\dots)_2 \times 2^{-24} \approx 0.95 \times 10^{-7}$. El error resultante del tiempo fue aproximadamente de treinta y cuatro centésimas de segundo después de funcionar durante 100 horas.

En 1996, el cohete Ariane 5 lanzado por la agencia espacial europea estalló 40 segundos después de que despegó en Kourou, Guayana Francesa. Una investigación determinó que la velocidad horizontal requería de la conversión de un número de punto flotante de 64 bits a un entero con signo de 16 bits. Esto falló, ya que el número era más grande que 32767, que era el entero más grande de este tipo que podría almacenarse en la memoria. El cohete y su carga se valuaron en 500 millones de dólares.

Más detalles acerca de estos desastres se pueden encontrar al investigar en la web. Hay otras historias interesantes de calamidades que se podrían haber evitado con una más cuidadosa programación en computadora, especialmente cuando se usa aritmética de punto flotante.

Resumen

(1) Un número de punto flotante de precisión simple en una computadora de longitud de palabra de 32 bits con representación estándar de punto flotante se almacena en una sola palabra con el patrón de bits

$$b_1 b_2 b_3 \cdots b_9 b_{10} b_{11} \cdots b_{32}$$

que se interpreta como el número real

$$(-1)^{b_1} \times 2^{(b_2 b_3 \dots b_9)_2} \times 2^{-127} \times (1.b_{10} b_{11} \dots b_{32})_2$$

(2) Un número de punto flotante de doble precisión en una computadora de longitud de palabra de 32 bits con representación estándar de punto flotante se almacena en dos palabras con el patrón de bits

$$b_1 b_2 b_3 \cdots b_9 b_{10} b_{11} b_{12} b_{13} \cdots b_{32} b_{33} b_{34} b_{35} \cdots \cdots b_{64}$$

que se interpreta como el número real

$$(-1)^{b_1} \times 2^{(b_2 b_3 \dots b_{12})_2} \times 2^{-1023} \times (1.b_{13} b_{14} \dots b_{64})_2$$

(3) La relación entre un número real x y el **número de máquina de punto flotante** $\text{fl}(x)$ se puede escribir como

$$\text{fl}(x) = x(1 + \delta) \quad (\lvert \delta \rvert \leq 2^{-24})$$

Si \odot denota cualquiera de las operaciones aritméticas, entonces escribimos

$$\text{fl}(x \odot y) = (x \odot y)(1 + \delta)$$

En estas ecuaciones, δ depende de x y y .

Problemas 2.1

1. Determine la representación de máquina con precisión simple en una computadora de longitud de palabra de 32 bits para los siguientes números decimales.

a. 2^{-30} **b.** 64.015625 **c.** -8×2^{-24}

2. Determine la representación de máquina con precisión simple y con doble precisión en una computadora de longitud de palabra de 32 bits de los siguientes números decimales:

a. $0.5, -0.5$ **b.** $0.125, -0.125$ **c.** $0.0625, -0.0625$ **d.** $0.03125, -0.03125$

- 3.** ¿Cuáles de éstos son números de máquina?

a. 10^{403} **b.** $1 + 2^{-32}$ **c.** $\frac{1}{5}$ **d.** $\frac{1}{10}$ **e.** $\frac{1}{256}$

4. Determine la representación de máquina con precisión simple y con doble precisión de los siguientes números decimales:

a. 1.0, -1.0 **b.** +0.0, -0.0 **c.** -9876.54321 **d.** 0.234375
e. 492.78125 **f.** 64.37109375 **g.** -285.75 **h.** 10^{-2}

5. Identifique los números de punto flotante correspondientes a las siguientes cadenas de bits:

- a. 0 00000000 000000000000000000000000000000
 - b. 1 00000000 000000000000000000000000000000
 - c. 0 11111111 000000000000000000000000000000
 - ^ad. 1 11111111 000000000000000000000000000000
 - e. 0 00000001 000000000000000000000000000000
 - f. 0 10000001 011000000000000000000000000000
 - g. 0 01111111 000000000000000000000000000000
 - h. 0 01111011 10011001100110011001100

6. ¿Cuáles son las representaciones de máquina de la cadena de bits para los siguientes números subnormales?

a. $2^{-127} + 2^{-128}$ **b.** $2^{-127} + 2^{-150}$ **c.** $2^{-127} + 2^{-130}$ **d.** $\sum_{k=127}^{150} 2^{-k}$

7. Determine el número de decimales que tienen las siguientes representaciones de máquina:
- [3F27E520]₁₆
 - [3BCDCA00]₁₆
 - [BF4F9680]₁₆
 - [CB187ABC]₁₆
8. Determine el número de decimales que tienen las siguientes representaciones de máquina:
- ^a[CA3F2900]₁₆
 - [C705A700]₁₆
 - [494F96A0]₁₆
 - ^ad.[4B187ABC]₁₆
 - [45223000]₁₆
 - [45607000]₁₆
 - ^ag.[C553E000]₁₆
 - [437F0000]₁₆
9. ¿Son éstas representaciones de máquina? ¿Por qué sí o por qué no?
- [4BAB2BEB]₁₆
 - [1A1AIA1A]₁₆
 - [FADEDEAD]₁₆
 - [CABE6G94]₁₆
10. La palabra computadora asociada con la variable Δ se presenta como [7F7FFFFF]₁₆, que es el número representable de punto flotante precisión simple más grande. ¿Cuál es el valor decimal de Δ ? La variable ϵ se presenta como [00800000]₁₆, que es el número positivo más pequeño. ¿Cuál es el valor decimal de ϵ ?
11. Liste el conjunto de números en el sistema de números de punto flotante que tenga representaciones binarias de la forma $\pm(0.b_1b_2) \times 2^k$, donde
- $k \in \{-1, 0\}$
 - $k \in \{-1, 1\}$
 - ^ac. $k \in \{-1, 0, 1\}$
12. ¿Cuáles son los números de máquina inmediatamente a la derecha y a la izquierda de 2^m ? ¿A qué distancia están de 2^m ?
13. En general, cuando se suma una lista de números de punto flotante, ocurrirá menos error de redondeo si los números se suman en orden creciente en magnitud. Dé algunos ejemplos que muestren este principio.
14. (Continuación) El principio del problema anterior *no es universalmente* válido. Considere una máquina decimal con dos dígitos decimales asignados a la mantisa. Muestre que se pueden sumar los cuatro números 0.25, 0.0034, 0.00051 y 0.061 con menos error de redondeo si *no* se agregan en orden creciente.
- ^a15. En el caso de que la máquina haga un subflujo, ¿cuál es el error relativo implicado al remplazar un número x por cero?
16. Considere una computadora que opera en base β y tiene n dígitos en la mantisa de sus números de punto flotante. Muestre que el redondeo de un número real x al número de máquina más cercano \tilde{x} implica un error relativo a lo más de $\frac{1}{2}\beta^{1-n}$. *Sugerencia:* imite el argumento del libro.
- ^a17. Considere una máquina decimal en la que cinco dígitos decimales están asignados a la mantisa. Dé un ejemplo, evitando el sobreflujo o el subflujo, de un número real x cuyo número de máquina más cercano \tilde{x} implica el error relativo más grande posible.
- ^a18. En una máquina de cinco decimales que redondea correctamente números al número de máquina más cercano, ¿qué números reales x tendrán la propiedad $\text{fl}(1.0 + x) = 1.0$?
- ^a19. Considere una computadora que opera en base β . Suponga que trunca números en lugar de redondearlos correctamente. Si sus números de punto flotante tienen una mantisa de n dígitos, ¿cuán grande es el error relativo al almacenar un número real en formato de máquina?

- “20.** ¿Cuál es el error de redondeo cuando representamos $2^{-1} + 2^{-25}$ con un número de máquina?
Nota: esto se refiere al error absoluto, no al error relativo.
- “21.** (Continuación) ¿Cuál es el error relativo de redondeo cuando redondeamos $2^{-1} + 2^{-26}$ para obtener el número de máquina más cercano?
- 22.** Si x es un número real dentro del rango de una computadora de longitud de palabra de 32 bits que se redondea y almacena, ¿qué puede pasar cuando se calcula x^2 ? Explique la diferencia entre $\text{fl}[\text{fl}(x)\text{fl}(x)]$ y $\text{fl}(x^2)$.
- 23.** Una máquina binaria que tiene 30 bits en la parte fraccionaria de cada número de punto flotante se diseña para redondear un número hacia arriba o hacia abajo correctamente para obtener el número de punto flotante más cercano. ¿Qué límite superior simple se puede dar para el error relativo en este proceso de redondeo?
- 24.** Una máquina decimal que tiene 15 cifras decimales en sus números de punto flotante está diseñada para truncar números. Si x es un número real en el rango de esta máquina y \tilde{x} es su representación de máquina, ¿qué límite superior se puede dar para $|x - \tilde{x}|/|x|$?
- “25.** Si x y y son números reales dentro del rango de una computadora de longitud de palabra de 32 bits y si xy está también en ese rango, ¿qué error relativo puede haber en el cálculo de xy de la máquina? *Sugerencia:* la máquina produce $\text{fl}[\text{fl}(x)\text{fl}(y)]$.
- “26.** Sean x y y números reales positivos que no son números de máquina pero están dentro del rango del exponente de una computadora de longitud de palabra de 32 bits. ¿Cuál es el error relativo más grande posible en la representación de máquina de $x + y^2$? Incluya los errores causados al obtener los números en la máquina, así como los errores en la aritmética.
- 27.** Demuestre que si x y y son números reales positivos que tienen los primeros n dígitos iguales en sus representaciones decimales, entonces y se aproxima a x con un error relativo menor que 10^{1-n} . ¿Lo contrario es cierto?
- 28.** Muestre que un límite burdo del error relativo de redondeo cuando n números de máquina se multiplican en una computadora de longitud de palabra de 32 bits es $(n-1)2^{-24}$.
- 29.** Muestre que $\text{fl}(x+y) = y$ en una computadora de longitud de palabra de 32 bits si x y y son números de máquina positivos y $x < y \times 2^{-25}$.
- “30.** Si se suman 1000 números de máquina distintos de cero en una computadora de longitud de palabra de 32 bits, ¿qué límite superior se puede dar para el error relativo de redondeo en el resultado? ¿Cuántos dígitos decimales en la respuesta pueden ser confiables?
- 31.** Suponga que $x = \sum_{i=1}^n a_i 2^{-i}$, donde $a_i \in \{-1, 0, 1\}$ es un número positivo. Muestre que x también se puede escribir en la forma $\sum_{i=1}^n b_i 2^{-i}$, donde $b_i \in \{0, 1\}$.
- 32.** Si x y y son números de máquina en una computadora de longitud de palabra de 32 bits y si $\text{fl}(x/y) = x/[y(1 + \delta)]$, ¿qué límite superior se puede poner sobre $|\delta|$?
- 33.** ¿Cuán grande es el hoyo en cero en una computadora de longitud de palabra de 32 bits?
- 34.** ¿Cuántos números de máquina hay en una computadora de longitud 32 bits? (Considere sólo números de punto flotante normalizados.)

- 35.** ¿Cuántos números de punto flotante normalizados están disponibles en una máquina binaria si se han asignado n bits a la mantisa y m bits al exponente? Suponga que se usan dos bits adicionales para los signos, como en una computadora de longitud 32 bits.
- 36.** Muestre con un ejemplo que en una computadora aritmética $a + (b + c)$ puede diferir de $(a + b) + c$.
- 37.** Considere una máquina decimal en la que los números de punto flotante tienen 13 cifras decimales. Suponga que los números están redondeados correctamente hacia arriba o hacia abajo al número de máquina más cercano. Dé el mejor límite para el error de redondeo, suponiendo que no haya subflujos ni sobreflujo. Por supuesto, use error relativo. ¿Qué sucede si siempre se truncan los números?
- 38.** Considere una computadora que usa números de cinco dígitos decimales. Sea que $\text{fl}(x)$ denote el número de máquina de punto flotante más cercano a x . Muestre que si $x = 0.53214\ 87513$ y $y = 0.53213\ 04421$, entonces la operación $\text{fl}(x) - \text{fl}(y)$ implica un gran error relativo. Calcúlelo.
- 39.** Dos números x y y que no son números de máquina se leen en una computadora de longitud de palabra de 32 bits. La máquina calcula xy^2 . ¿Qué clase de error relativo se puede esperar? Suponga que no hay subflujos ni sobreflujo.
- 40.** Sean x , y y z tres números de máquina en una computadora de longitud de palabra de 32 bits. Mediante análisis del error relativo en el peor caso, determine cuánto error de redondeo se debe esperar al formar $(xy)z$.
- 41.** Sean x y y números de máquina en una computadora de longitud de palabra de 32 bits. ¿Qué error relativo de redondeo se debe esperar en el cálculo de $x + y$? Si x es alrededor de 30 y y es alrededor de 250, ¿qué error absoluto se debe esperar en el cálculo de $x + y$?
- 42.** Cada número de máquina en una computadora de longitud de palabra de 32 bits se puede interpretar como la representación de máquina correcta de todo un *intervalo* de números reales. Describa este intervalo para los números de máquina $q \times 2^m$.
- 43.** ¿Es todo número de máquina en una computadora de longitud de palabra de 32 bits el promedio de otros dos números de máquina? Si no, describa los que no son promedio.
- 44.** Sean x y y números de máquina en una computadora de longitud de palabra de 32 bits. Sean u y v números reales en el rango de una computadora de longitud de palabra de 32 bits pero no números de máquina. Determine un límite superior objetivo del error relativo de redondeo cuando u y v se leen en la computadora y después úselo para calcular $(x + y)/(uv)$. Como siempre, ignore productos de dos o más números que tengan magnitudes tan pequeñas como 2^{-24} . Suponga que no ocurre sobreflujo o subflujos en este cálculo.
- 45.** Interprete lo siguiente:
- a.** $\text{fl}(x) = x(1 - \delta)$
 - b.** $\text{fl}(xy) = [x(1 + \delta)]y$
 - c.** $\text{fl}(xy) = x[y(1 + \delta)]$
 - d.** $\text{fl}(xy) = (x\sqrt{1 + \delta})(y\sqrt{1 + \delta})$
 - e.** $\text{fl}\left(\frac{x}{y}\right) = \frac{x(1 + \delta)}{y}$
 - f.** $\text{fl}\left(\frac{x}{y}\right) = \frac{x\sqrt{1 + \delta}}{y/\sqrt{1 + \delta}}$
 - g.** $\text{fl}\left(\frac{x}{y}\right) \approx \frac{x}{y(1 - \delta)}$
- 46.** Sean x y y números reales que no son números de máquina para una computadora de longitud de palabra de 32 bits y que se han redondeado para obtenerlos en la máquina. Suponga que

no hay sobreflujo o subflujo al obtener los valores (redondeados) en la máquina. (Así, los números están dentro del *rango* de una computadora de longitud de palabra de 32 bits, aunque no son números de máquina.) Encuentre un límite superior burdo del error relativo al calcular x^2 y 3^x . *Sugerencia:* decimos *límite superior burdo*, ya que se puede usar $(1 + \delta_1)(1 + \delta_2) \approx 1 + \delta_1 + \delta_2$ y aproximaciones similares. Asegúrese de incluir los errores implicados al obtener los números en la máquina, así como los errores que surgen de las operaciones aritméticas.

- 47. (Proyecto de investigación para el estudiante)** Escriba un artículo de investigación acerca del sistema estándar de números de puntos flotantes dando detalles adicionales sobre
- a. tipos de redondeo
 - b. números subnormales de punto flotante
 - c. precisión extendida
 - d. manejo de situaciones excepcionales

Problemas de cómputo 2.1

1. Imprima varios números, tanto enteros como reales, en formato octal y trate de explicar la representación de máquina usada en su computadora. Por ejemplo, examine $(0.1)_1 0$ y compare con los resultados dados al inicio de este capítulo.
2. Use su computadora para construir una tabla de tres funciones f , g y h definidas como sigue. Para cada entero n en el rango de 1 a 50, sea $f(n) = 1/n$. Entonces $g(n)$ se calcula sumando $f(n)$ a sí misma $n - 1$ veces. Por último, hacemos $h(n) = nf(n)$. Queremos ver los efectos de error de redondeo en estos cálculos. Use la función $\text{real}(n)$ para convertir una variable entero n en su forma real (punto flotante). Imprima la tabla con toda la precisión que pueda su computadora (en modo de precisión simple).
3. Prediga y después muestre qué valor imprimirá su computadora para $\sqrt{2}$ calculada con precisión simple. Repita para precisión doble o extendida. Explique.
4. Escriba un programa para determinar la épsilon de máquina ε dentro de un factor de 2 para precisión simple, doble y extendida.
5. Sea \mathcal{A} el conjunto de enteros positivos cuya representación decimal no tenga el dígito 0. Se sabe que la suma de los recíprocos de los elementos en \mathcal{A} es 23.10345. ¿Puede verificar esto numéricamente?
6. Escriba un código de computadora

integer function $nDigit(n, x)$

que regrese el enésimo dígito diferente de cero en la expresión decimal para el número real x .

7. Se sabe que la **serie armónica** $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$ diverge a $+\infty$. La enésima suma parcial tiende a $+\infty$ con la misma razón que $\ln(n)$. La **constante de Euler** se define como

$$\gamma = \lim_{n \rightarrow \infty} \left[\sum_{k=1}^n \frac{1}{k} - \ln(n) \right] \approx 0.57721$$

Si ejecutó un programa en su computadora durante una semana basado en el seudocódigo

```
real s, x
x  $\leftarrow$  1.0; s  $\leftarrow$  1.0
repeat
    x  $\leftarrow$  x + 1.0; s  $\leftarrow$  s + 1.0/x
end repeat
```

¿cuál es el valor más grande de *s* que se obtendría? Escriba y pruebe un ciclo de programa de 5000 pasos para calcular la constante de Euler. Imprima respuestas intermedias cada 100 pasos.

8. (Continuación) Pruebe que la constante de Euler, γ , también se puede representar por

$$\gamma = \lim_{m \rightarrow \infty} \left[\sum_{k=1}^m \frac{1}{k} - \ln\left(m + \frac{1}{2}\right) \right]$$

Escriba y pruebe un programa que use $m = 1, 2, 3, \dots, 5000$ para calcular γ con esta fórmula. La convergencia sería más rápida que en el cálculo anterior (véase el artículo de De Temple [1993]).

9. Determine la forma binaria de $\frac{1}{3}$. ¿Cuál es la representación de máquina correctamente redondeada en precisión simple en una computadora de longitud de palabra de 32 bits? Compruebe su respuesta en una máquina real con las instrucciones

x \leftarrow 1.0/3.0; **output** *x*

usando una longitud formal de 16 dígitos para el enunciado de salida.

10. Debido a su jalón gravitacional, la Tierra gana peso y volumen lentamente a partir del polvo espacial, meteoritos y cometas. Suponga que la Tierra es una esfera. Sea el radio $r_a = 7000$ kilómetros al inicio del año 1900 y sea r_b su radio a fines del año 2000. Suponga que $r_b = r_a + 0.000001$, un aumento de 1 milímetro. Usando una computadora, calcule cuánto volumen terrestre y área superficial ha aumentado durante el último siglo con los tres procedimientos siguientes (exactamente como se dan):

- a. $V_a = \frac{4}{3}\pi r_a^3$, $V_b = \frac{4}{3}\pi r_b^3$, $\delta_1 = V_b - V_a$ (diferencia en volumen esférico)
- b. $\delta_2 = \frac{4}{3}\pi(r_b - r_a)(r_b^2 + r_b r_a + r_a^2)$ (diferencia en volumen esférico)
- c. $h = r_b - r_a$, $\delta_3 = 4\pi r_a^2 h$ (diferencia en área superficial)

Primero use precisión simple y después doble precisión. Compare y analice sus resultados. (Este problema lo sugirió un lector anónimo.)

11. (**Proyecto de investigación para el estudiante**) Explore los desarrollos recientes en aritmética de punto flotante. En especial, aprenda acerca de precisión extendida tanto para números reales como para números complejos.

12. ¿Cuál es el entero más grande que su computadora puede manejar?

2.2 Pérdida de significancia

En esta sección mostramos cómo la pérdida de significancia en la resta con frecuencia se puede reducir o eliminar usando diferentes técnicas, como el uso de racionalización, series de Taylor, identidades trigonométricas, propiedades logarítmicas, doble precisión y reducción de rango. Estas son algunas de las técnicas que se pueden utilizar cuando se quiere proteger un cálculo contra la pérdida de precisión. Por supuesto, no podemos saber siempre que ha ocurrido una pérdida de significancia en un cálculo largo, pero debemos estar alertas a la posibilidad y, de ser factible, tomar precauciones para evitarla.

Dígitos significativos

Primero trataremos el difícil concepto de los **dígitos significativos** en un número. Suponga que x es un número real expresado en notación científica normalizada en el sistema decimal

$$x = \pm r \times 10^n \quad \left(\frac{1}{10} \leq r < 1 \right)$$

Por ejemplo, x podría ser

$$x = 0.3721498 \times 10^{-5}$$

Los dígitos 3, 7, 2, 1, 4, 9 y 8 usados para expresar r no tienen todos la misma significancia, ya que representan potencias de 10 diferentes. Así, decimos que 3 es el dígito más significativo y la significancia de los dígitos disminuye de izquierda a derecha. En este ejemplo, 8 es el dígito *menos* significativo.

Si x es un número real *matemáticamente exacto*, entonces su forma decimal aproximada se puede dar con tantos dígitos significativos como queramos. Por tanto, podemos escribir

$$\frac{\pi}{10} \approx 0.31415\ 92653\ 58979$$

y todos los dígitos dados son correctos. Sin embargo, si x es una *cantidad medida*, la situación es muy diferente. Toda cantidad medida implica un error cuya magnitud depende de la naturaleza del dispositivo de medición. De este modo, si se usa un metro no es razonable medir ninguna longitud con una precisión mejor que 1 milímetro. Por tanto, el resultado de medir, digamos, una ventana de vidrio con un metro no se debe reportar como 2.73594 metros. Sería un error. Sólo se deben reportar dígitos que se cree son correctos o que tienen error, a lo más, de unas cuantas unidades. Es una convención científica de que el dígito menos significativo en una cantidad medida debe tener error a lo más en cinco unidades; es decir, el resultado está redondeado correctamente.

Observaciones similares pertenecen a cantidades calculadas de cantidades medidas. Por ejemplo, si el lado de un cuadrado se reporta de $s = 0.736$ metros, entonces se puede suponer que el error no excede de unas cuantas unidades en la tercera cifra decimal. La diagonal de ese cuadrado es entonces

$$s\sqrt{2} \approx 0.10408\ 61182 \times 10^1$$

pero se debe reportar como 0.1041×10^1 o (más conservadoramente) 0.104×10^1 . La precisión infinita disponible en $\sqrt{2}$,

$$\sqrt{2} = 1.41421\ 35623\ 73095 \dots$$

no dan más precisión a $\sqrt{2}$ de la que ya está presente en s .

Pérdida de significancia causada por la computación

Quizá sea sorprendente que una pérdida de significancia pueda ocurrir dentro de la computadora. Es esencial para entender este proceso no confiar ciegamente en el resultado numérico de una computadora. Una de las causas más comunes para el deterioro en la precisión es la resta de una cantidad de otra cantidad casi igual. Este efecto es potencialmente muy serio y puede resultar catastrófico. Entre más cerca estén estos dos números, más pronunciado es el efecto.

Para ilustrar este fenómeno permítanos considerar el enunciado de asignación

$$y \leftarrow x - \sin(x)$$

y suponga que en algún punto en un programa de computadora este enunciado se ejecuta con un valor de $\frac{1}{15}$. Imagine además que nuestra computadora funciona con números de punto flotante y tenemos diez dígitos decimales. Entonces

$$\begin{aligned}x &\leftarrow 0.66666\ 66667 \times 10^{-1} \\ \sin(x) &\leftarrow 0.66617\ 29492 \times 10^{-1} \\ x - \sin(x) &\leftarrow 0.00049\ 37175 \times 10^{-1} \\ x - \sin(x) &\leftarrow 0.49371\ 75000 \times 10^{-4}\end{aligned}$$

En el último paso, el resultado se ha corrido a la forma de punto flotante normalizada. La computadora ha puesto tres ceros en los tres lugares decimales *menos* significativos. Nos referimos a estos como **ceros espurios**; *no* son dígitos significativos. De hecho, el valor correcto de los diez dígitos decimales es

$$\frac{1}{15} - \sin \frac{1}{15} \approx 0.49371\ 74327 \times 10^{-4}$$

Otra forma de interpretar esto es observar que el último dígito en $x - \sin(x)$ se deriva del décimo dígito en x y en $\sin(x)$. Cuando el undécimo dígito ya sea en x o $\sin(x)$ es 5, 6, 1, 8 o 9, los valores numéricos se redondean hacia arriba a diez dígitos, por lo que su décimo dígito pueden estar alterados en más de una unidad. Puesto que estos diez dígitos pueden tener error, el último dígito en $x - \sin(x)$ también puede tener error ¡que es éste!

EJEMPLO 1 Si $x = 0.37214\ 48693$ y $y = 0.37202\ 14371$, ¿cuál es el error relativo en el cálculo de $x - y$ en una computadora que tiene una exactitud de cinco dígitos decimales?

Solución Los números primero serían redondeados a $\tilde{x} = 0.37214$ y $\tilde{y} = 0.37202$. Entonces tenemos $\tilde{x} - \tilde{y} = 0.00012$, mientras que la respuesta correcta es $x - y = 0.00012\ 34322$. El error relativo implicado es

$$\frac{|(x - y) - (\tilde{x} - \tilde{y})|}{|x - y|} = \frac{0.00000\ 34322}{0.00012\ 34322} \approx 3 \times 10^{-2}$$

Esta magnitud de error relativo se debe juzgar muy grande cuando se compara con los errores relativos de \tilde{x} y de \tilde{y} . (No pueden exceder a $\frac{1}{2} \times 10^{-4}$ estimando burdamente y en este ejemplo, de hecho, son aproximadamente 1.3×10^{-5}). ■

Se debe subrayar que en este análisis no interviene la operación

$$\text{fl}(x - y) \leftarrow x - y$$

sino más bien la operación

$$\text{fl}[\text{fl}(x) - \text{fl}(y)] \leftarrow x - y$$

El error de redondeo en el caso anterior está dado por la ecuación

$$\text{fl}(x - y) = (x - y)(1 + \delta)$$

donde $|\delta| \leq 2^{-24}$ en una computadora de longitud de palabra de 32 bits y en una computadora de decimales de cinco dígitos en el ejemplo anterior $|\delta| \leq \frac{1}{2} \times 10^{-4}$.

En el ejemplo 1, observamos que la diferencia calculada de 0.00012 tiene sólo dos números significativos de exactitud, mientras que en general, se espera que los números y cálculos en esta computadora tengan cinco dígitos significativos de exactitud.

El remedio contra esta dificultad es primero prever que esto puede suceder y entonces programar nuevamente. La técnica más simple puede realizar parte de un cálculo con aritmética de precisión doble o extendida (que significa aproximadamente del doble de dígitos significativos), pero con frecuencia se requiere un cambio ligero en las fórmulas. Se darán algunos ejemplos de esto y el lector encontrará algunos más entre los problemas.

Considere el ejemplo 1, pero imagine que los cálculos para obtener x , y y $x - y$ se están haciendo con doble precisión. Suponga que en lo sucesivo se utiliza aritmética de precisión simple. En la computadora, todos los diez dígitos de x , y y $x - y$ se retendrán, pero al final $x - y$ se redondeará a su forma de cinco dígitos, que es 0.12343×10^{-3} . Esta respuesta tiene cinco dígitos significativos de exactitud, como nos gustaría. Por supuesto, el programador o analista debe conocer antes dónde será necesaria en el cálculo la aritmética de doble precisión. Programar cada cosa con doble precisión es mucho derroche si no es necesario. Este método tiene otro inconveniente: se pueden eliminar tantos dígitos significativos que aún la doble precisión podría no ayudar.

Teorema de pérdida de precisión

Antes de considerar otras técnicas para evitar este problema nos preguntamos lo siguiente: *exactamente ¿cuántos dígitos binarios significativos se pierden en la resta $x - y$ cuando x está cerca de y ?* La cercanía de x y y es convenientemente medida con $|1 - (y/x)|$. Este es el resultado:

■ TEOREMA 1

Teorema de pérdida de precisión

Sean x y y números de máquina de punto flotante normalizados, donde $x > y > 0$. Si se tiene que $2^{-p} \leq 1 - (y/x) \leq 2^{-q}$ para algunos enteros positivos p y q , entonces a lo más p y a lo menos q bits binarios significativos se pierden en la resta $x - y$.

Demostración

Demostramos la segunda parte del teorema y dejamos la primera como un ejercicio. Para esto, sea $x = r \times 2^n$ y $y = s \times 2^m$, donde $\frac{1}{2} \leq r, s < 1$. (Ésta es la forma binaria normalizada de punto flotante.)

Puesto que $y < x$, la computadora puede tener un *corrimiento* y después realizar la resta. En cualquier caso, se debe primero expresar y con el mismo exponente que x . Por tanto, $y = (s2^{m-n}) \times 2^n$ y

$$x - y = (r - s2^{m-n}) \times 2^n$$

La mantisa de este número satisface las ecuaciones y la desigualdad

$$r - s2^{m-n} = r \left(1 - \frac{s2^m}{r2^n}\right) = r \left(1 - \frac{y}{x}\right) < 2^{-q}$$

Por tanto, para normalizar la representación de $x - y$ es necesario un corrimiento de al menos q bits a la izquierda. Despues al menos q ceros (espurios) se colocan en el extremo derecho de la mantisa. Esto significa que al menos q bits de precisión se han perdido. ■

EJEMPLO 2 En la resta $37.59362\ 1 - 37.58421\ 6$, ¿cuántos bits significativos se perderán?

Solución Sea x el primer número y y el segundo. Entonces

$$1 - \frac{y}{x} = 0.00025\ 01754$$

Esto se encuentra entre 2^{-12} y 2^{-11} . Estos dos números son $0.00024\ 4$ y $0.00048\ 8$. Por tanto, se pierden al menos 11 pero no más que 12 bits. ■

Ahora presentamos un ejemplo en forma decimal: sea $x = .6353$ y $y = .6311$. Están cerca, y $1 - y/x = .00661 < 10^{-2}$. En la resta, tenemos $x - y = .0042$. Hay dos números significativos en la respuesta, aunque había cuatro números significativos en x y y .

Cómo evitar la pérdida de significancia en la resta

Ahora consideraremos varias técnicas que se pueden utilizar para evitar la pérdida de significancia que puede ocurrir en la resta. Considere la función

$$f(x) = \sqrt{x^2 + 1} - 1 \tag{1}$$

cuyos valores pueden ser requeridos para x cerca de cero. Puesto que $\sqrt{x^2 + 1} \approx 1$ cuando $x \approx 0$, vemos que hay una pérdida potencial de significancia en la resta. Sin embargo, la función se puede reescribir en la forma

$$f(x) = (\sqrt{x^2 + 1} - 1) \left(\frac{\sqrt{x^2 + 1} + 1}{\sqrt{x^2 + 1} + 1} \right) = \frac{x^2}{\sqrt{x^2 + 1} + 1} \tag{2}$$

racionalizando el numerador, es decir, eliminando el radical en el numerador. Este procedimiento permite eliminar términos y, en consecuencia, eliminar la resta. Por ejemplo, si usamos aritmética de cinco dígitos decimales y si $x = 10^{-3}$, entonces $f(x)$ se calculará incorrectamente como cero con la primera fórmula pero como $\frac{1}{2} \times 10^{-6}$ con la segunda. Si usamos la primera fórmula junto con doble precisión, la dificultad disminuye pero *no* se evade por completo. Por ejemplo, con doble precisión, tenemos el mismo problema cuando $x = 10^{-6}$.

Como otro ejemplo, suponga que los valores de

$$f(x) = x - \sin x \quad (3)$$

se requieren cerca de $x = 0$. Un programador descuidado podría codificar esta función exactamente como se indicó en la ecuación (3), sin darse cuenta de que ocurrirá una seria pérdida de exactitud. Recuerde del cálculo que

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

para ver que $\sin x \approx x$ cuando $x \approx 0$. Un remedio para este problema es usar la serie de Taylor para $\sin x$:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Se sabe que esta serie representa $\sin x$ para todos los valores reales de x . Para x cerca de cero, converge muy rápidamente. Usando esta serie podemos escribir la función f como

$$f(x) = x - \left(x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} - \dots \right) = \frac{x^3}{3!} - \frac{x^5}{5!} + \frac{x^7}{7!} - \dots \quad (4)$$

Vemos en esta ecuación dónde surge la dificultad original; a saber, para pequeños valores de x , el término x en la serie seno es mucho más grande que $x^3/3!$ y, por ende, más importante. Pero cuando se forma $f(x)$, este término de x dominante desaparece y quedan sólo los términos inferiores. La serie que inicia con $x^3/3!$ es muy eficiente para calcular $f(x)$ cuando x es pequeña.

En este ejemplo, se necesita más análisis para determinar el rango en el que se debe usar la serie (4) y el rango en el que se puede usar la fórmula (3). Usando el teorema de pérdida de precisión, vemos que la pérdida de bits en la resta de la fórmula (3) se puede limitar a lo más en *un* bit al restringir x de modo que $\frac{1}{2} \leq 1 - \sin x/x$. (Aquí estamos considerando sólo el caso cuando $\sin x > 0$.) Con una calculadora es fácil ver que x debe ser al menos 1.9. Por esto, para $|x| < 1.9$, usamos unos cuantos de los primeros términos de la serie (4) y para $|x| \geq 1.9$ usamos $f(x) = x - \sin x$. Se puede comprobar que en el peor de los casos ($x = 1.9$), diez términos en la serie dan $f(x)$ con un error a lo más de 10^{-16} (que es bastante bueno para doble precisión en una computadora de longitud de palabra de 32 bits.)

Para construir un procedimiento de función para $f(x)$, observe que los términos en la serie se pueden obtener por inducción mediante el algoritmo

$$\begin{cases} t_1 = \frac{x^3}{6} \\ t_{n+1} = \frac{-t_n x^2}{(2n+2)(2n+3)} \end{cases} \quad (n \geq 1)$$

Entonces las sumas parciales se pueden obtener por inducción por medio de

$$\begin{cases} s_1 = t_1 \\ s_{n+1} = s_n + t_{n+1} \end{cases} \quad (n \geq 1)$$

de forma que

$$s_n = \sum_{k=1}^n t_k = \sum_{k=1}^n (-1)^{k+1} \left[\frac{x^{2k+1}}{(2k+1)!} \right]$$

Un seudocódigo adecuado para una función se presenta aquí:

```

real function  $f(x)$ 
integer  $i, n \leftarrow 10$ ; real  $s, t, x$ 
if  $|x| \geq 1.9$  then
     $s \leftarrow x - \sin x$ 
    else
         $t \leftarrow x^3 / 6$ 
         $s \leftarrow t$ 
        for  $i = 2$  to  $n$  do
             $t \leftarrow -tx^2 / [(2i + 2)(2i + 3)]$ 
             $s \leftarrow s + t$ 
        end for
    end if
     $f \leftarrow s$ 
end function  $f$ 

```

EJEMPLO 3 ¿Cómo se pueden calcular valores exactos de la función

$$f(x) = e^x - e^{-2x}$$

en la vecindad de $x = 0$?

Solución Puesto que e^x y e^{-2x} son ambos iguales a 1 cuando $x = 0$, habrá una pérdida de significancia debida a la resta cuando x es casi cero. Insertando la serie de Taylor adecuada obtenemos

$$\begin{aligned} f(x) &= \left(1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots\right) - \left(1 - 2x + \frac{4x^2}{2!} - \frac{8x^3}{3!} + \dots\right) \\ &= 3x - \frac{3}{2}x^2 + \frac{3}{2}x^3 - \dots \end{aligned}$$

Una alternativa es escribir

$$\begin{aligned} f(x) &= e^{-2x}(e^{3x} - 1) \\ &= e^{-2x} \left(3x + \frac{9}{2!}x^2 + \frac{27}{3!}x^3 + \dots\right) \end{aligned}$$

Usando el teorema de pérdida de precisión encontramos que a lo más se pierde un bit en la resta $e^x - e^{-2x}$ cuando $x > 0$ y

$$\frac{1}{2} \leq 1 - \frac{e^{-2x}}{e^x}$$

Esta desigualdad es válida cuando $x \geq \frac{1}{3} \ln 2 = 0.23105$. Razonando de forma similar cuando $x < 0$ se muestra que para $x \leq -0.23105$, a lo más se pierde un bit. Por tanto, la serie se debe usar para $|x| < 0.23105$.

■

EJEMPLO 4 Juzgue el enunciado de asignación

$$y \leftarrow \cos^2(x) - \sin^2(x)$$

Solución Cuando se calcula $\cos^2(x) - \sin^2(x)$, habrá una pérdida de significancia en $x = \pi/4$ (y otros puntos). Se debe usar la identidad trigonométrica simple.

$$\cos 2\theta = \cos^2 \theta - \sin^2 \theta$$

Así, el enunciado de asignación se debe remplazar por

$$y \leftarrow \cos(2x)$$

EJEMPLO 5 Juzgue el enunciado de asignación

$$y \leftarrow \ln(x) - 1$$

Solución Si se usa la expresión $\ln x - 1$ para x cercanas a e , habrá una eliminación de dígitos y una pérdida de exactitud. Se pueden usar hechos elementales de los logaritmos para vencer la dificultad. Así, tenemos $y = \ln x - 1 = \ln x - \ln e = \ln(x/e)$. Éste es un enunciado de asignación adecuado

$$y \leftarrow \ln\left(\frac{x}{e}\right)$$

Reducción de rango

Otra causa de pérdida de números significativos es la evaluación de varias funciones de biblioteca con argumentos muy largos. Este problema es más sutil que el que se analizó. Ilustramos con la función seno.

Una propiedad básica de la función $\sin x$ es su **periodicidad**:

$$\sin x = \sin(x + 2n\pi)$$

para todos los valores reales de x y para todos los valores enteros de n . Debido a esta relación se necesita conocer sólo los valores de $\sin x$ en algún intervalo fijo de longitud 2π para calcular $\sin x$ para cualquier x . Esta propiedad se usa en la evaluación con computadora de $\sin x$ y se llama **reducción de rango**.

Ahora suponga que queremos evaluar $\sin(12532.14)$. Restando múltiplos enteros de 2π , encontramos que es igual a $\sin(3.47)$ si retenemos sólo dos dígitos decimales de exactitud. De $\sin(12532.14) = \sin(12532.14 - 2k\pi)$, queremos $12532 = 2k\pi$ y $k = 3989/2\pi \approx 1994$. Por tanto, obtenemos $12532.14 - 2(1994)\pi = 3.49$ y $\sin(12532.14) \approx \sin(3.49)$. Por ello, aunque nuestro argumento original 12532.14 tenía siete números significativos, el argumento reducido tiene sólo tres. Los dígitos restantes se eliminan en la resta de 3988π . Puesto que 3.47 tiene sólo tres números significativos, nuestro valor calculado de $\sin(12532.14)$ tendrá no más de tres números significativos. Esta disminución de precisión es inevitable si no hay manera de aumentar la precisión del argumento original. Si el argumento original (12532.14 en este ejemplo) se puede obtener con más números significativos, estos números adicionales se encontrarán en el argumento *reducido* (3.47 en este ejemplo). En algunos casos, ayudará programar con precisión doble o extendida.

EJEMPLO 5 Para $\sin x$, ¿cuántos bits binarios significativos se pierden en la reducción de rango al intervalo $[0, 2\pi]$?

Solución A partir de un argumento $x > 2\pi$, determinaremos un entero n que satisface la desigualdad $0 < x - 2n\pi < 2$. Entonces, para evaluar las funciones trigonométricas elementales, usamos

$f(x) = f(x - 2n\pi)$. En la resta $x - 2n\pi$ habrá una pérdida de significancia. Por el teorema de pérdida de precisión, se pierden al menos q bits si

$$1 - \frac{2n\pi}{x} \leq 2^{-q}$$

Puesto que

$$1 - \frac{2n\pi}{x} = \frac{x - 2n\pi}{x} < \frac{2\pi}{x}$$

concluimos que al menos se pierden q bits si $2\pi/x \leq 2^{-q}$. Dicho de otra manera, al menos q bits se pierden si $2^q \leq x/2\pi$ ■

Resumen

- (1) Para evitar la pérdida de significancia en la resta se puede reformular la expresión racionalizando, usando desarrollos de series o identidades matemáticas.
- (2) Si x y y son números de máquina positivos de punto flotante normalizados con

$$2^{-p} \leq 1 - \frac{y}{x} \leq 2^{-q}$$

entonces a lo más p y al menos q bits binarios significativos se pierden al calcular $x - y$. Observe que aquí se puede omitir la hipótesis $x > y$.

Referencias adicionales

Para mayor estudio y lectura de material relacionado con este capítulo, véase el apéndice B, así como las referencias siguientes: Acton [1996], Bornemann, Laurie, Wagon y Waldvogel [2004], Goldberg [1991], Higham [2002], Hedges [1983], Kincaid y Cheney [2002], Overton [2001], Salamin [1976], Wilkinson [1963] y otros listados en la bibliografía.

Problemas 2.2

1. ¿Cómo se puede evaluar la función $f(x) = \sqrt{x+4} - 2$ exactamente cuando x es pequeña?
2. Calcule $f(10^{-2})$ para la función $f(x) = e^x - x - 1$. La respuesta debe tener cinco números significativos y se puede obtener fácilmente con lápiz y papel. Compare esto con la evaluación directa de $f(10^{-2})$ usando $e^{0.01} \approx 1.0101$.
3. ¿Cuál es la mejor manera de calcular los valores de la función $f(x) = e^x - e$ si se necesita una precisión de máquina total? *Nota:* hay cierta dificultad cuando $x = 1$.
4. ¿Qué dificultad causaría la asignación siguiente?

$$y \leftarrow 1 - \sin x$$

Si es posible evádala sin recurrir a una serie de Taylor.

- 5.** La función seno hiperbólico está definida por $\operatorname{senh} x = \frac{1}{2}(e^x - e^{-x})$. ¿Qué desventaja podría haber al usar esta fórmula para obtener valores de la función? ¿Cómo se pueden calcular los valores de $\operatorname{senh} x$ con precisión de máquina total cuando $|x| \leq \frac{1}{2}$?

- 6.** Determine los primeros dos términos diferentes de cero en el desarrollo alrededor de cero para la función

$$f(x) = \frac{\tan x - \operatorname{sen} x}{x - \sqrt{1 + x^2}}$$

Dé un valor aproximado para $f(0.0125)$.

- 7.** Encuentre un método para calcular

$$y \leftarrow \frac{1}{x}(\operatorname{senh} x - \tanh x)$$

que evite la pérdida de significancia cuando x es pequeña. Encuentre identidades adecuadas para resolver este problema sin usar series de Taylor.

- 8.** Encuentre una forma de calcular valores exactos para

$$f(x) = \frac{\sqrt{1 + x^2} - 1}{x^2} - \frac{x^2 \operatorname{sen} x}{x - \tan x}$$

Determine $\lim_{x \rightarrow 0} f(x)$.

- 9.** Para algunos valores de x , el enunciado de asignación $y \leftarrow 1 - \cos x$ implica una dificultad. ¿Cuál es ésta?, ¿qué valores de x están implicados y qué solución propone?

- 10.** Para algunos valores de x , la función $f(x) = \sqrt{x^2 + 1} - x$ no se puede calcular exactamente usando esta fórmula. Explique y encuentre una manera de evitar la dificultad.

- 11.** El seno hiperbólico inverso está dado por $f(x) = \ln(x + \sqrt{x^2 + 1})$. Muestre cómo evitar la pérdida de significancia al calcular $f(x)$ cuando x es negativa. *Sugerencia:* encuentre y aproveche la relación entre $f(x)$ y $f(-x)$.

- 12.** En la mayoría de computadoras, se entrega una rutina muy exacta para $\cos x$. Se propone una rutina base para $\operatorname{sen} x$ en la fórmula $x = \pm\sqrt{1 - \cos^2 x}$. Desde el punto de vista de la precisión (no de la eficiencia), ¿qué problemas prevé y cómo puede evitarlos si insistimos en usar la rutina para $\cos x$?

- 13.** Juzgue y codifique de nuevo el enunciado de asignación $z \leftarrow \sqrt{x^4 + 4} - 2$ suponiendo que se necesitará z ocasionalmente para una x cercana a cero.

- 14.** ¿Cómo se pueden calcular exactamente los valores de la función $f(x) = \sqrt{x+2} - \sqrt{x}$ cuando x es grande?

- 15.** Escriba una función que calcule valores exactos de $f(x) = \sqrt[4]{x+4} - \sqrt[4]{x}$ para x positiva.

- 16.** Encuentre una forma de calcular $f(x) = (\cos x - e^{-x})/\operatorname{sen} x$ correctamente. Determine $f(0.008)$ correctamente con diez lugares decimales (redondeado).

- 17.** Sin usar series, ¿cómo podría calcular exactamente la función

$$f(x) = \frac{\operatorname{sen} x}{x - \sqrt{x^2 - 1}}$$

para evitar pérdida de significancia?

- 18.** Escriba procedimiento de función que regrese valores exactos de la función tangente hiperbólica

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

para todos los valores de x . Note la dificultad cuando $|x| < \frac{1}{2}$.

- 19.** Encuentre una buena manera de calcular $\sin x + \cos x - 1$ para x cercana cero.

- “20.** Encuentre una buena manera de calcular $\arctan x - x$ para x cercana a cero.

- 21.** Encuentre un buen límite para $|\ln x - x|$ usando series de Taylor y suponiendo que $|x| < \frac{1}{10}$.

- “22.** ¿Cómo calcularía $(e^{2x} - 1) / (2x)$ para evitar pérdida de significancia cerca de cero?

- 23.** Para cualquier $x_0 > -1$, la sucesión definida recursivamente por

$$x_{n+1} = 2^{n+1} (\sqrt{1 + 2^{-n} x_n} - 1) \quad (n \geq 0)$$

converge a $\ln(x_0 + 1)$. Arregle esta fórmula en una forma que evite pérdida de significancia.

- 24.** Indique cómo pueden ser útiles las siguientes fórmulas para arreglar cálculos a fin de evitar la pérdida de dígitos significativos.

a. $\sin x - \sin y = 2 \sin \frac{1}{2}(x - y) \cos \frac{1}{2}(x + ny)$

b. $\log x - \log y = \log(x/y)$ **c.** $e^{x-y} = e^x / e^y$ **d.** $1 - \cos x = 2 \sin^2(x/2)$

e. $\arctan x - \arctan y = \arctan \left(\frac{x - y}{1 + xy} \right)$

- 25.** ¿Cuál es la mejor manera de calcular $\tan x - x$ cuando x está cerca de cero?

- 26.** Encuentre maneras de calcular estas funciones sin pérdidas graves de números significativos:

a. $e^x - \sin x - \cos x$ **b.** $\ln(x) - 1$ **c.** $\log x - \log(1/x)$

d. $x^{-2}(\sin x - e^x + 1)$ **e.** $x - \operatorname{arctanh} x$

- 27.** Sea

$$a(x) = \frac{1 - \cos x}{\sin x} \quad b(x) = \frac{\sin x}{1 + \cos x} \quad c(x) = \frac{x}{2} + \frac{x^3}{24}$$

Muestre que $b(x)$ es idéntico a $a(x)$ y que $c(x)$ se aproxima a $a(x)$ en la vecindad de cero.

- “28.** En su computadora determine el rango de x para el que $(\sin x) / x \approx 1$ con precisión de máquina total. *Sugerencia:* use series de Taylor.

- “29.** El uso de la familiar fórmula cuadrática

$$x = \frac{1}{2a} \left(-b \pm \sqrt{b^2 - 4ac} \right)$$

ocasiona un problema cuando la ecuación cuadrática $x^2 - 10^5x + 1 = 0$ se resuelve con una máquina que tiene sólo ocho dígitos decimales. Investigue el ejemplo, observe la dificultad y proponga una solución. *Sugerencia:* hay un ejemplo similar en el libro.

- ^a30. Cuando se quieren valores exactos para las raíces de una ecuación cuadrática, puede ocurrir cierta pérdida de significancia si $b^2 \approx 4ac$. ¿Qué se puede hacer (si hay algo) para evitar esto cuando se escribe una rutina de cómputo?
31. Refiérase al análisis de la función $f(x) = x - \sin x$ presentado en el libro. Muestre que cuando $0 < x < 1.9$ no habrá pérdida indebida de significancia de la resta en la ecuación (3).
32. Analice el problema de calcular $\tan(10^{100})$. (Véase Gleick [1992], p. 178.)
33. Sean x y y dos números de máquina binarios normalizados de punto flotante. Suponga que $x = q \times 2^n, y = r \times 2^{n-1}, \frac{1}{2} \leq r, q < 1$ y $2q - 1 \geq r$. ¿Cuánta pérdida de significancia ocurre al restar $x - y$? Responda la misma pregunta cuando $2q - 1 < r$. Observe que el teorema de pérdida de precisión no es lo suficientemente fuerte para resolver este problema con precisión.
34. Demuestre la primera parte del teorema de pérdida de precisión.
35. Muestre que si x es un número de máquina en una computadora de 32 bits que satisface la desigualdad $x > \pi 2^{25}$, entonces $\sin x$ se calculará *sin* dígitos significativos.
36. Sean x y y dos números de máquina positivos de punto flotante normalizados en una computadora de 32 bits. Sea $x = q \times 2^m$ y $y = r \times 2^n$ con $\frac{1}{2} \leq r, q < 1$. Muestre que si $n = m$, entonces al menos se pierde un bit significativo en la resta $x - y$.
37. (**Proyecto de investigación para el estudiante**) Lea y analice la diferencia entre error *por eliminación*, un *mal* algoritmo y un problema *mal acondicionado*. Sugerencia: un ejemplo implica la ecuación cuadrática. Consulte Stewart [1996].
38. En una computadora de tres dígitos significativos, calcule, $\sqrt{9.01} - 3.00$ con tanta exactitud como sea posible.

Problemas de cómputo 2.2

1. Escriba una rutina para calcular las dos raíces x_1 y x_2 de la ecuación cuadrática $f(x) = ax^2 + bx + c = 0$ con constantes reales a, b y c y para evaluar $f(x_1)$ y $f(x_2)$. Use fórmulas que reduzcan error de redondeo y escriba un código eficiente. Pruebe su rutina con los siguientes valores de (a, b, c) : $(0, 0, 1); (0, 1, 0); (1, 0, 0); (0, 0, 0); (1, 1, 0); (2, 10, 1); (1, -4, 3.99999); (1, -8.01, 16.004); (2 \times 10^{17}, 10^{18}, 10^{17})$; y $(10^{-17}, -10^{17}, 10^{17})$.
2. (Continuación) Escriba y pruebe una rutina para resolver una ecuación cuadrática que pueda tener raíces complejas.
3. Cambie y pruebe el seudocódigo del libro para calcular $x - \sin x$ usando multiplicación anidada para evaluar la serie.
4. Escriba una rutina para la función $f(x) = e^x - e^{-2x}$ usando los ejemplos del libro como guía.
5. Escriba un código usando precisión doble o extendida para evaluar $f(x) = \cos(10^4 x)$ en el intervalo $[0, 1]$. Determine cuántos números significativos tendrán los valores de $f(x)$.

6. Escriba un procedimiento para calcular $f(x) = \sin x - 1 + \cos x$. La rutina debe producir casi una precisión total de máquina para toda x en el intervalo $[0, \pi/4]$. *Sugerencia:* la identidad trigonométrica $\sin^2 \theta = \frac{1}{2}(1 - \cos 2\theta)$ puede ser útil.
7. Escriba un procedimiento para calcular $f(x, y) = \int_1^x t^y$ para x y y arbitraria. *Nota:* observe el caso excepcional $y = -1$ y el problema numérico *cerca* del caso excepcional.
8. Suponga que queremos evaluar la función $f(x) = (x - \sin x)/x^3$ para valores de x cercanos a cero.
 - a. Escriba una rutina para esta función. Evalúe $f(x)$ diecisésis veces. Inicialmente, sea $x \leftarrow 1$ y después sea $x \leftarrow \frac{1}{10}x$ quince veces. Explique los resultados. *Nota:* la regla de L'Hôpital indica que $f(x)$ debe tender a $\frac{1}{6}$. Pruebe este código.
 - b. Escriba un procedimiento de función que produzca más valores exactos de $f(x)$ para todos los valores de x . Pruebe este código.
9. Escriba un programa para imprimir una tabla de la función $f(x) = 5 - \sqrt{25 + x^2}$ para $x = 0$ a 1 con pasos de 0.01. Asegúrese de que su programa produzca precisión de máquina total, pero no programe el problema con doble precisión. Explique los resultados.
- 10.** Escriba una rutina que calcule e^x al sumar n términos de la serie de Taylor hasta el $n+1$ ésimo término t sea tal que $|t| < \varepsilon = 10^{-6}$. Use el recíproco de e^x para valores negativos de x . Pruebe con los datos siguientes: 0, +1, -1, 0.5, -0.123, -25.5, -1776, 3.14159. Calcule el error relativo, el error absoluto y n para cada caso, usando la función exponencial en su sistema de cómputo para el valor exacto. Sume no más que 25 términos.
- 11.** (Continuación) El cálculo de e^x se puede reducir a sólo calcular e^u para $|u| < (\ln 2)/2$. Este algoritmo elimina potencias de 2 y calcula e^u en un rango donde la serie converge muy rápidamente. Está dado por

$$e^x = 2^m e^u$$

donde m y u se calculan con los pasos

$$\begin{aligned} z &\leftarrow x / \ln 2; & m &\leftarrow \text{integer}(z \pm \frac{1}{2}) \\ w &\leftarrow z - m; & u &\leftarrow w \ln 2 \end{aligned}$$

Aquí se usa el signo menos si $x < 0$ ya que $z < 0$. Incorpore esta técnica de reducción de rango en el código.

- 12.** (Continuación) Escriba una rutina que use reducción de rango $e^x = 2^m e^u$ y calcule e^u de la parte par de la *fracción gaussiana continuada*; es decir,

$$e^u = \frac{s + u}{s - u} \quad \text{donde} \quad s = 2 + u^2 \left(\frac{2520 + 28u^2}{15120 + 420u^2 + u^4} \right)$$

Pruebe con los datos dados en el problema de cómputo 2.2.10. *Nota:* algunos problemas de cómputo de esta sección tienen algoritmos más bien complicados para calcular diferentes funciones intrínsecas que corresponden a las usadas en realidad en un sistema de cómputo mainframe. Descripciones de éstas y otras funciones de biblioteca similares con frecuencia se encuentran en la documentación de apoyo de su sistema de cómputo.

- 13.** En muchas numéricas es muy importante el cálculo exacto del valor absoluto $|z|$ de un número complejo $z = a + bi$. Diseñe y realice un experimento de cómputo para comparar los tres es-

quemas siguientes:

a. $|z| = (a^2 + b^2)^{1/2}$

b. $|z| = v \left[1 + \left(\frac{w}{v} \right)^2 \right]^{1/2}$

c. $|z| = 2v \left[\frac{1}{4} + \left(\frac{w}{2v} \right)^2 \right]^{1/2}$

donde $v = \max\{|a|, |b|\}$ y $w = \min\{|a|, |b|\}$. Use números muy pequeños y grandes para el experimento.

- 14.** ¿Para qué rango de x es la aproximación $(e^x - 1)/2x \approx 0.5$ correcto con 15 dígitos decimales de exactitud? Usando esta información, escriba procedimiento de función para $(e^x - 1)/2x$, produciendo 15 decimales de exactitud en todo el intervalo $[-10, 10]$.

- 15.** En la teoría de series de Fourier, algunos números conocidos como **constantes de Lebesgue** resultan importantes. Una fórmula para éstos es

$$\rho_n = \frac{1}{2n+1} + \frac{2}{\pi} \sum_{k=1}^n \frac{1}{k} \tan \frac{\pi k}{2n+1}$$

Escriba y corra un programa para calcular $\rho_1, \rho_2, \dots, \rho_{100}$ con ocho dígitos decimales de exactitud. Después pruebe la validez de la desigualdad

$$0 \leq \frac{4}{\pi^2} \ln(2n+1) + 1 - \rho_n \leq 0.0106$$

- 16.** Calcule con precisión doble o extendida el número siguiente:

$$x = \left[\frac{1}{\pi} \ln(640320^3 + 744) \right]^2$$

¿Cuál es el punto de este problema? (Consulte Good [1972].)

- 17.** Escriba una rutina para calcular $\sin x$ para x en radianes como sigue. Primero, usando propiedades de la función seno, reduzca el rango de modo que $-\pi/2 \leq x \leq \pi/2$. Entonces, si $|x| < 10^{-8}$, haga $\sin x \approx x$; si $|x| > \pi/6$, haga $u = x/3$, calcule $\sin u$ con la fórmula de abajo y después haga $\sin x \approx [3 - 4\sin^2 u] \sin u$; si $|x| \leq \pi/6$, haga $u = x$ y calcule $\sin u$ como se muestra a continuación:

$$\sin u \approx u \left[\frac{1 - \left(\frac{29593}{207636} \right) u^2 + \left(\frac{34911}{7613320} \right) u^4 - \left(\frac{479249}{11511339840} \right) u^6}{1 + \left(\frac{1671}{69212} \right) u^2 + \left(\frac{97}{351384} \right) u^4 + \left(\frac{2623}{1644477120} \right) u^6} \right]$$

Intente determinar si la función seno en su sistema de cómputo usa este algoritmo. *Nota:* ésta es la aproximación racional de Padé para el seno.

- 18.** Escriba una rutina para calcular el logaritmo natural con el algoritmo que aquí se indica basado en los *racionales condensados y fracciones gaussianas continuadas* para $\ln x$ y pruebe para varios valores de x . Primero compruebe si $x = 1$ y regrese a cero si es así. Reduzca el rango de x al determinar n y r tal que $x = r \times 2^n$ con $\frac{1}{2} \leq r < 1$. Despues, haga $u = (r - \sqrt{2}/2)/(r + \sqrt{2}/2)$, y calcule $\ln[(1+u)/(1-u)]$ por la aproximación

$$\ln \left(\frac{1+u}{1-u} \right) \approx u \left(\frac{20790 - 21545.27u^2 + 4223.9187u^4}{10395 - 14237.635u^2 + 4778.8377u^4 - 230.41913u^6} \right)$$

que es válida para $|u| < 3 - 2\sqrt{2}$. Por último, haga

$$\ln x \approx \left(n - \frac{1}{2}\right) \ln 2 + \ln \left[\frac{1+u}{1-u} \right]$$

- 19.** Escriba una rutina para calcular la tangente de x en radianes, usando el algoritmo de abajo. Pruebe la rutina resultante en un rango de valores de x . Primero, el argumento x se reduce $|x| \leq \pi/2$ sumando o restando múltiplos de π . Si tenemos $0 \leq |x| \leq 1.7 \times 10^{-9}$, haga $\tan x \approx x$. Si $|x| > \pi/4$, haga $u = \pi/2 - x$; por otra parte, haga $u = x$. Ahora calcule la aproximación

$$\tan u \approx u \left(\frac{135135 - 17336.106u^2 + 379.23564u^4 - 1.0118625u^6}{135135 - 62381.106u^2 + 3154.9377u^4 + 28.17694u^6} \right)$$

Por último, si $|x| > \pi/4$, haga $\tan x \approx 1/\tan u$; si $|x| \leq \pi/4$, haga $\tan x \approx \tan u$. *Nota:* este algoritmo se obtiene del *racional condensado* y de la *fracción gaussiana continuada* para la función tangente.

- 20.** Escriba una rutina para calcular el arccsen x con base en el algoritmo siguiente, usando polinomios condensados para el arccsen. Si $|x| < 10^{-8}$, haga $\arccsen x \approx x$. Por otra parte, si $0 \leq x \leq \frac{1}{2}$, haga $u = x$, $a = 0$ y $b = 1$; si $\frac{1}{2} < x \leq \frac{1}{2}\sqrt{3}$, haga $u = 2x^2 - 1$, $a = \pi/4$ y $b = \frac{1}{2}$; si $\frac{1}{2}\sqrt{3} < x \leq \frac{1}{2}\sqrt{2 + \sqrt{3}}$, haga $u = 8x^4 - 8x^2 + 1$, $a = 3\pi/8$ y $b = \frac{1}{4}$; si $\frac{1}{2}\sqrt{2 + \sqrt{3}} < x \leq 1$, haga $u = \sqrt{\frac{1}{2}(1-x)}$, $a = \pi/2$ y $b = -2$. Ahora calcule la aproximación

$$\begin{aligned} \arccsen u \approx u &\left(1.0 + \frac{1}{6}u^2 + 0.075u^4 + 0.04464286u^6 + 0.03038182u^8 \right. \\ &+ 0.022375u^{10} + 0.01731276u^{12} + 0.01433124u^{14} \\ &+ 0.009342806u^{16} + 0.01835667u^{18} - 0.01186224u^{20} \\ &\left. + 0.03162712u^{22} \right) \end{aligned}$$

Por último, haga $\arccsen x \approx a + b \arccsen u$. Pruebe esta rutina para diferentes valores de x .

- 21.** Escriba y pruebe una rutina para calcular $\arctan x$ para x en radianes como sigue. Si $0 \leq x \leq 1.7 \times 10^{-9}$, haga $\arctan x \approx x$. Si $1.7 \times 10^{-9} < x \leq 2 \times 10^{-2}$, use la aproximación en serie

$$\arctan x \approx x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7}$$

Por otro lado, haga $y = x$, $a = 0$ y $b = 1$ si $0 \leq x \leq 1$; haga $y = 1/x$, $a = \pi/2$ y $b = -1$ si $1 < x$. Despues haga $c = \pi/16$ y $d = \tan c$ si $0 \leq y \leq \sqrt{2}$ y $c = 3\pi/16$ y $d = \tan c$ si $\sqrt{2} - 1 < y \leq 1$. Calcule $u = (y - d)/(1 + dy)$ y la aproximación

$$\arctan u \approx u \left(\frac{135135 + 171962.46u^2 + 52490.4832u^4 + 2218.1u^6}{135135 + 217007.46u^2 + 97799.3033u^4 + 10721.3745u^6} \right)$$

Por último, haga $\arctan x \approx a + b(c + \arctan u)$. *Nota:* este algoritmo usa el *racional condensado* y las *fracciones gaussianas continuadas*.

- 22.** Un algoritmo rápido para calcular $\arctan x$ con n bits de precisión para x en el intervalo $(0, 1]$ es como sigue. Haga $a = 2^{-n/2}$, $b = x/(1 + \sqrt{1 + x^2})$, $c = 1$ y $d = 1$. Despues repetidamente

actualice estas variables con estas fórmulas (en orden de izquierda a derecha y de arriba abajo):

real a, b, c, d

$$c \leftarrow \frac{2c}{1+a}; \quad d \leftarrow \frac{2ab}{1+b^2}; \quad d \leftarrow \frac{d}{1+\sqrt{1-d^2}}$$

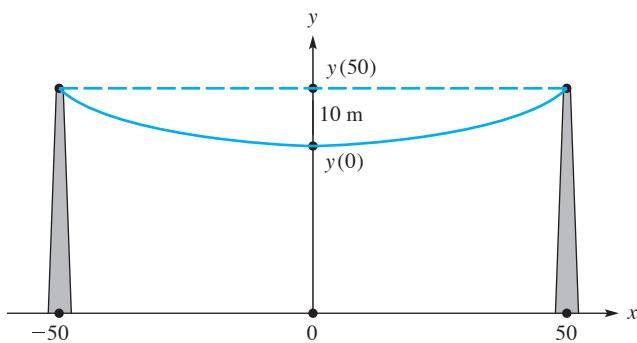
$$d \leftarrow \frac{b+d}{1-bd}; \quad b \leftarrow \frac{d}{1+\sqrt{1+d^2}}; \quad a \leftarrow \frac{2\sqrt{a}}{1+a}$$

Después de cada paso, imprima $f = c \ln [(1+b)/(1-b)]$. Pare cuando $1-a \leq 2^{-n}$. Escriba una rutina de doble precisión para implementar este algoritmo y pruébelo con diferentes valores de x . Compare el resultado con los obtenidos con la función arco tangente de su sistema de cómputo. *Nota:* este algoritmo rápido de precisión múltiple depende de la teoría de las *integrales elípticas*, usando la iteración aritmética-geométrica media y las *transformaciones ascendentes de Landen*. En Brent [1976] se analizan otros algoritmos rápidos para funciones trigonométricas.

23. En su computadora, muestre que en precisión simple tiene sólo seis dígitos decimales de exactitud si introduce 20 dígitos. Muestre que ir a doble precisión sólo es eficaz si todo el trabajo se hace en doble precisión. Por ejemplo, si usa $\text{pi} = 3.14$ o $\text{pi} = 22/7$, perderá toda la precisión que ha ganado al usar doble precisión. ¡Recuerde que el número de dígitos significativos en los resultados finales es lo que cuenta!
24. En algunos lenguajes de programación tales como Java y C++, muestre que la aritmética de modo mezclado puede conducir a resultados tales como $(4/3) * \text{pi} = \text{pi}$ cuando pi es un número de punto flotante, ya que la fracción dentro del paréntesis se calcula en modo entero.
25. **(Proyecto de investigación para el estudiante)** Investigue la aritmética de intervalos, lo cual tiene como objetivo obtener resultados con una precisión garantizada.

Localización de raíces de ecuaciones

Un cable de energía eléctrica está suspendido (de puntos de igual altura) de dos torres que están a una distancia de 100 metros. El cable cuelga 10 metros a la mitad. ¿Cuán largo es el cable?



Se sabe que la curva hecha por un cable suspendido es una **catenaria**. Cuando el eje y pasa por el punto mínimo, podemos suponer una ecuación de la forma $y = \lambda \cosh(x/\lambda)$. Aquí λ es un parámetro que debe ser determinado. Las condiciones del problema son que $y(50) = y(0) + 10$. Por tanto, obtenemos

$$\lambda \cosh\left(\frac{50}{\lambda}\right) = \lambda + 10$$

Con los métodos de este capítulo se halla que el parámetro es $\lambda = 126.632$. Después se sustituye este valor en la fórmula de la longitud de arco de la catenaria y se determina que la longitud es de 102.619 metros (véase el problema de cómputo 5.1.4).

3.1 Método de bisección

Introducción

Sea f una función con valores reales o complejos de una variable real o compleja. Un número r , real o complejo, para el que $f(r) = 0$ se llama una **raíz** de la ecuación o un **cero** de f . Por ejemplo, la función

$$f(x) = 6x^2 - 7x + 2$$

tiene $\frac{1}{2}$ y $\frac{2}{3}$ como ceros, como se puede comprobar sustituyendo directamente o escribiendo f en su forma factorizada:

$$f(x) = (2x - 1)(3x - 2)$$

Otro ejemplo, la función

$$g(x) = \cos 3x - \cos 7x$$

no sólo tiene el cero obvio $x = 0$, sino también todos los múltiplos enteros de $\pi/5$ y de $\pi/2$, lo que descubrimos al aplicar la identidad trigonométrica

$$\cos A - \cos B = 2 \sin\left[\frac{1}{2}(a + b)\right] \sin\left[\frac{1}{2}(b - a)\right]$$

Por tanto, encontramos

$$g(x) = 2 \sin(5x) \sin(2x)$$

¿Por qué es importante localizar raíces? Con frecuencia, la solución de un problema científico es un número del que tenemos poca información y sólo sabemos que satisface cierta ecuación. Puesto que toda ecuación puede escribirse de tal manera que la función se encuentre en un miembro y cero en el otro, el número que se quiere determinar debe ser un cero de la función. Así, si tenemos un conjunto de métodos para localizar ceros de funciones podremos resolver dichos problemas.

Ejemplificamos esta aseveración al usar un problema específico de ingeniería cuya solución es la raíz de una ecuación. En un cierto circuito eléctrico, el voltaje V y la corriente I están relacionados con dos ecuaciones de la forma

$$\begin{cases} I = a(e^{bV} - 1) \\ c = dI + V \end{cases}$$

en las que a, b, c y d son constantes. Para nuestro objetivo, se supone que se conocen estos cuatro números. Cuando se combinan estas ecuaciones al eliminarse I entre ellas, el resultado es una ecuación simple:

$$c = ad(e^{bV} - 1) + V$$

En un caso concreto, se puede reducir a

$$12 = 14.3(e^{2V} - 1) + V$$

y se requiere su solución. (Despejando se encuentra que en este caso $V \approx 0.299$.)

En algunos problemas en los que se busca la raíz de una ecuación podemos realizar el cálculo necesario con una calculadora manual. Pero ¿cómo podemos localizar ceros de funciones tan complicadas como éstas?

$$\begin{aligned} f(x) &= 3.24x^8 - 2.42x^7 + 10.34x^6 + 11.01x^2 + 47.98 \\ g(x) &= 2^{x^2} - 10x + 1 \\ h(x) &= \cosh(\sqrt{x^2 + 1}) - e^x + \log|\sin x| \end{aligned}$$

Se necesita un método numérico general que no dependa de las propiedades especiales de nuestras funciones. Por supuesto, la continuidad y la derivabilidad son propiedades especiales, pero son

atributos comunes de funciones que normalmente se encuentran. La clase de propiedad especial que quizás no podemos aprovechar fácilmente en códigos de propósito general está tipificado por la identidad trigonométrica mencionada en párrafos atrás.

Cientos de métodos están disponibles para localizar ceros de funciones y se han seleccionado tres de los más útiles para estudiarlos aquí: el método de bisección, el método de Newton y el método de la secante.

Sea f una función que tiene valores de signos opuestos en los dos extremos de un intervalo. Suponga también que f es continua en ese intervalo. Para fijar la notación, sea $a < b$ y $f(a) \cdot f(b) < 0$. Por ello f tiene una raíz en el intervalo (a, b) . En otras palabras, debe existir un número r que satisface las dos condiciones $a < r < b$ y $f(r) = 0$. ¿Cómo se llegó a esta conclusión? Se debe recordar el **teorema del valor intermedio**.^{*} Si x recorre un intervalo $[a, b]$, entonces los valores de $f(x)$ llenan por completo el intervalo entre $f(a)$ y $f(b)$. No se pueden omitir los valores intermedios. Por tanto, una función específica f debe tomar el valor cero en alguna parte del intervalo (a, b) , ya que $f(a)$ y $f(b)$ son de signos opuestos.

Algoritmo y seudocódigo de la bisección

El **método de bisección** aprovecha esta propiedad de las funciones continuas. En cada paso de este algoritmo tenemos un intervalo $[a, b]$ y los valores $u = f(a)$ y $v = f(b)$. Los números u y v satisfacen $uv < 0$. Ahora, construimos el punto medio del intervalo, $c = \frac{1}{2}(a + b)$, y calculamos $w = f(c)$. Puede ocurrir accidentalmente que $f(c) = 0$. Si esto pasa, se ha satisfecho el objetivo del algoritmo. En el caso usual, $w \neq 0$, y entonces $wu < 0$ o $wv < 0$. (¿Por qué?) Si $wu < 0$, podemos asegurar que una raíz de f existe en el intervalo $[a, c]$. Por tanto, almacenamos el valor de c en b y de w en v . Si $wu > 0$, entonces no podemos asegurar que f tiene una raíz en $[a, c]$, pero puesto que $wv < 0$, f debe tener una raíz en $[c, b]$. En este caso, almacenamos el valor de c en a y de w en u . En cualquier caso, la situación al final de este paso es exactamente igual que al principio excepto que el intervalo final es la mitad de largo que el intervalo inicial. Este paso ahora se puede repetir hasta que el intervalo sea satisfactoriamente pequeño, digamos $|b - a| < \frac{1}{2} \times 10^{-6}$. Al final, el mejor cálculo de la raíz sería $(a + b)/2$, donde $[a, b]$ es el último intervalo en el procedimiento.

Ahora permítanos construir el seudocódigo para llevar a cabo este procedimiento. No intentaremos crear un software de alta calidad con muchas “campanas y silbatos”, pero escribiremos el seudocódigo en la forma de un procedimiento para uso general. Esto le brindará a usted una oportunidad de repasar cómo se pueden conectar un programa principal y uno o más procedimientos.

Como una regla general, en rutinas de programación para localizar las raíces de funciones arbitrarias se deben evitar evaluaciones innecesarias de la función, ya que puede ser costoso evaluar una función dada en términos de tiempo de computadora. Por ello, cualquier valor de la función que se pueda necesitar posteriormente se debe guardar más que volver a calcular. Una programación descuidada del método de bisección podría violar este principio.

El procedimiento construido operará con una función arbitraria f . También se especifica un intervalo $[a, b]$, así como el número de pasos, $nmax$ el seudocódigo para dar.

* El **teorema del valor intermedio** establece lo siguiente: si la función f es continua en el intervalo cerrado $[a, b]$ y si $f(a) \leq y \leq f(b)$ o $f(b) \leq y \leq f(a)$, entonces existe un punto c tal que $a \leq c \leq b$ y $f(c) = y$.

n_{max} pasos en el algoritmo de la bisección siguiente:

```

procedure Bisección( $f, a, b, n_{max}, \varepsilon$ )
integer  $n, n_{max};$  real  $a, b, c, fa, fb, fc, error$ 
 $fa \leftarrow f(a)$ 
 $fb \leftarrow f(b)$ 
if sign( $fa$ ) = sign( $fb$ ) then
    output  $a, b, fa, fb$ 
    output “la función tiene el mismo signo en  $a$  y en  $b$ ”
    return
end if
 $error \leftarrow b - a$ 
for  $n = 0$  to  $n_{max}$  do
     $error \leftarrow error/2$ 
     $c \leftarrow a + error$ 
     $fc \leftarrow f(c)$ 
    output  $n, c, fc, error$ 
    if  $|error| < \varepsilon$  then
        output “convergencia”
        return
    end if
    if sign( $fa$ ) ≠ sign( $fc$ ) then
         $b \leftarrow c$ 
         $fb \leftarrow fc$ 
    else
         $a \leftarrow c$ 
         $fa \leftarrow fc$ 
    end if
end for
end procedure Bisección

```

Se han incorporado muchas modificaciones para enriquecer el seudocódigo. Por ejemplo, usamos fa, fb y fc como mnemónicos para u, v y w , respectivamente. También, ejemplificamos algunas técnicas de programación estructurada y algunas otras alternativas, tales como una prueba de convergencia. Por ejemplo, si u, v o w están cerca de cero, entonces uv o wu pueden hacer un subflujo. De manera similar, puede surgir una situación de sobreflujo. Una prueba que implica a la función intrínseca $sign$ se podría usar para evitar estas dificultades, así como una prueba que determine si $sign(u) \neq sign(v)$. Aquí, terminan las iteraciones si exceden n_{max} o si el límite de error (que analizaremos más tarde en esta sección) es menor que ε . Usted debe seguir los pasos en la rutina para ver qué hace lo que se dice.

Ejemplos

Ahora queremos mostrar cómo se pueden usar el seudocódigo de la bisección. Suponga que tenemos dos funciones y para cada una buscamos un cero en un intervalo dado:

$$\begin{aligned}f(x) &= x^3 - 3x + 1 && \text{en } [0, 1] \\g(x) &= x^3 - 2 \sin x && \text{en } [0.5, 2]\end{aligned}$$

Primero, escribimos dos funciones de procedimiento para calcular $f(x)$ y $g(x)$. Después introducimos los intervalos iniciales y el número de pasos que se darán en el programa principal. Puesto que se trata de un ejemplo sencillo, esta información se podría asignar directamente en el programa principal o con expresiones en los subprogramas más que para que la lea el programa. También, según el lenguaje de computadora que se esté usando, se necesita un enunciado o interface externos para decirle al compilador que el parámetro f en el procedimiento de la bisección *no* es una variable ordinaria con valores numéricos sino el nombre de una función de procedimiento definida externamente del programa principal. En este ejemplo, habría dos de estas funciones de procedimiento y dos llamadas al procedimiento de la bisección.

Un programa de llamada o programa principal que llama a la segunda rutina de la bisección se podría escribir como se muestra a continuación:

```

program Prueba_Bisección
integer n, nmax ← 20
real a, b, ε ←  $\frac{1}{2}10^{-6}$ 
external function f, g
    a ← 0.0
    b ← 1.0
    call Bisección(f, a, b, nmax, ε)
    a ← 0.5
    b ← 2.0
    call Bisección(g, a, b, nmax, ε)
end program Prueba_Bisección

real function f(x)
real x
    f ←  $x^3 - 3x + 1$ 
end function f

real function g(x)
real x
    g ←  $x^3 - 2 \operatorname{sen} x$ 
end function g

```

Los resultados de computadora para los pasos iterativos del método de bisección para $f(x)$:

n	c_n	$f(c_n)$	error
0	0.5	-0.375	0.5
1	0.25	0.266	0.25
2	0.375	-7.23×10^{-2}	0.125
3	0.3125	9.30×10^{-2}	6.25×10^{-2}
4	0.34375	9.37×10^{-3}	3.125×10^{-2}
⋮			
19	0.34729 67	-9.54×10^{-7}	9.54×10^{-7}
20	0.34729 62	3.58×10^{-7}	4.77×10^{-7}

También, los resultados para $g(x)$ son los siguientes:

n	c_n	$g(c_n)$	error
0	1.25	5.52×10^{-2}	0.75
1	0.875	-0.865	0.375
2	1.0625	-0.548	0.188
3	1.15625	-0.285	9.38×10^{-2}
4	1.203125	-0.125	4.69×10^{-2}
:			
19	1.2361827	-4.88×10^{-6}	1.43×10^{-6}
20	1.2361834	-2.15×10^{-6}	7.15×10^{-7}

Para comprobar estos resultados usamos procedimientos incorporados en software matemático como Matlab, Mathematica o Maple para encontrar que las raíces deseadas de f y g son 0.34729 63553 y 1.23618 3928, respectivamente. Puesto que f es un polinomio, podemos usar una rutina para determinar aproximaciones numéricas a todos los ceros (o raíces) de una función polinomial. Sin embargo, cuando se trata de funciones polinomiales más complicadas, existe generalmente un procedimiento no sistemático para determinar todos los ceros. En este caso, se puede usar una rutina para localizar ceros (uno a la vez), pero tenemos que especificar un punto en el que ha de empezar la búsqueda, y puntos diferentes de inicio pueden dar como resultado el mismo o diferentes ceros. Puede ser particularmente problemático encontrar todas las raíces de una función cuyo comportamiento no se conoce.

Análisis de convergencia

Ahora permítanos investigar la *exactitud* con la que el método de bisección determina una raíz de una función. Suponga que f es una función continua que toma valores de signo contrario en los extremos de un intervalo $[a_0, b_0]$. Entonces hay una raíz r en $[a_0, b_0]$, y si usamos el punto medio $c_0 = (a_0 + b_0)/2$ como nuestros cálculos de r , tenemos

$$|r - c_0| \leq \frac{b_0 - a_0}{2}$$

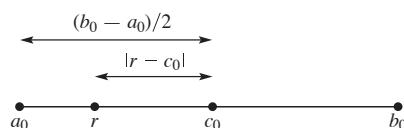
como se muestra en la figura 3.1. Si ahora se aplica el algoritmo de la bisección y si las cantidades calculadas se denotan por $a_0, b_0, c_0, a_1, b_1, c_1$ y así sucesivamente, entonces, por el mismo razonamiento,

$$|r - c_n| \leq \frac{b_n - a_n}{2} \quad (n \geq 0)$$

Puesto que los anchos de los intervalos se dividen entre 2 en cada paso, concluimos que

$$|r - c_n| \leq \frac{b_0 - a_0}{2^{n+1}} \tag{1}$$

FIGURA 3.1
Método de bisección:
ejemplificación
del error de
límite superior



Para resumir:

■ TEOREMA 1

Teorema del método de bisección

Si el algoritmo de bisección se aplica a una función continua f en un intervalo $[a, b]$, donde $f(a)f(b) < 0$, entonces, después de n pasos se habrá calculado una raíz aproximada con un error a lo más de $(b - a)/2^{n+1}$.

Si una tolerancia de error se ha señalado de antemano, se puede determinar el número de pasos requeridos en el método de bisección. Suponga que queremos $|r - c_n| < \varepsilon$. Entonces es necesario resolver la siguiente desigualdad para n :

$$\frac{b - a}{2^{n+1}} < \varepsilon$$

Al tomar logaritmos (con cualquier base conveniente), obtenemos

$$n > \frac{\log(b - a) - \log(2\varepsilon)}{\log 2} \quad (2)$$

EJEMPLO 1 ¿Cuántos pasos del algoritmo de bisección son necesarios para calcular una raíz de f con toda la precisión simple de la máquina en una computadora de longitud de palabra de 32 bits si $a = 16$ y $b = 17$?

Solución La raíz está entre los dos números binarios $a = (10\ 000.0)_2$ y $b = (10\ 001.0)_2$. Por tanto, ya conocemos cinco de los dígitos binarios en la respuesta. Puesto que podemos usar sólo 24 bits en total, quedan 19 bits por determinar. Queremos que el último sea correcto, así que buscamos que el error sea menor que 2^{-19} o 2^{-20} (siendo conservadores). Puesto que una computadora de longitud de palabra de 32 bits tiene una mantisa de 24 bits, podemos esperar que la respuesta tenga una exactitud de sólo 2^{-20} . De la ecuación anterior, queremos $(b - a)/2^{n+1} < \varepsilon$. En virtud de que $b - a = 1$ y $\varepsilon = 2^{-20}$, tenemos que $1/2^{n+1} < 2^{-20}$. Tomando recíprocos se obtiene $2^{n+1} > 2^{20}$, o $n \geq 20$. Alternativamente, podemos usar la ecuación (2), que en este caso es

$$n > \frac{\log 1 - \log 2^{-19}}{\log 2}$$

Usando una propiedad básica de logaritmos ($\log x^y = y \log x$), encontramos que $n \geq 20$. En este ejemplo, cada paso del algoritmo determina la raíz con un dígito binario adicional de precisión. ■

Una secuencia $\{x_n\}$ presenta **convergencia lineal** a un límite x si hay una constante C en el intervalo $[0, 1)$ tal que

$$|x_{n+1} - x| \leq C|x_n - x| \quad (n \geq 1) \quad (3)$$

Si esta desigualdad es válida para toda n , entonces

$$|x_{n+1} - x| \leq C|x_n - x| \leq C^2|x_{n-1} - x| \leq \cdots \leq C^n|x_1 - x|$$

Por esto, una consecuencia de la convergencia lineal es

$$|x_{n+1} - x| \leq AC^n \quad (0 \leq C < 1) \quad (4)$$

La sucesión producida por el método de bisección obedece a la desigualdad (4), como vemos de la ecuación (1). Sin embargo, la sucesión no necesariamente obedece a la desigualdad (3).

El método de bisección es la forma más simple de resolver una ecuación no lineal $f(x) = 0$. Obtiene la raíz restringiendo el intervalo en el que se encuentra una raíz y finalmente hace el intervalo muy pequeño. Ya que el método de bisección divide en dos el ancho del intervalo en cada paso, se puede predecir exactamente cuán largo será para encontrar la raíz dentro de cualquier grado de exactitud deseado. En el método de bisección, no todos los valores supuestos están más cerca de la raíz que el supuesto anterior, ya que no usa la naturaleza de la función misma. Con frecuencia el método de bisección se usa para acercarse a la raíz antes de cambiarse a un método más rápido.

Método de falsa posición (*regula falsi*) y modificaciones

El **método de falsa posición** conserva la característica principal del método de bisección: que una raíz está atrapada en una sucesión de intervalos de tamaño decreciente. Más que seleccionar el punto medio de cada intervalo, este método usa el punto donde las rectas secantes se intersecan con el eje x .

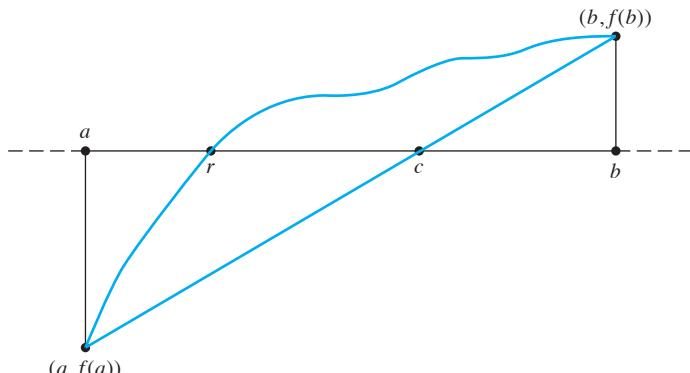


FIGURA 3.2
Método de falsa
posición

En la figura 3.2, la recta secante en el intervalo $[a, b]$ es la cuerda entre $(a, f(a))$ y $(b, f(b))$. Los dos triángulos rectángulos en la figura son *semejantes*, lo que significa que

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

Es fácil mostrar que

$$c = b - f(b) \left[\frac{a - b}{f(a) - f(b)} \right] = a - f(a) \left[\frac{b - a}{f(b) - f(a)} \right] = \frac{af(b) - bf(a)}{f(b) - f(a)}$$

Entonces calculamos $f(c)$ y continuamos al siguiente paso con el intervalo $[a, c]$ si $f(a)f(c) < 0$ o con el intervalo $[c, b]$ si $f(c)f(b) < 0$.

En el caso general, el **método de falsa posición** inicia con el intervalo $[a_0, b_0]$ que contiene una raíz: $f(a_0)$ y $f(b_0)$ son de signos contrarios. Este método usa intervalos $[a_k, b_k]$ que contienen raíces casi de la misma forma que lo hace el método de bisección. Sin embargo, en lugar de determinar el punto medio del intervalo, encuentra el punto donde la recta secante que une a $(a_k, f(a_k))$ con $(b_k, f(b_k))$ corta el eje x y después lo escoge como el nuevo punto extremo.

En el k ésimo paso, el método de falsa posición calcula

$$c_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$$

Si $f(a_k)$ y $f(c_k)$ tienen mismo signo, entonces $a_{k+1} = c_k$ y $b_{k+1} = c_k$; por otro lado, $a_{k+1} = a_k$ y $b_{k+1} = c_k$. El proceso se repite hasta que la raíz se approxima suficientemente bien.

Por algunas funciones, el método de falsa posición puede seleccionar repetidamente el mismo punto extremo y el proceso puede degradarse a convergencia lineal. Hay varios métodos para rectificar esto. Por ejemplo, cuando el mismo punto extremo se utiliza dos veces, el **método modificado de falsa posición** usa

$$c_k^{(m)} = \begin{cases} \frac{a_k f(b_k) - 2b_k f(a_k)}{f(b_k) - 2f(a_k)}, & \text{si } f(a_k)f(b_k) < 0 \\ \frac{2a_k f(b_k) - b_k f(a_k)}{2f(b_k) - f(a_k)}, & \text{si } f(a_k)f(b_k) > 0 \end{cases}$$

Por ello, más que seleccionar puntos en el mismo lado de la raíz como con lo hace el método regular de falsa posición, el método modificado de falsa posición cambia la pendiente de la recta porque está más cerca de la raíz (figura 3.3).

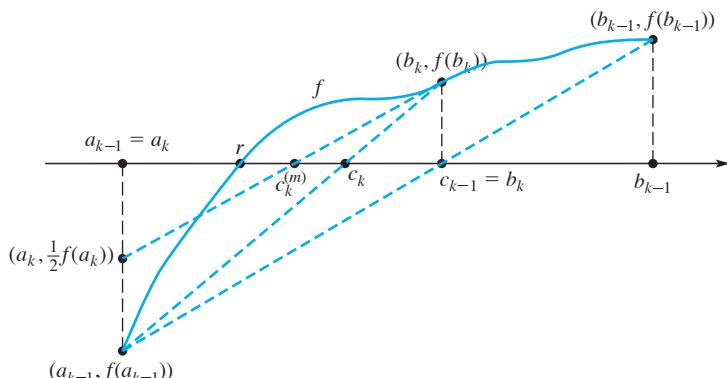


FIGURA 3.3
Método
modificado de
falsa posición

El método de bisección usa sólo el hecho de que $f(a)f(b) < 0$ para cada nuevo intervalo $[a, b]$, pero el método de falsa posición usa los valores de $f(a)$ y $f(b)$. Este es un ejemplo que muestra cómo se puede incluir información adicional en un algoritmo para construir uno mejor. En la sección siguiente, el método de Newton no usa sólo la función sino también su primera derivada.

Algunas variantes del método modificado de falsa posición tienen convergencia superlineal, la cual analizaremos sección 3.3. Consulte, por ejemplo, Ford [1995]. Otro método modificado de falsa posición remplaza las rectas secantes por rectas con cada vez menos pendiente hasta que la iteración cae en el lado opuesto de la raíz. (Véase Conte y De Boor [1980].) Las primeras versiones del método de falsa posición se remontan a un texto matemático chino (200 a.C. a 100 d.C.) y un texto matemático hindú (3 a.C.).

Resumen

(1) Para determinar un cero r de una función continua dada f en un intervalo $[a, b]$, n pasos del método de bisección producen una sucesión de intervalos $[a, b] = [a_0, b_0], [a_1, b_1], [a_2, b_2], \dots, [a_n, b_n]$, cada uno contiene la raíz deseada de la función. Los puntos medios de estos intervalos $c_0, c_1, c_2, \dots, c_n$ forma una sucesión de aproximaciones a la raíz, a saber, $c_i = \frac{1}{2}(a_i + b_i)$. En cada intervalo $[a_i, b_i]$, el error $e_i = r - c_i$ obedece la desigualdad

$$|e_i| \leq \frac{1}{2}(b_i - a_i)$$

y después de n pasos tenemos

$$|e_n| \leq \frac{1}{2^{n+1}}(b_0 - a_0)$$

(2) Para un tolerancia de error ε tal que $|e_n| < \varepsilon$, n pasos son necesarios, donde n satisface la desigualdad

$$n > \frac{\log(b - a) - \log 2\varepsilon}{\log 2}$$

(3) Para el k ésimo paso del método de falsa posición en el intervalo $[a_k, b_k]$, sea

$$c_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$$

Si $f(a_k)f(c_k) > 0$, $a_{k+1} = c_k$ y $b_{k+1} = b_k$; por otro lado, $a_{k+1} = a_k$ y $b_{k+1} = c_k$.

Problemas 3.1

- 1.** Encuentre donde se intersecan las gráficas de $y = 3x$ y $y = e^x$ al encontrar las raíces correctas de $e^x - 3x = 0$ con cuatro dígitos decimales.
- 2.** Dé una demostración gráfica de que la ecuación $\tan x = x$ tiene un número infinito de raíces. Determine una raíz precisamente y otra aproximadamente usando una gráfica. *Sugerencia:* use el método del problema anterior.
- 3.** Demuestre gráficamente que la ecuación $50\pi + \sin x = 100 \arctan x$ tiene un número infinito de soluciones.
- 4.** Por métodos gráficos, localice aproximaciones de todas las raíces de la ecuación no lineal $\ln(x + 1) + \tan(2x) = 0$.
- 5.** Dé un ejemplo de una función para la que el método de bisección *no* converja linealmente.
- 6.** Dibuje una gráfica de una función y discontinua aún así el método de bisección converja. Reita, para obtener una función que diverja.
- 7.** Pruebe la desigualdad (1).

- 8.** Si $a = 0.1$ y $b = 1.0$, ¿cuántos pasos del método de bisección se necesitan para determinar la raíz con un error a lo más de $\frac{1}{2} \times 10^{-8}$?
- 9.** Encuentre todas las raíces de $f(x) = \cos x - \cos 3x$. Use dos métodos diferentes.
- 10.** (Continuación) Encuentre la raíz o raíces de $\ln[(1+x)/(1-x^2)] = 0$.
- 11.** Si f tiene una inversa, entonces la ecuación $f(x) = 0$ se puede resolver al escribir simplemente $x = f^{-1}(0)$. ¿Esta observación elimina el problema de encontrar raíces de ecuaciones? Ejemplifique con $\sin x = 1/\pi$.
- 12.** ¿Cuántos dígitos binarios de precisión se ganan en cada paso del método de bisección? ¿Cuántos pasos se requieren para cada dígito decimal de precisión?
- 13.** Intente idear un criterio de parada del método de bisección para garantizar que la raíz se determina con un *error relativo* a lo más de ε .
- 14.** Denote los intervalos sucesivos que surgen en el método de bisección por $[a_0, b_0]$, $[a_1, b_1]$, $[a_2, b_2]$ y así sucesivamente.
- Muestre que $a_0 \leq a_1 \leq a_2 \leq \dots$ y que $b_0 \geq b_1 \geq b_2 \geq \dots$
 - Muestre que $b_n - a_n = 2^{-n}(b_0 - a_0)$.
 - Muestre que para toda n , $a_n b_n + a_{n-1} b_{n-1} = a_{n-1} b_n + a_n b_{n-1}$.
- 15.** (Continuación) ¿Puede suceder que $a_0 = a_1 = a_2 = \dots$?
- 16.** (Continuación) Sea $c_n = (a_n + b_n)/2$. Muestre que
- $$\lim_{n \rightarrow \infty} c_n = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n$$
- 17.** (Continuación) Considere el método de bisección con el intervalo inicial $[a_0, b_0]$. Muestre que después de diez pasos con este método,
- $$\left| \frac{1}{2}(a_{10} + b_{10}) - \frac{1}{2}(a_9 + b_9) \right| = 2^{-11}(b_0 - a_0)$$
- También determine cuántos pasos se requieren para garantizar una aproximación de una raíz con seis lugares decimales (redondeado).
- 18.** (Verdadero-falso) Si el método de bisección genera intervalos $[a_0, b_0]$, $[a_1, b_1]$ y así sucesivamente, ¿cuál de estas desigualdades son verdaderas para la raíz r que está calculando? Dé pruebas o contraejemplos en cada caso.
- $|r - a_n| \leq 2|r - b_n|$
 - $|r - a_n| \leq 2^{-n-1}(b_0 - a_0)$
 - $|r - \frac{1}{2}(a_n + b_n)| \leq 2^{-n-2}(b_0 - a_0)$
 - $0 \leq r - a_n \leq 2^{-n}(b_0 - a_0)$
 - $|r - b_n| \leq 2^{-n-1}(b_0 - a_0)$
- 19.** (Verdadero-falso) Usando la notación del libro, determine cuáles de estos enunciados son verdaderos y cuáles son generalmente falsos:
- $|r - c_n| < |r - c_{n-1}|$
 - $a_n \leq r \leq c_n$
 - $c_n \leq r \leq b_n$
 - $|r - a_n| \leq 2^{-n}$
 - $|r - b_n| \leq 2^{-n}(b_0 - a_0)$
- 20.** Pruebe que $|c_n - c_{n+1}| = 2^{-n-2}(b_0 - a_0)$.

- 21.** Si se aplica el método de bisección con el intervalo inicial $[a, a + 1]$ y $a \geq 2^m$, donde $m \geq 0$, ¿cuál es el número correcto de pasos para calcular la raíz con precisión total de máquina en una computadora de longitud de palabra de 32 bits?
- 22.** Si se aplica el método de bisección con el intervalo inicial $[2^m, 2^{m+1}]$, donde m es un entero positivo o negativo, ¿cuántos pasos se deben dar para calcular la raíz con la precisión total de máquina en una computadora de longitud de palabra de 32 bits?
- 23.** Todo polinomio de grado n tiene n ceros (contando las multiplicidades) en el plano complejo. ¿Todo polinomio real tiene n ceros reales? ¿Todo polinomio de grado infinito $f(x) = \sum_{n=0}^{\infty} a_n x^n$ tiene un número infinito de ceros?

Problemas de cómputo 3.1

- Usando el método de bisección, determine el punto de intersección de las curvas dadas por $y = x^3 - 2x + 1$ y $y = x^2$.
- Encuentre una raíz de la siguiente ecuación en el intervalo $[0, 1]$ usando el método de bisección: $9x^4 + 18x^3 + 38x^2 - 57x + 14 = 0$.
- Encuentre una raíz de la ecuación $\tan x = x$ en el intervalo $[4, 5]$ usando el método de bisección. ¿Qué sucede en el intervalo $[1, 2]$?
- Encuentre una raíz de la ecuación $6(e^x - x) = 6 + 3x^2 + 2x^3$ entre -1 y $+1$ usando el método de bisección.
- Use el método de bisección para encontrar un cero de la ecuación $\lambda \cosh(50/\lambda) = \lambda + 10$ con que empieza este capítulo.
- Programe el método de bisección como un procedimiento recursivo y pruébelo con uno o dos de los ejemplos del libro.
- Use el método de bisección para determinar raíces de estas funciones en los intervalos indicados. Procese las tres funciones en *una* corrida de computadora.

$$\begin{aligned} f(x) &= x^3 + 3x - 1 && \text{en } [0, 1] \\ g(x) &= x^3 - 2 \sin x && \text{en } [0.5, 2] \\ h(x) &= x + 10 - x \cosh(50/x) && \text{en } [120, 130] \end{aligned}$$

Encuentre cada raíz con precisión total de máquina. Use el número correcto de pasos, al menos aproximadamente. Repita usando el método de falsa posición.

- Pruebe las tres rutinas de bisección en $f(x) = x^3 + 2x^2 + 10x - 20$, con $a = 1$ y $b = 2$. El cero es 1.36880 8108. Al programar esta función polinomial, use multiplicación anidada. Repita usando el método modificado de falsa posición.
- Escriba un programa para encontrar un cero de una función f de la manera siguiente: en cada paso, un intervalo $[a, b]$ está dado y $f(a)f(b) < 0$. Después se calcula c como la raíz de la función lineal que concuerda con f en a y en b . Guardamos ya sea $[a, c]$ o $[c, b]$, dependiendo de si $f(a)f(c) < 0$ o $f(c)f(b) < 0$. Pruebe su programa con varias funciones.

- 10.** Escoja una rutina de su biblioteca de programación para resolver ecuaciones polinomiales y úsela para encontrar las raíces de la ecuación

$$\begin{aligned}x^8 - 36x^7 + 546x^6 - 4536x^5 + 22449x^4 - 67284x^3 \\+ 118124x^2 - 109584x + 40320 = 0\end{aligned}$$

Las raíces correctas son los enteros 1, 2, ..., 8. A continuación, resuelva la misma ecuación cuando el coeficiente de x^7 se cambia a -37. Observe cómo una perturbación menor en los coeficientes puede causar un cambio masivo en las raíces. Por tanto, las raíces son funciones **inestables** de los coeficientes. (Asegúrese de programar el problema permitiendo raíces complejas.) *Nota cultural:* esta es una versión simplificada del **polinomio de Wilkinson**, que se encuentra en el problema de cómputo 3.3.9.

- 11.** Se está usando un **eje circular de metal** para transmitir potencia. Se sabe que a una velocidad angular crítica ω dada, cualquier sacudida durante la rotación ocasiona que el eje se deforme o ceda. Esta es una situación peligrosa, ya que el eje podría romperse bajo el aumento de la fuerza centrífuga. Para encontrar esta velocidad crítica ω , primero debemos calcular un número x que satisface la ecuación

$$\tan x + \tanh x = 0$$

Después se usa este número en una fórmula para obtener ω . Resuelva para x ($x > 0$).

- 12.** Usando rutinas incorporadas en sistemas de software matemático como Matlab, Mathematica o Maple, encuentre las raíces para $f(x) = x^3 - 3x + 1$ en $[0, 1]$ y $g(x) = x^3 - \sin x$ en $[0.5, 2]$ con más dígitos de exactitud que los que se presentan en el libro.
- 13. (Problema de ingeniería)** Las ecuaciones no lineales se presentan en casi todos los campos de la ingeniería. Por ejemplo, suponga que una tarea dada se expresa en la forma $f(x) = 0$ y el objetivo es encontrar valores de x que satisfagan esta condición. Con frecuencia es difícil encontrar una solución explícita y una solución aproximada se busca con la ayuda de software matemático. Encuentre una solución de

$$f(x) = \frac{1}{\sqrt{2\pi}}e^{-(1/2)x^2} + \frac{1}{10} \sin(\pi x)$$

Trace la curva en el rango $[-3.5, 3.5]$ para valores de x y $[-0.5, 0.5]$ para valores $y = f(x)$.

- 14. (Problema de circuitos)** La carga en un circuito simple con una resistencia R , un capacitor C en serie con una batería de voltaje V se presenta por $Q = CV[1 - e^{-T/(RC)}]$, donde Q es la carga del capacitor y T es el tiempo necesario para obtener la carga. Queremos determinar la incógnita C . Por ejemplo, resuelva este problema

$$f(x) = [10x(1 - e^{-0.004/(2000x)}) - 0.00001]$$

Trace la curva. *Sugerencia:* puede ampliar la escala vertical usando $y = 10^5 f(x)$.

- 15. (Polinomios de ingeniería)** Ecuaciones como $A + Bx^2e^{Cx} = 0$ y $A + Bx + Cx^2 + Dx^3 + Ex^4 = 0$ se presentan en problemas de ingeniería. Usando software matemático, encuentre una o más soluciones de las siguientes ecuaciones y trace sus curvas:

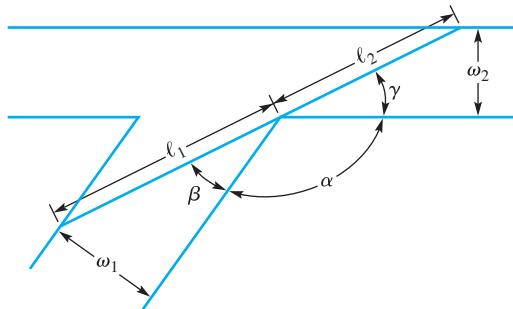
a. $2 - x^2e^{-0.385x} = 0$ **b.** $1 - 32x + 160x^2 - 256x^3 + 128x^4 = 0$

- 16. (Concreto reforzado)** En el diseño de concreto reforzado, cuando se considera tensión, es necesario resolver numéricamente una ecuación cuadrática como

$$24147.07.2x[450 - 0.822x(225)] - 265000000 = 0$$

Encuentre valores aproximados de las raíces.

- 17. (Problema de la tabla en el pasillo)** En una construcción, dos pasillos con anchos $w_1 = 9$ pies y $w_2 = 7$ pies se intersecan y se encuentran a un ángulo $\alpha = 125^\circ$, como se muestra:



Suponiendo una situación bidimensional, ¿cuál es la tabla más larga que puede girar? Ignore el espesor de la tabla. La relación entre los ángulos θ y la longitud de la tabla $\ell = \ell_1 + \ell_2$ es $\ell_1 = w_1 \csc(\beta)$, $\ell_2 = w_2 \csc(\gamma)$, $\beta = \pi - \alpha - \gamma$ y $\ell = w_1 \csc(\pi - \alpha - \gamma) + w_2 \csc(\gamma)$. La longitud máxima de la tabla que puede girar se encuentra al minimizar ℓ como una función de γ . Obteniendo la derivada y haciendo $d\ell/d\gamma = 0$, obtenemos

$$w_1 \cot(\pi - \alpha - \gamma) \csc(\pi - \alpha - \gamma) - w_2 \cot(\gamma) \csc(\gamma) = 0$$

Sustituya los valores conocidos y resuelva numéricamente la ecuación no lineal. Este problema es similar a un ejemplo en Gerald y Wheatley [1999].

- 18.** Encuentre el rectángulo de área máxima si sus vértices están en $(0, 0)$, $(x, 0)$, $(x, \cos x)$, $(0, \cos x)$. Suponga que $0 \leq x \leq \pi/2$.
- 19.** Programe el algoritmo de falsa posición y pruébelo con algunos ejemplos tales como varios de los problemas no lineales del libro o de los problemas de cómputo. Compare sus resultados con los obtenido con el método de bisección.
- 20.** Programe el método modificado de falsa posición, pruébelo y compárello con lo obtenido mediante el método de falsa posición con algunas funciones de ejemplo.

3.2 Método de Newton

El procedimiento que se conoce como método de Newton también se llama **iteración de Newton-Raphson**. Tiene una forma más general que la que aquí veremos, la cual se puede usar una para encontrar raíces de sistemas de ecuaciones. Verdaderamente, este es uno de los procedimientos

más importantes en el análisis numérico y su campo de aplicación se extiende a las ecuaciones diferenciales y ecuaciones integrales. Aquí se aplica a una sola ecuación de la forma $f(x) = 0$. Como antes, buscamos uno o más puntos en los que el valor de la función f es cero.

Interpretaciones del método de Newton

En el método de Newton, se supone desde el principio que la función f es derivable. Esto implica que la gráfica de f tiene una *pendiente* definida en cada punto y , por tanto, una recta tangente única. Ahora permítanos estudiar la siguiente idea simple. En un punto dado $(x_0, f(x_0))$ en la gráfica de f , hay una tangente, que es una bastante buena aproximación a la curva en la vecindad del punto. Analíticamente, esto significa que la función lineal

$$l(x) = f'(x_0)(x - x_0) + f(x_0)$$

está cerca de la función dada f cerca de x_0 . En x_0 , las dos funciones l y f concuerdan. Tomamos el cero de f como una aproximación del cero de l . El cero de l se encuentra fácilmente:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Por esto, empezando con el punto x_0 (al que podemos interpretar como una aproximación a la raíz buscada), pasemos a un nuevo punto x_1 obtenido de la fórmula anterior. Naturalmente, el proceso se puede repetir (iterando) para producir una sucesión de puntos:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}, \quad x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}, \quad \text{etc.}$$

En condiciones favorables, la sucesión de puntos tenderá a un cero de f .

En la figura 3.4 se muestra la geometría del método de Newton. La recta $y = l(x)$ es tangente a la curva $y = f(x)$. Ésta interseca el eje x en un punto x_1 . La pendiente de $l(x)$ es $f'(x_0)$.

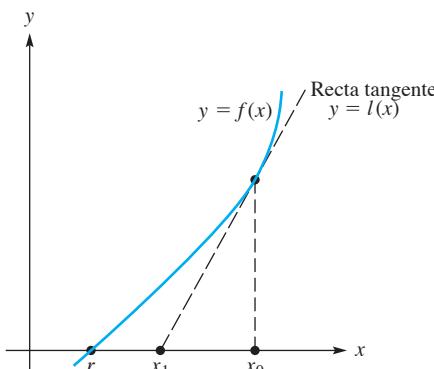


FIGURA 3.4
Método de
Newton

Hay otras maneras de interpretar el método de Newton. Suponga de nuevo que x_0 es una aproximación inicial a una raíz de f . Nos preguntamos: ¿qué corrección h se debe sumar a x_0 para obtener la raíz exactamente? Obviamente, queremos

$$f(x_0 + h) = 0$$

Si f es una función suficientemente bien comportada, se tendrá una serie de Taylor en x_0 [véase la ecuación (11) de la sección 1.2]. Así, podríamos escribir

$$f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \dots = 0$$

Por supuesto, no es fácil determinar h a partir de esta ecuación. Por tanto, nos rendimos a la expectativa de llegar a la raíz verdadera en un solo paso y buscamos sólo una aproximación para h . Ésta se puede obtener al despreciar los dos primeros términos en la serie:

$$f(x_0) + hf'(x_0) = 0$$

La h que resuelve esta *no* es la h que resuelve $f(x_0 + h) = 0$, pero es el número que se calcula más fácilmente

$$h = -\frac{f(x_0)}{f'(x_0)}$$

Nuestra nueva aproximación es entonces

$$x_1 = x_0 + h = x_0 - \frac{f(x_0)}{f'(x_0)}$$

y el proceso se puede repetir. En retrospectiva, vemos que después de todo no era necesaria la serie de Taylor, ya que sólo usamos los primeros dos términos. En el análisis que después faremos, se ha supuesto que f'' es continua en una vecindad de la raíz. Esta suposición nos permite calcular los errores del proceso.

Si el método de Newton se describe en términos de una sucesión x_0, x_1, \dots , entonces se aplica la siguiente definición **recursiva o inductiva**:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Naturalmente, la pregunta interesante es si

$$\lim_{n \rightarrow \infty} x_n = r$$

donde r es la raíz deseada.

EJEMPLO 1 Si $f(x) = x^3 - x + 1$ y $x_0 = 1$, ¿cuáles son x_1 y x_2 en la iteración de Newton?

Solución De la fórmula básica, $x_1 = x_0 - f(x_0)/f'(x_0)$. Ahora $f'(x) = 3x^2 - 1$, y así $f'(1) = 2$. También, encontramos $f(1) = 1$. Por tanto, tenemos $x_1 = 1 - \frac{1}{2} = \frac{1}{2}$. De manera similar, obtenemos $f(\frac{1}{2}) = \frac{5}{8}$, $f'(\frac{1}{2}) = -\frac{1}{4}$ y $x_2 = 3$

Seudocódigo

Un seudocódigo para el método de Newton puede escribirse como sigue:

```

procedure Newton( $f, f'$ ,  $x, nmax, \varepsilon, \delta$ )
integer  $n, nmax;$  real  $x, fx, fp, \varepsilon, \delta$ 
external function  $f, f'$ 
 $fx \leftarrow f(x)$ 
output  $0, x, fx$ 
for  $n = 1$  to  $nmax$  do
     $fp \leftarrow f'(x)$ 
    if  $|fp| < \delta$  then
        output "derivada pequeña"
        return
    end if
     $d \leftarrow fx/fp$ 
     $x \leftarrow x - d$ 
     $fx \leftarrow f(x)$ 
    output  $n, x, fx$ 
    if  $|d| < \varepsilon$  then
        output "convergencia"
        return
    end if
end for
end procedure Newton

```

Usando el valor inicial de x como punto de partida, realizamos un máximo de $nmax$ iteraciones del método de Newton. Los procedimientos se deben proporcionar para las funciones externas $f(x)$ y $f'(x)$. Los parámetros ε y δ se usan para controlar la convergencia y están relacionados con la exactitud deseada o a la precisión de máquina disponible.

Ilustración

Ahora mostraremos el método de Newton para localizar una raíz de $x^3 + x = 2x^2 + 3$. Aplicamos el método a la función $f(x) = x^3 - 2x^2 + x - 3$, iniciando con $x_0 = 3$. Por supuesto, $f'(x) = 3x^2 - 4x + 1$, y estas dos funciones se debe arreglar en forma anidada por eficiencia:

$$\begin{aligned}f(x) &= ((x - 2)x + 1)x - 3 \\f'(x) &= (3x - 4)x + 1\end{aligned}$$

Para ver con mayor detalle la convergencia rápida del método de Newton usamos aritmética con el doble de la precisión normal en el programa y obtenemos los resultados siguientes:

n	x_n	$f(x_n)$
0	3.0	9.0
1	2.4375	2.04
2	2.21303 27224 73144 5	0.256
3	2.17555 49386 14368 4	6.46×10^{-3}
4	2.17456 01006 55071 4	4.48×10^{-6}
5	2.17455 94102 93284 1	1.97×10^{-12}

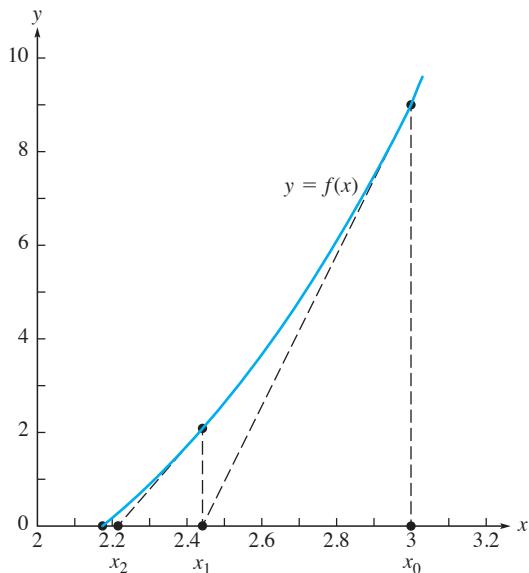


FIGURA 3.5
Tres pasos del
método de
Newton
 $f(x) =$
 $x^3 - 2x^2 + x - 3$

Observe la duplicación de la exactitud en $f(x)$ (y también en x) hasta que se encuentra la máxima precisión de la computadora. La figura 3.5 muestra la gráfica en computadora de tres iteraciones del método de Newton para este problema de ejemplo.

Usando software matemático que permite obtener raíces complejas como Matlab, Maple o Mathematica, encontramos que el polinomio tiene una sola raíz real, 2.17456 y un par de raíces conjugadas complejas, $-0.0872797 \pm 1.17131i$.

Análisis de convergencia

Cualquier persona que ha experimentado con el método de Newton, por ejemplo, al trabajar con algunos de los problemas de esta sección, habrá observado la notable rapidez en la convergencia de la sucesión a la raíz. Este fenómeno es también notable en el ejemplo que acabamos de ver. De hecho, el número de cifras correctas en la respuesta es casi el *doble* en cada paso. Por esto en el ejemplo anterior, tenemos primero 0 y después 1, 2, 3, 6, 12, 24, . . . dígitos exactos en cada iteración de Newton. Cinco o seis pasos del método de Newton con frecuencia bastan para producir precisión total de máquina en la determinación de una raíz. Hay una base teórica para este considerable desempeño, como ahora veremos.

Sea la función f , cuyo cero buscamos y que tiene dos derivadas continuas f' y f'' , y sea r un cero de f . Suponga además que r es un **simple cero**; que es, $f'(r) \neq 0$. Entonces el método de Newton, si inicia suficientemente cerca de r , **converge cuadráticamente** a r . Esto significa que los errores en los pasos sucesivos obedecen una desigualdad de la forma

$$|r - x_{n+1}| \leq c|r - x_n|^2$$

Estableceremos este hecho inmediatamente, pero primero puede ser útil una interpretación informal de la desigualdad.

Suponga, por simplicidad, que $c = 1$. Suponga también que x_n es una estimación de la raíz r que difiere de ésta, a lo más, en una unidad en el k ésimo lugar decimal. Esto significa que

$$|r - x_n| \leq 10^{-k}$$

Las dos desigualdades anteriores implican que

$$|r - x_{n+1}| \leq 10^{-2k}$$

En otras palabras, x_{n+1} difiere de r a lo más en una unidad en el $(2k)$ ésimo lugar decimal. ¡Por lo que x_{n+1} tiene aproximadamente el doble de dígitos correctos que x_n ! Esta es la duplicación de dígitos significativos referida anteriormente.

■ TEOREMA 1

Teorema del método de Newton

Si f , f' y f'' son continuas en la vecindad de una raíz r de f y si $f'(r) \neq 0$, entonces existe una δ positiva con esta propiedad: si el punto inicial en el método de Newton satisface $|r - x_0| \leq \delta$, entonces todos los puntos subsecuentes x_n satisfacen la misma desigualdad, convergen a r y lo hacen cuadráticamente; es decir,

$$|r - x_{n+1}| \leq c(\delta) |r - x_n|^2$$

donde $c(\delta)$ está dada por la ecuación (2) que se presenta a continuación.

Demostración Para establecer la convergencia cuadrática del método de Newton, sea $e_n = r - x_n$. Entonces la fórmula que define la sucesión $\{x_n\}$ da

$$e_{n+1} = r - x_{n+1} = r - x_n + \frac{f(x_n)}{f'(x_n)} = e_n + \frac{f(x_n)}{f'(x_n)} = \frac{e_n f'(x_n) + f(x_n)}{f'(x_n)}$$

Por el teorema de Taylor (véase la sección 1.2), existe un punto ξ_n situado entre x_n y r para el cual

$$0 = f(r) = f(x_n + e_n) = f(x_n) + e_n f'(x_n) + \frac{1}{2} e_n^2 f''(\xi_n)$$

(El subíndice en ξ_n enfatiza la dependencia sobre x_n). Esta última ecuación se puede reacomodar para escribirse

$$e_n f'(x_n) + f(x_n) = -\frac{1}{2} e_n^2 f''(\xi_n)$$

y si ésta se usa en la ecuación anterior para e_{n+1} , el resultado es

$$e_{n+1} = -\frac{1}{2} \left(\frac{f''(\xi_n)}{f'(x_n)} \right) e_n^2 \quad (1)$$

Esta es, al menos cualitativamente, la clase de ecuación que queremos. Continuando con el análisis, definimos una función

$$c(\delta) = \frac{1}{2} \frac{\max_{|x-r| \leq \delta} |f''(x)|}{\min_{|x-r| \leq \delta} |f'(x)|} \quad (\delta > 0) \quad (2)$$

En virtud de esta definición, podemos decir que, para cualesquier dos puntos x y ξ a una distancia δ de la raíz r , la desigualdad $\frac{1}{2} |f''(\xi)| / |f'(x)| \leq c(\delta)$ es válida. Ahora escoja δ tan pequeña que $\delta c(\delta) < 1$. Esto es posible porque conforme δ tiende a 0, $c(\delta)$ converge a $\frac{1}{2} |f''(r)| / |f'(r)|$, y así $\delta c(\delta)$ converge a 0. Recuerde que hemos supuesto que $f'(r) \neq 0$. Sea $\rho = \delta c(\delta)$. En el resto de este argumento, tenemos a δ , $c(\delta)$ y ρ fijas con $\rho < 1$.

Ahora suponga que algunas x_n que iteran están dentro de distancia δ de la raíz r . Tenemos

$$|e_n| = |r - x_n| \leq \delta \quad \text{y} \quad |\xi_n - r| \leq \delta$$

Por la definición de $c(\delta)$, se tiene que $\frac{1}{2}|f'(\xi_n)|/|f'(x_n)| \leq c(\delta)$. De la ecuación (1), ahora tenemos

$$|e_{n+1}| = \frac{1}{2} \left| \frac{f'(\xi_n)}{f'(x_n)} \right| e_n^2 \leq c(\delta) e_n^2 \leq \delta c(\delta) |e_n| = \rho |e_n|$$

Por tanto, x_{n+1} está también dentro de la distancia δ de r ya que

$$|r - x_{n+1}| = |e_{n+1}| \leq \rho |e_n| \leq |e_n| \leq \delta$$

Si el punto inicial x_0 se elige dentro de la distancia δ de r , entonces

$$|e_n| \leq \rho |e_{n-1}| \leq \rho^2 |e_{n-2}| \leq \cdots \leq \rho^n |e_0|$$

Puesto que $0 < \rho < 1$, $\lim_{n \rightarrow \infty} \rho^n = 0$ y $\lim_{n \rightarrow \infty} e_n = 0$. En otras palabras, obtenemos

$$\lim_{n \rightarrow \infty} x_n = r$$

En este proceso, tenemos $|e_{n+1}| \leq c(\delta)e_n^2$. ■

En el uso de Método de Newton, la consideración se debe dar a la adecuada elección de un punto de partida. Con frecuencia, se deben tener algunas pistas en la forma de la gráfica de la función. A veces una gráfica burda es adecuada, pero en otros casos una evaluación paso a paso de la función en diferentes puntos puede ser necesaria para encontrar un punto cerca de la raíz. Con frecuencia se usan inicialmente varios pasos del método de bissección para obtener un adecuado punto de partida y el método de Newton se emplea para mejorar la precisión.

Aunque el método de Newton es verdaderamente un invento maravilloso, su convergencia depende de hipótesis que son difíciles de comprobar a priori. Algunos ejemplos gráficos mostrarán lo que puede suceder. En la figura 3.6(a), la tangente a la gráfica de la función f en x_0 interseca el eje x en un punto distante de la raíz r y puntos sucesivos en la iteración de Newton retroceden

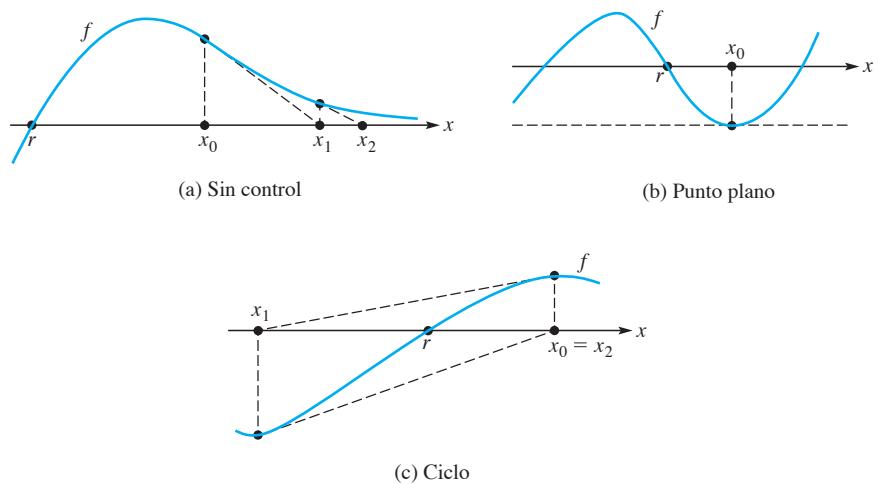


FIGURA 3.6
Falla del
método de
Newton debida
a puntos malos

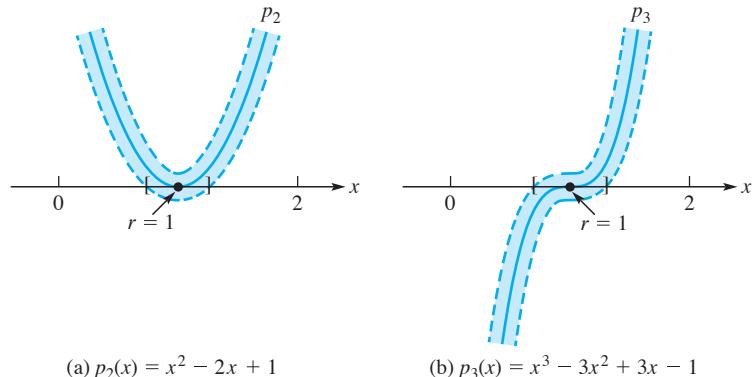
de r en lugar de converger a r . La dificultad se puede atribuir a una pobre elección del punto inicial x_0 ; éste *no* está suficientemente cerca de r . En la figura 3.6(b), la tangente a la curva es paralela al eje x y $x_1 = \pm\infty$, o a éste se le asigna el valor infinito de máquina en una computadora. En la figura 3.6(c) se muestran los valores del *ciclo*, ya que $x_2 = x_0$. En una computadora, los errores de redondeo o de precisión limitada pueden finalmente causar esta situación de desequilibrio tal que las iteraciones son en espiral hacia el interior y convergen o en espiral hacia afuera y divergen.

El análisis que establece la convergencia cuadrática revela otra hipótesis problemática; a saber, $f'(r) \neq 0$. Si $f'(r) = 0$, entonces r es un cero de f y f' . Este cero se llama un **cero múltiple** de f , en este caso, al menos un cero doble. ¡La iteración de Newton para un cero múltiple converge sólo linealmente! Comúnmente, no se sabría antes que el cero buscado era un cero múltiple. Sin embargo, si se sabía que la multiplicidad era m , el método de Newton podría acelerarse al modificar la ecuación

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)}$$

en la que m es la *multiplicidad* del cero en cuestión. La **multiplicidad** del cero r es al menos m tal que $f^{(k)}(r) = 0$ para $0 \leq k < m$, pero $f^{(m)}(r) \neq 0$. (Véase el problema 3.2.35.)

Como se muestra en la figura 3.7, la ecuación $p_2(x) = x^2 - 2x + 1 = 0$ tiene una raíz en 1 de multiplicidad 2 y la ecuación $p_3(x) = x^3 - 3x^2 + 3x - 1 = 0$ tiene una raíz en 1 de multiplicidad 3. Es instructivo trazar estas curvas. Las dos curvas son más bien planas en las raíces, lo que retrasa la convergencia del método regular de Newton. También, las figuras muestran las curvas de dos funciones no lineales con multiplicidades, así como también sus regiones de incertidumbre de las curvas. Por ello, las soluciones calculadas podrían estar en cualquier lugar dentro los intervalos indicados en el eje x . Esta es una indicación de la dificultad para obtener soluciones precisas de funciones no lineales con multiplicidades.



Sistemas de ecuaciones no lineales

Algunos problemas físicos implican la solución de sistemas de N ecuaciones no lineales con N incógnitas. Un método es **linealizar y resolver**, repetidamente. Esta es la misma estrategia que usa el método de Newton para resolver una sola ecuación no lineal. No es de extrañar que se pueda encontrar una extensión natural del método de Newton para sistemas no lineales. El tema de sistemas de ecuaciones no lineales requiere cierta familiaridad con matrices y sus inversas (véase el apéndice D).

En el caso general, un sistema de N ecuaciones no lineales con N incógnitas x_i se puede presentar en la forma

$$\begin{cases} f_1(x_1, x_2, \dots, x_N) = 0 \\ f_2(x_1, x_2, \dots, x_N) = 0 \\ \vdots \\ f_N(x_1, x_2, \dots, x_N) = 0 \end{cases}$$

Usando notación vectorial podemos escribir este sistema en una forma más elegante:

$$\mathbf{F}(\mathbf{X}) = \mathbf{0}$$

definiendo vectores columna como

$$\begin{aligned} \mathbf{F} &= [f_1, f_2, \dots, f_N]^T \\ \mathbf{X} &= [x_1, x_2, \dots, x_N]^T \end{aligned}$$

La extensión del método de Newton para sistemas no lineales es

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - [\mathbf{F}'(\mathbf{X}^{(k)})]^{-1} \mathbf{F}(\mathbf{X}^{(k)})$$

donde $\mathbf{F}'(\mathbf{X}^{(k)})$ es la **matriz jacobiana**, que se definirá inmediatamente. Tiene derivadas parciales de \mathbf{F} evaluadas en $\mathbf{X}^{(k)} = [x_1^{(k)}, x_2^{(k)}, \dots, x_N^{(k)}]^T$. Esta fórmula es similar a la versión anterior del método de Newton excepto que la expresión de la derivada no está en el denominador sino en el numerador como la inversa de una matriz. La forma computacional de la fórmula $\mathbf{X}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \dots, x_N^{(0)}]^T$ es una aproximación inicial del vector, tomada cerca de la solución del sistema no lineal y la inversa de la matriz jacobiana no se calcula sino más bien se resuelve un sistema de ecuaciones.

Ejemplificamos el desarrollo de este procedimiento usando tres ecuaciones no lineales

$$\begin{cases} f_1(x_1, x_2, x_3) = 0 \\ f_2(x_1, x_2, x_3) = 0 \\ f_3(x_1, x_2, x_3) = 0 \end{cases} \quad (3)$$

Recuerde el desarrollo de Taylor de tres variables para $i = 1, 2, 3$:

$$f_i(x_1 + h_1, x_2 + h_2, x_3 + h_3) = f_i(x_1, x_2, x_3) + h_1 \frac{\partial f_i}{\partial x_1} + h_2 \frac{\partial f_i}{\partial x_2} + h_3 \frac{\partial f_i}{\partial x_3} + \dots \quad (4)$$

donde las derivadas parciales se evalúan en el punto (x_1, x_2, x_3) . Aquí sólo se muestran los términos lineales con tamaños de paso h_i . Suponga que el vector $\mathbf{X}^{(0)} = (x_1^{(0)}, x_2^{(0)}, x_3^{(0)})^T$ es una solución aproximada de (3). Sea $\mathbf{H} = [h_1, h_2, h_3]^T$ una corrección calculada a la conjectura inicial de modo que $\mathbf{X}^{(0)} + \mathbf{H} = [x_1^{(0)} + h_1, x_2^{(0)} + h_2, x_3^{(0)} + h_3]^T$ es una solución aproximada mejor. Descartando los términos de orden superior en el desarrollo de Taylor (4), tenemos en notación vectorial

$$\mathbf{0} \approx \mathbf{F}(\mathbf{X}^{(0)} + \mathbf{H}) \approx \mathbf{F}(\mathbf{X}^{(0)}) + \mathbf{F}'(\mathbf{X}^{(0)})\mathbf{H} \quad (5)$$

donde la **matriz jacobiana** está definida por

$$\mathbf{F}'(\mathbf{X}^{(0)}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}$$

Aquí todas las derivadas parciales se evalúan en $\mathbf{X}^{(0)}$; a saber,

$$\frac{\partial f_i}{\partial x_j} = \frac{\partial f_i(\mathbf{X}^{(0)})}{\partial x_j}$$

También, suponemos que la matriz jacobiana $\mathbf{F}'(\mathbf{X}^{(0)})$ es no singular, por lo que su inversa existe. Resolviendo para \mathbf{H} en (5) tenemos

$$\mathbf{H} \approx -[\mathbf{F}'(\mathbf{X}^{(0)})]^{-1} \mathbf{F}(\mathbf{X}^{(0)})$$

Sea $\mathbf{X}^{(1)} = \mathbf{X}^{(0)} + \mathbf{H}$ la mejor aproximación después de la corrección; entonces llegamos a la primera iteración del método de Newton para sistemas no lineales

$$\mathbf{X}^{(1)} = \mathbf{X}^{(0)} - [\mathbf{F}'(\mathbf{X}^{(0)})]^{-1} \mathbf{F}(\mathbf{X}^{(0)})$$

En general, el método de Newton usa esta iteración:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - [\mathbf{F}'(\mathbf{X}^{(k)})]^{-1} \mathbf{F}(\mathbf{X}^{(k)})$$

En la práctica, la forma computacional del método de Newton no implica invertir la matriz jacobiana sino más bien resolver los **sistemas lineales jacobianos**

$$[\mathbf{F}'(\mathbf{X}^{(k)})] \mathbf{H}^{(k)} = -\mathbf{F}(\mathbf{X}^{(k)}) \quad (6)$$

La siguiente iteración del método de Newton es entonces

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{H}^{(k)} \quad (7)$$

Este es un **método de Newton para sistemas no lineales**. El sistema lineal (6) se puede resolver con los procedimientos *Gauss_Simple* que se analizan en el capítulo 7. Sistemas pequeños de orden 2 se pueden resolver fácilmente (véase el problema 3.2.39).

EJEMPLO 2 Como un ejemplo, podemos escribir un seudocódigo para resolver el siguiente sistema de ecuaciones no lineales usando una variante del método de Newton dado por (6) y (7):

$$\begin{cases} x + y + z = 3 \\ x^2 + y^2 + z^2 = 5 \\ e^x + xy - xz = 1 \end{cases} \quad (8)$$

Solución Con ojo avizor, usted inmediatamente ve que la solución de este sistema es $x = 0, y = 1, z = -2$. Pero en problemas más reales, la solución no es así de obvia. Queremos desarrollar un procedimiento

numérico para encontrar una solución. Aquí se muestra un seudocódigo:

```

 $\mathbf{X} = [0.1, 1.2, 2.5]^T$ 
for  $k = 1$  to 10 do
     $\mathbf{F} = \begin{bmatrix} x_1 + x_2 + x_3 - 3 \\ x_1^2 + x_2^2 + x_3^2 - 5 \\ e^{x_1} + x_1 x_2 - x_1 x_3 - 1 \end{bmatrix}$ 
     $\mathbf{J} = \begin{bmatrix} 1 & 1 & 1 \\ 2x_1 & 2x_2 & 2x_3 \\ e^{x_1} + x_2 - x_3 & x_1 & -x_1 \end{bmatrix}$ 
    solve  $\mathbf{JH} = \mathbf{F}$ 
     $\mathbf{X} = \mathbf{X} - \mathbf{H}$ 
end for

```

Cuando se programa y se ejecuta en una computadora, vemos que éste converge a $\mathbf{x} = (0, 1, 2)$, pero cuando cambiamos a un vector de inicio diferente, $(1, 0, 1)$, éste converge a otra raíz, $(1.2244, -0.0931, 1.8687)$. (¿Por qué?) ■

Podemos usar software matemático como Matlab, Maple o Mathematica y sus procedimientos incorporados para resolver el sistema de ecuaciones no lineales (8). El área importante de aplicación de resolver sistemas de ecuaciones no lineales se usa en el capítulo 16 en la minimización de funciones.

Cuencas de atracción de fractales

El campo de aplicación del método de Newton para determinar raíces complejas es uno de sus puntos sobresalientes. Sólo se necesita programarlo usando aritmética compleja.

Las fronteras del análisis numérico y de la dinámica no lineal se traslanan en algunas maneras intrigantes. La presentación con patrones de fractales generados por computadora, como los que se muestran en la figura 3.8, se pueden crear fácilmente con la ayuda de la iteración de Newton. Las imágenes resultantes muestran conjuntos elaboradamente entrelazados en el plano que

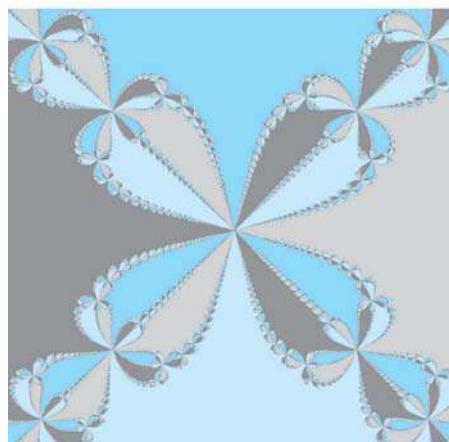


FIGURA 3.8
Cuencas de
atracción

son muy hermosos si se presentan en un monitor de computadora. Se inicia con un polinomio de variable compleja z . Por ejemplo, $p(z) = z^4 - 1$ es adecuado. Este polinomio tiene cuatro ceros, que son las cuatro raíces de la unidad. Cada uno de estos ceros tiene una **cuenca de atracción**, que es el conjunto de todos los puntos z_0 tales que la iteración de Newton, que inicia en z_0 , convergerá a cero. Estas cuatro cuencas de atracción están separadas de las otras, ya que si la iteración de Newton inicia en z_0 converge a ese cero y entonces no puede también converger a otro. Se podría naturalmente esperar que cada cuenca sea un conjunto simple rodeando al cero en el plano complejo. Pero esto está lejos de ser simple. Para ver qué son, podemos determinar sistemáticamente, para un gran número de puntos, que el cero de la iteración p de Newton converge a éste si inicia en z_0 . A los puntos de cada cuenca se les pueden asignar diferentes colores. Los (escasos) puntos para los que la iteración de Newton no converge se pueden dejar sin color. El problema de cómputo 3.2.27 sugiere cómo hacer esto.

Resumen

- (1) Para determinar un cero de una función continua y derivable función f , el **método de Newton** está dado por

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (n \geq 0)$$

Este requiere un valor inicial dado x_0 y dos evaluaciones de la función (para f y f') por paso.

- (2) Los errores están relacionados por

$$e_{n+1} = -\frac{1}{2} \left(\frac{f'(\xi_n)}{f'(x_n)} \right) e_n^2$$

lo que conduce a la desigualdad

$$|e_{n+1}| \leq c |e_n|^2$$

Esto significa que el método de Newton tiene comportamiento de **convergencia cuadrática** para x_0 suficientemente cercana a la raíz r .

- (3) Para un sistema $N \times N$ de ecuaciones no lineales $\mathbf{F}(\mathbf{X}) = \mathbf{0}$, el **método de Newton** se escribe como

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - [\mathbf{F}'(\mathbf{X}^{(k)})]^{-1} \mathbf{F}(\mathbf{X}^{(k)}) \quad (k \geq 0)$$

que implica la matriz jacobiana $\mathbf{F}'(\mathbf{X}^{(k)}) = \mathbf{J} = [(\partial f_i(\mathbf{X}^{(k)}) / \partial x_j)]_{N \times N}$. En la práctica, se resuelve el **sistema lineal jacobiano**

$$[\mathbf{F}'(\mathbf{X}^{(k)})] \mathbf{H}^{(k)} = -\mathbf{F}(\mathbf{X}^{(k)})$$

usando eliminación gaussiana y después se encuentra la siguiente iteración de la ecuación

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{H}^{(k)}$$

Referencias adicionales

Para detalles adicionales y gráficas de ejemplo, véase Kincaid y Cheney [2002] o Eppreanu y Greenside [1998]. Para otras referencias de fractales, véase Crilly, Earnshaw y Jones [1991], Feder [1998], Hastings y Sugihara [1993], y Novak [1998].

Además, un artículo de Ypma [1995] revisa el desarrollo histórico del método de Newton a través de notas, cartas y publicaciones de Isaac Newton, Joseph Raphson y Thomas Simpson.

Problemas 3.2

- 1.** Compruebe que cuando se usa el método de Newton para calcular \sqrt{R} (al resolver la ecuación $x^2 = R$), la sucesión de iteraciones está definida por

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{R}{x_n} \right)$$

- 2.** (Continuación) Muestre que si la sucesión $\{x_n\}$ está definida como en el problema anterior, entonces

$$x_{n+1}^2 - R = \left[\frac{x_n^2 - R}{2x_n} \right]^2$$

Interprete esta ecuación en términos de convergencia cuadrática.

- 3.** Escriba el método de Newton en forma simplificada para determinar el recíproco de la raíz cuadrada de un número positivo. Desarrolle dos iteraciones para aproximar $1/\pm\sqrt{5}$, iniciando con $x_0 = 1$ y $x_0 = -1$.
- 4.** Dos de los cuatro ceros de $x^4 + 2x^3 - 1x^2 + 3$ son positivos. Encuéntrelos con el método de Newton, correctos con dos números significativos.
- 5.** La ecuación $x - Rx^{-1} = 0$ tiene $x = \pm R^{1/2}$ como su solución. Establezca el esquema iterativo de Newton, en forma simplificada, para esta situación. Realice cinco pasos para $R = 25$ y $x_0 = -1$.
- 6.** Usando una calculadora, observe la lentitud con la que converge el método de Newton en el caso de $f(x) = (x - l)^m$ con $m = 8$ o 12 . Reconcilie esto con la teoría. Use $x_0 = 1.1$.
- 7.** ¿Qué función lineal $y = ax + b$ aproxima a $f(x) = \sin x$ mejor en la vecindad de $x = \pi/4$? ¿Cómo se relaciona este problema con el método de Newton?
- 8.** En los problemas 1.2.11 y 1.2.12, se sugieren varios métodos para calcular $\ln 2$. Compárelos con el uso del método de Newton aplicado a la ecuación $e^x = 2$.
- 9.** Defina una sucesión $x_{n+1} = x_n - \tan x_n$, con $x_0 = 3$. ¿A qué es igual el $\lim_{n \rightarrow \infty} x_n$?
- 10.** La fórmula de iteración $x_{n+1} = x_n - (\cos x_n)(\operatorname{sen} x_n) + R \cos^2 x_n$, donde R es una constante positiva, fue obtenida al aplicar el método de Newton a alguna función $f(x)$. ¿Qué era $f(x)$? ¿En qué se puede usar esta fórmula?
- 11.** Establezca el esquema iterativo de Newton en forma simplificada, sin implicar el recíproco de x , para la función $f(x) = xR - x^{-1}$. Realice tres pasos de este procedimiento usando $R = 4$ y $x_0 = -1$.

- 12.** Considere los procedimientos siguientes:

$$\text{a. } x_{n+1} = \frac{1}{3} \left(2x_n - \frac{r}{x_n^2} \right) \quad \text{b. } x_{n+1} = \frac{1}{2}x_n + \frac{1}{x_n}$$

¿Convergerán para cualquier punto inicial diferente de cero? Si es así, ¿a qué valores?

- 13.** Cada una de las funciones siguientes tiene a $\sqrt[3]{R}$ como un cero para cualquier número positivo real R . Determine las fórmulas para el método de Newton para cada una y cualesquier restricción necesaria en la elección de x_0 .

$$\begin{array}{lll} \text{a. } a(x) = x^3 - R & \text{b. } b(x) = 1/x^3 - 1/R & \text{c. } c(x) = x^2 - R/x \\ \text{d. } d(x) = x - R/x^2 & \text{e. } e(x) = 1 - R/x^3 & \text{f. } f(x) = 1/x - x^2/R \\ \text{g. } g(x) = 1/x^2 - x/R & \text{h. } h(x) = 1 - x^3/R & \end{array}$$

- 14.** Determine las fórmulas para el método de Newton para determinar una raíz de la función $f(x) = x - e/x$. ¿Cuál es la conducta de las iteraciones?

- 15.** Si el método de Newton se usa en $f(x) = x^3 - x + 1$ iniciando con $x_0 = 1$, ¿a qué será igual x_1 ?

- 16.** Localice la raíz de $f(x) = e^{-x} - \cos x$ que está cerca de $\pi/2$.

- 17.** Si se usa el método de Newton en $f(x) = x^5 - x^3 + 3$ y si $x_n = 1$, ¿a qué es igual x_{n+1} ?

- 18.** Determine la fórmula de la iteración de Newton para calcular la raíz cúbica de N/M para enteros diferentes de cero N y M .

- 19.** ¿Para qué valores iniciales el método de Newton converge si la función f es $f(x) = x^2/(1+x^2)$?

- 20.** Iniciando en $x = 3$, $x < 3$ o $x > 3$, analice qué pasa cuando el método de Newton se aplica a la función $f(x) = 2x^3 - 9x^2 + 12x + 15$.

- 21.** (Continuación) Repita para $f(x) = \sqrt{|x|}$, iniciando con $x < 0$ o con $x > 0$.

- 22.** Para determinar $x = \sqrt[3]{R}$, podemos resolver la ecuación $x^3 = R$ con el método de Newton. Escriba el ciclo que realiza este proceso, comenzando con la aproximación inicial $x_0 = R$.

- 23.** El **recíproco** de un número R se puede calcular sin división con la fórmula iterativa

$$x_{n+1} = x_n(2 - x_n R)$$

Establezca esta relación al aplicar el método de Newton a algunas $f(x)$. Inicie con $x_0 = 0.2$, calcule el recíproco de 4 correcto con seis dígitos decimales o más usando esta regla. Tabule el error en cada paso y observe la convergencia cuadrática.

- 24.** En una computadora moderna, los números de punto flotante tienen una mantisa de 48 bits. Además, el hardware de punto flotante puede realizar suma, resta, multiplicación y reciprocación, pero no división. Desafortunadamente, el hardware de reciprocación produce un resultado con menor exactitud que con precisión total, mientras que las otras operaciones producen resultados exactos con precisión total de punto flotante.

- a.** Muestre que el método de Newton se puede aplicar para encontrar un cero de la función $f(x) = 1 - 1/(ax)$. Esta proporcionará una aproximación a $1/a$ que es exacta con precisión total de punto flotante. ¿Cuántas iteraciones se requieren?

- b. Muestre cómo obtener una aproximación a b/a que sea exacta con precisión total de punto flotante.

25. El método de Newton para determinar \sqrt{R} es

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{R}{x_n} \right)$$

Desarrolle tres iteraciones de este esquema para calcular \sqrt{R} , iniciando con $x_0 = 1$ y del método de bisección para $\sqrt{2}$, empezando con el intervalo $[1, 2]$. ¿Cuántas iteraciones se necesitan para cada método con el fin de obtener una exactitud de 10^{-6} ?

26. (Continuación) El método de Newton para determinar \sqrt{R} , donde $R = AB$, da esta aproximación:

$$\sqrt{AB} \approx \frac{A+B}{4} + \frac{AB}{A+B}$$

Muestre que si $x_0 = A$ o B , entonces se necesitan dos iteraciones del método de Newton para obtener esta aproximación, mientras que si $x_0 = \frac{1}{2}(A+B)$, entonces sólo es necesaria una iteración.

27. Muestre que el método de Newton aplicado a $x^m - R$ y a $1 - (R/x^m)$ para determinar $\sqrt[m]{R}$ da como resultado dos similares pero todavía diferentes fórmulas iterativas. Aquí $R > 0$, $m \geq 2$. ¿Qué fórmula es mejor y por qué?

28. Usando una calculadora manual, realice tres iteraciones del método de Newton usando $x_0 = 1$ y $f(x) = 3x^3 + x^2 - 15x + 3$.

29. ¿Qué pasa si la iteración de Newton se aplica a $f(x) = \arctan x$ con $x_0 = 2$? Para qué valores iniciales convergerá el método de Newton? (Véase el problema de cómputo 3.2.7.)

30. El método de Newton se puede interpretar como sigue: suponga que $f(x+h) = 0$. Entonces $f'(x) \approx [f(x+h) - f(x)]/h = -f(x)/h$. Continúe este argumento.

31. Deduzca una fórmula para el método de Newton para la función $F(x) = f(x)/f'(x)$, donde $f(x)$ es una función con ceros simples que es tres veces continuamente derivable. Muestre que la convergencia del método resultante para cualquier cero r de $f(x)$ es al menos cuadrática. *Sugerencia:* aplique el resultado del libro a F , asegurándose que F tiene las propiedades requeridas.

32. La serie de Taylor para una función f se parece a esta:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \dots$$

Suponga que $f(x)$, $f'(x)$, y $f''(x)$ son calculadas fácilmente. Deduzca un algoritmo parecido al método de Newton que use tres términos de la serie de Taylor. El algoritmo debe tomar como entrada una aproximación a la raíz y producir como salida una mejor aproximación a ésta. Muestre que el método es cúbicamente convergente.

33. Para evitar calcular la derivada en cada paso en el método de Newton, se ha propuesto remplazar $f'(x_n)$ con $f'(x_0)$. Deduzca la rapidez de convergencia para este método.

34. Consulte al análisis del método de Newton y establezca que

$$\lim_{n \rightarrow \infty} (e_{n+1} e_n^{-2}) = -\frac{1}{2} \left[\frac{f''(r)}{f'(r)} \right]$$

¿Cómo se puede usar esto en un caso práctico para probar si la convergencia es cuadrática? Diseñe un ejemplo en el que $r, f'(r)$ y $f''(r)$ son todas conocidas y pruebe numéricamente la convergencia de $e_{n+1} e_n^{-2}$.

35. Muestre que en el caso de un cero de multiplicidad m , el **método modificado de Newton**

$$x_{n+1} = x_n - m \frac{f(x_n)}{f'(x_n)}$$

es cuadráticamente convergente. *Sugerencia:* use series de Taylor para $f(r + e_n)$ y $f'(r + e_n)$.

36. El **método de Steffensen** para resolver la ecuación $f(x) = 0$ usa la fórmula

$$x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)}$$

en el que $g(x) = \{f[x + f(x)] - f(x)\}/f(x)$. Es cuadráticamente convergente, como en el método de Newton. ¿Cuántas evaluaciones de función se necesitan por paso? Usando series de Taylor, muestre que $g(x) \approx f'(x)$ si $f(x)$ es pequeña y así relacione la iteración de Steffensen con la de Newton. ¿Qué ventaja tiene la de Steffensen? Establezca la convergencia cuadrática.

37. Una **generalización del método de Newton** propuesta es

$$x_{n+1} = x_n - \omega \frac{f(x_n)}{f'(x_n)}$$

donde la constante ω es un factor de aceleración elegido para aumentar la tasa de convergencia. ¿Para qué rango de valores de ω es una simple raíz r de $f(x)$ un **punto de atracción**; es decir, $|\omega f'(r)| < 1$, donde $g(x) = x - \omega f(x)/f'(x)$? Este método es cuadráticamente convergente sólo si $\omega = 1$, ya que $g'(r) \neq 0$ cuando $\omega \neq 1$.

38. Suponga que r es una raíz doble de $f(x) = 0$; es decir, $f(r) = f'(r) = 0$ pero $f''(r) \neq 0$, y suponga que f y todas sus derivadas hasta la segunda son continuas en alguna vecindad de r . Muestre que $e_{n+1} \approx e_n$ para el método de Newton y en consecuencia concluya que la tasa de convergencia es *lineal* cerca de una raíz doble. (Si la raíz tiene multiplicidad m , entonces $e_{n+1} \approx [(m-1)/m]e_n$).

39. (**Ecuaciones simultáneas no lineales**) Usando la serie de Taylor en dos variables (x, y) de la forma

$$f(x + h, y + k) = f(x, y) + hf_x(x, y) + kf_y(x, y) + \dots$$

donde $f_x = \partial f / \partial x$ y $f_y = \partial f / \partial y$, establezca que el método de Newton para resolver las dos ecuaciones simultáneas no lineales

$$\begin{cases} f(x, y) = 0 \\ g(x, y) = 0 \end{cases}$$

se puede describir con las fórmulas

$$x_{n+1} = x_n - \frac{fg_y - gf_y}{f_xg_y - g_xf_y}, \quad y_{n+1} = y_n - \frac{f_xg - g_xf}{f_xg_y - g_xf_y}$$

Aquí las funciones f, f_x y así sucesivamente son evaluadas en (x_n, y_n) .

- 40.** El método de Newton se puede definir para la ecuación $f(z) = g(x, y) + ih(x, y)$, donde $f(z)$ es una función analítica de la variable compleja $z = x + iy$ (x y y reales) y $g(x, y)$ y $h(x, y)$ son funciones reales para todas x y y . La derivada $f'(z)$ está dada por $f'(z) = g_x + ih_x = h_y - ig_y$ porque las **ecuaciones de Cauchy-Riemann** $g_x = h_y$ y $h_x = -g_y$ son válidas. Aquí las derivadas parciales están definidas como $g_x = \partial g / \partial x$, $g_y = \partial g / \partial y$ y así sucesivamente. Muestre que el método de Newton

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}$$

puede escribirse en la forma

$$x_{n+1} = x_n - \frac{gh_y - hg_y}{g_x h_y - g_y h_x}, \quad y_{n+1} = y_n - \frac{hg_x - gh_x}{g_x h_y - g_y h_x}$$

Aquí todas funciones están evaluadas en $z_n = x_n + iy_n$.

- 41.** Considere el algoritmo en el que el paso *uno* consta de dos pasos del método de Newton. ¿Cuál es su orden de convergencia?
- 42.** (Continuación) Usando la idea del problema anterior, muestre cómo podemos crear fácilmente métodos de orden superior arbitrario para resolver $f(x) = 0$. ¿Por qué el orden de un método no es el único criterio que se debe considerar al evaluar sus virtudes?
- 43.** Si queremos resolver la ecuación $2 - x = e^x$ usando la iteración de Newton, ¿cuáles son las ecuaciones y funciones que se deben codificar? Proponga un pseudocódigo para resolver este problema. Incluya un punto inicial conveniente y un criterio conveniente de parada.
- 44.** Suponga que queremos calcular $\sqrt{2}$ usando el método de Newton en la ecuación $x^2 = 2$ (en la forma obvia, directa). Si el punto inicial es $x_0 = \frac{7}{3}$, ¿cuál es el valor numérico de la corrección que se debe agregar a x_0 para obtener x_1 ? *Sugerencia:* la aritmética es muy fácil si usa cocientes de enteros.
- 45.** Aplique el método de Newton a la ecuación $f(x) = 0$ con $f(x)$ dada como se muestra a continuación. Encuentre qué pasa y por qué.

a. $f(x) = e^x$ b. $f(x) = e^x + x^2$

- 46.** Considere el método de Newton $x_{n+1} = x_n - f(x_n)/f'(x_n)$. Si la sucesión converge, entonces el punto límite es una solución. Explique por qué sí o por qué no.

Problemas de cómputo 3.2

- Usando el procedimiento de *Newton* y una sola corrida de computadora, pruebe su código con estos ejemplos: $f(t) = \tan t - t$ con $x_0 = 7$ y $e^t - \sqrt{t+9}$ con $x_0 = 2$. Imprima cada iteración y su correspondiente valor de la función.
- Escriba un sencillo programa autocontenido para aplicar el método de Newton a la ecuación $x^3 + 2x^2 + 10x = 20$, iniciando con $x_0 = 2$. Evalúe las adecuadas $f(x)$ y $f'(x)$ usando multiplicación anidada. Detenga el cálculo cuando dos puntos sucesivos difieran por $\frac{1}{2} \times 10^{-5}$ o algunas otras tolerancias convenientes cercanas a la capacidad de su máquina. Imprima todos los puntos intermedios y valores de la función. Ponga un límite superior de diez en el número de pasos.

3. (Continuación) Repita usando doble precisión y más pasos.

^a4. Encuentre la raíz de la ecuación

$$2x(1 - x^2 + x) \ln x = x^2 - 1$$

en el intervalo $[0, 1]$ con el método de Newton usando doble precisión. Haga una tabla que muestre el número de dígitos correctos en cada paso.

- ^a5. En 1685, John Wallis publicó un libro llamado *Algebra*, en el que describió un método diseñado por Newton para resolver ecuaciones. En forma ligeramente modificada, este método también lo publicó Joseph Raphson en 1690. Esta forma es la que ahora comúnmente se llama *método de Newton o método de Newton-Raphson*. Newton mismo analizó el método en 1669 y lo ilustró con la ecuación $x^3 - 2x - 5 = 0$. Wallis usó el mismo ejemplo. Encuentre una raíz de esta ecuación con doble precisión y así preserva la tradición de que cada estudiante de análisis numérico debía resolver esta venerable ecuación.
6. En mecánica celeste, la **ecuación de Kepler** es importante. Ésta se escribe como $x = y - \varepsilon \operatorname{sen} y$, en el que x la anomalía media de un planeta y su excentricidad anómala y ε la excentricidad de su órbita. Tomando $\varepsilon = 0.9$, construya una tabla de y para 30 valores igualmente espaciados de x en el intervalo $0 \leq x \leq \pi$. Use el método de Newton para obtener cada valor de y . La y correspondiente a una x se puede usar como el punto de partida para la iteración cuando x cambia ligeramente.
7. En el método de Newton, avanzamos en cada paso de un punto dado x a un nuevo punto $x - h$, donde $h = f(x)/f'(x)$. Un refinamiento que es fácilmente programado es este: si $|f(x - h)|$ no es más pequeño que $|f(x)|$, entonces rechace este valor de h y use en su lugar $h/2$. Pruebe este refinamiento.
8. Escriba un programa pequeño para calcular una raíz de la ecuación $x^3 = x^2 + x + 1$, usando el método de Newton. Tenga cuidado al elegir un valor inicial conveniente.
9. Encuentre la raíz de la ecuación $5(3x^4 - 6x^2 + 1) = 2(3x^5 - 5x^3)$ que está en el intervalo $[0, 1]$ usando el método de Newton y un programa pequeño.
10. Para cada ecuación, escriba un programa pequeño para calcular e imprimir ocho pasos del método de Newton para determinar una raíz positiva.
- ^aa. $x = 2 \operatorname{sen} x$ ^ab. $x^3 = \operatorname{sen} x + 7$ ^ac. $\operatorname{sen} x = 1 - x$
^ad. $x^5 + x^2 = 1 + 7x^3$ para $x \geq 2$
11. Escriba y pruebe a procedimiento recursivo para el método de Newton.
12. Reescriba y pruebe el procedimiento *Newton* para que sea una función de carácter y regrese palabras clave tales como *iteración*, *éxito*, *cerca de cero*, *máx-iteración*. Entonces se puede usar un enunciado *case* para imprimir los resultados.
13. ¿Le gustaría ver al número 0.55887 766 como resultado de un cálculo? Tome tres pasos en el método de Newton en $10 + x^3 - 12 \cos x = 0$ iniciando con $x_0 = 1$.
- ^a14. Escriba un pequeño programa para encontrar una raíz de la ecuación $e^{-x^2} = \cos x + 1$ en $[0, 4]$. ¿Qué pasa en el método de Newton si iniciamos con $x_0 = 0$ o con $x_0 = 1$?
15. Encuentre la raíz de la ecuación $\frac{1}{2}x^2 + x + 1 - e^x = 0$ usando el método de Newton, iniciando $x_0 = 1$ y considerando convergencia lenta.

- 16.** Usando $f(x) = x^5 - 9x^4 - x^3 + 17x^2 - 8x - 8$ y $x_0 = 0$, estudie y explique el comportamiento del método de Newton. *Sugerencia:* las iteraciones son inicialmente cíclicas.
- 17.** Encuentre el cero de la función $f(x) = x - \tan x$ que está más cerca de 99 (radianes) por el método de bisección y por el método de Newton. *Sugerencia:* se necesitan valores iniciales extremadamente exactos para esta función. Use la computadora para construir una tabla de valores de $f(x)$ alrededor de 99 para determinar la naturaleza de esta función.
- 18.** Usando el método de bisección, encuentre la raíz positiva de $2x(1+x^2)^{-1} = \arctan x$. Usando la raíz como x_0 , aplique el método de Newton para la función $\arctan x$. Interprete los resultados.
- 19.** Si la raíz de $f(x) = 0$ es una raíz doble, entonces el método de Newton se puede acelerar usando

$$x_{n+1} = x_n - 2 \frac{f(x_n)}{f'(x_n)}$$

Compare numéricamente la convergencia de este esquema con el método de Newton en una función con una raíz doble conocida.

- 20.** Programe y pruebe el método de Steffensen, como se describe en el problema 3.2.36.

- 21.** Considere el sistema no lineal

$$\begin{cases} f(x, y) = x^2 + y^2 - 25 = 0 \\ g(x, y) = x^2 - y - 2 = 0 \end{cases}$$

Usando software con capacidades de trazo de gráficas bidimensionales, muestre lo que pasa al resolver dicho sistema al trazar las gráficas de $f(x, y)$, $g(x, y)$ y muestre su intersección con plano (x, y) . Determine las raíces aproximadas de estas ecuaciones a partir de los resultados gráficos.

- 22.** Resuelva este par de ecuaciones simultáneas no lineales pero primero elimine y y después resuelva la ecuación resultante en x con el método de Newton. Inicie con el valor inicial $x_0 = 1.0$.

$$\begin{cases} x^3 - 2xy + y^7 - 4x^3y = 5 \\ y \sin x + 3x^2y + \tan x = 4 \end{cases}$$

- 23.** Usando las ecuaciones (7) y (8), codifique el método de Newton para sistemas no lineales. Pruebe su programa al resolver uno o más de los siguientes sistemas:
- Sistema del problema de cómputo 3.2.21.
 - Sistema del problema de cómputo 3.2.22.
 - Sistema (3) usando los valores iniciales $(0, 0, 0)$.
 - Usando los valores iniciales $(\frac{3}{4}, \frac{1}{2}, -\frac{1}{2})$, resuelva

$$\begin{cases} x + y + z = 0 \\ x^2 + y^2 + z^2 = 2 \\ x(y + z) = -1 \end{cases}$$

- e.** Usando los valores iniciales $(-0.01, -0.01)$, resuelva

$$\begin{cases} 4y^2 + 4y + 52x - 19 = 0 \\ 169x^2 + 3y^2 + 111x - 10y - 10 = 0 \end{cases}$$

f. Escoja valores iniciales y resuelva

$$\begin{cases} \sin(x+y) = e^{x-y} \\ \cos(x+6) = x^2 y^2 \end{cases}$$

24. Investigue el comportamiento del método de Newton para determinar raíces complejas de polinomios con coeficientes reales. Por ejemplo, el polinomio $p(x) = x^2 + 1$ tiene un par de raíces complejas conjugadas $\pm i$ y el método de Newton es $x_{n+1} = (x_n - l/x_n)$. Primero, programe este método usando aritmética real y números reales como valores iniciales. Segundo, modifique el programa usando aritmética compleja pero aún usando sólo valores iniciales reales. Por último, use números complejos como valores iniciales. Observe el comportamiento de las iteraciones en cada caso.

25. Usando el problema 3.2.40, encuentre una raíz compleja de cada una de las siguientes ecuaciones:

$$\text{a. } z^3 - z - 1 = 0$$

$$\text{b. } z^4 - 2z^3 - 2iz^2 + 4iz = 0$$

$$\text{c. } 2z^3 - 6(1+i)z^2 - 6(1-i) \equiv 0$$

d. $z = e^z$

Sugerencia: para el último inciso, use la relación de Euler $e^{iy} = \cos y + i \operatorname{sen} y$.

26. En el método de Newton, para determinar una raíz r de $f(x) = 0$, iniciamos con x_0 y calculamos la sucesión x_1, x_2, \dots usando la fórmula $x_{n+1} = x_n - f(x_n)/f'(x_n)$. Para que no sea necesario calcular la derivada en cada paso, se ha propuesto remplazar $f'(x_n)$ con $f'(x_0)$ en todos los pasos. También se ha sugerido que la derivada en la fórmula de Newton se calcule sólo un paso sí y otro no. Este método está dado por

$$\begin{cases} x_{2n+1} = x_{2n} - \frac{f(x_{2n})}{f'(x_{2n})} \\ x_{2n+2} = x_{2n+1} - \frac{f(x_{2n+1})}{f'(x_{2n})} \end{cases}$$

Numéricamente compare los dos métodos propuestos con el método de Newton para varias funciones simples que tienen raíces conocidas. Imprima el error de cada método en cada iteración para monitorear la convergencia. ¿Cuán bien trabajan los métodos propuestos?

- 27. (Cuenca de atracción)** Considere el polinomio complejo $z^3 - 1$, cuyos ceros son las tres raíces cúbicas de la unidad. Genere una imagen que muestre las tres cuencas de atracción en el plano complejo en la región cuadrada definida por $-1 \leq \text{Real}(z) \leq 1$ y $-1 \leq \text{Imaginario}(z) \leq 1$. Para hacer esto, use una malla de 1000×1000 pixeles dentro del cuadrado. El punto central de cada pixel se usa para iniciar la iteración del método de Newton. Asigne un color particular a la cuenca para cada pixel si se obtiene la convergencia a una raíz con $n_{\max} = 10$ iteraciones. El gran número de iteraciones sugerido se pueden evitar haciendo algunos análisis con la ayuda del teorema 1, puesto que las iteraciones se obtienen dentro de cierta vecindad de la raíz y la iteración se puede parar. El criterio de convergencia es comprobar que $|z_{n+1} - z_n| < \varepsilon$ y $|z_{n+1}^3 - 1| < \varepsilon$ con un pequeño valor como $\varepsilon = 10^{-4}$, así como con un número máximo de iteraciones. *Sugerencia:* es mejor depurar su programa y obtener una imagen cruda con sólo un número pequeño de pixeles como 10×10 .

- 28.** (Continuación) Repita para el polinomio $z^4 - 1 = 0$.

29. Escriba una **función real** $Sqrt(x)$ para calcular la raíz cuadrada de un argumento real x por el siguiente algoritmo: primero, reduzca el rango de x al encontrar un número real r y un entero m

tal que $x = 2^m r$ con $\frac{1}{4} \leq r < 1$. Ahora, calcule x_2 usando tres iteraciones del método de Newton dadas por

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{r}{x_n} \right)$$

+1

con la aproximación inicial especial

$$x_0 = 1.27235\,367 + 0.24269\,3281r - \frac{1.02966\,039}{1+r}$$

Entonces haga $\sqrt{x} \approx 2^m x_2$. Pruebe este algoritmo con varios valores de x . Obtenga un listado del código para la función raíz cuadrada en su sistema de cómputo. Al leer los comentarios, intente determinar qué algoritmo usa.

- 30.** El siguiente método tiene convergencia de tercer orden para calcular \sqrt{R}

$$x_{n+1} = \frac{x_n(x_n^2 + 3R)}{3x_n^2 + R}$$

Realice algunos experimentos numéricos usando este método y el método del problema anterior para ver si observa una diferencia en la tasa de convergencia. Use los mismos procedimientos iniciales de reducción de rango y aproximación inicial.

- 31.** Escriba **función real RaízCúbica(x)** para calcular la raíz cúbica de un argumento real x con el siguiente procedimiento: primero, determine un número real r y un entero m tal que $x = r2^{3m}$ con $\frac{1}{8} \leq r < 1$. Calcule x_4 usando cuatro iteraciones del método de Newton:

$$x_{n+1} = \frac{2}{3} \left(x_n + \frac{r}{2x_n^2} \right)$$

con el valor inicial especial

$$x_0 = 2.50292\,6 - \frac{8.04512\,5(r + 0.38775\,52)}{(r + 4.61224\,4)(r + 0.38775\,52) - 0.35984\,96}$$

Entonces haga $\sqrt[3]{x} \approx 2^m x_4$. Pruebe este algoritmo con varios valores de x .

- 32.** Use software matemático como en Maple o Mathematica para calcular diez iteraciones del método de Newton iniciando con $x_0 = 0$ para $f(x) = x^3 - 2x^2 + x - 3$. Con 100 lugares decimales de exactitud y después de nueve iteraciones, muestre que el valor de x es

$$\begin{aligned} 2.17455\,94102\,92980\,07420\,23189\,88695\,65392\,56759\,48725\,33708 \\ 24983\,36733\,92030\,23647\,64792\,75760\,66115\,28969\,38832\,0640 \end{aligned}$$

Muestre que los valores de la función en cada iteración son $9.0, 2.0, 0.26, 0.0065, 0.45 \times 10^{-5}, 0.22 \times 10^{-11}, 0.50 \times 10^{-24}, 0.27 \times 10^{-49}, 0.1 \times 10^{-98}$ y 0.1×10^{-98} . De nuevo observe que el número de dígitos de exactitud en el método de Newton se duplica (aproximadamente) con cada iteración una vez que esté suficientemente cerca de la raíz. (También, véase Bornemann, Wagon y Waldvogel [2004] para un desafío de 100 dígitos, el cual es un estudio de cálculo numérico de alta exactitud.)

- 33.** (Continuación) Use Maple o Mathematica para descubrir que esta raíz es exactamente

$$\sqrt[3]{\frac{79}{54} + \frac{1}{6}\sqrt{77}} + \frac{1}{9\sqrt[3]{\frac{79}{54} + \frac{1}{6}\sqrt{77}}} + \frac{2}{3}$$

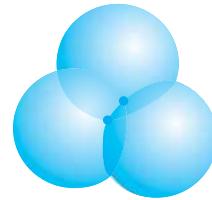
Obviamente, los resultados decimales son más interesantes para nosotros en nuestros estudios de métodos numéricos.

- 34.** (Continuación) Encuentre todas las raíces, incluidas las complejas.
- 35.** Numéricamente, encuentre todas las raíces del siguiente sistema de ecuaciones no lineales. Entonces trace las curvas para comprobar sus resultados:
- a.** $y = 2x^2 + 3x - 4, y = x^2 + 2x + 3$
 - b.** $y + x + 3 = 0, x^2 + y^2 = 17$
 - c.** $y = \frac{1}{2}x - 5, y = x^2 + 2x - 15$
 - d.** $xy = 1, x + y = 2$
 - e.** $y = x^2, x^2 + (y - 2)^2 = 4$
 - f.** $3x^2 + 2y^2 = 35, 4x^2 - 3y^2 = 24$
 - g.** $x^2 - xy + y^2 = 21, x^2 + 2xy - 8y^2 = 0$
- 36.** Aplique el método de Newton en estos problemas de prueba:
- a.** $f(x) = x^2$. *Sugerencia:* la primera derivada es cero en la raíz y la convergencia puede no ser cuadrática.
 - b.** $f(x) = x + x^{4/3}$. *Sugerencia:* no hay segunda derivada en la raíz y la convergencia puede no ser cuadrática.
 - c.** $f(x) = x + x^2 \operatorname{sen}(2/x)$ para $x \neq 0$ y $f(0) = 0$ y $f'(x) = 1 + 2x \operatorname{sen}(2/x) - 2\cos(2/x)$ para $x \neq 0$ y $f'(0) = 1$. *Sugerencia:* la derivada de esta función no es continua en la raíz y la convergencia puede fallar.
- 37.** Sea $\mathbf{F}(\mathbf{X}) = \begin{bmatrix} x_1^2 - x_2 + c \\ x_2^2 - x_1 + c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Cada ecuación componente $f_1(x) = 0$ y $f_2(x) = 0$ describe una parábola. Cualquier punto (x^*, y^*) donde estas dos parábolas se intersecan es una solución del sistema de ecuaciones no lineales. Usando el método de Newton para sistemas de ecuaciones no lineales, encuentre las soluciones para cada uno de estos valores del parámetro $c = \frac{1}{2}, \frac{1}{4}, -\frac{1}{2}, -1$. Dé la matriz jacobiana para cada una. También, para cada uno de estos valores trace las curvas resultantes mostrando los puntos de intersección. (Heath 2000, p. 218.)
- 38.** Sea $\mathbf{F}(\mathbf{X}) = \begin{bmatrix} x_1^2 + 2x_2 - 2 \\ x_1 + 4x_2^2 - 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. Resuelva este sistema no lineal iniciando con $\mathbf{X}^{(0)} = (1, 2)$. Dé la matriz jacobiana. También trace las curvas resultantes mostrando los puntos de intersección.
- 39.** Usando el método de Newton, encuentre los ceros de $f(z) = z^3 - z$ con estos valores iniciales $z^{(0)} = 1 + 1.5i, 1 + 1.1i, 1 + 1.2i, 1 + 1.3i$.
- 40.** Use el método Halley para producir una gráfica de las cuencas de atracción para $p(z) = z^6 - 1$. Compare con la figura 3.8.

- 41. (Proyecto sistema de posicionamiento global)** Cada vez que se usa un GPS, un sistema de ecuaciones no lineales de la forma

$$\begin{aligned}(x - a_1)^2 + (y - b_1)^2 + (z - c_1)^2 &= [(C(t_1) - D)]^2 \\(x - a_2)^2 + (y - b_2)^2 + (z - c_2)^2 &= [(C(t_2) - D)]^2 \\(x - a_3)^2 + (y - b_3)^2 + (z - c_3)^2 &= [(C(t_3) - D)]^2 \\(x - a_4)^2 + (y - b_4)^2 + (z - c_4)^2 &= [(C(t_4) - D)]^2\end{aligned}$$

se resuelve para las coordenadas (x, y, z) del receptor. Para cada satélite i , las posiciones son (a_i, b_i, c_i) y t_i es el tiempo de transmisión sincronizado del satélite. Además, C es la rapidez de la luz y D es la diferencia entre el tiempo de sincronización de los relojes del satélite y el reloj receptor fijo en tierra. Aunque sólo hay dos puntos en la intersección de las tres esferas (uno de los cuales se puede determinar que este en la posición deseada), se debe usar una cuarta esfera (satélite) para resolver la inexactitud en el reloj que tiene el receptor de bajo costo de la tierra. Explore diferentes maneras de resolver este sistema no lineal. Véase Hofmann-Wellenhof, Lichtenegger y Collins [2001], Sauer [2006] y Strang y Borre [1997].



- 42.** Use software matemático como Matlab, Maple o Mathematica y sus procedimientos incorporados para resolver el sistema de ecuaciones no lineales (8) del ejemplo 2. Además, trace las superficies dadas y la solución obtenida. *Sugerencia:* puede que necesite usar un punto inicial ligeramente perturbado $(0.5, 1.5, 0.5)$ para evitar una singularidad en la matriz jacobiana.

3.3 Método de la secante

Ahora consideraremos un procedimiento de propósito general que converge casi tan rápido como el método de Newton. Este método imita el de Newton pero evita el cálculo de derivadas. Recuerde que la iteración de Newton define a x_{n+1} en términos de x_n mediante la fórmula

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (1)$$

En el método de la secante remplazamos $f'(x_n)$ en la fórmula (1) mediante una aproximación que se calcula fácilmente. Puesto que la derivada está definida por

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

podemos decir que para h pequeña,

$$f'(x) \approx \frac{f(x + h) - f(x)}{h}$$

(En la sección 4.3 trataremos este tema nuevamente y aprenderemos que esta es una aproximación por diferencia finita a la primera derivada.) En particular, si $x = x_n$ y $h = x_{n-1} - x_n$, tenemos

$$f'(x_n) \approx \frac{f(x_{n-1}) - f(x_n)}{x_{n-1} - x_n} \quad (2)$$

Cuando se usa esto en la ecuación (1), el resultado define el **método de la secante**:

$$x_{n+1} = x_n - \left(\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right) f(x_n) \quad (3)$$

El método de la secante (parecido al de Newton) se puede usar para resolver también sistemas de ecuaciones.

El nombre del método se toma del hecho de que el miembro derecho de la ecuación (2) es la pendiente de una recta secante a la gráfica de f (figura 3.9). Por supuesto, el miembro izquierdo es la pendiente de una *recta tangente* a la gráfica de f . (De manera similar, el método de Newton podría haberse llamado el “método de la tangente”.)

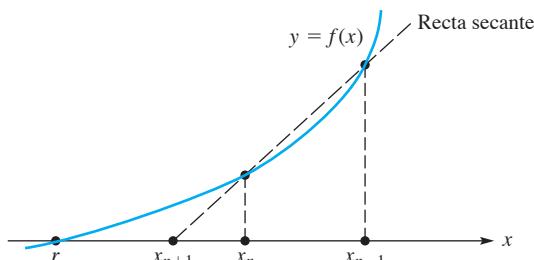


FIGURA 3.9
Método de la secante

Pocas observaciones acerca de la ecuación (3) están en orden. Obviamente, x_{n+1} depende de *dos* elementos anteriores de la sucesión. Así, para empezar se deben dar dos puntos (x_0 y x_1). La ecuación (3) puede entonces generar x_2, x_3, \dots . Al programar el método de la secante podríamos calcular y probar la cantidad $f(x_n) - f(x_{n-1})$. Si está cerca de cero, puede suceder un sobreflujo en la ecuación (3). Por supuesto, si el método tiene éxito los puntos x_n tenderán a un cero de f , por lo que $f(x_n)$ convergerá a cero. (Estamos suponiendo que f es continua.) También, $f(x_{n-1})$ convergerá a cero y, con más razón, $f(x_n) - f(x_{n-1})$ tenderá a cero. Si los términos $f(x_n)$ y $f(x_{n-1})$ tienen el mismo signo, los dígitos significativos adicionales se eliminan en la resta. Por ello, podríamos quizás detener la iteración cuando $|f(x_n) - f(x_{n-1})| \leq \delta |f(x_n)|$ con cierta tolerancia específica δ , tal como 1×10^{-6} (véase el problema de cómputo 3.3.18.)

Algoritmo de la secante

Un seudocódigo para $nmax$ pasos del método de la secante aplicado a la función f iniciando con el intervalo $[a, b] = [x_0, x_1]$ puede escribirse como sigue:

```

procedure Secante( $f, a, b, nmax, \epsilon$ )
  integer  $n, nmax$ ; real  $a, b, fa, fb, \epsilon, d$ 
  external function  $f$ 
   $fa \leftarrow f(a)$ 
   $fb \leftarrow f(b)$ 

```

```

if  $|fa| > |fb|$  then
     $a \longleftrightarrow b$ 
     $fa \longleftrightarrow fb$ 
end if
output 0,  $a, fa$ 
output 1,  $b, fb$ 
for  $n = 2$  to nmax do
    if  $|fa| > |fb|$  then
         $a \longleftrightarrow b$ 
         $fa \longleftrightarrow fb$ 
    end if
     $d \leftarrow (b - a)/(fb - fa)$ 
     $b \leftarrow a$ 
     $fb \leftarrow fa$ 
     $d \leftarrow d \cdot fa$ 
    if  $|d| < \epsilon$  then
        output "convergencia"
        return
    end if
     $a \leftarrow a - d$ 
     $fa \leftarrow f(a)$ 
    output  $n, a, fa$ 
end for
end procedure Secante

```

Aquí \longleftrightarrow significa intercambiar valores. Se intercambian los puntos finales $[a, b]$, si es necesario, para mantener $|f(a)| \leq |f(b)|$. Por tanto, los valores absolutos de la función son no crecientes; por esto, tenemos $|f(x_n)| \geq |f(x_{n+1})|$ para $n \geq 1$.

EJEMPLO 3 Si el método de la secante se usa en $p(x) = x^5 + x^3 + 3$ con $x_0 = -1$ y $x_1 = -1$, ¿a qué es igual x_8 ?

Solución La salida del programa de computadora correspondiente al pseudocódigo para el método de la secante es la siguiente (usamos una computadora de longitud de palabra de 32 bits).

n	x_n	$p(x_n)$
0	-1.0	1.0
1	1.0	5.0
2	-1.5	-7.97
3	-1.05575	0.512
4	-1.11416	-9.991×10^{-2}
5	-1.10462	7.593×10^{-3}
6	-1.10529	1.011×10^{-4}
7	-1.10530	2.990×10^{-7}
8	-1.10530	2.990×10^{-7}

Podemos usar software matemático para encontrar la única raíz real, -1.1053 y los dos pares de raíces complejas, $-0.319201 \pm 1.35008i$ y $0.871851 \pm 0.806311i$. ■

Análisis de convergencia

Las ventajas del método de la secante son que (después del primero paso) sólo se requiere una evaluación de la función por paso (en contraste con la iteración de Newton, que requiere dos) y que converge casi tan rápido como con el de Newton. Se puede mostrar que el método básico de la secante definido por la ecuación (3) obedece una ecuación de la forma

$$e_{n+1} = -\frac{1}{2} \left(\frac{f''(\xi_n)}{f'(\xi_n)} \right) e_n e_{n-1} \approx -\frac{1}{2} \left(\frac{f''(r)}{f'(r)} \right) e_n e_{n-1} \quad (4)$$

donde ξ_n y ζ_n están en el intervalo más pequeño que contiene a r , x_n y x_{n-1} . Por tanto, el cociente $e_{n+1}(e_n e_{n-1})^{-1}$ converge a $-\frac{1}{2} f''(r)/f'(r)$. La rapidez de convergencia de este método está, en general, entre la de la bisección y la del método de Newton.

Para demostrar la segunda parte de ecuación (4) comenzamos con la definición del método de la secante en la ecuación (3) y el error

$$\begin{aligned} e_{n+1} &= r - x_{n+1} \\ &= r - \frac{f(x_n)x_{n-1} - f(x_{n-1})x_n}{f(x_n) - f(x_{n-1})} \\ &= \frac{f(x_n)e_{n-1} - f(x_{n-1})e_n}{f(x_n) - f(x_{n-1})} \\ &= \left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right] \left[\frac{\frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}}}{x_n - x_{n-1}} \right] e_n e_{n-1} \end{aligned} \quad (5)$$

Por el teorema de Taylor, establecemos

$$f(x_n) = f(r - e_n) = f(r) - e_n f'(r) + \frac{1}{2} e_n^2 f''(r) + \mathcal{O}(e_n^3)$$

Puesto que $f(r) = 0$, tenemos

$$\frac{f(x_n)}{e_n} = -f'(r) + \frac{1}{2} e_n f''(r) + \mathcal{O}(e_n^2)$$

Cambiando el índice a $n - 1$ se obtiene

$$\frac{f(x_{n-1})}{e_{n-1}} = -f'(r) + \frac{1}{2} e_{n-1} f''(r) + \mathcal{O}(e_{n-1}^2)$$

Restando estas ecuaciones entre sí llegamos a

$$\frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}} = \frac{1}{2} (e_n - e_{n-1}) f''(r) + \mathcal{O}(e_{n-1}^2)$$

Puesto que $x_n - x_{n-1} = e_{n-1} - e_n$, llegamos a la ecuación

$$\frac{\frac{f(x_n)}{e_n} - \frac{f(x_{n-1})}{e_{n-1}}}{x_n - x_{n-1}} \approx -\frac{1}{2} f''(r)$$

La primera expresión entre corchetes en la ecuación (5) puede escribirse como

$$\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \approx \frac{1}{f'(r)}$$

Por tanto, hemos demostrado la segunda parte de la ecuación (4).

Dejamos el establecimiento de la primera parte de la ecuación (4) como un problema, ya que depende de algunos materiales que se cubrirán en el capítulo 4 (véase el problema 3.3.18).

De la ecuación (4), el orden de convergencia para el método de la secante se puede expresar en términos de la desigualdad

$$|e_{n+1}| \leq C |e_n|^\alpha \quad (6)$$

donde $\alpha = \frac{1}{2}(1 + \sqrt{5}) \approx 1.62$ es la **razón aurea**. Puesto que $\alpha > 1$, decimos que la convergencia es **superlineal**. Suponiendo que la desigualdad (6) es válida, podemos mostrar que el método de la secante converge bajo ciertas condiciones.

Sea $c = c(\delta)$ definida como en la ecuación (2) de la sección 3.2. Si $|r - x_n| \leq \delta$ y $|r - x_{n-1}| \leq \delta$, para alguna raíz r , entonces con la ecuación (4) se obtiene

$$|e_{n+1}| \leq c |e_n| |e_{n-1}| \quad (7)$$

Suponga que los puntos iniciales x_0 y x_1 están suficientemente cercanos a r tal que $c|e_0| \leq D$ y $c|e_1| \leq D$ para alguna $D < 1$. Entonces

$$\begin{aligned} c|e_1| &\leq D, \quad c|e_0| \leq D \\ c|e_2| &\leq c|e_1| c|e_0| \leq D^2 \\ c|e_3| &\leq c|e_2| c|e_1| \leq D^3 \\ c|e_4| &\leq c|e_3| c|e_2| \leq D^5 \\ c|e_5| &\leq c|e_4| c|e_3| \leq D^8 \\ &\text{etc.} \end{aligned}$$

En general, tenemos

$$|e_n| \leq c^{-1} D^{\lambda_{n+1}} \quad (8)$$

donde por inducción,

$$\begin{cases} \lambda_1 = 1, & \lambda_2 = 1 \\ \lambda_n = \lambda_{n-1} + \lambda_{n-2} & (n \geq 3) \end{cases} \quad (9)$$

Esta es la relación de recurrencia para generar la famosa **serie de Fibonacci**, 1, 1, 2, 3, 5, 8, ... Se puede demostrar que tienen la forma explícita sorprendente

$$\lambda_n = \frac{1}{\sqrt{5}} (\alpha^n - \beta^n) \quad (10)$$

donde $\alpha = \frac{1}{2}(1 + \sqrt{5})$ y $\beta = \frac{1}{2}(1 - \sqrt{5})$. Puesto que $D < 1$ y $\lambda_n \rightarrow \infty$, concluimos de la desigualdad (8) que $e_n \rightarrow 0$. Por tanto, $x_n \rightarrow r$ conforme $n \rightarrow \infty$ y el método de la secante converge a la raíz r si x_0 y x_1 están suficientemente cercanas a ésta.

A continuación, demostramos que la desigualdad (6) es de hecho *razonable*, no una prueba. De las ecuaciones (7), ahora tenemos

$$\begin{aligned}|e_{n+1}| &\leq c|e_n||e_{n-1}| \\&= c|e_n|^{\alpha}|e_n|^{1-\alpha}|e_{n-1}| \\&\approx c|e_n|^{\alpha}(c^{-1}D^{\lambda_{n+1}})^{1-\alpha}(c^{-1}D^{\lambda_n}) \\&= |e_n|^{\alpha}c^{\alpha-1}D^{\lambda_{n+1}(1-\alpha)+\lambda_n} \\&= |e_n|^{\alpha}c^{\alpha-1}D^{\lambda_{n+2}-\alpha\lambda_{n+1}}\end{aligned}$$

usando una aproximación a la desigualdad (8). En el último renglón usamos la relación de recurrencia (9). Ahora $\lambda_{n+2} - \alpha\lambda_{n+1}$ converge a cero (véase el problema 3.3.6). Por tanto, $c^{\alpha-1} D^{\lambda_{n+2}-\alpha\lambda_{n+1}}$ está acotado, digamos, por C , como una función de n . Por esto, tenemos

$$|e_{n+1}| \approx C|e_n|^{\alpha}$$

que es una aproximación razonable de la desigualdad (6).

Otra deducción (con un poco de *discusión acalorada*) para el orden de convergencia del método de la secante se puede dar usando una relación de recurrencia general. Con la ecuación (4) obtenemos

$$e_{n+1} \approx K e_n e_{n-1}$$

donde $K = -\frac{1}{2}f''(r)/f'(r)$. Podemos escribir esta como

$$|K e_{n+1}| \approx |K e_n| |K e_{n-1}|$$

Sea $z_i = \log |K e_i|$. Entonces queremos resolver la ecuación de recurrencia

$$z_{n+1} = z_n + z_{n-1}$$

donde z_0 y z_1 son arbitrarios. Esta es una relación de recurrencia lineal con coeficientes constantes similares a los de los números de Fibonacci (9) excepto que los primeros dos valores z_0 y z_1 no se conocen. La solución es de la forma

$$z_n = A\alpha^n + B\beta^n \tag{11}$$

donde $\alpha = \frac{1}{2}(1 + \sqrt{5})$ y $\beta = \frac{1}{2}(1 - \sqrt{5})$. Estas son las raíces de la ecuación cuadrática $\lambda^2 - \lambda - 1 = 0$. Puesto que $|\alpha| > |\beta|$, el término $A\alpha^n$ domina y podemos decir que

$$z_n \approx A\alpha^n$$

para n grandes y para alguna constante A . Por tanto, tenemos

$$|K e_n| \approx 10^{A\alpha^n}$$

Entonces se concluye que

$$|K e_{n+1}| \approx 10^{A\alpha^{n+1}} = (10^{A\alpha^n})^\alpha = |K e_n|^\alpha$$

Por tanto, tenemos

$$|e_{n+1}| \approx C|e_n|^\alpha \tag{12}$$

para n grande y para alguna constante C . De nuevo, la desigualdad (6) ¡está *esencialmente* establecida! Una rigurosa prueba de la desigualdad (6) es tediosa y muy larga.

Comparación de métodos

En este capítulo se han presentado tres métodos elementales para resolver una ecuación $f(x) = 0$. El método de bisección es confiable pero lento. El método de Newton es rápido pero con frecuencia sólo cerca de la raíz y requiere a f' . El método de la secante es casi tan rápido como el de Newton y no requiere conocer la derivada f' , la que puede no estar disponible o puede ser muy costoso calcular. El usuario del método de bisección debe proporcionar dos puntos en los que los signos de $f(x)$ difieren y la función f sólo necesita ser continua. Al usar el método de Newton, se debe especificar un punto de inicio cerca de la raíz y f debe ser derivable. El método de la secante requiere dos buenos puntos de inicio. El procedimiento de Newton se puede interpretar como la repetición de un procedimiento de dos pasos que se resume con la prescripción *linealiza y resuelve*. Esta estrategia es aplicable en muchos otros problemas numéricos y su importancia no se puede sobreestimar. Los dos métodos, el de Newton y el método de la secante fallan en cercar una raíz. El método modificado de falsa posición puede retener las ventajas de los dos métodos.

El método de la secante es con frecuencia más rápido para aproximar las raíces de funciones no lineales en comparación con el de bisección y de la falsa posición. A diferencia de estos dos métodos, los intervalos $[a_k, b_k]$ no tienen que estar en los lados opuestos de la raíz y tienen un cambio de signo. Por otra parte, la pendiente de la recta secante puede ser muy pequeña y con un paso puede moverse lejos del punto actual. El método de la secante puede fallar al encontrar una raíz de una función no lineal que tiene una pendiente pequeña cerca de la raíz, ya que la recta secante puede brincar una gran cantidad.

Para funciones agradables y suposiciones relativamente cercanas a la raíz, la mayoría de estos métodos requiere relativamente pocas iteraciones antes de acercarse a la raíz. Sin embargo, hay funciones patológicas que pueden causar problemas para cualquiera de estos métodos. Cuando seleccionamos un método para resolver un problema no lineal se deben considerar muchas cosas tales como lo que usted sabe acerca del comportamiento de la función, un intervalo $[a, b]$ que satisface $f(a)f(b) < 0$, la primera derivada de la función, una buena suposición inicial de la raíz deseada y así sucesivamente.

Esquemas híbridos

En un esfuerzo por encontrar el *mejor* algoritmo para determinar un cero de una función se han desarrollado varios métodos híbridos. Algunos de estos procedimientos combinan el método de bisección (usado durante las primeras iteraciones) ya sea con el método de la secante o el método de Newton. También, los esquemas adaptados se usan para monitorear las iteraciones y para realizar las reglas de parada. Más información sobre algunos métodos híbridos secante-bisección y métodos híbridos Newton-bisección con reglas adaptadas de parada se pueden encontrar en Bus y Dekker [1975], Dekker [1969], Kahaner, Moler y Nash [1989] y Novak, Ritter y Woźniakowski [1995].

Iteración de punto fijo

Para una ecuación no lineal $f(x) = 0$, buscamos un punto donde la curva f interseca el eje x ($y = 0$). Un enfoque alternativo es reformular el problema como un problema de punto fijo $x = g(x)$ para una función relacionada no lineal g . Para el problema de punto fijo, buscamos un punto donde la curva g interseca la recta diagonal $y = x$. Un valor de x tal que $x = g(x)$ es un **punto fijo** de g , ya que x no se cambia cuando g se aplica a ésta. Muchos algoritmos iterativos para resolver una ecuación no lineal $f(x) = 0$ están basados en un método iterativo de punto fijo $x^{(n+1)} = g(x^{(n)})$, donde g tiene puntos

fijos que son soluciones de $f(x) = 0$. Se selecciona un valor inicial $x^{(0)}$ y el método iterativo se aplica repetidamente hasta que converge suficientemente bien.

EJEMPLO 2 Aplique el procedimiento de punto fijo, donde $g(x) = 1 + 2/x$, iniciando con $x^{(0)} = 1$, para calcular un cero de la función no lineal $f(x) = x^2 - x - 2$. Gráficamente, trace el proceso de convergencia.

Solución El método de punto fijo es

$$x^{(n+1)} = 1 + \frac{2}{x^{(n)}}$$

Ocho pasos del algoritmo iterativo son $x^{(0)} = 1, x^{(1)} = 3, x^{(2)} = 5/3, x^{(3)} = 11/5, x^{(4)} = 21/11, x^{(5)} = 43/21, x^{(6)} = 85/43, x^{(7)} = 171/85$ y $x^{(8)} = 341/171 \approx 1.99415$. En la figura 3.10 vemos estos pasos de espiral dentro del punto fijo 2.

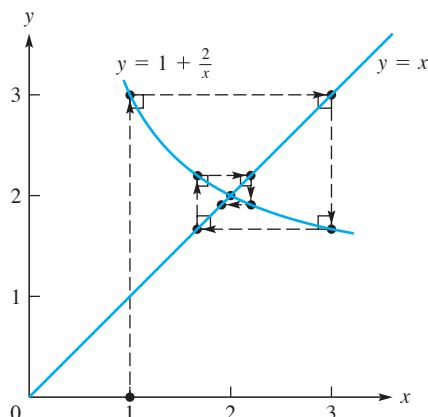


FIGURA 3.10
Iteraciones para el punto fijo
 $f(x) = x^2 - x - 2$

Para una ecuación no lineal dada $f(x) = 0$, puede haber muchos problemas de punto fijo equivalentes $x = g(x)$ con diferentes funciones g , algunos mejores que otros. Una forma simple de caracterizar el comportamiento de un método iterativo $x^{(n+1)} = g(x^{(n)})$ es *localmente convergente* para x^* si $x^* = g(x^*)$ y $|g'(x^*)| < 1$. *Localmente convergente* significa que hay un intervalo que contiene a $x^{(0)}$ tal que el método de punto fijo converge para cualquier valor inicial $x^{(0)}$ dentro del intervalo. Si $|g'(x^*)| > 1$, entonces el método de punto fijo diverge para cualquier punto de inicio $x^{(0)}$ distinto de x^* . Los métodos iterativos de punto fijo se usan comúnmente en la práctica para resolver varios problemas de ciencia y de ingeniería. De hecho, la teoría del punto fijo puede simplificar la prueba de la convergencia del método de Newton.

Resumen

(1) El **método de la secante** para determinar un cero r de una función $f(x)$ se escribe como

$$x_{n+1} = x_n - \left(\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right) f(x_n)$$

para $n \geq 1$, que requiere dos valores iniciales x_0 y x_1 . Después del primer paso, sólo es necesaria una nueva evaluación de función por paso.

(2) Después de $n + 1$ pasos del método de la secante, el error de la iteración $e_i = r - x_i$ obedece la ecuación

$$e_{n+1} = -\frac{1}{2} \left(\frac{f'(\xi_n)}{f'(\zeta_n)} \right) e_n e_{n-1}$$

que conduce a la aproximación

$$|e_{n+1}| \approx C |e_n|^{1/2(1+\sqrt{5})} \approx C |e_n|^{1.62}$$

Por tanto, el método de la secante tiene un comportamiento de **convergencia superlineal**.

Referencias adicionales

Como lectura complementaria y estudio, véase Barnsley [2006], Bus y Dekker [1975], Dekker [1969], Dennis y Schnabel [1983], Epureanu y Greenside [1998], Fauvel, Flood, Shortland y Wilson [1988], Feder [1988], Ford [1995], Householder [1970], Kelley [1995], Lozier y Olver [1994], Nericke y Haegemans [1976], Novak, Ritter y Woźniakowski [1995], Ortega y Rheinboldt [1970], Ostrowski [1966], Rabinowitz [1970], Traub [1964], Westfall [1995] e Ypma [1995].

Problemas 3.3

- 1.** Calcule un valor aproximado para $4^{3/4}$ usando un paso del método de la secante con $x_0 = 3$ y $x_1 = 2$.
- 2.** Si usamos el método de la secante en $f(x) = x^3 - 2x + 2$ iniciando con $x_0 = 0$ y $x_1 = 1$, ¿a qué es igual x_2 ?
- 3.** Si el método de la secante se usa en $f(x) = x^5 + x^3 + 3$ y si $x_{n-2} = 0$ y $x_{n-1} = 0$, ¿a qué es igual x_n ?
- 4.** Si $x_{n+1} = x_n + (2 - e^{x_n})(x_n - x_{n-1})/(e^{x_n} - e^{x_{n-1}})$ con $x_0 = 0$ y $x_1 = 0$, ¿a qué es igual $\lim_{n \rightarrow \infty} x_n$?
- 5.** Usando el método de bisección, el método de Newton y el método de la secante, encuentre la raíz positiva más grande correcta a tres lugares decimales de $x^3 - 5x + 3 = 0$. (Todas raíces están en $[-3, +3]$).
- 6.** Demuestre que en el primer análisis del método de la secante, $\lambda_{n+1} - a\lambda_n$ converge a cero cuando $n \rightarrow \infty$.
- 7.** Establezca la ecuación (10).
- 8.** Escriba la deducción del orden de convergencia del método de la secante que usa relaciones de recurrencia; esto es, encuentre las constantes A y B en la ecuación (11) y complete los detalles para obtener la ecuación (12).

- 9.** ¿Cuál es la fórmula adecuada para determinar raíces cuadradas usando el método de la secante? (Consulte al problema 3.2.1.)

- 10.** La fórmula para el método de la secante también se puede escribir como

$$x_{n+1} = \frac{x_{n-1}f(x_n) - x_nf(x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Establezca esto y explique por qué es inferior a la ecuación (3) en un programa de computadora.

- 11.** Demuestre que si las iteraciones en el método de Newton convergen a un punto r para el que $f'(r) \neq 0$, entonces $f(r) = 0$. Establezca el mismo enunciado para el método de la secante. *Sugerencia:* en la segunda parte, es útil el teorema del valor medio del cálculo diferencial. Este es el caso $n = 0$ en el teorema de Taylor.

- 12.** Un método para determinar un cero de una función dada f procede como sigue. Se eligen dos aproximaciones iniciales de x_0 y x_1 a cero, se fija el valor de x_0 y las iteraciones sucesivas están dadas por

$$x_{n+1} = x_n - \left(\frac{x_n - x_0}{f(x_n) - f(x_0)} \right) f(x_n)$$

Este proceso convergerá a un cero de f bajo ciertas condiciones. Demuestre que la rapidez de convergencia a un cero simple es *lineal* bajo algunas condiciones.

- 13.** Pruebe las sucesiones siguientes para diferentes tipos de convergencia (es decir, lineal, superlineal o cuadrática), donde $n = 1, 2, 3, \dots$

$$\text{a. } x_n = n^{-2} \quad \text{b. } x_n = 2^{-n} \quad \text{c. } x_n = 2^{-2^n}$$

$$\text{d. } x_n = 2^{-a_n} \text{ con } a_0 = a_1 = 1 \text{ y } a_{n+1} = a_n + a_{n-1} \text{ para } n \geq 2$$

- 14.** Este problema y los siguientes tres se relacionan con el método de la **iteración funcional**. El método de la iteración funcional es el siguiente: iniciando con cualquier x_0 , definimos $x_{n+1} = f(x_n)$, donde $n = 0, 1, 2, \dots$. Muestre que si f es continua y si la sucesión $\{x_n\}$ converge, entonces su límite es un punto fijo de f .

- 15.** (Continuación) Demuestre que si f es una función definida en toda la recta real cuya derivada satisface $|f'(x)| \leq c$ con una constante c menor que 1, entonces el método de la iteración funcional produce un punto fijo de f . *Sugerencia:* para establecer esto, es útil el teorema del valor medio de la sección 1.2.

- 16.** (Continuación) Con una calculadora, intente el método de la iteración funcional con $f(x) = x/2 + 1/x$, tomando $x_0 = 1$. ¿Cuál es el límite de la sucesión resultante?

- 17.** (Continuación) Usando iteración funcional, muestre que la ecuación $10 - 2x + \sin x = 0$ tiene una raíz. Localice la raíz aproximadamente por medio de una gráfica. Iniciando con su raíz aproximada, use la iteración funcional para obtener la raíz exactamente usando una calculadora. *Sugerencia:* escriba la ecuación en la forma $x = 5 + \frac{1}{2} \sin x$.

- 18.** Establezca la primera parte de la ecuación (4) usando la ecuación (5). *Sugerencia:* use la relación entre diferencias divididas y derivadas de la sección 4.2.

Problemas de cómputo 3.3

- ^a1. Use el método de la secante para encontrar el cero cerca de -0.5 de $f(x) = e^x - 3x^2$. Esta función también tiene un cero cerca de 4 . Encuentre este cero positivo usando el método de Newton.

2. Escriba

```
procedure Secante(f, x1, x2, epsi, delta, maxf, x, ierr)
```

que usa el método de la secante para resolver $f(x) = 0$. Los parámetros de entrada son los siguientes: f es el nombre de la función dada; x_1 y x_2 son las estimaciones iniciales de la solución; $epsi$ es una tolerancia positiva tal que la iteración se detiene si la diferencia entre dos iteraciones consecutivas es menor que este valor; $delta$ es una tolerancia positiva tal que la iteración se detiene si un valor de la función es menor en magnitud que este valor; y $maxf$ es un entero positivo que acota el número de evaluaciones de la función permitida. Los parámetros de salida son los siguientes: x es la estimación final de la solución e $ierr$ es una bandera de error de entero que indica si una prueba de tolerancia fue violada. Pruebe esta rutina usando la función del problema de cómputo 3.3.1. Imprima la estimación final de la solución y el valor de la función en este punto.

3. Encuentre un cero de una de las funciones dadas en la introducción de este capítulo usando uno de los métodos presentados en este capítulo.
4. Escriba y pruebe un procedimiento recursivo para el método de la secante.
5. Ejecute de nuevo el ejemplo de esta sección con $x_0 = 0$ y $x_1 = 1$. Explique cualquier resultado inusual.
6. Escriba un programa sencillo para comparar el método de la secante con el método de Newton para determinar una raíz de cada función.

$$\text{a. } x^3 - 3x + 1 \text{ con } x_0 = 2 \quad \text{b. } x^3 - 2 \sin x \text{ con } x_0 = \frac{1}{2}$$

Use el valor x_1 del método de Newton como el segundo punto de inicio para el método de la secante. Imprima cada iteración para los dos métodos.

- ^a7. Escriba un programa sencillo para encontrar la raíz de $f(x) = x^3 + 2x^2 + 10x - 20$ usando el método de la secante con valores iniciales $x_0 = 2$ y $x_1 = -1$. Déjelo correr a lo más 20 pasos e incluya también una prueba de parada. Compare el número de pasos necesario aquí con el número de pasos necesarios con el método de Newton. ¿La convergencia es cuadrática?
8. Pruebe el método de la secante con el conjunto de funciones $f_k(x) = 2e^{-k}x + 1 - 3e^{-k}x$ para $k = 1, 2, 3, \dots, 10$. Use los puntos de inicio 0 y 1 en cada caso.
- ^a9. Un ejemplo de Wilkinson [1963] muestra que diminutas alteraciones en los coeficientes de un polinomio pueden tener efectos masivos en las raíces. Sea

$$f(x) = (x - 1)(x - 2) \cdots (x - 20)$$

que se conoce como el **polinomio de Wilkinson**. Los ceros de f son, por supuesto, los enteros $1, 2, \dots, 20$. Trate de determinar qué le ocurre al cero $r = 20$ cuando la función se altera como $f(x) - 10^{-8}x^{19}$. *Sugerencia:* el método de la secante en doble precisión localizará un cero en el intervalo $[20, 21]$.

- 10.** Pruebe el método de la secante en un ejemplo en el que r , $f'(r)$ y $f''(r)$ se conocen de antemano. Supervise los cocientes $e_{n+1}/(e_n e_{n-1})$ para ver si convergen a $-\frac{1}{2}f''(r)/f'(r)$. La función $f(x) = \arctan x$ es adecuada para este experimento.

- 11.** Usando una función de su elección, compruebe numéricamente que el método iterativo

$$x_{n+1} = x_n - \frac{f(x_n)}{\sqrt{[f'(x_n)]^2 - f(x_n)f''(x_n)}}$$

es cúbicamente convergente en una raíz simple pero sólo linealmente convergente en una raíz múltiple.

- 12.** Pruebe numéricamente si el **método de Olver**, dado por

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{1}{2} \frac{f''(x_n)}{f'(x_n)} \left[\frac{f(x_n)}{f'(x_n)} \right]^2$$

es cúbicamente convergente a una raíz de f . Intente establecer que lo es.

- 13.** (Continuación) Repita para el **método de Halley**

$$x_{n+1} = x_n - \frac{1}{a_n} \quad \text{con} \quad a_n = \frac{f'(x_n)}{f(x_n)} - \frac{1}{2} \left[\frac{f''(x_n)}{f'(x_n)} \right]$$

- 14. (Algoritmo de Moler-Morrison)** El cálculo de una aproximación para $\sqrt{x^2 + y^2}$ que no requiere raíces cuadradas, se puede hacer como sigue:

```

real function  $f(x, y)$ 
integer  $n$ ; real  $a, b, c, x, y$ 
 $f \leftarrow \max\{|x|, |y|\}$ 
 $a \leftarrow \min\{|x|, |y|\}$ 
for  $n = 1$  to 3 do
     $b \leftarrow (a/f)^2$ 
     $c \leftarrow b/(4 + b)$ 
     $f \leftarrow f + 2cf$ 
     $a \leftarrow ca$ 
end for
end function  $f$ 

```

Pruebe el algoritmo con algunos casos simples tales como $(x, y) = (3, 4)$, $(-5, 12)$ y $(7, -24)$. Entonces escriba una rutina que use la función $f(x, y)$ para aproximar la **norma euclíadiana** de un vector $x = (x_1, x_2, \dots, x_n)$; es decir, el número no negativo $\|x\| = (x_1^2 + x_2^2 + \dots + x_n^2)^{1/2}$.

- 15.** Estudie las siguientes funciones iniciando con cualquier valor inicial de x_0 en el dominio $[0, 2]$ e iterando $x_{n+1} = F(x_n)$. Primero use una calculadora y después una computadora. Explique los resultados.

- a.** Use la **función tienda de campaña**

$$F(x) = \begin{cases} 2x & \text{si } 2x < 1 \\ 2x - 1 & \text{si } 2x \geq 1 \end{cases}$$

- b.** Repita usando la función

$$F(x) = 10x \ (\text{módulo } 1)$$

Sugerencia: no se sorprenda de la conducta caótica. El lector interesado puede aprender más acerca de la dinámica de los mapas unidimensionales leyendo artículos como el de Bassien [1998].

16. Muestre cómo el método de la secante se puede utilizar para resolver sistemas de ecuaciones como las de los problemas 3.2.21–3.2.23.
17. (**Proyecto de investigación estudiantil**) El **método de Muller** es un algoritmo para calcular soluciones de una ecuación $f(x) = 0$. Es similar al método de la secante en que remplaza localmente a f por una función simple y encuentra una raíz de ésta. Naturalmente, este paso se repite. La función simple elegida en el método de Muller es un polinomio cuadrático, p , que interpola f en los tres puntos más recientes. Después de que p se ha determinado, se calculan sus raíces y se elige una de ellas como el punto siguiente en la sucesión. Puesto que esta función cuadrática puede tener raíces complejas, el algoritmo se debe programar pensando en esto. Suponga que los puntos x_{n-2} , x_{n-1} y x_n se han calculado. Haga

$$p(x) = a(x - x_n)(x - x_{n-1}) + b(x - x_n) + c$$

donde a , b y c están determinados de modo que p interpola a f en los tres puntos mencionados antes. Despues encuentre las raíces de p y tome x_{n+1} como la raíz de p más cercana a x_n . Al inicio, los tres puntos deben ser suministrados por el usuario. Programe el método, permitiendo números complejos en todo. Pruebe su programa en el ejemplo

$$p(x) = x^3 + x^2 - 10x - 10$$

Si los primeros tres puntos son 1, 2, 3, entonces debe encontrar que el polinomio es $p(x) = 7(x - 3)(x - 2) + 14(x - 3) - 4$ y $x_4 = 3.17971\ 086$. Despues, pruebe su código con un polinomio que tenga coeficientes reales pero algunas raíces complejas.

18. Programe y pruebe el código para el algoritmo de la secante después de incorporarle el criterio de parada descrito en el libro.
19. Usando software matemático como Matlab, Mathematica y Maple, encuentre el cero real del polinomio $p(x) = x^5 + x^3 + 3$. Obtenga más dígitos de exactitud que los que se muestran en la solución del ejemplo 1 del libro.
20. (Continuación) Usando software matemático que permite raíces complejas, encuentre todos los ceros del polinomio.
21. Programe un método híbrido para resolver varios problemas no lineales presentados como ejemplos en el libro y compare sus resultados con los dados.
22. Encuentre los puntos fijos para cada una de las siguientes funciones:
 - a. $e^x + 1$
 - b. $e^{-x} - x$
 - c. $x^2 - 4 \operatorname{sen} x$
 - d. $x^3 + 6x^2 + 11x - 6$
 - e. $\operatorname{sen} x$

23. Para la ecuación no lineal $f(x) = x^2 - x - 2 = 0$ con raíces 1 y 2, escriba cuatro problemas de punto fijo $x = g(x)$ que sean equivalentes. Trace la gráfica de todas y muestre que todas intersectan la recta $x = y$. También, trace la gráfica de los pasos de la convergencia de cada una de estas iteraciones de punto fijo para diferentes valores iniciales $x^{(0)}$. Muestre que el comportamiento de estos esquemas de punto fijo puede variar fuertemente: convergencia lenta, convergencia rápida y divergencia.

Interpolación y diferenciación numérica

La viscosidad del agua se ha determinado experimentalmente a temperaturas diferentes, como se indica en la tabla siguiente:

Temperatura	0°	5°	10°	15°
Viscosidad	1.792	1.519	1.308	1.140

A partir de esta tabla, ¿cómo podemos estimar un valor razonable para la viscosidad a una temperatura de 8°?

El método de interpolación polinomial, descrito en la sección 4.1, se puede usar para crear un polinomio de grado 3 que toma los valores en la tabla. Este polinomio debe proporcionar valores intermedios aceptables para temperaturas no tabuladas. El valor del polinomio en el punto 8° es de 1.386.

4.1 Interpolación polinomial

Observaciones preliminares

Planteamos tres problemas relacionados con la representación de funciones para dar una indicación de la materia en este capítulo, en el capítulo 9 (acerca de splines) y en el capítulo 12 (sobre mínimos cuadrados).

Primero, suponga que tenemos una tabla de valores numéricos de una función:

x	x_0	x_1	...	x_n
y	y_0	y_1	...	y_n

¿Es posible encontrar una fórmula simple y conveniente que reproduzca los puntos dados exactamente?

El segundo problema es similar, pero se supone que la tabla de valores numéricos dada está contaminada por errores, como puede ocurrir si los valores provienen de un experimento de física. Ahora nos preguntamos por una fórmula que represente los datos (aproximadamente) y, si es posible, filtre los errores.

Como un tercer problema, una función f está dada, quizás en la forma de un procedimiento computacional, pero es costoso evaluarla. En este caso, nos preguntamos por otra función g que sea simple de evaluar y produzca una aproximación razonable a f . A veces en este problema, queremos que g se aproxime a f con precisión total de máquina.

En todos estos problemas, se puede obtener una función simple p que represente o aproxime a la tabla o función f dadas. La representación p siempre se puede tomar como un polinomio, aunque también se pueden usar muchos otros tipos de funciones simples. Una vez que se ha obtenido una función simple p , se puede usar en lugar de f en muchas situaciones. Por ejemplo, la integral de f se podría estimar con la integral de p y, generalmente, esta última debe ser más fácil de evaluar.

En muchas situaciones, una solución polinomial a los problemas delineados antes no será satisfactoria desde un punto de vista práctico, se deben considerar otras clases de funciones. En este libro se analiza otra clase de funciones versátiles: las funciones spline (véase el capítulo 9). Este capítulo trata exclusivamente con polinomios y el capítulo 12 analiza familias de funciones lineales generales, en las que los splines y los polinomios son ejemplos importantes.

La forma obvia en la que un polinomio puede *fallar* como una solución práctica a uno de los problemas anteriores es que su grado puede ser irracionalmente alto. Por ejemplo, si la tabla considerada contiene 1000 entradas, se puede requerir un polinomio de grado 999 para representarla. También los polinomios pueden tener el sorprendente defecto de ser sumamente oscilatorios. Si la tabla se representa exactamente por medio de un polinomio p , entonces $p(x_i) = y_i$ para $0 \leq i \leq n$. Para otros puntos diferentes a los x_i dados, sin embargo, $p(x)$ puede ser una representación muy pobre de la función a partir de la cual surge la tabla. El ejemplo de la sección 4.2, que implica la función Runge, ilustra este fenómeno.

Interpolación polinomial

Comenzaremos de nuevo con una tabla de valores:

x	x_0	x_1	\cdots	x_n
y	y_0	y_1	\cdots	y_n

y suponemos que los de x_i forman un conjunto de $n + 1$ puntos distintos. La tabla representa $n + 1$ puntos en el plano cartesiano y queremos encontrar una curva polinomial que pase por todos ellos. Por tanto, buscamos determinar un polinomio que esté definido para *toda* x y tome los valores correspondientes de y_i para cada una de las $n + 1$ x_i distintas en esta tabla. Un polinomio p para el que $p(x_i) = y_i$ cuando $0 \leq i \leq n$ **interpola** la tabla. Los puntos x_i se llaman **nodos**.

Considere el primer caso y más simple, $n = 0$. Aquí, una función constante resuelve el problema. En otras palabras, el polinomio p de grado 0 definido por la ecuación $p(x) = y_0$ reproduce la tabla de un nodo.

El siguiente caso más simple sucede cuando $n = 1$. Puesto que una recta puede pasar por dos puntos, una función lineal es capaz de resolver el problema. Explícitamente, el polinomio p definido por

$$\begin{aligned} p(x) &= \left(\frac{x - x_1}{x_0 - x_1} \right) y_0 + \left(\frac{x - x_0}{x_1 - x_0} \right) y_1 \\ &= y_0 + \left(\frac{y_1 - y_0}{x_1 - x_0} \right) (x - x_0) \end{aligned}$$

es de primer grado (a lo más) y reproduce la tabla. Ello significa (en este caso) que $p(x_0) = y_0$ y $p(x_1) = y_1$, que es fácil de comprobar. Esta p se usa para **interpolación lineal**.

EJEMPLO 1 Encuentre el polinomio de menor grado que interpola esta tabla:

x	1.4	1.25
y	3.7	3.9

Solución Por la ecuación anterior, el polinomio que se busca es

$$\begin{aligned} p(x) &= \left(\frac{x - 1.25}{1.4 - 1.25} \right) 3.7 + \left(\frac{x - 1.4}{1.25 - 1.4} \right) 3.9 \\ &= 3.7 + \left(\frac{3.9 - 3.7}{1.25 - 1.4} \right) (x - 1.4) \\ &= 3.7 - \frac{4}{3}(x - 1.4) \end{aligned}$$

■

Como podemos ver, un polinomio de interpolación se puede escribir en una variedad de formas; entre ellas son conocidas la forma de Newton y la forma de Lagrange. La forma de Newton es quizás la más conveniente y eficiente; sin embargo, conceptualmente, la forma de Lagrange tiene varias ventajas. Comenzaremos con la forma de Lagrange, ya que puede ser más fácil de entender.

Polinomio de interpolación: forma de Lagrange

Suponga que queremos interpolar funciones arbitrarias en un conjunto de nodos fijos x_0, x_1, \dots, x_n . Primero definimos un sistema de $n + 1$ polinomios especiales de grado n conocidos como **polinomios cardinales** en la teoría de interpolación. Éstos se denotan por $\ell_0, \ell_1, \dots, \ell_n$ y tienen la propiedad

$$\ell_i(x_j) = \delta_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$$

Una vez que están disponibles, podemos interpolar *cualquier* función f por la **forma de Lagrange de interpolación polinomial**:

$$p_n(x) = \sum_{i=0}^n \ell_i(x) f(x_i) \tag{1}$$

Esta función p_n , al ser una combinación lineal de los polinomios ℓ_i , es en sí misma un polinomio de grado a lo más n . Además, cuando evaluamos p_n en x_i obtenemos $f(x_i)$:

$$p_n(x_j) = \sum_{i=0}^n \ell_i(x_j) f(x_i) = \ell_j(x_j) f(x_j) = f(x_j)$$

Por tanto, p_n es el polinomio de interpolación para la función f en los nodos x_0, x_1, \dots, x_n . Ahora sólo resta escribir la fórmula para el **polinomio cardinal** ℓ_i , que es

$$\ell_i(x) = \prod_{\substack{j \neq i \\ j=0}}^n \left(\frac{x - x_j}{x_i - x_j} \right) \quad (0 \leq i \leq n) \tag{2}$$

Esta fórmula indica que $\ell_i(x)$ es el producto de n factores lineales:

$$\ell_i(x) = \left(\frac{x - x_0}{x_i - x_0} \right) \left(\frac{x - x_1}{x_i - x_1} \right) \cdots \left(\frac{x - x_{i-1}}{x_i - x_{i-1}} \right) \left(\frac{x - x_{i+1}}{x_i - x_{i+1}} \right) \cdots \left(\frac{x - x_n}{x_i - x_n} \right)$$

(Los denominadores son sólo números; la variable x se presenta únicamente en los numeradores.) Por tanto, ℓ_i es un polinomio de grado n . Observe que cuando $\ell_i(x)$ se evalúa en $x = x_i$, cada factor en la ecuación anterior será 1. Por tanto, $\ell_i(x_i) = 1$. Pero cuando se evalúa $\ell_i(x)$ en cualquier otro nodo, digamos, x_j , uno de los factores en la ecuación anterior será 0 y $\ell_i(x_j) = 0$ para $i \neq j$.

La figura 4.1 muestra algunos de los primeros polinomios cardinales de Lagrange: $\ell_0(x)$, $\ell_1(x)$, $\ell_2(x)$, $\ell_3(x)$ y $\ell_4(x)$.

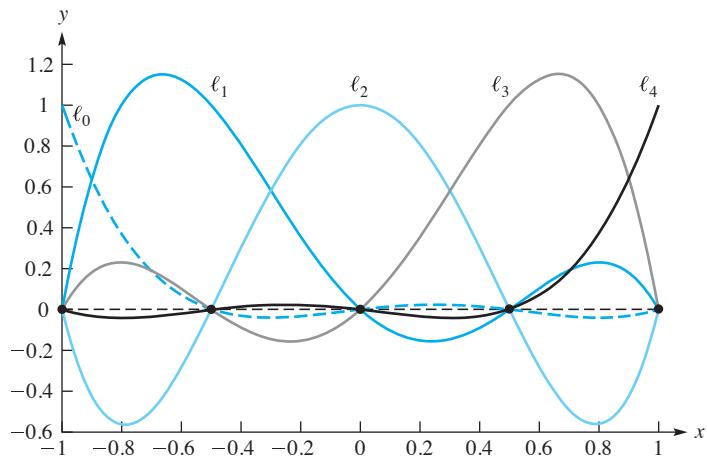


FIGURA 4.1
Algunos de
los primeros
polinomios
cardinales
de
Lagrange

EJEMPLO 2 Escriba los polinomios cardinales adecuados al problema de interpolación de la siguiente tabla y dé la forma de Lagrange del polinomio de interpolación:

x	$\frac{1}{3}$	$\frac{1}{4}$	1
$f(x)$	2	-1	7

Solución Usando la ecuación (2) tenemos

$$\ell_0(x) = \frac{(x - \frac{1}{4})(x - 1)}{(\frac{1}{3} - \frac{1}{4})(\frac{1}{3} - 1)} = -18 \left(x - \frac{1}{4} \right) (x - 1)$$

$$\ell_1(x) = \frac{(x - \frac{1}{3})(x - 1)}{(\frac{1}{4} - \frac{1}{3})(\frac{1}{4} - 1)} = 16 \left(x - \frac{1}{3} \right) (x - 1)$$

$$\ell_2(x) = \frac{(x - \frac{1}{3})(x - \frac{1}{4})}{(1 - \frac{1}{3})(1 - \frac{1}{4})} = 2 \left(x - \frac{1}{3} \right) \left(x - \frac{1}{4} \right)$$

Por tanto, el polinomio de interpolación en la forma de Lagrange es

$$p_2(x) = -36 \left(x - \frac{1}{4} \right) (x - 1) - 16 \left(x - \frac{1}{3} \right) (x - 1) + 14 \left(x - \frac{1}{3} \right) \left(x - \frac{1}{4} \right)$$

Existencia de la interpolación de polinomios

La fórmula de interpolación de Lagrange demuestra la existencia de un polinomio de interpolación para cualquier tabla de valores. Hay otra forma constructiva de demostrar este hecho y conduce a una fórmula diferente.

Suponga que tenemos éxito en determinar un polinomio p que reproduce *parte* de la tabla. Suponga que $p(x_i) = y_i$ para $0 \leq i \leq k$. Vamos a intentar sumar a p otro término que permitirá al nuevo polinomio reproducir una entrada más en la tabla. Consideremos

$$p(x) + c(x - x_0)(x - x_1) \cdots (x - x_k)$$

donde c es una constante por determinar. Este es sin duda un polinomio. También reproduce los primeros k puntos en la tabla, ya que p mismo lo hace y la parte sumada toma el valor de 0 en cada uno de los puntos x_0, x_1, \dots, x_k . (Su forma se elige exactamente por esta razón.) Ahora ajustamos el parámetro c por lo que el nuevo polinomio toma el valor y_{k+1} en x_{k+1} . Imponiendo esta condición, obtenemos

$$p(x_{k+1}) + c(x_{k+1} - x_0)(x_{k+1} - x_1) \cdots (x_{k+1} - x_k) = y_{k+1}$$

El valor adecuado de c se puede obtener a partir de esta ecuación, ya que ninguno de los factores $x_{k+1} - x_i$, para $0 \leq i \leq k$, puede ser cero. Recuerde nuestra suposición original que las x_i son todas distintas.

Este análisis es un ejemplo de razonamiento inductivo. Hemos demostrado que el proceso se puede iniciar y que se puede continuar. Por tanto, se ha justificado parcialmente el siguiente enunciado formal:

TEOREMA 1

Teorema de la existencia de la interpolación polinomial

Si los puntos x_0, x_1, \dots, x_n son distintos, entonces para los valores reales arbitrarios y_0, y_1, \dots, y_n , hay un único polinomio p de grado a lo más n tal que $p(x_i) = y_i$ para $0 \leq i \leq n$.

Aún se deben establecer dos partes de este enunciado formal. Primera, el grado del polinomio aumenta a lo más en 1 en cada paso del argumento inductivo. Al inicio, el grado era a lo más 0, por lo que al final, el grado es a lo más n .

Segunda, establecemos la unicidad del polinomio p . Suponga que otro polinomio q afirma que cumple lo que p ; es decir, q es también de grado a lo más n y satisface $q(x_i) = y_i$ para $0 \leq i \leq n$. Entonces el polinomio $p - q$ es de grado a lo más n y toma el valor 0 en x_0, x_1, \dots, x_n . Recuerde, sin embargo, que un polinomio *distinto de cero* de grado n puede tener a lo más n raíces. Concluimos que $p = q$, lo que establece la unicidad de p .

Interpolación polinomial: forma de Newton

En el ejemplo 2 encontramos la forma de Lagrange del polinomio de interpolación:

$$p_2(x) = -36\left(x - \frac{1}{4}\right)(x - 1) - 16\left(x - \frac{1}{3}\right)(x - 1) + 14\left(x - \frac{1}{3}\right)\left(x - \frac{1}{4}\right)$$

Esto se puede simplificar a

$$p_2(x) = -\frac{79}{6} + \frac{349}{6}x - 38x^2$$

Ahora aprenderemos que este polinomio se puede escribir en otra forma llamada forma de Newton anidada:

$$p_2(x) = 2 + \left(x - \frac{1}{3}\right) \left[36 + \left(x - \frac{1}{4}\right)(-38)\right]$$

Esto implica el menor número de operaciones aritméticas elementales y se recomienda para evaluar $p_2(x)$. No se puede sobrevalorar que las formas de Newton y Lagrange son exactamente dos deducciones diferentes para el mismo polinomio precisamente. La forma de Newton tiene la ventaja de fácil extensibilidad para acomodar datos adicionales.

El análisis anterior proporciona un método para construir un polinomio de interpolación. El método se conoce como **algoritmo de Newton** y el polinomio resultante es la forma del polinomio de interpolación de Newton.

EJEMPLO 3 Usando el algoritmo de Newton, encuentre el polinomio de interpolación de menor grado para esta tabla:

x	0	1	-1	2	-2
y	-5	-3	-15	39	-9

Solución En la construcción se presentarán cinco polinomios sucesivos; se etiquetan con p_0, p_1, p_2, p_3 y p_4 . El polinomio p_0 se define como

$$p_0(x) = -5$$

El polinomio p_1 tiene la forma

$$p_1(x) = p_0(x) + c(x - x_0) = -5 + c(x - 0)$$

La condición de interpolación impuesta en p_1 es que $p_1(1) = -3$. Por tanto, tenemos $-5 + c(1 - 0) = -3$. Por tanto, $c = 2$ y p_1 es

$$p_1(x) = -5 + 2x$$

El polinomio p_2 tiene la forma

$$p_2(x) = p_1(x) + c(x - x_0)(x - x_1) = -5 + 2x + cx(x - 1)$$

La condición de interpolación impuesta en p_2 es que $p_2(-1) = -15$. Por tanto, tenemos $-5 + 2(-1) + c(-1)(-1 - 1) = -15$. Esta produce $c = -4$, por lo que

$$p_2(x) = -5 + 2x - 4x(x - 1)$$

Los pasos restantes son similares y el resultado final es la forma del polinomio de interpolación de Newton:

$$p_4(x) = -5 + 2x - 4x(x - 1) + 8x(x - 1)(x + 1) + 3x(x - 1)(x + 1)(x - 2)$$



Más tarde desarrollaremos un mejor algoritmo para construir el polinomio de interpolación de Newton. No obstante, el método que acabamos de explicar es sistemático e implica muy poco cálculo. Una característica importante es que cada nuevo polinomio en el algoritmo se obtiene a partir de su antecesor al sumar un nuevo término. Por tanto, al terminar, el polinomio final presenta todos los polinomios anteriores como constituyentes.

Forma anidada

Antes de continuar, permítanos reescribir la forma del polinomio de interpolación de Newton para una evaluación eficiente.

EJEMPLO 4 Escriba el polinomio p_4 del ejemplo 3 en forma *anidada* y úselo para evaluar $p_4(3)$.

Solución Escribimos p_4 como

$$p_4(x) = -5 + x(2 + (x - 1)(-4 + (x + 1)(8 + (x - 2)3)))$$

Por tanto,

$$\begin{aligned} p_4(3) &= -5 + 3(2 + 2(-4 + 4(8 + 3))) \\ &= 241 \end{aligned}$$

Otra solución, también en forma anidada, es

$$p_4(x) = -5 + x(4 + x(-7 + x(2 + 3x)))$$

a partir de la cual obtenemos

$$p_4(3) = -5 + 3(4 + 3(-7 + 3(2 + 3 \cdot 3))) = 241$$

Esta forma se obtiene al expandir y factorizar sistemáticamente el polinomio original. Se conoce también como una **forma anidada** y su evaluación es por **multiplicación anidada**. ■

Para describir la multiplicación anidada de una manera formal (para que se pueda traducir en un código), considere un polinomio general en la forma de Newton. Podría ser

$$\begin{aligned} p(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots \\ &\quad + a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) \end{aligned}$$

La forma anidada de $p(x)$ es

$$\begin{aligned} p(x) &= a_0 + (x - x_0)(a_1 + (x - x_1)(a_2 + \cdots + (x - x_{n-1})a_n)) \cdots \\ &= (\cdots ((a_n(x - x_{n-1}) + a_{n-1})(x - x_{n-2}) + a_{n-2}) \cdots)(x - x_0) + a_0 \end{aligned}$$

El **polinomio de interpolación de Newton** se puede escribir en forma resumida como

$$p_n(x) = \sum_{i=0}^n a_i \prod_{j=0}^{i-1} (x - x_j) \tag{3}$$

Aquí $\prod_{j=0}^{-1} (x - x_j)$ se traduce como 1. También, podemos escribirlo como

$$p_n(x) = \sum_{i=0}^n a_i \pi_i(x)$$

donde

$$\pi_i(x) = \prod_{j=0}^{i-1} (x - x_j) \tag{4}$$

En la figura 4.2 se muestran algunos de los primeros polinomios de Newton: $\pi_0(x)$, $\pi_1(x)$, $\pi_2(x)$, $\pi_3(x)$, $\pi_4(x)$ y $\pi_5(x)$.

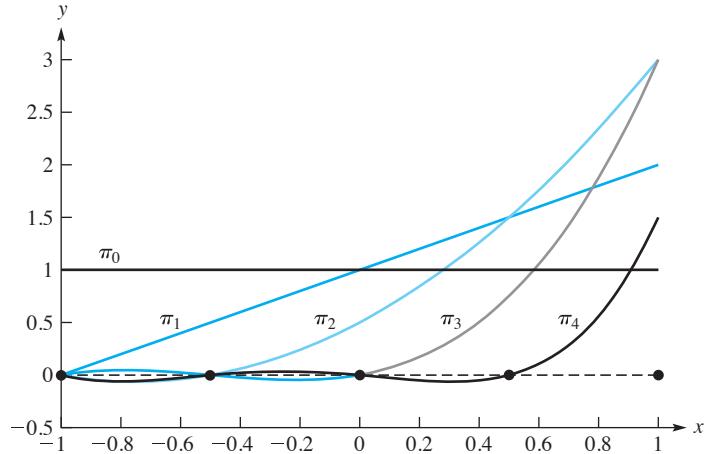


FIGURA 4.2
Algunos de los primeros polinomios de Newton

Al evaluar $p(t)$ para un valor numérico dado de t , naturalmente comenzamos con el paréntesis más interno, formando sucesivamente las cantidades siguientes:

$$\begin{aligned} v_0 &= a_n \\ v_1 &= v_0(t - x_{n-1}) + a_{n-1} \\ v_2 &= v_1(t - x_{n-2}) + a_{n-2} \\ &\vdots \\ v_n &= v_{n-1}(t - x_0) + a_0 \end{aligned}$$

La cantidad v_n es ahora $p(t)$. En el siguiente seudocódigo no se necesita una variable subindizada para v_i . Más bien, podemos escribir

```
integer i, n;  real t, v;  real array (ai)0:n, (xi)0:n
v ← an
for i = n - 1 to 0 step -1 do
    v ← v(t - xi) + ai
end for
```

Aquí, el arreglo $(a_i)_{0:n}$ contiene los coeficientes $n + 1$ de la forma del polinomio de interpolación de Newton (3) de grado a lo más n , y el arreglo $(x_i)_{0:n}$ contiene los $n + 1$ nodos x_i .

Cálculo de coeficientes a_i usando diferencias divididas

Ahora regresamos al problema de determinar los coeficientes a_0, a_1, \dots, a_n eficientemente. De nuevo iniciamos con una tabla de valores de una función f :

x	x_0	x_1	x_2	\cdots	x_n
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$	\cdots	$f(x_n)$

Los puntos x_0, x_1, \dots, x_n se suponen diferentes, pero no se ha hecho suposición acerca de su ubicación en la recta real.

Antes establecimos que para cada $n = 0, 1, \dots$, existe un polinomio único p_n tal que

- El grado de p_n es a lo más n .
- $p_n(x_i) = f(x_i)$ para $i = 0, 1, \dots, n$.

Se ha mostrado que p_n se puede expresar en la forma de Newton

$$\begin{aligned} p_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots \\ &\quad + a_n(x - x_0) \cdots (x - x_{n-1}) \end{aligned}$$

Una observación muy importante acerca de p_n es que los coeficientes a_0, a_1, \dots , no dependen de n . En otras palabras, p_n se obtiene de p_{n-1} sumando un término más, sin alterar los coeficientes que ya están en p_{n-1} . Este es porque comenzamos esperando que p_n pudiera expresarse en la forma

$$p_n(x) = p_{n-1}(x) + a_n(x - x_0) \cdots (x - x_{n-1})$$

y descubrimos que esto de hecho era posible.

Una forma de determinar sistemáticamente los coeficientes desconocidos a_0, a_1, \dots, a_n es hacer x igual a x_0, x_1, \dots, x_n en la forma de Newton (3) y a continuación escribir las ecuaciones resultantes:

$$\left\{ \begin{array}{l} f(x_0) = a_0 \\ f(x_1) = a_0 + a_1(x_1 - x_0) \\ f(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\ \text{etc.} \end{array} \right. \quad (5)$$

La forma compacta de las ecuaciones (5) es

$$f(x_k) = \sum_{i=0}^k a_i \prod_{j=0}^{i-1} (x_k - x_j) \quad (0 \leq k \leq n) \quad (6)$$

Las ecuaciones (5) se pueden resolver para las a_i correspondientes, iniciando con a_0 . Entonces vemos que a_0 depende de $f(x_0)$, que a_1 depende de $f(x_0)$ y $f(x_1)$, y así sucesivamente. En general, a_k depende de $f(x_0), f(x_1), \dots, f(x_n)$. En otras palabras, a_k depende de los valores de f en los nodos x_0, x_1, \dots, x_k . La notación tradicional es

$$a_k = f[x_0, x_1, \dots, x_k] \quad (7)$$

Esta ecuación define $f[x_0, x_1, \dots, x_k]$. La cantidad $f[x_0, x_1, \dots, x_k]$ se llama **diferencia dividida de orden k** para f . Observe también que los coeficientes a_0, a_1, \dots, a_k están determinados en *forma única* por el sistema (6). De hecho, no hay otra elección posible para a_0 que $a_0 = f(x_0)$. De manera similar, ahora no hay otra elección para a_1 que $[f(x_1) - a_0]/(x_1 - x_0)$ y así sucesivamente. Usando las ecuaciones (5), vemos que algunas de las primeras diferencias divididas se pueden escribir como

$$\begin{aligned} a_0 &= f(x_0) \\ a_1 &= \frac{f(x_1) - a_0}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\ a_2 &= \frac{f(x_2) - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} \end{aligned}$$

EJEMPLO 5 Para la tabla

x	1	-4	0
$f(x)$	3	13	-23

determine las cantidades $f[x_0]$, $f[x_0, x_1]$ y $f[x_0, x_1, x_2]$.

Solución Escribimos el sistema de ecuaciones (5) para este caso concreto:

$$\begin{cases} 3 = a_0 \\ 13 = a_0 + a_1(-5) \\ -23 = a_0 + a_1(-1) + a_2(4) \end{cases}$$

La solución es $a_0 = 3$, $a_1 = -2$ y $a_2 = 7$. Por tanto, para esta función, $f[1] = 3$, $f[1, -4] = -2$ y $f[1, -4, 0] = 7$. ■

Con esta nueva notación, la **forma de Newton del polinomio de interpolación** se convierte en

$$p_n(x) = \sum_{i=0}^n \left\{ f[x_0, x_1, \dots, x_i] \prod_{j=0}^{i-1} (x - x_j) \right\} \quad (8)$$

con la convención usual que $\prod_{j=0}^{-1} (x - x_j) = 1$. Observe que el coeficiente de x^n en p_n es $f[x_0, x_1, \dots, x_n]$, ya que el término x^n sólo se presenta en $\prod_{j=0}^{n-1} (x - x_j)$. Se deduce que si f es un polinomio de grado $\leq n-1$, entonces $f[x_0, x_1, \dots, x_n] = 0$.

Regresamos a la cuestión de cómo calcular las diferencias divididas requeridas $f[x_0, x_1, \dots, x_n]$. A partir del sistema (5) o (6), es evidente que este cálculo se puede ejecutar *recursivamente*. Sólo resolvemos la ecuación (6) para a_k como sigue:

$$f(x_k) = a_k \prod_{j=0}^{k-1} (x_k - x_j) + \sum_{i=0}^{k-1} a_i \prod_{j=0}^{i-1} (x_k - x_j)$$

y

$$a_k = \frac{f(x_k) - \sum_{i=0}^{k-1} a_i \prod_{j=0}^{i-1} (x_k - x_j)}{\prod_{j=0}^{k-1} (x_k - x_j)}$$

Usando la ecuación (7), tenemos

$$f[x_0, x_1, \dots, x_k] = \frac{f(x_k) - \sum_{i=0}^{k-1} f[x_0, x_1, \dots, x_i] \prod_{j=0}^{i-1} (x_k - x_j)}{\prod_{j=0}^{k-1} (x_k - x_j)} \quad (9)$$

■ ALGORITMO 1 Un algoritmo para calcular las diferencias divididas de f

- Sea $f[x_0] = f(x_0)$
 - Para $k = 1, 2, \dots, n$, calcule $f[x_0, x_1, \dots, x_n]$ por la ecuación (9).
- (10)

EJEMPLO 6 Usando el algoritmo (10), escriba las fórmulas para $f[x_0]$, $f[x_0, x_1]$, $f[x_0, x_1, x_2]$ y $f[x_0, x_1, x_2, x_3]$.

Solución

$$\begin{aligned}f[x_0] &= f(x_0) \\f[x_0, x_1] &= \frac{f(x_1) - f[x_0]}{x_1 - x_0} \\f[x_0, x_1, x_2] &= \frac{f(x_2) - f[x_0] - f[x_0, x_1](x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\f[x_0, x_1, x_2, x_3] &= \frac{f(x_3) - f[x_0] - f[x_0, x_1](x_3 - x_0) - f[x_0, x_1, x_2](x_3 - x_0)(x_3 - x_1)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)}\end{aligned}$$



El algoritmo (10) se programa fácilmente y se pueden calcular las diferencias divididas $f[x_0]$, $f[x_0, x_1]$, \dots , $f[x_0, x_1, \dots, x_n]$ con un costo de $\frac{1}{2}n(3n + 1)$ sumas, $(n - 1)(n - 2)$ multiplicaciones y n divisiones excluyendo las operaciones aritméticas de los índices. Ahora presentaremos un método más refinado para el cual el pseudocódigo requiere sólo tres expresiones y cuesta sólo $\frac{1}{2}n(n + 1)$ divisiones y $n(n + 1)$ sumas.

La esencia del nuevo método es el siguiente teorema notable:

■ TEOREMA 2

Propiedad recursiva de diferencias divididas

Las diferencias divididas obedecen la fórmula

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, x_2, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0} \quad (11)$$

Demostración

Ya que $f[x_0, x_1, \dots, x_k]$ fue definida igual al coeficiente a_k en la forma de Newton del polinomio de interpolación p_k de la ecuación (3), podemos decir que $f[x_0, x_1, \dots, x_k]$ es el coeficiente de x^k en el polinomio p_k de grado $\leq k$, que interpola a f en x_0, x_1, \dots, x_k . De manera similar, $f[x_1, x_2, \dots, x_k]$ es el coeficiente de x^{k-1} en el polinomio q de grado $\leq k - 1$, que interpola a f en x_1, x_2, \dots, x_k . Asimismo, $f[x_0, x_1, \dots, x_{k-1}]$ es el coeficiente de x^{k-1} en el polinomio p_{k-1} de grado $\leq k - 1$, que interpola a f en x_0, x_1, \dots, x_{k-1} . Los tres polinomios p_k , q y p_{k-1} están íntimamente relacionados. De hecho,

$$p_k(x) = q(x) + \frac{x - x_k}{x_k - x_0} [q(x) - p_{k-1}(x)] \quad (12)$$

Para establecer la ecuación (12), observe que el miembro derecho es un polinomio de grado a lo más k . Evalúelo en x_i , para $1 \leq i \leq k - 1$, dando como resultado $f(x_i)$:

$$\begin{aligned}q(x_i) + \frac{x_i - x_k}{x_k - x_0} [q(x_i) - p_{k-1}(x_i)] &= f(x_i) + \frac{x_i - x_k}{x_k - x_0} [f(x_i) - f(x_i)] \\&= f(x_i)\end{aligned}$$

De manera similar, al evaluar éste en x_0 y x_k se obtiene $f(x_0)$ y $f(x_k)$, respectivamente. Por la unicidad de los polinomios interpolados, el miembro derecho de la ecuación (12) debe ser $p_k(x)$ y se establece la ecuación (12).

Completando el argumento para justificar la ecuación (11), tomamos el coeficiente de x^k en los dos lados de la ecuación (12). Ello resulta en la (11). De hecho, vemos que $f[x_1, x_2, \dots, x_k]$ es el coeficiente de x^{k-1} en q , y $f[x_0, x_1, \dots, x_{k-1}]$ es el coeficiente de x^{k-1} en p_{k-1} . ■

Observe que $f[x_0, x_1, \dots, x_k]$ no se cambia si se permutan los nodos x_0, x_1, \dots, x_k ; así, por ejemplo, $f[x_0, x_1, x_2] = f[x_1, x_2, x_0]$. La razón es que $f[x_0, x_1, x_2]$ es el coeficiente de x^2 en el polinomio cuadrático interpolado f en x_0, x_1, x_2 , mientras que $f[x_1, x_2, x_0]$ es el coeficiente de x^2 en el polinomio cuadrático interpolado f en x_1, x_2, x_0 . Estos dos polinomios son, por supuesto, iguales. Un enunciado formal en el lenguaje matemático se expresa como:

■ TEOREMA 3

Teorema de la invarianza

La diferencia dividida $f[x_0, x_1, \dots, x_k]$ es invariante bajo todas las permutaciones de los argumentos x_0, x_1, \dots, x_k .

Puesto que las variables x_0, x_1, \dots, x_k y k son arbitrarias, la fórmula recursiva (11) también se puede escribir como

$$f[x_i, x_{i+1}, \dots, x_{j-1}, x_j] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_j] - f[x_i, x_{i+1}, \dots, x_{j-1}]}{x_j - x_i} \quad (13)$$

Las primeras tres diferencias divididas son, por tanto

$$\begin{aligned} f[x_i] &= f(x_i) \\ f[x_i, x_{i+1}] &= \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \\ f[x_i, x_{i+1}, x_{i+2}] &= \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} \end{aligned}$$

Usando la fórmula (13) podemos construir una tabla de diferencias divididas para una función f . Se acostumbra arreglarla de la forma siguiente (aquí $n = 3$):

x	$f[]$	$f[,]$	$f[, ,]$	$f[, , ,]$
x_0	$f[x_0]$			
x_1	$f[x_1]$	$f[x_0, x_1]$		
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
x_3	$f[x_3]$	$f[x_2, x_3]$		

En la tabla, los coeficientes a lo largo de la diagonal superior son los necesarios para constituir la forma de Newton del polinomio de interpolación (3).

EJEMPLO 7 Construya un diagrama de diferencia dividida para la función f dada en la siguiente tabla y escriba la forma de Newton del polinomio de interpolación.

x	1	$\frac{3}{2}$	0	2
$f(x)$	3	$\frac{13}{4}$	3	$\frac{5}{3}$

Solución La primera entrada es $f[x_0, x_1] = (\frac{13}{4} - 3) / (\frac{3}{2} - 1) = \frac{1}{2}$. Después de completar la columna 3, la primera entrada de la columna 4 es

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\frac{1}{6} - \frac{1}{2}}{0 - 1} = \frac{1}{3}$$

El diagrama completo es

x	$f[]$	$f[,]$	$f[, ,]$	$f[, , ,]$
1	3			
$\frac{3}{2}$	$\frac{13}{4}$	$\frac{1}{2}$	$\frac{1}{3}$	
0	3	$\frac{1}{6}$	$-\frac{5}{3}$	
2	$\frac{5}{3}$	$-\frac{2}{3}$		-2

Por tanto, obtenemos

$$p_3(x) = 3 + \frac{1}{2}(x - 1) + \frac{1}{3}(x - 1)(x - \frac{3}{2}) - 2(x - 1)(x - \frac{3}{2})x$$



Algoritmos y seudocódigo

Ahora veamos el siguiente algoritmo. Supongamos que una tabla para f está dada en los puntos x_0, x_1, \dots, x_n y que todas las diferencias divididas $a_{ij} \equiv f[x_i, x_{i+1}, \dots, x_j]$ deben ser calculadas. El siguiente seudocódigo logra esto:

```

integer  $i, j, n$ ; real array  $(a_{ij})_{0:n \times 0:n}, (x_i)_{0:n}$ 
for  $i = 0$  to  $n$  do
     $a_{i0} \leftarrow f(x_i)$ 
end for
for  $j = 1$  to  $n$  do
    for  $i = 0$  to  $n - j$  do
         $a_{ij} \leftarrow (a_{i+1, j-1} - a_{i, j-1}) / (x_{i+j} - x_i)$ 
    end for
end for

```

Observe que los coeficientes del polinomio de interpolación (3) se almacenan en el primer renglón del arreglo $(a_{ij})_{0:n \times 0:n}$.

Si las diferencias divididas se calculan sólo en la construcción de la forma de Newton del polinomio de interpolación

$$p_n(x) = \sum_{i=0}^n a_i \prod_{j=0}^{i-1} (x - x_j)$$

donde $a_i = f[x_0, x_1, \dots, x_i]$, no hay necesidad de almacenarlas todas. Sólo se requiere almacenar $f[x_0], f[x_0, x_1], \dots, f[x_0, x_1, \dots, x_n]$.

Cuando se usa un arreglo unidimensional $(a_i)_{0:n}$, las diferencias divididas se pueden sobreescribir cada vez a partir de la última ubicación de almacenamiento hacia atrás, por lo que, al final, sólo permanecen los coeficientes deseados. En este caso, la cantidad de cálculos es igual que en el

caso anterior, pero los requisitos de almacenamiento son menores. (¿Por qué?) Aquí se presenta un pseudocódigo para hacer esto:

```

integer  $i, j, n$ ; real array  $(a_i)_{0:n}, (x_i)_{0:n}$ 
for  $i = 0$  to  $n$  do
     $a_i \leftarrow f(x_i)$ 
end for
for  $j = 1$  to  $n$  do
    for  $i = n$  to  $j$  step  $-1$  do
         $a_i \leftarrow (a_i - a_{i-1}) / (x_i - x_{i-j})$ 
    end for
end for

```

Este algoritmo es más enredado y se invita al lector para comprobarlo, digamos, en el caso $n = 3$.

Para los experimentos numéricos sugeridos en los problemas de cómputo, los siguientes dos procedimientos deben ser satisfactorios. Al primero se le llama *Coef*. Requiere como entrada el número n y valores tabulados en los arreglos (x_i) y (y_i) . Recuerde que el número de puntos en la tabla es $n + 1$. El procedimiento entonces calcula los coeficientes requeridos en el polinomio de interpolación de Newton, almacenándolos en el arreglo (a_i) .

```

procedure Coef( $n, (x_i), (y_i), (a_i)$ )
integer  $i, j, n$ ; real array  $(x_i)_{0:n}, (y_i)_{0:n}, (a_i)_{0:n}$ 
for  $i = 0$  to  $n$  do
     $a_i \leftarrow y_i$ 
end for
for  $j = 1$  to  $n$  do
    for  $i = n$  to  $j$  step  $-1$  do
         $a_i \leftarrow (a_i - a_{i-1}) / (x_i - x_{i-j})$ 
    end for
end for
end procedure Coef

```

La segunda es la función *Eval*. Requiere como entrada el arreglo (x_i) de la tabla original y el arreglo (a_i) , que es la *salida* de *Coef*. El arreglo (a_i) contiene los coeficientes para la forma de Newton del polinomio de interpolación. Por último, como entrada, un solo valor real para t dada. Entonces la función regresa el valor del polinomio de interpolación en t .

```

real function Eval( $n, (x_i), (a_i), t$ )
integer  $i, n$ ; real  $t, temp$ ; real array  $(x_i)_{0:n}, (a_i)_{0:n}$ 
 $temp \leftarrow a_n$ 
for  $i = n - 1$  to  $0$  step  $-1$  do
     $temp \leftarrow (temp)(t - x_i) + a_i$ 
end for
 $Eval \leftarrow temp$ 
end function Eval

```

Ya que los coeficientes del polinomio de interpolación necesitan calcularse sólo una vez, llamamos primero a *Coef* y después todas las llamadas sucesivas para evaluar este polinomio se cumplen con *Eval*. Observe que sólo el argumento t se debe cambiar entre llamadas sucesivas a la función *Eval*.

EJEMPLO 8 Escriba un seudocódigo para la forma de Newton del polinomio de interpolación p para $\sin x$ en diez puntos equidistantes en el intervalo $[0, 1.6875]$. El código encuentra el valor máximo de $|\sin x - p(x)|$ sobre un fino conjunto de puntos igualmente espaciados en el mismo intervalo.

Solución Si tomamos diez puntos, incluidos los extremos del intervalo, entonces creamos nueve subintervalos, cada uno de longitud $h = 0.1875$. Los puntos son entonces $x_i = ih$ para $i = 0, 1, \dots, 9$. Después de obtener el polinomio, dividimos cada subintervalo en cuatro paneles y evaluamos $|\sin x - p(x)|$ en 37 puntos (llamado t en el seudocódigo). Éstos son $t_j = jh/4$ para $j = 0, 1, \dots, 36$. Aquí se presenta un adecuado programa principal en seudocódigo que llama a los procedimientos *Coef* y *Eval* antes dados:

```
program Prueba_Evaluación_Coeficientes
integer j, k, n, j_max;  real e, h, p, e_max, p_max, t_max,
real array (x_i)_{0:n}, (y_i)_{0:n}, (a_i)_{0:n}
n ← 9
h ← 1.6875/n
for k = 0 to n do
    x_k ← kh
    y_k ← sin(x_k)
end for
call Coef(n, (x_i), (y_i), (a_i))
output (a_i); e_max ← 0
for j = 0 to 4n do
    t ← jh/4
    p ← Eval(n, (x_i)_n, (a_i)_n, t)
    e ← |sin(t) - p|
    output j, t, p, e
    if e > e_max then
        j_max ← j; t_max ← t; p_max ← p; e_max ← e
    end if
end for
output j_max, t_max, p_max, e_max
end program Prueba_Evaluación_Coeficientes
```

El primer coeficiente en la forma de Newton del polinomio de interpolación es 0 (¿por qué?) y los otros varían en magnitud a partir de aproximadamente 0.99 a 0.18×10^{-5} . La desviación entre $\sin x$ y $p(x)$ es prácticamente cero en cada nodo de interpolación. (Debido a los errores de redondeo, no son exactamente cero.) De la salida de la computadora, el error más grande es $j_{\text{máx}} = 35$, donde $\sin(1.640625) \approx 0.9975631$ con un error de 1.19×10^{-7} . ■

Matriz de Vandermonde

Otra forma de ver la interpolación es que para un conjunto de $n + 1$ puntos de datos dado $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, queremos expresar una función de interpolación $f(x)$ como una combinación lineal de un conjunto de *funciones base* $\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_n$, de modo que

$$f(x) \approx c_0\varphi_0(x) + c_1\varphi_1(x) + c_2\varphi_2(x) + \cdots + c_n\varphi_n(x)$$

Aquí se determinarán los coeficientes $c_0, c_1, c_2, \dots, c_n$. Queremos que la función f interpole los datos (x_i, y_i) . Esto significa que tenemos ecuaciones lineales de la forma

$$f(x_i) = c_0\varphi_0(x_i) + c_1\varphi_1(x_i) + c_2\varphi_2(x_i) + \cdots + c_n\varphi_n(x_i) = y_i$$

para cada $i = 0, 1, 2, \dots, n$. Este es un sistema de ecuaciones lineales

$$\mathbf{Ac} = \mathbf{y}$$

Aquí, las entradas en la matriz de coeficientes \mathbf{A} están dadas por $a_{ij} = \varphi_j(x_i)$, que es el valor de la j -ésima función base evaluada en el i -ésimo punto de datos. El lado derecho del vector \mathbf{y} contiene los valores conocidos de datos y_i , y los componentes del vector \mathbf{c} son los coeficientes desconocidos c_i . Los sistemas de ecuaciones lineales se analizan en los capítulos 7 y 8.

Los polinomios son las funciones base más simples y comunes. La base natural para \mathbb{P}_n consta de los *monomios*

$$\varphi_0(x) = 1, \varphi_1(x) = x, \varphi_2(x) = x^2, \dots, \varphi_n(x) = x^n$$

La figura 4.3 muestra algunos de los primeros monomios: $1, x, x^2, x^3, x^4$ y x^5 .

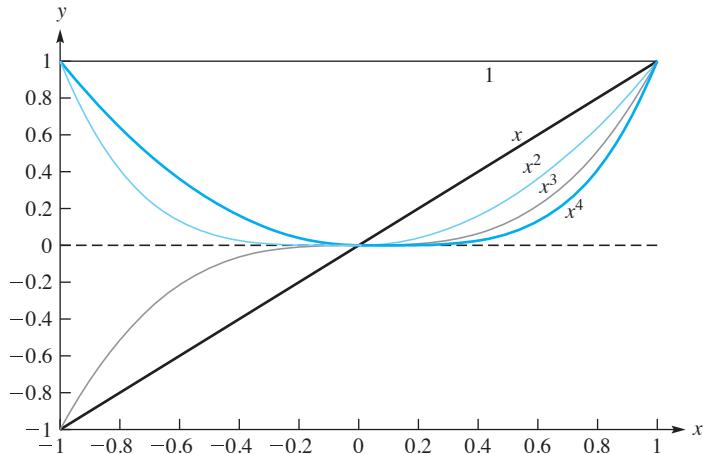


FIGURA 4.3
Algunos de
los primeros
monomios

Por consiguiente, un polinomio dado p_n tiene la forma

$$p_n(x) = c_0 + c_1x + c_2x^2 + \cdots + c_nx^n$$

El correspondiente sistema lineal $\mathbf{Ac} = \mathbf{y}$ tiene la forma

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

La matriz de coeficientes se llama *matriz de Vandermonde*. Se puede demostrar que esta matriz es no singular siempre que los puntos $x_0, x_1, x_2, \dots, x_n$ sean distintos. Por ello, podemos, en teoría, resolver el sistema para el polinomio de interpolación. Aunque la matriz de Vandermonde es no singular, está mal condicionada conforme n aumenta. Para n grande, los monomios son menos distinguibles entre sí, como se muestra en la figura 4.4. Por otra parte, en este caso las columnas de la matriz de Vandermonde serán casi linealmente dependientes. Los polinomios de alto orden con frecuencia oscilan fuertemente y son muy sensibles a pequeños cambios en los datos.

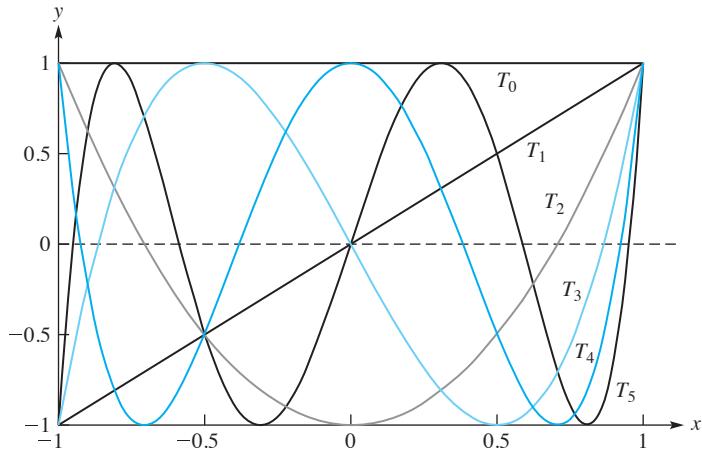


FIGURA 4.4
Algunos de
los primeros
polinomios de
Chebyshev

Como se mostró en las figuras 4.1, 4.2 y 4.3, se han analizado tres opciones para las funciones base: los polinomios cardinales de Lagrange $\ell_i(x)$, los polinomios de Newton $\pi_i(x)$ y los monomios. Resulta que hay mejores opciones para las funciones base; a saber, los polinomios de Chebyshev tienen características más deseables.

Los polinomios de Chebyshev juegan un importante papel en matemáticas debido a que tienen varias propiedades especiales, como la relación de recurrencia

$$\begin{cases} T_0(x) = 1, T_1(x) = x \\ T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x) \end{cases}$$

para $i = 2, 3, 4$ y así sucesivamente. Por tanto, los primeros cinco polinomios de Chebyshev son

$$\begin{aligned} T_0(x) &= 1, & T_1(x) &= x, & T_2(x) &= 2x^2 - 1, & T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1, & T_5(x) &= 16x^5 - 20x^3 + 5x \end{aligned}$$

Como se muestra en la figura 4.4, las curvas para estos polinomios son bastante diferentes entre sí. Los polinomios de Chebyshev con frecuencia se utilizan en el intervalo $[-1, 1]$. Con cambios de variable se pueden usar en cualquier intervalo, pero los resultados serán más complicados.

Una de las propiedades importantes de los polinomios de Chebyshev es la de igual oscilación. Observe en la figura 4.4 que los puntos finales sucesivos de los polinomios de Chebyshev son de igual magnitud y alternan en signo. Esta propiedad tiende a distribuir el error uniformemente cuando los polinomios de Chebyshev se usan como las funciones base. En la interpolación de polinomios para funciones continuas es particularmente ventajoso escoger como los puntos de interpolación las raíces o los puntos finales de un polinomio de Chebyshev. Esto causa que el máximo error en el intervalo de interpolación sea minimizado. Un ejemplo de esto se presenta en la sección 4.2. En la sección 12.2 analizamos los polinomios de Chebyshev con más detalle.

Interpolación inversa

Un proceso que se llama **interpolación inversa** con frecuencia se usa para aproximar una función inversa. Suponga que se han calculado los valores $y_i = f(x_i)$ en x_0, x_1, \dots, x_n . Usando la tabla

y	y_0	y_1	\cdots	y_n
x	x_0	x_1	\cdots	x_n

formamos el polinomio de interpolación

$$p(y) = \sum_{i=0}^n c_i \prod_{j=0}^{i-1} (y - y_j)$$

La relación original, $y = f(x)$, tiene una inversa bajo ciertas condiciones. Esta inversa se está aproximando por $x = p(y)$. Se pueden usar los procedimientos *Coef* y *Eval* para realizar la interpolación inversa al invertir los argumentos x y y en secuencia de llamado para *Coef*.

La interpolación inversa se puede usar para encontrar dónde una función dada f tiene una raíz o *cero*. Esto significa invertir la ecuación $f(x) = 0$. Proponemos hacer esto creando una tabla de valores $(f(x_i), x_i)$ e interpolando con un polinomio, p . Por tanto, $p(y) = x_i$. Los puntos x_i se deben elegir cerca de la raíz desconocida, r . La raíz aproximada está entonces dada por $r \approx p(0)$. Véase la figura 4.5 para un ejemplo de función $y = f(x)$ y su función inversa $x = g(y)$ con la raíz $r = g(0)$.

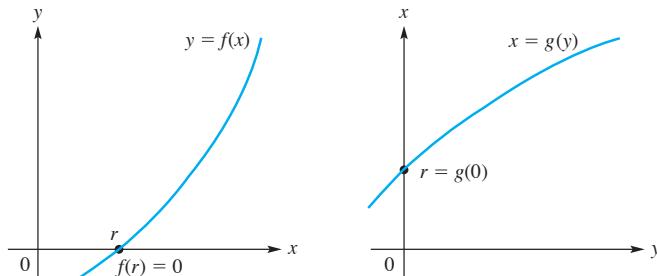


FIGURA 4.5

Función
 $y = f(x)$ y
función inversa
 $x = g(y)$

EJEMPLO 9 Para un caso concreto, sea la tabla de valores conocidos

y	-0.57892 00	-0.36263 70	-0.18491 60	-0.03406 42	0.09698 58
x	1.0	2.0	3.0	4.0	5.0

Encuentre el polinomio de interpolación inversa.

Solución Los nodos en este problema son los puntos en el renglón de la tabla con el rótulo y , y los valores de la función que se interpolan están en el renglón x . El polinomio resultante es

$$p(y) = 0.25y^4 + 1.2y^3 + 3.69y^2 + 7.39y + 4.24747\,0086$$

y $p(0) = 4.24747\,0086$. Sólo se muestra el último coeficiente con todos los dígitos con los que se realizó el cálculo, ya que es el único que se necesita para el problema en cuestión. ■

Interpolación polinomial con el algoritmo de Neville

Otro método para obtener un polinomio de interpolación a partir de una tabla de valores dada

x	x_0	x_1	\cdots	x_n
y	y_0	y_1	\cdots	y_n

fue dado por Neville. Se construye el polinomio en pasos, igual que lo hace el algoritmo de Newton. Los polinomios constituyentes tienen propiedades de interpolación propias.

Sea $P_{a, b, \dots, s}(x)$ el polinomio de interpolación de los datos dados en una secuencia de nodos x_a, x_b, \dots, x_s . Iniciamos con polinomios constantes $p_i(x) = f(x_i)$. Seleccionando dos nodos x_i y x_j con $i > j$, definimos recursivamente

$$P_{u, \dots, v}(x) = \left(\frac{x - x_j}{x_i - x_j} \right) P_{u, \dots, j-1, j+1, \dots, v}(x) + \left(\frac{x_i - x}{x_i - x_j} \right) P_{u, \dots, i-1, i+1, \dots, v}(x)$$

Usando esta fórmula repetidamente podemos crear un arreglo de polinomios:

x_0	$P_0(x)$					
x_1	$P_1(x)$	$P_{0,1}(x)$				
x_2	$P_2(x)$	$P_{1,2}(x)$	$P_{0,1,2}(x)$			
x_3	$P_3(x)$	$P_{2,3}(x)$	$P_{1,2,3}(x)$	$P_{0,1,2,3}(x)$		
x_4	$P_4(x)$	$P_{3,4}(x)$	$P_{2,3,4}(x)$	$P_{1,2,3,4}(x)$	$P_{0,1,2,3,4}(x)$	

Aquí, cada polinomio sucesivo se puede determinar a partir de dos polinomios adyacentes en la columna anterior.

Podemos simplificar la notación haciendo

$$S_{ij}(x) = P_{i-j, i-j+1, \dots, i-1, i}(x)$$

donde $S_{ij}(x)$ para $i \leq j$ denota al polinomio de interpolación de grado j en los $j+1$ nodos $x_{i-j}, x_{i-j+1}, \dots, x_{i-1}, x_i$. Ahora podemos reescribir la relación de recurrencia anterior como

$$S_{ij}(x) = \left(\frac{x - x_{i-j}}{x_i - x_{i-j}} \right) S_{i,j-1}(x) + \left(\frac{x_i - x}{x_i - x_{i-j}} \right) S_{i-1,j-1}(x)$$

Así, la representación del arreglo será

x_0	$S_{00}(x)$					
x_1	$S_{10}(x)$	$S_{11}(x)$				
x_2	$S_{20}(x)$	$S_{21}(x)$	$S_{22}(x)$			
x_3	$S_{30}(x)$	$S_{31}(x)$	$S_{32}(x)$	$S_{33}(x)$		
x_4	$S_{40}(x)$	$S_{41}(x)$	$S_{42}(x)$	$S_{43}(x)$	$S_{44}(x)$	

Para probar algunos resultados teóricos, cambiamos la notación haciendo el subíndice igual al grado del polinomio. Al inicio, definimos polinomios constantes (es decir, polinomios de grado 0) como $P_i^0(x) = y_i$ para $0 \leq i \leq n$. Entonces definimos

$$P_i^j(x) = \left(\frac{x - x_{i-j}}{x_i - x_{i-j}} \right) P_i^{j-1}(x) + \left(\frac{x_i - x}{x_i - x_{i-j}} \right) P_{i-1}^{j-1}(x) \quad (14)$$

En esta ecuación, los superíndices son simples índices, no exponentes. El rango de j es $1 \leq j \leq n$, mientras que el de i es $j \leq i \leq n$. La fórmula (14) la veremos de nuevo, en forma ligeramente distinta, en la teoría de splines B en la sección 9.3.

Las propiedades de interpolación de estos polinomios se presentan en el siguiente resultado.

■ TEOREMA 4

Propiedades de interpolación

Los polinomios P_i^j definidos antes se interpolan como se muestra a continuación

$$P_i^j(x_k) = y_k \quad (0 \leq i - j \leq k \leq i \leq n) \quad (15)$$

Demostración Usamos inducción en j . Cuando $j = 0$, la expresión en la ecuación (15) se escribe como

$$P_i^0(x_k) = y_k \quad (0 \leq i \leq k \leq i \leq n)$$

En otras palabras, $P_i^0(x_i) = y_i$, que es verdadera por la definición de P_i^0 .

Ahora suponga, como una hipótesis de inducción, que para alguna $j \geq i$,

$$P_i^{j-1}(x_k) = y_k \quad (0 \leq i - j + 1 \leq k \leq i \leq n)$$

Para probar el caso siguiente en la ecuación (15) comenzaremos por comprobar los dos casos extremos para k , a saber, $k = i - j$ y $k = i$. Tenemos, por la ecuación (14),

$$\begin{aligned} P_i^j(x_{i-j}) &= \left(\frac{x_i - x_{i-j}}{x_i - x_{i-j}} \right) P_{i-1}^{j-1}(x_{i-j}) \\ &= P_{i-1}^{j-1}(x_{i-j}) = y_{i-j} \end{aligned}$$

La última igualdad se justifica por la hipótesis de inducción. Es necesario observar que $0 \leq i - 1 - j + 1 \leq i - j \leq i - 1 \leq n$. De la misma manera, calcule

$$\begin{aligned} P_i^j(x_i) &= \left(\frac{x_i - x_{i-j}}{x_i - x_{i-j}} \right) P_i^{j-1}(x_i) \\ &= P_i^{j-1}(x_i) = y_i \end{aligned}$$

Aquí, usando la hipótesis de inducción, observe que $0 \leq i - j + 1 \leq i \leq i \leq n$.

Ahora sea $i - j < k < i$. Entonces

$$P_i^j(x_k) = \left(\frac{x_k - x_{i-j}}{x_i - x_{i-j}} \right) P_i^{j-1}(x_k) + \left(\frac{x_i - x_k}{x_i - x_{i-j}} \right) P_{i-1}^{j-1}(x_k)$$

En esta ecuación, $P_i^{j-1}(x_k) = y_k$ por la hipótesis de inducción, puesto que $0 \leq i-j+1 \leq k \leq i \leq n$. De la misma manera, $P_{i-1}^{j-1}(x_k) = y_k$, ya que $0 \leq i-1-j+1 \leq k \leq i-1 \leq n$. Por tanto, tenemos

$$P_i^j(x_k) = \left(\frac{x_k - x_{i-j}}{x_i - x_{i-j}} \right) y_k + \left(\frac{x_i - x_k}{x_i - x_{i-j}} \right) y_k = y_k \quad \blacksquare$$

Se tiene un algoritmo en pseudocódigo para evaluar $P_0^n(t)$ cuando se da una tabla de valores:

```

integer  $i, j, n$ ; real array  $(x_i)_{0:n}, (y_i)_{0:n}, (S_{ij})_{0:n \times 0:n}$ 
for  $i = 0$  to  $n$ 
     $S_{i0} \leftarrow y_i$ 
end for
for  $j = 1$  to  $n$ 
    for  $i = j$  to  $n$ 
         $S_{ij} \leftarrow [(t - x_{i-j})S_{i,j-1} + (x_i - t)S_{i-1,j-1}] / (x_i - x_{i-j})$ 
    end for
end for
return  $S_{0n}$ 
```

Comenzaremos el algoritmo al encontrar el nodo más cercano al punto t en el que se hace la evaluación. En general, la interpolación es más exacta cuando se hace esto.

Interpolación de funciones de dos variables

Los métodos que se han analizado para interpolar funciones de una variable mediante polinomios se amplían en *algunos* casos de funciones de dos o más variables. Un caso importante ocurre cuando una función $(x, y) \mapsto f(x, y)$ se aproxima en un rectángulo. Esto conduce a lo que se conoce como **interpolación de producto tensorial**. Suponga que el rectángulo es el producto cartesiano de dos intervalos: $[a, b] \times [\alpha, \beta]$. Es decir, las variables x y y corren a lo largo de los intervalos $[a, b]$ y $[\alpha, \beta]$, respectivamente. Escoja n nodos x_i en $[a, b]$ y defina los *polinomios de Lagrange*

$$\ell_i(x) = \prod_{\substack{j \neq i \\ j=1}}^n \frac{x - x_j}{x_i - x_j} \quad (1 \leq i \leq n)$$

De manera similar, elegimos m nodos y_i en $[\alpha, \beta]$ y definimos

$$\bar{\ell}_i(y) = \prod_{\substack{j \neq i \\ j=1}}^m \frac{y - y_j}{y_i - y_j} \quad (1 \leq i \leq m)$$

Entonces la función

$$P(x, y) = \sum_{i=1}^n \sum_{j=1}^m f(x_i, y_j) \ell_i(x) \bar{\ell}_j(y)$$

es un polinomio en dos variables que interpola f en los *puntos de la malla* (x_i, y_j) . Hay nm de esos puntos de interpolación. La demostración de la propiedad de interpolación es bastante simple debi-

do a que $\ell_i(x_q) = \delta_{iq}$ y $\bar{\ell}_j(y_p) = \delta_{jp}$. Por consiguiente,

$$\begin{aligned} P(x_q, y_p) &= \sum_{i=1}^n \sum_{j=1}^m f(x_i, y_j) \ell_i(x_q) \bar{\ell}_j(y_p) \\ &= \sum_{i=1}^n \sum_{j=1}^m f(x_i, y_j) \delta_{iq} \delta_{jp} = f(x_q, y_p) \end{aligned}$$

El mismo procedimiento se puede usar con splines interpolantes (o de hecho con cualquier otro tipo de función).

Resumen

(1) La forma de Lagrange del polinomio de interpolación es

$$p_n(x) = \sum_{i=0}^n \ell_i(x) f(x_i)$$

con **polinomios cardinales**

$$\ell_i(x) = \prod_{\substack{j \neq i \\ j=0}}^n \left(\frac{x - x_j}{x_i - x_j} \right) \quad (0 \leq i \leq n)$$

que obedecen la **ecuación de la delta de Kronecker**

$$\ell_i(x_j) = \delta_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$$

(2) La forma de Newton del polinomio de interpolación es

$$p_n(x) = \sum_{i=0}^n a_i \prod_{j=0}^{i-1} (x - x_j)$$

con **diferencias divididas**

$$a_i = f[x_0, x_1, \dots, x_i] = \frac{f[x_1, x_2, \dots, x_i] - f[x_0, x_1, \dots, x_{i-1}]}{x_i - x_0}$$

Estas son dos formas diferentes del único polinomio p de grado n que interpola una tabla de $n + 1$ pares de puntos $(x_i, f(x_i))$ para $0 \leq i \leq n$.

(3) Podemos ejemplificar esto con una pequeña tabla para $n = 2$:

x	x_0	x_1	x_2
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$

El polinomio de interpolación es

$$\begin{aligned} p_2(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \\ &\quad + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \end{aligned}$$

Obviamente, $p_2(x_0) = f(x_0)$, $p_2(x_1) = f(x_1)$ y $p_2(x_2) = f(x_2)$. Ahora, formamos la tabla de diferencias divididas:

x_0	$f(x_0)$	$f[x_0, x_1]$	
x_1	$f(x_1)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$
x_2	$f(x_2)$		

Usando las entradas de diferencias divididas de la diagonal superior, tenemos

$$p_n(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

De nuevo, se puede mostrar fácilmente que $p_2(x_0) = f(x_0)$, $p_2(x_1) = f(x_1)$ y $p_2(x_2) = f(x_2)$.

(4) Podemos usar interpolación polinomial inversa para encontrar un valor aproximado de una raíz r de la ecuación $f(x) = 0$ a partir de una tabla de valores (x_i, y_i) para $1 \leq i \leq n$. Aquí estamos suponiendo que los valores de la tabla están en la vecindad de este cero de la función f . Invirtiendo la tabla de valores, usamos la tabla inversa de valores (y_i, x_i) para determinar el polinomio de interpolación llamado $p_n(y)$. Ahora evaluando éste en 0, encontramos un valor que aproxime el cero deseado, a saber, $r \approx p_n(0)$ y $f(p_n(0)) \approx f(r) = 0$.

(5) Otros métodos avanzados de interpolación polinomial que se analizaron son el **algoritmo de Neville** y la **interpolación de una función de dos variables**.

Problemas 4.1

1. Use el proceso de interpolación de Lagrange para obtener un polinomio de menor grado que tome estos valores:

x	0	2	3	4
y	7	11	28	63

2. (Continuación) Arregle de nuevo los puntos de la tabla del problema anterior y encuentre la forma de Newton del polinomio de interpolación. Muestre que los polinomios obtenidos son idénticos, aunque sus formas pueden diferir.
3. Para los cuatro nodos de interpolación $-1, 1, 3, 4$, ¿cuáles son las funciones ℓ_i (2) requeridas en el procedimiento de interpolación de Lagrange? Dibuje las gráficas de estas cuatro funciones para mostrar sus propiedades esenciales.
4. Compruebe que los polinomios

$$p(x) = 5x^3 - 27x^2 + 45x - 21, \quad q(x) = x^4 - 5x^3 + 8x^2 - 5x + 3$$

interpolan los datos

x	1	2	3	4
y	2	1	6	47

y explique por qué esto no viola la parte de unicidad del teorema de existencia del polinomio de interpolación.

- 5.** Compruebe que los polinomios

$$p(x) = 3 + 2(x - 1) + 4(x - 1)(x + 2), \quad q(x) = 4x^2 + 6x - 7$$

son los dos polinomios de interpolación para la siguiente tabla y explique por qué esto no viola la parte de unicidad del teorema de existencia del polinomio de interpolación.

x	1	-2	0
y	3	-3	-7

- 6.** Halle el polinomio p de menor grado que toma estos valores: $p(0) = 2$, $p(2) = 4$, $p(3) = -4$, $p(5) = 82$. Use diferencias divididas para obtener el polinomio correcto. *No* es necesario escribir el polinomio en la forma estándar $a_0 + a_1x + a_2x^2 + \dots$.
- 7.** Complete las siguientes tablas de diferencias divididas y úselas para obtener polinomios de grado 3 que interpolan los valores indicados de la función:

a.

x	$f[]$	$f[,]$	$f[, ,]$	$f[, , ,]$
-1	2			
1	-4		2	
3	6			
5	10	2		

b.

x	$f[]$	$f[,]$	$f[, ,]$	$f[, , ,]$
-1	2			
1	-4			
3	46			
4	99.5	53.5		

Escriba los polinomios finales en una forma de calcular más eficiente.

- 8.** Encuentre un polinomio de interpolación para esta tabla:

x	1	2	2.5	3	4
y	-1	- $\frac{1}{3}$	$\frac{3}{32}$	$\frac{4}{3}$	25

- 9.** Dados los datos

x	0	1	2	4	6
$f(x)$	1	9	23	93	259

realice lo siguiente.

- a.** Construya la tabla de diferencias divididas.

- b.** Usando el polinomio de interpolación de Newton, encuentre una aproximación a $f(4.2)$.

Sugerencia: use polinomios que empiecen con 9 y que impliquen factores $(x - 1)$.

- 10. a.** Construya el polinomio de interpolación de Newton para los datos que se muestran.

x	0	2	3	4
y	7	11	28	63

b. Sin simplificarlo, escriba el polinomio obtenido en forma anidada para fácil evaluación.

- 11.** A partir de datos del censo, la población aproximada de Estados Unidos era de 150.7 millones en 1950, 179.3 millones en 1960, 203.3 millones en 1970, 226.5 millones en 1980 y 249.6 millones en 1990. Usando el polinomio de interpolación de Newton para estos datos, encuentre un valor aproximado para la población en 2000. Luego use el polinomio para calcular la población en 1920 a partir de estos datos. ¿Qué conclusión debe sacarse?

- 12.** El polinomio $p(x) = x^4 - x^3 + x^2 - x + 1$ tiene los siguientes valores:

x	-2	-1	0	1	2	3
$p(x)$	31	5	1	1	11	61

Encuentre un polinomio q que tome estos valores:

x	-2	-1	0	1	2	3
$q(x)$	31	5	1	1	11	30

Sugerencia: esto se puede hacer con poco trabajo.

- 13.** Use el método de diferencias divididas para obtener un polinomio de menor grado que se ajuste a los valores que se muestran.

a.

x	0	1	2	-1	3
y	-1	-1	-1	-7	5

b.

x	1	3	-2	4	5
y	2	6	-1	-4	2

- 14.** Encuentre el polinomio de interpolación para estos datos:

x	1.0	2.0	2.5	3.0	4.0
$f(x)$	-1.5	-0.5	0.0	0.5	1.5

- 15.** Se sospecha que la tabla

x	-2	-1	0	1	2	3
y	1	4	11	16	13	-4

provine de un polinomio cúbico. ¿Cómo se puede probar? Explique.

- 16.** Existe un único polinomio $p(x)$ de grado 2 o menor tal que $p(0) = 0$, $p(1) = 1$ y $p'(\alpha) = 2$ para cualquier valor de α entre 0 y 1 (inclusive) excepto un valor de α , digamos, α_0 . Determine α_0 , y dé este polinomio para $\alpha \neq \alpha_0$.

- 17.** Usando dos métodos, determine el polinomio de grado 2 o menor cuya gráfica pasa por los puntos $(0, 1.1)$, $(1, 2)$, y $(2, 4.2)$. Compruebe que llega al mismo resultado.

- 18.** Desarrolle la tabla de diferencias divididas a partir de los datos dados. Escriba el polinomio de interpolación y arréglo de nuevo para un cálculo rápido sin simplificación.

x	0	1	3	2	5
$f(x)$	2	1	5	6	-183

Punto de control: $f[1, 3, 2, 5] = -7$.

^a19. Sea $f(x) = x^3 + 2x^2 + x + 1$. Encuentre el polinomio de grado 4 que interpola los valores de f en $x = -2, -1, 0, 1, 2$. Encuentre el polinomio de grado 2 que interpola los valores de f en $x = -1, 0, 1$.

20. Sin usar una tabla de diferencias divididas, deduzca y simplifique el polinomio de menor grado que toma estos valores:

x	-2	-1	0	1	2
y	2	14	4	2	2

21. (Continuación) Encuentre un polinomio que tome los valores mostrados en el problema anterior y que tiene en $x = 3$ el valor 10. *Sugerencia:* sume un polinomio conveniente a $p(x)$ en el problema anterior.

^a22. Encuentre un polinomio de menor grado que tome estos valores:

x	1.73	1.82	2.61	5.22	8.26
y	0	0	7.8	0	0

Sugerencia: arregle de nuevo la tabla de modo que el valor distinto de cero de y esté en la última entrada o piense en una mejor forma.

23. Forme una tabla de diferencias divididas para la tabla siguiente y explique lo que ocurre.

x	1	2	3	1
y	3	5	5	7

24. La interpolación simple del polinomio en dos dimensiones no siempre es posible. Por ejemplo, suponga que los datos siguientes se representaron usando un polinomio de primer grado en x y en y , $p(t) = a + bt + ct$, donde $t = (x, y)$:

t	(1, 1)	(3, 2)	(5, 3)
$f(t)$	3	2	6

Demuestre que esto no es posible.

25. Considere una función $f(x)$ tal que $f(2) = 1.5713, f(3) = 1.5719, f(5) = 1.5738$ y $f(6) = 1.5751$. Calcule $f(4)$ usando un polinomio de interpolación de segundo grado y uno de tercer grado. Redondee los resultados finales a cuatro lugares decimales. ¿Existe alguna ventaja aquí al usar un polinomio de tercer grado?

26. Use interpolación inversa para encontrar un valor aproximado de x tal que $f(x) = 0$ a partir de la siguiente tabla de valores para f . Analice lo que ocurre y saque una conclusión.

x	-2	-1	1	2	3
$f(x)$	-31	5	1	11	61

^a27. Encuentre un polinomio $p(x)$ de grado a lo más 3 tal que $p(0) = 1, p(1) = 0, p'(0) = 0$ y $p'(-1) = -1$.

^a28. Usando una tabla de logaritmos, obtenemos los siguientes valores de $\log x$ que se indican en los puntos tabulares:

x	1	1.5	2	3	3.5	4
$\log x$	0	0.17609	0.30103	0.47712	0.54407	0.60206

Forme una tabla de diferencias divididas a partir de estos valores. Interpole para $\log 2.4$ y $\log 1.2$ usando polinomios de interpolación de tercer grado en la forma de Newton.

29. Muestre que las diferencias divididas son mapeos lineales; es decir,

$$(\alpha f + \beta g)[x_0, x_1, \dots, x_n] = \alpha f[x_0, x_1, \dots, x_n] + \beta g[x_0, x_1, \dots, x_n]$$

Sugerencia: use inducción.

30. Muestre que otra forma para el polinomio p_n de grado a lo más n que toma valores y_0, y_1, \dots, y_n en las abscisas x_0, x_1, \dots, x_n es

$$\sum_{i=0}^n f[x_n, x_{n-1}, \dots, x_{n-i}] \prod_{j=0}^{i-1} (x - x_{n-j})$$

31. Use la unicidad del polinomio de interpolación para comprobar que

$$\sum_{i=0}^n f(x_i) \ell_i(x) = \sum_{i=0}^n f[x_0, x_1, \dots, x_i] \prod_{j=0}^{i-1} (x - x_j)$$

32. (Continuación) Muestre que la siguiente fórmula explícita es válida para diferencias divididas:

$$f[x_0, x_1, \dots, x_n] = \sum_{i=0}^n f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n (x_i - x_j)^{-1}$$

Sugerencia: si dos polinomios son iguales, los coeficientes para cada x^n son iguales.

33. Compruebe directamente que

$$\sum_{i=0}^n \ell_i(x) = 1$$

para el caso $n = 1$. Después establezca el resultado para valores arbitrarios de n .

34. Escriba la forma de Lagrange (1) del polinomio de interpolación de grado a lo más 2 que intercala $f(x)$ en x_0, x_1 y x_2 , donde $x_0 < x_1 < x_2$.

35. (Continuación) Escriba la forma de Newton del polinomio de interpolación $p_2(x)$ y muestre que es equivalente a la forma de Lagrange.

36. (Continuación) Muestre directamente que

$$p_2''(x) = 2f[x_0, x_1, x_2]$$

37. (Continuación) Muestre directamente para espaciamiento uniforme $h = x_1 - x_0 = x_2 - x_1$ que

$$f[x_0, x_1] = \frac{\Delta f_0}{h} \quad y \quad f[x_0, x_1, x_2] = \frac{\Delta^2 f_0}{2h^2}$$

donde $\Delta f_i = f_{i+1} - f_i$, $\Delta^2 f_i = \Delta f_{i+1} - \Delta f_i$ y $f_i = f(x_i)$.

38. (Continuación) Establezca la forma de **diferencia hacia adelante de Newton** del polinomio de interpolación con espaciamiento uniforme

$$p_2(x) = f_0 + \binom{s}{1} \Delta f_0 + \binom{s}{2} \Delta^2 f_0$$

donde $x = x_0 + sh$. Aquí, $\binom{s}{m}$ es el coeficiente binomial $[s!]/[(s-m)!m!]$ y $s! / (s-m)! = s(s-1)(s-2) \cdots (s-m+1)$ ya que s puede ser cualquier número real y $m!$ tiene la definición usual puesto que m es un entero.

- “39.”** (Continuación) De la siguiente tabla de valores de $\ln x$, interpole para obtener $\ln 2.352$ y $\ln 2.387$ usando la forma de diferencia hacia adelante de Newton del polinomio de interpolación:

x	$f(x)$	Δf	$\Delta^2 f$
2.35	0.85442	0.00424	
2.36	0.85866	0.00423	-0.00001
2.37	0.86289	0.00421	-0.00002
2.38	0.86710	0.00419	-0.00002
2.39	0.87129		

Usando los valores correctamente redondeados $\ln 2.352 \approx 0.85527$ y $\ln 2.387 \approx 0.87004$, muestre que la fórmula de diferencia hacia adelante es más exacta cerca de la parte superior de la tabla que lo que está cerca de la parte inferior.

- “40.”** Cuente el número de multiplicaciones, divisiones y sumas/restas en la generación de la tabla de diferencias divididas que tiene $n+1$ puntos.

- 41.** Compruebe directamente que para cualesquiera tres distintos puntos x_0, x_1 y x_2 ,

$$f[x_0, x_1, x_2] = f[x_2, x_0, x_1] = f[x_1, x_2, x_0]$$

Compare este argumento con el del libro.

- “42.”** Sea p un polinomio de grado n . ¿Qué es $p[x_0, x_1, \dots, x_{n+1}]$?

- 43.** Demuestre que si f es continuamente derivable en el intervalo $[x_0, x_1]$, entonces $f[x_0, x_1] = f'(c)$ para alguna c en (x_0, x_1) .

- 44.** Si f es un polinomio de grado n , muestre que en una tabla de diferencias divididas para f , la n -ésima columna tiene un solo valor constante, una columna que contiene entradas $f[x_i, x_{i+1}, \dots, x_{i+n}]$.

- “45.”** Determine si la siguiente afirmación es verdadera o falsa. Si x_0, x_1, \dots, x_n son diferentes, entonces para valores reales arbitrarios y_0, y_1, \dots, y_n hay un único polinomio p_{n+1} de grado $\leq n+1$ tal que $p_{n+1}(x_i) = y_i$ para toda $i = 0, 1, \dots, n$.

- 46.** Muestre que si una función g interpola la función f en x_0, x_1, \dots, x_{n-1} y h interpola f en x_1, x_2, \dots, x_n , entonces

$$g(x) + \frac{x_0 - x}{x_n - x_0} [g(x) - h(x)]$$

interpola f en x_0, x_1, \dots, x_n .

- 47. (Determinante de Vandermonde)** Usando $f_i = f(x_i)$, muestre lo siguiente:

$$\text{a. } f[x_0, x_1] = \begin{vmatrix} 1 & f_0 \\ 1 & f_1 \\ 1 & x_0 \\ 1 & x_1 \end{vmatrix}$$

$$\text{b. } f[x_0, x_1, x_2] = \begin{vmatrix} 1 & x_0 & f_0 \\ 1 & x_1 & f_1 \\ 1 & x_2 & f_2 \\ 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{vmatrix}$$

Problemas de cómputo 4.1

- 1.** Pruebe el procedimiento dado en el libro para determinar la forma de Newton del polinomio de interpolación. Por ejemplo, considere esta tabla:

x	1	2	3	-4	5
y	2	48	272	1182	2262

Encuentre el polinomio de interpolación y compruebe que $p(-1) = 12$.

- 2.** Encuentre el polinomio de grado 10 que interpola la función $\arctan x$ en 11 puntos igualmente espaciados en el intervalo $[1, 6]$. Imprima los coeficientes en la forma de Newton del polinomio. Calcule e imprima la diferencia entre el polinomio y la función en 33 puntos igualmente espaciados en el intervalo $[0, 8]$. ¿Qué conclusión se puede sacar?
- 3.** Escriba un programa sencillo usando el procedimiento *Coef* que interpola e^x usando un polinomio de grado 10 en $[0, 2]$ y después compare el polinomio para \exp en 100 puntos.
- 4.** Use como datos de entrada al procedimiento *Coef* la lluvia anual en su lugar de residencia para cada uno de los últimos 5 años. Usando la función *Eval*, prediga la lluvia para este año. ¿La respuesta es razonable?
- 5.** Una tabla de valores de una función f está dada en los puntos $x_i = i/10$ para $0 \leq i \leq 100$. Con el fin de obtener una gráfica de f con la ayuda de un trazador automático, los valores de f se requieren en los puntos $z_i = i/20$ para $0 \leq i \leq 200$. Escriba un procedimiento para hacer esto, usando un polinomio cúbico interpolado con nodos x_i, x_{i+1}, x_{i+2} y x_{i+3} , para calcular f en $\frac{1}{2}(x_{i+1} + x_{i+2})$. Para z_i y z_{199} , use el polinomio cúbico asociado con z_3 y z_{197} , respectivamente. Compare esta rutina con *Coef* para una función dada.
- 6.** Escriba rutinas análogas para *Coef* y *Eval* usando la forma de Lagrange del polinomio de interpolación. Pruebe en el ejemplo dado en esta sección en 20 puntos con $h/2$. ¿La forma de Lagrange tiene alguna ventaja sobre la forma de Newton?
- 7. (Continuación)** Diseñe y realice un experimento numérico para comparar la exactitud de las formas de los polinomios de interpolación de Newton y de Lagrange en valores de todo el intervalo $[x_0, x_n]$.
- 8.** Reescriba y pruebe las rutinas *Coef* y *Eval* para que el arreglo (a) no se use. *Sugerencia:* cuando los elementos en el arreglo (y) no son necesariamente grandes, almacene las diferencias divididas en sus lugares.
- 9.** Escriba un procedimiento para realizar interpolación inversa para resolver ecuaciones de la forma $f(x) = 0$. Pruebelo en el ejemplo de la introducción al inicio de este capítulo.

10. Para el ejemplo 8, compare los resultados de su código con los del libro. Hágalo nuevamente usando interpolación lineal basado en diez puntos equidistantes. ¿Cómo se comparan los errores en los puntos intermedios? Trace curvas para visualizar la diferencia entre interpolación lineal y una interpolación de polinomio de grado superior.
11. Use software matemático como Matlab, Maple o Mathematica para encontrar un polinomio de interpolación para los puntos $(0, 0), (1, 1), (2, 2.001), (3, 3), (4, 4), (5, 5)$. Evalúe el polinomio en el punto $x = 14$ o $x = 20$ para mostrar que ligeros errores de redondeo en los datos pueden conducir a resultados sospechosos en la extrapolación.
12. Use software matemático simbólico como Matlab, Maple o Mathematica para generar el polinomio de interpolación para los puntos de datos del ejemplo 3. Trace el polinomio y los puntos de datos.
13. (Continuación) Repita estas instrucciones usando el ejemplo 7.
14. Realice los detalles del ejemplo 8 escribiendo un programa de computadora que trace la gráfica de los puntos de datos y la curva para el polinomio de interpolación.
15. (Continuación) Repita las instrucciones del problema 14 en el ejemplo 9.
16. Usando software matemático, realice los detalles y compruebe los resultados del ejemplo de la introducción de este capítulo.
17. **(Interpolación de Padé)** Encuentre una función racional de la forma

$$g(x) = \frac{a + bx}{1 + cx}$$

que interpole la función $f(x) = \arctan(x)$ en los puntos $x_0 = 1, x_1 = 2$ y $x_2 = 3$. En los mismos ejes, trace las gráficas de f y g , usando líneas de guiones y puntos, respectivamente.

4.2 Errores en la interpolación polinomial

Cuando una función f es aproximada en un intervalo $[a, b]$ mediante un polinomio de interpolación p , la diferencia entre f y p será (teóricamente) cero en cada nodo de interpolación. Una expectativa natural es que la función estará bien aproximada en todos los puntos intermedios y que a medida que el número de nodos aumenta, esto será cada vez mejor.

En la historia de las matemáticas numéricas, ocurrió un severo choque cuando se demostró que esta expectativa estaba mal fundada. Por supuesto, si la función que se está approximando no es continua, entonces puede no coincidir en todas las $p(x)$ y $f(x)$ excepto en los nodos.

- EJEMPLO 1** Considere estos cinco puntos de datos: $(0, 8), (1, 12), (3, 2), (4, 6), (8, 0)$. Construya y trace el polinomio de interpolación usando los dos puntos exteriores. Repita este proceso sumando un punto adicional a la vez hasta que se incluyan todos los puntos. ¿Qué conclusiones puede sacar?

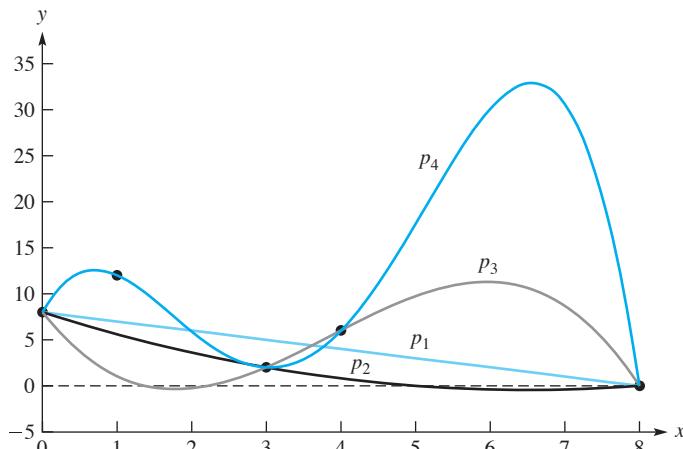


FIGURA 4.6
Polinomios de interpolación sobre datos puntuales

Solución El primer polinomio de interpolación es la recta entre los puntos más exteiros $(0, 8)$ y $(8, 0)$. Entonces agregamos los puntos $(3, 2)$, $(4, 5)$ y $(1, 12)$ en este orden y se traza una curva para cada punto adicional. Todos estos polinomios se muestran en la figura 4.6. Esperamos que una curva suave pase por estos puntos sin fluctuaciones amplias, pero esto no ocurre (¿por qué?) Puede parecer contradictorio, pero conforme agregamos más puntos, ¡la situación se vuelve peor en vez de mejor! La razón de esto proviene de la naturaleza de los polinomios de grado superior. Un polinomio de grado n tiene n ceros. Si todos estos puntos raíz son reales, entonces la curva corta el eje x n veces. La curva resultante debe hacer varias vueltas para que esto ocurra, dando como resultado fuertes oscilaciones. En el capítulo 9 analizamos el ajuste de puntos de datos con curvas spline. ■

Función de Dirichlet

Como un ejemplo patológico considere la así llamada **función de Dirichlet** f , definida igual a 1 en cada punto irracional y 0 en cada punto racional. Si elegimos nodos que son números racionales, entonces $p(x) \equiv 0$ y $f(x) - p(x) = 0$ para todos los valores racionales de x , pero $f(x) - p(x) = 1$ para todos los valores irracionales de x .

Sin embargo, si la función f es bien comportada, ¿no podemos suponer que las diferencias $|f(x) - p(x)|$ serán pequeñas cuando el número de nodos interpolados sea grande? La respuesta todavía es *no*, ¡aun para funciones que tienen derivadas continuas de todos los órdenes en el intervalo!

Función de Runge

Un ejemplo específico de este notable fenómeno se da al usar la función de **Runge**:

$$f(x) = (1 + x^2)^{-1} \quad (1)$$

en el intervalo $[-5, 5]$. Sea p_n el polinomio que interpola esta función en $n + 1$ puntos igualmente espaciados en el intervalo $[-5, 5]$, incluidos los puntos finales. Entonces

$$\lim_{n \rightarrow \infty} \max_{-5 \leq x \leq 5} |f(x) - p_n(x)| = \infty$$

Por tanto, el efecto de exigir el acuerdo de f y p_n en más y más puntos *aumenta* el error en los puntos de no nodos y el error realmente *¡aumenta* más allá de todos los límites!

La moraleja de este ejemplo, entonces, es que la interpolación del polinomio de alto grado con muchos nodos es una operación arriesgada; los polinomios resultantes pueden ser muy insatisfactorios como representaciones de funciones a menos que el conjunto de nodos se elija con mucho cuidado.

Usted puede fácilmente observar el fenómeno que acabamos de describir usando los sencillos códigos que ya desarrollamos en este capítulo. Lea el problema de cómputo 4.2.1 para una sugerencia de experimento numérico. En un estudio más avanzado de este tema, se mostraría que la divergencia de los polinomios con frecuencia se puede atribuir al hecho de que los nodos están igualmente espaciados. De nuevo, contrario a la intuición, los nodos igualmente distribuidos son a menudo una muy pobre elección en interpolación. Una mucho mejor elección para $n + 1$ nodos en $[-1, 1]$ es el conjunto de los **nodos de Chebyshev**:

$$x_i = \cos\left[\left(\frac{2i+1}{2n+2}\right)\pi\right] \quad (0 \leq i \leq n)$$

El conjunto correspondiente de nodos en un intervalo arbitrario $[a, b]$ se deduciría a partir de un mapeo lineal para obtener

$$x_i = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left[\left(\frac{2i+1}{2n+2}\right)\pi\right] \quad (0 \leq i \leq n)$$

Observe que estos nodos están numerados de derecha a izquierda. Puesto que la teoría no depende de cualquier ordenamiento particular de los nodos, esto no es problema.

Una gráfica simple ilustra mejor este fenómeno. De nuevo, considere la ecuación (1) en el intervalo $[-5, 5]$. Primero, elegimos nueve nodos igualmente espaciados y usamos las rutinas *Coef* y *Eval* con un trazador automático para graficar p_8 . Como se muestra en la figura 4.7, la curva resultante toma valores negativos, los que, ¡por supuesto, $f(x)$ no tiene! Agregando más nodos igualmente espaciados y, por tanto, obteniendo un polinomio de más alto grado, sólo empeora las cosas con salvajes oscilaciones. En la figura 4.8 se usan nueve nodos de Chebyshev y el polinomio resultante es una curva más suave. Sin embargo, los splines cúbicos (que se analizan en el capítulo 9) producen aún un mejor ajuste de curva.

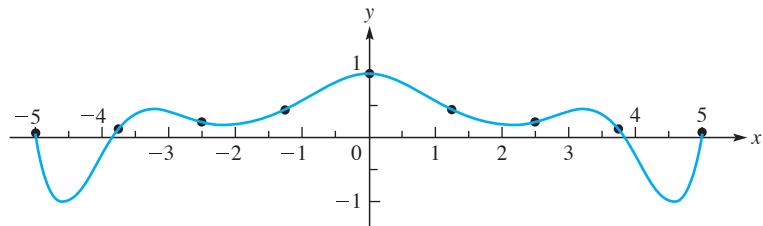


FIGURA 4.7
Polinomio de interpolación con nueve nodos igualmente espaciados

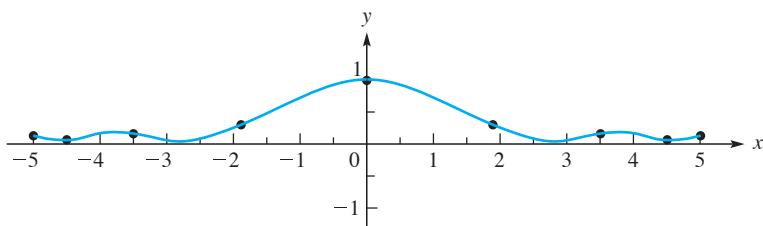
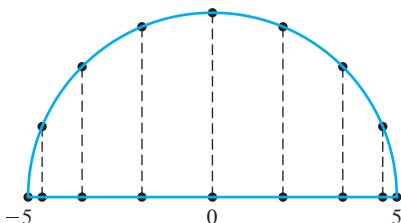


FIGURA 4.8
Polinomio de interpolación con nueve nodos de Chebyshev

FIGURA 4.9
Interpolación con puntos de Chebyshev



Los nodos de Chebyshev se obtienen al tomar puntos igualmente espaciados en un semicírculo y proyectándolos abajo en el eje horizontal, como se muestra en la figura 4.9.

Teoremas de errores de interpolación

Es posible evaluar los errores de interpolación usando una fórmula que implica la $(n + 1)$ ésima derivada de la función que se está interpolando. Este es el enunciado formal:

■ TEOREMA 1

Errores de interpolación I

Si p es el polinomio de grado a lo más n que interpola a f en los $n + 1$ nodos distintos x_0, x_1, \dots, x_n pertenecientes a un intervalo $[a, b]$ y si $f^{(n+1)}$ es continua, entonces para cada x en $[a, b]$, existe una ξ en (a, b) para la cual

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^n (x - x_i) \quad (2)$$

Demostración

Observe primero que la ecuación (2) es obviamente válida si x es uno de los nodos x_i ya que entonces los dos lados de la ecuación se reducen a cero. Si x no es un nodo, lo fijamos en el resto del análisis y se define

$$\begin{aligned} w(t) &= \prod_{i=0}^n (t - x_i) && \text{(polinomio en la variable } t\text{)} \\ c &= \frac{f(x) - p(x)}{w(x)} && \text{(constante)} \\ \varphi(t) &= f(t) - p(t) - cw(t) && \text{(función en la variable } t\text{)} \end{aligned} \quad (3)$$

Note que c está bien definida, ya que $w(x) \neq 0$ (x no es un nodo). Observe también que φ toma el valor 0 en los $n + 2$ puntos x_0, x_1, \dots, x_n y x . Ahora invocando al **teorema de Rolle**,* que establece que entre cualesquiera dos raíces de φ , ahí se debe presentar una raíz de φ' . Por tanto, φ' tiene al menos $n + 1$ raíces. Usando un razonamiento similar, φ'' tiene al menos n raíces, φ''' tiene al menos $n - 1$ raíces y así sucesivamente. Por último, de esto se puede inferir que $\varphi^{(n+1)}$ debe tener al menos una raíz. Sea ξ una raíz de $\varphi^{(n+1)}$.

***Teorema de Rolle:** Sea f una función continua en $[a, b]$ y derivable en (a, b) . Si $f(a) = f(b) = 0$, entonces $f'(c) = 0$ para algún punto c en (a, b) .

Todas las raíces que se están considerando en este argumento están en (a, b) . Por tanto,

$$0 = \varphi^{(n+1)}(\xi) = f^{(n+1)}(\xi) - p^{(n+1)}(\xi) - cw^{(n+1)}(\xi)$$

En esta ecuación, $p^{(n+1)}(\xi) = 0$ porque p es un polinomio de grado $\leq n$. También, $w^{(n+1)}(\xi) = (n+1)!$ porque $w(t) = t^{n+1} + (\text{ términos de orden inferior en } t)$. Por tanto, tenemos

$$0 = f^{(n+1)}(\xi) - c(n+1)! = f^{(n+1)}(\xi) - \frac{(n+1)!}{w(x)}[f(x) - p(x)]$$

Esta ecuación es un rearrreglo de la ecuación (2). ■

Un caso especial que con frecuencia surge es uno en el cual los nodos de interpolación están igualmente espaciados.

LEMA 1

Lema de límite superior

Suponga que $x_i = a + ih$ para $i = 0, 1, \dots, n$ y que $h = (b - a)/n$. Entonces para cualquier $x \in [a, b]$

$$\prod_{i=0}^n |x - x_i| \leq \frac{1}{4} h^{n+1} n! \quad (4)$$

Demostración Para establecer esta desigualdad, fije x y escoja j tal que $x_j \leq x \leq x_{j+1}$. Es un ejercicio de cálculo (problema 4.2.2) demostrar que

$$|x - x_j||x - x_{j+1}| \leq \frac{h^2}{4} \quad (5)$$

Usando la ecuación (5), tenemos

$$\prod_{i=0}^n |x - x_i| \leq \frac{h^2}{4} \prod_{i=0}^{j-1} (x - x_i) \prod_{i=j+2}^n (x_i - x)$$

El bosquejo en la figura 4.10 muestra un caso típico de nodos igualmente espaciados y puede ser útil. Ya que $x_j \leq x \leq x_{j+1}$, tenemos además

$$\prod_{i=0}^n |x - x_i| \leq \frac{h^2}{4} \prod_{i=0}^{j-1} (x_{j+1} - x_i) \prod_{i=j+2}^n (x_i - x_j)$$

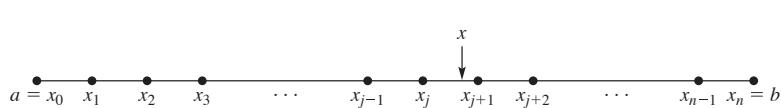


FIGURA 4.10
Localización típica de x en nodos igualmente espaciados

Ahora use el hecho de que $x_i = a + ih$. Entonces tenemos $x_{j+1} - x_i = (j - i + 1)h$ y $x_i - x_j = (i - j)h$. Por tanto,

$$\begin{aligned}\prod_{i=0}^n |x - x_i| &\leq \frac{h^2}{4} h^j h^{n-(j+2)+1} \prod_{i=0}^{j-1} (j - i + 1) \prod_{i=j+2}^n (i - j) \\ &\leq \frac{1}{4} h^{n+1} (j + 1)! (n - j)! \leq \frac{1}{4} h^{n+1} n!\end{aligned}$$

En el último paso, usemos el hecho de que si $0 \leq j \leq n - 1$, entonces $(j + 1)! (n - j)! \leq n!$. Esto, también, se deja como un ejercicio (problema 4.2.3). Por tanto, se establece la desigualdad (4). ■

Podemos ahora encontrar un límite en el error de interpolación.

■ TEOREMA 2

Errores de interpolación II

Sea f una función tal que $f^{(n+1)}$ es continua en $[a, b]$ y satisface que $|f^{(n+1)}(x)| \leq M$. Sea p el polinomio de grado $\leq n$ que interpola f en $n + 1$ nodos igualmente espaciados en $[a, b]$, incluidos los puntos finales. Entonces en $[a, b]$,

$$|f(x) - p(x)| \leq \frac{1}{4(n+1)} M h^{n+1} \quad (6)$$

donde $h = (b - a)/n$ es el espaciamiento entre nodos.

Demostración Use el teorema 1 de los errores de interpolación y la desigualdad (4) del lema 1. ■

Este teorema da límites superiores laxos en los errores de interpolación para diferentes valores de n . Usando otros medios, se pueden encontrar límites superiores más estrictos para valores pequeños de n . (Consulte el problema 4.2.5.) Si los nodos no están uniformemente espaciados entonces se puede encontrar un mejor límite al usar los nodos de Chebyshev.

EJEMPLO 2 Evalúe el error si $\sin x$ se remplaza usando un polinomio de interpolación que tiene diez nodos igualmente espaciados en $[0, 1.6875]$. (Véase el ejemplo 8 relacionado en la sección 4.1).

Solución Use el teorema 2 de errores de interpolación, tomando $f(x) = \sin x$, $n = 9$, $a = 0$ y $b = 1.6875$. Ya que $f^{(10)}(x) = -\sin x$, $|f^{(10)}(x)| \leq 1$. Por tanto, en la ecuación (6), podemos hacer $M = 1$. El resultado es

$$|\sin x - p(x)| \leq 1.34 \times 10^{-9}$$

Por tanto, $p(x)$ representa $\sin x$ en este intervalo con un error de a lo más dos unidades en el noveno lugar decimal. Por ende, el polinomio de interpolación que tiene diez nodos igualmente espaciados en el intervalo $[0, 1.6875]$ aproxima $\sin x$ con al menos ocho dígitos decimales de exactitud. De hecho, una cuidadosa comprobación en una computadora revelaría que el polinomio es exacto con aún más lugares decimales. (¿Por qué?) ■

La expresión del error en el polinomio de interpolación puede también darse en términos de diferencias divididas:

■ TEOREMA 3

Errores de interpolación III

Si p es el polinomio de grado n que interpola la función f en los nodos x_0, x_1, \dots, x_n , entonces para cualquier x que no es un nodo,

$$f(x) - p(x) = f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (x - x_i)$$

Demostración Sea t cualquier punto, diferente de un nodo, donde $f(t)$ está definida. Sea q el polinomio de grado $\leq n+1$ que interpola f en x_0, x_1, \dots, x_n, t . Usando la fórmula de interpolación de la forma de Newton [ecuación (8) de la sección 4.1], tenemos

$$q(x) = p(x) + f[x_0, x_1, \dots, x_n, t] \prod_{i=0}^n (x - x_i)$$

Puesto que $q(t) = f(t)$, esto produce en una vez

$$f(t) = p(t) + f[x_0, x_1, \dots, x_n, t] \prod_{i=0}^n (t - x_i)$$



El siguiente teorema muestra que hay una relación entre diferencias divididas y derivadas.

■ TEOREMA 4

Diferencias divididas y derivadas

Si $f^{(n)}$ es continua en $[a, b]$ y si x_0, x_1, \dots, x_n , son cualesquier $n+1$ puntos distintos en $[a, b]$, entonces para alguna ξ en (a, b) ,

$$f[x_0, x_1, \dots, x_n] = \frac{1}{n!} f^{(n)}(\xi)$$

Demostración Sea p el polinomio de grado $\leq n-1$ que interpola f en x_0, x_1, \dots, x_{n-1} . Usando el teorema 1 de errores de interpolación, existe un punto ξ tal que

$$f(x_n) - p(x_n) = \frac{1}{n!} f^{(n)}(\xi) \prod_{i=0}^{n-1} (x_n - x_i)$$

Usando el teorema 3 de errores de interpolación obtenemos

$$f(x_n) - p(x_n) = f[x_0, x_1, \dots, x_{n-1}, x_n] \prod_{i=0}^{n-1} (x_n - x_i)$$



Como una consecuencia inmediata de este teorema, observamos que todas las diferencias divididas de orden superior son cero para un polinomio.

COROLARIO 1**Diferencias divididas**

Si f es un polinomio de grado n , entonces todas las diferencias divididas son cero para $i \leq n + 1$.

EJEMPLO 3 ¿Existe un polinomio cúbico que tome estos valores?

x	1	-2	0	3	-1	7
y	-2	-56	-2	4	-16	376

Solución Si existe tal polinomio, sus diferencias divididas de cuarto orden $f[, , ,]$ serían cero. Formamos una tabla de diferencias divididas para comprobar esta posibilidad:

x	$f[]$	$f[,]$	$f[, ,]$	$f[, , ,]$	$f[, , , ,]$
1	-2				
-2	-56	18	-9		
0	-2	27	-5	2	
3	4	2	-3	2	0
-1	-16	5	11	2	0
7	376	49			

Los datos se pueden representar usando un polinomio cúbico debido a que las diferencias divididas de cuarto orden $f[, , ,]$ son cero. De la fórmula de la forma de Newton de interpolación, este polinomio es

$$p_3(x) = -2 + 18(x - 1) - 9(x - 1)(x + 2) + 2(x - 1)(x + 2)x$$



Resumen

- (1) La función de Runge $f(x) = 1/(1 + x^2)$ en el intervalo $[-5, 5]$ muestra que la interpolación del polinomio de alto grado y espaciamiento uniforme de nodos puede no ser satisfactoria. Los nodos de Chebyshev para el intervalo $[a, b]$ están dados por

$$x_i = \frac{1}{2}(a + b) + \frac{1}{2}(b - a) \cos \left[\left(\frac{2i + 1}{2n + 2} \right) \pi \right]$$

- (2) Existe una relación entre diferencias y derivadas:

$$f[x_0, x_1, \dots, x_n] = \frac{1}{n!} f^{(n)}(\xi)$$

(3) Las expresiones para errores en la interpolación de polinomio son

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) \prod_{i=0}^n (x - x_i)$$

$$f(x) - p(x) = f[x_0, x_1, \dots, x_n, x] \prod_{i=0}^n (x - x_i)$$

(4) Para $n + 1$ nodos igualmente espaciados, un límite superior del error está dado por

$$|f(x) - p(x)| \leq \frac{M}{4(n+1)} \left(\frac{b-a}{n} \right)^{n+1}$$

Aquí M es un límite superior de $|f^{(n+1)}(x)|$ cuando $a \leq x \leq b$.

(5) Si f es un polinomio de grado n , entonces todas las diferencias divididas $f[x_0, x_1, \dots, x_i]$ son cero para $i \geq n + 1$.

Problemas 4.2

- ^a1. Use una tabla de diferencias divididas para mostrar que los datos siguientes se pueden representar usando un polinomio de grado 3:

x	-2	-1	0	1	2	3
y	1	4	11	16	13	-4

2. Complete en detalle la prueba de desigualdad (4) usando la demostración de la desigualdad (5).
3. (Continuación) Complete en detalle la prueba de la desigualdad (4) mediante la demostración de que $(j+1)!(n-j)! \leq n!$ si $0 \leq j \leq n-1$. Se puede usar un argumento de inducción y de simetría.
4. Para nodos no uniformemente distribuidos $a = x_0 < x_1 < \dots < x_n = b$, donde $h = \max_{1 \leq i \leq n} \{(x_i - x_{i-1})\}$, demuestre que la desigualdad (4) es verdadera.
5. Usando el teorema 1, muestre directamente que el error máximo de interpolación está limitado por las expresiones siguientes y compárelas con las de los límites dados cuando se usa el teorema 2:
- a. $\frac{1}{8}h^2 M$ para interpolación lineal, donde $h = x_1 - x_0$ y $M = \max_{x_0 \leq x \leq x_1} |f''(x)|$.
 - b. $\frac{1}{9\sqrt{3}}h^3 M$ para interpolación cuadrática, donde $h = x_1 - x_0 = x_2 - x_1$ y $M = \max_{x_0 \leq x \leq x_2} |f''(x)|$.
 - c. $\frac{3}{128}h^4 M$ para interpolación cúbica, donde $h = x_1 - x_0 = x_2 - x_1 = x_3 - x_2$ y $M = \max_{x_0 \leq x \leq x_3} |f''(x)|$.
6. ¿Con qué exactitud podemos determinar $\sin x$ usando interpolación lineal, dando una tabla de $\sin x$ con diez lugares decimales, para x en $[0, 2]$ con $h = 0.01$?
7. (Continuación) Dados los datos

x	$\sin x$	$\cos x$
0.70	0.64421 76872	0.76484 21873
0.71	0.65183 37710	0.75836 18760

encuentre valores aproximados de $\sin 0.705$ y $\cos 0.702$ usando interpolación lineal. ¿Cuál es el error?

- 8.** **Interpolación lineal** en una tabla de valores de una función significa lo siguiente: si $y_0 = f(x_0)$ y $y_1 = f(x_1)$ son valores tabulados y si $x_0 < x < x_1$, entonces un valor interpolado de $f(x)$ es $y_0 + [(y_1 - y_0)/(x_1 - x_0)](x - x_0)$, como se explicó al comienzo de la sección 4.1. Se necesita una tabla de valores de $\cos x$ para que la interpolación lineal produzca una exactitud de cinco lugares decimales para cualquier valor de x en $[0, \pi]$. Suponga que los valores tabulados están igualmente espaciados y determine el número mínimo de entradas necesarias en esta tabla.
- 9.** Se usa un polinomio de interpolación de grado 20 para aproximar e^{-x} en el intervalo $[0, 2]$. ¿Cuán exacto será? (Use 21 nodos uniformes, incluidos los puntos finales del intervalo. Compare los resultados usando los teoremas 1 y 2.)
- 10.** Sea que la función $f(x) = \ln x$ se aproxime usando un polinomio de interpolación de grado 9 con diez nodos uniformemente distribuidos en el intervalo $[1, 2]$. ¿Qué límite se puede poner en el error?
- 11.** En el primer teorema de errores de interpolación, muestre que si $x_0 < x_1 < \dots < x_n$ y $x_0 < x < x_n$, entonces $x_0 < \xi < x_n$.
- 12.** (Continuación) En el mismo teorema, considere ξ como una función de x , muestre que $f^{(n)}[\xi(x)]$ es una función continua de x . *Nota:* es necesario que $\xi(x)$ no sea una función continua de x .
- 13.** Suponga que $\cos x$ se aproxima usando un polinomio de interpolación de grado n , usando $n+1$ nodos igualmente espaciados en el intervalo $[0, 1]$. ¿Qué exactitud tiene la aproximación? (Expresé su respuesta en términos de n .) ¿Qué exactitud tiene la aproximación cuando $n=9$? ¿Para qué valores de n es el error menor que 10^{-7} ?
- 14.** En interpolación con $n+1$ nodos igualmente espaciados en un intervalo, podríamos usar $x_i = a + (2i+1)h/2$, donde $0 \leq i \leq n-1$ y $h = (b-a)/n$. ¿Qué límite se puede dar ahora para $\prod_{i=0}^n |x - x_i|$ cuando $a \leq x \leq b$? *Nota:* no estamos pidiendo que los puntos finales sean nodos.
- 15.** Usando la ecuación (3), demuestre que
- $$w'(t) = \sum_{i=0}^n \prod_{j \neq i}^n (t - x_j) \quad w'(x_i) = \prod_{j=0}^n (x_i - x_j)$$
- 16.** ¿Cualquier polinomio p de grado a lo más n obedece la siguiente ecuación? Explique por qué sí o por qué no.
- $$p(x) = \sum_{i=0}^n p[x_0, x_1, \dots, x_i] \prod_{j=0}^{i-1} (x - x_j)$$
- Sugerencia:* use la unicidad del polinomio de interpolación.
- 17.** Encuentre un polinomio p que tome estos valores: $p(1) = 3$, $p(2) = 1$, $p(0) = -5$. Puede usar cualquier método que desee. Puede dejar al polinomio en cualquier forma conveniente, no necesariamente en la forma *estándar*, $\sum_{k=1}^n c_k x^k$. A continuación, encuentre un nuevo polinomio q que tome estos mismos tres valores y $q(3) = 7$.
- 18.** Para el caso $n = 2$, establezca el teorema 4 y el corolario 1 directamente.

Problemas de cómputo 4.2

- Usando 21 nodos igualmente espaciados en el intervalo $[-5, 5]$, encuentre el polinomio de interpolación p de grado 20 para la función $f(x) = (x^2 + 1)^{-1}$. Imprima los valores de $f(x)$ y $p(x)$ en los 41 puntos igualmente espaciados, incluidos los nodos. Observe la gran diferencia entre $f(x)$ y $p(x)$.
- (Continuación) Desarrolle el experimento del problema de cómputo anterior usando los nodos de Chebyshev $x_i = 5 \cos(i\pi/20)$, donde $0 \leq i \leq 20$, y los nodos $x_i = 5 \cos[(2i+1)\pi/42]$, donde $0 \leq i \leq 20$. Registre sus conclusiones.
- Usando los procedimientos correspondientes del seudocódigo en el libro, encuentre un polinomio de grado 13 que interpole $f(x) = \arctan x$ en el intervalo $[-1, 1]$. Pruebe numéricamente tomando 100 puntos para determinar la exactitud del polinomio de aproximación.
- (Continuación) Escriba una función para $\arctan x$ que use el polinomio del problema de cómputo anterior. Si x no está en el intervalo $[-1, 1]$, use la fórmula $1/\tan \theta = \cot \theta = \tan(\pi/2 - \theta)$.
- Aproxime $\arcsen x$ en el intervalo $[-1/\sqrt{2}, 1/\sqrt{2}]$ usando un polinomio de interpolación de grado 15. Determine la exactitud de la aproximación usando pruebas numéricas. Use nodos igualmente espaciados.
- (Continuación) Escriba una función para $\arcsen x$ usando el polinomio del problema de cómputo anterior. Use $\sin(\pi/2 - \theta) = \cos \theta = \sqrt{1 - \sin^2 \theta}$ si x está en el intervalo $|x| < 1/\sqrt{2}$.
- Sea $f(x) = \max\{0, 1 - x\}$. Trace la función f . Después encuentre polinomios de interpolación p de grados 2, 4, 8, 16 y 32 para f en el intervalo $[-4, 4]$, usando nodos igualmente espaciados. Imprima la diferencia $f(x) - p(x)$ en 128 puntos igualmente espaciados. Luego repita el problema usando nodos de Chebyshev.
- Usando *Coef*, *Eval* y un trazador automático, ajuste un polinomio que pase por los siguientes datos:

x	0.0	0.60	1.50	1.70	1.90	2.1	2.30	2.60	2.8	3.00
y	-0.8	-0.34	0.59	0.59	0.23	0.1	0.28	1.03	1.5	1.44

¿La curva resultante parece un buen ajuste? Explique.

- Encuentre el polinomio p de grado ≤ 10 que interpola $|x|$ en $[-1, 1]$ en 11 puntos igualmente espaciados. Imprima la diferencia $|x| - p(x)$ en 41 puntos igualmente espaciados. Después haga lo mismo con los nodos de Chebyshev. Compare.
- ¿Por qué son generalmente mejores los nodos de Chebyshev que los nodos igualmente espaciados en la interpolación de polinomios? La respuesta se encuentra en el término $\prod_{i=0}^n (x - x_i)$ que se presenta en la fórmula de error. Si $x_i = \cos[(2i+1)\pi/(2n+2)]$, entonces

$$\left| \prod_{i=0}^n (x - x_i) \right| \leq 2^{-n}$$

para toda x en $[-1, 1]$. Realice un experimento numérico para demostrar la desigualdad dada para $n = 3, 7, 15$.

11. (**Proyecto de investigación estudiantil**) Examine el tema de interpolación de datos multivariados dispersos, tales como los que se presentan en geofísica y otras áreas.
 12. Use software matemático como el que se encuentra en Matlab, Maple o Mathematica para reproducir las figuras 4.7 y 4.8.
 13. Use software matemático simbólico como Maple o Mathematica para generar el polinomio de interpolación para los puntos de datos del ejemplo 2. Trace el polinomio y los puntos de datos.
 14. Use software gráfico para trazar cuatro o cinco puntos que ocurren para generar un polinomio de interpolación que presenta un gran detalle de oscilaciones. Esta pieza de software le permite hacer *clic* con el mouse en tres o cuatro puntos que visualmente forman parte de una curva suave. Después use el polinomio de interpolación de Newton para trazar la curva que pasa por estos puntos. Entonces agregue otro punto que esté un poco distante de la curva y reajuste todos los puntos. Repita, agregando otros puntos. Después de que se han agregado unos cuantos puntos de esta manera, usted debe tener evidencia de que los polinomios pueden oscilar fuertemente.
-

4.3 Cálculo de derivadas y extrapolación de Richardson

Un experimento numérico delineado en el capítulo 1 (al final de la sección 1.1, p. 10) mostró que determinar la derivada de una función f en un punto x *no* es un problema numérico trivial. Específicamente, si $f(x)$ se puede calcular con sólo n dígitos de precisión, es difícil calcular $f'(x)$ numéricamente con n dígitos de precisión. Esta dificultad se puede ver en la resta de cantidades que son casi iguales. En esta sección se ofrecen varias alternativas para el cálculo numérico de $f'(x)$ y $f''(x)$.

Fórmulas de primera derivada mediante series de Taylor

Primero, considere de nuevo el método obvio basado en la definición de $f'(x)$. Consiste en seleccionar uno o más valores pequeños de h y escribir

$$f'(x) \approx \frac{1}{h}[f(x + h) - f(x)] \quad (1)$$

¿Qué error está implícito en esta fórmula? Para encontrarlo use el teorema de Taylor de la sección 1.2:

$$f(x + h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(\xi)$$

Rearreglando esta ecuación se obtiene

$$f'(x) = \frac{1}{h}[f(x + h) - f(x)] - \frac{1}{2}hf''(\xi) \quad (2)$$

Por tanto, veamos que la aproximación (1) tiene un término de error $-\frac{1}{2}hf''(\xi) = \mathcal{O}(h)$, donde ξ está en el intervalo que tiene puntos finales x y $x + h$.

La ecuación (2) muestra que, en general, conforme $h \rightarrow 0$, la diferencia entre $f'(x)$ y el cálculo de $h^{-1}[f(x + h) - f(x)]$ tienden a cero con la misma rapidez que h lo hace—es decir, $\mathcal{O}(h)$. Por supuesto, si $f''(x) = 0$, entonces el término de error será $\frac{1}{6}h^2 f'''(\gamma)$, que converge a cero un poco más rápido en $\mathcal{O}(h^2)$. Pero con frecuencia $f''(x)$ no es cero.

La ecuación (2) da el **error de truncamiento** para este procedimiento numérico, a saber, $-\frac{1}{2}hf''(\xi)$. Este error se presenta aun si los cálculos se realizan con precisión *infinita*; esto se debe a nuestra imitación del proceso de límite matemático mediante una fórmula de aproximación. Más (y peores) errores se deben esperar cuando se realizan cálculos en una computadora con longitud de palabra finita.

EJEMPLO 1 En la sección 1.1, el programa llamado *First* usa la regla unilateral (1) para aproximar la primera derivada de la función $f(x) = \sin x$ en $x = 0.5$. Explique qué ocurre cuando se realiza un gran número de iteraciones, digamos $n = 50$.

Solución ¡Hay una pérdida total de todos los dígitos significativos! Cuando examinamos con cuidado los resultados de la computadora hallamos que, de hecho, se encontró una buena aproximación $f'(0.5) \approx 0.87758$, pero se deterioró conforme continuó el proceso. Esto lo ocasionó la resta de dos cantidades casi iguales $f(x+h)$ y $f(x)$, lo cual resultó en una pérdida de dígitos significativos, así como un aumento de este efecto a partir de la división usando un pequeño valor de h . ¡Debemos parar las iteraciones pronto! Donde parar un proceso iterativo es un tema común en algoritmos numéricos. En este caso, se pueden supervisar las iteraciones para determinar cuándo se paran, a saber, cuando dos iteraciones sucesivas están dentro de una tolerancia prescrita. Alternativamente, podemos usar el término de error de truncamiento. Si queremos seis dígitos significativos de exactitud en los resultados, hacemos

$$\left| -\frac{1}{2}hf''(\xi) \right| \leq \frac{1}{2}4^{-n} < \frac{1}{2}10^{-6}$$

ya que $|f''(x)| < 1$ y $h = 1/4^n$. Encontramos $n > 6/\log 4 \approx 9.97$. Por ello, debemos parar después de cerca de diez pasos en el proceso. (El menor error de 3.1×10^{-9} se encontró en la iteración 14.) ■

Como vimos en el método de Newton (capítulo 3) y veremos en el método de Romberg (capítulo 5), es ventajoso que la convergencia del proceso numérico ocurra con potencias superiores de alguna cantidad que tienda a cero. En la situación actual, queremos una aproximación a $f'(x)$ en el cual el error se comporte como $\mathcal{O}(h^2)$. Tal método es fácilmente obtenido con la ayuda de las dos series de Taylor siguientes:

$$\begin{cases} f(x+h) = f(x) + hf'(x) + \frac{1}{2!}h^2f''(x) + \frac{1}{3!}h^3f'''(x) + \frac{1}{4!}h^4f^{(4)}(x) + \dots \\ f(x-h) = f(x) - hf'(x) + \frac{1}{2!}h^2f''(x) - \frac{1}{3!}h^3f'''(x) + \frac{1}{4!}h^4f^{(4)}(x) - \dots \end{cases} \quad (3)$$

Restando, obtenemos

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{2}{3!}h^3f'''(x) + \frac{2}{5!}h^5f^{(5)}(x) + \dots$$

Esto conduce a una fórmula de aproximación de $f'(x)$ muy importante:

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] - \frac{h^2}{3!}f'''(x) - \frac{h^4}{5!}f^{(5)}(x) - \dots \quad (4)$$

Que de otro modo se expresa como,

$$f'(x) \approx \frac{1}{2h}[f(x+h) - f(x-h)] \quad (5)$$

con un error cuyo término principal es $-\frac{1}{6}h^2f'''(x)$, que lo hace $\mathcal{O}(h^2)$.

Usando el teorema de Taylor con su término de error podríamos haber obtenido las dos expresiones siguientes:

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{1}{2}h^2f''(x) + \frac{1}{6}h^3f'''(\xi_1) \\ f(x-h) &= f(x) - hf'(x) + \frac{1}{2}h^2f''(x) - \frac{1}{6}h^3f'''(\xi_2) \end{aligned}$$

Entonces la resta conduciría a

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] - \frac{1}{6}h^2\left[\frac{f'''(\xi_1) + f'''(\xi_2)}{2}\right]$$

El término de error aquí se puede simplificar usando el siguiente razonamiento: la expresión $\frac{1}{2}[f'''(\xi_1) + f'''(\xi_2)]$ es el promedio de dos valores de f''' en el intervalo $[x-h, x+h]$. Por tanto, se encuentra entre el menor y el más grande valor de f''' en este intervalo. Si f''' es continua en este intervalo, entonces este valor promedio se supone en algún punto ξ . Por tanto, la fórmula con su término de error se puede escribir como

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] - \frac{1}{6}h^2f'''(\xi)$$

Esto se basa en la única suposición de que f''' es continua en $[x-h, x+h]$. Esta fórmula para derivación numérica resulta ser muy útil en la solución numérica de ciertas ecuaciones diferenciales, como veremos en el capítulo 14 (sobre problemas de valores en la frontera) y en el capítulo 15 (acerca de ecuaciones diferenciales parciales).

EJEMPLO 2 Modifique el programa *First* de la sección 1.1 para que use la fórmula de la diferencia central (5) para aproximar la primera derivada de la función $f(x) = \sin x$ en $x = 0.5$.

Solución Usando el término de error de truncamiento para la fórmula de la diferencia central (5) hacemos

$$\left| -\frac{1}{6}h^2f'''(\xi) \right| \leq \frac{1}{6}4^{-2n} < \frac{1}{2}10^{-6}$$

o $n > (6 - \log 3)/\log 16 \approx 4.59$. Obtenemos una buena aproximación después de cerca de cinco iteraciones con esta fórmula de orden superior. (El menor error de 3.6×10^{-12} estaba en el paso 9.)

Extrapolación de Richardson

Regresando ahora a la ecuación (4), la escribimos en una forma más simple:

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] + a_2h^2 + a_4h^4 + a_6h^6 + \dots \quad (6)$$

en la que las constantes a_2, a_4, \dots dependen de f y x . Cuando dicha información está disponible acerca de un proceso numérico, es posible usar una técnica poderosa conocida como *extrapolación de Richardson* para obtener más exactitud del método. Este procedimiento ahora se explicará usando la ecuación (6) como nuestro modelo.

Manteniendo a f y a x fijos, definimos una función de h usando la fórmula

$$\varphi(h) = \frac{1}{2h}[f(x+h) - f(x-h)] \quad (7)$$

De la ecuación (6), vemos que $\varphi(h)$ es una aproximación a $f'(x)$ con error de orden $\mathcal{O}(h^2)$. Nuestro objetivo es calcular $\lim_{h \rightarrow 0} \varphi(h)$, ya que esta es la cantidad $f'(x)$ que queremos en primer lugar. Si

elegimos una función f y graficamos $\varphi(h)$ para $h = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$, entonces obtenemos una gráfica (problema de cómputo 4.3.5). Cerca de cero, donde no podemos realmente calcular el valor de φ a partir de la ecuación (7), φ es aproximadamente una función cuadrática de h , ya que los términos de orden superior a partir de la ecuación (6) son despreciables. La extrapolación de Richardson busca calcular el valor limitante en 0 a partir de algunos valores calculados de $\varphi(h)$ cerca de 0. Obviamente, podemos tomar cualquier sucesión conveniente h_n que converja a cero, calcular $\varphi(h_n)$ de la ecuación (7) y usarlas como aproximaciones a $f'(x)$.

Pero se puede hacer algo mucho más inteligente. Suponga que calculamos $\varphi(h)$ para algunas h y después calculamos $\varphi(h/2)$. Usando la ecuación (6), tenemos

$$\begin{aligned}\varphi(h) &= f'(x) - a_2h^2 - a_4h^4 - a_6h^6 - \dots \\ \varphi\left(\frac{h}{2}\right) &= f'(x) - a_2\left(\frac{h}{2}\right)^2 - a_4\left(\frac{h}{2}\right)^4 - a_6\left(\frac{h}{2}\right)^6 - \dots\end{aligned}$$

Podemos eliminar el término dominante en la serie de error usando simple álgebra. Para hacerlo, multipliquemos la segunda ecuación por 4 y le restamos la primera ecuación. El resultado es

$$\varphi(h) - 4\varphi\left(\frac{h}{2}\right) = -3f'(x) - \frac{3}{4}a_4h^4 - \frac{15}{16}a_6h^6 - \dots$$

Dividimos entre -3 y la rearreglamos para obtener

$$\varphi\left(\frac{h}{2}\right) + \frac{1}{3}\left[\varphi\left(\frac{h}{2}\right) - \varphi(h)\right] = f'(x) + \frac{1}{4}a_4h^4 + \frac{5}{16}a_6h^6 + \dots$$

Este es un descubrimiento maravilloso. Simplemente al agregar $\frac{1}{3}[\varphi(h/2) - \varphi(h)]$ a $\varphi(h/2)$ hemos mejorado claramente la precisión a $\mathcal{O}(h^4)$ debido a que la serie de error que acompaña esta nueva combinación empieza con $\frac{1}{4}a_4h^4$. Como h será pequeña, este es un dramático mejoramiento.

Podemos repetir este proceso haciendo

$$\Phi(h) = \frac{4}{3}\varphi\left(\frac{h}{2}\right) - \frac{1}{3}\varphi(h)$$

Entonces a partir de la deducción anterior tenemos que

$$\begin{aligned}\Phi(h) &= f'(x) + b_4h^4 + b_6h^6 + \dots \\ \Phi\left(\frac{h}{2}\right) &= f'(x) + b_4\left(\frac{h}{2}\right)^4 + b_6\left(\frac{h}{2}\right)^6 + \dots\end{aligned}$$

Podemos combinar estas ecuaciones para eliminar el primer término de la serie de error

$$\Phi(h) - 16\Phi\left(\frac{h}{2}\right) = -15f'(x) + \frac{3}{4}b_6h^6 + \dots$$

Por tanto, tenemos

$$\Phi\left(\frac{h}{2}\right) + \frac{1}{15}\left[\Phi\left(\frac{h}{2}\right) - \Phi(h)\right] = f'(x) - \frac{1}{20}b_6h^5 + \dots$$

Este es otro mejoramiento claro en la precisión a $\mathcal{O}(h^6)$. Y ahora, para colmo, observe que el mismo procedimiento se puede repetir una y otra vez para *matar* los términos más y más superiores del error. Esta es la **extrapolación de Richardson**.

Esencialmente la misma situación surge en la deducción del algoritmo de Romberg del capítulo 5. Por tanto, es deseable tener aquí un análisis general del procedimiento. Iniciamos con una ecuación que incluye las dos situaciones. Sea j una función tal que

$$\varphi(h) = L - \sum_{k=1}^{\infty} a_{2k} h^{2k} \quad (8)$$

donde los coeficientes a_{2k} no son conocidos. La ecuación (8) no se interpreta como la *definición* de φ sino más bien como una *propiedad* que tiene φ . Se supone que $\varphi(h)$ se puede calcular para cualquier $h > 0$ y que nuestro objetivo es aproximar L exactamente usando φ .

Escoja una h conveniente y calcule los números

$$D(n, 0) = \varphi\left(\frac{h}{2^n}\right) \quad (n \geq 0) \quad (9)$$

De la ecuación (8), tenemos

$$D(n, 0) = L + \sum_{k=1}^{\infty} A(k, 0) \left(\frac{h}{2^n}\right)^{2k}$$

donde $A(k, 0) = -a_{2k}$. Estas cantidades $D(n, 0)$ dan una cruda estimación del número desconocido $L = \lim_{x \rightarrow 0} j(x)$. Estimaciones más exactas se obtienen mediante la extrapolación de Richardson. La fórmula de extrapolación es

$$D(n, m) = \frac{4^m}{4^m - 1} D(n, m - 1) - \frac{1}{4^m - 1} D(n - 1, m - 1) \quad (1 \leq m \leq n) \quad (10)$$

■ TEOREMA 1

Teorema de la extrapolación de Richardson

Las cantidades $D(n, m)$ definidas en el proceso de extrapolación de Richardson (10) obedecen la ecuación

$$D(n, m) = L + \sum_{k=m+1}^{\infty} A(k, m) \left(\frac{h}{2^n}\right)^{2k} \quad (0 \leq m \leq n) \quad (11)$$

Demostración

La ecuación (11) es verdadera usando la hipótesis si $m = 0$. Para el propósito de una prueba induktiva, *suponemos* que la ecuación (11) es válida para un valor arbitrario de $m - 1$ y probamos que la ecuación (11) es entonces válida para m . Ahora, a partir de ecuaciones (10) y (11) para un valor fijo m , tenemos

$$\begin{aligned} D(n, m) &= \frac{4^m}{4^m - 1} \left[L + \sum_{k=m}^{\infty} A(k, m - 1) \left(\frac{h}{2^n}\right)^{2k} \right] \\ &\quad - \frac{1}{4^m - 1} \left[L + \sum_{k=m}^{\infty} A(k, m - 1) \left(\frac{h}{2^{n-1}}\right)^{2k} \right] \end{aligned}$$

Después de la simplificación queda

$$D(n, m) = L + \sum_{k=m}^{\infty} A(k, m - 1) \left(\frac{4^m - 4^k}{4^m - 1}\right) \left(\frac{h}{2^n}\right)^{2k} \quad (12)$$

Por tanto, nos lleva a definir

$$A(k, m) = A(k, m - 1) \left(\frac{4^m - 4^k}{4^m - 1} \right)$$

Al mismo tiempo, observe que $A(m, m) = 0$. Por tanto, la ecuación (12) se puede escribir como

$$D(n, m) = L + \sum_{k=m+1}^{\infty} A(k, m) \left(\frac{h}{2^n} \right)^{2k}$$

La ecuación (11) es válida para m y la inducción está completa. ■

La importancia de la ecuación (11) es que la suma *comienza* con el término $(h/2^n)^{2m+2}$. Puesto que $h/2^n$ es pequeño, esto indica que los números $D(n, m)$ se acercan a L muy rápidamente, a saber,

$$D(n, m) = L + O \left(\frac{h^{2(m+1)}}{2^{2n(m+1)}} \right)$$

En la práctica, se pueden acomodar las cantidades en un arreglo triangular bidimensional como se muestra a continuación:

$$\begin{array}{ccccccc} D(0, 0) & & & & & & \\ D(1, 0) & D(1, 1) & & & & & \\ D(2, 0) & D(2, 1) & D(2, 2) & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ D(N, 0) & D(N, 1) & D(N, 2) & \cdots & D(N, N) & & \end{array} \quad (13)$$

Las principales tareas para generar tal arreglo son las siguientes:

■ ALGORITMO 2 Extrapolación de Richardson

1. Escriba una función para φ .
2. Decida valores convenientes para N y h .
3. Para $i = 0, 1, \dots, N$, calcule $D(i, 0) = \varphi(h/2^i)$.
4. Para $0 \leq i \leq j \leq N$, calcule

$$D(i, j) = D(i, j - 1) + (4^j - 1)^{-1} [D(i, j - 1) - D(i - 1, j - 1)]$$

Observe que en este algoritmo, el cálculo de $D(i, \varphi)$ sigue la ecuación (10) pero se ha reacomodado ligeramente para mejorar sus propiedades numéricas.

EJEMPLO 3 Escriba un procedimiento para calcular la derivada de una función en un punto usando la ecuación (5) y la extrapolación de Richardson.

Solución La entrada al procedimiento será una función f , un punto específico x , un valor de h y un número n que significa cuántos renglones en el arreglo (13) se calcularán. La salida será el arreglo (13).

Aquí se presenta un seudocódigo conveniente:

```

procedure Derivada (f, x, n, h, (dij))
integer i, j, n;  real h, x;  real array (dij)0:n×0:n
external function f
for i = 0 to n do
    di0 ← [f(x + h) − f(x − h)]/(2h)
    for j = 1 to i do
        di,j ← di,j-1 + (di,j-1 − di-1,j-1)/(4j − 1)
    end for
    h ← h/2
end for
end procedure Derivada

```

Para probar el procedimiento, elija $f(x) = \sin x$, donde $x_0 = 1.23095\ 94154$ y $h = 1$. Entonces $f'(x) = \cos x$ y $f'(x_0) = \frac{1}{3}$. Un seudocódigo se escribe como sigue:

```

program Prueba_de_Derivada
real array (dij)0:n×0:n;  external function f
integer n ← 10;  real h ← 1;  x ← 1.23095 94154
call Derivada (f, x, n, h, (dij))
output (dij)
end program Prueba_de_Derivada

real function f(x)
real x
f ← sin(x)
end function f

```

Le invitamos a programar el seudocódigo y ejecutarlo en una computadora. La salida de la computadora es el arreglo triangular (d_{ij}) con índices $0 \leq j \leq i \leq 10$. El valor más exacto es $(d_{4,1}) = 0.33333\ 33433$. Los valores d_{i0} , que se obtienen usando únicamente las ecuaciones (7) y (9) sin ninguna extrapolación, no son tan exactos, al no tener más de cuatro dígitos correctos. ■

Ahora hay software matemático disponible con capacidades de manipulación algebraica. Usándolo podríamos escribir un programa de cómputo para encontrar derivadas simbólicamente para una más grande clase de funciones, probablemente todas las que encontraría en un curso de cálculo. Por ejemplo, podríamos comprobar los resultados numéricos anteriores primero determinando la derivada exactamente y después evaluando la respuesta numérica $\cos(1.23095\ 94154) \approx 0.33333\ 33355$, ya que $(\frac{1}{3}) \approx 1.23095\ 941543$. Por supuesto, los procedimientos analizados en esta sección son para aproximar derivadas que no se pueden determinar exactamente.

Fórmulas de primera derivada mediante interpolación de polinomios

Se puede usar un importante artificio general para aproximar derivadas (así como integrales y otras cantidades). La función f primero se aproxima usando un polinomio p , por lo que $f \approx p$.

Entonces simplificando procedemos a la aproximación $f'(x) \approx p'(x')$ como una consecuencia. Por supuesto, esta estrategia se deberá usar *muy cuidadosamente* porque el comportamiento del polinomio de interpolación puede ser oscilatorio.

En la práctica, el polinomio de aproximación p con frecuencia se determina usando interpolación en pocos puntos. Por ejemplo, suponga que p es el polinomio de grado a lo más 1 que interpola f en dos nodos, x_0 y x_1 . Entonces de la ecuación (8) de la sección 4.1 con $n = 1$, tenemos

$$p_1(x) = f(x_0) + f[x_0, x_1](x - x_0)$$

Por tanto,

$$f'(x) \approx p'_1(x) = f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (14)$$

Si $x_0 = x$ y $x_1 = x + h$ (véase la figura 4.11), esta fórmula es la que se consideró antes, a saber, la ecuación (1):

$$f'(x) \approx \frac{1}{h}[f(x + h) - f(x)] \quad (15)$$

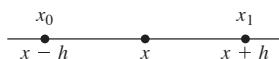
FIGURA 4.11

Diferencia hacia adelante:
dos nodos



FIGURA 4.12

Diferencia central:
dos nodos



Ahora considere la interpolación con tres nodos, x_0 , x_1 y x_2 . El polinomio de interpolación se obtiene a partir de la ecuación (8) de la sección 4.1:

$$p_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

y su derivada es

$$p'_2(x) = f[x_0, x_1] + f[x_0, x_1, x_2](2x - x_0 - x_1) \quad (17)$$

Aquí el miembro derecho consta de dos términos. El primero es la estimación anterior de la ecuación (14) y el segundo es un refinamiento o término de corrección.

Si se usa la ecuación (17) para evaluar $f'(x)$ cuando $x = \frac{1}{2}(x_0 + x_1)$, como en la ecuación (16), entonces el término de corrección en la ecuación (17) es cero. Por tanto, el primer término en este caso debe ser más exacto que los de los otros casos, ya que el término de corrección no agrega nada. Por ello la ecuación (16) es más exacta que la (15).

Un análisis de los errores de este procedimiento general es el siguiente: suponga que p_n es el polinomio de menor grado que interpola f en los nodos x_0, x_1, \dots, x_n . Entonces de acuerdo

con el primer teorema de errores de interpolación de la sección 4.2,

$$f(x) - p_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi) w(x)$$

donde ξ es dependiente de x y $w(x) = (x - x_0)(x - x_1) \cdots (x - x_n)$. Derivando se obtiene

$$f'(x) - p'_n(x) = \frac{1}{(n+1)!} w(x) \frac{d}{dx} f^{(n+1)}(\xi) + \frac{1}{(n+1)!} f^{(n+1)}(\xi) w'(x) \quad (18)$$

Aquí tuvimos que suponer que $f^{(n+1)}(\xi)$ es derivable como una función de x , un hecho que se conoce si $f^{(n+2)}$ existe y es continua.

La primera observación acerca de la fórmula del error en la ecuación (18) es que $w(x)$ se hace cero en cada nodo, por lo que si la evaluación está en un nodo x_i , la ecuación resultante es más simple:

$$f'(x_i) = p'_n(x_i) + \frac{1}{(n+1)!} f^{(n+1)}(\xi) w'(x_i)$$

Por ejemplo, tomando sólo dos puntos x_0 y x_1 , obtenemos con $n = 1$ e $i = 0$,

$$\begin{aligned} f'(x_0) &= f[x_0, x_1] + \frac{1}{2} f'(\xi) \frac{d}{dx} [(x - x_0)(x - x_1)] \Big|_{x=x_0} \\ &= f[x_0, x_1] + \frac{1}{2} f'(\xi) (x_0 - x_1) \end{aligned}$$

Esta es la ecuación (2) encubierta cuando $x_0 = x$ y $x_1 = x + h$. Resultados similares se obtienen con $n = 1$ e $i = 1$.

La segunda observación acerca de la ecuación (18) es que se volverá más simple si x se elige como un punto donde $w'(x) = 0$. Por ejemplo, si $n = 1$, entonces w es una función cuadrática que se hace cero en los dos nodos x_0 y x_1 . Debido a que una parábola es simétrica con respecto a su eje, $w'[(x_0 + x_1)/2] = 0$. La fórmula resultante es

$$f'\left(\frac{x_0 + x_1}{2}\right) = f[x_0, x_1] - \frac{1}{8}(x_1 - x_0)^2 \frac{d}{dx} f'(\xi)$$

Como un ejemplo final, considere cuatro puntos de interpolación: x_0, x_1, x_2 y x_3 . El polinomio de interpolación de la ecuación (8) de la sección 4.1 con $n = 3$ es

$$\begin{aligned} p_3(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &\quad + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \end{aligned}$$

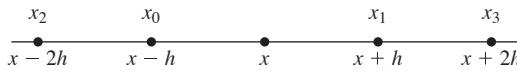
Su derivada es

$$\begin{aligned} p'_3(x) &= f[x_0, x_1] + f[x_0, x_1, x_2](2x - x_0 - x_1) \\ &\quad + f[x_0, x_1, x_2, x_3]((x - x_1)(x - x_2) \\ &\quad + (x - x_0)(x - x_2) + (x - x_0)(x - x_1)) \end{aligned}$$

Un caso especial útil ocurre si $x_0 = x - h$, $x_1 = x + h$, $x_2 = x - 2h$ y $x_3 = x + 2h$ (véase la figura 4.13). La fórmula resultante es

$$f'(x) \approx -\frac{2}{3h} [f(x + h) - f(x - h)] - \frac{1}{12h} [f(x + 2h) - f(x - 2h)]$$

FIGURA 4.13
Diferencia central: cuatro nodos



Esta se puede arreglar en una forma en la que probablemente se deba calcular con un término principal más un término de corrección o refinamiento:

$$\begin{aligned} f'(x) &\approx \frac{1}{2h} [f(x+h) - f(x-h)] \\ &\quad - \frac{1}{12h} \{f(x+2h) - 2[f(x+h) - f(x-h)] - f(x-2h)\} \end{aligned} \quad (19)$$

El término de error es $-\frac{1}{30}h^4 f^{(v)}(\xi) = \mathcal{O}(h^4)$.

Fórmulas de segunda derivada mediante series de Taylor

En la solución numérica de ecuaciones diferenciales con frecuencia es necesario aproximar segundas derivadas. Deduciremos las fórmulas más importantes para obtenerlas. Simplemente *sume* las dos series de Taylor (3) para $f(x+h)$ y $f(x-h)$. El resultado es

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + 2 \left[\frac{1}{4!} h^4 f^{(4)}(x) + \dots \right]$$

Cuando ésta se reacomoda queda

$$f''(x) = \frac{1}{h^2} [f(x+h) - 2f(x) + f(x-h)] + E$$

donde la serie de error es

$$E = -2 \left[\frac{1}{4!} h^2 f^{(4)}(x) + \frac{1}{6!} h^4 f^{(6)}(x) + \dots \right]$$

Realizando el mismo proceso usando la fórmula de Taylor con un residuo, se puede mostrar que E está también dado por

$$E = -\frac{1}{12} h^2 f^{(4)}(\xi)$$

para alguna ξ en el intervalo $(x-h, x+h)$. Por tanto, tenemos la aproximación

$$f''(x) \approx \frac{1}{h^2} [f(x+h) - 2f(x) + f(x-h)] \quad (20)$$

con error $\mathcal{O}(h^2)$.

EJEMPLO 4 Repita el ejemplo 2 usando la fórmula de la diferencia central (20) para aproximar la segunda derivada de la función $f(x) = \sin x$ en el punto dado $x = 0.5$.

Solución Usando el término de error de truncamiento, hacemos

$$\left| -\frac{1}{12} h^2 f^{(4)}(\xi) \right| \leq \frac{1}{12} 4^{-2n} < \frac{1}{2} 10^{-6}$$

y obtenemos $n > (6 - \log 6)/\log 16 \approx 4.34$. Por tanto, el programa modificado *First* encuentra una buena aproximación de $f''(0.5) \approx -0.47942$ después de cerca de cuatro iteraciones. (El menor error de 3.1×10^{-9} se obtuvo en la iteración 6.)

Fórmulas aproximadas de derivadas de orden superior se pueden obtener usando puntos desigualmente espaciados como en los nodos de Chebyshev. Recientemente, se han desarrollado paquetes de software para derivar automáticamente funciones que son expresadas usando un programa de computadora. Producen derivadas reales sólo con errores de redondeo y no con errores de discretización.

Ruido en cálculos

Un tema interesante es cómo afecta el ruido en la evaluación de $f(x)$ en el cálculo de derivadas cuando se usan las fórmulas estándar.

Las fórmulas para las derivadas se deducen con la expectativa de que la evaluación de la función en cualquier punto es posible, con *precisión total*. Entonces la derivada aproximada que se produce usando la fórmula difiere de la derivada real una cantidad que se llama el **término de error**, que implica el espaciamiento de los puntos de la muestra y algunas más altas derivadas de la función.

Si hay errores en los valores de la función (**ruido**), ¡pueden viciar todo el proceso! Estos errores podrían aplastar el error inherente en las fórmulas. El error inherente surge porque en la deducción de las fórmulas se truncó una serie de Taylor después de sólo unos pocos términos. Esto se llama **error de truncamiento** y está presente aun si la evaluación de la función en los puntos requeridos de la muestra es absolutamente correcta.

Por ejemplo, considere la fórmula

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(\xi)$$

El término con h^2 es el término de error. El punto ξ es un punto cercano (desconocido). Si $f(x+h)$ y $f(x-h)$ tienen error a lo más de d , entonces se puede ver que la fórmula producirá un valor para $f'(x)$ que tienen un error de d/h , que es grande cuando h es pequeña. El ruido vicia completamente el proceso si d es grande.

Para un caso numérico específico, suponga que $h = 10^{-2}$ y $|f'''(s)| \leq 6$. Entonces el error de truncamiento, E , satisface $|E| \leq 10^{-4}$. La derivada calculada a partir de la fórmula con precisión total está dentro de 10^{-4} de la derivada real. Suponga, sin embargo, que hay ruido en la evaluación de $f(x \pm h)$ de magnitud $d = h$. El valor correcto de $[f(x+h) - f(x-h)]/(2h)$ puede diferir del valor con ruido por $(2d)/(2h) = 1$.

Resumen

(1) Hemos deducido fórmulas para aproximar la primera y la segunda derivadas. Para $f'(x)$, una fórmula unilateral es

$$f'(x) \approx \frac{1}{h}[f(x+h) - f(x)]$$

con término de error $-\frac{1}{2}hf''(\xi)$. Una fórmula de la diferencia central es

$$f'(x) \approx \frac{1}{2h}[f(x+h) - f(x-h)]$$

con error $-\frac{1}{6}h^2 f'''(\xi) = \mathcal{O}(h^2)$. Una fórmula de diferencia central con un término de corrección es

$$\begin{aligned} f'(x) &\approx \frac{1}{2h} [f(x+h) - f(x-h)] \\ &\quad - \frac{1}{12h} [f(x+2h) - 2f(x+h) + 2f(x-h) - f(x-2h)] \end{aligned}$$

con término de error $-\frac{1}{30}h^4 f^{(v)}(\xi) = \mathcal{O}(h^4)$.

(2) Para $f''(x)$, una fórmula de diferencia central es

$$f''(x) \approx \frac{1}{h^2} [f(x+h) - 2f(x) + f(x-h)]$$

con término de error $-\frac{1}{12}h^2 f^{(4)}(\xi)$.

(3) Si $\varphi(h)$ es una de estas fórmulas con serie de error $a_2h^2 + a_4h^4 + a_6h^6 + \dots$, entonces podemos aplicar la **extrapolación de Richardson** como sigue

$$\begin{cases} D(n, 0) = \varphi(h/2^n) \\ D(n, m) = D(n, m-1) + [D(n, m-1) - D(n-1, m-1)]/(4^m - 1) \end{cases}$$

con términos de error

$$D(n, m) = L + \mathcal{O}\left(\frac{h^{2(m+1)}}{2^{2n(m+1)}}\right)$$

Referencias adicionales del capítulo 4

Para un estudio adicional, véase Gautschi [1990], Goldstine [1977], Griewank [2000], Groetsch [1998], Rivlin [1990] y Whittaker y Robinson [1944].

Problemas 4.3

*a*1. Determine el término de error para la fórmula

$$f'(x) \approx \frac{1}{4h} [f(x+3h) - f(x-h)]$$

*a*2. Usando series de Taylor, establezca el término de error para la fórmula

$$f'(0) \approx \frac{1}{2h} [f(2h) - f(0)]$$

3. Deduzca la fórmula de aproximación

$$f'(x) \approx \frac{1}{2h} [4f(x+h) - 3f(x) - f(x+2h)]$$

y muestre que su término de error es de la forma $\frac{1}{3}h^2 f'''(\xi)$.

*a*4. ¿Puede encontrar una fórmula de aproximación para $f'(x)$ que tiene término de error $\mathcal{O}(h^3)$ y que implica sólo dos evaluaciones de la función f ? Pruebe o refute.

5. Promediando la fórmula de diferencia hacia adelante $f'(x) \approx [f(x+h) - f(x)]/h$ y la fórmula de diferencia hacia atrás $f'(x) \approx [f(x) - f(x-h)]/h$, cada una con término de error

$\mathcal{O}(h)$, se obtiene la fórmula de la diferencia central $f'(x) \approx [f(x+h) - f(x-h)]/(2h)$ con error $\mathcal{O}(h^2)$. *Sugerencia:* determine al menos el primer término en la serie de error para cada fórmula.

6. Juzgue el siguiente análisis. Usando la fórmula de Taylor, tenemos

$$\begin{aligned} f(x+h) - f(x) &= hf'(x) + \frac{h^2}{2}f''(\xi) + \frac{h^3}{6}f'''(\xi) \\ f(x-h) - f(x) &= -hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\xi) \end{aligned}$$

Por adición, obtenemos una expresión *exacta* para $f''(x)$:

$$f(x+h) + f(x-h) - 2f(x) = h^2 f''(x)$$

7. Juzgue el siguiente análisis. Usando la fórmula de Taylor, tenemos

$$\begin{aligned} f(x+h) - f(x) &= hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(\xi_1) \\ f(x-h) - f(x) &= -hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\xi_2) \end{aligned}$$

Por tanto,

$$\frac{1}{h^2} [f(x+h) - 2f(x) + f(x-h)] = f''(x) + \frac{h}{6} [f'''(\xi_1) - f'''(\xi_2)]$$

El error en la fórmula de aproximación para f'' es por tanto $\mathcal{O}(h)$.

8. Deduzca las dos fórmulas

a. $f'(x) \approx \frac{1}{4h}[f(x+2h) - f(x-2h)]$

b. $f''(x) \approx \frac{1}{4h^2}[f(x+2h) - 2f(x) + f(x-2h)]$

y establezca fórmulas para los errores en su uso.

9. Deduzca las siguientes reglas para estimar derivadas:

a. $f'''(x) \approx \frac{1}{2h^3}[f(x+2h) - 2f(x+h) + 2f(x-h) - f(x-2h)]$

b. $f^{(4)}(x) \approx \frac{1}{h^4}[f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)]$

y sus términos de error. ¿Cuál es más exacta? *Sugerencia:* considere la serie de Taylor para $D(h) \equiv f(x+h) - f(x-h)$ y $S(h) \equiv f(x+h) + f(x-h)$.

10. Establezca la fórmula

$$f''(x) \approx \frac{2}{h^2} \left[\frac{f(x_0)}{(1+\alpha)} - \frac{f(x_1)}{\alpha} + \frac{f(x_2)}{\alpha(\alpha+1)} \right]$$

de las dos maneras siguientes, usando los puntos desigualmente espaciados $x_0 < x_1 < x_2$, donde $x_1 - x_0 = h$ y $x_2 - x_1 = ah$. Observe que esta fórmula se reduce a la fórmula de diferencia central estándar (20) cuando $\alpha = 1$.

- Aproxime $f'(x)$ usando la forma de Newton del polinomio de interpolación de grado 2.
- Calcule los coeficientes indeterminados A , B y C en la expresión $f''(x) \approx Af(x_0) + Bf(x_1) + Cf(x_2)$ al hacerla exacta para los tres polinomios 1 , $x - x_1$ y $(x - x_1)^2$ y por tanto exacta para todos los polinomios de grado ≤ 2 .

“11. (Continuación) Usando series de Taylor, muestre que

$$f'(x_1) = \frac{f(x_2) - f(x_0)}{x_2 - x_0} + (\alpha - 1)\frac{h}{2}f''(x_1) + \mathcal{O}(h^2)$$

Establezca que el error para la aproximación de $f'(x_1)$ usando $[f(x_2) - f(x_0)]/(x_2 - x_0)$ es $\mathcal{O}(h^2)$ cuando x_1 está a la mitad entre x_0 y x_2 pero sólo $\mathcal{O}(h)$ en cualquier otro caso.

“12. Un cierto cálculo requiere de una fórmula de aproximación para $f'(x) + f''(x)$. ¿Cuán bien funciona la expresión

$$\left(\frac{2+h}{2h^2}\right)f(x+h) - \left(\frac{2}{h^2}\right)f(x) + \left(\frac{2-h}{2h^2}\right)f(x-h)$$

Deduzca esta aproximación y su término de error.

“13. Los valores de una función f están dados en tres puntos x_0 , x_1 y x_2 . Si se usa un polinomio cuadrático de interpolación para calcular $f'(x)$ en $x = \frac{1}{2}(x_0 + x_1)$, ¿qué fórmula resultará?

14. Considere la ecuación (19).

- Complete los detalles de su deducción.
- Usando series de Taylor, deduzca su término de error.

15. Muestre cómo trabajaría la extrapolación de Richardson en la fórmula (20).

“16. Si $\varphi(h) = L - c_1h - c_2h^2 - c_3h^3 - \dots$, entonces, ¿qué combinación de $\varphi(h)$ y $\varphi(h/2)$ debe dar un cálculo exacto de L ?

17. (Continuación) Establezca y pruebe un teorema análogo al teorema de extrapolación de Richardson para la situación del problema anterior.

18. Si $\varphi(h) = L - c_1h^{1/2} - c_2h^{2/2} - c_3h^{3/2} - \dots$, entonces, ¿qué combinación de $\varphi(h)$ y $\varphi(h/2)$ deberá dar un cálculo exacto de L ?

19. Muestre que la extrapolación de Richardson se puede realizar para dos valores cualesquiera de h . Por tanto, si $\varphi(h) = L - \mathcal{O}(h^p)$, entonces a partir de $\varphi(h_1)$ y $\varphi(h_2)$ se da un cálculo más exacto de L usando

$$\varphi(h_2) + \frac{h_2^p}{h_1^p - h_2^p}[\varphi(h_2) - \varphi(h_1)]$$

“20. Considere una función φ tal que $\lim_{h \rightarrow 0} \varphi(h) = L$ y $L - \varphi(h) \approx ce^{-lh}$ para alguna constante c . Combinando $\varphi(h)$, $\varphi(h/2)$ y $\varphi(h/3)$, encuentre un cálculo exacto de L .

- 21.** Considere la fórmula de aproximación

$$f'(x) \approx \frac{3}{2h^3} \int_{-h}^h tf(x+t) dt$$

Determine su término de error. ¿La función f tiene que ser derivable para que la fórmula tenga sentido? *Sugerencia:* este es un método nuevo para hacer derivación numérica. Si usted está interesado, puede leer más acerca de la **derivada generalizada de Lanczos** en Groetsch [1998].

- 22.** Deduzca los términos de error para $D(3, 0)$, $D(3, 1)$, $D(3, 2)$ y $D(3, 3)$.
- 23.** La derivación y la integración son procesos mutuamente inversos. La derivación es un problema inherentemente sensible en el cual pequeños cambios en los datos pueden causar grandes cambios en los resultados. La integración es un proceso de suavización y es inherentemente estable. Represente dos funciones que tengan derivadas muy diferentes pero integrales definidas iguales y viceversa.
- 24.** Establezca los términos de errores para estas reglas:

- a. $f'''(x) \approx \frac{1}{2h^3} [3f(x+h) - 10f(x) + 12f(x-h) - 6f(x-2h) + f(x-3h)]$
- b. $f'(x) + \frac{h}{2}f'' \approx \frac{1}{h} [f(x+h) - f(x)]$
- c. $f^{(iv)}(x) \approx \frac{1}{h^4} \left[\frac{4}{3}f(x+3h) - 6f(x+2h) + 12f(x+h) \right]$ si $f(x) = f'(x) = 0$.

Problemas de cómputo 4.3

- 1.** Pruebe el procedimiento *Derivada* en las siguientes funciones en los puntos indicados en una sola corrida de la computadora. Interprete los resultados.
- a. $f(x) = \cos x$ en $x = 0$
 - b. $f(x) = \arctan x$ en $x = 1$
 - c. $f(x) = |x|$ en $x = 0$
- 2.** (Continuación) Escriba y pruebe un procedimiento similar a *Derivada* que calcule $f''(x)$ al repetir la extrapolación de Richardson.
- 3.** Encuentre $f'(0.25)$ tan exactamente como sea posible, usando sólo la función correspondiente al seudocódigo que se muestra a continuación y un método de derivación numérica:

```

real function f(x)
integer i; real a, b, c, x
a ← 1; b ← cos(x)
for i = 1 to 5 do
    c ← b
    b ← √(ab)
    a ← (a + c)/2
end for
f ← 2 arctan(1)/ a
end function f

```

4. Realice un experimento numérico para comparar la exactitud de las fórmulas (5) y (19) en una función f cuya derivada se puede calcular exactamente. Tome una secuencia de valores para h , como 4^{-n} con $0 \leq n \leq 12$.
5. Usando el análisis de la interpretación geométrica de la extrapolación de Richardson, produzca una gráfica para mostrar que $\varphi(h)$ parece una curva cuadrática en h .
6. Use software simbólico matemático como Maple o Mathematica para establecer el primer término en la serie de error para la ecuación (19).
7. Use software matemático como el que se encuentra en Matlab, Maple o Mathematica para repetir el ejemplo 1.

Integración numérica

En la teoría de campo eléctrico se ha demostrado que el campo magnético inducido por una corriente que fluye en una espira de alambre circular tiene una intensidad

$$H(x) = \frac{4I}{r^2 - x^2} \int_0^{\pi/2} \left[1 - \left(\frac{x}{r} \right)^2 \sin^2 \theta \right]^{1/2} d\theta$$

donde I es la corriente, r el radio de la espira y x la distancia del centro al punto donde se está calculando la intensidad magnética ($0 \leq x \leq r$). Si I , r y x están dados, tenemos una formidable integral para evaluar. Esta es una **integral elíptica** y no se expresa en términos de funciones familiares. Pero H se puede calcular exactamente con los métodos expuesto en este capítulo. Por ejemplo, si $I = 15.3$, $r = 120$ y $x = 84$, encontramos $H = 1.35566\,1135$ con exactitud a nueve decimales.

5.1 Sumas inferior y superior

El cálculo elemental se centra en gran parte en dos procesos importantes de las matemáticas: derivación e integración. En la sección 1.1 se consideró brevemente la derivación numérica; se estudió de nuevo en la sección 4.3. En este capítulo se examina el proceso de integración desde el punto de vista de la matemática numérica.

Integrales definidas e indefinidas

Se acostumbra distinguir dos tipos de integrales: la integral definida y la indefinida. La **integral indefinida** de una función es *otra función* o una clase de funciones, mientras que la **integral definida** de una función en un intervalo fijo es un *número*. Por ejemplo,

$$\begin{aligned} \text{Integral indefinida: } & \int x^2 dx = \frac{1}{3}x^3 + C \\ \text{Integral definida: } & \int_0^2 x^2 dx = \frac{8}{3} \end{aligned}$$

En realidad, una función no tiene sólo una sino varias integrales indefinidas. Éstas difieren entre sí por las constantes. Así, en el ejemplo anterior, se puede asignar cualquier valor constante

a C y el resultado sigue siendo una integral indefinida. En cálculo elemental, el concepto de una integral indefinida es idéntico al de una antiderivada. Una **antiderivada** de una función f es cualquier función F que tiene la propiedad de que $F' = f$.

Las integrales definida e indefinida están relacionadas con el **teorema fundamental del cálculo**,* que establece que $\int_a^b f(x) dx$ se puede calcular al encontrar primero una antiderivada F de f y después evaluando $F(b) - F(a)$. Así, usando la notación tradicional, tenemos

$$\int_1^3 (x^2 - 2) dx = \left(\frac{x^3}{3} - 2x \right) \Big|_1^3 = \left(\frac{27}{3} - 6 \right) - \left(\frac{1}{3} - 2 \right) = \frac{14}{3}$$

Como otro ejemplo del teorema fundamental del cálculo podemos escribir

$$\begin{aligned}\int_a^b F'(x) dx &= F(b) - F(a) \\ \int_a^x F'(t) dt &= F(x) - F(a)\end{aligned}$$

Si esta segunda ecuación se deriva con respecto a x , el resultado es (y aquí hemos puesto $f = F'$)

$$\frac{d}{dx} \int_a^x f(t) dt = f(x)$$

Esta última ecuación muestra que $\int_a^x f(t) dt$ debe ser una antiderivada (integral indefinida) de f .

La técnica anterior para calcular las integrales definidas es virtualmente lo único que se enfatiza en cálculo elemental. La integral definida de una función, sin embargo, tiene una interpretación como el área bajo una curva y así la existencia de un valor numérico para $\int_a^b f(x) dx$ no debe depender lógicamente de nuestra limitada habilidad para encontrar antiderivadas. Así, por ejemplo,

$$\int_0^1 e^{x^2} dx$$

tiene un valor numérico exacto a pesar del hecho de que no hay una función elemental F tal que $F'(x) = e^{x^2}$. Por las observaciones anteriores, e^{x^2} tiene antiderivadas, una de las cuales es

$$F(x) = \int_0^x e^{t^2} dt$$

Sin embargo, esta forma de la función F no ayuda a determinar el valor numérico buscado.

Sumas inferior y superior

La existencia de la integral definida de una función no negativa f en un intervalo cerrado $[a, b]$ está basada en una interpretación de la integral como el área bajo una curva de f . La integral definida se determina por medio de dos conceptos, las *sumas inferiores* de f y las *sumas superiores* de f ; estas son aproximaciones al área bajo una curva.

***Teorema fundamental del cálculo:** Si f es continua en el intervalo $[a, b]$ y F es una antiderivada de f , entonces

$$\int_a^b f(x) dx = F(b) - F(a)$$

Sea P una **partición** del intervalo $[a, b]$ dada por

$$P = \{a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b\}$$

con puntos de partición $x_0, x_1, x_2, \dots, x_n$ que dividen el intervalo $[a, b]$ en n subintervalos $[x_i, x_{i+1}]$. Ahora denotamos por m_i el **límite inferior más grande** (*infimum* o \inf) de $f(x)$ en el subintervalo $[x_i, x_{i+1}]$. Simbólicamente,

$$m_i = \inf \{f(x) : x_i \leq x \leq x_{i+1}\}$$

Del mismo modo, denotamos con M_i el **límite superior más pequeño** (*supremum* o \sup) de $f(x)$ en $[x_i, x_{i+1}]$. Así,

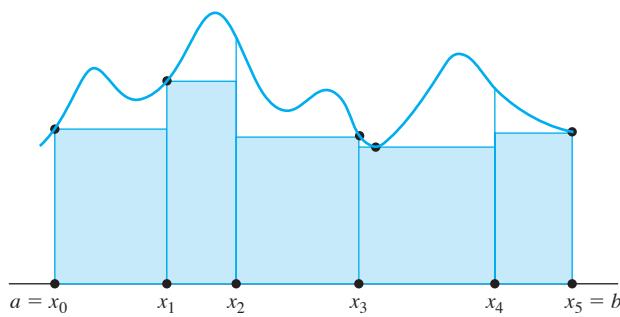
$$M_i = \sup \{f(x) : x_i \leq x \leq x_{i+1}\}$$

Las **sumas inferiores** y las **sumas superiores** de f corresponden a la partición P y están definidas como

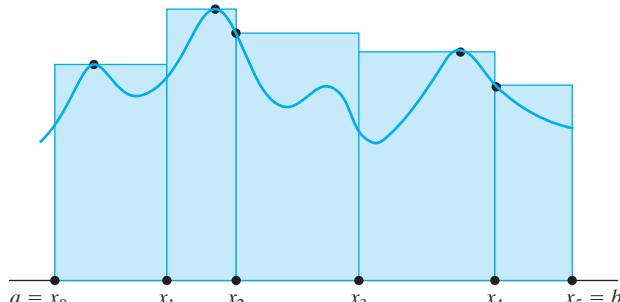
$$L(f; P) = \sum_{i=0}^{n-1} m_i(x_{i+1} - x_i)$$

$$U(f; P) = \sum_{i=0}^{n-1} M_i(x_{i+1} - x_i)$$

Si f es una función positiva, estas dos cantidades se pueden interpretar como cálculos del área bajo la curva de f . Estas sumas se muestran en la figura 5.1.



(a) Sumas inferiores



(b) Sumas superiores

FIGURA 5.1
Ilustración de
sumas inferior
y superior

EJEMPLO 1 ¿Cuáles son los valores numéricos de la sumas superior y inferior para $f(x) = x^2$ en el intervalo $[0, 1]$ si la partición es $P = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$?

Solución Queremos el valor de

$$U(f; P) = M_0(x_1 - x_0) + M_1(x_2 - x_1) + M_2(x_3 - x_2) + M_3(x_4 - x_3)$$

Puesto que f es creciente en $[0, 1]$, $M_0 = f(x_1) = \frac{1}{16}$. De manera similar, $M_1 = f(x_2) = \frac{1}{4}$, $M_2 = f(x_3) = \frac{9}{16}$ y $M_3 = f(x_4) = 1$. Los anchos de los subintervalos son todos iguales a $\frac{1}{4}$. Por tanto,

$$U(f; P) = \frac{1}{4} \left(\frac{1}{16} + \frac{1}{4} + \frac{9}{16} + 1 \right) = \frac{15}{32}$$

De la misma manera, encontramos que $m_0 = f(x_0) = 0$, $m_1 = \frac{1}{16}$, $m_2 = \frac{1}{4}$ y $m_3 = \frac{9}{16}$. Por tanto,

$$L(f; P) = \frac{1}{4} \left(0 + \frac{1}{16} + \frac{1}{4} + \frac{9}{16} \right) = \frac{7}{32}$$

Si no teníamos otra forma de calcular $\int_0^1 x^2 dx$, el mejor cálculo sería un valor a medio camino entre $U(f; P)$ y $L(f; P)$. Este número es $\frac{11}{32}$. El valor correcto es $\frac{1}{3}$ y el error es $\frac{11}{32} - \frac{1}{3} = \frac{1}{96}$. ■

Es intuitivamente claro que la suma superior *sobreestima* el área bajo la curva y la suma inferior la *subestima*. Por tanto, la expresión $\int_a^b f(x) dx$, que estamos tratando de definir, es *requerida* para satisfacer la desigualdad básica

$$L(f; P) \leq \int_a^b f(x) dx \leq U(f; P) \quad (1)$$

para todas las particiones P . Resulta que si f es una función *continua* definida en $[a, b]$, entonces la desigualdad (1) en efecto define la integral. Es decir, hay un y sólo un número real que es mayor o igual que todas las sumas inferiores de f y menor o igual que todas las sumas superiores de f . Este único número (que depende de f , a , y b) está definido como $\int_a^b f(x) dx$. La integral también existe si f es monótona creciente en $[a, b]$ o monótona decreciente en $[a, b]$.

Funciones integrables de Riemann

Consideremos el límite superior más pequeño (*supremum*) del conjunto de todos los números $L(f; P)$ que se obtienen cuando a P se le permite variar sobre todas las particiones del intervalo $[a, b]$. Éste se abrevia $\sup_P L(f; P)$. De manera similar, consideremos el límite inferior más grande (*infimum*) de $U(f; P)$ cuando P varía sobre todas las particiones de $[a, b]$. Esto se denota por $\inf_P U(f; P)$. Ahora si estos dos números son iguales, es decir, si

$$\inf_P U(f; P) = \sup_P L(f; P) \quad (2)$$

entonces decimos que f es una función **integrable de Riemann** en $[a, b]$ y se define $\int_a^b f(x) dx$ es el valor común obtenido en la ecuación (2). El resultado importante mencionado se puede enunciar formalmente como sigue:

■ TEOREMA 1

Teorema de la integral de Riemann

Toda función continua definida en un intervalo cerrado y acotado en la recta real es una función integrable de Riemann.

Hay muchas funciones que *no* son funciones integrables de Riemann. La más simple se conoce como **función de Dirichlet**:

$$d(x) = \begin{cases} 0 & \text{si } x \text{ es racional} \\ 1 & \text{si } x \text{ es irracional} \end{cases}$$

Para cualquier intervalo $[a, b]$ y para cualquier partición P de $[a, b]$, tenemos $L(d; P) = 0$ y $U(d; P) = b - a$. Por tanto,

$$0 = \sup_P L(d; P) < \inf_P U(d; P) = b - a$$

En cálculo, se ha demostrado no sólo que la integral de Riemann de una función continua en $[a, b]$ existe, sino también que se puede obtener con dos límites:

$$\lim_{n \rightarrow \infty} L(f; P_n) = \int_a^b f(x) dx = \lim_{n \rightarrow \infty} U(f; P_n)$$

en la que P_0, P_1, \dots es cualquier sucesión de particiones con la propiedad de que la longitud del subintervalo más largo en P_n converge a cero cuando $n \rightarrow \infty$. Además, si esto se arregla para que P_{n+1} se obtenga de P_n al agregarle (y no borrarle) nuevos puntos, entonces las sumas inferiores convergen *hacia arriba* en la integral y las sumas superiores convergen *hacia abajo* en la integral. Desde el punto de vista numérico, esta es una característica deseable del proceso porque en cada paso, un intervalo que contiene el número desconocido $\int_a^b f(x) dx$ estará disponible. Además, estos intervalos reducen el ancho en cada paso sucesivo.

Ejemplos y seudocódigo

El proceso apenas descrito puede realizarse fácilmente en una computadora. Para ilustrarlo, seleccionamos la función $f(x) = e^{-x^2}$ y el intervalo $[0, 1]$; es decir, consideremos

$$\int_0^1 e^{-x^2} dx \tag{3}$$

Esta función es muy importante en estadística, pero su integral indefinida no se puede obtener con técnicas elementales de cálculo. Para particiones, tomamos puntos igualmente espaciados en $[0, 1]$. Así, si hay n subintervalos en P_n , entonces definimos $P_n = \{x_0, x_1, \dots, x_n\}$, donde $x_i = ih$ para $0 \leq i \leq n$ y $h = 1/n$. Puesto que e^{-x^2} es *decreciente* en $[0, 1]$, el menor valor de f en el subintervalo $[x_i, x_{i+1}]$ se presenta en x_{i+1} . De manera similar, el valor más grande se presenta en x_i . Por tanto, $m_i = f(x_{i+1})$ y $M_i = f(x_i)$. Poniendo esto en las fórmulas para las sumas superior e inferior obtenemos para esta función

$$\begin{aligned} L(f; P_n) &= \sum_{i=0}^{n-1} hf(x_{i+1}) = h \sum_{i=0}^{n-1} e^{-x_{i+1}^2} \\ &= \\ U(f; P_n) &= \sum_{i=0}^{n-1} hf(x_i) = h \sum_{i=0}^{n-1} e^{-x_i^2} \end{aligned}$$

Puesto que estas sumas son casi iguales, es más económico calcular $L(f; P_n)$ con la fórmula y obtener $U(f; P_n)$ al observar que

$$U(f; P_n) = hf(x_0) + L(f; P_n) - hf(x_n) = L(f; P_n) + h(1 - e^{-1})$$

La última ecuación también muestra que el intervalo definido por la desigualdad (1) tiene un ancho de $h(1 - e^{-1})$ para este problema.

Aquí se presenta un seudocódigo para realizar este experimento con $n = 1000$:

```

program Sumas
integer i; real h, suma, suma_inferior, suma_superior
integer n ← 1000; real a ← 0, b ← 1
h ← (b - a)/n
suma ← 0
for i = n to 1 step -1 do
    x ← a + ih
    suma ← suma + f(x)
end for
suma_inferior ← (suma)h
suma_superior ← suma_inferior + h[f(a) - f(b)]
output suma_inferior, suma_superior
end program Sumas

real function f(x)
real x
f ← e-x2
end function f

```

Un breve comentario acerca del seudocódigo puede ser útil. Primero, no se necesita una variable subindizada para los puntos x_i . Cada punto está etiquetado con x . Después de que se ha definido y usado, no se necesita guardar. Después, observe que el programa ha sido escrito para que sólo se deba cambiar un renglón del código si se requiere otro valor de n . Por último, se agregan los números e^{-x^2} en orden de magnitud *ascendente* para reducir el error de redondeo. Sin embargo, los errores de redondeo en la computadora son despreciables comparados con el error en nuestro cálculo final de la integral. Este código se puede usar con cualquier función que sea *decreciente* en $[a, b]$ porque con esta suposición, $U(f; P)$ se puede obtener fácilmente de $L(f; P)$ (véase el problema 5.1.4).

El programa de computadora que corresponde al seudocódigo produce como salida los valores siguientes de las sumas inferior y superior:

$$\text{suma_inferior} = 0.74651, \quad \text{suma_superior} = 0.74714$$

En este momento, se le pide a usted que programe este experimento o uno parecido. El experimento muestra cómo la computadora puede imitar la definición abstracta de la integral de Riemann, al menos en casos en los que los números m_i y M_i se pueden obtener fácilmente. Otra conclusión que se puede obtener del experimento es que la traducción directa de una definición en un algoritmo de computadora puede dejar mucho que desechar *en precisión*. Con 999 evaluaciones de la función, el error absoluto aún es aproximadamente de 0.0003. Pronto veremos que algoritmos más complejos (como el de Romberg) mejoran esta situación dramáticamente.

Un buen valor aproximado para la integral de la ecuación (3) se puede calcular a partir de saber que esta integral está relacionada con la **función de error**

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Usando un software matemático adecuado obtenemos

$$\int_0^1 e^{-x^2} dx = \frac{1}{2} \sqrt{\pi} \operatorname{erf}(1) \approx 0.7468241330$$

Los sistemas de software matemático como Maple y Matlab tienen la función de error. Sin embargo, nos interesa el aprender acerca de algoritmos para aproximar integrales que sólo se pueden evaluar numéricamente.

En los problemas de este capítulo hemos usado diferentes integrales bien conocidas para exemplificar la integración numérica. Muchas de estas integrales han sido ampliamente investigadas y tabuladas. Los ejemplos son integrales elípticas, la integral seno, la integral de Fresnel, la integral logarítmica, la función de error y funciones Bessel. En el mundo real, cuando uno se enfrenta con una integral intimidatoria, la primera pregunta que surge es si ya se le ha estudiado y quizás tabulado. El primer lugar para buscar es el *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, editado por M. Abramowitz y I. Stegun [1964]. En el análisis numérico moderno, estas tablas son de uso limitado debido a la disponibilidad de paquetes de software como Matlab, Maple y Mathematica. No obstante, en raras ocasiones, se han encontrado problemas en los que se obtiene la respuesta equivocada cuando se usan dichos paquetes.

EJEMPLO 2 Si la integral

$$\int_0^\pi e^{\cos x} dx$$

se va a calcular con error absoluto menor que $\frac{1}{2} \times 10^{-3}$, y si estamos usando sumas superior e inferior con una partición uniforme, ¿cuántos subintervalos se necesitan?

Solución El integrando, $f(x) = e^{\cos x}$, es una función decreciente en el intervalo $[0, \pi]$. Por tanto, en las fórmulas para $U(f; P)$ y $L(f; P)$, tenemos

$$m_i = f(x_{i+1}) \quad \text{y} \quad M_i = f(x_i)$$

Sea que P denote la partición de $[0, \pi]$ por $n + 1$ puntos igualmente espaciados, $0 = x_0 < \dots < x_n = \pi$. Entonces habrán n subintervalos, todos de ancho π/n . Por tanto,

$$L(f; P) = \frac{\pi}{n} \sum_{i=0}^{n-1} m_i = \frac{\pi}{n} \sum_{i=0}^{n-1} f(x_{i+1}) \tag{4}$$

$$U(f; P) = \frac{\pi}{n} \sum_{i=0}^{n-1} M_i = \frac{\pi}{n} \sum_{i=0}^{n-1} f(x_i) \tag{5}$$

El valor correcto de la integral se encuentra en el intervalo entre $L(f; P)$ y $U(f; P)$. Tomamos el *punto medio* del intervalo como el mejor cálculo y así obtenemos un error de a lo más $\frac{1}{2}[U(f; P) - L(f; P)]$ —es decir, la longitud de la mitad del intervalo. Para encontrar el criterio de error impuesta en el problema, debemos tener

$$\frac{1}{2}[U(f; P) - L(f; P)] < \frac{1}{2} \times 10^{-3}$$

De las fórmulas (4) y (5) podemos calcular la diferencia entre las sumas superior e inferior. Esto conduce a $(\pi/n)(e^1 - e^{-1}) < 10^{-3}$. Con la ayuda de una calculadora, determine que n debe ser al menos 7385. ■

Por razones históricas, las fórmulas para aproximar las integrales definidas se llaman **reglas**. Las sumas superior e inferior dan lugar a las reglas de los rectángulos izquierda y derecha, la regla del punto medio, la regla del trapezio y muchas otras reglas, algunas de las cuales se encuentran en los problemas y capítulos subsecuentes de este libro. Una gran colección de estas **reglas de cuadratura** se pueden encontrar en Abramowitz y Stegun [1964], *Standard Mathematical Tables*, que tiene su origen en un proyecto de trabajo del gobierno de los Estados Unidos conducido durante la depresión de los 1930s.

La palabra **cuadratura** tiene varios significados tanto en matemática como en astronomía. En el diccionario, el primer significado matemático es el proceso de encontrar un cuadrado cuya área sea igual al área encerrada por una curva dada. El significado matemático general es el proceso de determinar el área de una superficie, especialmente la limitada por una curva. Usamos esta acepción principalmente para significar la aproximación del área bajo una curva usando un procedimiento de integración numérica.

Resumen

(1) Sea $P = \{a = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = b\}$ una **partición** del intervalo $[a, b]$, que lo divide en n subintervalos $[x_i, x_{i+1}]$. La **suma inferior** y la **suma superior** de f corresponden a la partición dada P son

$$L(f; P) = \sum_{i=0}^{n-1} m_i(x_{i+1} - x_i)$$

$$U(f; P) = \sum_{i=0}^{n-1} M_i(x_{i+1} - x_i)$$

donde m_i es el **límite inferior más grande** y M_i es el **límite superior más pequeño** de $f(x)$ en el subintervalo $[x_i, x_{i+1}]$, a saber,

$$m_i = \inf\{f(x) : x_i \leq x \leq x_{i+1}\}$$

$$M_i = \sup\{f(x) : x_i \leq x \leq x_{i+1}\}$$

(2) Tenemos

$$L(f; P) \leq \int_a^b f(x) dx \leq U(f; P)$$

Problemas 5.1

- ^a1. Si calculamos $\int_0^1 (x^2 + 2)^{-1} dx$ usando una suma inferior con la partición $P = \{0, \frac{1}{2}, 1\}$, ¿cuál es el resultado?
- 2. ¿Cuál es el resultado si calculamos $\int_1^2 x^{-1} dx$ usando la suma superior con la partición $P = \{1, \frac{3}{2}, 2\}$?
- 3. Calcule un valor aproximado de $\int_0^a [(e^x - 1)/x] dx$ para $a = 10^{-4}$ correcto con 14 lugares decimales (redondeado). *Sugerencia:* use series de Taylor.

4. Para una función decreciente $f(x)$ en un intervalo $[a, b]$ con n subintervalos uniformes, demuestre que la diferencia entre la suma superior y la suma inferior está dada por la expresión $[(b - a)/n][f(a) - f(b)]$.

5. (Continuación) Repita el problema anterior para una función creciente.

6. Si las sumas superior e inferior se usan con puntos regularmente espaciados para calcular $\int_2^5 (dx / \log x)$, ¿cuántos puntos se necesitan si se tiene una exactitud de $\frac{1}{2} \times 10^{-4}$?

7. Sea f una función creciente. Si la integral $\int_0^1 f(x) dx$ se calcula usando el método de sumas superior e inferior, tomando n puntos igualmente espaciados, ¿cuál es el peor error posible?

8. Si f es una función (estrictamente) creciente en $[a, b]$, y si $\alpha = f(a)$ y $\beta = f(b)$, entonces $f^{-1}(x)$ está bien definida para $\alpha \leq x \leq \beta$. Descubra la relación entre $\int_a^b f(x) dx$ y $\int_\alpha^\beta f^{-1}(x) dx$.

9. Demuestre que si $\theta_i \geq 0$ y $\sum_{i=0}^n \theta_i = 1$, entonces $\sum_{i=0}^n \theta_i a_i$ se encuentra entre el menor y el mayor de los números a_i .

10. Establezca la **regla del punto medio compuesto** para calcular una integral:

$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} (x_{i+1} - x_i) f\left[\frac{1}{2}(x_{i+1} + x_i)\right]$$

11. (Continuación) Encuentre la relación entre la regla del punto medio y las sumas superior e inferior.

12. (Continuación) Establezca que la **regla del punto medio compuesto para subintervalos iguales** está dada por

$$\int_a^b f(x) dx \approx h \sum_{i=0}^{n-1} f\left(x_i + \frac{1}{2}h\right)$$

donde $h = (b - a)/n$, $x_i = a + ih$ y $0 \leq i \leq n$.

Problemas de cómputo 5.1

1. Escriba un procedimiento de propósito general para calcular integrales de funciones decrecientes mediante el método de sumas superior e inferior con una partición uniforme. De al procedimiento la secuencia de llamado

real function *Integral(f, a, b, ε, n, suma_inferior, suma_superior)*

donde f es el nombre de la función, a y b son los puntos finales del intervalo y $ε$ es la tolerancia. El procedimiento determina n tal que $suma_superior - suma_inferior < 2ε$. El procedimiento regresa el promedio de *suma_superior* y *suma_inferior*. Pruebelo en la integral seno del siguiente problema de cómputo, usando $ε = \frac{1}{2} \times 10^{-5}$.

- a2.** Calcule la integral definida $\int_0^1 x^{-1} \sin x \, dx$ al calcular las sumas superior e inferior, usando 800 puntos en el intervalo. El integrando se define igual a 1 en $x = 0$. La función es decreciente y este hecho se debe demostrar con cálculo. (Para una función decreciente $f, f' < 0$).
- Nota:* la función

$$\text{Si}(x) = \int_0^x t^{-1} \sin t \, dt$$

es una importante función especial conocida como la **integral seno**. Ésta se representa con una serie de Taylor que converge para todos los valores reales o complejos de x . La forma más fácil de obtener esta serie es comenzar con la serie para $\sin t$, dividir entre t e integrar término por término:

$$\begin{aligned}\text{Si}(x) &= \int_0^x t^{-1} \sin t \, dt = \int_0^x \sum_{n=0}^{\infty} (-1)^n \frac{t^{2n}}{(2n+1)!} \, dt \\ &= \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!(2n+1)} = x - \frac{x^3}{18} + \frac{x^5}{600} - \frac{x^7}{35280} + \dots\end{aligned}$$

Esta serie es rápidamente convergente. Por ejemplo, de sólo los términos que se muestran, $\text{Si}(1)$ se calcula igual a 0.94608 27 con un error de, a lo más, cuatro unidades en el último dígito mostrado.

- 3. La integral logarítmica** es una función matemática especial definida por la ecuación

$$\text{li}(x) = \int_2^x \frac{dt}{\ln t}$$

Para x grandes, el número de enteros primos menores que o iguales a x se approxima por $\text{li}(x)$. Por ejemplo, hay 46 primos menores que 200 y $\text{li}(200)$ es más o menos 50. Encuentre $\text{li}(200)$ con tres cifras significativas por medio de las sumas superior e inferior. Determine el número de puntos de partición necesario *antes* de ejecutar el programa.

- a4.** De cálculo, la longitud de una curva es $\int_a^b \sqrt{1 + [f'(x)]^2} \, dx$, donde f es una función cuya gráfica es la curva en el intervalo $a \leq x \leq b$.
- Encuentre la longitud de la elipse $y^2 + 4x^2 = 1$. Use la simetría de la elipse.
 - Compruebe la aproximación numérica dada para la longitud de arco en el ejemplo de la introducción al principio del capítulo 3.
- 5.** Usando un sistema de software matemático que tenga la función de error erf, encuentre una aproximación numérica de $\int_0^1 e^{-x^2} \, dx$ con toda la precisión disponible. También, trace la gráfica de la función de error.
- 6.** (Continuación) Evalúe la integral $\int_0^1 e^{-x^2} \, dx$ usando una rutina de integración numérica en un sistema de software matemático como Matlab. Compare los resultados con los obtenidos antes.

5.2 Regla del trapecio

El siguiente método considerado es una mejora sobre el burdo método de la sección anterior. Además, es un importante ingrediente del algoritmo de Romberg de la siguiente sección.

Este método se llama la **regla del trapecio** y se basa en un cálculo del área bajo una curva usando trapecios. Nuevamente, el cálculo de $\int_a^b f(x) dx$ se realiza al dividir primero el intervalo $[a, b]$ en subintervalos de acuerdo con la partición $P = \{a = x_0 < x_1 < x_2 < \dots < x_n = b\}$. Para cada una de las particiones del intervalo (no es necesario que los puntos de partición x_i estén uniformemente espaciados), se obtiene un cálculo de la integral con la regla del trapecio. La denotamos por $T(f; P)$. La figura 5.2 muestra qué son los trapecios. Un trapecio típico tiene el subintervalo $[x_i, x_{i+1}]$ como su base y los dos lados verticales son $f(x_i)$ y $f(x_{i+1})$ (véase la figura 5.3). El área es igual a la base por la altura promedio y tenemos la **regla del trapecio básica** para el subintervalo $[x_i, x_{i+1}]$:

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx \frac{1}{2}(x_{i+1} - x_i)[f(x_i) + f(x_{i+1})]$$

Por tanto, el área total de todos los trapecios es

$$\int_a^b f(x) dx \approx \frac{1}{2} \sum_{i=0}^{n-1} (x_{i+1} - x_i)[f(x_i) + f(x_{i+1})]$$

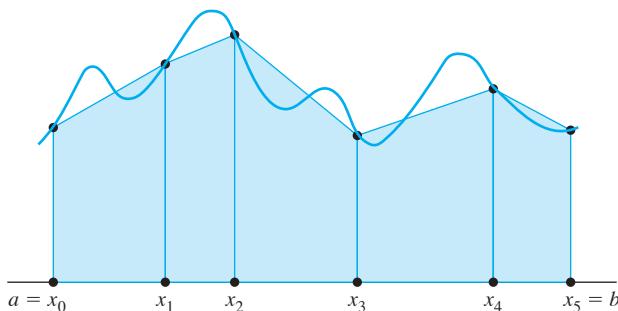


FIGURA 5.2
Regla del
trapecio

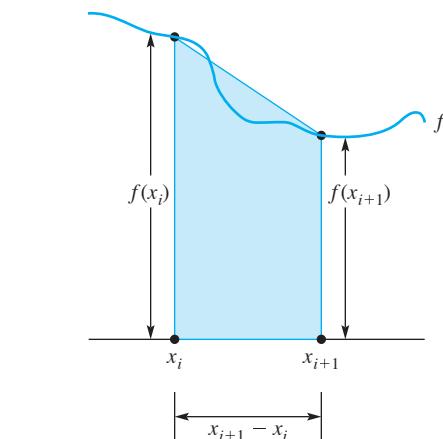


FIGURA 5.3
Trapecio típico

Esta fórmula se llama la **regla del trapecio compuesto**. La regla del trapecio compuesto es fácil de entender: en cada subintervalo $[x_i, x_{i+1}]$ multiplicamos $(x_{i+1} - x_i)$ veces el promedio de $f(x_i)$ y $f(x_{i+1})$.

Espaciado uniforme

En la práctica y en el algoritmo de Romberg (que se aborda en la sección siguiente), la regla del trapecio se usa con una partición *uniforme* del intervalo. Esto significa que los puntos de división x_i están igualmente espaciados: $x_i = a + ih$, donde $h = (b - a)/n$ y $0 \leq i \leq n$. Piense en h como el tamaño de paso en el proceso. En este caso, la fórmula para $T(f; P)$ se puede dar en forma más simple porque $x_{i+1} - x_i = h$. Así, obtenemos

$$T(f; P) = \frac{h}{2} \sum_{i=0}^{n-1} [f(x_i) + f(x_{i+1})]$$

Se debe enfatizar que para economizar la cantidad de aritmética, la fórmula preferible desde el punto de vista de los cálculos para la regla del trapecio compuesto es

$$\int_a^b f(x) dx \approx T(f; P) = h \left\{ \frac{1}{2}[f(x_0) + f(x_n)] + \sum_{i=1}^{n-1} f(x_i) \right\} \quad (1)$$

Aquí, hemos expandido la suma y agrupado los términos semejantes en la nueva suma. Para ejemplificar, consideremos la integral

$$\int_0^1 e^{-x^2} dx$$

que se approximó en la sección 5.1 mediante sumas inferior y superior. Aquí se presenta un pseudocódigo para la ecuación (1) con $n = 60$ y $f(x) = e^{-x^2}$:

```

program Trapecio
integer i; real h, suma, x
integer n ← 60; real a ← 0, b ← 1
h ← (b - a)/ n
suma ← ½[f(a) + f(b)]
for i = 1 to n - 1 do
    x ← a + i h
    suma ← suma + f(x)
end for
suma ← (suma)h
output suma
end Trapecio

real function f(x)
real x
f ← 1/ e-x2
end function f

```

La salida de la computadora para el valor aproximado de la integral es 0.74681.

EJEMPLO 1 Calcule

$$\int_0^1 (\sin x/x) dx$$

usando la regla del trapecio compuesto con seis puntos uniformes (compare con el problema de cómputo 5.1.2).

Solución Los valores de la función se arreglan en una tabla como se muestra a continuación:

x_i	$f(x_i)$
0.0	1.00000
0.2	0.99335
0.4	0.97355
0.6	0.94107
0.8	0.89670
1.0	0.84147

Observe que hemos asignado el valor $\sin x/x = 1$ en $x = 0$. Entonces

$$\begin{aligned} T(f; P) &= 0.2 \sum_{i=1}^4 f(x_i) + (0.1)[f(x_0) + f(x_5)] \\ &= (0.2)(3.80467) + (0.1)(1.84147) \\ &= 0.94508 \end{aligned}$$

Este resultado no es exacto para todos los dígitos que se muestran, como se podría esperar, ya que se utilizaron sólo cinco subintervalos. Usando software matemático obtenemos Si $(I) \approx 0.94608\ 30704$. (consulte problema de cómputo 5.1.2). Más tarde veremos cómo determinar un valor adecuado para n para obtener la exactitud deseada al usar la regla del trapecio. ■

Análisis de error

La siguiente tarea es analizar el error cometido al usar la regla del trapecio para calcular una integral. Estableceremos el siguiente resultado.

■ TEOREMA 2

Teorema de precisión de la regla del trapecio

Si f'' existe y es continua en el intervalo $[a, b]$ y si la regla del trapecio compuesto T con espaciamiento uniforme h se usa para calcular la integral $I = \int_a^b f(x) dx$, entonces para alguna ζ , en (a, b) ,

$$I - T = -\frac{1}{12}(b-a)h^2 f''(\zeta) = O(h^2)$$

Demostración El primer paso en el análisis es demostrar el resultado enunciado cuando $a = 0$, $b = 1$ y $h = 1$. En este caso, tenemos que demostrar que

$$\int_0^1 f(x) dx - \frac{1}{2}[f(0) + f(1)] = -\frac{1}{12} f''(\zeta) \quad (2)$$

Esto se establece fácilmente con la ayuda de la fórmula de error para la interpolación polinomial (véase la sección 4.2). Para usar esta fórmula, sea p el polinomio de grado 1 que interpola f en 0 y

1. Entonces p está dado por

$$p(x) = f(0) + [f(1) - f(0)]x$$

Por tanto,

$$\begin{aligned}\int_0^1 p(x) dx &= f(0) + \frac{1}{2}[f(1) - f(0)] \\ &= \frac{1}{2}[f(0) + f(1)]\end{aligned}$$

Por la fórmula de error que gobierna la interpolación polinomial [ecuación (2) de la sección 4.2], tenemos (aquí, por supuesto, $n = 1$, $x_0 = 0$ y $x_1 = 1$)

$$f(x) - p(x) = \frac{1}{2}f''[\xi(x)]x(x - 1)$$

donde $\xi(x)$ depende de x en $(0, 1)$. De la ecuación (3), se tiene que

$$\int_0^1 f(x) dx - \int_0^1 p(x) dx = \frac{1}{2} \int_0^1 f''[\xi(x)]x(x - 1) dx \quad (3)$$

Se puede demostrar que $f''[\xi(x)]$ es continua al resolver la ecuación (3) para $f''[\xi(x)]$ y comprobando la continuidad (véase el problema 4.2.12). Observe que $x(x - 1)$ no cambia de signo en el intervalo $[0, 1]$. Por tanto, por el **teorema del valor medio para integrales**,* hay un punto $x = s$ para el que

$$\begin{aligned}\int_0^1 f''[\xi(x)]x(x - 1) dx &= f''[\xi(s)] \int_0^1 x(x - 1) dx \\ &= -\frac{1}{6}f''(\xi)\end{aligned}$$

Poniendo todas estas ecuaciones juntas obtenemos la ecuación (2). De la ecuación (2), haciendo un cambio de variable, obtenemos la **regla del trapecio básica** con su término de error:

$$\int_a^b f(x) dx = \frac{b-a}{2}[f(a) + f(b)] - \frac{1}{12}(b-a)^3 f''(\xi) \quad (4)$$

Los detalles de esta son los siguientes: sea $g(t) = f(a + t(b-a))$ y $x = a + (b-a)t$. Así, conforme t recorre el intervalo $[0, 1]$, x recorre el intervalo $[a, b]$. También, $dx = (b-a)dt$, $g'(t) = f'[a + t(b-a)](b-a)$ y $g''(t) = f''[a + t(b-a)](b-a)^2$. Por tanto, por la ecuación (2),

$$\begin{aligned}\int_a^b f(x) dx &= (b-a) \int_0^1 f[a + t(b-a)] dt \\ &= (b-a) \int_0^1 g(t) dt \\ &= (b-a) \left\{ \frac{1}{2}[g(0) + g(1)] - \frac{1}{12}g''(\xi) \right\} \\ &= \frac{b-a}{2}[f(a) + f(b)] - \frac{(b-a)^3}{12}f''(\xi)\end{aligned}$$

***Teorema del valor medio para integrales:** sea f continua en $[a, b]$ y suponga que g es una función integrable de Riemann en $[a, b]$. Si $g(x) \geq 0$ en $[a, b]$, entonces existe un punto ξ tal que $a \leq \xi \leq b$ y $\int_a^b f(x)g(x) dx = f(\xi) \int_a^b g(x) dx$.

Esta es la regla del trapecio y el término de error para el intervalo $[a, b]$ con sólo un subintervalo, el cual abarca todo el intervalo. Así, el término de error es $\mathcal{O}(h^3)$, donde $h = b - a$. Aquí, ξ está en (a, b) .

Ahora sea el intervalo $[a, b]$ dividido en n subintervalos iguales por puntos x_0, x_1, \dots, x_n con espaciamiento h . Aplicando la fórmula (4) para el subintervalo $[x_i, x_{i+1}]$, tenemos

$$\int_{x_i}^{x_{i+1}} f(x) dx = \frac{h}{2}[f(x_i) + f(x_{i+1})] - \frac{1}{12}h^3 f''(\xi_i) \quad (5)$$

donde $x_i < \xi_i < x_{i+1}$. Usamos este resultado sobre el intervalo $[a, b]$ para obtener la **regla del trapecio compuesto**

$$\begin{aligned} \int_a^b f(x) dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx \\ &= \frac{h}{2} \sum_{i=0}^{n-1} [f(x_i) + f(x_{i+1})] - \frac{h^3}{12} \sum_{i=0}^{n-1} f''(\xi_i) \end{aligned} \quad (6)$$

El término final en la ecuación (6) es el término de error y se puede simplificar en la siguiente forma: puesto que $h = (b - a)/n$, el término de error para la regla del trapecio compuesto es

$$-\frac{h^3}{12} \sum_{i=0}^{n-1} f''(\xi_i) = -\frac{b-a}{12} h^2 \left[\frac{1}{n} \sum_{i=0}^{n-1} f''(\xi_i) \right] = -\frac{b-a}{12} h^2 f''(\zeta)$$

Aquí, hemos razonado que el promedio $[1/n] \sum_{i=0}^{n-1} f''(\xi_i)$ se encuentra entre los valores menor y mayor de f'' en el intervalo (a, b) . Por tanto, por el **teorema del valor intermedio**,* esto es $f''(\zeta)$ para algún punto ζ en (a, b) . Esto completa nuestra demostración de la fórmula de error. ■

EJEMPLO 2 Use una serie de Taylor para representar el error en la regla del trapecio básica por una serie infinita.

Solución La ecuación (4) es equivalente a

$$\int_a^{a+h} f(x) dx = \frac{h}{2}[f(a) + f(a+h)] - \frac{1}{12}h^3 f''(\xi)$$

Sea

$$F(t) = \int_a^t f(x) dx$$

La serie de Taylor para F es

$$F(a+h) = F(a) + hF'(a) + \frac{h^2}{2}F''(a) + \frac{h^3}{3!}F'''(a) + \cdots$$

***Teorema del valor intermedio:** si la función g es continua en un intervalo $[a, b]$, entonces para cada c entre $g(a)$ y $g(b)$, hay un punto ξ en $[a, b]$ para el que $g(x) = c$.

Por el teorema fundamental del cálculo (p. 181), $F' = f$, y observamos que $F(a) = 0$, $F'' = f'$, $F''' = f''$ y así sucesivamente. Por tanto, tenemos

$$\int_a^{a+h} f(x) dx = hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{3!}f''(a) + \dots$$

La serie de Taylor para f es

$$f(a + h) = f(a) + hf'(a) + \frac{h^2}{2}f''(a) + \frac{h^3}{3!}f'''(a) + \dots$$

Sumando $f(a)$ a ambos lados de esta ecuación y después multiplicando por $h/2$, obtenemos

$$\frac{h}{2}[f(a) + f(a + h)] = hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{4}f''(a) + \dots$$

Restando queda

$$\int_a^{a+h} f(x) dx - \frac{h}{2}[f(a) + f(a + h)] = -\frac{1}{12}h^3f''(a) + \dots$$
■

Aplicación de la fórmula de error

¿Cómo puede usarse una fórmula de error de modo similar a la que acabamos de deducir? Nuestra primera aplicación es predecir cuán pequeña debe ser h para lograr una precisión especificada en la regla del trapecio.

EJEMPLO 3 Si la regla del trapecio compuesto se usa para calcular

$$\int_0^1 e^{-x^2} dx$$

con un error de a lo más $\frac{1}{2} \times 10^{-4}$, ¿cuántos puntos se deben usar?

Solución La fórmula de error es

$$-\frac{b-a}{12}h^2 f''(\xi)$$

En este ejemplo, $f(x) = e^{-x^2}$, $f'(x) = -2xe^{-x^2}$ y $f''(x) = (4x^2 - 2)e^{-x^2}$. Así, $|f''(x)| \leq 2$ en el intervalo $[0, 1]$, y el error en valor absoluto no será mayor que $\frac{1}{6}h^2$. Para tener un error de a lo más $\frac{1}{2} \times 10^{-4}$, requerimos

$$\frac{1}{6}h^2 \leq \frac{1}{2} \times 10^{-4} \quad \text{o} \quad h \leq 0.01732$$

En este ejemplo, $h = 1/n$, así que requerimos $n \geq 58$. Por tanto, 59 o más puntos ciertamente producirán la exactitud deseada.

■

EJEMPLO 4 ¿Cuántos subintervalos se necesitan para aproximar

$$\int_0^1 \frac{\sin x}{x} dx$$

con error que no exceda a $\frac{1}{2} \times 10^{-5}$ usando la regla del trapecio compuesto? Aquí, el integrando $f(x) = x^{-1} \sin x$ está definido igual a 1 cuando x es 0.

Solución Deseamos establecer un límite en $|f''(x)|$ para x en el rango $[0, 1]$. Obtener derivadas de la manera usual no es satisfactorio, ya que cada término contiene x con una potencia negativa y es difícil encontrar un límite superior en $|f''(x)|$. Sin embargo, usando series de Taylor tenemos

$$\begin{aligned} f(x) &= 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \frac{x^8}{9!} - \dots \\ f'(x) &= -\frac{2x}{3!} + \frac{4x^3}{5!} - \frac{6x^5}{7!} + \frac{8x^7}{9!} - \dots \\ f''(x) &= -\frac{2}{3!} + \frac{3 \times 4x^2}{5!} - \frac{5 \times 6x^4}{7!} + \frac{7 \times 8x^6}{9!} - \dots \end{aligned}$$

Así, en el intervalo $[0, 1]$, $|f''(x)|$ no puede ser mayor que $\frac{1}{2}$ porque

$$\frac{2}{3!} + \frac{3 \times 4}{5!} + \frac{5 \times 6}{7!} + \frac{7 \times 8}{9!} + \dots < \frac{1}{3} + \frac{1}{10} + \frac{1}{24} \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots \right) < \frac{1}{2}$$

Por tanto, el término de error $|(b-a)h^2 f''(\xi)/12|$ no puede ser mayor que $h^2/24$. Para que esto sea menor que $\frac{1}{2} \times 10^{-5}$, es suficiente tomar $h < \sqrt{1.2} \times 10^{-2}$ o $n > (1/\sqrt{1.2})10^2$. Este análisis nos induce a tomar 92 subintervalos.

Fórmula recursiva del trapecio para subintervalos iguales

En la siguiente sección requerimos una fórmula para la regla del trapecio compuesto cuando el intervalo $[a, b]$ está subdividido en 2^n partes iguales. Por la fórmula (1), tenemos

$$\begin{aligned} T(f; P) &= h \sum_{i=1}^{n-1} f(x_i) + \frac{h}{2} [f(x_0) + f(x_n)] \\ &= h \sum_{i=1}^{n-1} f(a + ih) + \frac{h}{2} [f(a) + f(b)] \end{aligned}$$

Si ahora remplazamos n con 2^n y usamos $h = (b-a)/2^n$, la fórmula anterior será

$$R(n, 0) = h \sum_{i=1}^{2^n-1} f(a + ih) + \frac{h}{2} [f(a) + f(b)] \quad (7)$$

Aquí, hemos introducido la notación que se usará en la sección 5.3 en el algoritmo de Romberg, a saber, $R(n, 0)$, y que denota el resultado de aplicar la regla del trapecio compuesto con 2^n subintervalos iguales.

En el algoritmo de Romberg, también será necesario tener un medio para calcular $R(n, 0)$ a partir de $R(n-1, 0)$ sin implicar evaluaciones innecesarias de f . Por ejemplo, el cálculo de $R(2, 0)$ utiliza los valores de f en los cinco puntos $a, a + (b-a)/4, a + 2(b-a)/4, a + 3(b-a)/4$ y b . Para calcular $R(3, 0)$, necesitamos valores de f en estos cinco puntos, así como en cuatro puntos nuevos: $a + (b-a)/8, a + 3(b-a)/8, a + 5(b-a)/8$ y $a + 7(b-a)/8$ (véase la figura 5.4). El cálculo debe aprovechar el resultado previamente obtenido. Ahora se explicará cómo hacerlo.

Si $R(n-1, 0)$ se ha calculado y $R(n, 0)$ se está calculando, usamos la identidad

$$R(n, 0) = \frac{1}{2} R(n-1, 0) + \left[R(n, 0) - \frac{1}{2} R(n-1, 0) \right]$$

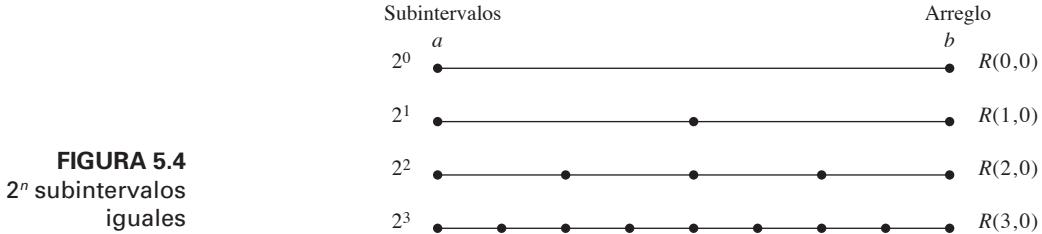


FIGURA 5.4
 2^n subintervalos
iguales

Es deseable calcular la expresión entre corchetes con tan poco trabajo adicional como sea posible. Fijando $h = (b - a)/2^n$ para el análisis y haciendo

$$C = \frac{h}{2} [f(a) + f(b)]$$

tenemos, de la ecuación (7),

$$R(n, 0) = h \sum_{i=1}^{2^n-1} f(a + ih) + C \quad (8)$$

$$R(n-1, 0) = 2h \sum_{j=1}^{2^{n-1}-1} f(a + 2jh) + 2C \quad (9)$$

Observe que los subintervalos para $R(n-1, 0)$ son del *doble* del tamaño de los de $R(n, 0)$. Ahora, de las ecuaciones (8) y (9) tenemos

$$\begin{aligned} R(n, 0) - \frac{1}{2} R(n-1, 0) &= h \sum_{i=1}^{2^n-1} f(a + ih) - h \sum_{j=1}^{2^{n-1}-1} f(a + 2jh) \\ &= h \sum_{k=1}^{2^{n-1}} f[a + (2k-1)h] \end{aligned}$$

Aquí, hemos considerado el hecho de que cada término en la primera suma que corresponde a un valor *par* de i es *eliminado* por un término de la segunda suma. Esto deja sólo términos que corresponden a valores *impares* de i .

Para resumir:

■ TEOREMA 2

Fórmula recursiva del trapecio

Si $R(n-1, 0)$ está disponible, entonces $R(n, 0)$ se puede calcular con la fórmula

$$R(n, 0) = \frac{1}{2} R(n-1, 0) + h \sum_{k=1}^{2^{n-1}} f[a + (2k-1)h] \quad (n \geq 1) \quad (10)$$

usando $h = (b - a)/2^n$. Aquí, $R(0, 0) = \frac{1}{2}(b - a)[f(a) + f(b)]$.

Esta fórmula nos permite calcular una sucesión de aproximaciones para una integral definida usando la regla del trapecio sin evaluar de nuevo el integrando en los puntos donde ya ha sido evaluado.

Integración multidimensional

Aquí daremos una breve explicación de la *integración numérica multidimensional*. Por simplicidad, ejemplificaremos con la regla del trapecio para el intervalo $[0, 1]$, usando $n + 1$ puntos igualmente espaciados. El tamaño de paso es, por tanto, $h = 1/n$. La regla del trapecio compuesto es entonces

$$\int_0^1 f(x) dx \approx \frac{1}{2h} \left[f(0) + 2 \sum_{i=1}^{n-1} f\left(\frac{i}{n}\right) + f(1) \right]$$

Escribimos esto en la forma

$$\int_0^1 f(x) dx \approx \sum_{i=0}^n C_i f\left(\frac{i}{n}\right)$$

donde

$$C_i = \begin{cases} 1/(2h), & i = 0 \\ 1/h, & 0 < i < n \\ 1/(2h), & i = n \end{cases}$$

El error es $\mathcal{O}(h^2) = \mathcal{O}(n^{-2})$ para funciones que tengan una segunda derivada continua.

Si uno se enfrenta con una **integración bidimensional sobre un cuadrado unitario** entonces la regla del trapecio se puede aplicar dos veces:

$$\begin{aligned} \int_0^1 \int_0^1 f(x, y) dx dy &\approx \int_0^1 \sum_{a_1=0}^n C_{a_1} f\left(\frac{a_1}{n}, y\right) dy \\ &= \sum_{a_1=0}^n C_{a_1} \int_0^1 f\left(\frac{a_1}{n}, y\right) dy \\ &\approx \sum_{a_1=0}^n C_{a_1} \sum_{a_2=0}^n C_{a_2} f\left(\frac{a_1}{n}, \frac{a_2}{n}\right) \\ &= \sum_{a_1=0}^n \sum_{a_2=0}^n C_{a_1} C_{a_2} f\left(\frac{a_1}{n}, \frac{a_2}{n}\right) \end{aligned}$$

El error aquí es de nuevo $\mathcal{O}(h^2)$, ya que cada una de las dos aplicaciones de la regla del trapecio implica un error de $\mathcal{O}(h^2)$.

De la misma manera, podemos integrar una función de k variables. La notación adecuada es el vector $x = (x_1, x_2, \dots, x_k)^T$ para la variable independiente. Ahora la región se toma como un cubo k dimensional $[0, 1]^k \equiv [0, 1] \times [0, 1] \times \dots \times [0, 1]$. Entonces obtenemos una **regla de integración numérica multidimensional**

$$\int_{[0,1]^k} f(\mathbf{x}) d\mathbf{x} \approx \sum_{a_1=0}^n \sum_{a_2=0}^n \dots \sum_{a_k=0}^n C_{a_1} C_{a_2} \dots C_{a_k} f\left(\frac{a_1}{n}, \frac{a_2}{n}, \dots, \frac{a_k}{n}\right)$$

El error aún es $\mathcal{O}(h^2) = \mathcal{O}(n^{-2})$, siempre que f tenga derivadas parciales continuas $\partial^2 f / \partial x_i^2$.

Además del error implicado, uno debe considerar el esfuerzo, o trabajo, requerido para lograr un nivel de exactitud deseado. El trabajo en el caso de una variable es $\mathcal{O}(n)$. En el caso de dos variables es $\mathcal{O}(n^2)$ y es $\mathcal{O}(n^k)$ para k variables. El error, ahora expresado como una función del

número de nodos $N = n^k$, es

$$\mathcal{O}(h^2) = \mathcal{O}(n^{-2}) = \mathcal{O}\left((n^k)^{-2/k}\right) = \mathcal{O}(N^{-2/k})$$

Así, la calidad de la aproximación numérica de la integral decrece muy rápidamente conforme aumenta el número de variables k . Expresado en otros términos, si se mantiene un orden constante de exactitud mientras que el número de variables k aumenta, el número de nodos debe aumentar como n^k . Estas observaciones indican por qué el método de Monte Carlo para integración numérica se vuelve más atractivo para integración de alta dimensión. (Este tema se analiza en el capítulo 13.)

Resumen

(1) Para calcular $\int_a^b f(x) dx$, divida el intervalo $[a, b]$ en subintervalos de acuerdo con la partición $P = \{a = x_0 < x_1 < x_2 < \dots < x_n = b\}$. La **regla del trapecio básica** para el subintervalo $[x_i, x_{i+1}]$ es

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx A_i = \frac{1}{2}(x_{i+1} - x_i)[f(x_i) + f(x_{i+1})]$$

donde el error es $-\frac{1}{12}(x_{i+1} - x_i)^3 f''(\xi_i)$. La **regla del trapecio compuesto** es

$$\int_a^b f(x) dx \approx T(f; P) = \sum_{i=0}^{n-1} A_i = \frac{1}{2} \sum_{i=0}^{n-1} (x_{i+1} - x_i)[f(x_i) + f(x_{i+1})]$$

donde el error es $-\frac{1}{12} \sum_{i=1}^n (x_{i+1} - x_i)^2 f''(\xi_i)$.

(2) Para un espaciamiento uniforme de nodos en el intervalo $[a, b]$, hacemos $x_i = a + ih$, donde $h = (b - a)/n$ y $0 \leq i \leq n$. La **regla del trapecio compuesto con espaciamiento uniforme** es

$$\int_a^b f(x) dx \approx T(f; P) = \frac{h}{2}[f(x_0) + f(x_n)] + h \sum_{i=1}^{n-1} f(x_i)$$

donde el error es $-\frac{1}{12}(b - a)^2 f''(\zeta)$.

(3) Para espaciamiento uniforme de nodos en el intervalo $[a, b]$ con 2^n subintervalos, hacemos $h = (b - a)/2^n$ y tenemos

$$\begin{cases} R(0, 0) = \frac{1}{2}(b - a)[f(a) + f(b)] \\ R(n, 0) = h \sum_{i=1}^{2^n-1} f(a + ih) + \frac{h}{2}[f(a) + f(b)] \end{cases}$$

Podemos calcular la primera columna del arreglo $R(n, 0)$ recursivamente con la **fórmula recursiva del trapecio**:

$$R(n, 0) = \frac{1}{2}R(n-1, 0) + h \sum_{k=1}^{2^{n-1}} f[a + (2k-1)h]$$

(4) Para la **integración bidimensional sobre el cuadrado unitario** se puede aplicar dos veces la regla del trapecio:

$$\int_0^1 \int_0^1 f(x, y) dx dy \approx \sum_{\alpha_1=0}^n \sum_{\alpha_2=0}^n C_{\alpha_1} C_{\alpha_2} f\left(\frac{\alpha_1}{n}, \frac{\alpha_2}{n}\right)$$

con error $\mathcal{O}(h^2)$. Para un cubo k -dimensional $[0, 1]^k = [0, 1] \times [0, 1] \times \cdots \times [0, 1]$, una **regla de integración numérica multidimensional** es

$$\int_{[0,1]^k} f(\mathbf{x}) d\mathbf{x} \approx \sum_{\alpha_1=0}^n \sum_{\alpha_2=0}^n \cdots \sum_{\alpha_k=0}^n C_{\alpha_1} C_{\alpha_2} \cdots C_{\alpha_k} f\left(\frac{\alpha_1}{n}, \frac{\alpha_2}{n}, \dots, \frac{\alpha_k}{n}\right)$$

con error $\mathcal{O}(h^2) = \mathcal{O}(n^{-2})$.

Problemas 5.2

a1. ¿Cuál es el valor numérico de la regla del trapecio compuesto aplicada a la función recíproca $f(x) = x^{-1}$ usando los puntos $1, \frac{4}{3}$ y 2 ?

a2. Calcule un valor aproximado de $\int_0^1 (x^2 + 1)^{-1} dx$ usando la regla del trapecio compuesto con tres puntos. Despues compare con el valor real de la integral. Entonces, determine la fórmula de error y compruebe numéricamente un límite superior en ésta.

3. (Continuación) Ya ha calculado $R(1, 0)$ en el problema anterior, calcule $R(2, 0)$ usando la fórmula (10).

4. Obtenga un límite superior en el error absoluto cuando calculamos $\int_0^6 \sin x^2 dx$ usando la regla del trapecio compuesto con 101 puntos igualmente espaciados.

5. Si la regla del trapecio compuesto se usa para calcular $\int_{-1}^2 \sin x dx$ con $h = 0.01$, dé un límite objetivo para el error.

a6. Considere la función $f(x) = |x|$ en el intervalo $[-1, 1]$. Calcule los resultados de aplicar las reglas siguientes para aproximar $\int_{-1}^1 f(x) dx$. Explique las diferencias en los resultados y compare con la solución verdadera. Use las

- a.** sumas inferiores **b.** sumas superiores **c.** regla del trapecio compuesto

con espaciamientos uniformes $h = 2, 1, \frac{1}{2}, \frac{1}{4}$.

a7. ¿Cuán largo debe ser n si la regla del trapecio compuesto en la ecuación (1) se está usando para calcular $\int_0^{\pi} \sin x dx$ con error $\leq 10^{-12}$? ¿Calcularemos que es muy grande o muy pequeño?

8. ¿Qué fórmula resulta al usar la regla del trapecio compuesto en $f(x) = x^2$, con intervalo $[0, 1]$ y $n + 1$ puntos igualmente espaciados? Simplifique su resultado usando el hecho de que $1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{1}{6}n(2n+1)(n+1)$. Muestre que conforme $M \rightarrow \infty$, el cálculo trapezoidal converge al valor correcto, $\frac{1}{3}$.

9. Pruebe que si una función es cóncava hacia abajo, entonces la regla del trapecio subestima la integral.

- 10.** Calcule dos valores aproximados para $\int_1^2 dx/x^2$ usando $h = \frac{1}{2}$ con sumas inferiores y la regla del trapecio compuesto.
- 11.** Considere $\int_1^2 dx/x^3$. ¿Cuál es el resultado de usar la regla del trapecio compuesto con los puntos de partición $1, \frac{3}{2}$ y 2 ?
- 12.** Si la regla del trapecio compuesto se usa con $h = 0.01$ para calcular $\int_2^5 \sin x dx$, ¿qué valor numérico no excederá el error? (Use el valor absoluto de error.) Dé la mejor respuesta basada en la fórmula de error.
- 13.** Aproxime $\int_0^2 2^x dx$ usando la regla del trapecio compuesto con $h = \frac{1}{2}$.
- 14.** Considere $\int_0^1 dx/(x^2 + 2)$. ¿Cuál es el resultado de usar la regla del trapecio compuesto con $0, \frac{1}{2}$ y 1 como puntos de partición?
- 15.** ¿Cuál es un límite razonable en el error cuando usamos la regla del trapecio compuesto en $\int_0^4 \cos x^3 dx$ tomando 201 puntos igualmente espaciados (incluidos los puntos finales)?
- 16.** Queremos aproximar $\int_1^2 f(x) dx$ a partir de la tabla de valores

x	1	$\frac{5}{4}$	$\frac{3}{2}$	$\frac{7}{4}$	2
$f(x)$	10	8	7	6	5

Calcule una estimación con la regla del trapecio compuesto. ¿Se pueden calcular sumas superior e inferior a partir de los datos dados?

- 17.** Considere la integral $I(h) \equiv \int_a^{a+h} f(x) dx$. Establezca una expresión para el término de error para cada una de las siguientes reglas:
- a.** $I(h) \approx hf(a + h)$ **b.** $I(h) \approx hf(a + h) - \frac{1}{2}h^2 f'(a)$
c. $I(h) \approx hf(a)$ **d.** $I(h) \approx hf(a) - \frac{1}{2}h^2 f'(a)$

Para cada una, determine la regla general correspondiente y los términos de error para la integral $\int_a^b f(x) dx$, donde la partición es uniforme; es decir, $x_i = a + ih$ y $h = (b - a)/n$ para $0 \leq i \leq n$.

- 18.** Obtenga las siguientes expresiones para los términos de error para la **regla del punto medio**

a. $\int_a^{a+\frac{1}{2}h} f(x) dx \approx hf\left(a + \frac{1}{2}h\right)$ (un subintervalo)

b. $\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} h_i f\left(x_i + \frac{1}{2}h_i\right)$ (n subintervalos iguales)

c. $\int_a^b f(x) dx \approx h \sum_{i=0}^{n-1} f\left[a + \left(i + \frac{1}{2}\right)h\right]$ (n subintervalos uniformes)

donde $h_i = x_{i+1} - x_i$ y $h = (b - a)/n$. (La regla del punto medio se introdujo en los problemas 5.1.10–5.1.12.)

- 19.** Demuestre que existen coeficientes w_0, w_1, \dots, w_n que dependen de x_0, x_1, \dots, x_n y de a, b tal que

$$\int_a^b p(x) dx = \sum_{i=0}^n w_i p(x_i)$$

para todos los polinomios p de grado $\leq n$. *Sugerencia:* use la forma de Lagrange de los polinomios de interpolación de la sección 4.1.

- 20.** Demuestre que cuando se aplica la regla del trapecio compuesto a $\int_a^b e^x dx$ usando puntos igualmente espaciados, el error relativo es exactamente $1 - (h/2) - [h/(e^k - 1)]$.

- 21.** Sea f una función decreciente en $[a, b]$. Sea P una partición del intervalo. Demuestre que

$$T(f; P) = \frac{1}{2}[L(f; P) + U(f; P)]$$

donde T , L y U son la regla del trapecio, las sumas inferiores y las sumas superiores, respectivamente.

- 22.** Demuestre que para cualquier función f y cualquier partición P ,

$$L(f; P) \leq T(f; P) \leq U(f; P)$$

- 23.** Sea f una función continua y sea P_n , para $n = 0, 1, \dots$, particiones de $[a, b]$ tales que el ancho del subintervalo más largo en P_n converge a cero cuando $n \rightarrow \infty$. Demuestre que $T(f; P_n)$ converge a $\int_a^b f(x) dx$ cuando $n \rightarrow \infty$. *Sugerencia:* use el problema anterior y hechos conocidos acerca de las sumas superior e inferior.

- 24.** Dé un ejemplo de una función f y una partición P para la cual $L(f; P)$ es un mejor cálculo de $\int_a^b f(x) dx$ que $T(f; P)$.

- 25.** Se dice que una función es **convexa** si su gráfica entre cualesquiera dos puntos se encuentra debajo de la cuerda dibujada entre esos dos puntos. ¿Cuál es la relación de $L(f; P)$, $U(f; P)$, $T(f; P)$ y $\int_a^b f(x) dx$ para esa función?

- 26.** ¿Cuán grande debe ser n si se usa la regla del trapecio compuesto con subintervalos iguales para calcular a $\int_0^2 e^{-x^2} dx$ con un error que no excede a 10^{-6} ? Primero encuentre un cálculo burdo de n usando la fórmula de error. Después determine el menor valor posible para n .

- 27.** Demuestre que

$$\int_a^b f(x) dx - \frac{b-a}{2}[f(a) + f(b)] = -\sum_{k=3}^{\infty} \frac{k-2}{2 \times k!} (b-a)^k f^{(k-1)}(a)$$

- 28.** La **regla del rectángulo (izquierdo) compuesto** para integración numérica es parecida a las sumas superior e inferior pero más simple:

$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i)$$

Aquí, la partición es $P = \{a = x_0 < x_1 < x_2 < \dots < x_n = b\}$. Demuestre que la regla del rectángulo converge a la integral cuando $n \rightarrow \infty$.

- 29.** (Continuación) La **regla del rectángulo compuesto con espacioamiento uniforme** se escribe como sigue:

$$\int_a^b f(x) dx \approx h \sum_{i=0}^{n-1} f(x_i)$$

donde $h = (b - a)/n$ y $x_i = a + ih$ para $0 \leq i \leq n$. Encuentre una expresión para el error implicado en esta última fórmula.

- 30.** De los dos problemas anteriores, la regla del rectángulo básico para un solo intervalo está dada por

$$\int_a^b f(x) dx = (b - a)f(a) + \frac{1}{2}(b - a)^2 f'(\xi)$$

Establezca la regla del rectángulo y su término de error cuando el intervalo $[a, b]$ se partitiona en 2^n subintervalos uniformes, cada uno de ancho h . Simplifique los resultados.

- 31.** En la regla del trapecio compuesto, no es necesario que el espaciamiento sea uniforme. Establezca la fórmula

$$\int_a^b f(x) dx \approx \frac{1}{2} \sum_{i=1}^{n-1} (h_{i-1} + h_i) f(x_i) + \frac{1}{2} [h_0 f(x_0) + h_{n-1} f(x_n)]$$

donde $h_i = x_{i+1} - x_i$ y $a = x_0 < x_1 < x_2 < \dots < x_n = b$.

- 32.** (Continuación) Establezca la siguiente fórmula de error para la **regla del trapecio compuesto con espaciamiento desigual** de puntos:

$$\int_a^b f(x) dx = \sum_{i=0}^{n-1} \frac{h_i}{2} [f(x_i) + f(x_{i+1})] - \frac{1}{12}(b - a)h^2 f''(\xi)$$

donde $\xi \in (a, b)$, $h_i = x_{i+1} - x_i$, y $\min_i h_i \leq h \leq \max_i h_i$. (La regla del trapecio compuesto con espaciamiento no uniforme se introdujo en el problema anterior).

- 33.** Cuántos puntos debemos usar en la regla del trapecio en el cálculo de un valor aproximado de $\int_0^1 e^{x^2} dx$ si la respuesta está dentro de 10^{-6} del valor correcto? *Sugerencia:* recuerde la fórmula de error para la regla del trapecio: $-\frac{1}{12}h^2(b - a)f''(\xi)$. Puede usar un cálculo burdo, como $2 < e < 3$. Explique qué está haciendo. Por último, queremos un valor adecuado de n , el número de puntos.

Problemas de cómputo 5.2

- 1.** Escriba

real function Trapecio-Regular (f, a, b, n)

para calcular $\int_a^b f(x) dx$ usando la regla del trapecio compuesto con n subintervalos iguales.

- 2.** (Continuación) Pruebe el código escrito en el problema de cómputo anterior en las siguientes funciones. En cada caso, compare con la respuesta correcta.

$$\text{a. } \int_0^\pi \sin x dx \quad \text{b. } \int_0^1 e^x dx \quad \text{c. } \int_0^1 \arctan x dx$$

- 3.** Calcule p a partir de una integral de la forma $c \int_a^b dx / (1 + x^2)$.

- 4.** Calcule un valor aproximado para la integral $\int_0^{0.8} (\sin x / x) dx$.

5. Calcule estas integrales usando valores pequeños y grandes para los límites inferior y superior y aplicando un método numérico. Después calcúlelas haciendo primero el cambio de variable indicado.

- $\int_0^\infty e^{-x^{2/2}} dx = \sqrt{\frac{\pi}{2}}$, usando $x = -\ln t$ **(integral gaussiana/probabilidad)**
- $\int_0^\infty x^{-1} \sin x dx = \frac{\pi}{2}$, usando $x = t^{-1}$ **(integral seno)**
- $\int_0^\infty \sin x^2 dx = \frac{1}{2} \sqrt{\frac{\pi}{2}}$, usando $x = \tan t$ **(integral seno de Fresnel)**

Aquí y en otras partes hemos usado diferentes integrales bien conocidas como ejemplos de prueba de esquemas de integración numérica. Algunas de estas integrales están tabuladas y se pueden encontrar en las tablas en Abramowitz y Stegun [1964].

- Usando una rutina de integración numérica en un sistema de software matemático como Matlab, encuentre un valor aproximado para la integral $\sin x / x$. Compare el valor aproximado que se obtiene con el valor de $\text{Si}(l)$ si el sistema contiene esta función. Haga una gráfica del integrando.
- Use la regla del trapecio compuesto con 59 subintervalos para comprobar numéricamente que la aproximación obtenida concuerda con los resultados del ejemplo 3.
- Usando un sistema de software matemático, compruebe la aproximación numérica de la integral dada en el ejemplo que viene al empezar el capítulo.

5.2 Algoritmo de Romberg

Descripción

El *algoritmo de Romberg* produce un arreglo triangular de números, los cuales son cálculos numéricos de la integral definida $\int_a^b f(x) dx$. El arreglo se denota aquí con la notación

$$\begin{array}{ccccccc}
 R(0,0) & & & & & & \\
 R(1,0) & R(1,1) & & & & & \\
 R(2,0) & R(2,1) & R(2,2) & & & & \\
 R(3,0) & R(3,1) & R(3,2) & R(3,3) & & & \\
 \vdots & \vdots & \vdots & \vdots & \ddots & & \\
 R(n,0) & R(n,1) & R(n,2) & R(n,3) & \cdots & R(n,n) &
 \end{array}$$

La primera columna de esta tabla tiene cálculos de la integral obtenidos con la fórmula recursiva del trapecio con valores decrecientes del tamaño de paso. Explícitamente, $R(n, 0)$ es el resultado de aplicar la regla del trapecio con 2^n subintervalos iguales. El primero de ellos, $R(0, 0)$, se obtuvo con sólo un trapecio:

$$R(0,0) = \frac{1}{2}(b-a)[f(a) + f(b)]$$

De manera similar, $R(1, 0)$ se obtuvo con dos trapecios:

$$\begin{aligned} R(1, 0) &= \frac{1}{4}(b-a) \left[f(a) + f\left(\frac{a+b}{2}\right) \right] + \frac{1}{4}(b-a) \left[f\left(\frac{a+b}{2}\right) + f(b) \right] \\ &= \frac{1}{4}(b-a)[f(a) + f(b)] + \frac{1}{2}(b-a) f\left(\frac{a+b}{2}\right) \\ &= \frac{1}{2}R(0, 0) + \frac{1}{2}(b-a) f\left(\frac{a+b}{2}\right) \end{aligned}$$

Estas fórmulas concuerdan con las que se desarrollaron en la sección anterior. En particular, observe que $R(n, 0)$ se obtiene fácilmente de $R(n-1, 0)$ si se usa la ecuación (10) de la sección 5.2; es decir,

$$R(n, 0) = \frac{1}{2}R(n-1, 0) + h \sum_{k=1}^{2^{n-1}} f[a + (2k-1)h] \quad (1)$$

donde $h = (b-a)/2^n$ y $n \geq 1$.

A partir de la segunda columna el arreglo de Romberg se generó con la fórmula de extrapolación

$$R(n, m) = R(n, m-1) + \frac{1}{4^m - 1} [R(n, m-1) - R(n-1, m-1)] \quad (2)$$

con $n \geq 1$ y $m \geq 1$. Esta fórmula se deducirá después usando la teoría de extrapolación de Richardson expuesta en la sección 4.3.

EJEMPLO 1 Si $R(4, 2) = 8$ y $R(3, 2) = 1$, ¿a qué es igual $R(4, 3)$?

Solución De la ecuación (2), tenemos

$$\begin{aligned} R(4, 3) &= R(4, 2) + \frac{1}{63} [R(4, 2) - R(3, 2)] \\ &= 8 + \frac{1}{63} (8 - 1) = \frac{73}{9} \end{aligned}$$



Seudocódigo

Ahora el objetivo es desarrollar fórmulas de cálculo para el **algoritmo de Romberg**. Al remplazar n con i y m con j en la ecuación (2), obtenemos, para $i \geq 1$ y $j \geq 1$,

$$R(i, j) = R(i, j-1) + \frac{1}{4^j - 1} [R(i, j-1) - R(i-1, j-1)]$$

y

$$R(i, 0) = \frac{1}{2}R(i-1, 0) + h \sum_{k=1}^{2^{i-1}} f[a + (2k-1)h]$$

Los límites de la suma son $1 \leq k \leq 2^i - 1$, así que $1 \leq 2k-1 \leq 2^i - 1$.

Una forma de generar el arreglo de Romberg es calcular un número de términos razonable en la primera columna, $R(0, 0)$ hasta $R(n, 0)$ y después usar la fórmula de extrapolación (2) para construir las columnas $1, 2, \dots, n$ en orden. Otra forma es calcular el arreglo renglón por renglón. Observe, por ejemplo, que $R(1, 1)$ se puede calcular con la fórmula de extrapolación tan pronto como estén disponibles $R(1, 0)$ y $R(0, 0)$. El procedimiento *Romberg* calcula, renglón por renglón, n

renglones y columnas del arreglo de Romberg para una función f y un intervalo específico $[a, b]$:

```

procedure Romberg(f, a, b, n, (rij))
integer i, j, k, n; real a, b, h, suma; real array (rij)0:n × 0:n
external function f
h ← b - a
r00 ← (h/2)[f(a) + f(b)]
for i = 1 to n do
    h ← h/2
    suma ← 0
    for k = 1 to 2i - 1 step 2 do
        suma ← suma + f(a + kh)
    end for
    ri0 ←  $\frac{1}{2}r_{i-1,0} + (suma)h$ 
    for j = 1 to i do
        rij ← ri,j-1 + (ri,j-1 - ri-1,j-1)/(4j - 1)
    end for
end for
end procedure Romberg

```

Este procedimiento se usa con un programa principal y un procedimiento de función (para calcular los valores de la función f). En el programa principal y quizás en el procedimiento *Romberg* se debe incluir alguna interfaz específica del lenguaje para indicar que el primer argumento es una función externa. Recuerde que como se describió en el algoritmo de Romberg, el número de subintervalos es 2^n . Así, se debe elegir un pequeño valor de n , por ejemplo $n = 5$. Un programa más complejo incluiría pruebas automáticas para finalizar el cálculo tan pronto como el error alcance una tolerancia preasignada.

Como un ejemplo, se puede aproximar π usando el procedimiento *Romberg* con $n = 5$ para obtener una aproximación numérica para la integral

$$\int_0^1 \frac{4}{1+x^2} dx$$

Obtenemos los siguientes resultados:

Fórmula de Euler-Maclaurin

Aquí explicamos la fuente de la ecuación (2), que se usa para construir las columnas sucesivas del arreglo de Romberg. Comenzamos con una fórmula que expresa el error en la regla del trapecio sobre 2^{n-1} subintervalos:

$$\int_a^b f(x) dx = R(n-1, 0) + a_2 h^2 + a_4 h^4 + a_6 h^6 + \dots \quad (3)$$

Aquí, $h = (b - a)/2^{n-1}$ y los coeficientes a_i dependen de f pero no de h . Esta ecuación es una forma de la **fórmula de Euler-Maclaurin** y se presenta aquí sin demostración (véase Young y Gregory [1972]). En esta ecuación, $R(n-1, 0)$ denota un típico elemento de la primera columna f en el arreglo de Romberg; por tanto, este es un cálculo trapezoidal de la integral. Observe particularmente que el error se expresa en potencias de h^2 , y la serie del error es $\mathcal{O}(h^2)$. Para nuestro propósito, no es necesario conocer los coeficientes, pero, de hecho, tienen expresiones definidas en términos de f y sus derivadas. Para que la teoría funcione fácilmente, se supone que f tiene derivadas de todos los órdenes en el intervalo $[a, b]$.

Usted ahora debe recordar la teoría de extrapolación de Richardson como se describió en la sección 4.3. La teoría es aplicable debido a la ecuación (3). En la ecuación (8) de la sección 4.3, $L = \phi(h) + \sum_{k=1}^{\infty} a_{2k} h^{2k}$. Aquí, L es el valor de la integral y $\phi(h)$ es $R(n-1, 0)$, el cálculo trapezoidal de L usando subintervalos de tamaño h . La ecuación (10) de la sección 4.3 presenta la fórmula aproximada de extrapolación, que en esta situación es la ecuación (2).

Revisemos brevemente este procedimiento. Al remplazar n con $n + 1$ y h con $h/2$ en la ecuación (3) queda

$$\int_a^b f(x) dx = R(n, 0) + \frac{1}{4}a_2 h^2 + \frac{1}{16}a_4 h^4 + \frac{1}{64}a_6 h^6 + \dots \quad (4)$$

Restando la ecuación (3) de cuatro veces la ecuación (4) obtenemos

$$\int_a^b f(x) dx = R(n, 1) - \frac{1}{4}a_4 h^4 - \frac{5}{16}a_6 h^6 - \dots \quad (5)$$

donde

$$R(n, 1) = R(n, 0) + \frac{1}{3}[R(n, 0) - R(n-1, 0)] \quad (n \geq 1)$$

Observe que este es el primer caso ($m = 1$) de la fórmula de extrapolación (2). Ahora $R(n, 1)$ se debe ser considerablemente más exacta que $R(n, 0)$ o $R(n-1, 0)$, ya que su fórmula de error inicia con un término h^4 . Por tanto, la serie del error es ahora $\mathcal{O}(h^4)$. Este proceso se puede repetir usando la ecuación (5) ligeramente modificada como el punto inicial, es decir, con n remplazada por $n - 1$ y con h sustituida por $2h$. Después combine las dos ecuaciones adecuadamente para eliminar el término h^4 . El resultado es una nueva combinación de los elementos de la columna 2 en el arreglo de Romberg:

$$\int_a^b f(x) dx = R(n, 2) + \frac{1}{4^3}a_6 h^6 + \frac{21}{4^5}a_8 h^8 + \dots \quad (6)$$

donde

$$R(n, 2) = R(n, 1) + \frac{1}{15}[R(n, 1) - R(n-1, 1)] \quad (n \geq 2)$$

que concuerda con la ecuación (2) cuando $m = 2$. Así, $R(n, 2)$ es un aproximación aún más exacta de la integral, ya que su serie del error es $\mathcal{O}(h^6)$.

La suposición básica de la que depende este análisis es que la ecuación (3) es válida para la función f que se está integrando. Por supuesto, en la práctica, usamos un número pequeño de renglones en el algoritmo de Romberg y sólo se necesita este número de términos en la ecuación (3).

Aquí se presenta el teorema que gobierna la situación:

TEOREMA 1

Fórmula de Euler-Maclaurin y término de error

Si $f^{(2m)}$ existe y es continua en el intervalo $[a, b]$, entonces

$$\int_a^b f(x) dx = \frac{h}{2} \sum_{i=0}^{n-1} [f(x_i) + f(x_{i+1})] + E$$

donde $h = (b - a)/n$, $x_i = a + ih$ para $0 \leq i \leq n$ y

$$E = \sum_{k=1}^{m-1} A_{2k} h^{2k} [f^{(2k-1)}(a) - f^{(2k-1)}(b)] - A_{2m} (b - a) h^{2m} f^{(2m)}(\xi)$$

para alguna ξ en el intervalo (a, b) .

En este teorema, las A_k son constantes (relacionadas con los **números de Bernoulli**) y ξ es algún punto en el intervalo (a, b) . Si usted está interesado debe consultar a Young y Gregory [1972, vol. 1, p. 374]. Resulta que las A_k se pueden definir por la ecuación

$$\frac{x}{e^x - 1} = \sum_{k=0}^{\infty} A_k x^k \quad (7)$$

Observe que en la fórmula de Euler-Maclaurin, el miembro derecho contiene la regla del trapecio y un término de error, E . Además, E se puede expresar como una suma finita en potencias ascendentes de h^2 . Este teorema da la justificación formal (y los detalles) de la ecuación (3).

Si el integrando f no tiene un gran número de derivadas pero es al menos una función integrable de Riemann, entonces el algoritmo de Romberg aún converge en el siguiente sentido: el límite de cada *columna* en el arreglo es igual a la integral:

$$\lim_{n \rightarrow \infty} R(n, m) = \int_a^b f(x) dx \quad (m \geq 0)$$

La convergencia de la primera columna se justifica fácilmente refiriéndose a las sumas superior e inferior (véase el problema 5.2.23). Una vez establecida la convergencia de la primera columna, la de las columnas restantes se puede probar usando la ecuación (2) (véan los problemas 5.3.24 y 5.3.25).

En la práctica, no podemos saber si la función f cuya integral buscamos satisface el criterio de suavidad del que depende la teoría. Entonces no se sabe si la ecuación (3) es válida para f . Una forma de probar esto en el curso del algoritmo de Romberg es calcular los cocientes

$$\frac{R(n, m) - R(n - 1, m)}{R(n + 1, m) - R(n, m)}$$

y observar si son cercanos a 4^{m+1} . Permítanos comprobar, al menos para el caso $m = 0$, que este cociente es casi 4 para una función que obedece la ecuación (3).

Si restamos la ecuación (4) de la (3), el resultado es

$$R(n, 0) - R(n - 1, 0) = \frac{3}{4} a_2 h^2 + \frac{15}{16} a_4 h^4 + \frac{63}{64} a_6 h^6 + \dots \quad (8)$$

Si escribimos la misma ecuación para el *siguiente* valor de n , entonces la h de esta ecuación es un medio del valor de h usado en la ecuación (8). Por tanto,

$$R(n+1, 0) - R(n, 0) = \frac{3}{4^2} a_2 h^2 + \frac{15}{16^2} a_4 h^4 + \frac{63}{64^2} a_6 h^6 + \dots \quad (9)$$

Las ecuaciones (8) y (9) ahora se usan para expresar el cociente mencionado antes:

$$\begin{aligned} \frac{R(n, 0) - R(n-1, 0)}{R(n+1, 0) - R(n, 0)} &= 4 \left[\frac{1 + \frac{5}{4} \left(\frac{a_4}{a_2} \right) h^2 + \frac{21}{16} \left(\frac{a_6}{a_2} \right) h^4 + \dots}{1 + \frac{5}{4^2} \left(\frac{a_4}{a_2} \right) h^2 + \frac{21}{16^2} \left(\frac{a_6}{a_2} \right) h^4 + \dots} \right] \\ &= 4 \left[1 + \frac{15}{4^2} \left(\frac{a_4}{a_2} \right) h^2 + \dots \right] \end{aligned}$$

Para valores pequeños de h , esta expresión es casi 4.

Extrapolación general

Para concluir, regresamos al proceso de extrapolación, es decir, al corazón del algoritmo de Romberg. El proceso es la extrapolación de Richardson, que se analizó en la sección 4.3. Este es un ejemplo de un dictamen general en matemática numérica que si se sabe algo acerca de los errores en un proceso, entonces dicho conocimiento se puede aprovechar para mejorar el proceso.

El único tipo de extrapolación ejemplificada hasta ahora (en esta sección y en la sección 4.3) fue la así llamada extrapolación h^2 , que se aplica a un proceso numérico en el que la serie del error es de la forma

$$E = a_2 h^2 + a_4 h^4 + a_6 h^6 + \dots$$

En este caso, los errores se comportan como $\mathcal{O}(h^2)$ cuando $h \rightarrow 0$, pero la idea básica de la extrapolación de Richardson tiene aplicabilidad mucho más amplia. Podríamos aplicar extrapolación si supiéramos, por ejemplo, que

$$E = ah^\alpha + bh^\beta + ch^\gamma + \dots$$

siempre que $0 < \alpha < \beta < \gamma < \dots$. Es suficiente ver cómo eliminar el primer término de la expansión del error porque los pasos sucesivos serían similares.

Suponga por tanto que

$$L = \varphi(h) + ah^\alpha + bh^\beta + ch^\gamma + \dots \quad (10)$$

Aquí, L es una entidad matemática que es aproximada por una fórmula $\varphi(h)$ que depende de h con la serie del error $ah^\alpha + bh^\beta + \dots$. Se deduce que

$$L = \varphi\left(\frac{h}{2}\right) + a\left(\frac{h}{2}\right)^\alpha + b\left(\frac{h}{2}\right)^\beta + c\left(\frac{h}{2}\right)^\gamma + \dots$$

Por tanto, si multiplicamos esta por 2^α , obtenemos

$$2^\alpha L = 2^\alpha \varphi\left(\frac{h}{2}\right) + ah^\alpha + 2^\alpha b\left(\frac{h}{2}\right)^\beta + 2^\alpha c\left(\frac{h}{2}\right)^\gamma + \dots$$

Restando la ecuación (10) de esta ecuación nos libramos del término h^α :

$$(2^\alpha - 1)L = 2^\alpha \varphi\left(\frac{h}{2}\right) - \varphi(h) + (2^{\alpha-\beta} - 1)bh^\beta + (2^{\alpha-\gamma} - 1)ch^\gamma + \dots$$

Reescribimos esto como

$$L = \frac{2^\alpha}{2^\alpha - 1} \varphi\left(\frac{h}{2}\right) - \frac{1}{2^\alpha - 1} \varphi(h) + \tilde{b}h^\beta + \tilde{c}h^\gamma + \dots \quad (11)$$

Así, la combinación lineal especial

$$\frac{2^\alpha}{2^\alpha - 1} \varphi\left(\frac{h}{2}\right) - \frac{1}{2^\alpha - 1} \varphi(h) = \varphi\left(\frac{h}{2}\right) + \frac{1}{2^\alpha - 1} \left[\varphi\left(\frac{h}{2}\right) - \varphi(h) \right] \quad (12)$$

debe ser una aproximación más exacta a L que cualquiera $\varphi(h)$ o $\varphi(h/2)$, ya que su serie del error, en las ecuaciones (10) y (11), mejora de $\mathcal{O}(h^\alpha)$ a $\mathcal{O}(h^\beta)$ cuando $h \rightarrow 0$ y $\beta > \alpha > 0$. Observe que cuando $\alpha = 2$, la combinación en la ecuación (12) es la que ya hemos usado para la segunda columna en el arreglo de Romberg.

La extrapolación del mismo tipo se puede usar en situaciones aún más generales, como se ejemplifica a continuación (y en los problemas).

EJEMPLO 2 Si φ es una función con la propiedad

$$\varphi(x) = L + a_1x^{-1} + a_2x^{-2} + a_3x^{-3} + \dots$$

¿cómo se puede calcular L usando extrapolación de Richardson?

Solución Obviamente, $L = \lim_{x \rightarrow \infty} \varphi(x)$; así, L se puede calcular al evaluar $\varphi(x)$ para una sucesión de valores cada vez más grandes de x . Para usar extrapolación, escribimos

$$\begin{aligned} \varphi(x) &= L + a_1x^{-1} + a_2x^{-2} + a_3x^{-3} + \dots \\ \varphi(2x) &= L + 2^{-1}a_1x^{-1} + 2^{-2}a_2x^{-2} + 2^{-3}a_3x^{-3} + \dots \\ 2\varphi(2x) &= 2L + a_1x^{-1} + 2^{-1}a_2x^{-2} + 2^{-2}a_3x^{-3} + \dots \\ 2\varphi(2x) - \varphi(x) &= L - 2^{-1}a_2x^{-2} - 3 \cdot 2^{-2}a_3x^{-3} - \dots \end{aligned}$$

Así, una vez que hemos calculado $\varphi(x)$ y $\varphi(2x)$, podemos calcular una nueva función $\psi(x) = 2\varphi(2x) - \varphi(x)$. Debe ser una mejor aproximación a L porque su serie del error inicia con x^2 y es $\mathcal{O}(x^2)$ cuando $x \rightarrow \infty$. Este proceso se puede repetir, como en el algoritmo de Romberg. ■

Aquí se presenta un ejemplo concreto del ejemplo anterior. Queremos calcular $\lim_{x \rightarrow \infty} \varphi(x)$ de la tabla siguiente de valores numéricos:

x	1	2	4	8	16	32	64	128
$\varphi(x)$	21.1100	16.4425	14.3394	13.3455	12.8629	12.6253	12.5073	12.4486

Una hipótesis tentativa es que φ tiene la forma del ejemplo anterior. Cuando calculamos los valores de la función $\psi(x) = 2\varphi(2x) - \varphi(x)$, obtenemos una nueva tabla de valores:

x	1	2	4	8	16	32	64
$\psi(x)$	11.7750	12.2363	12.3516	12.3803	12.3877	12.3893	12.3899

Por tanto parece razonable creer que el valor de $\lim_{x \rightarrow \infty} \varphi(x)$ es aproximadamente 12.3899. Si hacemos otra extrapolación, debemos calcular $\theta(x) = [4\psi(2x) - \psi(x)]/3$; los valores para

esta tabla son

x	1	2	4	8	16	32
$\theta(x)$	12.3901	12.3900	12.3899	12.3902	12.3898	12.3901

Para la precisión de los datos dados, concluimos que $\lim_{x \rightarrow \infty} \varphi(x) = 12.3900$ dentro del error de redondeo.

Resumen

(1) Usando la regla recursiva del trapecio, encontramos que la primera columna del **algoritmo de Romberg** es

$$R(n, 0) = \frac{1}{2} R(n - 1, 0) + h \sum_{k=1}^{2^{n-1}} f[a + (2k - 1)h]$$

donde $h = (b - a)/2^n$ y $n \geq 1$. La segunda y las columnas sucesivas en el arreglo de Romberg se generan con la fórmula de extrapolación de Richardson y son

$$R(n, m) = R(n, m - 1) + \frac{1}{4^m - 1} [R(n, m - 1) - R(n - 1, m - 1)]$$

con $n \geq 1$ y $m \geq 1$. El error es $\mathcal{O}(h^2)$ para la primera columna, $\mathcal{O}(h^4)$ para la segunda columna, $\mathcal{O}(h^6)$ para la tercera columna y así sucesivamente. Compruebe los cocientes

$$\frac{R(n, m) - R(n - 1, m)}{R(n + 1, m) - R(n, m)} \approx 4^{m+1}$$

para probar si el algoritmo está funcionando.

(2) Si la expresión L es aproximada por medio de $\varphi(h)$ y si estas entidades están relacionadas con la serie del error

$$L = \varphi(h) + ah^\alpha + bh^\beta + ch^\gamma + \dots$$

entonces una aproximación más exacta es

$$L \approx \varphi\left(\frac{h}{2}\right) + \frac{1}{2^\alpha - 1} \left[\varphi\left(\frac{h}{2}\right) - \varphi(h) \right]$$

con error $\mathcal{O}(h^\beta)$.

Referencias adicionales

Para un estudio adicional, consulte Abramowitz y Stegun [1964], Clenshaw y Curtis [1960], Davis y Rabinowitz [1984], de Boor [1971], Dixon [1974], Fraser y Wilson [1966], Gentleman [1972], Ghizetti y Ossicini [1970], Havie [1969], Kahaner [1971], Krylov [1962], O'Hara y Smith [1968], Stroud [1974] y Stroud y Secrest [1966].

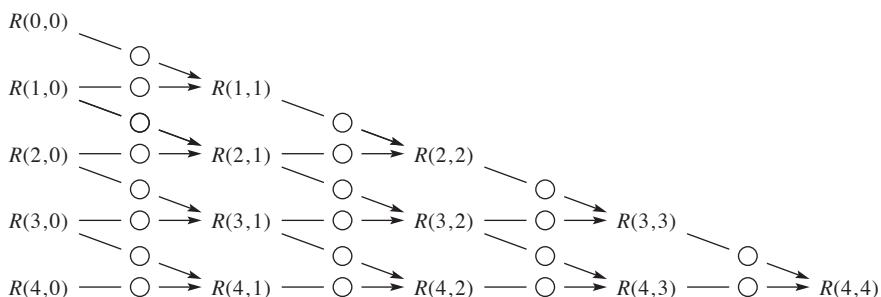
Problemas 5.3

- ^a1. ¿Qué es $R(5, 3)$ si $R(5, 2) = 12$ y $R(4, 2) = -51$, en el algoritmo de Romberg?
2. Si $R(3, 2) = -54$ y $R(4, 2) = 72$, ¿qué es $R(4, 3)$?
3. Calcule $R(5, 2)$ de $R(3, 0) = R(4, 0) = 8$ y $R(5, 0) = -4$.
4. Sea $f(x) = 2^x$. Aproxime $\int_0^4 f(x) dx$ por la regla del trapecio usando los puntos de partición 0, 2 y 4. Repita usando puntos de partición 0, 1, 2, 3 y 4. Ahora aplique extrapolación de Romberg para obtener una mejor aproximación.
- ^a5. Por el algoritmo de Romberg, aproxime $\int_0^2 4 dx / (1 + x^2)$ evaluando $R(1, 1)$.
6. Usando el esquema de Romberg, establezca un valor numérico para la aproximación

$$\int_0^1 e^{-(10x)^2} dx \approx R(1, 1)$$

Calcule la aproximación con sólo tres lugares decimales de exactitud.

- ^a7. Vamos a utilizar el método de Romberg para calcular $\int_0^1 \sqrt{x} \cos x dx$. ¿El método funcionará? ¿Funcionará bien? Explique.
- ^a8. Al combinar $R(0, 0)$ y $R(1, 0)$ para la partición $P = \{-h < 0 < h\}$, determine $R(1, 1)$.
9. En cálculo, se desarrolla una técnica de integración por sustitución. Por ejemplo, si se hace la sustitución $x = z^2$ en la integral $\int_0^1 (e^x / \sqrt{x}) dx$, el resultado es $2 \int_0^1 e^{z^2} dz$. Compruébela y analice los aspectos numéricos de este ejemplo. ¿Qué forma es más probable que produzca una respuesta más exacta mediante el método de Romberg?
- ^a10. ¿Cuántas evaluaciones de la función (integrando) son necesarias si se va a construir el arreglo de Romberg con n renglones y n columnas?
11. Usando la ecuación (2), escriba en los círculos del diagrama siguiente los coeficientes usados en el algoritmo de Romberg:



12. Deduzca la regla de cuadratura para $R(1, 1)$ en términos de la función f evaluada en los puntos de partición a , $a + h$ y $a + 2h$, donde $h = (b - a)/2$. Haga lo mismo para $R(n, 1)$, con $h = (b - a)/2^n$.

- 13.** (Continuación) Deduzca la regla de cuadratura $R(2, 2)$ en términos de la función f evaluada en $a, a + h, a + 2h, a + 3h$ y b , donde $h = (b - a)/4$.
- 14.** Queremos calcular $X = \lim_{n \rightarrow \infty} S_n$ y ya hemos calculado los dos números $u = S_{10}$ y $v = S_{30}$. Se sabe que $X = S_n + Cn^{-3}$. ¿A qué es igual X en términos de u y v ?
- 15.** Suponga que queremos calcular $Z = \lim_{h \rightarrow 0} f(h)$ y que calculamos $f(1), f(2^{-1}), f(2^{-2}), f(2^{-3}), \dots, f(2^{-10})$. Entonces suponemos que también se sabe que $Z = f(h) + ah^2 + bh^4 + ch^6$. Demuestre cómo obtener un cálculo mejorado de Z a partir de los 11 números ya calculados. Demuestre cómo Z se puede determinar exactamente a partir de cualesquiera 4 de los 11 números calculados.
- 16.** Demuestre cómo funciona la extrapolación de Richardson en una sucesión x_1, x_2, x_3, \dots que converge a L cuando $n \rightarrow \infty$ de tal forma que $L - x_n = a_2 n^{-2} + a_3 n^{-3} + a_4 n^{-4} + \dots$
- 17.** Sea x_n una sucesión que converge a L cuando $n \rightarrow \infty$. Si se sabe que $L - x_n$ es de la forma $a_3 n^{-3} + a_4 n^{-4} + \dots$ (los coeficientes son desconocidos), ¿cómo puede converger la sucesión si se acelera tomando combinaciones de x_n y x_{n+1} ?
- 18.** Si el algoritmo de Romberg está operando en una función que tiene derivadas continuas de todos los órdenes en el intervalo de integración, entonces ¿cuál es un límite de la cantidad $|\int_a^b f(x) dx - R(n, m)|$ en términos de h ?
- 19.** Demuestre que la forma exacta de la ecuación (5) es
- $$\int_a^b f(x) dx = R(n, 1) - \sum_{j=1}^{\infty} \left(\frac{4^j - 1}{3 \times 4^j} \right) a_{2j+2} h^{2j+2}$$
- 20.** Deduzca la ecuación (6) y demuestre que su forma exacta es
- $$\int_a^b f(x) dx = R(n, 2) + \sum_{j=2}^{\infty} \left(\frac{4^j - 1}{3 \times 4^j} \right) \left(\frac{4^{j-1} - 1}{15 \times 4^{j-1}} \right) a_{2j+2} h^{2j+2}$$
- 21.** Use el hecho de que los coeficientes en la ecuación (3) tienen la forma
- $$a_k = c_k [f^{(k-1)}(b) - f^{(k-1)}(a)]$$
- para probar que $\int_a^b f(x) dx = R(n, m)$ si f es un polinomio de grado $\leq 2m - 2$.
- 22.** En el algoritmo de Romberg, $R(n, 0)$ denota un cálculo de $\int_a^b f(x) dx$ con subintervalos de tamaño $h = (b - a)/2^n$. Si se supiera que
- $$\int_a^b f(x) dx = R(n, 0) + a_3 h^3 + a_6 h^6 + \dots$$
- ¿cómo tendríamos que modificar el algoritmo de Romberg?
- 23.** Demuestre que si f'' es continua, entonces la primera columna en el arreglo de Romberg converge a la integral de tal forma que el error en el enésimo paso está acotado en magnitud por una constante multiplicada por 4^{-n} .
- 24.** Suponga que la primera columna del arreglo de Romberg converge a $\int_a^b f(x) dx$. Demuestre que la segunda columna también lo hace.

- 25.** (Continuación) En el problema anterior, hemos establecido la propiedad elemental de que si $\lim_{n \rightarrow \infty} R(n, 0) = \int_a^b f(x) dx$, entonces $\lim_{n \rightarrow \infty} R(n, 1) = \int_a^b f(x) dx$. Demuestre que

$$\lim_{n \rightarrow \infty} R(n, 2) = \lim_{n \rightarrow \infty} R(n, 3) = \cdots = \lim_{n \rightarrow \infty} R(n, n) = \int_a^b f(x) dx$$

- 26. a.** Usando la fórmula (7), demuestre que los coeficientes de Euler-Maclaurin no se pueden generar recursivamente.

$$A_0 = 1, \quad A_k = - \sum_{j=1}^k \frac{A_{k-j}}{(j+1)!}$$

- b.** Determine A_k para $1 \leq k \leq 6$.

- *27.** Evalúe E en el teorema de la fórmula de Euler-Maclaurin para este caso especial: $a = 0$, $b = 2\pi$, $f(x) = 1 + \cos 4x$, $n = 4$ y m arbitrario.

Problemas de cómputo 5.3

- *1.** Calcule ocho renglones y columnas en el arreglo de Romberg para $\int_{1.3}^{2.19} x^{-1} \sin x dx$.

- 2.** Diseñe y realice un experimento usando el algoritmo de Romberg. *Sugerencias:* para una función que tiene muchas derivadas continuas en el intervalo, el método debería funcionar bien. Primero, intente con una función de estas. Si elige una cuya integral se puede calcular por otros medios, tendrá un entendimiento mejor de la exactitud del algoritmo de Romberg. Por ejemplo, intente con las integrales definidas

$$\int (1+x)^{-1} dx = \ln(1+x) \quad \int e^x dx = e^x$$

y

$$\int (1+x^2)^{-1} dx = \arctan x$$

- 3.** Pruebe el algoritmo de Romberg en una *mala* función, como \sqrt{x} en $[0, 1]$. ¿Por qué es mala?
- 4.** El número transcendental π es igual al área de un círculo cuyo radio es 1. Demuestre que

$$8 \int_0^{1/\sqrt{2}} (\sqrt{1-x^2} - x) dx = \pi$$

con la ayuda de un diagrama y use esta integral para aproximar π con el método de Romberg.

- *5.** Aplique el método de Romberg para calcular $\int_0^\pi (2 + \sin 2x)^{-1} dx$. Observe la gran precisión que se obtuvo en la primera columna del arreglo, es decir, por los cálculos trapezoidales simples.
- *6.** Calcule $\int_0^\pi x \cos 3x dx$ con el algoritmo de Romberg usando $n = 6$. ¿Cuál es la respuesta correcta?
- *7.** Una integral de la forma $\int_0^\infty f(x) dx$ se puede transformar en una integral en un intervalo finito haciendo un cambio de variable. Compruebe, por ejemplo, que la sustitución $x = -\ln y$ cambia la integral $\int_0^\infty f(x) dx$ en $\int_0^1 y^{-1} f(-\ln y) dy$. Use esta idea para calcular $\int_0^\infty [e^{-x}/(1+x^2)] dx$ usando el algoritmo de Romberg, con 128 evaluaciones de la función transformada.

- 8.** Con el algoritmo de Romberg, calcule

$$\int_0^\infty e^{-x} \sqrt{1 - \sin x} dx$$

- 9.** Calcule

$$\int_0^1 \frac{\sin x}{\sqrt{x}} dx$$

Con el algoritmo de Romberg. *Sugerencia:* considere hacer un cambio de variable.

- 10.** Calcule $\log 2$ usando el algoritmo de Romberg en una integral adecuada.

- 11.** La función Bessel de orden 0 está definida por la ecuación

$$J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin \theta) d\theta$$

Calcule $J_0(1)$ aplicando el algoritmo de Romberg a la integral.

- 12.** Codifique de nuevo el procedimiento de Romberg para que *todos* los resultados de la regla del trapecio se calculen y almacenén en la primera columna. Después en un procedimiento aparte,

procedures *Extrapolar* ($n, (r)$)

realice la extrapolación de Richardson y almacene los resultados en la parte triangular inferior del arreglo (r) . ¿Cuáles son las ventajas y desventajas de este procedimiento sobre la rutina que se presenta en el libro? Pruebe con las dos integrales $\int_0^4 dx/(1+x)$ y $\int_{-1}^1 e^x dx$ usando sólo una corrida de computadora.

- 13. (Proyecto de investigación estudiantil)** Estudie el método de Clenshaw-Curtis para la cuadratura numérica. Si es posible, lea el artículo original de Clenshaw y Curtis [1960] y después programe el método. Si se programa bien, debe ser superior al método de Romberg en muchos casos. Para mayor información de éste, consulte los artículos de Dixon [1974], Fraser y Wilson [1966], Gentleman [1972], Havie [1969], Kahaner [1971] y O'Hara y Smith [1968].

- 14. (Proyecto de investigación estudiantil)** La integración numérica es un problema ideal para usar en una *computadora en paralelo*, puesto que el intervalo de integración se puede subdividir en subintervalos y en cada uno de ellos la integral se puede aproximar simultáneamente independientemente uno de otro. Investigue cómo se puede hacer en paralelo la integración numérica. Si tiene acceso a una computadora en paralelo o puede simular una computadora en paralelo con un conjunto de PC, escriba un programa en paralelo para aproximar p usando el ejemplo estándar

$$\int_0^1 (1 + x^2)^{-1} dx$$

con una regla básica como la del punto medio. Varíe el número de procesadores usado y el número de subintervalos. Puede leer acerca de computación en paralelo en libros como Pacheco [1997], Quinn [1994] y otros o en cualquiera de los numerosos sitios de internet.

- 15.** Use un sistema de software matemático con capacidades simbólicas como Mathematica para comprobar la relación entre A_k y los números de Bernoulli para $k = 6$.

Temas adicionales de integración numérica

Algunas integrales de prueba interesantes (cuyo valor numérico se conoce) son

$$\int_0^1 \frac{dx}{\sqrt{\sin x}} \quad \int_0^\infty e^{-x^3} dx \quad \int_0^1 x|\sin(1/x)| dx$$

Una característica importante que es deseable en un esquema de integración numérica es la capacidad de tratar con funciones que tienen peculiaridades, como ser infinita en algún punto o ser sumamente oscilatoria en algunos subintervalos. Surge otro caso especial cuando el intervalo de integración es infinito. En este capítulo se introducen otros métodos de integración numérica: las fórmulas de cuadratura de Gauss o gaussiana y un esquema adaptado basado en la regla de Simpson. Las fórmulas gaussianas pueden usarse con frecuencia cuando el integrando tiene una singularidad en un extremo del intervalo. El código adaptado de Simpson es *robusto* en el sentido de que puede concentrarse en los cálculos de algunas partes problemáticas del intervalo, donde el integrando puede tener un comportamiento inesperado. Los procedimientos de cuadratura robustos detectan automáticamente singularidades o fluctuaciones rápidas en el integrando y las tratan de forma apropiada.

6.1 Regla de Simpson y adaptable de Simpson

Regla básica de Simpson

La regla básica del trapecio para aproximar $\int_a^b f(x) dx$ se fundamenta en el cálculo del área bajo de la curva en el intervalo $[a, b]$ usando un trapecio. La función de integración de $f(x)$ se considera una línea recta entre $f(a)$ y $f(b)$. La fórmula de integración numérica es la de la forma

$$\int_a^b f(x) dx \approx Af(a) + Bf(b)$$

donde se seleccionan los valores de A y B de tal modo que la fórmula aproximada que resulta integre correctamente cualquier función lineal. Basta integrar exactamente las dos funciones 1 y x , ya que un polinomio a lo más de grado uno es una combinación lineal de estos dos monomios. Para

simplificar los cálculos, sea $a = 0$ y $b = 1$ y encuentre una fórmula del tipo siguiente:

$$\int_0^1 f(x) dx \approx Af(0) + Bf(1)$$

Por tanto, se deben satisfacer estas ecuaciones:

$$\begin{aligned} f(x) = 1 &: \int_0^1 dx = A + B \\ f(x) = x &: \int_0^1 x dx = \frac{1}{2} = B \end{aligned}$$

La solución es $A = B = \frac{1}{2}$ y la fórmula de integración es

$$\int_0^1 f(x) dx \approx \frac{1}{2}[f(0) + f(1)]$$

Se obtiene la **regla básica del trapecio** usando un mapeo lineal $y = (b - a)x + a$ de $[0, 1]$ a $[a, b]$:

$$\int_a^b f(x) dx \approx \frac{1}{2}(b - a)[f(a) + f(b)]$$

Véase la figura 6.1 para una ilustración gráfica.

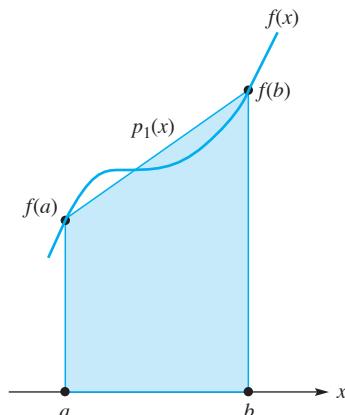


FIGURA 6.1
Regla básica
del trapecio

La siguiente generalización obvia es tomar dos subintervalos $[a, \frac{a+b}{2}]$ y $[\frac{a+b}{2}, b]$ y aproximar $\int_a^b f(x) dx$ al tomar la función de integración $f(x)$ como un polinomio cuadrático que pasa por los tres puntos $f(a)$, $f\left(\frac{a+b}{2}\right)$ y $f(b)$. Permitámos buscar una fórmula de integración numérica del siguiente tipo:

$$\int_a^b f(x) dx \approx Af(a) + Bf\left(\frac{a+b}{2}\right) + Cf(b)$$

Se supone que la función f es continua en el intervalo $[a, b]$. Los coeficientes A , B y C se eligen de tal forma que la fórmula anterior da los valores correctos para la integral siempre que f sea un polinomio cuadrático. Basta integrar correctamente las tres funciones 1 , x y x^2 , ya que un polinomio a lo más de grado 2 es una combinación lineal de estos tres monomios. Para simplificar los cálculos

sea $a = -1$ y $b = 1$ y considere la ecuación

$$\int_{-1}^1 f(x) dx \approx Af(-1) + Bf(0) + Cf(1)$$

Por ende, estas ecuaciones se deben satisfacer:

$$\begin{aligned} f(x) = 1: \quad & \int_{-1}^1 dx = 2 = A + B + C \\ f(x) = x: \quad & \int_{-1}^1 x dx = 0 = -A + C \\ f(x) = x^2: \quad & \int_{-1}^1 x^2 dx = \frac{2}{3} = A + C \end{aligned}$$

La solución es $A = \frac{1}{3}$, $C = \frac{1}{3}$ y $B = \frac{4}{3}$. La fórmula resultante es

$$\int_{-1}^1 f(x) dx \approx \frac{1}{3}[f(-1) + 4f(0) + f(1)]$$

Usando un mapeo lineal $y = \frac{1}{2}(b - a) + \frac{1}{2}(a + b)$ de $[-1, 1]$ a $[a, b]$, obtenemos la **regla básica de Simpson** en el intervalo $[a, b]$:

$$\int_a^b f(x) dx \approx \frac{1}{6}(b - a) \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

Véase la figura 6.2 para una ilustración.

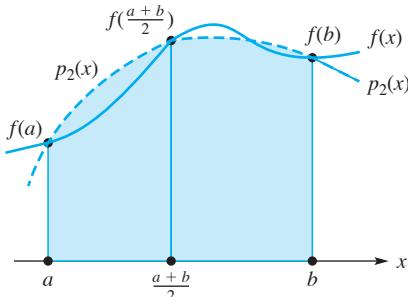


FIGURA 6.2
Regla básica de Simpson

La figura 6.3 muestra en forma gráfica la diferencia entre la regla del trapecio y la regla de Simpson.

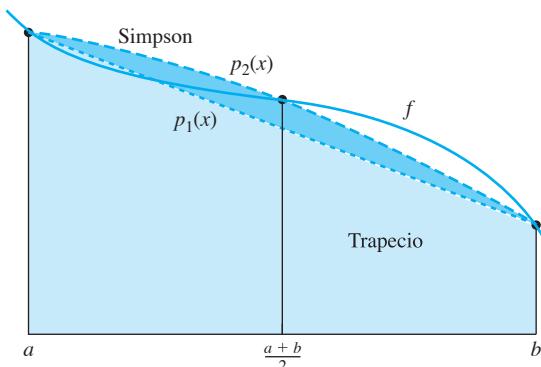


FIGURA 6.3
Ejemplo de la regla del trapecio contra la regla de Simpson

EJEMPLO 1 Encuentre los valores aproximados de la integral

$$\int_{-1}^1 e^{-x^2} ds$$

usando la regla básica del trapecio y la regla básica de Simpson. Tome cinco dígitos significativos

Solución Sea $a = 0$ y $b = 1$. Con la regla básica del trapecio (1) obtenemos

$$\int_0^1 e^{-x^2} ds \approx \frac{1}{2} [e^0 + e^{-1}] \approx 0.5[1 + 0.36788] = 0.68394$$

que es correcta con sólo un lugar decimal significativo (redondeado). Con la regla básica de Simpson (2) encontramos

$$\begin{aligned} \int_0^1 e^{-x^2} ds &\approx \frac{1}{6} [e^0 + 4e^{-0.25} + e^{-1}] \\ &\approx 0.16667[1 + 4(0.77880) + 0.36788] = 0.7472 \end{aligned}$$

que es correcta con tres lugares decimales significativos (redondeado). Recuerde que $\int_0^1 e^{-x^2} dx = \frac{1}{2}\sqrt{\pi}\text{erf}(1) \approx 0.74682$. ■

Regla de Simpson

Una regla de integración numérica en dos subintervalos iguales con puntos de partición a , $a + h$ y $a + 2h = b$ es la ampliamente usada **regla básica de Simpson**:

$$\int_a^{a+2h} f(x) dx \approx \frac{h}{3}[f(a) + 4f(a+h) + f(a+2h)] \quad (1)$$

La regla de Simpson calcula exactamente la integral de un polinomio de interpolación de segundo grado sobre un intervalo de longitud $2h$ con tres puntos; a saber, los dos extremos y el punto medio. Puede deducirse integrando en el intervalo $[0, 2h]$ el polinomio cuadrático de Lagrange que pasa por los puntos $(0, f(0))$, $(h, f(h))$ y $(2h, f(2h))$:

$$\int_0^{2h} f(x) dx \approx \int_0^{2h} p(x) dx = \frac{h}{3}[f(0) + 4f(h) + f(2h)]$$

donde

$$p(x) = \frac{1}{2h^2}(x-h)(x-2h)f(0) - \frac{1}{h^2}x(x-2h)f(h) + \frac{1}{2h^2}x(x-h)f(2h)$$

El término de error en la regla de Simpson se puede establecer usando la serie de Taylor de la sección 1.2:

$$f(a+h) = f + hf' + \frac{1}{2!}h^2f'' + \frac{1}{3!}h^3f''' + \frac{1}{4!}h^4f^{(4)} + \dots$$

donde las funciones f, f', f'', \dots en el lado derecho se evalúan en a . Ahora sustituyendo h por $2h$, tenemos

$$f(a+2h) = f + 2hf' + 2h^2f'' + \frac{4}{3}h^3f''' + \frac{2^4}{4!}h^4f^{(4)} + \dots$$

Con estas dos series se obtiene

$$f(a) + 4f(a+h) + f(a+2h) = 6f + 6hf' + 4h^2f'' + 2h^3f''' + \frac{20}{4!}h^4f^{(4)} + \dots$$

y, así, tenemos

$$\begin{aligned} \frac{h}{3}[f(a) + 4f(a+h) + f(a+2h)] &= 2hf + 2h^2f' + \frac{4}{3}h^3f'' \\ &\quad + \frac{2}{3}h^4f''' + \frac{20}{3 \cdot 4!}h^5f^{(4)} + \dots \end{aligned} \quad (2)$$

Por lo tanto, tenemos una serie para el miembro derecho de la ecuación (1). Ahora vamos a encontrar una para el miembro izquierdo. La serie de Taylor para $F(a+2h)$ es

$$\begin{aligned} F(a+2h) &= F(a) + 2hF'(a) + 2h^2F''(a) + \frac{4}{3}h^3F'''(a) \\ &\quad + \frac{2}{3}h^4F^{(4)}(a) + \frac{2^5}{5!}h^5F^{(5)}(a) + \dots \end{aligned}$$

Sea

$$F(x) = \int_a^x f(t) dt$$

Por el teorema fundamental del cálculo, $F' = f$. Observamos que $F(a) = 0$ y $F(a+2h)$ es la integral del lado izquierdo de la ecuación (1). Como $F'' = f'$, $F''' = f''$ y así sucesivamente, tenemos

$$\int_a^{a+2h} f(x) dx = 2hf + 2h^2f' + \frac{4}{3}h^3f'' + \frac{2}{3}h^4f''' + \frac{2^5}{5 \cdot 4!}h^5f^{(4)} + \dots \quad (3)$$

Restando la ecuación (2) de la ecuación (3) obtenemos

$$\int_a^{a+2h} f(x) dx = \frac{h}{3}[f(a) + 4f(a+h) + f(a+2h)] - \frac{h^5}{90}f^{(4)} - \dots$$

Un análisis más detallado muestra que el término de error para la regla básica de Simpson (1) es $-(h^5/90)f^{(4)}(\xi) = \mathcal{O}(h^5)$ cuando $h \rightarrow 0$, para alguna ξ entre a y $a+2h$. Podemos reescribir la **regla básica de Simpson** en el intervalo $[a, b]$ como

$$\int_a^b f(x) dx \approx \frac{(b-a)}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

con término de error

$$-\frac{1}{90} \left(\frac{b-a}{2} \right)^5 f^{(4)}(\xi)$$

para alguna ξ en (a, b) .

Regla compuesta de Simpson

Suponga que el intervalo $[a, b]$ está subdividido en un número par de subintervalos, digamos n , cada uno de ancho $h = (b-a)/n$. Entonces los puntos de partición son $x_i = a + ih$ para $0 \leq i \leq n$,

donde n es divisible entre 2. Ahora del cálculo básico tenemos

$$\int_a^b f(x) dx = \sum_{i=1}^{n/2} \int_{a+2(i-1)h}^{a+2ih} f(x) dx$$

Usando la regla básica de Simpson tenemos, para el lado derecho,

$$\begin{aligned} & \approx \sum_{i=1}^{n/2} \frac{h}{3} \{f(a + 2(i - 1)h) + 4f(a + (2i - 1)h) + f(a + 2ih)\} \\ & = \frac{h}{3} \left\{ f(a) + \sum_{i=1}^{(n/2)-1} f(a + 2ih) + 4 \sum_{i=1}^{n/2} f(a + (2i - 1)h) \right. \\ & \quad \left. + \sum_{i=1}^{(n/2)-1} f(a + 2ih) + f(b) \right\} \end{aligned}$$

Así, obtenemos

$$\int_a^b f(x) dx \approx \frac{h}{3} \left\{ [f(a) + f(b)] + 4 \sum_{i=1}^{n/2} f[a + (2i - 1)h] + 2 \sum_{i=1}^{(n-2)/2} f(a + 2ih) \right\}$$

donde $h = (b - a)/n$. El término de error es

$$-\frac{1}{180}(b - a)h^4 f^{(4)}(\xi)$$

Muchas fórmulas para integración numérica tienen cálculos de error que implican derivadas de la función que se está integrando. Un punto importante que frecuentemente se pasa por alto es que los cálculos de error dependen de que la función tenga derivadas. Así que si se integra una función definida en partes, la integración numérica debe partirse a lo largo de la región para coincidir con las regiones de suavidad de la función. Otro aspecto importante es que ningún polinomio se hará infinito en el plano finito, por lo que cualquier técnica de integración que utilice polinomios para aproximar el integrando fallará en la obtención de buenos resultados sin que se realice un trabajo extra en las singularidades integrables.

Un esquema adaptable de Simpson

Ahora desarrollamos un esquema adaptado basado en la regla de Simpson para obtener una aproximación numérica a la integral

$$\int_a^b f(x) dx$$

En este algoritmo adaptado, la partición del intervalo $[a, b]$, no se selecciona antes sino que se determina automáticamente. La partición se genera adaptándose así a los más y más pequeños subintervalos que se usan en algunas partes del intervalo y a los pocos y grandes subintervalos que se usan en otras partes.

En el proceso adaptado, dividimos el intervalo $[a, b]$ en dos subintervalos y entonces decidimos si cada uno de ellos se divide en más subintervalos. Este proceso continua hasta que se obtiene alguna exactitud específica en todo el intervalo $[a, b]$. Puesto que el integrando f pueden variar en su comportamiento en el intervalo $[a, b]$, no esperamos que la partición final sea uniforme, sino que varíe en la densidad de los puntos de partición.

Esto es necesario para desarrollar la prueba para decidir si los subintervalos se deben seguir dividiendo. Una aplicación de la regla de Simpson en el intervalo $[a, b]$ se puede escribir como

$$I \equiv \int_a^b f(x) dx = S(a, b) + E(a, b)$$

donde

$$S(a, b) = \frac{(b-a)}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

y

$$E(a, b) = -\frac{1}{90} \left(\frac{b-a}{2}\right)^5 f^{(4)}(a) + \dots$$

Haciendo $h = b - a$ tenemos

$$I = S^{(1)} + E^{(1)} \quad (4)$$

donde

$$S^{(1)} = S(a, b)$$

y

$$\begin{aligned} E^{(1)} &= -\frac{1}{90} \left(\frac{h}{2}\right)^5 f^{(4)}(a) + \dots \\ &= -\frac{1}{90} \left(\frac{h}{2}\right)^5 C \end{aligned}$$

Aquí suponemos que $f^{(4)}$ permanece con un valor constante C en todo el intervalo $[a, b]$. Ahora, dos aplicaciones de la regla de Simpson en el intervalo $[a, b]$ dan

$$I = S^{(2)} + E^{(2)} \quad (5)$$

donde

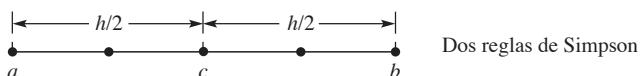
$$S^{(2)} = S(a, c) + S(c, b)$$

donde $c = (a + b)/2$, como en la figura 6.4, y

$$\begin{aligned} E^{(2)} &= -\frac{1}{90} \left(\frac{h/2}{2}\right)^5 f^{(4)}(a) + \dots - \frac{1}{90} \left(\frac{h/2}{2}\right)^5 f^{(4)}(c) + \dots \\ &= -\frac{1}{90} \left(\frac{h/2}{2}\right)^5 [f^{(4)}(a) + f^{(4)}(c)] + \dots \\ &= -\frac{1}{90} \left(\frac{1}{2^5}\right) \left(\frac{h}{2}\right)^5 (2C) = \frac{1}{16} \left[-\frac{1}{90} \left(\frac{h}{2}\right)^5 C\right] \end{aligned}$$



FIGURA 6.4
Regla de Simpson



De nuevo, usamos la suposición de que $f^{(4)}$ permanece con un valor constante C en todo el intervalo $[a, b]$. Encontramos que

$$16E^{(2)} = E^{(1)}$$

Restando la ecuación (5) de la (4) obtenemos

$$S^{(2)} - S^{(1)} = E^{(1)} - E^{(2)} = 15E^{(2)}$$

De esta ecuación y de la ecuación (4) tenemos

$$I = S^{(2)} + E^{(2)} = S^{(2)} + \frac{1}{15}(S^{(2)} - S^{(1)})$$

Este valor de I es el mejor que tenemos en este paso y usamos la desigualdad

$$\frac{1}{15}|S^{(2)} - S^{(1)}| < \varepsilon \quad (6)$$

para guiar el proceso de adaptación.

Si la prueba (6) no se satisface, el intervalo $[a, b]$ se divide en dos subintervalos, $[a, c]$ y $[c, b]$, donde c es el punto medio $c = (a + b)/2$. En cada uno de estos subintervalos, de nuevo se usa la prueba (6) con ε sustituido por $\varepsilon/2$, por lo que la tolerancia resultante será ε en todo el intervalo $[a, b]$. Un procedimiento recursivo maneja esto muy bien.

Para ver por qué tomamos $\varepsilon/2$ en cada subintervalo, recuerde que

$$I = \int_a^b f(x) dx = \int_a^c f(x) dx + \int_c^b f(x) dx = I_{\text{izquierdo}} + I_{\text{derecho}}$$

Si S es la suma de las aproximaciones $S_{\text{izquierdo}}^{(2)}$ sobre $[a, c]$ y $S_{\text{derecho}}^{(2)}$ sobre $[c, b]$, tenemos

$$\begin{aligned} |I - S| &= |I_{\text{izquierdo}} + I_{\text{derecho}} - S_{\text{izquierdo}}^{(2)} - S_{\text{derecho}}^{(2)}| \\ &\leq |I_{\text{izquierdo}} - S_{\text{izquierdo}}^{(2)}| + |I_{\text{derecho}} - S_{\text{derecho}}^{(2)}| \\ &= \frac{1}{15}|S_{\text{izquierdo}}^{(2)} - S_{\text{izquierdo}}^{(1)}| + \frac{1}{15}|S_{\text{derecho}}^{(2)} - S_{\text{derecho}}^{(1)}| \end{aligned}$$

usando la ecuación (6). Por tanto, si requerimos

$$\frac{1}{15}|S_{\text{izquierdo}}^{(2)} - S_{\text{izquierdo}}^{(1)}| \leq \frac{\varepsilon}{2} \quad \text{y} \quad \frac{1}{15}|S_{\text{derecho}}^{(2)} - S_{\text{derecho}}^{(1)}| \leq \frac{\varepsilon}{2}$$

entonces $|I - S| \leq \varepsilon$ en todo el intervalo $[a, b]$.

Ahora describimos un procedimiento recursivo adaptado de Simpson. El intervalo $[a, b]$ está particionado en cuatro subintervalos de ancho $(b - a)/4$. Dos aproximaciones de Simpson se calculan usando subintervalos de doble ancho y cuatro subintervalos de ancho simple; es decir,

$$\begin{aligned} \text{un_simpson} &\leftarrow \frac{h}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \\ \text{dos_simpson} &\leftarrow \frac{h}{12} \left[f(a) + 4f\left(\frac{a+c}{2}\right) + 2f(c) + 4f\left(\frac{c+b}{2}\right) + f(b) \right] \end{aligned}$$

donde $h = b - a$ y $c = (a + b)/2$.

Según la desigualdad (6), si un_simpson y dos_simpson coinciden dentro de 15ε , entonces el intervalo $[a, b]$ no se necesita subdividir aún más para obtener una aproximación exacta de la integral $\int_a^b f(x) dx$. En este caso, el valor de $[16(\text{dos_simpson}) - (\text{un_simpson})]/15$ se utiliza como el valor aproximado de la integral en el intervalo $[a, b]$. Si no se ha obtenido la precisión deseada

para la integral, entonces el intervalo $[a, b]$ se divide por la mitad. Los subintervalos $[a, c]$ y $[c, b]$, donde $c = (a + b)/2$, se utilizan en una llamada recursiva al procedimiento adaptado de Simpson con tolerancia $\epsilon/2$ en cada uno. Este procedimiento termina cuando todos los subintervalos satisfacen la desigualdad (6). Alternativamente, también se utiliza un número máximo de niveles permisibles de la subdivisión de intervalos para terminar el procedimiento antes de tiempo. El procedimiento recursivo proporciona una manera elegante y simple de hacer un seguimiento de los subintervalos que satisfacen la prueba de la tolerancia y los que necesitan dividirse aún más.

Ejemplo de uso del procedimiento adaptable de Simpson

El programa principal para llamar al procedimiento adaptable Simpson se puede presentar mejor en términos de un ejemplo concreto. Se requiere un valor aproximado de la integral

$$\int_0^{\frac{5}{4}\pi} \left[\frac{\cos(2x)}{e^x} \right] dx \quad (7)$$

se desea con precisión $\frac{1}{2} \times 10^{-3}$.

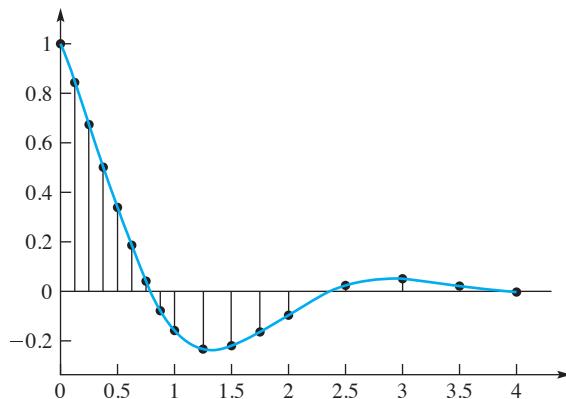


FIGURA 6.5
Integración
adaptada de
 $\int_0^{\frac{5}{4}\pi} \cos(2x)/e^x dx$

En la figura 6.5, se muestra la gráfica de la función integrando. Vemos que esta función tiene muchas vueltas y curvas, de modo que determinar con exactitud el área bajo la curva puede ser difícil. Se escribe para el integrando un procedimiento de función f . Su nombre es el primer argumento en el procedimiento y aquí y en el programa principal se necesitan enunciados de interfaz. Otros argumentos son los valores de los límites superior e inferior de A y B de la integral, la precisión deseada, el nivel de subintervalo actual y el máximo nivel de profundidad. Aquí se presenta el pseudocódigo:

```

recursive real function Simpson( $f, a, b, \epsilon$  nivel, nivel_máx)
    result(resultado_de_simpson)
    integer nivel, nivel_máx;      real a, b, c, d, e, h
    external function f
    nivel  $\leftarrow$  nivel + 1
    h  $\leftarrow$  b - a
    c  $\leftarrow$  (a + b)/2
    un_simpson  $\leftarrow$  h[f(a) + 4f(c) + f(b)]/6
    d  $\leftarrow$  (a + c)/2
    e  $\leftarrow$  (c + b)/2
  
```

```

dos_simpson ← h[f(a) + 4f(d) + 2f(c) + 4f(e) + f(b)]/ 12
if nivel ≥ nivel_máx then
    resultado_de_simpson ← dos_simpson
    output “se ha alcanzado el nivel máximo”
else
    if |dos_simpson - un_simpson| < 15ε then
        resultado_de_simpson ← dos_simpson + (dos_simpson - un_simpson)/15
    else
        simpson_izquierdo ← Simpson(f, a, c, ε/ 2, nivel, nivel_máx)
        simpson_derecho ← Simpson(f, c, b, ε/ 2, nivel, nivel_máx)
        resultado_de_simpson ← simpson_izquierdo + simpson_derecho
    end if
end if
end function Simpson

```

Al escribir un programa controlador de este seudocódigo y ejecutarlo en un computadora se obtiene un valor aproximado de 0.208 para la integral (7). El procedimiento adaptado de Simpson utiliza un número diferente de paneles para distintas partes de la curva como se muestra en la figura 6.5.

Reglas de Newton-Cotes

Las fórmulas de cuadratura de Newton-Cotes para aproximar $\int_a^b f(x) dx$ se obtienen mediante el cálculo de la función de integración $f(x)$ mediante polinomios de interpolación. Las reglas son cerradas cuando implican valores de la función en los extremos del intervalo de integración. De lo contrario, se dice que están abiertas.

Algunas **reglas de Newton-Cotes cerradas** con términos de error son las siguientes. Aquí, $a = x_0$, $b = x_n$, $h = (b - a)/n$, $x_i = x_0 + ih$ para $i = 0, 1, \dots, n$, donde $h = (b - a)/n$, $f_i = f(x_i)$ y $a = x_0 < \xi < x_n = b$ en los términos de error.

Regla del trapecio:

$$\int_{x_0}^{x_1} f(x) dx = \frac{1}{2}h[f_0 + f_1] - \frac{1}{12}h^3 f''(\xi)$$

Regla de Simpson de $\frac{1}{3}$:

$$\int_{x_0}^{x_2} f(x) dx = \frac{1}{3}h[f_0 + 4f_1 + f_2] - \frac{1}{90}h^5 f^{(4)}(\xi)$$

Regla de Simpson de $\frac{3}{8}$:

$$\int_{x_0}^{x_3} f(x) dx = \frac{3}{8}h[f_0 + 3f_1 + 3f_2 + f_3] - \frac{3}{80}h^5 f^{(4)}(\xi)$$

Regla de Boole:

$$\int_{x_0}^{x_4} f(x) dx = \frac{2}{45}h[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4] - \frac{8}{945}h^7 f^{(6)}(\xi)$$

Regla cerrada de Newton-Cotes de seis puntos:

$$\int_{x_0}^{x_5} f(x) dx = \frac{5}{288}h[19f_0 + 75f_1 + 50f_2 + 50f_3 + 75f_4 + 19f_5] - \frac{275}{12096}h^7 f^{(6)}(\xi)$$

Algunas de las **reglas abiertas de Newton-Cotes** son las siguientes:

Regla del punto medio:

$$\int_{x_0}^{x_2} f(x) dx = 2hf_1 + \frac{1}{24}h^3 f''(\xi)$$

Regla abierta de Newton-Cotes de dos puntos:

$$\int_{x_0}^{x_3} f(x) dx = \frac{3}{2}h[f_1 + f_2] + \frac{1}{4}h^3 f''(\xi)$$

Regla abierta de Newton-Cotes de tres puntos:

$$\int_{x_0}^{x_4} f(x) dx = \frac{4}{3}h[2f_1 - f_2 + 2f_3] + \frac{28}{90}h^5 f^{(4)}(\xi)$$

Regla abierta de Newton-Cotes de cuatro puntos:

$$\int_{x_0}^{x_5} f(x) dx = \frac{5}{24}h[11f_1 + f_2 + f_3 + 11f_4] + \frac{95}{144}h^5 f^{(4)}(\xi)$$

Regla abierta de Newton-Cotes de cinco puntos:

$$\int_{x_0}^{x_6} f(x) dx = \frac{6}{20}h[11f_1 - 14f_2 + 26f_3 - 14f_4 + 11f_5] - \frac{41}{140}h^7 f^{(6)}(\xi)$$

Con los años, se han deducido muchas fórmulas de Newton-Cotes y se han compilado en el manual de Abramowitz y Stegun [1964], que está disponible en línea. En lugar de usar reglas de orden alto de Newton-Cotes que se deducen utilizando un solo polinomio en todo el intervalo, es preferible utilizar una regla compuesta basada en una regla básica de orden inferior de Newton-Cotes. Rara vez hay alguna ventaja en usar una abierta en lugar de una regla cerrada que implica el mismo número de nodos. Sin embargo, las reglas abiertas tienen aplicaciones en la integración de una función con singularidades en los extremos y en la solución numérica de ecuaciones diferenciales ordinarias, como se analiza en los capítulos 10 y 11.

Antes del uso generalizado de las computadoras, las de Newton-Cotes fueron las reglas de cuadratura más comúnmente utilizadas, ya que implican fracciones que son fáciles de usar en cálculos a mano. Las reglas de cuadratura gaussianas de la siguiente sección utilizan menos evaluaciones de la función con términos de error de alto orden. El hecho de que suponga nodos que implican números irracionales ya no es un inconveniente en las computadoras modernas.

Resumen

(1) En el intervalo $[a, b]$, la **regla básica de Simpson** es

$$\int_a^b f(x) dx \approx S(a, b) = \frac{(b-a)}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

Con término de error $-\frac{1}{90}[\frac{1}{2}(b-a)]^5 f^{(4)}(\xi)$ para alguna ξ en (a, b) . Haciendo $h = (b-a)/2$, otra forma de la **regla básica de Simpson** es

$$\int_a^{a+2h} f(x) dx \approx \frac{h}{3}[f(a) + 4f(a+h) + f(a+2h)]$$

con término de error $-\frac{1}{90}h^5 f^{(4)}(\xi)$.

(2) La regla compuesta de Simpson de $\frac{1}{3}$ en n (par) subintervalos

$$\begin{aligned} \int_a^b f(x) dx &\approx \frac{h}{3}[f(a) + f(b)] + \frac{4h}{3} \sum_{i=1}^{n/2} f[a + (2i-1)h] \\ &+ \frac{2h}{3} \sum_{i=1}^{(n-2)/2} f(a + 2ih) \end{aligned}$$

donde $h = (b-a)/n$ y el término de error general es $-\frac{1}{180}(b-a)h^4 f^{(4)}(\xi)$.

(3) En el intervalo $[a, b]$ con $c = \frac{1}{2}(a+b)$, la prueba

$$\frac{1}{15}|S(a, c) + S(c, b) - S(a, b)| < \varepsilon$$

se puede usar en un **algoritmo adaptado de Simpson**.

(4) Las reglas de cuadratura de Newton-Cotes abarcan muchas reglas de cuadratura comunes, como la regla del trapecio, la de Simpson y la del punto medio.

Problemas 6.1

a1. Calcule $\int_0^1 (1+x^2)^{-1} dx$ con la regla básica de Simpson, usando los tres puntos de partición $x = 0, 0.5$ y 1 . Compare con la solución verdadera.

2. Considere la integral $\int_0^1 \sin(\pi x^2/2) dx$. Supongamos que queremos integrar numéricamente con un error de magnitud menor que 10^{-3} .

a. ¿Qué ancho h se necesita si queremos utilizar la regla compuesta del trapecio?

b. ¿La regla compuesta de Simpson?

c. ¿La regla compuesta de Simpson de $\frac{3}{8}$?

3. Una función f tiene los valores que se muestran.

x	1	1.25	1.5	1.75	2
$f(x)$	10	8	7	6	5

a. Use la regla de Simpson y los valores de la función en $x = 1, 1.5$ y 2 para aproximar $\int_1^2 f(x) dx$.

b. Repita el inciso anterior, usando $x = 1, 1.25, 1.5, 1.75$ y 2 .

c. Use los resultados de los incisos **a** y **b** junto con los términos de error para establecer una mejor aproximación. *Sugerencia:* suponga un término de error constante Ch^4 .

- d. Repita los incisos anteriores utilizando sumas inferiores, sumas superiores y la regla del trapecio. Compare estos resultados con los de la regla de Simpson.
- 4.** Halle un valor aproximado de $\int_1^2 x^{-1} dx$ usando la regla compuesta de Simpson con $h = 0.25$. Dé un límite en el error.
- 5.** Utilice la regla de Simpson y su fórmula de error para demostrar que si un polinomio cúbico y un polinomio cuadrático se cortan en tres puntos equidistantes, entonces las dos áreas encerradas son iguales.
- 6.** Para la **regla compuesta de Simpson de $\frac{1}{3}$** sobre n (par) subintervalos, deduzca el término de error general

$$-\frac{1}{180}(b-a)h^4 f^{(4)}(\xi)$$

para alguna $\xi \in (a, b)$.

- 7.** (Continuación) La regla compuesta de Simpson para el cálculo de $\int_a^b f(x) dx$ se puede escribir como

$$S_{n-1} = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 4f(x_{n-1}) + f(x_n)]$$

donde $x_i = a + ih$ para $0 \leq i \leq n$ y $h = (b-a)/n$ con n par. Su error es de la forma Ch^4 . Demuestre cómo dos valores de S_k se pueden combinar para obtener un cálculo más exacto de la integral.

- 8.** Un esquema de integración numérica que no es tan conocido es el de la **regla básica de Simpson de $\frac{3}{8}$** sobre tres subintervalos:

$$\int_a^{a+3h} f(x) dx \approx \frac{3h}{8}[f(a) + 3f(a+h) + 3f(a+2h) + f(a+3h)]$$

Establezca el término de error para esta regla y explique por qué ésta se considera menos importante que la regla de Simpson.

- 9.** (Continuación) Utilizando el problema anterior, establezca la regla compuesta de Simpson de $\frac{3}{8}$ sobre n (divisible entre 3) subintervalos. Deduzca el término de error general.

- 10.** Escriba los detalles de la deducción de la regla de Simpson.

- 11.** Encuentre una fórmula del tipo

$$\int_0^1 f(x) dx \approx \alpha f(0) + \beta f(1)$$

que dé los valores correctos para $f(x) = 1$ y $f(x) = x^2$. ¿Su fórmula da el valor correcto cuando $f(x) = x$?

- 12.** Si es posible, encuentre una fórmula

$$\int_{-1}^1 f(x) dx \approx \alpha f(-1) + \beta f(0) + \gamma f(1)$$

que dé el valor correcto para $f(x) = x, x^2$ y x^3 . ¿Integra correctamente las funciones $x \mapsto 1, x^4$ y x^5 ?

- 13.** Use mapeos lineales de $[0, 1]$ a $[a, b]$ y de $[-1, 1]$ a $[a, b]$ para justificar la regla del trapecio y la regla básica de Simpson en términos generales, respectivamente.

Problemas de cómputo 6.1

- 1.** Encuentre valores aproximados para las dos integrales

$$4 \int_0^1 \frac{dx}{1+x^2} \quad 8 \int_0^{1/\sqrt{2}} (\sqrt{1-x^2} - x) dx$$

Use la función recursiva *Simpson* con $\varepsilon = \frac{1}{2} \times 10^{-5}$ y *nivel_máx* = 4. Dibuje las curvas del integrando $f(x)$ en cada caso y muestre cómo *Simpson* partitiona los intervalos. Usted puede querer imprimir los intervalos en los cuales los nuevos valores se agregan a *resultado_de_Simpson*, en la función *Simpson* y también imprimir los valores de $f(x)$ en todo el intervalo $[a, b]$ para dibujar las curvas.

- 2.** Descubra cómo ahorrar evaluaciones de función en la función *Simpson*, de modo que el integrando $f(x)$ se evalúe sólo una vez en cada punto de la partición. Pruebe el código modificado con el ejemplo del libro, es decir,

$$\int_0^{2\pi} \cos(2x)e^{-x} dx$$

con $\varepsilon = 5.0 \times 10^{-5}$ y *nivel_máx* = 4.

- 3.** Modifique y pruebe el seudocódigo de esta sección para que almacene los puntos de la partición y los valores de la función. Usando un trazador automático y el código modificado, repita el problema de cómputo anterior y represente los puntos de partición resultantes y los valores de la función.

- 4.** Escriba y pruebe un código similar al de esta sección, pero basado en una regla de Newton-Cotes diferente.

- 5.** Usando software matemático como Matlab, Maple o Mathematica, escriba y ejecute un programa de computadora para encontrar un valor aproximado de la integral de la ecuación (7). Interprete los mensajes de advertencia. Trate de obtener una aproximación más exacta con más dígitos de precisión usando parámetros adicionales (opcional) del procedimiento.

- 6.** Codifique y ejecute el algoritmo recursivo de Simpson. Use la integral (7) para una prueba.

- 7.** Considere la integral

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx$$

Debido a que tiene singularidades en los extremos del intervalo $[-1, 1]$, las reglas cerradas no se pueden utilizar. Aplique todas las reglas abiertas de Newton-Cotes. Compare y explique estos resultados numéricos con la solución verdadera, que es $\int_{-1}^1 (1-x^2)^{-1/2} dx = \arcsen x|_{-1}^1 = \pi$.

6.2 Fórmulas de cuadratura gaussiana

Descripción

La mayoría de las fórmulas de integración numérica se ajustan al siguiente patrón:

$$\int_a^b f(x) dx \approx A_0 f(x_0) + A_1 f(x_1) + \cdots + A_n f(x_n) \quad (1)$$

En esta sección, cada fórmula de integración numérica es de esta forma. Para utilizar esta fórmula, sólo se necesitan conocer los **nodos** x_0, x_1, \dots, x_n y los **pesos** A_0, A_1, \dots, A_n . Existen tablas que listan los valores numéricos de los nodos y los pesos para importantes casos especiales.

¿De dónde provienen las fórmulas tales como la (1)? Una fuente importante es la teoría de la interpolación polinomial que se presentó en el capítulo 4. Si se han fijado los nodos, entonces existe una correspondiente fórmula de interpolación de Lagrange:

$$p(x) = \sum_{i=0}^n f(x_i) \ell_i(x) \quad \text{donde} \quad \ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \left(\frac{x - x_j}{x_i - x_j} \right)$$

Esta fórmula [ecuaciones (1) y (2) de la sección 4.1] proporciona un polinomio p de grado a lo más n que interpola f en los nodos; es decir, $p(x_i) = f(x_i)$ para $0 \leq i \leq n$. Si las circunstancias son favorables, p será una buena aproximación a f y $\int_a^b p(x) dx$ será una buena aproximación a $\int_a^b f(x) dx$. Por tanto,

$$\int_a^b f(x) dx \approx \int_a^b p(x) dx = \sum_{i=0}^n f(x_i) \int_a^b \ell_i(x) dx = \sum_{i=0}^n A_i f(x_i) \quad (2)$$

donde hemos puesto

$$A_i = \int_a^b \ell_i(x) dx$$

Por la forma en que se ha deducido la fórmula (2), sabemos que dará los valores correctos para la integral de cada polinomio a lo más de grado n .

EJEMPLO 1 Determine la fórmula de cuadratura de la forma (1) cuando el intervalo es $[-2, 2]$ y los nodos son $-1, 0$ y 1 .

Solución Las funciones ℓ_i están dadas antes. Así, tenemos

$$\ell_0(x) = \prod_{j=1}^2 \left(\frac{x - x_j}{x_0 - x_j} \right) = \frac{1}{2}x(x - 1)$$

De manera similar, $\ell_1(x) = -(x + 1)(x - 1)$ y $\ell_2(x) = \frac{1}{2}x(x + 1)$. Los pesos se obtienen al integrar estas funciones. Por ejemplo,

$$A_0 = \int_{-2}^2 \ell_0(x) dx = \frac{1}{2} \int_{-2}^2 (x^2 - x) dx = \frac{8}{3}$$

De manera equivalente, $A_1 = -\frac{4}{3}$ y $A_2 = \frac{8}{3}$. Por tanto, la fórmula de cuadratura es

$$\int_{-2}^2 f(x) dx \approx \frac{8}{3}f(-1) - \frac{4}{3}f(0) + \frac{8}{3}f(1)$$

Como una revisión del trabajo, se puede comprobar que la fórmula da valores exactos para las tres funciones $f(x) = 1, x$ y x^2 . En álgebra lineal, la fórmula proporciona valores correctos para cualquier polinomio de segundo grado. ■

Cambio de intervalos

Las reglas gaussianas para integración numérica se dan generalmente en un intervalo tal como $[0, 1]$ o $[-1, 1]$. Con frecuencia, queremos utilizar estas reglas en un intervalo diferente! Podemos deducir una fórmula en cualquier otro intervalo haciendo un cambio lineal de variables. Si la primera fórmula es exacta para polinomios de cierto grado, lo mismo puede decirse de la segunda. Veamos cómo se logra esto.

Suponga que está dada una fórmula de integración numérica:

$$\int_c^d f(t) dt \approx \sum_{i=0}^n A_i f(t_i)$$

No importa de dónde provenga esta fórmula; sin embargo, supongamos que es exacta para todos los polinomios a lo más de grado m . Si se necesita una fórmula para cualquier otro intervalo, digamos, $[a, b]$, primero definimos una función lineal λ de t , tal que si t recorre $[c, d]$, entonces $\lambda(t)$ recorrerá $[a, b]$. La función λ está dada explícitamente por

$$\lambda(t) = \left(\frac{b-a}{d-c}\right)t + \left(\frac{ad-bc}{d-c}\right)$$

Ahora en la integral

$$\int_a^b f(x) dx$$

cambiamos la variable, $x = \lambda(t)$. Entonces $dx = \lambda'(t)dt = (b-a)(d-c)^{-1} dt$, y así tenemos

$$\begin{aligned} \int_a^b f(x) dx &= \left(\frac{b-a}{d-c}\right) \int_c^d f(\lambda(t)) dt \\ &\approx \left(\frac{b-a}{d-c}\right) \sum_{i=0}^n A_i f(\lambda(t_i)) \end{aligned}$$

Por tanto, tenemos

$$\int_a^b f(x) dx \approx \left(\frac{b-a}{d-c}\right) \sum_{i=0}^n A_i f\left(\left(\frac{b-a}{d-c}\right)t_i + \left(\frac{ad-bc}{d-c}\right)\right)$$

Observe que como λ es lineal, $f(\lambda(t))$ es un polinomio en t si f es un polinomio y los grados son iguales. Por tanto, la nueva fórmula es exacta para polinomios a lo más de grado m .

Nodos gaussianos y pesos

En el análisis anterior los nodos son arbitrarios, aunque por razones prácticas, deben pertenecer al intervalo en el que se realizará la integración. El gran matemático Karl Friedrich Gauss (1777–1855) descubrió que, mediante una colocación especial de los nodos, la exactitud del proceso de integración numérica podría ser mucho mayor. Aquí se presenta el notable resultado de Gauss.

■ TEOREMA 1

Teorema de la cuadratura gaussiana

Sea q un polinomio no trivial de grado $n + 1$ tal que

$$\int_a^b x^k q(x) dx = 0 \quad (0 \leq k \leq n)$$

Sean x_0, x_1, \dots, x_n los ceros de q . Entonces la fórmula

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i f(x_i) \quad \text{donde} \quad A_i = \int_a^b \ell_i(x) dx \quad (3)$$

con estas x_i como nodos serán exacta para todos los polinomios de grado a lo más $2n + 1$. Además, los nodos se encuentran en el intervalo abierto (a, b) .

Demostración (Demostramos sólo el primer argumento). Sea f cualquier polinomio de grado $\leq 2n + 1$. Dividiendo f entre q obtenemos un cociente p y un residuo r , ambos a lo más de grado n . Así

$$f = pq + r$$

Por nuestra hipótesis, $\int_a^b q(x)p(x) dx = 0$. Además, puesto que cada x_i es una raíz de q , tenemos $f(x_i) = p(x_i)q(x_i) + r(x_i) = r(x_i)$. Por último, como r tiene a lo más grado n , la fórmula (3) dará $\int_a^b r(x) dx$ exactamente. Por tanto,

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^b p(x)q(x) dx + \int_a^b r(x) dx = \int_a^b r(x) dx \\ &= \sum_{i=0}^n A_i r(x_i) = \sum_{i=0}^n A_i f(x_i) \end{aligned}$$



Para resumir: con nodos arbitrarios, la fórmula (3) será exacta para todos los polinomios de grado $\leq n$. Con los nodos gaussianos, la fórmula (3) será exacta para todos los polinomios de grado $\leq 2n + 1$.

Las fórmulas de cuadratura que surgen como aplicaciones de este teorema se llaman **fórmulas de cuadratura gaussiana o de Gauss-Legendre**. Hay una fórmula diferente para cada intervalo $[a, b]$ y para cada valor de n . Hay también fórmulas de Gauss más generales para dar valores aproximados a las integrales, tales como

$$\int_0^\infty f(x)e^{-x} dx \quad \int_{-1}^1 f(x)(1-x^2)^{1/2} dx \quad \int_{-\infty}^\infty f(x)e^{-x^2} dx \quad \text{etc.}$$

A continuación deduciremos una fórmula gaussiana que no es muy complicada.

EJEMPLO 2 Determine la fórmula de cuadratura gaussiana con tres nodos gaussianos y tres pesos para la integral $\int_{-1}^1 f(x) dx$.

Solución Debemos encontrar el polinomio q mencionado en el teorema de la cuadratura de gaussiana y después calcular sus raíces. El grado de q es 3, por lo que q tiene la forma de

$$q(x) = c_0 + c_1x + c_2x^2 + c_3x^3$$

Las condiciones que debe satisfacer q son

$$\int_{-1}^1 q(x) dx = \int_{-1}^1 xq(x) dx = \int_{-1}^1 x^2q(x) dx = 0$$

Si hacemos $c_0 = c_2 = 0$, entonces $q(x) = c_1x + c_3x^3$ y así

$$\int_{-1}^1 q(x) dx = \int_{-1}^1 x^2q(x) dx = 0$$

debido a que la integral de una función impar sobre un intervalo simétrico es 0. Para obtener c_1 y c_3 imponemos la condición

$$\int_{-1}^1 x(c_1x + c_3x^3) dx = 0$$

Una solución conveniente de esta es $c_1 = -3$ y $c_3 = 5$. (Ya que es una ecuación homogénea, cualquier múltiplo de una solución es otra solución. Tomamos los enteros más pequeños que funcionan.) Por lo tanto, se obtiene

$$q(x) = 5x^3 - 3x$$

Las raíces de q son $-\sqrt{3/5}, 0$ y $\sqrt{3/5}$. Estas son, pues, los nodos de Gauss para la fórmula de cuadratura deseada.

Para obtener los pesos A_0, A_1 y A_2 se utiliza un procedimiento conocido como el **método de coeficientes indeterminados**. Queremos seleccionar A_0, A_1 y A_2 en la fórmula

$$\int_{-1}^1 f(x) dx \approx A_0f\left(-\sqrt{\frac{3}{5}}\right) + A_1f(0) + A_2f\left(\sqrt{\frac{3}{5}}\right) \quad (4)$$

por lo que la igualdad aproximada (\approx) es una igualdad exacta ($=$) siempre que f sea de la forma $ax^2 + bx + c$. Puesto que la integración es un proceso lineal, la fórmula (4) será exacta para todos los polinomios de grado ≤ 2 si es exacta para estas tres: $1, x$ y x^2 . Arreglamos los cálculos en una forma tabular.

f	Miembro izquierdo	Miembro derecho
1	$\int_{-1}^1 dx = 2$	$A_0 + A_1 + A_2$
x	$\int_{-1}^1 x dx = 0$	$-\sqrt{\frac{3}{5}}A_0 + \sqrt{\frac{3}{5}}A_2$
x^2	$\int_{-1}^1 x^2 dx = \frac{2}{3}$	$\frac{3}{5}A_0 + \frac{3}{5}A_2$

El miembro izquierdo de la ecuación (4) será igual al miembro derecho para todos los polinomios de segundo grado, cuando A_0, A_1 y A_2 satisfacen las ecuaciones

$$\begin{cases} A_0 + A_1 + A_2 = 2 \\ A_0 - A_2 = 0 \\ A_0 + A_2 = \frac{10}{9} \end{cases}$$

Los pesos son $A_0 = A_2 = \frac{5}{9}$ y $A_1 = \frac{8}{9}$. Por tanto, la fórmula final es

$$\int_{-1}^1 f(x) dx \approx \frac{5}{9} f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{\frac{3}{5}}\right) \quad (5)$$

Integrará correctamente todos los polinomios incluidos los de quinto grado. Por ejemplo, $\int_{-1}^1 x^4 dx = \frac{2}{5}$ y la fórmula también produce el valor $\frac{2}{5}$ para esta función. ■

Con la transformación $t = [2x - (b + a)]/(b - a)$, una regla de cuadratura gaussiana de la forma

$$\int_{-1}^1 f(t) dt \approx \sum_{i=0}^n A_i f(t_i)$$

se puede usar en el intervalo $[a, b]$; es decir,

$$\int_a^b f(x) dx = \frac{1}{2}(b - a) \int_{-1}^1 f\left[\frac{1}{2}(b - a)t + \frac{1}{2}(b + a)\right] dt \quad (6)$$

EJEMPLO 3 Use las fórmulas (5) y (6) para aproximar la integral

$$\int_0^1 e^{-x^2} dx$$

Solución Puesto que $a = 0$ y $b = 1$, tenemos

$$\begin{aligned} \int_0^1 f(x) dx &= \frac{1}{2} \int_{-1}^1 f\left(\frac{1}{2}t + \frac{1}{2}\right) dt \\ &= \frac{1}{2} \left[\frac{5}{9} f\left(\frac{1}{2} - \frac{1}{2}\sqrt{\frac{3}{5}}\right) + \frac{8}{9} f\left(\frac{1}{2}\right) + \frac{5}{9} f\left(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{3}{5}}\right) \right] \end{aligned}$$

Haciendo $f(x) = e^{-x^2}$ tenemos

$$\begin{aligned} \int_0^1 e^{-x^2} dx &\approx \frac{5}{18} e^{-0.112701665^2} + \frac{4}{9} e^{-0.5^2} + \frac{5}{18} e^{-0.887298335^2} \\ &\approx 0.746814584 \end{aligned}$$

Comparando contra la solución verdadera $\frac{1}{2}\sqrt{\pi}\text{erf}(1) \approx 0.7468241330$, encontramos que el error en la solución calculada es aproximadamente 10^{-5} , que es excelente, considerando que sólo habían tres evaluaciones de la función. ■

Polinomios de Legendre

Se podría decir mucho más acerca de las fórmulas de cuadratura de Gauss. En particular, hay métodos eficaces para la generación de los polinomios especiales cuyas raíces son usadas como nodos

en la fórmula de cuadratura. Si nos especializamos en la integral $\int_{-1}^1 f(x) dx$ y la normalizamos para que q_n sea tal que $q_n(1) = 1$, entonces estos polinomios se llaman **polinomios de Legendre**. Así, las raíces de los polinomios de Legendre son los nodos de la cuadratura de Gauss en el intervalo $[-1, 1]$. Los primeros polinomios de Legendre son

$$\begin{aligned} q_0(x) &= 1 \\ q_1(x) &= x \\ q_2(x) &= \frac{3}{2}x^2 - \frac{1}{2} \\ q_3(x) &= \frac{5}{2}x^3 - \frac{3}{2}x \end{aligned}$$

Se pueden generar con la relación de recurrencia de tres términos:

$$q_n(x) = \left(\frac{2n-1}{n}\right)xq_{n-1}(x) - \left(\frac{n-1}{n}\right)q_{n-2}(x) \quad (n \geq 2) \quad (7)$$

Sin nuevas ideas, podemos tratar las integrales de la forma $\int_a^b f(x)w(x) dx$. Aquí, $w(x)$ debe ser una función positiva fija en (a, b) para la que todas las integrales $\int_a^b x^n w(x) dx$ existen, para $n = 0, 1, 2, \dots$. Ejemplos importantes para el intervalo $[-1, 1]$ están dados por $w(x) = (1-x^2)^{-1/2}$ y $w(x) = (1-x^2)^{1/2}$. El teorema correspondiente es éste:

■ TEOREMA 2

Teorema de cuadratura gaussiana pesada

Sea q un polinomio diferente de cero de grado $n+1$ tal que

$$\int_a^b x^k q(x) w(x) dx = 0 \quad (0 \leq k \leq n)$$

Sean x_0, x_1, \dots, x_n las raíces de q . Entonces la fórmula

$$\int_a^b f(x) w(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

donde

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad \text{y} \quad A_i = \int_a^b \ell_i(x) w(x) dx$$

será exacta siempre que f sea un polinomio de grado a lo más $2n+1$.

Los nodos y pesos para diversos valores de n en la fórmula de cuadratura de Gauss

$$\int_{-1}^1 f(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

están dados en la tabla 6.1. Los valores numéricos de los nodos y los pesos para diversos valores de n hasta el 95 se puede encontrar en Abramowitz y Stegun [1964]. Véase también Stroud y Secrest [1966]. Como estos nodos y pesos son en su mayoría números irracionales, no se utilizan en cálculos a mano que se refieren a reglas simples que implican enteros y valores racionales. Sin embargo, en programas de cálculo automático, no importa si una fórmula

TABLA 6.1 Nodos y pesos de cuadratura gaussiana

n	Nodos x_i	Pesos A_i
2	$-\sqrt{\frac{1}{3}}$ $+\sqrt{\frac{1}{3}}$	1 1
3	$-\sqrt{\frac{3}{5}}$ 0 $+\sqrt{\frac{3}{5}}$	$\frac{5}{9}$ $\frac{8}{9}$ $\frac{5}{9}$
4	$-\sqrt{\frac{1}{7}(3 - 4\sqrt{0.3})}$ $-\sqrt{\frac{1}{7}(3 + 4\sqrt{0.3})}$ $+\sqrt{\frac{1}{7}(3 - 4\sqrt{0.3})}$ $+\sqrt{\frac{1}{7}(3 + 4\sqrt{0.3})}$	$\frac{1}{2} + \frac{1}{12}\sqrt{\frac{10}{3}}$ $\frac{1}{2} - \frac{1}{12}\sqrt{\frac{10}{3}}$ $\frac{1}{2} + \frac{1}{12}\sqrt{\frac{10}{3}}$ $\frac{1}{2} - \frac{1}{12}\sqrt{\frac{10}{3}}$
5	$-\sqrt{\frac{1}{9}\left(5 - 2\sqrt{\frac{10}{7}}\right)}$ $-\sqrt{\frac{1}{9}\left(5 + 2\sqrt{\frac{10}{7}}\right)}$ 0 $+\sqrt{\frac{1}{9}\left(5 - 2\sqrt{\frac{10}{7}}\right)}$ $+\sqrt{\frac{1}{9}\left(5 + 2\sqrt{\frac{10}{7}}\right)}$	$0.3\left(\frac{-0.7 + 5\sqrt{0.7}}{-2 + 5\sqrt{0.7}}\right)$ $0.3\left(\frac{0.7 + 5\sqrt{0.7}}{2 + 5\sqrt{0.7}}\right)$ $\frac{128}{225}$ $0.3\left(\frac{-0.7 + 5\sqrt{0.7}}{-2 + 5\sqrt{0.7}}\right)$ $0.3\left(\frac{0.7 + 5\sqrt{0.7}}{2 + 5\sqrt{0.7}}\right)$

parece elegante, y las fórmulas de cuadratura de Gauss suelen dar una mayor precisión con menos evaluaciones de la función. La elección de la fórmula de cuadratura depende de la aplicación específica que se considere y usted debe consultar las directrices en las referencias más avanzadas. Véase, por ejemplo, Davis y Rabinowitz [1984], Ghizetti y Ossicini [1970], o Krylov [1962].

Integrales con singularidades

Si el intervalo de integración es ilimitado o la función de integración es ilimitada, entonces se deben utilizar procedimientos especiales para obtener aproximaciones precisas de las integrales.

Un método para el manejo de una singularidad en la función de la integración consiste en cambiar las variables para eliminar la singularidad y luego utilizar una técnica de aproximación estándar. Por ejemplo, obtenemos

$$\int_0^1 \frac{dx}{e^x \sqrt{x}} = 2 \int_0^1 \frac{dt}{e^{t^2}}$$

y

$$\int_0^{\pi/2} \frac{\cos x}{\sqrt{x}} dx = 2 \int_0^{\sqrt{\pi/2}} \cos t^2 dt$$

usando $x = t^2$. Algunas otras transformaciones útiles son $x = -\log t$, $x = t/(1-t)$, $x = \tan t$ y $x = \sqrt{(1+t)/(1-t)}$.

Un caso importante donde las fórmulas gaussianas tienen una ventaja se presenta en la integración de una función que es infinita en un extremo del intervalo. La razón de esta ventaja es que los nodos de la cuadratura de Gauss siempre son los puntos interiores del intervalo. Así, por ejemplo, al calcular

$$\int_0^1 \frac{\sin x}{x} dx$$

podemos utilizar con seguridad el enunciado $y \leftarrow \sin x/x$ con una fórmula gaussiana, ya que el valor en $x=0$ no será necesario. Integrales más difíciles como

$$\int_0^1 \frac{\sqrt[3]{x^2 - 1}}{\sqrt{\sin(e^x - 1)}} dx$$

se pueden calcular directamente con una fórmula gaussiana, a pesar de la singularidad en 0. Por supuesto, nos estamos refiriendo a las integrales que están bien definidas y limitadas, a pesar de una singularidad. Un caso típico es

$$\int_0^1 \frac{dx}{\sqrt{x}}$$

Resumen

(1) Las reglas de cuadratura gaussiana con nodos x_i y pesos A_i son de la forma

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

donde los pesos son

$$A_i = \int_a^b \ell_i(x) dx \quad \ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \left(\frac{x - x_j}{x_i - x_j} \right)$$

Si q es un polinomio no trivial de grado $n + 1$ tal que

$$\int_a^b x^k q(x) dx = 0 \quad (0 \leq k \leq n)$$

entonces los nodos x_0, x_1, \dots, x_n son los ceros de q . Además, los nodos se encuentran en el intervalo abierto (a, b) . La regla es exacta para todos los polinomios de grado a lo más $2n + 1$.

(2) Utilice la fórmula siguiente para cambiar una regla de integración del intervalo $[c, d]$ al $[a, b]$:

$$\int_a^b f(x) dx \approx \left(\frac{b-a}{d-c} \right) \sum_{i=0}^n A_i f \left(\left(\frac{b-a}{d-c} \right) x_i + \left(\frac{ad-bc}{d-c} \right) \right)$$

(3) Algunas reglas de integración gaussianas son

$$\begin{aligned} \int_{-1}^1 f(x) dx &\approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \\ \int_{-1}^1 f(x) dx &\approx \frac{5}{9} f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9} f(0) + \frac{5}{9} f\left(\sqrt{\frac{3}{5}}\right) \end{aligned}$$

(4) Las **reglas de cuadratura gaussianas pesadas** son de la forma

$$\int_a^b f(x) w(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

donde los pesos son

$$A_i = \int_a^b \ell_i(x) w(x) dx$$

Si q es un polinomio distinto de cero de grado $n + 1$ tal que

$$\int_a^b x^k q(x) w(x) dx = 0 \quad (0 \leq k \leq n)$$

entonces los nodos x_0, x_1, \dots, x_n son las raíces de q . La regla es exacta siempre que f sea un polinomio a lo más de grado $2n + 1$.

(5) Si tenemos una fórmula básica de integración numérica para el intervalo $[-1, 1]$ como

$$\int_{-1}^1 f(t) dt \approx \sum_{i=0}^m A_i f(t_i)$$

se puede emplear en un intervalo arbitrario $[c, d]$ usando un cambio de variables. Para convertir al intervalo $[c, d]$, cambie las variables al escribir $x = \beta t + \alpha$, donde $\alpha = \frac{1}{2}(c+d)$ y $\beta = \frac{1}{2}(d-c)$. Observe que cuando $t = -1$, entonces $x = c$ y cuando $t = +1$, entonces $x = d$. También,

debemos utilizar $dx = \beta dt$. Sustituyendo, tenemos las siguientes fórmulas:

$$\int_c^d f(x) dx = \beta \int_{-1}^1 f(\beta t + \alpha) dt \approx \beta \sum_{i=0}^m A_i f(\beta t_i + \alpha)$$

Si queremos encontrar una regla compuesta para el intervalo $[a, b]$ con $m/2$ aplicaciones de la regla básica, usamos

$$\int_a^b f(x) dx = \sum_{j=1}^{n/2} \int_{x_{2(j-1)}}^{x_{2j}} f(x) dx$$

y determinamos

$$\int_a^b f(x) dx \approx h \sum_{j=1}^{n/2} \sum_{i=0}^m A_i f [ht_i + t_{2i-1}]$$

donde $h = t_{2i} - t_{2i-1} = t_{2i-1} - t_{2i-2}$.

Referencias adicionales

Para ahondar en el tema consulte las referencias siguientes: Abell y Braselton [1993], Abramowitz y Stegun [1964], Acton [1990], Atkinson [1993], Clenshaw y Curtis [1960], Davis y Rabinowitz [1984], De Boor [1971], Dixon [1974], Fraser y Wilson [1966], Gander y Gautschi [2000], Gentleman [1972], Ghizetti y Ossicini [1970], Havig [1969], Kahaner [1971], Krylov [1962], O'Hara y Smith [1968], Stroud [1974] y Stroud [1966].

Problemas 6.2

- 1.** Una regla de cuadratura gaussiana para el intervalo $[-1, 1]$ se puede utilizar en el intervalo $[a, b]$ mediante la aplicación de una transformación lineal adecuada. Aproxime

$$\int_0^2 e^{-x^2} dx$$

usando la regla transformada de la tabla 6.1 con $n = 1$.

- 2.** Usando la tabla 6.1, muestre directamente que la regla de cuadratura gaussiana es exacta para los polinomios $1, x, x^2, \dots, x^{2n+1}$ cuando

a. $n = 1$ **b.** $n = 3$ **c.** $n = 4$

- 3.** ¿Para qué alto grado del polinomio es verdadera la fórmula (5)? Compruebe su respuesta al seguir el método de coeficientes indeterminados hasta que una ecuación no se cumpla.

- 4.** Compruebe partes de la tabla 6.1 encontrando las raíces de q_n y usando el método de los coeficientes indeterminados para establecer la fórmula de cuadratura gaussiana en el intervalo $[-1, 1]$ para los siguientes:

a. $n = 1$ **b.** $n = 3$ **c.** $n = 4$

^a5. Construya una regla de la forma:

$$\int_{-1}^1 f(x) dx \approx \alpha f\left(-\frac{1}{2}\right) + \beta f(0) + \gamma f\left(\frac{1}{2}\right)$$

que sea exacta para todos los polinomios de grado ≤ 2 ; es decir, determine valores para α, β y γ . *Sugerencia:* haga la relación exacta para $1, x$ y x^2 y encuentre una solución de las ecuaciones resultantes. Si ésta es exacta para estos polinomios, lo es para todos los polinomios de grado ≤ 2 .

^a6. Establezca una fórmula de integración numérica de la forma

$$\int_a^b f(x) dx \approx Af(a) + Bf'(b)$$

que sea exacta para polinomios del mayor grado posible.

^a7. Deduzca una fórmula para $\int_a^{a+h} f(x) dx$ en términos de evaluaciones de la función $f(a)$, $f(a+h)$ y $f(a+2h)$ que sea correcta para polinomios de tan alto grado como sea posible. *Sugerencia:* use polinomios $1, x-a, (x-a)^2$ y así sucesivamente.

^a8. Deduzca una fórmula de la forma

$$\int_a^b f(x) dx \approx w_0 f(a) + w_1 f(b) + w_2 f'(a) + w_3 f'(b)$$

que sea exacta para polinomios del más alto grado posible.

^a9. Deduzca la regla de la cuadratura gaussiana de la forma

$$\int_{-1}^1 f(x) x^2 dx \approx af(-\alpha) + bf(0) + cf(\alpha)$$

que sea exacta para todos los polinomios de tan alto grado como sea posible; es decir, determine α, a, b y c .

^a10. Determine una fórmula de la forma

$$\int_0^h f(x) dx \approx w_0 f(0) + w_1 f(h) + w_2 f''(0) + w_3 f''(h)$$

que sea exacta para polinomios de tan alto grado como sea posible.

^a11. Deduzca una fórmula de integración numérica de la forma

$$\int_{x_{n-1}}^{x_{n+1}} f(x) dx \approx Af(x_n) + Bf'(x_{n-1}) + Cf''(x_{n+1})$$

para puntos uniformemente espaciados x_{n-1}, x_n y x_{n+1} con espaciamiento h . La fórmula debe ser exacta para polinomios de tan alto grado como sea posible. *Sugerencia:* considere

$$\int_{-h}^h f(x) dx \approx Af(0) + Bf'(-h) + Cf''(h)$$

^a12. Con el método de los coeficientes indeterminados, deduzca una fórmula de integración numérica de la forma

$$\int_{-2}^{+2} |x| f(x) dx \approx Af(-1) + Bf(0) + Cf(+1)$$

que sea exacta para polinomios de grado ≤ 2 . ¿Es exacta para polinomios de grado mayor que 2?

- 13.** Determine A, B, C y D para una fórmula de la forma

$$Af(-h) + Bf(0) + Cf(h) = hDf(h) + \int_{-h}^h f(x) dx$$

que sea exacta para polinomios de tan alto grado como sea posible.

- 14.** La regla de integración numérica

$$\int_0^{3h} f(x) dx \approx \frac{3h}{8} [f(0) + 3f(h) + 3f(2h) + f(3h)]$$

es exacta para polinomios de grado $\leq n$. Determine el valor más grande de n para el cual esta afirmación es cierto.

- 15. (Fórmulas de Adams-Bashforth-Moulton)** Compruebe que las fórmulas de integración numérica

$$\begin{aligned} \text{a. } \int_t^{t+h} g(s) ds &\approx \frac{h}{24} [55g(t) - 59g(t-h) + 37g(t-2h) - 9g(t-3h)] \\ \text{b. } \int_t^{t+h} g(s) ds &\approx \frac{h}{24} [9g(t+h) + 19g(t) - 5g(t-h) + g(t-2h)] \end{aligned}$$

son exactas para polinomios de tercer grado. *Nota:* estas dos fórmulas también se pueden deducir al remplazar los dos integrandos g con dos polinomios de interpolación del capítulo 4 usando los nodos $(t, t-h, t-2h, t-3h)$ o los nodos $(t+h, t, t-h, t-2h)$, respectivamente.

- 16.** Sea una fórmula de cuadratura dada en la forma

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

¿Cuál es la fórmula correspondiente para $\int_0^1 f(x) dx$?

- 17.** Usando las reglas de la tabla 6.1, determine las reglas generales para aproximar integrales de la forma $\int_a^b f(x) dx$.

Problemas de cómputo 6.2

- 1.** Escriba un programa para evaluar una integral $\int_a^b f(x) dx$ usando la fórmula (5).

- 2.** (Continuación) Usando el mismo programa, calcule valores aproximados de las integrales

$$\begin{aligned} \text{a. } \int_0^1 dx / \sqrt{x} & \quad \text{b. } \int_0^2 e^{-\cos^2 x} dx \end{aligned}$$

- 3.** (Continuación) Calcule $\int_0^1 x^{-1} \sin x dx$ por la fórmula gaussiana (5) adecuadamente modificada.

4. Escriba un procedimiento de evaluación de $\int_a^b f(x) dx$ al subdividir primero el intervalo en n subintervalos iguales y, a continuación, usando los tres puntos de la fórmula gaussiana (5) modificada para aplicarla a distintos subintervalos n . La función f y el entero n se proporcionará en el procedimiento.
5. (Continuación) Pruebe el procedimiento escrito en el problema de cómputo anterior en los siguientes ejemplos:

$$\text{a. } \int_0^1 x^5 dx \quad (n = 1, 2, 10) \qquad \text{b. } \int_0^1 x^{-1} \sin x dx \quad (n = 1, 2, 3, 4)$$

6. Aplique y compare las reglas compuestas del trapecio, del punto medio, gaussiana de dos puntos y la regla de Simpson de $\frac{1}{3}$ para la aproximación de la integral

$$\int_0^{2\pi} e^{-x} \cos x dx \approx 0.499066278634$$

usando las 32 aplicaciones de cada regla básica.

7. Codifique y pruebe un procedimiento de **integración gaussiana de dos puntos** que se adapte para aproximar la integral

$$\int_1^3 100x^{-1} \sin(10x^{-1}) dx \approx -18.79829683678703$$

Escriba tres procedimientos usando doble precisión:

- a. **procedimiento Gauss(f, a, b)** de dos puntos de Gauss
- b. **procedimiento adaptado no recursivo Adaptado_Inicial(f, a, b)** que inicializa las variables *suma* y *profundidad* a cero y llama al **procedimiento recursivo Adaptado($f, suma, a, b, profundidad$)**
- c. **procedimiento recursivo Adaptado($f, suma, a, b, profundidad$)** que comprueba si la profundidad máxima se supera; de ser así, se imprime un mensaje de error y se detiene, si no, continúa dividiendo el intervalo $[a, b]$ por la mitad y llama al procedimiento *Gauss* en el subintervalo izquierdo, el subintervalo derecho y todo el intervalo, a continuación, se comprueba si se acepta la prueba de tolerancia, si lo es, agrega el valor aproximado a todo el intervalo de la variable *suma*, de lo contrario, llama al procedimiento recursivo *Adaptado* en los subintervalos izquierdo y derecho, además de aumentar el valor de la variable *profundidad*. La prueba de tolerancia comprueba si la diferencia en valor absoluto entre el valor aproximado en todo el intervalo y la suma de los valores aproximados en el subintervalo izquierdo y en el subintervalo derecho es menor que la variable *tolerance*.

Imprima la contribución de cada subintervalo y la profundidad con la que se acepta el valor aproximado en el subintervalo. Use una profundidad máxima de 100 subintervalos y pare la subdivisión de subintervalos cuando la tolerancia sea menor que 10^{-7} .

8. Calcule las tres integrales que se mencionaron en los casos de prueba de la introducción de este capítulo:

$$\text{a. } \int_0^1 \frac{dx}{\sqrt{\sin x}} \qquad \text{b. } \int_0^\infty e^{-x^3} dx \qquad \text{c. } \int_0^1 |x| \sin(1/x) dx$$

Para determinar si los resultados calculados son exactos, use dos programas diferentes, ya sea Matlab, Maple o Mathematica para hacer estos cálculos.

- 9.** (Continuación) Otro método para el cálculo de la integral $\int_0^1 x|\sin(1/x)| dx$ es con un cambio de variable. Conviértala en la integral $\int_1^\infty |\sin(t)|/t^3$ y después escríbala como la suma de las integrales de 1 a π , π a 2π y $2k\pi$ a $2(k+1)\pi$, para $k = 1, 2, 3, \dots$. Para obtener 12 decimales de precisión, deje que k corra hasta 112536. Sumar las subintegrales en orden de menor a mayor debe dar mejores errores de redondeo. Tomar 10,000 pasos puede requerir alrededor de cinco minutos de tiempo de máquina, pero el error no debe ser de más de dos dígitos en el décimo lugar decimal. Las dos primeras integrales parciales deben calcularse fuera del ciclo y después agregarlas en la suma al final. Usando el programa Matlab `quad`, integre la integral original y, después, programe este método alternativo.

- 10.** Use las fórmulas de cuadratura gaussianas en estos casos de prueba:

$$\begin{array}{ll} \textbf{a. } \int_0^1 \frac{\log(1-x)}{x} dx = -\frac{\pi^2}{6} & \textbf{b. } \int_0^1 \frac{\log(1+x)}{x} dx = \frac{\pi^2}{12} \\ \textbf{c. } \int_0^1 \frac{\log(1+x^2)}{x} dx = \frac{\pi^2}{24} & \end{array}$$

Este problema ilustra integrales con singularidades en los puntos extremos. Las integrales se pueden calcular numéricamente usando cuadratura gaussiana. Los valores conocidos nos permiten probar el proceso. (Véase Haruki y Haruki [1983] y Jeffrey [2000].)

- 11.** Supongamos que queremos calcular $\int_a^b f(x) dx$. Dividimos el intervalo $[a, b]$ en n subintervalos de tamaño uniforme $h = (b-a)/n$, donde n es divisible entre 2. Sean los nodos $x = a_i + ih$ para $0 \leq i \leq n$. Considere las siguientes reglas de integración numérica

Regla compuesta del trapecio (no es necesario que sea n par)

$$\int_a^b f(x) dx \approx \frac{1}{2}h [f(a) + f(b)] + h \sum_{i=1}^{n-1} f(x_i)$$

Regla compuesta de Simpson de $\frac{1}{3}$ (n par)

$$\begin{aligned} \int_a^b f(x) dx &\approx \frac{1}{3}h [f(a) + f(b)] + \frac{4}{3}hf(b-a) \\ &+ \frac{2}{3}h \sum_{i=1}^{\frac{n}{2}-1} [2f(x_{2i-1}) + f(x_{2i})] \end{aligned}$$

Regla compuesta gaussiana de tres puntos (n par)

$$\begin{aligned} \int_a^b f(x) dx &\approx h \sum_{i=1}^{n/2} \left[\frac{5}{9}f\left(x_{2i-1} - h\sqrt{\frac{3}{5}}\right) \right. \\ &\quad \left. + \frac{5}{9}f\left(x_{2i-1} + h\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(x_{2i-1}) \right] \end{aligned}$$

Escriba y ejecute programas de cómputo para obtener la aproximación numérica a la integral $\int_0^{2\pi} [\cos(2x)/e^x] dx$ usando estas reglas con $n = 120$. Use la solución verdadera

$\frac{1}{5}(1 - e^{-2\pi})$ calculada con doble precisión para obtener los errores absolutos en estos resultados.

12. (Continuación) Repita el problema anterior utilizando todas las reglas de la tabla 6.1 y compare los resultados.

13. (Proyecto de investigación estudiantil) Desde un punto de vista práctico, investigue nuevos algoritmos para la integración numérica que están asociados con los nombres de Clenshaw y Curtis [1960], Kronrod [1964] y Patterson [1968]. Los dos últimos son métodos adaptados de cuadratura de Gauss que proporcionan cálculos de error basados en la evaluación y la reutilización de los resultados en los *puntos de Kronrod*. Véase QUADPACK por Pessens, de Doncker, Überhuber y Kahaner [1983] y también Laurie [1997], Ammar, Calvetti, Reichel y [1999] y Calvetti, Golub, Gragg y Reichel [2000] para ejemplos.

14. Considere la integral

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx$$

Puesto que tiene singularidades en los extremos del intervalo $[-1, 1]$, no se pueden usar las reglas cerradas. Aplique todas las reglas gaussianas abiertas de la tabla 6.1. Compare y explique estos resultados numéricos con la solución verdadera, que es $\int_{-1}^1 (1-x^2)^{-1/2} dx = \arcsen x|_{-1}^1 = \pi$.

15. Use integración numérica para comprobar o refutar cada una de las conjeturas siguientes:

- | | | |
|---|--|--|
| a. $\int_0^1 \frac{4}{1+x^2} dx = \pi$ | b. $\int_0^1 \sqrt{x} \log(x) dx = -\frac{4}{9}$ | c. $\int_0^1 \sqrt{x^3} dx = \frac{2}{5}$ |
| d. $\int_0^1 \frac{1}{1+10x^2} dx = \frac{4}{5}$ | e. $\int_{-9}^{100} \frac{1}{\sqrt{ x }} dx = 26$ | f. $\int_0^{10} 25e^{-25x} dx = 1$ |
| g. $\int_0^1 \log(x) dx = -1$ | | |

Sistemas de ecuaciones lineales

Una red eléctrica simple contiene una serie de resistencias y una sola fuente de fuerza electromotriz: batería, como se observa en la figura 7.1. Utilizando las leyes de Kirchhoff y la ley de Ohm, se puede escribir un sistema de ecuaciones lineales que gobiernan este circuito. Si x_1, x_2, x_3 y x_4 son las corrientes de las espiras como se muestra, entonces, las ecuaciones son

$$\begin{cases} 15x_1 - 2x_2 - 6x_3 = 300 \\ -2x_1 + 12x_2 - 4x_3 - x_4 = 0 \\ -6x_1 - 4x_2 + 19x_3 - 9x_4 = 0 \\ -x_2 - 9x_3 + 21x_4 = 0 \end{cases}$$

Sistemas de ecuaciones de este tipo, aun las que tienen cientos de incógnitas, se pueden resolver utilizando los métodos desarrollados en este capítulo. La solución del sistema anterior es

$$x_1 = 26.5 \quad x_2 = 9.35 \quad x_3 = 13.3 \quad x_4 = 6.13$$

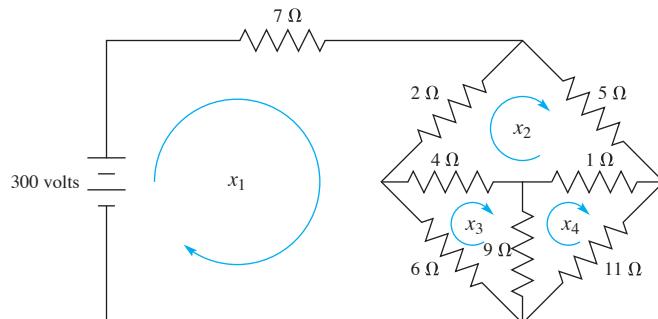


Figura 7.1
Red eléctrica

7.1 Eliminación gaussiana simple

Uno de los problemas fundamentales en muchas aplicaciones científicas y de ingeniería consiste en resolver un sistema lineal algebraico $A\mathbf{x} = \mathbf{b}$ para el vector de incógnitas \mathbf{x} cuando se conoce la matriz de coeficientes A y el vector del miembro derecho \mathbf{b} . Estos sistemas surgen naturalmente en

varias aplicaciones, tales como la aproximación de ecuaciones no lineales con ecuaciones lineales o ecuaciones diferenciales con ecuaciones algebraicas. La piedra angular de muchos de los métodos numéricos empleados para resolver una variedad de problemas prácticos de cálculo es la solución eficiente y precisa de los sistemas lineales. El sistema de ecuaciones lineales algebraicas $\mathbf{Ax} = \mathbf{b}$ puede o no tener una solución y si la tiene, puede o no puede ser única. La eliminación gaussiana es el método estándar para resolver el sistema lineal usando una calculadora o una computadora. Sin duda, este método es familiar para la mayoría de los lectores, ya que es la forma más sencilla de resolver un sistema lineal a mano. Cuando el sistema no tiene solución, se utilizan otros métodos, como la recta de mínimos cuadrados, que se analiza en el capítulo 14. En este capítulo y la mayor parte del siguiente suponemos que la matriz de coeficientes \mathbf{A} es $n \times n$ invertible (no singular).

Con un método matemático puro, la solución al problema $\mathbf{Ax} = \mathbf{b}$ es simplemente $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, donde \mathbf{A}^{-1} es la matriz inversa. Pero en la mayoría de las aplicaciones, es aconsejable resolver el sistema directamente con el vector incógnita \mathbf{x} en lugar de calcular explícitamente la matriz inversa.

En matemáticas aplicadas y en muchas aplicaciones, puede ser una tarea desalentadora, aun para la más grande y rápida de las computadoras, resolver con precisión sistemas muy grandes que implican miles o millones de incógnitas. Algunas de las preguntas son las siguientes: ¿cómo almacenar estos grandes sistemas en la computadora? ¿Cómo sabemos que las respuestas calculadas son correctas? ¿Cuál es la precisión de los resultados calculados? ¿Puede fallar el algoritmo? ¿Cuánto tiempo tomará calcular las respuestas? ¿Cuál es el recuento de la operación asintótica del algoritmo? ¿El algoritmo será inestable en algunos sistemas? ¿La inestabilidad se puede controlar pivoteando? (Permutar el orden de los renglones de la matriz se llama **pivotear**.) ¿Cuál estrategia de pivoteo se debe usar? ¿Cómo sabemos si la matriz está mal condicionada y si las respuestas son exactas?

La eliminación gaussiana transforma un sistema lineal en una forma triangular superior, que es más fácil de resolver. Este proceso, a su vez, equivale a encontrar la factorización $\mathbf{A} = \mathbf{LU}$, donde \mathbf{L} es una matriz triangular inferior y \mathbf{U} es una matriz triangular superior. Esta factorización es especialmente útil cuando la solución de muchos de los sistemas lineales implican la misma matriz de coeficientes pero diferentes lados derechos, lo que ocurre en diversas aplicaciones.

Cuando la matriz de coeficientes \mathbf{A} tiene una estructura especial, como simétrica, definida positiva, de forma triangular, en banda, en bloque o escasa, el enfoque general de eliminación gaussiana con pivoteo parcial se debe modificar o reescribir específicamente para el sistema. Cuando la matriz de coeficientes tiene predominantemente entradas iguales a cero, el sistema es disperso y los métodos iterativos pueden implicar mucho menos memoria de la computadora que la eliminación gaussiana. Vamos a abordar muchas de estas cuestiones en este capítulo y en el siguiente.

Nuestro objetivo en este capítulo es desarrollar un buen programa para resolver un sistema de n ecuaciones lineales con n incógnitas:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \cdots + a_{in}x_n = b_i \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n \end{array} \right. \quad (1)$$

En forma compacta, este sistema se puede escribir simplemente como

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (1 \leq i \leq n)$$

En estas ecuaciones, a_{ij} y b_i son números reales prescritos (datos) y se determinarán las incógnitas x_j . Los subíndices de la letra a están separados por una coma sólo si es necesario para tener mayor claridad, por ejemplo, en $a_{32,75}$ pero no en a_{ij} .

Un gran ejemplo numérico

En esta sección se explica la forma más simple de la eliminación gaussiana. El adjetivo **sencillo** se aplica porque esta forma no suele ser la adecuada para el cálculo automático a menos que se realicen cambios esenciales, como en la sección 7.2. Se ilustra la eliminación gaussiana sencilla con un ejemplo específico que tiene cuatro ecuaciones y cuatro incógnitas:

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 16 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 = 26 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 = -19 \\ -6x_1 + 4x_2 + x_3 - 18x_4 = -34 \end{cases} \quad (2)$$

En el primer paso del procedimiento de eliminación, un múltiplo de la primera ecuación se resta de la segunda, tercera y cuarta ecuaciones para así eliminar x_1 de ellas. Por tanto, queremos crear ceros como coeficientes para cada x_1 debajo de la primera (donde ahora están 12, 3 y -6). Es evidente que hay que restar 2 veces la primera ecuación de la segunda. (Este multiplicador es simplemente el cociente $\frac{12}{6}$.) Asimismo, debemos restar $\frac{1}{2}$ vez la primera ecuación de la tercera. (De nuevo, este multiplicador es exactamente $\frac{3}{6}$.) Por último, se debe restar una vez la primera ecuación de la cuarta. Cuando se ha hecho todo esto, el resultado es

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 16 \\ -4x_2 + 2x_3 + 2x_4 = -6 \\ -12x_2 + 8x_3 + x_4 = -27 \\ 2x_2 + 3x_3 - 14x_4 = -18 \end{cases} \quad (3)$$

Observe que la primera ecuación no se modificó en este proceso, aunque se utiliza para producir los coeficientes cero en las otras ecuaciones. En este contexto, se le llama **ecuación pivote**.

Note también que los sistemas (2) y (3) son *equivalentes* en el sentido técnico siguiente: la solución de (2) también es una solución de (3) y viceversa. Esto es consecuencia a la vez del hecho de que si se agregan cantidades iguales a las cantidades iguales, las cantidades resultantes son iguales. Se puede obtener el sistema (2) del sistema (3) sumando 2 por la primera ecuación a la segunda y así sucesivamente.

En el segundo paso del proceso, mentalmente ignoramos la primera ecuación y la primera columna de los coeficientes. Esto deja un sistema de tres ecuaciones con tres incógnitas. Ahora se repite el mismo proceso con la ecuación de arriba en el sistema más pequeño como la ecuación pivote actual. Así, comenzamos por restar tres veces la segunda ecuación de la tercera. (El multiplicador es exactamente el cociente $\frac{-12}{-4}$.) Después restamos $-\frac{1}{2}$ vez por la segunda ecuación

de la cuarta. Después de hacer la aritmética, se llega a

$$\left\{ \begin{array}{rcl} 6x_1 - 2x_2 + 2x_3 + 4x_4 & = & 16 \\ - 4x_2 + 2x_3 + 2x_4 & = & -6 \\ 2x_3 - 5x_4 & = & -9 \\ 4x_3 - 13x_4 & = & -21 \end{array} \right. \quad (4)$$

El paso final consiste en restar dos veces la tercera ecuación de la cuarta. El resultado es

$$\left\{ \begin{array}{rcl} 6x_1 - 2x_2 + 2x_3 + 4x_4 & = & 16 \\ - 4x_2 + 2x_3 + 2x_4 & = & -6 \\ 2x_3 - 5x_4 & = & -9 \\ - 3x_4 & = & -3 \end{array} \right. \quad (5)$$

Se dice que este sistema está en forma **triangular superior**. Es equivalente al sistema (2).

Esto completa la primera fase (**eliminación hacia adelante**) en el algoritmo gaussiano. La segunda fase (**sustitución hacia atrás**) resolverá el sistema (5) para las incógnitas *inimando en la parte inferior*. Así, de la cuarta ecuación, obtenemos la última incógnita

$$x_4 = \frac{-3}{-3} = 1$$

Sustituyendo $x_4 = 1$ en la tercera ecuación se obtiene

$$2x_3 - 5 = -9$$

y encontramos la siguiente a la última incógnita

$$x_3 = \frac{-4}{2} = -2$$

y así sucesivamente. La solución es

$$x_1 = 3 \quad x_2 = 1 \quad x_3 = -2 \quad x_4 = 1$$

Algoritmo

Para simplificar el análisis, se escribe el sistema (1) en la forma de matriz vectorial. Los elementos coeficientes a_{ij} forman un arreglo cuadrado o matriz de $n \times n$. Las incógnitas x_i y los elementos del lado derecho b_i forman arreglos o vectores $n \times 1$.* (Véase el apéndice D para la notación y conceptos de álgebra lineal.) Por tanto, tenemos que

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix} \quad (6)$$

*Para ahorrar espacio, en ocasiones escribimos un vector como $[x_1, x_2, \dots, x_n]^T$, donde la T es un símbolo para la **transpuesta**, que nos dice que este es un arreglo o vector de $n \times 1$ y no $1 \times n$, como indica sin el símbolo de transpuesta.

o

$$Ax = b$$

Las operaciones entre las *ecuaciones* corresponden a operaciones entre *renglones* en esta notación. Vamos a utilizar estos dos términos indistintamente.

Ahora vamos a organizar el algoritmo de eliminación gaussiana simple para el sistema general, que tiene n ecuaciones y n incógnitas. En este algoritmo, los datos originales se sobreescriben con los nuevos valores calculados. En la fase de eliminación hacia adelante del proceso, hay $n - 1$ pasos principales. El primero de estos pasos utiliza la primera ecuación para producir $n - 1$ ceros como los coeficientes para cada x_1 en todo menos en la primera ecuación. Esto se hace al restar múltiplos adecuados de la primera ecuación de las otras. En este proceso, nos referimos a la primera ecuación como la primera **ecuación pivote** y a a_{11} como el primer **elemento pivote**. Para cada una de las ecuaciones restantes ($2 \leq i \leq n$), se calcula

$$\begin{cases} a_{ij} \leftarrow a_{ij} - \left(\frac{a_{i1}}{a_{11}} \right) a_{1j} & (1 \leq j \leq n) \\ b_i \leftarrow b_i - \left(\frac{a_{i1}}{a_{11}} \right) b_1 \end{cases}$$

El símbolo \leftarrow indica una *sustitución*. Así, el contenido de la ubicación de memoria asignada a a_{ij} se sustituye por $a_{ij} - (a_{i1}/a_{11})a_{1j}$ y así sucesivamente. Esto se logra mediante el siguiente renglón de pseudocódigo:

$$a_{ij} \leftarrow a_{ij} - (a_{i1}/a_{11})a_{1j}$$

Observe que las cantidades (a_{i1}/a_{11}) son los **multiplicadores**. El nuevo coeficiente de x_1 en la i ésta ecuación será cero, ya que $a_{i1} - (a_{i1}/a_{11})a_{1j} = 0$.

Después del primer paso, el sistema será de la forma

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{i2} & a_{i3} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{n2} & a_{n3} & \cdots & a_{nn} \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_i \\ \vdots \\ b_n \end{array} \right]$$

A partir de aquí no vamos a modificar la primera ecuación ni vamos a alterar ninguno de los coeficientes para x_1 (ya que una multiplicación por 0 restada de 0 sigue siendo 0). Por lo tanto, mentalmente se puede ignorar el primer renglón y la primera columna y se repite el proceso en el sistema más pequeño. Con la segunda ecuación como ecuación pivote, calculamos para cada una de las ecuaciones restantes ($3 \leq i \leq n$)

$$\begin{cases} a_{ij} \leftarrow a_{ij} - \left(\frac{a_{i2}}{a_{22}} \right) a_{2j} & (2 \leq j \leq n) \\ b_i \leftarrow b_i - \left(\frac{a_{i2}}{a_{22}} \right) b_2 \end{cases}$$

Justo antes del k ésmo paso en la eliminación hacia adelante, el sistema se presentará de la siguiente manera:

$$\left[\begin{array}{ccccccc|c} a_{11} & a_{12} & a_{13} & \cdots & \cdots & \cdots & a_{1n} & x_1 \\ 0 & a_{22} & a_{23} & \cdots & \cdots & \cdots & a_{2n} & x_2 \\ 0 & 0 & a_{33} & \cdots & \cdots & \cdots & a_{3n} & x_3 \\ \vdots & \vdots & \vdots & \ddots & & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{kk} & \cdots & a_{kj} & \cdots & a_{kn} & x_k \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{ik} & \cdots & a_{ij} & \cdots & a_{in} & x_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nk} & \cdots & a_{nj} & \cdots & a_{nn} & x_n \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_k \\ \vdots \\ b_i \\ \vdots \\ b_n \end{array} \right]$$

Aquí, se ha creado una cuña de coeficientes cero y las primeras k ecuaciones se han procesado y ahora estan corregidas. Utilizando la k -ésima como ecuación pivote, seleccionamos multiplicadores para crear ceros como coeficientes para cada x_j debajo del coeficiente a_{kk} . Por lo tanto, calculamos para cada una de las ecuaciones restantes ($k+1 \leq i \leq n$)

$$\begin{cases} a_{ij} \leftarrow a_{ij} - \left(\frac{a_{ik}}{a_{kk}} \right) a_{kj} & (k \leq j \leq n) \\ b_i \leftarrow b_i - \left(\frac{a_{ik}}{a_{kk}} \right) b_k \end{cases}$$

Obviamente, debemos suponer que todos los divisores en este algoritmo son diferentes de cero.

Seudocódigo

Consideremos ahora el seudocódigo para la eliminación hacia adelante. El arreglo de coeficientes se almacena como una matriz de doble subíndice (a_{ij}); el lado derecho del sistema de ecuaciones se almacena en un solo arreglo subindizado (b_i); se calcula la solución y se almacena en un arreglo de un solo subíndice (x_i). Es fácil ver que los siguientes renglones de seudocódigo realizan la fase de eliminación hacia adelante de la eliminación gaussiana simple:

```
integer i, j, k;  real array (aij)1:n × 1:n, (bi)1:n
for k = 1 to n - 1 do
    for i = k + 1 to n do
        for j = k to n do
            aij ← aij - (aik/akk)akj
        end for
        bi ← bi - (aik/akk)bk
    end for
end for
```

Puesto que el multiplicador a_{ik}/a_{kk} no depende de j , debe moverse fuera del ciclo j . Obsérve también que los nuevos valores en la columna k serán 0, al menos en teoría, porque cuando

$j = k$, tenemos

$$a_{ik} \leftarrow a_{ik} - (a_{ik}/a_{kk})a_{kk}$$

Como esperamos que esto sea 0, carece de interés su cálculo. El lugar donde se está creando el 0 es un buen sitio para almacenar el multiplicador. Si se ponen en práctica estas observaciones, el pseudocódigo será:

```

integer  $i, j, k$ ; real  $xmult$ ; real array  $(a_{ij})_{1:n \times 1:n}, (b_i)_{1:n}$ 
for  $k = 1$  to  $n - 1$  do
    for  $i = k + 1$  to  $n$  do
         $xmult \leftarrow a_{ik}/a_{kk}$ 
         $a_{ik} \leftarrow xmult$ 
        for  $j = k + 1$  to  $n$  do
             $a_{ij} \leftarrow a_{ij} - (xmult)a_{kj}$ 
        end for
         $b_i \leftarrow b_i - (xmult)b_k$ 
    end for
end for

```

Aquí, los multiplicadores se almacenan porque forman parte de la factorización LU que puede ser útil en algunas aplicaciones. Este tema se analiza en la sección 8.1.

Al comienzo de la sustitución hacia atrás, el sistema lineal es de la forma

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \\ a_{33}x_3 + \cdots + a_{3n}x_n = b_3 \\ \vdots \quad \vdots \\ a_{ii}x_i + a_{i,i+1}x_{i+1} + \cdots + a_{in}x_n = b_i \\ \vdots \quad \vdots \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1} \\ a_{nn}x_n = b_n \end{array} \right.$$

donde las a_{ij} y las b_i no son las originales del sistema (6) sino, más bien, son las que han sido alteradas por el proceso de eliminación.

La sustitución hacia atrás empieza con la solución de la ecuación de la enésima ecuación para x_n :

$$x_n = \frac{b_n}{a_{nn}}$$

Entonces, usando la $(n - 1)$ éSIMA ecuación, resolvemos para x_{n-1} :

$$x_{n-1} = \frac{1}{a_{n-1,n-1}} (b_{n-1} - a_{n-1,n}x_n)$$

Seguimos trabajando hacia arriba, recuperando cada x_i mediante la fórmula

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^n a_{ij}x_j \right) \quad (i = n-1, n-2, \dots, 1) \quad (7)$$

Este es el seudocódigo para hacer esto:

```

integer i, j, n; real suma; real array (aij)1:n × 1:n, (xi)1:n
xn ← bn / ann
for i = n - 1 to 1 step -1 do
    suma ← bi
    for j = i + 1 to n do
        suma ← suma - aijxj
    end for
    xi ← suma / aii
end for
```

Ahora juntamos estas partes de seudocódigo para formar un procedimiento, llamado *Gauss_Simple*, cuyo objetivo es resolver un sistema de n ecuaciones lineales con n incógnitas por el método de eliminación gaussiana simple. Este seudocódigo sirve a un solo objetivo didáctico, en la siguiente sección se desarrollará un seudocódigo más robusto.

```

procedure Gauss_Simple(n, (aij), (bi), (xi))
integer i, j, k, n; real suma, xmult
real array (aij)1:n × 1:n, (bi)1:n, (xi)1:n
for k = 1 to n - 1 do
    for i = k + 1 to n do
        xmult ← aik / akk
        aik ← xmult
        for j = k + 1 to n do
            aij ← aij - (xmult)akj
        end for
        bi ← bi - (xmult)bk
    end for
end for
xn ← bn / ann
for i = n - 1 to 1 step -1 do
    suma ← bi
    for j = i + 1 to n do
        suma ← suma - aijxj
    end for
    xi ← suma / aii
end for
end procedure Gauss_Simple
```

Antes de dar un ejemplo de prueba, examinemos el cálculo fundamental de nuestro seudocódigo, a saber, un ciclo for triplemente anidado que contiene una operación de sustitución:

```

for  $k \dots \dots \dots$  do
    for  $i \dots \dots \dots$  do
        for  $j \dots \dots \dots$  do
             $a_{ij} \leftarrow a_{ij} - (a_{ik}/a_{kk})a_{kj}$ 
        end do
    end do
end do

```

En este caso, debemos esperar que todas las cantidades estén infectadas con el error de redondeo. Tal error de redondeo en a_{kj} se multiplica por el factor (a_{ik}/a_{kk}) . Este factor es grande si el elemento pivote $|a_{kk}|$ es pequeño con respecto a $|a_{ik}|$. Por lo tanto, podemos concluir, tentativamente, que los pequeños elementos pivote conducen a multiplicadores grandes y a peores errores de redondeo.

Prueba del seudocódigo

Una buena manera de probar un procedimiento es crear un problema artificial, cuya solución se conoce de antemano. A veces el problema de prueba incluirá un parámetro que se puede cambiar para variar la dificultad. El siguiente ejemplo ilustra esto.

Fijando un valor de n , se define el polinomio

$$p(t) = 1 + t + t^2 + \dots + t^{n-1} = \sum_{j=1}^n t^{j-1}$$

Los coeficientes de este polinomio son todos iguales a 1. Vamos a tratar de recuperar estos coeficientes conocidos de n valores del polinomio. Utilizamos los valores de $p(t)$ en los enteros $t = 1 + i$ para $i = 1, 2, \dots, n$. Si los coeficientes del polinomio se designan por x_1, x_2, \dots, x_n , debemos tener

$$\sum_{j=1}^n (1+i)^{j-1} x_j = \frac{1}{i} [(1+i)^n - 1] \quad (1 \leq i \leq n) \quad (8)$$

En este caso, hemos utilizado la fórmula de la suma de una serie geométrica en el lado derecho, es decir,

$$p(1+i) = \sum_{j=1}^n (1+i)^{j-1} = \frac{(1+i)^n - 1}{(1+i) - 1} = \frac{1}{i} [(1+i)^n - 1] \quad (9)$$

Haciendo $a_{ij} = (1+i)^{j-1}$ y $b_i = [(1+i)^n - 1]/i$ en la ecuación (8), tenemos un sistema lineal.

EJEMPLO 1 Escribimos un seudocódigo para un caso de prueba específico que resuelva el sistema de la ecuación (8) para diferentes valores de n .

Solución Puesto que se puede utilizar el procedimiento de eliminación gaussiana simple, todo lo que se necesita es un programa que llame. Hemos decidido utilizar $n = 4, 5, 6, 7, 8, 9, 10$ para la prueba.

Aquí se presenta un pseudocódigo adecuado:

```

program Prueba_EGS
integer parameter m  $\leftarrow$  10
integer i, j, n; real array,  $(a_{ij})_{1:m \times 1:m}$ ,  $(b_i)_{1:m}$ ,  $(x_i)_{1:m}$ 
for n = 4 to 10 do
    for i = 1 to n do
        for j = 1 to n do
             $a_{ij} \leftarrow (i + 1)^{j-1}$ 
        end for
         $b_i \leftarrow [(i + 1)^n - 1]/i$ 
    end for
    call Gauss_Simple(n,  $(a_{ij})$ ,  $(b_i)$ ,  $(x_i)$ )
    output n,  $(x_i)_{1:n}$ 
end for
end program Prueba_EGS

```

Cuando se ejecutó este pseudocódigo en una máquina que lleva aproximadamente siete dígitos decimales de exactitud, la solución se obtuvo con precisión completa hasta que n alcanzó el 9 y entonces la solución calculada no sirvió ¡ya que uno de los componentes mostró un error relativo del 16120%! (Escriba y ejecute un programa de computadora para verlo por usted mismo.) ■

La matriz de coeficientes de este sistema lineal es un ejemplo de una matriz mal acondicionada llamada **matriz de Vandermonde** y esto explica el hecho de que el sistema se puede resolver con precisión utilizando eliminación gaussiana simple. Lo que es sorprendente es que ¡el problema ocurre de repente! Cuando $n \geq 9$, el error de redondeo que está presente en el cálculo de x_i se propaga y se amplifica a través de la fase de sustitución hacia atrás, de modo que la mayoría de los valores calculados de x_i carecen de valor. Inserte algunos enunciados intermedios de impresión en el código para que vea por usted mismo lo que está pasando aquí. (Véase Gautschi [1990] para obtener más información acerca de la matriz de Vandermonde y su naturaleza mal acondicionada.)

Vectores residual y de error

Para un sistema lineal $Ax = b$ que tiene una solución verdadera x y una solución calculada \tilde{x} , definimos

$$\begin{aligned} e &= \tilde{x} - x && \text{vector de error} \\ r &= A\tilde{x} - b && \text{vector residual} \end{aligned}$$

Una relación importante entre el vector error y el vector residual es

$$Ae = r$$

Suponga que dos estudiantes utilizan diferentes sistemas de cómputo y resuelven el mismo sistema lineal, $Ax = b$. El algoritmo y la precisión que usa cada estudiante no se conocen. Cada uno vehementemente dice tener la respuesta correcta, pero las dos soluciones calculadas \tilde{x} y \hat{x} ¡son totalmente diferentes! ¿Cómo determinamos cuál, si hay alguna, es la solución calculada correcta?

Podemos *comprobar* las soluciones sustituyendo en el sistema original, que es lo mismo que calcular los **vectores residuales** $\tilde{r} = A\tilde{x} - b$ y $\hat{r} = A\hat{x} - b$. Por supuesto, las soluciones calcu-

ladas no son exactas porque cada una debe tener algunos errores de redondeo. Así que nos gustaría aceptar la solución con el vector residual más pequeño. Sin embargo, si supiéramos la solución exacta \mathbf{x} , entonces simplemente compararíamos las soluciones calculadas con la solución exacta, que es lo mismo que calcular los **vectores de error** $\mathbf{e} = \tilde{\mathbf{x}} - \mathbf{x}$ y $\hat{\mathbf{e}} = \hat{\mathbf{x}} - \mathbf{x}$. Ahora, la solución calculada que produce el vector de error más pequeño podría seguramente ser la mejor respuesta.

Puesto que la solución exacta generalmente no se conoce en las aplicaciones, uno tendería a aceptar la solución calculada que tiene el vector residual más pequeño. Pero esto puede no ser la mejor solución calculada si el problema original es sensible a los errores de redondeo, es decir, si está mal acondicionado. De hecho, la pregunta de si una solución calculada para un sistema lineal es una buena solución es muy difícil y está más allá del alcance de este libro. El problema 7.1.5 puede dar una idea de la dificultad de evaluar la exactitud de las soluciones calculadas de los sistemas lineales.

Resumen

(1) El procedimiento básico de **eliminación hacia adelante**, utilizando la ecuación k para operar en las ecuaciones $k + 1, k + 2, \dots, n$ es

$$\begin{cases} a_{ij} \leftarrow a_{ij} - (a_{ik}/a_{kk})a_{kj} & (k \leq j \leq n, k < i \leq n) \\ b_i \leftarrow b_i - (a_{ik}/a_{kk})b_k \end{cases}$$

Aquí suponemos $a_{kk} \neq 0$. El procedimiento básico de **sustitución hacia atrás** es

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^n a_{ij}x_j \right) \quad (i = n-1, n-2, \dots, 1)$$

(2) Cuando se resuelve el sistema lineal $\mathbf{Ax} = \mathbf{b}$, si la solución verdadera o exacta es \mathbf{x} y la solución aproximada o calculada es $\tilde{\mathbf{x}}$, entonces las cantidades importantes son

$$\begin{array}{ll} \text{vectores de error} & \mathbf{e} = \tilde{\mathbf{x}} - \mathbf{x} \\ \text{vectores residuales} & \mathbf{r} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b} \end{array}$$

Problemas 7.1

1. Muestre que el sistema de ecuaciones

$$\begin{cases} x_1 + 4x_2 + \alpha x_3 = 6 \\ 2x_1 - x_2 + 2\alpha x_3 = 3 \\ \alpha x_1 + 3x_2 + x_3 = 5 \end{cases}$$

tiene una solución única cuando $\alpha = 0$, no tiene solución cuando $\alpha = -1$ y un número infinito de soluciones cuando $\alpha = 1$. También, investigue la situación correspondiente cuando el lado derecho se remplaza por ceros.

- ^a2. ¿Para qué valores de α la eliminación gaussiana simple produce respuestas erróneas para este sistema?

$$\begin{cases} x_1 + x_2 = 2 \\ \alpha x_1 + x_2 = 2 + \alpha \end{cases}$$

Explique qué pasa en la computadora.

3. Aplique eliminación gaussiana simple para estos ejemplos y explique las fallas. Resuelva si es posible los sistemas por otros medios.

^a**a.** $\begin{cases} 3x_1 + 2x_2 = 4 \\ -x_1 - \frac{2}{3}x_2 = 1 \end{cases}$

^a**b.** $\begin{cases} 6x_1 - 3x_2 = 6 \\ -2x_1 + x_2 = -2 \end{cases}$

c. $\begin{cases} 0x_1 + 2x_2 = 4 \\ x_1 - x_2 = 5 \end{cases}$

d. $\begin{cases} x_1 + x_2 + 2x_3 = 4 \\ x_1 + x_2 + 0x_3 = 2 \\ 0x_1 + x_2 + x_3 = 0 \end{cases}$

- ^a4. Resuelva el siguiente sistema de ecuaciones, con sólo cuatro cifras significativas en cada paso del cálculo, y compare su respuesta con la solución obtenida cuando se conservan ocho cifras significativas. Sea consistente siempre al redondear el número de cifras significativas que lleva o corta.

$$\begin{cases} 0.1036x_1 + 0.2122x_2 = 0.7381 \\ 0.2081x_1 + 0.4247x_2 = 0.9327 \end{cases}$$

- ^a5. Considere

$$A = \begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{bmatrix}, \quad b = \begin{bmatrix} 0.217 \\ 0.254 \end{bmatrix}$$

$$\tilde{x} = \begin{bmatrix} 0.999 \\ -1.001 \end{bmatrix}, \quad \hat{x} = \begin{bmatrix} 0.341 \\ -0.087 \end{bmatrix}$$

Calcule vectores residuales $\tilde{r} = A\tilde{x} - b$ y $\hat{r} = A\hat{x} - b$ y decida cuál de \tilde{x} y \hat{x} es el mejor vector solución. Ahora calcule los vectores de error $e = \tilde{x} - x$ y $\hat{e} = \hat{x} - x$, donde $x = [1, -1]^T$ es la solución exacta. Analice las implicaciones de este ejemplo.

6. Considere el sistema

$$\begin{cases} 10^{-4}x_1 + x_2 = b_1 \\ x_1 + x_2 = b_2 \end{cases}$$

donde $b_1 \neq 0$ y $b_2 \neq 0$. Su solución exacta es

$$x_1 = \frac{-b_1 + b_2}{1 - 10^{-4}}, \quad x_2 = \frac{b_1 - 10^{-4}b_2}{1 - 10^{-4}}$$

- ^aa. Sea $b_1 = 1$ y $b_2 = 2$. Resuelva este sistema usando eliminación gaussiana simple con tres dígitos (redondeado) aritméticos y compare con la solución exacta $x_1 = 1.00010\ldots$ y $x_2 = 0.999899\ldots$

- ^ab. Repita el inciso anterior después intercambiando el orden de las dos ecuaciones.

- ^ac. Encuentre valores de b_1 y b_2 en el sistema original para que la eliminación gaussiana simple no dé respuestas pobres.

7. Resuelva cada uno de los siguientes sistemas mediante la eliminación gaussiana simple, es decir, eliminación y sustitución hacia adelante y hacia atrás. Lleve cuatro cifras significativas.

$$\text{a. } \begin{cases} 3x_1 + 4x_2 + 3x_3 = 10 \\ x_1 + 5x_2 - x_3 = 7 \\ 6x_1 + 3x_2 + 7x_3 = 15 \end{cases}$$

$$\text{b. } \begin{cases} 3x_1 + 2x_2 - 5x_3 = 0 \\ 2x_1 - 3x_2 + x_3 = 0 \\ x_1 + 4x_2 - x_3 = 4 \end{cases}$$

$$\text{c. } \begin{bmatrix} 1 & -1 & 2 & 1 \\ 3 & 2 & 1 & 4 \\ 5 & 8 & 6 & 3 \\ 4 & 2 & 5 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}$$

$$\text{d. } \begin{cases} 3x_1 + 2x_2 - x_3 = 7 \\ 5x_1 + 3x_2 + 2x_3 = 4 \\ -x_1 + x_2 - 3x_3 = -1 \end{cases}$$

$$\text{e. } \begin{cases} x_1 + 3x_2 + 2x_3 + x_4 = -2 \\ 4x_1 + 2x_2 + x_3 + 2x_4 = 2 \\ 2x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ x_1 + 2x_2 + 4x_3 + x_4 = -1 \end{cases}$$

Problemas de cómputo 7.1

- Programe y corra el ejemplo en el libro e inserte algunos enunciados de impresión (print) para ver qué está pasando.
- Reescriba y pruebe el procedimiento *Gauss_Simple* para que sea orientado por columna; es decir, para que el primer índice de a_{ij} varíe en el ciclo más interno.
- Defina una matriz A de $n \times n$ con la ecuación $a_{ij} = i + j$. Defina \mathbf{b} con la ecuación $b_i = i + 1$. Resuelva $A\mathbf{x} = \mathbf{b}$ usando el procedimiento *Gauss_Simple*. ¿Qué debería ser \mathbf{x} ?
- Defina un arreglo de $n \times n$ mediante $a_{ij} = -1 + 2 \min\{i, j\}$. Despues establezca el arreglo (b_i) de tal forma que la solución del sistema $\sum_{j=1}^n a_{ij}x_j = b_i$ ($1 \leq i \leq n$) sea $x_j = 1$ ($1 \leq j \leq n$). Pruebe el procedimiento *Gauss_Simple* de este sistema para un valor moderado de n , digamos $n = 15$.
- Escriba y pruebe una versión del procedimiento *Gauss_Simple* en el cual
 - Un intento de división entre 0 se indica con un error de regreso.
 - La solución \mathbf{x} se coloca en el arreglo (b_i).
- Escriba una versión de aritmética compleja de *Gauss_Simple* declarando ciertas variables complejas y haciendo otros cambios necesarios en el código. Considere el sistema lineal complejo

$$A\mathbf{x} = \mathbf{b}$$

Donde

$$A = \begin{bmatrix} 5 + 9i & 5 + 5i & -6 - 6i & -7 - 7i \\ 3 + 3i & 6 + 10i & -5 - 5i & -6 - 6i \\ 2 + 2i & 3 + 3i & -1 + 3i & -5 - 5i \\ 1 + i & 2 + 2i & -3 - 3i & 4i \end{bmatrix}$$

Resuelva este sistema cuatro veces con los siguientes vectores \mathbf{b} :

$$\begin{bmatrix} -10 + 2i \\ -5 + i \\ -5 + i \\ -5 + i \end{bmatrix}, \quad \begin{bmatrix} 2 + 6i \\ 4 + 12i \\ 2 + 6i \\ 2 + 6i \end{bmatrix}, \quad \begin{bmatrix} 7 - 3i \\ 7 - 3i \\ 0 \\ 7 - 3i \end{bmatrix}, \quad \begin{bmatrix} -4 - 8i \\ -4 - 8i \\ -4 - 8i \\ 0 \end{bmatrix}$$

Compruebe que las soluciones son $z = \lambda^{-1}b$ para escalares λ . Los números λ se llaman **valores propios** y las soluciones z son **vectores propios** de A . En general, el vector b no se conoce y la solución del problema de $Az = \lambda z$ no se puede obtener con un programa de solución de ecuaciones lineales.

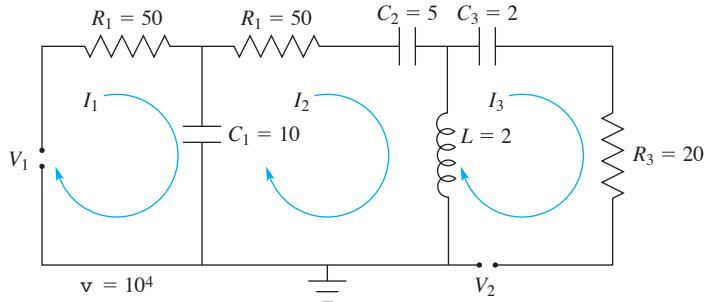
7. (Continuación) Un problema común de ingeniería eléctrica es el cálculo de las corrientes en un circuito eléctrico. Por ejemplo, el circuito que se muestra en la figura con R_i (ohms), C_i (microfarads), L (milihenries) y w (hertz) conduce al sistema

$$\begin{cases} (50 - 10i)I_1 + (50)I_2 + (50)I_3 = V_1 \\ (10i)I_1 + (10 - 10i)I_2 + (10 - 20i)I_3 = 0 \\ - (30i)I_2 + (20 - 50i)I_3 = -V_2 \end{cases}$$

Seleccione V_1 igual a 100 milivoltos y resuelva los dos casos:

- a.** Los dos voltajes están en fase; es decir, $V_2 = V_1$.
b. El segundo voltaje está un cuarto de un ciclo adelante del primero; es decir, $V_2 = iV_1$.

Use la versión de aritmética compleja de *Gauss_Simple* y en cada caso, resuelva el sistema para la amplitud (en miliampères) y la fase (en grados) para cada corriente I_k . *Sugerencia:* cuando $I_k = \text{Re}(I_k) + i\text{Im}(I_k)$, la amplitud es $|I_k|$, y la fase es $(180^\circ/\pi) \arctan[\text{Im}(I_k)/\text{Re}(I_k)]$. Dibuje un diagrama que muestre por qué esto es así.



8. Seleccione un valor razonable de n y genere un arreglo aleatorio a de $n \times n$ usando un generador de números aleatorios. Defina el arreglo b tal que la solución del sistema

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (1 \leq i \leq n)$$

sea $x_j = j$, donde $1 \leq j \leq n$. Pruebe el algoritmo gaussiano simple en este sistema. *Sugerencia:* puede usar la función *Random*, que se analiza en el capítulo 13, para generar los elementos aleatorios del arreglo (a_{ij}) .

9. Realice la prueba descrita en el libro para el procedimiento *Gauss_Simple* pero *invierta* el orden de las ecuaciones. *Sugerencia:* es suficiente remplazar en el código i con $n - i + 1$ en los lugares adecuados.
10. Resuelva el sistema lineal dado en el ejemplo de inicio de este capítulo usando *Gauss_Simple*.
11. Use software matemático como el de las rutinas incorporadas en Matlab, Maple o Mathematica para resolver directamente al sistema lineal (2).

7.2 Eliminación gaussiana con pivoteo escalado parcial

La eliminación gaussiana simple puede fallar

Para ver por qué el algoritmo de eliminación gaussiana simple no es satisfactorio considere el siguiente sistema:

$$\begin{cases} 0x_1 + x_2 = 1 \\ x_1 + x_2 = 2 \end{cases} \quad (1)$$

El seudocódigo que construimos en la sección 7.1 intentaría restar un múltiplo de la primera ecuación a la segunda para producir un 0 como el coeficiente de x_1 en la segunda ecuación. Esto, por supuesto, es imposible, ya que el algoritmo falla si $a_{11} = 0$.

Si un procedimiento numérico en realidad falla para algunos valores de los datos, el procedimiento es probablemente poco confiable para los valores de los datos *cerca* de los valores donde falla. Para probar esta afirmación, considere el sistema

$$\begin{cases} \varepsilon x_1 + x_2 = 1 \\ x_1 + x_2 = 2 \end{cases} \quad (2)$$

en el que ε es un pequeño número diferente de 0. Ahora, el algoritmo simple de la sección 7.1 funciona y después de la eliminación hacia adelante se obtiene el sistema

$$\begin{cases} \varepsilon x_1 + x_2 = 1 \\ (1 - \varepsilon^{-1})x_2 = 2 - \varepsilon^{-1} \end{cases} \quad (3)$$

En la sustitución hacia atrás, la aritmética es la siguiente:

$$x_2 = \frac{2 - \varepsilon^{-1}}{1 - \varepsilon^{-1}} \approx 1, \quad x_1 = \varepsilon^{-1}(1 - x_2) \approx 0$$

Ahora ε^{-1} será grande, por lo que si este cálculo se realiza con una computadora que tiene una longitud de palabra fija, entonces para valores pequeños de ε , tanto $(2 - \varepsilon^{-1})$ como $(1 - \varepsilon^{-1})$ se calculan como $-\varepsilon^{-1}$.

Por ejemplo, en una máquina de 8 dígitos decimales con un acumulador de 16 dígitos, cuando $\varepsilon = 10^{-9}$, se tiene que $\varepsilon^{-1} = 10^9$. Para restar, la computadora debe interpretar los números como

$$\begin{aligned} \varepsilon^{-1} &= 10^9 = 0.10000\ 000 \times 10^{10} = 0.10000\ 00000\ 00000\ 0 \times 10^{10} \\ 2 &= 0.20000\ 000 \times 10^1 = 0.00000\ 00002\ 00000\ 0 \times 10^{10} \end{aligned}$$

Por lo tanto, $(\varepsilon^{-1} - 2)$ inicialmente se calcula como $0.09999\ 99998\ 00000\ 0 \times 10^{10}$ y después se redondea a $0.10000\ 000 \times 10^{10} = \varepsilon^{-1}$.

Concluimos que para los valores de ε suficientemente cercanos a 0, la computadora calcula x_2 como 1 y entonces a x_1 como 0. Como la solución *correcta* es

$$x_1 = \frac{1}{1 - \varepsilon} \approx 1, \quad x_2 = \frac{1 - 2\varepsilon}{1 - \varepsilon} \approx 1$$

el error relativo en la solución calculada para x_1 es extremadamente grande: 100%.

En realidad, el algoritmo de eliminación gaussiana simple funciona bien en los sistemas (1) y (2) si se permutan primero las ecuaciones:

$$\begin{cases} x_1 + x_2 = 2 \\ 0x_1 + x_2 = 1 \end{cases}$$

y

$$\begin{cases} x_1 + x_2 = 2 \\ \varepsilon x_1 + x_2 = 1 \end{cases}$$

El primer sistema es fácil de resolver obteniendo $x_2 = 1$ y $x_1 = 2 - x_2 = 1$. Además, el segundo de estos sistemas se convierte en

$$\begin{cases} x_1 + x_2 = 2 \\ (1 - \varepsilon)x_2 = 1 - 2\varepsilon \end{cases}$$

después de la eliminación hacia adelante. Luego de la sustitución hacia atrás, la solución se calcula como

$$x_2 = \frac{1 - 2\varepsilon}{1 - \varepsilon} \approx 1, \quad x_1 = 2 - x_2 \approx 1$$

Observe que no tenemos que reorganizar las ecuaciones en el sistema: sólo es necesario seleccionar un diferente renglón pivote. La dificultad en el sistema (2) no se debe simplemente a que ε es pequeña, sino a que es pequeña con respecto a otros coeficientes del mismo renglón. Para comprobar esto, considere

$$\begin{cases} x_1 + \varepsilon^{-1}x_2 = \varepsilon^{-1} \\ x_1 + x_2 = 2 \end{cases} \tag{4}$$

El sistema (4) es matemáticamente equivalente al (2). El algoritmo de eliminación gaussiana simple aquí falla. Este produce el sistema triangular

$$\begin{cases} x_1 + \varepsilon^{-1}x_2 = \varepsilon^{-1} \\ (1 - \varepsilon^{-1})x_2 = 2 - \varepsilon^{-1} \end{cases}$$

Y después, en la sustitución hacia atrás, se obtiene el resultado erróneo

$$x_2 = \frac{2 - \varepsilon^{-1}}{1 - \varepsilon^{-1}} \approx 1, \quad x_1 = \varepsilon^{-1} - \varepsilon^{-1}x_2 \approx 0$$

Esta situación se puede resolver intercambiando las dos ecuaciones en (4):

$$\begin{cases} x_1 + x_2 = 2 \\ x_1 + \varepsilon^{-1}x_2 = \varepsilon^{-1} \end{cases}$$

Ahora, el algoritmo de eliminación gaussiana simple se puede aplicar, lo cual resultará en el sistema

$$\begin{cases} x_1 + x_2 = 2 \\ (\varepsilon^{-1} - 1)x_2 = \varepsilon^{-1} - 2 \end{cases}$$

La solución es

$$x_2 = \frac{\varepsilon^{-1} - 2}{\varepsilon^{-1} - 1} \approx 1, \quad x_1 = 2 - x_2 \approx 1$$

que es la solución correcta.

Pivoteo parcial y pivoteo completo parcial

La eliminación gaussiana con **pivoteo parcial** selecciona el renglón pivote como el que tiene la entrada pivote con valor absoluto máximo en las columnas principales de la submatriz reducida. Se intercambian dos renglones para mover el renglón designado a la posición del renglón pivote. La eliminación gaussiana con **pivoteo completo** selecciona la entrada pivote como la entrada pivote máxima de todas las entradas de la submatriz. (Esto complica las cosas porque se han reorganizado algunas de las incógnitas.) Para lograr esto se intercambian dos renglones y dos columnas. En la práctica, el pivoteo parcial es casi tan bueno como el pivoteo completo y requiere un trabajo mucho menor. Consulte Wilkinson [1963] para más detalles acerca de este tema. Basta escoger el número más grande en magnitud como se hace en pivoteo parcial puede funcionar bien, pero aquí escalamiento de renglón no desempeña un papel: no se consideran los tamaños relativos de las entradas en un renglón. Los sistemas de ecuaciones con coeficientes de tamaños diferentes pueden causar dificultades y deben ser vistos con recelo. A veces, una estrategia de escalada pueden mejorar estos problemas. En este libro se presenta la eliminación gaussiana con pivoteo parcial escalado y el seudocódigo contiene un esquema de pivoteo implícito.

En ciertas situaciones, la eliminación gaussiana con la estrategia de pivoteo parcial simple puede conducir a una solución incorrecta. Considere la matriz aumentada

$$\left[\begin{array}{cc|c} 2 & 2c & 2c \\ 1 & 1 & 2 \end{array} \right]$$

donde c es un parámetro que puede tomar valores numéricos muy grandes y las variables son x y y . El primer renglón se selecciona como el renglón pivote al elegir el número más grande en la primera columna. Como el multiplicador es $1/2$, un paso en el proceso de reducción de renglones nos lleva a

$$\left[\begin{array}{cc|c} 2 & 2c & 2c \\ 0 & 1 - c & 2 - c \end{array} \right]$$

Supongamos ahora que estamos trabajando con una computadora de longitud de palabra limitada. Así, en esta computadora se obtiene $1 - c \approx -c$ y $2 - c \approx -c$. En consecuencia, la computadora contiene estos números:

$$\left[\begin{array}{cc|c} 2 & 2c & 2c \\ 0 & -c & -c \end{array} \right]$$

Así, como la solución, obtenemos $y = 1$ y $x = 0$, mientras que la solución correcta es $x = y = 1$.

Por otra parte, la eliminación gaussiana con pivoteo parcial escalado selecciona el segundo renglón como el renglón pivote. Las constantes de escalamiento son $(2c, 1)$ y el mayor de los dos cocientes para seleccionar el renglón pivote de $\{2/(2c), 1\}$ es el segundo. Ahora el multiplicador es 2 y un paso en el proceso de reducción de renglones nos lleva a

$$\left[\begin{array}{cc|c} 0 & 2c - 2 & 2c - 4 \\ 1 & 1 & 2 \end{array} \right]$$

En nuestra computadora de longitud de la palabra limitada, encontramos $2c - 2 \approx 2c$ y $2c - 4 \approx 2c$. Por consiguiente, la computadora contiene estos números:

$$\left[\begin{array}{cc|c} 0 & 2c & 2c \\ 1 & 1 & 2 \end{array} \right]$$

Ahora obtenemos la solución correcta, $y = 1$ y $x = 1$.

Eliminación gaussiana con pivoteo escalado parcial

Estos ejemplos simples deben dejar claro que el *orden* con que se tratan las ecuaciones afecta significativamente la exactitud del algoritmo de eliminación en la computadora. En el algoritmo de eliminación gaussiana simple usamos la primera ecuación para eliminar x_1 de las ecuaciones que le seguían. Luego usamos la segunda ecuación para eliminar x_2 de las ecuaciones que le seguían y así sucesivamente. El orden en que se utilizan las ecuaciones como ecuaciones pivote es el orden **natural** ($1, 2, \dots, n$). Observe que la última ecuación (ecuación número n) *no* se utiliza como una ecuación que funciona en el orden natural: en ningún momento múltiplos de ésta se restan de otras ecuaciones en el algoritmo simple.

De los ejemplos anteriores es evidente que se requiere una estrategia de selección de los nuevos pivotes en cada etapa de eliminación gaussiana. Tal vez el mejor método es el **pivoteo completo**, que implica búsquedas en todas las entradas de las submatrices para la entrada con valor absoluto más grande y después intercambia renglones y columnas para moverlas a la posición de pivote. Esto resulta bastante caro, ya que implica una gran cantidad de búsqueda y movimiento de datos. Sin embargo, sólo la búsqueda de la primera columna de la submatriz en cada etapa logra la mayor parte de lo que se necesita (evitando pivotes pequeños o nulos). Esto es **pivoteo parcial** y es el método más común. Esto no implica un examen de los elementos de los renglones, ya que sólo se fija en las entradas de la columna. Defendemos una estrategia que simule una escalada de los vectores renglón y luego seleccione un elemento pivote que sea relativamente la entrada más grande en una columna. Además, en lugar de intercambiar los renglones para mover el elemento deseado en la posición de pivote, utilizamos un arreglo indizado para evitar el movimiento de datos. Este procedimiento no es tan caro como el pivoteo completo y va más allá del pivoteo parcial al incluir un examen de todos los elementos de la matriz original. Por supuesto, se pueden utilizar otras estrategias para la selección de los elementos pivote.

El algoritmo de eliminación gaussiana que ahora describiremos usa ecuaciones en un orden que está determinado por el sistema real que se está resolviendo. Por ejemplo, si se le pidió al algoritmo resolver los sistemas (1) o (2), el orden en que se utilizarían las ecuaciones como ecuaciones pivote no sería el orden natural $\{1, 2\}$, sino más bien, $\{2, 1\}$. Este orden lo determina automáticamente el programa de la computadora. El orden en el que se emplean las ecuaciones se denota por el vector renglón $[\ell_1, \ell_2, \dots, \ell_n]$, donde ℓ_n no se utiliza realmente en la fase de eliminación hacia adelante. Aquí, los ℓ_i son números enteros del 1 al n en un posible orden diferente. Llamamos a $\ell = [\ell_1, \ell_2, \dots, \ell_n]$ el **vector de índices**. La estrategia que se describe a continuación para determinar el vector índice se denomina **pivoteo parcial escalado**.

En un principio, se debe calcular un **factor de escala** para cada ecuación en el sistema. En referencia con la notación de la sección 7.1, definimos

$$s_i = \max_{1 \leq j \leq n} |a_{ij}| \quad (1 \leq i \leq n)$$

Estos números n se registran en el **vector de escala** $s = [s_1, s_2, \dots, s_n]$.

Al comenzar el proceso de eliminación hacia adelante, no se usa de manera arbitraria la primera ecuación como la ecuación pivote. En su lugar, usamos la ecuación para la que el cociente $|a_{i,1}|/s_i$ es mayor. Sea ℓ_1 el primer índice para el que este cociente es mayor. Ahora, múltiplos adecuados de la ecuación ℓ_1 se restan de las otras ecuaciones para crear ceros como los coeficientes para cada x_1 , excepto en la ecuación pivote.

La mejor manera de hacer el seguimiento de los índices es la siguiente: al principio, se define al vector índice ℓ como $[\ell_1, \ell_2, \dots, \ell_n] = [1, 2, \dots, n]$. Seleccione j como el primer índice

asociado con el cociente más grande del conjunto:

$$\left\{ \frac{|a_{\ell_i 1}|}{s_{\ell_i}} : 1 \leq i \leq n \right\}$$

Ahora intercambie ℓ_j con ℓ_1 en el vector índice ℓ . A continuación, use los multiplicadores

$$\frac{a_{\ell_i 1}}{a_{\ell_1 1}}$$

por el renglón ℓ_1 y réstelos de las ecuaciones ℓ_i para $2 \leq i \leq n$. Es importante señalar que sólo se intercambian las entradas en ℓ y *no* las ecuaciones. Esto elimina el tiempo y el proceso innecesario de mover los coeficientes de las ecuaciones por toda la memoria de la computadora.

En el segundo paso, los cocientes

$$\left\{ \frac{|a_{\ell_i, 2}|}{s_{\ell_i}} : 2 \leq i \leq n \right\}$$

se revisan meticulosamente. Si j es el primer índice para el cociente más grande, se intercambia ℓ_j con ℓ_2 en ℓ . Despues los multiplicadores

$$\frac{a_{\ell_i 2}}{a_{\ell_2 2}}$$

por la ecuación ℓ_2 se restan de las ecuaciones ℓ_i para $3 \leq i \leq n$.

En el paso k , se selecciona j como el primer índice correspondiente al más grande de los cocientes,

$$\left\{ \frac{|a_{\ell_i k}|}{s_{\ell_i}} : k \leq i \leq n \right\}$$

y se intercambia ℓ_j y ℓ_k en el vector índice ℓ . Despues los multiplicadores

$$\frac{a_{\ell_i k}}{a_{\ell_k k}}$$

por la ecuación pivote ℓ_k se restan de las ecuaciones ℓ_i para $k + 1 \leq i \leq n$.

Observe que los factores de escala *no* cambian después de cada paso de pivote. Intuitivamente, uno podría pensar que después de cada paso en el algoritmo gaussiano los coeficientes (modificados) restantes se deben utilizar para volver a calcular los factores de escala en lugar de utilizar el vector de escala original. Por supuesto, esto se podría hacer, pero en general se cree que los cálculos adicionales implicados en este procedimiento no valen la pena en la mayoría de los sistemas lineales. Se invita al lector a estudiar este tema (véase el problema de cómputo 7.2.16).

EJEMPLO 1

Resuelva este sistema de ecuaciones lineales:

$$\begin{cases} 0.0001x + y = 1 \\ x + y = 2 \end{cases}$$

sin usar pivoteo, pivoteo parcial y pivoteo parcial escalado. Realícelo con a lo más cinco dígitos significativos de precisión (redondeo) para ver cómo los cálculos con precisión finita y los errores de redondeo afectan las estimaciones.

- Solución** Por sustitución directa, es fácil comprobar que la solución verdadera es $x = 1.0001$ y $y = 0.99990$ con cinco dígitos significativos.

Sin pivoteo, la primera ecuación en el sistema original es la ecuación pivote y el multiplicador es $xmult = 1/0.0001 = 10000$. Multiplicando la primera ecuación por este multiplicador y restando el resultado a la segunda ecuación, los cálculos necesarios son $(10000)(0.0001) - 1 = 0$, $(10000)(1) - 1 = 9999$ y $(10000)(1) - 2 = 9998$. El nuevo sistema de ecuaciones es

$$\begin{cases} 0.0001x + y = 1 \\ 9999y = 9998 \end{cases}$$

De la segunda ecuación, obtenemos $y = 9998/9999 \approx 0.99990$. Usando este resultado y la primera ecuación encontramos $0.0001x = 1 - y = 1 - 0.99990 = 0.0001$ y $x = 0.0001/0.0001 = 1$. Observe que hemos perdido el último dígito significativo en el valor correcto de x .

Repetimos la solución con pivoteo parcial en el sistema original. Examinando la primera columna de los coeficientes de x ($0.0001, 1$), vemos que la segunda es más grande, por lo que se utiliza la segunda ecuación como la ecuación pivote. Podemos intercambiar las dos ecuaciones, con lo que se obtiene

$$\begin{cases} x + y = 2 \\ 0.0001x + y = 1 \end{cases}$$

El multiplicador es $xmult = 0.0001/1 = 0.0001$. Este múltiplo de la primera ecuación se resta de la segunda ecuación. Los cálculos son $(-0.0001)(1) + 0.0001 = 0$, $(0.0001)(1) - 1 = 0.99990$ y $(0.0001)(2) - 1 = 0.99980$. El nuevo sistema de ecuaciones es

$$\begin{cases} x + y = 2 \\ 0.99990y = 0.99980 \end{cases}$$

Obtenemos $y = 0.99980/0.99990 \approx 0.99990$. Ahora, usando la segunda ecuación y este valor encontramos $x = 2 - y = 2 - 0.99990 = 1.0001$. Ambos valores calculados de x y de y son correctos con cinco dígitos significativos.

Repetimos la solución con pivoteo parcial escalado en el sistema original. Puesto que las constantes de escalamiento son $s = (1, 1)$ y los cocientes para la determinación de la ecuación pivote son $(0.0001/1, 1/1)$, la segunda ecuación es ahora la ecuación pivote. No se intercambian realmente las ecuaciones, pero se puede trabajar con un arreglo de índice $\ell = (2, 1)$ que nos dice que usemos la segunda ecuación como la primera ecuación pivote. El resto de los cálculos son como el anterior para pivoteo parcial. Los valores calculados de x y y son correctos con cinco dígitos significativos.

No podemos prometer que el pivoteo parcial escalado será mejor que el pivoteo parcial, pero está claro que tiene algunas ventajas. Por ejemplo, supongamos que alguien quiere obligar a la primera ecuación en el sistema original a ser la ecuación pivote y la multiplica por un número grande como 20000, de lo que resulta

$$\begin{cases} 2x + 20000y = 20000 \\ x + y = 2 \end{cases}$$

El pivoteo parcial ignora el hecho de que los coeficientes en la primera ecuación difieren de los otros en varios órdenes de magnitud y selecciona la primera ecuación como la ecuación pivote. Sin embargo, el pivoteo parcial escalado utiliza las constantes de escalamiento (20000, 1) y los cocientes para determinar que las ecuaciones pivote son $(2/20000, 1/1)$. El pivoteo parcial escalado continua ¡seleccionando a la segunda ecuación como la ecuación pivote!

Un gran ejemplo numérico

No estamos completamente listos para escribir un seudocódigo, pero veamos lo que ha sido descrito en un ejemplo concreto. Consideré

$$\begin{bmatrix} 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \\ 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -19 \\ -34 \\ 16 \\ 26 \end{bmatrix} \quad (5)$$

El vector índice es $\ell = [1, 2, 3, 4]$ inicialmente. El vector de escala no cambia en todo el procedimiento y es $s = [13, 18, 6, 12]$. Para determinar el primer renglón pivote, veamos los cuatro cocientes:

$$\left\{ \frac{|a_{\ell_i,1}|}{s_{\ell_i}} : i = 1, 2, 3, 4 \right\} = \left\{ \frac{3}{13}, \frac{6}{18}, \frac{6}{6}, \frac{12}{12} \right\} \approx \{0.23, 0.33, 1.0, 1.0\}$$

Se selecciona el índice j como la *primera* ocurrencia del mayor valor de estos cocientes. En este ejemplo, el mayor de estos se produce para el índice $j = 3$. Por ello, el renglón tres es la ecuación pivote en el paso 1 ($k = 1$) del proceso de eliminación. En el vector índice ℓ , las entradas ℓ_k y ℓ_j se intercambian de forma que el nuevo vector índice es $\ell = [3, 2, 1, 4]$. Así, la ecuación pivote es ℓ_3 , que es $\ell_1 = 3$. Ahora se restan múltiplos adecuados de la tercera ecuación de las otras ecuaciones para crear ceros como coeficientes de x_1 en cada una de esas ecuaciones. Explícitamente, $\frac{1}{2}$ veces el renglón tres se resta del renglón uno, -1 veces el renglón tres se resta del renglón dos y 2 veces el renglón tres se resta del renglón cuatro. El resultado es

$$\begin{bmatrix} 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \\ 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -27 \\ -18 \\ 16 \\ -6 \end{bmatrix}$$

En el paso siguiente ($k = 2$), usamos el vector índice $\ell = [3, 2, 1, 4]$ y se analizan los cocientes correspondientes a los renglones dos, uno y cuatro:

$$\left\{ \frac{|a_{\ell_i,2}|}{s_{\ell_i}} : i = 2, 3, 4 \right\} = \left\{ \frac{2}{18}, \frac{12}{13}, \frac{4}{12} \right\} \approx \{0.11, 0.92, 0.33\}$$

buscando el de mayor valor. Encontramos que el mayor es el segundo cociente y por lo tanto hacemos $j = 2$ e intercambiamos ℓ_k con ℓ_j en el vector índice. Así, éste se convierte en $\ell = [3, 1, 2, 4]$. La ecuación pivote para el paso 2 en la eliminación es ahora el renglón uno, y $\ell_2 = 1$. A continuación, los múltiplos de la primera ecuación se restan de la segunda y de la cuarta ecuaciones. Los múltiplos adecuados son $-\frac{1}{6}$ y $\frac{1}{3}$, respectivamente. El resultado es

$$\begin{bmatrix} 0 & -12 & 8 & 1 \\ 0 & 0 & \frac{13}{3} & -\frac{83}{6} \\ 6 & -2 & 2 & 4 \\ 0 & 0 & -\frac{2}{3} & \frac{5}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -27 \\ -\frac{45}{2} \\ 16 \\ 3 \end{bmatrix}$$

El tercero y último paso ($k = 3$) es examinar los cocientes correspondientes a los renglones dos y cuatro:

$$\left\{ \frac{|a_{\ell_i,3}|}{s_{\ell_i}} : i = 3, 4 \right\} = \left\{ \frac{13/3}{18}, \frac{2/3}{12} \right\} \approx \{0.24, 0.06\}$$

con el vector índice $\ell = [3, 1, 2, 4]$. El valor más grande es el primero, por lo que hacemos $j = 3$. Como este paso es $k = 3$, intercambiar ℓ_k con ℓ_j no altera el vector índice, $\ell = [3, 1, 2, 4]$. La ecuación pivote es el renglón dos y $\ell_3 = 2$, y restamos $-\frac{2}{13}$ veces la segunda ecuación de la cuarta ecuación. Así, la fase de eliminación hacia adelante termina dando el sistema final

$$\left[\begin{array}{cccc} 0 & -12 & 8 & 1 \\ 0 & 0 & \frac{13}{3} & -\frac{83}{6} \\ 6 & -2 & 2 & 4 \\ 0 & 0 & 0 & -\frac{6}{13} \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -27 \\ -\frac{45}{2} \\ 16 \\ -\frac{6}{13} \end{bmatrix}$$

El orden en que se seleccionaron las ecuaciones pivote se presenta en el vector índice final $\ell = [3, 1, 2, 4]$.

Ahora, leyendo las entradas en el vector índice de la última a la primera, tenemos el orden en el que se va a realizar la sustitución hacia atrás. La solución se obtiene usando la ecuación $\ell_4 = 4$ para determinar x_4 , y después la ecuación $\ell_3 = 2$ para encontrar x_3 y así sucesivamente. Realizando los cálculos obtenemos

$$\begin{aligned} x_4 &= \frac{1}{-6/13}[-6/13] = 1 \\ x_3 &= \frac{1}{13/3}[(-45/2) + (83/6)(1)] = -2 \\ x_2 &= \frac{1}{-12}[-27 - 8(-2) - 1(1)] = 1 \\ x_1 &= \frac{1}{6}[16 + 2(1) - 2(-2) - 4(1)] = 3 \end{aligned}$$

Por tanto, la solución es

$$\mathbf{x} = [3 \quad 1 \quad -2 \quad 1]^T$$

Seudocódigo

El algoritmo como se programó realiza la fase de eliminación hacia adelante en el arreglo de coeficientes (a_{ij}) solamente. El arreglo del lado derecho (b_i) se trata en la siguiente fase. Este método se adoptó porque es más eficiente si se deben resolver varios sistemas con el mismo arreglo (a_{ij}), pero con diferentes arreglos (b_i). Como queremos tratar a (b_i) más tarde, es necesario no sólo almacenar el arreglo de índices sino también los diferentes multiplicadores que se utilizaron. Éstos son convenientemente almacenados en el arreglo (a_{ij}) en las posiciones donde se habían creado las entradas 0. Estos multiplicadores son útiles en la construcción de la factorización LU de la matriz A , como se explica en la sección 8.1.

Ahora estamos listos para escribir un procedimiento para la eliminación hacia adelante con pivoteo parcial escalado. Nuestro método consiste en modificar el procedimiento *Gauss_Simple* de la sección 7.1 introduciendo los arreglos escalados e indizados. El procedimiento que realiza la eliminación gaussiana con pivoteo parcial escalado del arreglo cuadrado (a_{ij}) se llama *Gauss*. Su secuencia de llamado es $(n, (a_{ij}), (\ell_i))$, donde (a_{ij}) es el arreglo de coeficientes de $n \times n$ y (ℓ_i) es el arreglo de índices ℓ . En el seudocódigo, (s) es el arreglo de escala, s .

```

procedure Gauss( $n, (a_{ij}), (\ell_i)$ )
integer  $i, j, k, n$ ; real  $r, rmax, smax, xm$ 
real array  $(a_{ij})_{1:n \times 1:n}, (\ell_i)_{1:n}$ ; real array allocate  $(s_i)_{1:n}$ 
for  $i = 1$  to  $n$  do
     $\ell_i \leftarrow i$ 
     $smax \leftarrow 0$ 
    for  $j = 1$  to  $n$  do
         $smax \leftarrow \max(smax, |a_{ij}|)$ 
    end for
     $s_i \leftarrow smax$ 
end for
for  $k = 1$  to  $n - 1$  do
     $rmax \leftarrow 0$ 
    for  $i = k$  to  $n$  do
         $r \leftarrow |a_{\ell_i, k}| / s_{\ell_i}$ 
        if ( $r > rmax$ ) then
             $rmax \leftarrow r$ 
             $j \leftarrow i$ 
        end if
    end for
     $\ell_j \leftrightarrow \ell_k$ 
    for  $i = k + 1$  to  $n$  do
         $xmult \leftarrow a_{\ell_i, k} / a_{\ell_k, k}$ 
         $a_{\ell_i, k} \leftarrow xm$ 
        for  $j = k + 1$  to  $n$  do
             $a_{\ell_i, j} \leftarrow a_{\ell_i, j} - (xm) a_{\ell_k, j}$ 
        end for
    end for
end for
deallocate array  $(s_i)$ 
end procedure Gauss

```

Ahora se presenta una explicación detallada del procedimiento anterior. En el primer ciclo, se está estableciendo la forma inicial del arreglo de índices, a saber, $\ell_i = i$. Entonces se calcula el arreglo de escala (s_i) .

El enunciado **for** $k = 1$ **to** $n - 1$ **do** inicia el ciclo principal exterior. El índice k es el subíndice de la variable cuyos coeficientes se harán 0 en el arreglo (a_{ij}) ; es decir, k es el índice de la columna en la que se crean los nuevos ceros. Recuerde que los ceros en el arreglo (a_{ij}) en realidad no aparecen porque los lugares de almacenamiento se utilizan para los multiplicadores. Este hecho se puede ver en renglón del procedimiento en el que xm se almacena en el arreglo (a_{ij}) (véase la sección 8.1 en la factorización *LU* de A para qué se hace esto).

Una vez que se ha establecido k , la primera tarea es seleccionar el renglón pivote correcto, lo que se realiza mediante el cálculo de $|a_{\ell_i, k}| / s_{\ell_i}$ para $i = k, k + 1, \dots, n$. El siguiente conjunto de renglones en el pseudocódigo es el cálculo de este cociente mayor, llamado $rmax$ en la rutina y el índice j donde se presenta. Después, se intercambian ℓ_k y ℓ_j en el arreglo (ℓ_i) .

Las modificaciones aritméticas en el arreglo (a_{ij}) , debido a la resta de múltiplos del renglón ℓ_k de los renglones $\ell_{k+1}, \ell_{k+2}, \dots, \ell_n$ ocurren todas en los renglones finales. En primer lugar, el multiplicador se calcula y se almacena y después se hace la resta en un ciclo.

Precaución: los valores en el arreglo (a_{ij}) que resultan de la *salida* del procedimiento *Gauss* no son los mismos que los del arreglo (a_{ij}) en la *entrada*. Si se debe conservar el arreglo original, se debe guardar un duplicado del mismo en otro arreglo.

En el procedimiento *Gauss_Simple* de eliminación gaussiana simple de la sección 7.1, se ha modificado el lado derecho b durante la fase de eliminación hacia delante; sin embargo, esto no se hizo en el procedimiento *Gauss*. Por tanto, necesitamos actualizar b antes de considerar la fase de sustitución hacia atrás. Por simplicidad, hablamos de la actualización de b primero para la eliminación simple hacia adelante. Excluyendo el seudocódigo *Gauss_Simple* que implica al arreglo (b) en la fase de eliminación hacia adelante, obtenemos

```

for  $k = 1$  to  $n - 1$  do
    for  $i = k + 1$  to  $n$  do
         $b_i = b_i - a_{ik}b_k$ 
    end for
end for

```

Esto actualiza el arreglo (b) con base en los multiplicadores almacenados del arreglo (a_{ij}) . Cuando se hace el pivoteo parcial escalado en la fase de eliminación hacia delante, como en el procedimiento *Gauss*, los multiplicadores para cada paso no están uno abajo de otro en el arreglo (a_{ij}) , sino que están mezclados. Para desentrañar esta situación, todo lo que tenemos que hacer es introducir el arreglo de índices (ℓ_i) en el seudocódigo anterior:

```

for  $k = 1$  to  $n - 1$  do
    for  $i = k + 1$  to  $n$  do
         $b_{\ell_i} = b_{\ell_i} - a_{\ell_i k}b_{\ell_k}$ 
    end for
end for

```

Después de que se ha procesado el arreglo b en la eliminación hacia adelante, se realiza el proceso de sustitución hacia atrás. Se comienza resolviendo la ecuación

$$a_{\ell_n, n}x_n = b_{\ell_n} \quad (6)$$

donde

$$x_n = \frac{b_{\ell_n}}{a_{\ell_n, n}}$$

Entonces la ecuación

$$a_{\ell_{n-1}, n-1}x_{n-1} + a_{\ell_{n-1}, n}x_n = b_{\ell_{n-1}}$$

se resuelve para x_{n-1} :

$$x_{n-1} = \frac{1}{a_{\ell_{n-1}, n-1}}(b_{\ell_{n-1}} - a_{\ell_{n-1}, n}x_n)$$

Después de que se han determinado $x_n, x_{n-1}, \dots, x_{i+1}$, se encuentra x_i de la ecuación

$$a_{\ell_i,i}x_i + a_{\ell_i,i+1}x_{i+1} + \cdots + a_{\ell_i,n}x_n = b_{\ell_i}$$

cuya solución es

$$x_i = \frac{1}{a_{\ell_i,i}} \left(b_{\ell_i} - \sum_{j=i+1}^n a_{\ell_i,j}x_j \right) \quad (7)$$

Excepto por la presencia del vector de índices ℓ_i , esto es similar a la fórmula de sustitución hacia atrás (7) en la sección 7.1 obtenida en la eliminación gaussiana simple.

El procedimiento para procesar el arreglo b y realizar la fase de sustitución hacia atrás se presenta a continuación:

```

procedure Solución(n,(aij),(ℓi),(bi),(xi))
integer i, k, n;    real suma
real array (aij)1:n x 1:n, (ℓi)1:n, (bi)1:n, (xi)1:n
for k = 1 to n - 1 do
    for i = k + 1 to n do
        bℓ_i ← bℓ_i - aℓ_i,kbℓ_k
    end for
end for
xn ← bℓ_n / aℓ_n,n
for i = n - 1 to 1 step -1 do
    suma ← bℓ_i
    for j = i + 1 to n do
        suma ← suma - aℓ_i,jxj
    end for
    xi ← suma / aℓ_i,i
end for
end procedure Solución

```

Aquí, en el primer ciclo se realiza el proceso de eliminación hacia adelante en el arreglo (b) , usando los arreglos (a_{ij}) y (ℓ_i) que resultan del procedimiento *Gauss*. En el siguiente renglón se realiza la solución de la ecuación (6). La parte final realiza la ecuación (7). La variable *suma* es una variable temporal para acumular los términos encerrados entre paréntesis.

Como con la mayoría de los seudocódigos de este libro, los que se encuentran en este capítulo sólo contienen los ingredientes básicos para un buen software matemático. No son adecuados como códigos de *producción* por varias razones. Por ejemplo, se omiten los procedimientos para la optimización de código. Además, los procedimientos no advierten las dificultades que puedan surgir, tales como ¡la división entre cero! El software de propósito general debe ser **robusto**, es decir, deben prever todas las situaciones posibles y tratar cada una de una manera prescrita (véase el problema de cómputo 7.2.11).

Conteo de operaciones largas

La solución de grandes sistemas de ecuaciones lineales puede ser costosa en tiempo de máquina. Para entender por qué, vamos a realizar un conteo de las operaciones en los dos algoritmos de los códigos que se han dado. Contamos sólo las multiplicaciones y divisiones (operaciones largas)

porque consumen más tiempo que las sumas. Además, agrupamos multiplicaciones y divisiones a pesar de que la división es más lenta que la multiplicación. En las computadoras modernas, todas las operaciones de punto flotante se hacen en hardware, de modo que las operaciones largas pueden no ser tan importantes, pero esto aún da una idea del costo operacional de la eliminación gaussiana.

Considere primero el procedimiento *Gauss*. En el paso 1, la elección de un elemento pivote requiere el cálculo de los n cocientes, es decir, n divisiones. Después, para los renglones $\ell_2, \ell_3, \dots, \ell_n$ primero calculamos un multiplicador y después lo restamos del renglón ℓ_i tras multiplicarlo por ℓ_1 . El cero que se está creando en este proceso *no* se calcula. Así, la eliminación requiere $n - 1$ multiplicaciones por renglón. Si se incluyen el cálculo del multiplicador, hay n operaciones largas (multiplicaciones o divisiones) por renglón. Hay $n - 1$ renglones por procesar con un total de $n(n - 1)$ operaciones. Si sumamos el costo del cálculo de los cocientes, se necesita un total de n^2 operaciones para el paso 1.

El siguiente paso es parecido al paso 1, excepto que el renglón ℓ_1 no se afecta, ni se crea y almacena la columna de multiplicadores en el paso 1. Por lo tanto, el paso 2 requiere $(n - 1)^2$ multiplicaciones o divisiones porque opera en un sistema sin el renglón ℓ_1 y sin la columna 1. Siguiendo este razonamiento, podemos concluir que el número total de operaciones largas para el procedimiento *Gauss* es

$$n^2 + (n - 1)^2 + (n - 2)^2 + \cdots + 4^2 + 3^2 + 2^2 = \frac{n}{6}(n + 1)(2n + 1) - 1 \approx \frac{n^3}{3}$$

(La deducción de esta fórmula se describe en el problema 7.2.16.) Observe que el número de operaciones largas en este procedimiento crece como $n^3/3$, el término dominante.

Ahora considere el procedimiento *Solución*. El procesamiento hacia adelante del arreglo (b_i) implica $n - 1$ pasos. El primer paso contiene $n - 1$ multiplicaciones, el segundo contiene $n - 2$ multiplicaciones y así sucesivamente. El total de procesamiento hacia adelante del arreglo (b_i) por tanto es

$$(n - 1) + (n - 2) + \cdots + 3 + 2 + 1 = \frac{n}{2}(n - 1)$$

(véase el problema 7.2.15.) En el procedimiento de sustitución hacia atrás está implicada una operación larga en el primer paso, dos en el segundo paso y así sucesivamente. El total es de

$$1 + 2 + 3 + \cdots + n = \frac{n}{2}(n + 1)$$

Así, el procedimiento *Solución* implica n^2 operaciones. Para resumir:

TEOREMA 1

Teorema de operaciones largas

La fase de eliminación hacia adelante del algoritmo de eliminación gaussiana con pivoteo parcial escalado, si se aplicara sólo al arreglo de coeficientes de $n \times n$, implica aproximadamente $n^3/3$ operaciones largas (multiplicaciones o divisiones). Resolver para x requiere n^2 operaciones largas adicionales.

Una forma intuitiva de pensar en este resultado es que el algoritmo de eliminación gaussiana implica un triple ciclo for anidado. Así, una estructura algorítmica $\mathcal{O}(n^3)$ está conduciendo el proceso de eliminación y el trabajo está muy influenciado por el cubo del número de ecuaciones y de incógnitas.

Estabilidad numérica

La **estabilidad numérica** de un algoritmo numérico está relacionada con la exactitud del procedimiento. Un algoritmo puede tener diferentes niveles de estabilidad numérica porque muchos de los cálculos se pueden lograr de varias maneras que son equivalentes algebraicamente pero que pueden producir resultados diferentes. Es deseable un algoritmo numérico robusto con un alto nivel de estabilidad numérica. La eliminación gaussiana es estable numéricamente para las matrices diagonales estrictamente dominantes o para las matrices simétricas definidas positivas. (Estas son propiedades que se presentarán en las secciones 7.3 y 8.1, respectivamente.) Para matrices en general con una estructura densa, la eliminación gaussiana con pivoteo parcial suele ser numéricamente estable en la práctica. Sin embargo, existen ejemplos patológicos inestables en los que puede fallar. Para más detalles consulte Golub y Van Loan [1996] y Highman [1996].

Una primera versión de la eliminación gaussiana se puede encontrar en un libro chino de matemáticas que data del año 150 a.C.

Escalamiento

No se debe confundir *escalamiento* en la eliminación gaussiana (que *no* es recomendable) con el análisis de pivoteo parcial *escalado* en la eliminación gaussiana.

La palabra **escalamiento** tiene más de un significado. En realidad, podría significar dividir cada fila entre su elemento máximo en valor absoluto. Desde luego, no abogamos por ello. En otras palabras, no se recomienda en absoluto el escalamiento de la matriz. Sin embargo, calculamos un arreglo de escala y lo usamos para seleccionar el elemento pivote en la eliminación gaussiana con pivoteo parcial escalado. En realidad no se escalan los renglones; sólo mantenemos un vector de las “normas infinitas de renglón”, es decir, el elemento con mayor valor absoluto de cada renglón. Esto y la necesidad de un vector de índices para seguir la pista de los renglones pivote hacen al algoritmo algo complicado, pero ese es el precio por lograr un cierto grado de robustez en el procedimiento.

El simple ejemplo de 2×2 en la ecuación (4) muestra que el escalamiento no ayuda en la elección de un buen renglón pivote. En este ejemplo no se usa el escalamiento. El escalamiento de los renglones se contempla en el problema 7.2.23 y en el problema de cómputo 7.2.17. Observe que este procedimiento requiere al menos n^2 operaciones aritméticas. Insistimos, no lo estamos recomendando para un código de propósito general.

Algunos códigos en realidad mueven los renglones en función del almacenamiento. Debido a que no se debe hacer en la práctica, no lo hacemos en el código, ya que podría ser engañoso. Además, para evitar inducir a error al lector ocasional, llamamos a nuestro algoritmo inicial (en la sección anterior) *simple*, esperando que nadie se confunda con un código confiable.

Resumen

(1) En el ejercicio de eliminación gaussiana, el pivoteo parcial es muy recomendable para evitar pivotes cero y pivotes pequeños. En la eliminación gaussiana con pivoteo parcial escalado, se utiliza un **vector de escala** $s = [s_1, s_2, \dots, s_n]^T$ en el que

$$s_i = \max_{1 \leq j \leq n} |a_{ij}| \quad (1 \leq i \leq n)$$

y un **vector de índices** $\ell = [\ell_1, \ell_2, \dots, \ell_n]^T$, que inicialmente se establece como $\ell = [1, 2, \dots, n]^T$. El vector o arreglo de escala se establece una vez al comienzo del algoritmo. Los elementos en el vector o arreglo de índices están intercambiados en lugar de los renglones de la matriz A , lo que

reduce la cantidad de movimiento de datos considerablemente. El paso clave en el procedimiento de pivoteo es seleccionar que j sea el primer índice asociado con el cociente más grande en el conjunto

$$\left\{ \frac{|a_{\ell_i,k}|}{s_{\ell_i}} : k \leq i \leq n \right\}$$

e intercambiar ℓ_j con ℓ_k en el arreglo de índices ℓ . Después use los multiplicadores

$$\frac{a_{\ell_i,k}}{a_{\ell_k,k}}$$

por los renglones ℓ_k y réstelos de las ecuaciones ℓ_i para $k+1 \leq i \leq n$. La **eliminación hacia atrás** de la ecuación ℓ_i para $\ell_{k+1} \leq \ell_i \leq \ell_n$ es

$$\begin{cases} a_{\ell_i,j} \leftarrow a_{\ell_i,j} - (a_{\ell_i,k}/a_{kk})a_{kj} & (\ell_k \leq \ell_j \leq \ell_n) \\ b_{\ell_i} \leftarrow b_{\ell_i} - (a_{\ell_i,k}/a_{kk})b_{\ell_k} \end{cases}$$

Los pasos que implica el vector b generalmente se realizan por separado poco antes de la fase de sustitución hacia atrás, lo que llamamos *actualización* del lado derecho. La **sustitución hacia atrás** es

$$x_i = \frac{1}{a_{\ell_i,i}} \left(b_{\ell_i} - \sum_{j=i+1}^n a_{\ell_i,j}x_j \right) \quad (i = n, n-1, n-2, \dots, 1)$$

(2) Para que un sistema de ecuaciones lineales de $n \times n$, $Ax = b$, la fase de eliminación hacia adelante de la eliminación gaussiana con pivoteo parcial escalado implica aproximadamente $n^3/3$ operaciones largas (multiplicaciones o divisiones), mientras que la sustitución hacia atrás sólo requiere n^2 operaciones largas.

Problemas 7.2

- ^a1. Muestre cómo la eliminación gaussiana con pivoteo parcial escalado funciona con la siguiente matriz A :

$$\begin{bmatrix} 2 & 3 & -4 & 1 \\ 1 & -1 & 0 & -2 \\ 3 & 3 & 4 & 3 \\ 4 & 1 & 0 & 4 \end{bmatrix}$$

- ^a2. Resuelva el siguiente sistema mediante eliminación gaussiana con pivoteo parcial escalado:

$$\begin{bmatrix} 1 & -1 & 2 \\ -2 & 1 & -1 \\ 4 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ -1 \end{bmatrix}$$

Muestre matrices intermedias en cada paso.

- ^a3. Realice eliminación gaussiana con pivoteo parcial escalado de la matriz

$$\begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 3 & -1 \\ 3 & -3 & 0 & 6 \\ 0 & 2 & 4 & -6 \end{bmatrix}$$

Muestre matrices intermedias.

4. Considere la matriz

$$\begin{bmatrix} -0.0013 & 56.4972 & 123.4567 & 987.6543 \\ 0.0000 & -0.0145 & 8.8990 & 833.3333 \\ 0.0000 & 102.7513 & -7.6543 & 69.6869 \\ 0.0000 & -1.3131 & -9876.5432 & 100.0001 \end{bmatrix}$$

Identifique la entrada que se utilizará como el siguiente elemento pivote de la eliminación gaussiana simple, de la eliminación gaussiana con pivoteo parcial (el vector de escala es [1, 1, 1, 1]) y de la eliminación gaussiana con pivoteo parcial escalado (el vector de escala es [987.6543, 46.79, 256.29, 1.096]).

5. Sin usar la computadora, determine el contenido final del arreglo (a_{ij}) después de que el procedimiento *Gauss* ha procesado el arreglo siguiente. Indique los multiplicadores subrayándolos.

$$\begin{bmatrix} 1 & 3 & 2 & 1 \\ 4 & 2 & 1 & 2 \\ 2 & 1 & 2 & 3 \\ 1 & 2 & 4 & 1 \end{bmatrix}$$

6. Si el algoritmo de eliminación gaussiana con pivoteo parcial escalado se utiliza en la matriz que se muestra, ¿cuál es el vector de escala? ¿Cuál es el segundo renglón pivote?

$$\begin{bmatrix} 4 & 7 & 3 \\ 1 & 3 & 2 \\ 2 & -4 & -1 \end{bmatrix}$$

7. Si el algoritmo de eliminación gaussiana con pivoteo parcial escalado se utiliza en el ejemplo mostrado, ¿cuál renglón será seleccionado como el tercer renglón pivote?

$$\begin{bmatrix} 8 & -1 & 4 & 9 & 2 \\ 1 & 0 & 3 & 9 & 7 \\ -5 & 0 & 1 & 3 & 5 \\ 4 & 3 & 2 & 2 & 7 \\ 3 & 0 & 0 & 0 & 9 \end{bmatrix}$$

8. Resuelva el sistema

$$\begin{cases} 2x_1 + 4x_2 - 2x_3 = 6 \\ x_1 + 3x_2 + 4x_3 = -1 \\ 5x_1 + 2x_2 = 2 \end{cases}$$

mediante la eliminación gaussiana con pivoteo parcial escalado. Muestre los resultados intermedios en cada paso, en particular, muestre los vectores de escala y de índices.

9. Considere el sistema lineal

$$\begin{cases} 2x_1 + 3x_2 = 8 \\ -x_1 + 2x_2 - x_3 = 0 \\ 3x_1 + 2x_3 = 9 \end{cases}$$

Resuelva para x_1, x_2, x_3 mediante eliminación gaussiana con pivoteo parcial escalado. Muestre matrices y vectores intermedios.

“10. Considere el sistema lineal de ecuaciones

$$\begin{cases} -x_1 + x_2 - 3x_4 = 4 \\ x_1 + 3x_3 + x_4 = 0 \\ x_2 - x_3 - x_4 = 3 \\ 3x_1 + x_3 + 2x_4 = 1 \end{cases}$$

Resuelva este sistema mediante eliminación gaussiana con pivoteo parcial escalado. Muestre todos los pasos intermedios y escriba el vector de índices en cada paso.

11. Considere eliminación gaussiana con pivoteo parcial escalado aplicada a la matriz de coeficientes

$$\left[\begin{array}{ccccc} \# & \# & \# & \# & 0 \\ \# & \# & \# & 0 & \# \\ 0 & \# & \# & \# & 0 \\ 0 & \# & 0 & \# & 0 \\ \# & 0 & 0 & \# & \# \end{array} \right]$$

donde cada # denota un elemento diferente de cero. Circule las ubicaciones de los elementos en los que se almacenarán los multiplicadores y marque con una f aquellos que se van a llenar. El vector de índices final es $\ell = [2, 3, 1, 5, 4]$.

12. Repita el problema 7.1.6a usando eliminación gaussiana con pivoteo parcial escalado.

13. Resuelva cada uno de los siguientes sistemas usando la eliminación gaussiana con pivoteo parcial escalado, con cuatro cifras significativas. ¿Cuáles son los contenidos del arreglo de índices en cada paso?

a. $\begin{cases} 3x_1 + 4x_2 + 3x_3 = 10 \\ x_1 + 5x_2 - x_3 = 7 \\ 6x_1 + 3x_3 + 7x_3 = 15 \end{cases}$

b. $\begin{cases} 3x_1 + 2x_2 - 5x_3 = 0 \\ 2x_1 - 3x_2 + x_3 = 0 \\ x_1 + 4x_2 - x_3 = 4 \end{cases}$

c. $\left[\begin{array}{cccc} 1 & -1 & 2 & 1 \\ 3 & 2 & 1 & 4 \\ 5 & 8 & 6 & 3 \\ 4 & 2 & 5 & 3 \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[\begin{array}{c} 1 \\ 1 \\ 1 \\ -1 \end{array} \right]$

d. $\begin{cases} 3x_1 + 2x_2 - x_3 = 7 \\ 5x_1 + 3x_2 + 2x_3 = 4 \\ -x_1 + x_2 - 3x_3 = -1 \end{cases}$

e. $\begin{cases} x_1 + 3x_2 + 2x_3 + x_4 = -2 \\ 4x_1 + 2x_2 + x_3 + 2x_4 = 2 \\ 2x_1 + x_2 + 2x_3 + 3x_4 = 1 \\ x_1 + 2x_2 + 4x_3 + x_4 = -1 \end{cases}$

14. Usando pivoteo parcial escalado, muestre cómo la computadora resuelve el siguiente sistema de ecuaciones. Muestre el arreglo de escala, diga cómo se seleccionaron los renglones *pivot* y realice los cálculos. Incluya el arreglo de índices para cada paso. No hay fracciones en la solución correcta, excepto ciertos cocientes que se deben considerar al seleccionar los pivotes. Debe seguir exactamente el código de pivoteo parcial escalado, excepto que puede incluir el lado derecho del sistema en sus cálculos conforme avance.

$$\begin{cases} 2x_1 - x_2 + 3x_3 + 7x_4 = 15 \\ 4x_1 + 4x_2 + 7x_4 = 11 \\ 2x_1 + x_2 + x_3 + 3x_4 = 7 \\ 6x_1 + 5x_2 + 4x_3 + 17x_4 = 31 \end{cases}$$

- 15.** Deduzca la fórmula

$$\sum_{k=1}^n k = \frac{n}{2}(n + 1)$$

Sugerencia: haga $S = \sum_{k=1}^n k$; también observe que

$$\begin{aligned} 2S &= (1 + 2 + \cdots + n) + [n + (n - 1) + \cdots + 2 + 1] \\ &= (n + 1) + (n + 1) + \cdots \end{aligned}$$

o use inducción.

- 16.** Deduzca la fórmula

$$\sum_{k=1}^n k^2 = \frac{n}{6}(n + 1)(2n + 1)$$

Sugerencia: la inducción es probablemente más fácil.

- 17.** Cuente el número de operaciones en el siguiente seudocódigo:

```
real array  $(a_{ij})_{1:n \times 1:n}, (x_{ij})_{1:n \times 1:n}$ 
real  $z$ ; integer  $i, j, n$ 
for  $i = 1$  to  $n$  do
    for  $j = 1$  to  $i$  do
         $z = z + a_{ij}x_{ij}$ 
    end for
end for
```

- 18.** Cuente el número de divisiones en el procedimiento *Gauss*. Cuente el número de multiplicaciones. Cuente el número de sumas o restas. Usando tiempos de ejecución en microsegundos (multiplicación 1, división 2.9, suma 0.4, resta 0.4), escriba una función de n que represente el tiempo usado en estas operaciones aritméticas.

- 19.** Considerando sólo operaciones largas y suponga que el tiempo de ejecución es de 1 microsegundo para todas las operaciones largas; luego dé los tiempos de ejecución y los costos aproximados para el procedimiento *Gauss* cuando $n = 10, 10^2, 10^3, 10^4$. Utilice únicamente el término dominante en el conteo de operación. Estime los costos a \$500 por hora.

- 20.** (Continuación) ¿Cuánto tiempo utilizaría la computadora para resolver 2000 ecuaciones usando eliminación gaussiana con pivoteo parcial escalado? ¿Cuánto costaría? Dé una estimación aproximada basada en los tiempos de operación.

- 21.** Después de procesar una matriz A usando el procedimiento *Gauss*, ¿cómo pueden los resultados utilizarse para resolver un sistema de ecuaciones de la forma $A^T\mathbf{x} = \mathbf{b}$?

- 22.** ¿Qué cambios harían más eficiente el procedimiento *Gauss* si la división fuera mucho más lenta que la multiplicación?

- 23.** La matriz $A = (a_{ij})_{n \times n}$ es de **renglones equilibrados** si se escala de modo que

$$\max_{1 \leq j \leq n} |a_{ij}| = 1 \quad (1 \leq i \leq n)$$

En la solución de un sistema de ecuaciones $A\mathbf{x} = \mathbf{b}$, podemos producir un sistema equivalente en el que la matriz es de renglones equilibrados dividiendo la i éSIMA ecuación entre $\max_{1 \leq j \leq n} |a_{ij}|$.

^a Resuelva el sistema de ecuaciones

$$\begin{bmatrix} 1 & 1 & 2 \times 10^9 \\ 2 & -1 & 10^9 \\ 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

con eliminación gaussiana con pivoteo parcial escalado.

- b.** Resuelva usando eliminación gaussiana simple de renglón equilibrado. ¿Las respuestas son iguales? ¿Por qué sí o por qué no?
- 24.** Resuelva cada sistema usando pivoteo parcial y pivoteo parcial escalado con cuatro dígitos significativos. También encuentre las soluciones verdaderas.

a. $\begin{cases} 0.004000x + 69.13y = 69.17 \\ 4.281x - 5.230y = 41.91 \end{cases}$

b. $\begin{cases} 40.00x + 691300y = 691700 \\ 4.281x - 5.230y = 41.91 \end{cases}$

c. $\begin{cases} 0.003000x + 59.14y = 59.17 \\ 5.291x - 6.130y = 46.78 \end{cases}$

d. $\begin{cases} 30.00x + 591400y = 591700 \\ 5.291x - 6.130y = 46.78 \end{cases}$

e. $\begin{cases} 0.7000x + 1725y = 1739 \\ 0.4352x - 5.433y = 5.278 \end{cases}$

f. $\begin{cases} 0.8000x + 1825y = 2040 \\ 0.4321x - 5.432y = 7.531 \end{cases}$

Problemas de cómputo 7.2

- 1.** Pruebe el ejemplo numérico del libro usando el algoritmo gaussiano simple y el algoritmo gaussiano con pivoteo parcial escalado.

- 2.** Considere el sistema

$$\begin{bmatrix} 0.4096 & 0.1234 & 0.3678 & 0.2943 \\ 0.2246 & 0.3872 & 0.4015 & 0.1129 \\ 0.3645 & 0.1920 & 0.3781 & 0.0643 \\ 0.1784 & 0.4002 & 0.2786 & 0.3927 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0.4043 \\ 0.1550 \\ 0.4240 \\ 0.2557 \end{bmatrix}$$

Resuévalo con eliminación gaussiana con pivoteo parcial escalado utilizando los procedimientos *Gauss* y *Solución*.

- 3.** (Continuación) Supongamos que se cometió un error cuando se escribió la matriz de coeficientes en el problema de cómputo 7.2.2 y que un sólo dígito no estaba bien escrito, es decir, 0.3645 se convirtió en 0.3345. Resuelva este sistema y observe el efecto de estos pequeños cambios. Explique.

- 4.** La **matriz de Hilbert** de orden n está definida por $a_{ij} = (i + j - 1)^{-1}$ para $1 \leq i, j \leq n$. A menudo se utiliza con fines de prueba debido a su naturaleza mal acondicionada. Defina $b_i = \sum_{j=1}^n a_{ij}$. Entonces, la solución del sistema de ecuaciones $\sum_{j=1}^n a_{ij}x_j = b_i$ para $1 \leq i \leq n$ es $\mathbf{x} = [1, 1, \dots, 1]^T$. Compruebe esto. Seleccione un valor de n en el rango de $2 \leq n \leq 15$, resuelva el sistema de ecuaciones para \mathbf{x} usando los procedimientos *Gauss* y *Solución*, y vea si el resultado es el previsto. Haga el caso $n = 2$ a mano para ver qué dificultades ocurren en la computadora.

- 5.** Defina el arreglo (a_{ij}) de $n \times n$ mediante $a_{ij} = -1 + 2 \max\{i, j\}$. Establezca el arreglo (b_i) de tal manera que la solución del sistema $A\mathbf{x} = \mathbf{b}$ sea $x_i = 1$ para $1 \leq i \leq n$. Pruebe los procedimientos *Gauss* y *Solución* en este sistema para un valor moderado de n ; por ejemplo, $n = 30$.

- 6.** Seleccione un valor modesto de n , digamos, $5 \leq n \leq 20$ y sea $a_{ij} = (i-1)^{j-1}$ y $b_{ij} = i-1$. Resuelva el sistema $\mathbf{Ax} = \mathbf{b}$ en la computadora. Al ver la salida, intuya cuál es la solución correcta. Establezca algebraicamente que su suposición es correcta. Cuente los errores en la solución calculada.

- 7.** Para un valor fijo de n de 2 a 4, sea

$$a_{ij} = (i + j)^2 \quad b_i = ni(i + n + 1) + \frac{1}{6}n(1 + n(2n + 3))$$

Muestre que el vector $\mathbf{x} = [1, 1, \dots, 1]^T$ resuelve el sistema $\mathbf{Ax} = \mathbf{b}$. Pruebe si los procedimientos *Gauss* y *Solución* pueden calcular correctamente \mathbf{x} para $n = 2, 3, 4$. Explique lo que sucede.

- 8.** Usando cada valor de n de 2 a 9, resuelva el sistema de $n \times n \mathbf{Ax} = \mathbf{b}$, donde \mathbf{A} y \mathbf{b} se definen por

$$a_{ij} = (i + j - 1)^7 \quad b_i = p(n + i - 1) - p(i - 1)$$

donde

$$p(x) = \frac{x^2}{24}(2 + x^2(-7 + n^2(14 + n(12 + 3n))))$$

Explique qué sucede.

- 9.** Resuelva el siguiente sistema usando los procedimientos *Gauss* y *Solución* y después utilice el procedimiento *Simple_de_Gauss*. Compare los resultados y explique.

$$\begin{bmatrix} 0.0001 & -5.0300 & 5.8090 & 7.8320 \\ 2.2660 & 1.9950 & 1.2120 & 8.0080 \\ 8.8500 & 5.6810 & 4.5520 & 1.3020 \\ 6.7750 & -2.2530 & 2.9080 & 3.9700 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 9.5740 \\ 7.2190 \\ 5.7300 \\ 6.2910 \end{bmatrix}$$

- 10.** Sin cambiar la lista de parámetros, reescriba y pruebe el procedimiento *Gauss* del modo que lo hace tanto con eliminación hacia adelante y sustitución hacia atrás. Aumente el tamaño de la matriz (a_{ij}) y guarde el arreglo del lado derecho (b_i) en la $(n+1)$ éSIMA columna de (a_{ij}) . También, regrese la solución de esta columna.

- 11.** Modifique los procedimientos *Gauss* y *Solución* de manera que sean más robustos. Dos cambios que se sugieren son los siguientes: (i) Sáltese la eliminación si $a_{\ell,i,k} = 0$ y (ii) agregue un parámetro de error *ierr* a la lista de parámetros y realice la comprobación de errores (por ejemplo, división entre cero o un renglón de ceros). Pruebe el código modificado en sistemas lineales de diferentes tamaños.

- 12.** Reescriba los procedimientos *Gauss* y *Solución* de modo que sean orientados a columnas, es decir, para que todos los ciclos internos varíen el primer índice de (a_{ij}) . En algunos sistemas de cómputo, esta aplicación puede evitar la paginación o el cambio entre alta velocidad y memoria secundaria y ser más eficiente para matrices de gran tamaño.

- 13.** La memoria de computadora se puede minimizar con el uso de un método de almacenamiento diferente cuando la matriz de coeficientes es simétrica. Una matriz simétrica de $n \times n$, $\mathbf{A} = (a_{ij})$ tiene la propiedad que $a_{ij} = a_{ji}$, de modo que sólo los elementos sobre y debajo de la diagonal principal se deben almacenar en un vector de longitud $n(n+1)/2$. Los elementos de la matriz \mathbf{A} se

colocan en un vector $\mathbf{v} = (v_k)$ en este orden: $a_{11}, a_{21}, a_{31}, a_{32}, a_{33}, \dots, a_{n,n}$. Almacenar una matriz de esta manera se conoce como **modo de almacenamiento simétrico** y efectúa un ahorro de $n(n-1)/2$ posiciones de memoria. Aquí, $a_{ij} = v_k$, donde $k = \frac{1}{2}(i-1) + j$ para $i \geq j$. Compruebe estos enunciados.

Escriba y pruebe los procedimientos *Simetría de Gauss*($n, (\mathbf{v}_i), (\ell_i)$) y *Solución_Simétrica*($n, (\mathbf{v}_i), (\ell_i), (b_i)$), que son análogos a los procedimientos *Gauss* y *Solución* excepto que la matriz de coeficientes se almacena en el modo de almacenamiento simétrico en un arreglo unidimensional (\mathbf{v}_i) y la solución se regresó al arreglo (b_i).

- 14.** El **determinante** de una matriz cuadrada puede calcularse fácilmente con la ayuda del procedimiento *Gauss*. Requerimos tres hechos sobre determinantes. Primero, el determinante de una matriz triangular es el producto de los elementos de su diagonal. Segundo, si un múltiplo de un renglón se añade a otro renglón, el determinante de la matriz no cambia. Tercero, si se intercambian dos renglones en una matriz, el signo del determinante cambia. El procedimiento *Gauss* se puede *interpretar* como un procedimiento para reducir una matriz a la forma triangular superior al intercambiar renglones y sumar múltiplos de un renglón a otro. Escriba una función $\det(n, (a_{ij}))$ que calcule el determinante de una matriz de $n \times n$. Que llame al procedimiento *Gauss* y utilice los arreglos (a_{ij}) y (ℓ_i) que resultan de esa llamada. Compruebe numéricamente la función \det utilizando las siguientes matrices de prueba con diferentes valores de n :

a. $a_{ij} = |i - j| \quad \det(\mathbf{A}) = (-1)^{n-1}(n-1)2^{n-2}$

b. $a_{ij} = \begin{cases} 1 & j \geq i \\ -j & j < i \end{cases} \quad \det(\mathbf{A}) = n!$

c. $\begin{cases} a_{ij} = a_{j1} = n^{-1} & j \geq 1 \\ a_{ij} = a_{i-1,j} + a_{i,j-1} & i, j \geq 2 \end{cases} \quad \det(\mathbf{A}) = n^{-n}$

- 15.** (Continuación) Sobreflujo y subflujo se pueden producir en la evaluación de los determinantes mediante este procedimiento. Para evitar esto, se puede calcular $\log |\det(\mathbf{A})|$ como la suma de términos $\log |a_{\ell_i,i}|$ y utilizar la función exponencial al final. Repita los experimentos numéricos del problema de cómputo 7.2.14 utilizando esta idea.
- 16.** Pruebe una modificación del procedimiento *Gauss* en la que el arreglo de escala se calcula nuevamente en cada paso (cada nuevo valor de k) de la fase de eliminación hacia adelante. Trate de construir un ejemplo para el cual en este procedimiento se produzcan menos errores de redondeo que en el método de pivoteo parcial escalado presentado en el libro con un arreglo escalado fijo. En general, se cree que los cálculos adicionales que están implicados en este procedimiento no valen la pena para la mayoría de los sistemas no lineales.
- 17.** (Continuación) Modifique y pruebe el procedimiento *Gauss* para que el sistema original esté inicialmente con renglones equilibrados, es decir, esté escalado para que el elemento máximo en cada renglón sea 1.
- 18.** Modifique y pruebe los procedimientos *Gauss* y *Solución* de manera que realicen un pivoteo *completo* escalado, es decir, el elemento pivote se selecciona de todos los elementos de la submatriz, no sólo los de la columna k éSIMA. Lleve registro del orden de las incógnitas en el arreglo solución en otro arreglo de índices, ya que no se determinarán en el orden x_n, x_{n-1}, \dots, x_1 .

19. Compare las soluciones calculadas numéricamente de los dos siguientes sistemas lineales:

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1.0 & 0.5 & 0.333333 & 0.25 & 0.2 \\ 0.5 & 0.333333 & 0.25 & 0.2 & 0.166667 \\ 0.333333 & 0.25 & 0.2 & 0.166667 & 0.142857 \\ 0.25 & 0.2 & 0.166667 & 0.142857 & 0.125 \\ 0.2 & 0.166667 & 0.142857 & 0.125 & 0.111111 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Resuelva ambos sistemas usando tanto eliminación gaussiana de precisión simple con pivoteo parcial escalado. Para cada sistema, calcule las normas $\ell_2 \|\mathbf{u}\|_2 = \sqrt{\sum_{i=1}^n u_i^2}$ del vector residual $\tilde{\mathbf{r}} = A\tilde{\mathbf{x}} - \mathbf{b}$ y del vector de error $\tilde{\mathbf{e}} = \tilde{\mathbf{x}} - \mathbf{x}$, donde $\tilde{\mathbf{x}}$ es la solución calculada y \mathbf{x} es la solución o verdadera exacta. Para el primer sistema, la solución exacta es $\mathbf{x} = [25, -300, 1050, -1400, 630]^T$ y para el segundo sistema, la solución exacta, con seis dígitos decimales de precisión, es $\mathbf{x} = [26.9314, -336.018, 1205.11, -1634.03, 744.411]^T$. No cambie los datos de entrada del segundo sistema para incluir más que el número de dígitos que se muestra. Analice los resultados. ¿Qué ha aprendido?

20. (Continuación) Repita el problema de cómputo anterior, pero establezca $a_{ij} \leftarrow 7560 a_{ij}$ y $b_i \leftarrow 7560 b_i$ para cada sistema antes de resolver.

21. Escriba versiones de aritmética compleja de los procedimientos *Gauss* y *Solución*, declarando ciertas variables complejas y haciendo otros cambios necesarios en el código. Pruebelos en los sistemas lineales complejos dados en el problema de cómputo 7.1.6.

22. (Continuación) Resuelva los sistemas lineales complejos que se presentan en el problema de cómputo 7.1.7.

23. El hecho de que en los dos problemas anteriores se le pidieran las soluciones de sistemas lineales complejos puede llevarlo a creer que *debe* tener versiones complejas de los procedimientos *Gauss* y *Solución*. No es así. Un sistema complejo $A\mathbf{x} = \mathbf{b}$ también se puede escribir como un sistema real de $2n \times 2n$:

$$\sum_{j=1}^n [\operatorname{Re}(a_{ij})\operatorname{Re}(x_j) - \operatorname{Im}(a_{ij})\operatorname{Im}(x_j)] = \operatorname{Re}(b_i) \quad (1 \leq i \leq n)$$

$$\sum_{j=1}^n [\operatorname{Re}(a_{ij})\operatorname{Im}(x_j) + \operatorname{Im}(a_{ij})\operatorname{Re}(x_j)] = \operatorname{Im}(b_i) \quad (1 \leq i \leq n)$$

Repita esos dos problemas usando esta idea y los dos procedimientos de esta sección. (Aquí, Re denota la parte real e Im la parte imaginaria.)

24. (Proyecto de investigación estudiantil) El algoritmo de Gauss-Huard es una variante del algoritmo de Gauss-Jordan para resolver sistemas lineales densos. Ambos algoritmos reducen el sistema a un sistema diagonal equivalente. Sin embargo, el método de Gauss-Jordan hace más operaciones de punto flotante que el de eliminación gaussiana, mientras que el mé-

todo de Gauss-Huard no. Para conservar estabilidad, el método de Gauss-Huard incorpora una estrategia de pivoteo usando intercambio de columnas. Un análisis de error indica que el método de Gauss-Huard es tan estable como el de eliminación de Gauss-Jordan con una estrategia adecuada de pivoteo. Lea acerca de estos algoritmos en los artículos de Dekker y Hoffmann [1989], Dekker, Hoffmann y Potma [1997], Hoffmann [1989] y Huard [1979]. Realice algunos experimentos numéricos programando y probando los algoritmos de Gauss-Jordan y de Gauss-Huard en algunos sistemas lineales densos.

25. Resuelva el sistema (5) utilizando las rutinas de software matemático basado en la eliminación gaussiana como las de Matlab, Maple o Mathematica. Hay un gran número de programas de computadora y paquetes de software para resolver sistemas lineales, cada uno de los cuales pueden utilizar una estrategia de pivoteo ligeramente diferente.

7.3 Sistemas tridiagonales y en banda

En muchas aplicaciones, incluidas algunas que se consideran más adelante, se encuentran sistemas lineales muy grandes que tienen una estructura **en banda**. Las matrices en banda con frecuencia se presentan en la solución de ecuaciones diferenciales ordinarias y parciales. Es ventajoso desarrollar un código de cómputo diseñado específicamente para dichos sistemas lineales, ya que reducen la cantidad de almacenamiento utilizado.

De importancia práctica es el sistema **tridiagonal**. Aquí, todos los elementos distintos de cero en la matriz de coeficientes deben estar en la diagonal principal o en las dos diagonales justo arriba y abajo de ella (normalmente llamadas **superdiagonal** y **subdiagonal**, respectivamente):

$$\left[\begin{array}{ccc|c} d_1 & c_1 & & b_1 \\ a_1 & d_2 & c_2 & b_2 \\ & a_2 & d_3 & c_3 \\ & \ddots & \ddots & \vdots \\ & & a_{i-1} & d_i & c_i \\ & & & \ddots & \ddots \\ & & & a_{n-2} & d_{n-1} & c_{n-1} \\ & & & a_{n-1} & d_n & \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_i \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \quad (1)$$

(No todos los elementos que se presentan en las diagonales son ceros.) Una matriz tridiagonal se caracteriza por la condición $a_{ij} = 0$ si $|i - j| \geq 2$. En general, se dice que una matriz tiene una **estructura en banda** si hay un número entero k (menor que n) tal que $a_{ij} = 0$ cuando $|i - j| \geq k$.

Los requisitos de almacenamiento para una matriz en banda son menores que los de una matriz general del mismo tamaño. Así, una matriz diagonal de $n \times n$ requiere sólo n ubicaciones de memoria en la computadora y una matriz tridiagonal sólo requiere $3n - 2$. Este hecho es importante si se están utilizando matrices en banda de orden muy grande.

Para matrices en banda, el algoritmo de eliminación gaussiana puede ser muy eficaz si se sabe de antemano que no es necesario el pivoteo. Esta situación se presenta con suficiente frecuencia para justificar procedimientos especiales. A continuación, desarrollamos un código para el sistema tridiagonal y damos una lista para el sistema *pentadiagonal* (en el que $a_{ij} = 0$ si $|i - j| \geq 3$).

Sistemas tridiagonales

La rutina que ahora se describe se llama procedimiento *Tri*. Está diseñado para resolver un sistema de n ecuaciones lineales con n incógnitas, como se muestra en la ecuación (1). Tanto en la fase de eliminación hacia delante como en la fase de sustitución hacia atrás se incorporan al procedimiento *y no* se utiliza pivoteo, es decir, las ecuaciones pivotadas son las dadas por el orden natural $\{1, 2, \dots, n\}$. Así, se utiliza la eliminación gaussiana simple.

En el paso 1, se resta a_1/d_1 veces el renglón 1 del renglón 2, creando un 0 en la posición a_1 . Sólo las entradas d_2 y b_2 están alteradas. Observe que c_2 no se altera. En el paso 2, se repite el proceso, utilizando el nuevo renglón 2 como renglón pivote. A continuación se presenta cómo se alteran los d_i y los b_i en cada paso:

$$\begin{cases} d_2 \leftarrow d_2 - \left(\frac{a_1}{d_1} \right) c_1 \\ b_2 \leftarrow b_2 - \left(\frac{a_1}{d_1} \right) b_1 \end{cases}$$

En general, se obtiene

$$\begin{cases} d_i \leftarrow d_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) c_{i-1} \\ b_i \leftarrow b_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) b_{i-1} \quad (2 \leq i \leq n) \end{cases}$$

Al final de la fase de eliminación hacia adelante la forma del sistema es el siguiente:

$$\left[\begin{array}{cccccc} d_1 & c_1 & & & & & \\ & d_2 & c_2 & & & & \\ & & d_3 & c_3 & & & \\ & & & \ddots & \ddots & & \\ & & & & d_i & c_i & \\ & & & & & \ddots & \ddots \\ & & & & & & d_{n-1} & c_{n-1} \\ & & & & & & & d_n \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_i \\ \vdots \\ b_{n-1} \\ b_n \end{array} \right]$$

Por supuesto, los b_i y los d_i no son lo que eran al comienzo de este proceso, pero los c_i sí. La fase de sustitución hacia atrás resuelve para x_n, x_{n-1}, \dots, x_1 de la siguiente manera:

$$\begin{aligned} x_n &\leftarrow \frac{b_n}{d_n} \\ x_{n-1} &\leftarrow \frac{1}{d_{n-1}} (b_{n-1} - c_{n-1}x_n) \end{aligned}$$

Finalmente, obtenemos

$$x_i \leftarrow \frac{1}{d_i} (b_i - c_i x_{i+1}) \quad (i = n-1, n-2, \dots, 1)$$

En el procedimiento *Tri* para un sistema tridiagonal, utilizamos sólo los arreglos unidimensionales (a_i), (d_i) y (c_i) para las diagonales en la matriz de coeficientes y en el arreglo (b_i) para el lado derecho; la solución se almacena en el arreglo (x_i).

```

procedure Tri(n,(ai),(di),(ci),(bi),(xi))
integer i,n; real xmult
real array (ai)1:n,(di)1:n,(ci)1:n,(bi)1:n,(xi)1:n
for i = 2 to n do
    xmult  $\leftarrow$  ai-1 / di-1
    di  $\leftarrow$  di - (xmult)ci-1
    bi  $\leftarrow$  bi - (xmult)bi-1
end for
 $x_n \leftarrow b_n / d_n$ 
for i = n - 1 to 1 step -1 do
    xi  $\leftarrow$  (bi - cixi+1) / di
end for
end procedure Tri

```

Observe que se han cambiado los datos originales en los arreglos (*d_i*) y (*b_i*).

Un sistema tridiagonal simétrico surge en el desarrollo del spline cúbico del capítulo 9 y en otros lugares. Un sistema tridiagonal general simétrico tiene la forma

$$\begin{bmatrix} d_1 & c_1 & & & & \\ c_1 & d_2 & c_2 & & & \\ & c_2 & d_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & c_{i-1} & d_i & c_i \\ & & & & \ddots & \ddots & \ddots \\ & & & & c_{n-2} & d_{n-1} & c_{n-1} \\ & & & & & c_{n-1} & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_i \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \quad (2)$$

También se podría sobrescribir el vector *b* del lado derecho con el vector solución *x*. Por lo tanto, un sistema lineal simétrico se puede resolver con una llamada al procedimiento de la forma

call *Tri*(*n*,(*c_i*),(*d_i*),(*c_i*),(*b_i*),(*b_i*))

que reduce el número de arreglos lineales de cinco a tres.

Dominio estrictamente diagonal

Puesto que el procedimiento *Tri* no implica pivoteo, es natural preguntarse si es probable que falle. Se pueden dar ejemplos simples para mostrar la falla debido a que se intenta hacer la división entre cero, aunque la matriz de coeficientes de la ecuación (1) es no singular. Por otra parte, no es fácil dar las condiciones más débiles posibles de esta matriz para garantizar el éxito del algoritmo. Nos contentamos con una propiedad que es fácil de comprobar y comúnmente encontrar. Si la matriz de coeficientes tridiagonal tiene dominio diagonal, entonces el procedimiento *Tri* no encontrará divisores cero.

DEFINICIÓN 1

Dominio estrictamente diagonal

Una matriz general $A = (a_{ij})_{n \times n}$ tiene **dominio estrictamente diagonal** si

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad (1 \leq i \leq n)$$

En el caso del sistema tridiagonal de la ecuación (1), el dominio estrictamente diagonal significa simplemente que (con $a_0 = a_n = 0$)

$$|d_i| > |a_{i-1}| + |c_i| \quad (1 \leq i \leq n)$$

Vamos a comprobar que la fase de eliminación hacia adelante en el procedimiento *Tri* conserva el dominio estrictamente diagonal. La nueva matriz de coeficientes obtenida de la eliminación gaussiana tiene elementos 0 donde originalmente estaban los a_i y los nuevos elementos de la diagonal se determinan recursivamente con

$$\begin{cases} \hat{d}_1 = d_1 \\ \hat{d}_i = d_i - \left(\frac{a_{i-1}}{\hat{d}_{i-1}} \right) c_{i-1} \quad (2 \leq i \leq n) \end{cases}$$

donde \hat{d}_i denota un nuevo elemento de la diagonal. Los elementos c_i no se alteran. Ahora se supone que $|d_i| > |a_{i-1}| + |c_i|$ y queremos asegurarnos que $|\hat{d}_i| > |c_i|$. Obviamente, esto es cierto para $i = 1$ porque $\hat{d}_1 = d_1$. Si es cierto para el índice $i - 1$ (es decir, $|\hat{d}_{i-1}| > |c_{i-1}|$), entonces es cierto para el índice i porque

$$\begin{aligned} |\hat{d}_i| &= \left| d_i - \left(\frac{a_{i-1}}{\hat{d}_{i-1}} \right) c_{i-1} \right| \\ &\geq |d_i| - |a_{i-1}| \frac{|c_{i-1}|}{|\hat{d}_{i-1}|} \\ &> |a_{i-1}| + |c_i| - |a_{i-1}| = |c_i| \end{aligned}$$

Aunque el número de operaciones largas en la eliminación gaussiana en matrices completas es $\mathcal{O}(n^3)$, sólo es $\mathcal{O}(n)$ para las matrices tridiagonales. Además, la estrategia de pivoteo escalado, no se necesita en sistemas tridiagonales con dominio estrictamente diagonal.

Sistemas pentadiagonales

Los principios mostrados con el procedimiento *Tri* se puede aplicar a matrices en banda más ancha con elementos distintos de cero. A continuación se presenta un procedimiento llamado *Penta* para

resolver el sistema de cinco diagonales:

$$\left[\begin{array}{cccccc} d_1 & c_1 & f_1 & & & \\ a_1 & d_2 & c_2 & f_2 & & \\ e_1 & a_2 & d_3 & c_3 & f_3 & \\ & e_2 & a_3 & d_4 & c_4 & f_4 \\ & \ddots & \ddots & \ddots & \ddots & \\ & & e_{i-2} & a_{i-1} & d_i & c_i & f_i \\ & & & \ddots & \ddots & \ddots & \ddots \\ & & & e_{n-4} & a_{n-3} & d_{n-2} & c_{n-2} & f_{n-2} \\ & & & e_{n-3} & a_{n-2} & d_{n-1} & c_{n-1} & \\ & & & & e_{n-2} & a_{n-1} & d_n & \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_i \\ \vdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_i \\ \vdots \\ b_{n-2} \\ b_{n-1} \\ b_n \end{bmatrix}$$

En el seudocódigo, se coloca el vector solución en el arreglo (x_i) . Además, no se debería utilizar esta rutina si $n \leq 4$. (¿Por qué?)

```

procedure Penta( $n, (e_i), (a_i), (d_i), (c_i), (f_i), (b_i), (x_i)$ )
integer  $i, n$ ; real  $r, s, xmult$ 
real array  $(e_i)_{1:n}, (a_i)_{1:n}, (d_i)_{1:n}, (c_i)_{1:n}, (f_i)_{1:n}, (b_i)_{1:n}, (x_i)_{1:n}$ 
 $r \leftarrow a_1$ 
 $s \leftarrow a_2$ 
 $t \leftarrow e_1$ 
for  $i = 2$  to  $n - 1$  do
     $xmult \leftarrow r / d_{i-1}$ 
     $d_i \leftarrow d_i - (xmult)c_{i-1}$ 
     $c_i \leftarrow c_i - (xmult)f_{i-1}$ 
     $b_i \leftarrow b_i - (xmult)b_{i-1}$ 
     $xmult \leftarrow t / d_{i-1}$ 
     $r \leftarrow s - (xmult)c_{i-1}$ 
     $d_{i+1} \leftarrow d_{i+1} - (xmult)f_{i-1}$ 
     $b_{i+1} \leftarrow b_{i+1} - (xmult)b_{i-1}$ 
     $s \leftarrow a_{i+1}$ 
     $t \leftarrow e_i$ 
end for
 $xmult \leftarrow r / d_{n-1}$ 
 $d_n \leftarrow d_n - (xmult)c_{n-1}$ 
 $x_n \leftarrow (b_n - (xmult)b_{n-1}) / d_n$ 
 $x_{n-1} \leftarrow (b_{n-1} - c_{n-1}x_n) / d_{n-1}$ 
for  $i = n - 2$  to  $1$  step  $-1$  do
     $x_i \leftarrow (b_i - f_i x_{i+2} - c_i x_{i+1}) / d_i$ 
end for
end procedure Penta

```

Para poder resolver sistemas pentadiagonales simétricos con el mismo código y con un mínimo de almacenamiento, hemos utilizado las variables r, s y t para almacenar temporalmente algunos datos en lugar de sobrescribir en los arreglos. Esto nos permite resolver un sistema pentadiagonal

simétrico con una llamada al procedimiento de la forma

```
call Penta(n,( fi),( ci),( di),( ci),( fi),( bi),( bi))
```

que reduce el número de arreglos lineales de siete a cuatro. Por supuesto, los datos originales en algunos de estos arreglos se corromperán. La solución calculada se almacenará en el arreglo (b_i) . En este caso, suponemos que todos los arreglos lineales llenan con ceros la longitud n para no exceder las dimensiones del arreglo en el seudocódigo.

Sistemas pentadiagonales de bloque

Muchos de los problemas matemáticos implican matrices con estructura de bloque. En muchos casos, hay ventajas en la explotación de la estructura del bloque en la solución numérica. Esto es particularmente cierto en la solución numérica de ecuaciones diferenciales parciales.

Podemos considerar un sistema pentadiagonal como un bloque de un sistema tridiagonal

$$\begin{bmatrix} \mathbf{D}_1 & \mathbf{C}_1 & & & \\ A_1 & \mathbf{D}_2 & \mathbf{C}_2 & & \\ & A_2 & \mathbf{D}_3 & \mathbf{C}_3 & \\ & & \ddots & \ddots & \ddots \\ & & & A_{i-1} & \mathbf{D}_i & \mathbf{C}_i \\ & & & & \ddots & \ddots & \ddots \\ & & & & A_{n-2} & \mathbf{D}_{n-1} & \mathbf{C}_{n-1} \\ & & & & A_{n-1} & \mathbf{D}_n & \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_i \\ \vdots \\ X_{n-1} \\ X_n \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \mathbf{B}_3 \\ \vdots \\ \mathbf{B}_i \\ \vdots \\ \mathbf{B}_{n-1} \\ \mathbf{B}_n \end{bmatrix}$$

donde

$$\mathbf{D}_i = \begin{bmatrix} d_{2i-1} & c_{2i-1} \\ a_{2i-1} & d_{2i} \end{bmatrix}, \quad A_i = \begin{bmatrix} e_{2i-1} & c_{2i-1} \\ 0 & e_{2i} \end{bmatrix}, \quad \mathbf{C}_i = \begin{bmatrix} f_{2i-1} & 0 \\ c_{2i-1} & f_{2i} \end{bmatrix}$$

En este caso, suponemos que n es par, digamos $n = 2m$. Si n no es par, entonces el sistema se puede completar con una ecuación extra $x_{n+1} = 1$ para que el número de renglones sea par.

El algoritmo para este sistema de bloque tridiagonal es similar al de los sistemas tridiagonales. Por lo tanto, tenemos la fase de eliminación hacia adelante

$$\begin{cases} \mathbf{D}_i \leftarrow \mathbf{D}_i - A_{i-1} \mathbf{D}_{i-1}^{-1} \mathbf{C}_{i-1} \\ \mathbf{B}_i \leftarrow \mathbf{B}_i - A_{i-1} \mathbf{D}_{i-1}^{-1} \mathbf{B}_{i-1} \end{cases} \quad (2 \leq i \leq m)$$

y la fase de sustitución hacia atrás

$$\begin{cases} X_n \leftarrow \mathbf{D}_n^{-1} \mathbf{B}_n \\ X_i \leftarrow \mathbf{D}_i^{-1} (\mathbf{B}_i - \mathbf{C}_i X_{i+1}) \end{cases} \quad (m-1 \leq i \leq 1)$$

Aquí,

$$\mathbf{D}_i^{-1} = \frac{1}{\Delta} \begin{bmatrix} d_{2i} & -c_{2i-1} \\ -a_{2i-1} & d_{2i-1} \end{bmatrix}$$

donde $\Delta = d_{2i}d_{2i-1} - a_{2i-1}c_{2i-1}$.

El código para la solución de un sistema pentadiagonal que utiliza este procedimiento de bloque se deja como ejercicio (problema de cómputo 7.3.21). Los resultados del código de bloque pentadiagonal son los mismos que con el procedimiento *Penta*, con excepción del error de redon-

deo. También, este procedimiento puede emplearse para sistemas pentadiagonales simétricos (en los que las subdiagonales son las mismas que las superdiagonales).

En el capítulo 16 se analizan ecuaciones diferenciales parciales bidimensiones elípticas. Por ejemplo, la ecuación de Laplace se define sobre el cuadrado unitario. Una red de puntos de 3×3 se coloca sobre la región cuadrada unitaria, y los puntos se acomodan siguiendo el orden natural (de izquierda a derecha y hacia arriba) como se muestra en la figura 7.2.

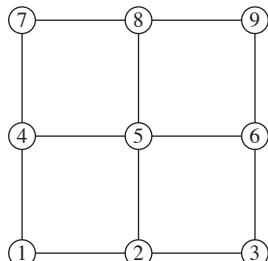


FIGURA 7.2
Red de puntos
con un orden
natural

En la ecuación de Laplace, las segundas derivadas parciales se aproximan con fórmulas de diferencia finita centradas de segundo orden. Esto da como resultado un sistema de ecuaciones lineales de 9×9 que tiene una matriz de coeficientes dispersa con este patrón distinto de cero:

$$\mathbf{A} = \left[\begin{array}{ccc|ccc|ccc} \times & \times & & \times & & & & & \\ \times & \times & \times & & \times & & & & \\ & \times & \times & & & \times & & & \\ \hline & & & \times & \times & \times & & & \\ & & & \times & \times & \times & & & \\ & & & & \times & \times & & & \\ \hline & & & & & & \times & \times & \\ & & & & & & \times & \times & \times \\ & & & & & & & \times & \times \end{array} \right]$$

Aquí, las entradas distintas de cero en la matriz se indicarán mediante el símbolo \times y los elementos nulos están en blanco. Esta matriz es en bloque tridiagonal y cada bloque distinto de cero es tridiagonal o diagonal. Otros ordenamientos del resultado de los puntos de la red dan como resultado matrices dispersas con diferentes patrones.

Resumen

(1) Para sistemas en banda, como el tridiagonal, el pentadiagonal y otros, es habitual desarrollar algoritmos especiales para que ejecuten la eliminación gaussiana, ya que *no* se necesita el pivoteo parcial en muchas aplicaciones. El procedimiento de eliminación hacia adelante para un sistema lineal tridiagonal $A = \text{tridiagonal}[(a_i), (d_i), (c_i)]$ es

$$\begin{cases} d_i \leftarrow d_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) c_{i-1} \\ b_i \leftarrow b_i - \left(\frac{a_{i-1}}{d_{i-1}} \right) b_{i-1} \quad (2 \leq i \leq n) \end{cases}$$

El procedimiento de sustitución hacia atrás es

$$x_i \leftarrow \frac{1}{d_i}(b_i - c_i x_{i+1}) \quad (i = n-1, n-2, \dots, 1)$$

(2) Una matriz con **dominio estrictamente diagonal** $A = (a_{ij})_{n \times n}$ es aquella en la que la magnitud de la entrada en diagonal es más grande que la suma de las magnitudes de las entradas fuera de la diagonal en el mismo renglón y esto es cierto para todos los renglones, a saber,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad (1 \leq i \leq n)$$

Para matrices de coeficientes tridiagonales con dominio estrictamente diagonal el pivoteo parcial *no* es necesario porque *no* se encontrarán divisores cero.

(3) Los procedimientos de eliminación y sustitución hacia atrás para un sistema lineal pentadiagonal $A =$ pentadiagonal $[(e_i), (a_i), (d_i), (c_i), (f_i)]$ son similares a los de un sistema tridiagonal.

Referencias adicionales

Para un estudio adicional de los sistemas lineales, véase Coleran y Van Loan [1988], Dekker y Hoffmann [1989] Dekker, Hoffmann y Potma [1997], Dongarra, Duff, Sorenson y Van der Vorst [1990], Forsythe y Moler [1967], Gallivan y colaboradores [1990], Golub y Van Loan [1996], Hoffmann [1989] Jennings [1977], Meyer [2000], Noble y Daniel [1988], Stewart [1973, 1996, 1998a, 1998b, 2001] y Watkins [1991].

Problemas 7.3

1. ¿Qué sucede con el sistema tridiagonal (1) si se utiliza la eliminación gaussiana con pivoteo parcial para resolver el problema? En general, ¿qué sucede en un sistema en banda?
2. Cuente las operaciones aritméticas largas que intervienen en los procedimientos:
 a. *Tri* b. *Penta*
3. ¿Cuántos lugares de almacenamiento se necesitan para un sistema de n ecuaciones lineales, si la matriz de coeficientes tiene estructura en banda tal que $a_{ij} = 0$ para $|i - j| \geq k + 1$?
4. Dé un ejemplo de un sistema de ecuaciones lineales en forma tridiagonal que no se puede resolver sin pivoteo.
5. Cuál es el aspecto de una matriz A si sus elementos satisfacen $a_{ij} = 0$ cuando:
 a. $j < i - 2$ b. $j > i + 1$
6. Considere una matriz con dominio estrictamente diagonal cuyos elementos satisfacen $a_{ij} = 0$ cuando $i > j + 1$. ¿La eliminación gaussiana sin pivoteo mantiene el dominio estrictamente diagonal? ¿Por qué sí o por qué no?
7. Sea A una matriz de la forma (1) tal que $a_i c_i > 0$ para $1 \leq i \leq n-1$. Encuentre la forma general de la matriz diagonal $D = \text{diag}(a_i)$ con $a_i \neq 0$ tal que $D^{-1}AD$ sea simétrica. ¿Cuál es la forma general $D^{-1}AD$?

Problemas de cómputo 7.3

1. Reescriba el procedimiento *Tri* usando solamente cuatro arreglos (a_i), (d_i), (c_i) y (b_i), y almacene la solución en el arreglo (b_i). Pruebe el código con dos sistemas tridiagonales, uno simétrico y otro no simétrico.
2. Repita el problema de cómputo anterior con el procedimiento *Penta* con seis arreglos (e_i) (a_i), (d_i), (c_i), (f_i) y (b_i). Utilice el ejemplo con que inicia este capítulo como uno de los casos de prueba.
3. Escriba y pruebe un procedimiento especial para resolver un sistema en el que $a_i = c_i = 1$ para toda i .
4. Utilice el procedimiento *Tri* para resolver el siguiente sistema de 100 ecuaciones. Compare la solución numérica con la solución exacta obvia.

$$\begin{cases} x_1 + 0.5x_2 &= 1.5 \\ 0.5x_{i-1} + x_i + 0.5x_{i+1} &= 2.0 \quad (2 \leq i \leq 99) \\ 0.5x_{99} + x_{100} &= 1.5 \end{cases}$$

5. Resuelva el sistema

$$\begin{cases} 4x_1 - x_2 &= -20 \\ x_{j-1} - 4x_j + x_{j+1} &= 40 \quad (2 \leq j \leq n-1) \\ -x_{n-1} + 4x_n &= -20 \end{cases}$$

usando el procedimiento *Tri* con $n = 100$.

6. Sea A una matriz tridiagonal de 50×50

$$\begin{bmatrix} 5 & -1 & & & & \\ -1 & 5 & -1 & & & \\ & -1 & 5 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 5 & -1 \\ & & & & -1 & 5 \end{bmatrix}$$

Considere el problema $Ax = b$ para 50 vectores diferentes b de la forma

$$[1, 2, \dots, 49, 50]^T \quad [2, 3, \dots, 50, 1]^T \quad [3, 4, \dots, 50, 1, 2]^T \quad \dots$$

Escriba y pruebe un código eficiente para resolver este problema. *Sugerencia:* reescriba el procedimiento *Tri*.

7. Reescriba y pruebe el procedimiento *Tri* para que realice eliminación gaussiana con pivoteo parcial escalado. *Sugerencia:* puede necesitar otras matrices de almacenamiento temporal.
8. Reescriba y pruebe *Penta* para que haga eliminación gaussiana con pivoteo parcial escalado. ¿Vale la pena hacerlo?
9. Usando las ideas ilustradas en *Penta*, escriba un procedimiento para resolver sistemas de siete diagonales. Pruebe con diferentes sistemas de ese tamaño.

- 10.** Considere el sistema de ecuaciones ($n = 7$)

$$\begin{bmatrix} d_1 & & & & a_6 & a_7 \\ & d_2 & & & & \\ & & d_3 & a_5 & & \\ & & & d_4 & & \\ & & a_3 & & d_5 & \\ a_1 & & a_2 & & & d_6 \\ & & & & & d_7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}$$

Para n impar, escriba y pruebe

procedure $X_Gauss(n, (a_i), (d_i), (b_i))$

que hace la fase de eliminación hacia adelante de la eliminación gaussiana (sin pivoteo parcial escalado) y

procedure $X_Solve(n, (a_i), (d_i), (b_i), (x_i))$

que hace la sustitución hacia atrás para sistemas cruzados de esta forma.

- 11.** Considere el sistema triangular inferior de $n \times n Ax = b$, donde $A = (a_{ij})$ y $a_{ij} = 0$ para $i < j$.

- a.** Escriba un algoritmo (en términos matemáticos) para resolver x con sustitución hacia adelante.
- b.** Escriba

producere $Sutitución_hacia_adelante(n, (a_i), (b_i), (x_i))$

que use este algoritmo.

- c.** Determine el número de divisiones, multiplicaciones y sumas (o restas) en el uso de este algoritmo para resolver para x .
- d.** ¿Puede usarse la eliminación gaussiana con pivoteo parcial para resolver tal sistema?

- 12. (Algoritmo tridiagonal normalizado)** Construya un algoritmo para el manejo de sistemas tridiagonales en el que se utilice el procedimiento de eliminación gaussiana normalizada sin pivoteo. En este proceso, cada renglón pivote se divide entre el elemento de la diagonal antes de que un múltiplo del renglón se reste de los renglones sucesivos. Escriba las ecuaciones implicadas en la fase de eliminación hacia adelante y almacene las entradas de la parte superior de la diagonal del arreglo (c_j) y las entradas en el arreglo del lado derecho (b_j). Escriba las ecuaciones para la fase de sustitución hacia atrás, almacenando la matriz de la solución en el arreglo (b_j). Codifique y pruebe este procedimiento. ¿Cuáles son sus ventajas y desventajas?

- 13.** Para un sistema tridiagonal de $(2n) \times (2n)$, escriba y pruebe un procedimiento que proceda de la siguiente manera. En la fase de eliminación hacia delante, la rutina al mismo tiempo elimina los elementos en la subdiagonal de la parte superior a la media y en la superdiagonal desde la parte interior a la media. En la fase de sustitución hacia atrás, las dos incógnitas se determinan a la vez del centro hacia fuera.

- 14.** (Continuación) Reescriba y pruebe el procedimiento en el problema de cómputo anterior para una matriz tridiagonal general de $n \times n$.

15. Suponga que

procedure *Tri_Normal*(*n*,(*a_i*),(*d_i*),(*c_i*),(*b_i*),(*x_i*))

realiza el algoritmo de eliminación gaussiana normalizada del problema de cómputo 7.3.12 y que

procedure *Tri_2n*(*n*,(*a_i*),(*d_i*),(*c_i*),(*b_i*),(*x_i*))

realiza el algoritmo mencionado en el problema de cómputo 7.3.13. Utilizando una rutina cronometrada en su computadora, compare *Tri*, *Tri_Normal*, y *Tri_2n* para determinar cuál de ellos es más rápido para el sistema tridiagonal

$$\begin{aligned} a_i &= i(n - i + 1), & c_i &= (i + 1)(n - i - 1), \\ d_i &= (2i + 1)n - i - 2i, & b_i &= i \end{aligned}$$

con un valor grande y par de *n*. *Nota:* los algoritmos matemáticos pueden comportarse de manera diferente en computadoras en paralelo y en computadoras de vectores. En general, los cálculos en paralelo alteran completamente nuestras nociones convencionales acerca de qué es lo mejor o lo más eficiente.

16. Considere un sistema especial bidiagonal lineal de la siguiente forma (ilustrado con *n* = 7) con elementos diferentes de cero en la diagonal:

$$\left[\begin{array}{ccccccc} d_1 & & & & & & \\ a_1 & d_2 & & & & & \\ & a_2 & d_3 & & & & \\ & & a_3 & d_4 & a_4 & & \\ & & & d_5 & a_5 & & \\ & & & & d_6 & a_6 & \\ & & & & & d_7 & \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{array} \right]$$

Escriba y pruebe

producere *Bi_Diagonal*(*n*,(*a_i*),(*d_i*),(*b_i*))

para resolver el sistema general de orden *n* (impar). Guarde la solución en el arreglo *b* y suponga que todos los arreglos son de longitud *n*. No utilice eliminación hacia adelante porque el sistema puede ser resuelto fácilmente sin él.

17. Escriba y pruebe

producere *Tridiagonal_hacia_atrás*(*n*,(*a_i*),(*d_i*),(*c_i*),(*b_i*),(*x_i*))

para la solución de un sistema tridiagonal hacia atrás de ecuaciones lineales de la forma

$$\left[\begin{array}{ccccc} & a_1 & d_1 & & \\ & a_2 & d_2 & c_1 & \\ a_3 & d_3 & c_2 & & \\ \vdots & \vdots & \ddots & & \\ a_{n-1} & d_{n-1} & c_{n-1} & & \\ d_n & c_{n-1} & & & \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{array} \right] = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{array} \right]$$

usando eliminación gaussiana sin pivoteo.

18. Una matriz superior de Hessenberg es de la forma

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{32} & a_{33} & \cdots & a_{3n} & \\ \ddots & \ddots & \ddots & & \\ a_{n,n-1} & & a_{nn} & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

Escriba un procedimiento para la solución de este sistema y pruébelo con un sistema que tenga 10 o más ecuaciones.

19. Una matriz de coeficientes de $n \times n$ en bandas con ℓ subdiagonales y m superdiagonales se puede almacenar con el **modo de almacenamiento en banda** en un arreglo $n \times (\ell + m + 1)$. La matriz se guarda con la estructura de la diagonal y los renglones intacta y con casi todos los elementos cero sin guardar. Si la matriz de $n \times n$ en banda original tenía la forma que se muestra en la figura, entonces el arreglo $n \times (\ell + m + 1)$ en el modo de almacenamiento en banda sería como se muestra. La diagonal principal sería la $\ell + 1$ éSIMA columna del nuevo arreglo. Escriba y pruebe el procedimiento para resolver un sistema lineal con la matriz de coeficientes almacenada en el modo de almacenamiento en banda.
20. Una matriz de coeficientes de $n \times n$ en banda con m subdiagonales y m superdiagonales se puede almacenar en el **modo de almacenamiento en banda simétrico** en un arreglo de $n \times (m + 1)$. Sólo la diagonal principal y las subdiagonales se almacenan de modo que la diagonal principal es la última columna en la nueva matriz, como se muestra en la figura. Escriba y pruebe el procedimiento para resolver un sistema lineal con la matriz de coeficientes almacenados en el modo de almacenamiento en banda simétrico.
21. Escriba un código para la solución de sistemas de bloque pentadiagonales y pruébelo en sistemas con submatrices en bloque. Compárelo con *Penta* empleando sistemas simétricos y no simétricos.
22. (**Filtro de spline no periódico**) La ecuación de filtro para el filtro de spline no periódico está dada por el sistema de $n \times n$

$$(I + \alpha^4 Q)w = z$$

donde la matriz es

$$Q = \begin{bmatrix} 1 & -2 & 1 & & & \\ -2 & 5 & -4 & 1 & & \\ 1 & -4 & 6 & -4 & 1 & \\ \ddots & \ddots & \ddots & \ddots & \ddots & \\ & 1 & -4 & 6 & -4 & -1 \\ & & 1 & -4 & 5 & -2 \\ & & & 1 & -2 & 1 \end{bmatrix}$$

Aquí el parámetro $\alpha = l/[2 \operatorname{sen}(\pi \Delta x / \lambda_c)]$ implica los valores de medición del perfil, dimensiones y longitud de onda en un intervalo de muestreo. La solución w da el valor para el perfil de los componentes de onda larga y $z - w$ son aquellos de los componentes de onda corta. Utilice este sistema para probar el código *Penta* utilizando valores distintos de α . *Sugerencia:* para los sistemas de prueba, seleccione un vector solución simple como $w = [1, -1, 1, -1, \dots, 1]^T$ con un valor modesto de n y luego calcule el lado derecho de la multiplicación matricial de vectores $z = (I + \alpha^4 Q)w$.

- 23.** (Continuación, filtro de spline periódico) La ecuación de filtro para el filtro de spline periódico está dada por el sistema de $n \times n$

$$(I + \alpha^4 \hat{Q})\hat{w} = \hat{z}$$

donde la matriz es

$$\hat{Q} = \begin{bmatrix} 6 & -4 & 1 & & & 1 & -4 \\ -4 & 6 & -4 & 1 & & & 1 \\ 1 & -4 & 6 & -4 & 1 & & \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ 1 & & & 1 & -4 & 6 & -4 \\ -4 & 1 & & & 1 & -4 & 6 \end{bmatrix}$$

Los filtros de spline periódico se utilizan en los casos de perfiles cerrados de filtrado. Haciendo uso de la simetría, modifique el seudocódigo *Penta* para manejar este sistema y después codifique y pruébelo.

- 24.** Use software matemático como Matlab, Maple o Mathematica para generar un sistema tridiagonal y resuélvalo. Por ejemplo, utilice el sistema tridiagonal de $5 \times 5 A = \text{Banda_Matriz}(-1, 2, 1)$ con el lado derecho $b = [1, 4, 9, 16, 25]^T$.

Temas adicionales referentes a sistemas de ecuaciones lineales

En aplicaciones que implican ecuaciones diferenciales parciales surgen grandes sistemas lineales con matrices de coeficientes dispersas como

$$\mathbf{A} = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{bmatrix}$$

La eliminación gaussiana puede causar que se llenen los elementos nulos con valores distintos de cero. Por otro lado, los métodos iterativos conservan su estructura dispersa.

8.1 Factorizaciones matriciales

Un sistema de ecuaciones lineales de $n \times n$ se puede escribir en forma de matriz

$$\mathbf{Ax} = \mathbf{b} \quad (1)$$

donde la matriz \mathbf{A} de coeficientes tiene la forma

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}$$

Nuestro principal objetivo es mostrar que el algoritmo gaussiano simple aplicado a A produce una factorización de A como el simple producto de dos matrices, una *triangular inferior* unitaria.

$$\mathbf{L} = \begin{bmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \vdots & \ddots & \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \cdots & 1 \end{bmatrix}$$

y la otra *triangular superior*.

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{bmatrix}$$

En pocas palabras, nos referimos a esto como una **factorización LU** de A , es decir, $A = LU$.

Ejemplo numérico

El sistema de ecuaciones (2) de la sección 7.1 se puede escribir de manera concisa en forma de matriz:

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ 26 \\ -19 \\ -34 \end{bmatrix} \quad (2)$$

Además, las operaciones que conducen de este sistema a la ecuación (5) de la sección 7.1, es decir, al sistema

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -3 \end{bmatrix} \quad (3)$$

podrían efectuarse por una multiplicación de matrices apropiada. La fase de eliminación hacia adelante se puede interpretar iniciando en (1) y procediendo a

$$\mathbf{M}\mathbf{A}\mathbf{x} = \mathbf{M}\mathbf{b} \quad (4)$$

donde \mathbf{M} es una matriz elegida de modo que $\mathbf{M}\mathbf{A}$ es la matriz de coeficientes para el sistema (3). Por lo tanto, tenemos

$$\mathbf{M}\mathbf{A} = \begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \equiv \mathbf{U}$$

que es una matriz triangular superior.

El primer paso de la eliminación gaussiana simple da como resultado la ecuación (3) de la sección 7.1 o el sistema

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -27 \\ -18 \end{bmatrix}$$

Este paso se puede lograr mediante la multiplicación de (1) por una matriz triangular inferior \mathbf{M}_1 :

$$\mathbf{M}_1 \mathbf{A} \mathbf{x} = \mathbf{M}_1 \mathbf{b}$$

donde

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Observe la forma especial de \mathbf{M}_1 . Los elementos de la diagonal son todos 1 y los únicos elementos distintos de cero están en la primera columna. Estos números son los *negativos de los multiplicadores* situados en las posiciones donde se han creado ceros como coeficientes en el paso 1 de la fase de eliminación hacia adelante. Continuando, el paso 2 da como resultado la ecuación (4) de la sección 7.1 o el sistema

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ -9 \\ -21 \end{bmatrix}$$

que es equivalente a

$$\mathbf{M}_2 \mathbf{M}_1 \mathbf{A} \mathbf{x} = \mathbf{M}_2 \mathbf{M}_1 \mathbf{b}$$

donde

$$\mathbf{M}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{bmatrix}$$

Una vez más, \mathbf{M}_2 difiere de una matriz identidad por la presencia de los negativos de los multiplicadores en la segunda columna de la diagonal hacia abajo. Por último, el paso 3 resulta en el sistema (3), lo que equivale a

$$\mathbf{M}_3 \mathbf{M}_2 \mathbf{M}_1 \mathbf{A} \mathbf{x} = \mathbf{M}_3 \mathbf{M}_2 \mathbf{M}_1 \mathbf{b}$$

donde

$$\mathbf{M}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

Ahora, la fase de eliminación hacia adelante está completa y con

$$\mathbf{M} = \mathbf{M}_3 \mathbf{M}_2 \mathbf{M}_1 \quad (5)$$

tenemos el sistema de coeficientes triangular superior (3).

Utilizando las ecuaciones (4) y (5) podemos dar una interpretación diferente de la fase de eliminación hacia adelante de la eliminación gaussiana simple. Ahora vemos que

$$\begin{aligned} \mathbf{A} &= \mathbf{M}^{-1} \mathbf{U} \\ &= \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \mathbf{M}_3^{-1} \mathbf{U} \\ &= \mathbf{L} \mathbf{U} \end{aligned}$$

Puesto que cada \mathbf{M}_k tiene una forma especial, su inversa se obtiene simplemente ¡cambiando los signos de las entradas de multiplicador negativo! Por lo tanto, tenemos que

$$\begin{aligned} \mathbf{L} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{bmatrix} \end{aligned}$$

Es un tanto sorprendente que \mathbf{L} sea una matriz triangular inferior unitaria compuesta por los multiplicadores. Observe que en la formación de \mathbf{L} , no determinamos a \mathbf{M} primero y después calculamos $\mathbf{M}^{-1} = \mathbf{L}$. (¿Por qué?)

Es fácil comprobar que

$$\begin{aligned} \mathbf{L} \mathbf{U} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \\ &= \begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} = \mathbf{A} \end{aligned}$$

Vemos que \mathbf{A} está **factorizada o descompuesta** en una matriz triangular inferior unitaria \mathbf{L} y una matriz triangular superior \mathbf{U} . La matriz \mathbf{L} se compone de multiplicadores situados en las posiciones de los elementos que aniquilaron de \mathbf{A} , de los elementos unidad de la diagonal y de elementos 0 triangulares superiores. De hecho, ahora sabemos la forma general de \mathbf{L} y sólo se puede escribir directamente a través de los multiplicadores *sin* formar las \mathbf{M}_k y las \mathbf{M}_k^{-1} . La matriz \mathbf{U} es triangular superior (generalmente no tiene una diagonal unitaria) y es la matriz de coeficientes final después de que se ha completado la fase de eliminación hacia adelante.

Cabe señalar que en el seudocódigo *Gauss_Simple* de la sección 7.1 se sustituye la matriz de coeficientes original con su factorización \mathbf{LU} . Los elementos de \mathbf{U} están en la parte triangular superior del arreglo (a_{ij}) incluida la diagonal. Las entradas debajo de la diagonal principal en \mathbf{L} (es decir, los multiplicadores) se encuentran debajo de la diagonal principal en el arreglo (a_{ij}) . Puesto que se sabe que \mathbf{L} tiene una diagonal de unos, nada se pierde al no almacenarlos. [En efecto, ¡de todas formas nos quedamos sin espacio en el arreglo (a_{ij}) !]]

Deducción formal

Para ver formalmente cómo la eliminación gaussiana (en forma simple) conduce a una factorización \mathbf{LU} es necesario demostrar que cada operación de renglón usada en el algoritmo se puede

efectuar al multiplicar \mathbf{A} por la izquierda por una matriz elemental. En concreto, si queremos restar λ veces el renglón p del renglón q , primero aplicamos esta operación a la matriz identidad de $n \times n$ para crear una matriz elemental \mathbf{M}_{qp} . Entonces se forma la matriz producto $\mathbf{M}_{qp}\mathbf{A}$.

Antes de continuar, vamos a verificar que $\mathbf{M}_{qp}\mathbf{A}$ se obtiene de restar λ veces el renglón p del renglón q de la matriz \mathbf{A} . Supongamos que $p < q$ (en el algoritmo simple, esto no siempre es cierto). Entonces, los elementos de $\mathbf{M}_{qp} = (m_{ij})$ son

$$m_{ij} = \begin{cases} 1 & \text{si } i = j \\ -\lambda & \text{si } i = q \text{ y } j = p \\ 0 & \text{en todos los demás casos} \end{cases}$$

Por lo tanto, los elementos de $\mathbf{M}_{qp}\mathbf{A}$ están dados por

$$(\mathbf{M}_{qp}\mathbf{A})_{ij} = \sum_{s=1}^n m_{is}a_{sj} = \begin{cases} a_{ij} & \text{si } i \neq q \\ a_{qj} - \lambda a_{pj} & \text{si } i = q \end{cases}$$

El q -ésimo renglón de $\mathbf{M}_{qp}\mathbf{A}$ es la suma del renglón q -ésimo de \mathbf{A} y $-\lambda$ veces el renglón del p -ésimo de \mathbf{A} , como se quería demostrar.

El k -ésimo paso de la eliminación gaussiana corresponde a la matriz \mathbf{M}_k que es el producto de $n - k$ matrices elementales:

$$\mathbf{M}_k = \mathbf{M}_{nk} \mathbf{M}_{n-1,k} \cdots \mathbf{M}_{k+1,k}$$

Observe que cada matriz elemental \mathbf{M}_{ik} aquí es triangular inferior porque $i > k$ y, por lo tanto, k es también inferior triangular. Si llevamos a cabo el proceso de eliminación gaussiana hacia adelante en \mathbf{A} , el resultado será una matriz triangular superior \mathbf{U} . Por otra parte, el resultado se obtiene aplicando una serie de factores tales como \mathbf{M}_k a la izquierda de \mathbf{A} . Por lo tanto, todo el proceso se resume a escribir

$$\mathbf{M}_{n-1} \cdots \mathbf{M}_2 \mathbf{M}_1 \mathbf{A} = \mathbf{U}$$

Puesto que cada \mathbf{M}_k es invertible, tenemos

$$\mathbf{A} = \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \cdots \mathbf{M}_{n-1}^{-1} \mathbf{U}$$

Cada \mathbf{M}_k es triangular inferior con unos en su diagonal principal (triangular inferior unitaria). Cada inversa \mathbf{M}_k^{-1} tiene la misma propiedad y lo mismo es cierto de su producto. Por lo tanto, la matriz

$$\mathbf{L} = \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \cdots \mathbf{M}_{n-1}^{-1} \quad (6)$$

es triangular inferior unitaria y tenemos

$$\mathbf{A} = \mathbf{LU}$$

Esta es la llamada **factorización LU** de \mathbf{A} . Nuestra construcción de la misma depende de *no* encontrar ningún divisor 0 en el algoritmo. Es fácil dar ejemplos de matrices que no tienen factorización LU ; una de las más simples es

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

(Véase el problema 8.1.4.)

TEOREMA 1**Teorema factorización LU**

Sea $\mathbf{A} = (a_{ij})$ una matriz de $n \times n$. Supongamos que la fase de eliminación hacia adelante del algoritmo gaussiano simple se aplica a \mathbf{A} sin encontrar divisores 0. Sea que la matriz resultante se denota por $\tilde{\mathbf{A}} = (\tilde{a}_{ij})$. Si

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \tilde{a}_{21} & 1 & 0 & \cdots & 0 \\ \tilde{a}_{31} & \tilde{a}_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{n,n-1} & 1 \end{bmatrix}$$

y

$$\mathbf{U} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \cdots & \tilde{a}_{1n} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ 0 & 0 & \tilde{a}_{33} & \cdots & \tilde{a}_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \tilde{a}_{nn} \end{bmatrix}$$

entonces $\mathbf{A} = \mathbf{LU}$.

Demostración Definimos el algoritmo gaussiano formalmente como sigue. Sea $\mathbf{A}^{(1)} = \mathbf{A}$. Entonces calculamos $\mathbf{A}^{(2)}$, $\mathbf{A}^{(3)}, \dots, \mathbf{A}^{(n)}$ recursivamente con el algoritmo gaussiano simple, siguiendo estas ecuaciones:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} \quad (\text{si } i \leq k \text{ o } j < k) \quad (7)$$

$$a_{ij}^{(k+1)} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad (\text{si } i > k \text{ y } j = k) \quad (8)$$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \left(\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right) a_{kj}^{(k)} \quad (\text{si } i > k \text{ y } j > k) \quad (9)$$

Estas ecuaciones describen en forma precisa la fase de eliminación hacia adelante del algoritmo de la eliminación gaussiana simple. Por ejemplo, la ecuación (7) establece que en el procedimiento de $\mathbf{A}^{(k)}$ a $\mathbf{A}^{(k+1)}$, no alteramos los renglones $1, 2, \dots, k$, o las columnas $1, 2, \dots, k-1$. La ecuación (8) muestra cómo se calculan los multiplicadores y se almacenan al pasar de $\mathbf{A}^{(k)}$ a $\mathbf{A}^{(k+1)}$. Por último, la ecuación (9) muestra cómo los múltiplos del renglón k se restan de los renglones $k+1, k+2, \dots, n$ para obtener $\mathbf{A}^{(k+1)}$ de $\mathbf{A}^{(k)}$.

Observe que un $\mathbf{A}^{(n)}$ es el resultado final del proceso. (Se denota como $\tilde{\mathbf{A}}$ en el enunciado del teorema). Las definiciones formales de $\mathbf{L} = (\ell_{ik})$ y $\mathbf{U} = (u_{kj})$ son, por lo tanto

$$\ell_{ik} = 1 \quad (i = k) \quad (10)$$

$$\ell_{ik} = a_{ik}^{(n)} \quad (k < i) \quad (11)$$

$$\ell_{ik} = 0 \quad (k > i) \quad (12)$$

$$u_{kj} = a_{kj}^{(n)} \quad (j \geq k) \quad (13)$$

$$u_{kj} = 0 \quad (j < k) \quad (14)$$

Ahora sacamos algunas consecuencias de estas ecuaciones. En primer lugar, se concluye inmediatamente de la ecuación (7) que

$$a_{ij}^{(i)} = a_{ij}^{(i+1)} = \cdots = a_{ij}^{(n)} \quad (15)$$

Asimismo, tenemos, de la ecuación (7),

$$a_{ij}^{(j+1)} = a_{ij}^{(j+2)} = \cdots = a_{ij}^{(n)} \quad (j < n) \quad (16)$$

De las ecuaciones (16) y (8), ahora tenemos

$$a_{ij}^{(n)} = a_{ij}^{(j+1)} = \frac{a_{ij}^{(j)}}{a_{jj}^{(j)}} \quad (j < n) \quad (17)$$

De las ecuaciones (17) y (11), se deduce que

$$\ell_{ik} = a_{ik}^{(n)} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad (k < i) \quad (18)$$

De las ecuaciones (13) y (15), tenemos

$$u_{kj} = a_{kj}^{(n)} = a_{kj}^{(k)} \quad (k \leq j) \quad (19)$$

Con la ayuda de todas estas ecuaciones podemos demostrar que $LU = A$. En primer lugar, consideremos el caso $i \leq j$. Entonces

$$\begin{aligned} (\mathbf{L}\mathbf{U})_{ij} &= \sum_{k=1}^n \ell_{ik} u_{kj} && [\text{definición de multiplicación}] \\ &= \sum_{k=1}^i \ell_{ik} u_{kj} && [\text{por la ecuación (12)}] \\ &= \sum_{k=1}^{i-1} \ell_{ik} u_{kj} + u_{ij} && [\text{por la ecuación (10)}] \\ &= \sum_{k=1}^{i-1} \left[\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right] a_{kj}^{(k)} + a_{ij}^{(i)} && [\text{por las ecuaciones (18) y (19)}] \\ &= \sum_{k=1}^{i-1} \left[a_{ij}^{(k)} - a_{ij}^{(k+1)} \right] + a_{ij}^{(i)} && [\text{por la ecuación (9)}] \\ &= a_{ij}^{(1)} = a_{ij} \end{aligned}$$

En el caso que falta, $i > j$, tenemos

$$\begin{aligned}
 (\mathbf{L}\mathbf{U})_{ij} &= \sum_{k=1}^n \ell_{ik} u_{kj} && [\text{definición de multiplicación}] \\
 &= \sum_{k=1}^j \ell_{ik} u_{kj} && [\text{por la ecuación (14)}] \\
 &= \sum_{k=1}^j \left[\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right] a_{kj}^{(k)} && [\text{por las ecuaciones (18) y (19)}] \\
 &= \sum_{k=1}^{j-1} \left[\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \right] a_{kj}^{(k)} + a_{ij}^{(j)} && \\
 &= \sum_{k=1}^{j-1} \left[a_{ij}^{(k)} - a_{ij}^{(k+1)} \right] + a_{ij}^{(j)} && [\text{por la ecuación (9)}] \\
 &= a_{ij}^{(1)} = a_{ij}
 \end{aligned}$$

■

Seudocódigo

El siguiente es el seudocódigo para realizar la factorización LU , que a veces se llama **factorización de Doolittle**:

```

integer i, k, n;  real array (aij)1:n × 1:n, (ℓij)1:n × 1:n, (uij)1:n × 1:n
for k = 1 to n do
    ℓkk ← 1
    for j = k to n do
        ukj ← akj - ∑s=1k-1 ℓks usj
    end do
    for i = k + 1 to n do
        ℓik ← (aik - ∑s=1k-1 ℓis usk) / ukk
    end do
end do

```

Resolución de sistemas lineales usando factorización LU

Una vez que se tiene la factorización LU de A podemos resolver el sistema

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

escribiendo

$$\mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{b}$$

Después resolvemos dos sistemas triangulares:

$$\mathbf{L}\mathbf{z} = \mathbf{b} \tag{20}$$

para \mathbf{z} y

$$\mathbf{U}\mathbf{x} = \mathbf{z} \quad (21)$$

para \mathbf{x} . Esto es particularmente útil para problemas que implican la misma matriz de coeficientes \mathbf{A} y muchos vectores diferentes en el lado derecho \mathbf{b} .

Puesto que \mathbf{L} es triangular inferior unitaria, \mathbf{z} se obtiene con el seudocódigo

```
integer  $i, n$ ; real array  $(b_i)_{1:n}, (\ell_{ij})_{1:n \times 1:n}, (z_i)_{1:n}$ 
 $z_1 \leftarrow b_1$ 
for  $i = 2$  to  $n$  do
     $z_i \leftarrow b_i - \sum_{j=1}^{i-1} \ell_{ij} z_j$ 
end for
```

Asimismo, \mathbf{x} se obtiene con el seudocódigo

```
integer  $i, n$ ; real array  $(u_{ij})_{1:n \times 1:n}, (x_i)_{1:n}, (z_i)_{1:n}$ 
 $x_n \leftarrow z_n / u_{nn}$ 
for  $i = n-1$  to  $1$  step  $-1$  do
     $x_i \leftarrow \left( z_i - \sum_{j=i+1}^n u_{ij} x_j \right) / u_{ii}$ 
end for
```

El primero de estos dos algoritmos se aplica en la fase hacia adelante de la eliminación gaussiana en el vector \mathbf{b} del lado derecho. [Recordemos que el arreglo ℓ_{ij} son los *multiplicadores* que se han guardado en el arreglo (a_{ij})]. La forma más fácil de verificar esta afirmación es usar la ecuación (6) y volver a escribir la ecuación

$$\mathbf{L}\mathbf{z} = \mathbf{b}$$

en la forma

$$\mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \cdots \mathbf{M}_{n-1}^{-1} \mathbf{z} = \mathbf{b}$$

De ésta, obtenemos inmediatamente

$$\mathbf{z} = \mathbf{M}_{n-1} \cdots \mathbf{M}_2 \mathbf{M}_1 \mathbf{b}$$

Así, las mismas operaciones que se utilizan para reducir \mathbf{A} a \mathbf{U} se van a utilizar en \mathbf{b} para obtener \mathbf{z} .

Otra forma de resolver la ecuación (20) es observar que lo que debe hacerse es formar

$$\mathbf{M}_{n-1} \mathbf{M}_{n-2} \cdots \mathbf{M}_2 \mathbf{M}_1 \mathbf{b}$$

Esto puede lograrse utilizando sólo el arreglo (b_i) , poniendo los resultados de nuevo en \mathbf{b} , es decir,

$$\mathbf{b} \leftarrow \mathbf{M}_k \mathbf{b}$$

Sabemos que M_k parece que se compone de multiplicadores negativos que se han guardado en la matriz (a_{ij}) . En consecuencia, tenemos

$$M_k b = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -a_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -a_{ik} & & & 1 \\ & & \vdots & & & \ddots \\ & & -a_{nk} & & & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_k \\ b_{k+1} \\ \vdots \\ b_i \\ \vdots \\ b_n \end{bmatrix}$$

Las entradas de b_1 a b_k no se cambian con esta multiplicación, mientras que b_i (para $i \geq k+1$) se sustituye por $-a_{ik}b_k + b_i$. Por lo tanto, el siguiente seudocódigo actualiza el arreglo (b_i) con base en los multiplicadores almacenados en el arreglo a :

```
integer i, k, n;  real array (aij)1:n x 1:n, (bi)1:n
for k = 1 to n - 1 do
    for i = k + 1 to n do
        bi ← bi - aikbk
    end for
end for
```

Este seudocódigo debe ser familiar. Es el proceso de actualización de \mathbf{b} de la sección 7.2.

El algoritmo para resolver la ecuación (21) es la fase de sustitución hacia atrás del proceso de eliminación gaussiana simple.

Factorización LDL^T

En la factorización LDL^T , L es triangular inferior unitaria y D es una matriz diagonal. Esta factorización se puede realizar si A es simétrica y tiene una factorización LU ordinaria, con L triangular inferior unitaria. Para ver esto, empezamos con

$$LU = A = A^T = (LU)^T = U^T L^T$$

Puesto que L es triangular inferior unitaria, es invertible y podemos escribir $U = L^{-1}U^T L^T$. Entonces $U(L^T)^{-1} = L^{-1}U^T$. Puesto que el lado derecho de esta ecuación es triangular inferior y el lado izquierdo es triangular superior, ambos lados son diagonales, digamos, D . De la ecuación $U(L^T)^{-1} = D$, tenemos $U = DL^T$ y $A = LU = LDL^T$.

Ahora deducimos el seudocódigo para obtener la factorización LDL^T de una matriz simétrica A en la que L es triangular inferior unitaria y D es diagonal. En nuestro análisis, se escribe a_{ij} , como elementos genéricos de A y ℓ_{ij} como elementos genéricos de L . La diagonal de D tiene

elementos d_{ii} , o d_i . De la ecuación $\mathbf{A} = \mathbf{LDL}^T$, tenemos

$$\begin{aligned} a_{ij} &= \sum_{v=1}^n \sum_{\mu=1}^n \ell_{iv} d_{v\mu} \ell_{\mu j}^T \\ &= \sum_{v=1}^n \sum_{\mu=1}^n \ell_{iv} d_v \delta_{v\mu} \ell_{j\mu} \\ &= \sum_{v=1}^n \ell_{iv} d_v \ell_{jv} \quad (1 \leq i, j \leq n) \end{aligned}$$

Use el hecho de que $\ell_{ij} = 0$ cuando $j > i$ y $\ell_{ii} = 0$ para continuar el argumento

$$a_{ij} = \sum_{v=1}^{\min(i,j)} \ell_{iv} d_v \ell_{jv} \quad (1 \leq i, j \leq n)$$

Suponga ahora que $j \leq i$. Entonces

$$\begin{aligned} a_{ij} &= \sum_{v=1}^j \ell_{iv} d_v \ell_{jv} \\ &= \sum_{v=1}^{j-1} \ell_{iv} d_v \ell_{jv} + \ell_{ij} d_j \ell_{jj} \\ &= \sum_{v=1}^{j-1} \ell_{iv} d_v \ell_{jv} + \ell_{ij} d_j \quad (1 \leq j \leq i \leq n) \end{aligned}$$

En particular, sea $j = i$. Obtenemos

$$a_{ii} = \sum_{v=1}^{i-1} \ell_{iv} d_v \ell_{iv} + d_i \quad (1 \leq i \leq n)$$

Equivalentemente, tenemos

$$d_i = a_{ii} - \sum_{v=1}^{i-1} d_v \ell_{iv}^2 \quad (1 \leq i \leq n)$$

Casos particulares de esto son

$$\begin{aligned} d_1 &= a_{11} \\ d_2 &= a_{22} - d_1 \ell_{21}^2 \\ d_3 &= a_{33} - d_1 \ell_{31}^2 - d_2 \ell_{32}^2 \\ &\text{etc.} \end{aligned}$$

Ahora podemos limitar nuestra atención a los casos $1 \leq j \leq i \leq n$, donde tenemos

$$a_{ij} = \sum_{v=1}^{j-1} \ell_{iv} d_v \ell_{jv} + \ell_{ij} d_j \quad (1 \leq j < i \leq n)$$

Resolviendo para ℓ_{ij} , obtenemos

$$\ell_{ij} = \left[a_{ij} - \sum_{v=1}^{j-1} \ell_{iv} d_v \ell_{jv} \right] / d_j \quad (1 \leq j < i \leq n)$$

Tomando $j = 1$, tenemos

$$\ell_{i1} = a_{i1} / d_1 \quad (2 \leq i \leq n)$$

Esta fórmula produce la columna uno en L . Tomando $j = 2$, tenemos

$$\ell_{i2} = (a_{i2} - \ell_{i1} d_1 \ell_{12}) / d_2 \quad (3 \leq i \leq n)$$

Esta fórmula produce la columna dos en L . El algoritmo formal para la factorización LDL^T es el siguiente:

```

integer i, j, n, v; real array (aij)1:n × 1:n, (ℓij)1:n × 1:n, (di)1:n
for j = 1 to n
    ℓjj = 1
    dj = ajj - ∑v=1j-1 dv ℓjv2
    for i = j + 1 to n
        ℓji = 0
        ℓij = (aij - ∑v/1j-1 ℓiv dv ℓjv) / dj
    end for
end for

```

EJEMPLO 1 Determine la factorización LDL^T de la matriz

$$A = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Solución Primero, determinamos la factorización LU :

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{2}{3} & 1 & 0 \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 & 2 & 1 \\ 0 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} = LU$$

A continuación, se sacan los elementos diagonales de U y se colocan en una matriz diagonal D , escribimos

$$U = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & \frac{3}{4} & 0 & 0 \\ 0 & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} = DL^T$$

Claramente, tenemos $A = LDL^T$.



Factorización de Cholesky

Cualquier matriz simétrica que tiene una factorización LU en la que L es triangular inferior unitaria, tiene una factorización LDL^T . La factorización de Cholesky $A = LL^T$ es una consecuencia simple de esta para el caso en el que A es simétrica y definida positiva.

Suponga en la factorización $A = LU$ que la matriz L es triangular inferior y la matriz U es triangular superior. Cuando L es triangular inferior *unitaria*, ésta se llama **factorización de Doolittle**. Cuando U es triangular superior *unitaria*, lleva el nombre de **factorización de Crout**. En el caso en que A es simétrica definida positiva y $U = L^T$, se llama la **factorización de Cholesky**. El matemático André Louis Cholesky demostró el siguiente resultado.

TEOREMA 2

Teorema de Cholesky acerca de la factorización LL^T

Si A es una matriz real, simétrica y definida positiva, entonces tiene una factorización única, $A = LL^T$, en la que L es triangular inferior con una diagonal positiva.

Recuerde que una matriz A es **simétrica y definida positiva** si $A = A^T$ y $x^T A x > 0$ para cada vector x distinto de cero. Se deduce que A es no singular porque, obviamente A , no puede mapear cualquier vector distinto de cero en 0. Además, considerando vectores especiales de la forma $x = (x_1, x_2, \dots, x_k, 0, 0, \dots, 0)^T$, vemos que los menores principales de A son también definidos positivos. El teorema 1 implica que A tiene una descomposición LU . Por la simetría de A , entonces tenemos, del análisis anterior, que $A = LDL^T$. Se puede demostrar que D es definida positiva y por lo tanto sus elementos d_{ii} son positivos. Denotando por $D^{1/2}$ a la matriz diagonal cuyos elementos diagonales son $\sqrt{d_{ii}}$, tenemos $A = \tilde{L} \tilde{L}^T$ donde $\tilde{L} \equiv LD^{1/2}$, que es la factorización de Cholesky. Dejamos la prueba de unicidad al lector.

El algoritmo para la factorización de Cholesky es un caso especial del algoritmo general de factorización LU . Si A es real, simétrica y definida positiva, por el teorema 2, tiene una factorización única de la forma $A = LL^T$, en la que L es triangular inferior y tiene diagonal positiva. Así, en la ecuación $A = LU$, $U = L^T$. En el k -ésimo paso del algoritmo general, la entrada en la diagonal se calcula por

$$\ell_{kk} = \left(a_{kk} - \sum_{s=1}^{k-1} \ell_{ks}^2 \right)^{1/2} \quad (22)$$

El algoritmo para la **factorización de Cholesky** será entonces el siguiente:

```

integer  $i, k, n, s$ ; real array  $(a_{ij})_{1:n \times 1:n}, (\ell_{ij})_{1:n \times 1:n}$ 
for  $k = 1$  to  $n$  do
     $\ell_{kk} \leftarrow \left( a_{kk} - \sum_{s=1}^{k-1} \ell_{ks}^2 \right)^{1/2}$ 
    for  $i = k + 1$  to  $n$  do
         $\ell_{ik} \leftarrow \left( a_{ik} - \sum_{s=1}^{k-1} \ell_{is} \ell_{ks} \right) / \ell_{kk}$ 
    end do
end do

```

El teorema 2 garantiza que $\ell_{kk} > 0$. Observe que la ecuación (22) nos da el siguiente límite:

$$a_{kk} = \sum_{s=1}^k \ell_{ks}^2 \geq \ell_{kj}^2 \quad (j \leq k)$$

de lo que concluimos que

$$|\ell_{kj}| \leq \sqrt{a_{kk}} \quad (1 \leq j \leq k)$$

Por lo tanto, cualquier elemento de \mathbf{L} está anotado por la raíz cuadrada del elemento diagonal correspondiente en \mathbf{A} . Esto implica que los elementos de \mathbf{L} no se hacen grandes con respecto a \mathbf{A} , aun sin ningún tipo de pivoteo. En el algoritmo de Cholesky (y en los algoritmos de Doolittle), el producto punto de vectores se debe calcular con doble precisión para evitar la acumulación de errores de redondeo.

EJEMPLO 2 Determine la factorización de Cholesky de la matriz del ejemplo 1.

Solución Usando los resultados del ejemplo 1, escribimos

$$\mathbf{A} = \mathbf{LDL}^T = (\mathbf{LD}^{1/2})(\mathbf{D}^{1/2}\mathbf{L}^T) = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$$

donde

$$\begin{aligned} \tilde{\mathbf{L}} &= \mathbf{LD}^{1/2} \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{2}{3} & 1 & 0 \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & \frac{1}{2}\sqrt{3} & 0 & 0 \\ 0 & 0 & \sqrt{\frac{2}{3}} & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \\ &= \begin{bmatrix} 2 & 0 & 0 & 0 \\ \frac{3}{2} & \frac{1}{2}\sqrt{3} & 0 & 0 \\ 1 & \frac{1}{3}\sqrt{3} & \sqrt{\frac{2}{3}} & 0 \\ \frac{1}{2} & \frac{1}{6}\sqrt{3} & \frac{1}{2}\sqrt{\frac{2}{3}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 2.0000 & 0 & 0 & 0 \\ 1.5000 & 0.8660 & 0 & 0 \\ 1.0000 & 0.5774 & 0.8165 & 0 \\ 0.5000 & 0.2887 & 0.4082 & 0.7071 \end{bmatrix} \end{aligned}$$

Claramente, $\tilde{\mathbf{L}}$ es la matriz triangular inferior en la factorización de Cholesky $\mathbf{A} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$. ■

Múltiples lados derechos

Muchos paquetes de software para resolver sistemas lineales permiten la entrada de varios lados derechos. Supongamos una matriz \mathbf{B} de $n \times m$

$$\mathbf{B} = [\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)}]$$

en la que cada columna corresponde al lado derecho de los m sistemas lineales

$$\mathbf{Ax}^{(j)} = \mathbf{b}^{(j)}$$

para $1 \leq j \leq m$. Así, podemos escribir

$$\mathbf{A}[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}] = [\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \dots, \mathbf{b}^{(m)}]$$

o

$$\mathbf{A}\mathbf{X} = \mathbf{B}$$

Por ejemplo, el procedimiento *Gauss* puede utilizarse una vez para producir una factorización de \mathbf{A} y el procedimiento *Simple* puede usarse m veces con los vectores del lado derecho $\mathbf{b}^{(j)}$ para encontrar los m vectores solución $\mathbf{x}^{(j)}$ para $1 \leq j \leq m$. Puesto que la fase de factorización se puede hacer con $\frac{1}{3}n^3$ operaciones largas mientras que cada una de las fases de sustitución hacia atrás requiere n^2 operaciones largas, todo este proceso se puede hacer con $\frac{1}{3}n^3 + mn^2$ operaciones largas. Esto es mucho menor que $m(\frac{1}{3}n^3 + n^2)$, que es lo que tomaría si cada uno de los m sistemas lineales se resolvieran por separado.

Cálculo de \mathbf{A}^{-1}

En algunas aplicaciones, como en estadística, puede ser necesario calcular la inversa de una matriz \mathbf{A} y presentarla de forma explícita como \mathbf{A}^{-1} . Esto puede hacerse mediante el uso de los procedimientos *Gauss_Simple*. Si una matriz \mathbf{A} de $n \times n$ tiene una inversa, esta es una matriz \mathbf{X} de $n \times n$ con la propiedad de que

$$\mathbf{A}\mathbf{X} = \mathbf{I} \quad (23)$$

donde \mathbf{I} es la matriz identidad. Si $\mathbf{x}^{(j)}$ denota la columna j de \mathbf{X} e $\mathbf{I}^{(j)}$ denota la j -ésima columna de \mathbf{I} , entonces la ecuación matricial (23) se puede escribir como

$$\mathbf{A}[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}] = [\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \dots, \mathbf{I}^{(n)}]$$

Esto puede escribirse como n sistemas lineales de ecuaciones de la forma

$$\mathbf{A}\mathbf{x}^{(j)} = \mathbf{I}^{(j)} \quad (1 \leq j \leq n)$$

Ahora use el procedimiento *Gauss* una vez para producir una factorización de \mathbf{A} y use el procedimiento *Simple* n veces con los vectores del lado derecho $\mathbf{I}^{(j)}$ para $1 \leq j \leq n$. Esto equivale a resolver, una a la vez, las columnas de \mathbf{A}^{-1} , que son $\mathbf{x}^{(j)}$. Por lo tanto,

$$\mathbf{A}^{-1} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}]$$

Un consejo en el cálculo de la inversa de una matriz: en la solución de un sistema lineal $\mathbf{Ax} = \mathbf{b}$, no es recomendable determinar \mathbf{A}^{-1} y luego calcular el producto matriz-vector $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ porque esto requiere muchos cálculos innecesarios, en comparación con la solución directa de $\mathbf{Ax} = \mathbf{b}$ para \mathbf{x} .

Ejemplo con uso de paquetes de software

Una **matriz de permutación** es una matriz \mathbf{P} de $n \times n$ que surge de la matriz identidad al permutar sus renglones. Luego resulta que la permutación de los renglones de una matriz \mathbf{A} de $n \times n$ se puede realizar con una multiplicación de \mathbf{A} a la izquierda de \mathbf{P} . Toda matriz de permutación es singular, puesto que los renglones aun forman una base de \mathbb{R}^n . Cuando se realiza la eliminación gaussiana con pivoteo de renglones en una matriz \mathbf{A} , el resultado se puede expresar como

$$\mathbf{PA} = \mathbf{LU}$$

donde \mathbf{L} es triangular inferior y \mathbf{U} es triangular superior. La matriz \mathbf{PA} es \mathbf{A} con sus renglones rearrreglados. Si tenemos la factorización \mathbf{LU} de \mathbf{PA} , ¿cómo podemos resolver el sistema $\mathbf{Ax} = \mathbf{b}$?

Primero, escríbalo como

$$\mathbf{P} \mathbf{A} \mathbf{x} = \mathbf{P} \mathbf{b}$$

entonces $\mathbf{L} \mathbf{U} \mathbf{x} = \mathbf{P} \mathbf{b}$. Sea $\mathbf{y} = \mathbf{U} \mathbf{x}$, por lo que nuestro problema ahora es

$$\mathbf{L} \mathbf{y} = \mathbf{P} \mathbf{b}$$

$$\mathbf{U} \mathbf{x} = \mathbf{y}$$

La primera ecuación es fácil de resolver para \mathbf{y} , a continuación, la segunda ecuación es fácil de resolver para \mathbf{x} . Los sistemas de software matemático como Matlab, Maple y Mathematica producen factorizaciones de la forma $\mathbf{PA} = \mathbf{LU}$ con una instrucción.

EJEMPLO 3 Use sistemas de software matemático como Matlab, Maple y Mathematica para encontrar la factorización \mathbf{LU} de esta matriz:

$$\mathbf{A} = \begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \quad (24)$$

Solución Primero, se usa Maple y se encuentra esta factorización:

$$\mathbf{A} = \mathbf{L} \mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{bmatrix} \begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix}$$

Después, usamos Matlab y se encuentra una factorización diferente:

$$\begin{aligned} \mathbf{P} \mathbf{A} &= \widehat{\mathbf{L}} \widehat{\mathbf{U}} \\ \widehat{\mathbf{L}} &= \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0.2500 & 1.0000 & 0 & 0 \\ -0.5000 & 0 & 1.0000 & 0 \\ 0.5000 & -0.1818 & 0.0909 & 1.0000 \end{bmatrix} \\ \widehat{\mathbf{U}} &= \begin{bmatrix} 12.0000 & -8.0000 & 6.0000 & 10.0000 \\ 0 & -11.0000 & 7.5000 & 0.5000 \\ 0 & 0 & 4.0000 & -13.0000 \\ 0 & 0 & 0 & 0.2727 \end{bmatrix} \\ \mathbf{P} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

donde \mathbf{P} es una matriz de permutación correspondiente a la estrategia de pivoteo utilizada. Por último, se utiliza Mathematica para crear esta descomposición \mathbf{LU} :

$$\begin{bmatrix} 3 & -13 & 9 & 3 \\ -2 & -22 & 19 & -12 \\ 2 & -\frac{12}{11} & \frac{52}{11} & -\frac{166}{11} \\ 4 & -2 & \frac{22}{13} & -\frac{6}{13} \end{bmatrix}$$

La salida es en un esquema de almacenamiento compacto que contiene la matriz triangular inferior y la matriz triangular superior en una sola matriz. Sin embargo, el arreglo de almacenamiento puede ser complicado porque los renglones están generalmente permutados durante la factorización para hacer que el proceso de solución sea estable numéricamente. Compruebe que este factorización corresponde a la permutación de los renglones de la matriz A en el orden 3, 4, 1, 2. ■

Resumen

(1) Si $A = (a_{ij})$ es una matriz de $n \times n$ de tal manera que la fase de eliminación hacia adelante del algoritmo gaussiano simple se puede aplicar sin encontrar divisores cero, entonces la matriz resultante se puede denotar por $\tilde{A} = (\tilde{a}_{ij})$, donde

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \tilde{a}_{21} & 1 & 0 & \cdots & 0 \\ \tilde{a}_{31} & \tilde{a}_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{n,n-1} & 1 \end{bmatrix}$$

y

$$\mathbf{U} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13} & \cdots & \tilde{a}_{1n} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ 0 & 0 & \tilde{a}_{33} & \cdots & \tilde{a}_{3n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \tilde{a}_{nn} \end{bmatrix}$$

Esta es la **factorización LU** de A , de modo que $A = LU$, donde L es triangular inferior unitaria y U es triangular superior. Cuando realizamos el proceso de eliminación gaussiana hacia adelante en A , el resultado es la matriz triangular superior U . La matriz L es la matriz triangular unitaria inferior cuyos elementos son negativos de los multiplicadores en las ubicaciones de los elementos que eran cero.

(2) También se puede dar una descripción formal de la siguiente manera. La matriz U se puede obtener mediante la aplicación de una serie de matrices \mathbf{M}_k a la izquierda de A . El k ésimo paso de la eliminación gaussiana corresponde a una matriz triangular inferior unitaria \mathbf{M}_k , que es el producto de $n - k$ matrices elementales

$$\mathbf{M}_k = \mathbf{M}_{nk} \mathbf{M}_{n-1,k} \cdots \mathbf{M}_{k+1,k}$$

donde cada matriz elemental \mathbf{M}_{ik} es unitaria triangular inferior. Si $\mathbf{M}_{qp} A$ se obtiene restando λ veces el renglón p del renglón q en una matriz A con $p < q$, entonces los elementos de $\mathbf{M}_{qp} = (m_{ij})$ son

$$m_{ij} = \begin{cases} 1 & \text{si } i = j \\ -\lambda & \text{si } i = q \text{ y } j = p \\ 0 & \text{en todos los otros casos} \end{cases}$$

El proceso completo de eliminación gaussiana se resume al escribir

$$\mathbf{M}_{n-1} \cdots \mathbf{M}_2 \mathbf{M}_1 A = U$$

Puesto que cada M_k es invertible, tenemos

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U$$

Cada M_k es una matriz triangular inferior unitaria y lo mismo es cierto de cada inversa M_k^{-1} , así como de sus productos. Por lo tanto, la matriz

$$L = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1}$$

es triangular inferior unitaria.

(3) Para matrices simétricas, tenemos la factorización LDL^T y para matrices simétricas definidas positivas, tenemos la factorización LL^T , que también se conoce como factorización de Cholesky.

(4) Si se tiene la factorización LU de A podemos resolver el sistema

$$Ax = b$$

al resolver dos sistemas triangulares:

$$\begin{cases} Ly = b & \text{para } y \\ Ux = y & \text{para } x \end{cases}$$

Esto es útil para problemas que implican la misma matriz de coeficientes A y muchos vectores b del lado derecho. Por ejemplo, sea B una matriz de $m \times n$ de la forma

$$B = [b^{(1)}, b^{(2)}, \dots, b^{(m)}]$$

donde cada columna corresponde a la parte derecha de los m sistemas lineales

$$Ax^{(j)} = b^{(j)} \quad (1 \leq j \leq m)$$

Así, podemos escribir

$$A[x^{(1)}, x^{(2)}, \dots, x^{(m)}] = [b^{(1)}, b^{(2)}, \dots, b^{(m)}]$$

o

$$AX = B$$

Un caso especial de esto es para calcular la inversa de una matriz invertible A de $n \times n$. Escribimos

$$AX = I$$

donde I es la matriz identidad. Si $x^{(j)}$ denota la j ésima columna de X e $I^{(j)}$ denota la j ésima columna de I , esto se puede escribir como

$$A[x^{(1)}, x^{(2)}, \dots, x^{(n)}] = [I^{(1)}, I^{(2)}, \dots, I^{(n)}]$$

o como n sistemas de ecuaciones lineales de la forma

$$Ax^{(j)} = I^{(j)} \quad (1 \leq j \leq n)$$

Podemos utilizar la factorización LU para resolver estos n sistemas de manera eficiente, obteniendo

$$A^{-1} = [x^{(1)}, x^{(2)}, \dots, x^{(n)}]$$

(5) Cuando se realiza eliminación gaussiana con pivoteo de renglones en una matriz A , el resultado se expresa como

$$PA = LU$$

donde P es una **matriz de permutación**, L es triangular inferior unitaria y U es triangular superior. Aquí, la matriz PA es A con sus renglones intercambiados. Podemos resolver el sistema $Ax = b$ al resolver

$$\begin{cases} Ly = Pb & \text{para } y \\ Ux = y & \text{para } x \end{cases}$$

Problemas 8.1

- 1.** Usando eliminación gaussiana simple, factorice las matrices siguientes en la forma $A = LU$, donde L es una matriz triangular inferior unitaria y U es una matriz triangular superior.

a. $A = \begin{bmatrix} 3 & 0 & 3 \\ 0 & -1 & 3 \\ 1 & 3 & 0 \end{bmatrix}$

b. $A = \begin{bmatrix} 1 & 0 & \frac{1}{3} & 0 \\ 0 & 1 & 3 & -1 \\ 3 & -3 & 0 & 6 \\ 0 & 2 & 4 & -6 \end{bmatrix}$

c. $A = \begin{bmatrix} -20 & -15 & -10 & -5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

- 2.** Considere la matriz

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 0 & 9 & 4 & 0 \\ 5 & 0 & 8 & 10 \end{bmatrix}$$

- a. Determine una matriz triangular inferior unitaria M y una matriz triangular superior U tal que $MA = U$.
 b. Determine una matriz triangular inferior unitaria L y una matriz triangular superior U tal que $A = LU$. Muestre que $ML = I$, por lo que $L = M^{-1}$.

- 3.** Considere la matriz

$$A = \begin{bmatrix} 25 & 0 & 0 & 0 & 1 \\ 0 & 27 & 4 & 3 & 2 \\ 0 & 54 & 58 & 0 & 0 \\ 0 & 108 & 116 & 0 & 0 \\ 100 & 0 & 0 & 0 & 24 \end{bmatrix}$$

- a. Determine la matriz triangular inferior unitaria M y la matriz triangular superior U tal que $MA = U$.
 b. Determine $M^{-1} = L$ tal que $A = LU$.

- 4.** Considere la matriz

$$A = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 1 & 1 \\ 3 & 2 & 1 \end{bmatrix}$$

- a. Demuestre que A no puede factorizarse como el producto de una matriz triangular inferior unitaria y una matriz triangular superior.
 b. Intercambie los renglones de A para que se pueda hacer esto.

5. Considere la matriz

$$A = \begin{bmatrix} a & 0 & 0 & z \\ 0 & b & 0 & 0 \\ 0 & x & c & 0 \\ w & 0 & y & d \end{bmatrix}$$

- a. Determine una matriz triangular inferior unitaria M y una matriz triangular superior U tal que $MA = U$.
 b. Determine una matriz triangular inferior L' y una matriz triangular superior unitaria U' tal que $A = L'U'$.

6. Considere la matriz

$$A = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}$$

Factorice A de las formas siguientes:

- a. $A = LU$, donde L es triangular inferior unitaria y U es triangular superior.
 b. $A = LDU'$, donde L es triangular inferior unitaria, D es diagonal y U' es triangular superior unitaria.
 c. $A = L'U'$, donde L' es triangular inferior y U' es triangular superior unitaria.
 d. $A = (L')^T(L')^T$, donde L'' es triangular inferior.
 e. Evalúe el determinante de A . Sugerencia: $\det(A) = \det(L) \det(D) \det(U') = \det(D)$.

7. Considere la matriz de Hilbert de 3×3

$$A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

Repita el problema anterior usando esta matriz.

8. Encuentre la descomposición LU , donde L es triangular inferior unitaria, para

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & -1 \\ -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

9. Considere

$$A = \begin{bmatrix} 2 & -1 & 2 \\ 2 & -3 & 3 \\ 6 & -1 & 8 \end{bmatrix}$$

a. Encuentre la factorización matricial $A = LDU'$, donde L es triangular inferior unitaria, D es diagonal y U' es triangular superior unitaria.

b. Use esta descomposición de A para resolver $Ax = b$, donde $b = [-2, -5, 0]^T$.

10. Repita el problema anterior para

$$A = \begin{bmatrix} -2 & 1 & -2 \\ -4 & 3 & -3 \\ 2 & 2 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 4 \\ 4 \end{bmatrix}$$

11. Considere el sistema de ecuaciones

$$\left\{ \begin{array}{l} 6x_1 = 12 \\ 6x_2 + 3x_1 = -12 \\ 7x_3 - 2x_2 + 4x_1 = 14 \\ 21x_4 + 9x_3 - 3x_2 + 5x_1 = -2 \end{array} \right.$$

a. Resuelva para x_1, x_2, x_3 y x_4 (en orden) con sustitución hacia adelante.

b. Escriba este sistema en notación matricial $Ax = b$, donde $x = [x_1, x_2, x_3, x_4]^T$. Determine la factorización LU , $A = LU$, donde L es triangular inferior unitaria y U es triangular superior.

12. Dadas

$$A = \begin{bmatrix} 3 & 2 & -1 \\ 5 & 3 & 2 \\ -1 & 1 & -3 \end{bmatrix}, \quad L^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{5}{3} & 1 & 0 \\ -8 & 5 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 3 & 2 & -1 \\ 0 & -\frac{1}{3} & \frac{11}{3} \\ 0 & 0 & 15 \end{bmatrix}$$

obtenga la inversa de A al resolver $UX^{(j)} = L^{-1}I^{(j)}$ para $j = 1, 2, 3$.

13. Usando el sistema de ecuaciones (2), forme $M = M_3M_2M_1$ y determine M^{-1} . Compruebe que $M^{-1} = L$. ¿Por qué esto, en general, no es una buena idea?

14. Considere la matriz $A =$ tridiagonal (a_{ij}) , donde $a_{ii} \neq 0$.

a. Establezca el algoritmo

```

integer i
real array (aij)1:n × 1:n, (ℓij)1:n × 1:n, (uij)1:n × 1:n
ℓ11 ← a11
for i = 2 to 4 do
    ℓi,i-1 ← ai,i-1
    ui-1,i ← ai-1,i / ℓi-1,i-1
    ℓi,i ← ai,i - ℓi,i-1ui-1,i
end for
```

para determinar los elementos de una matriz tridiagonal inferior $L = (\ell_{ij})$ y una matriz tridiagonal superior *unitaria* $U = (u_{ij})$ tal que $A = LU$.

b. Establezca el algoritmo

```

integer  $i$ ; real array  $(a_{ij})_{1:n \times 1:n}, (\ell_{i,j})_{1:n \times 1:n}, (u_{i,j})_{1:n \times 1:n}$ 
 $u_{11} \leftarrow a_{11}$ 
for  $i = 2$  to  $4$  do
     $u_{i-1,i} \leftarrow a_{i-1,i}$ 
     $\ell_{i,i-1} \leftarrow a_{i,i-1} / u_{i-1,i-1}$ 
     $u_{i,j} \leftarrow a_{i,i} - \ell_{i,i-1}u_{i-1,i}$ 
end for

```

para determinar los elementos de una matriz triangular inferior unitaria $L = (\ell_{ij})$ y una matriz tridiagonal superior unitaria $U = (u_{ij})$ tal que $A = LU$.

Con ciclos extendidos, podemos generalizar estos algoritmos a matrices tridiagonales de $n \times n$.

- 15.** Muestre que la ecuación $AX = B$ se puede resolver por eliminación gaussiana con pivoteo parcial escalado en $(n^3/3) + mn^2 + \mathcal{O}(n^2)$ multiplicaciones y divisiones, donde, A , X y B son matrices de orden $n \times n$, $n \times m$ y $n \times m$, respectivamente. Por lo tanto, si B es de $n \times n$, entonces la matriz solución X de $n \times n$ se puede encontrar con eliminación gaussiana con pivoteo parcial escalado con $\frac{4}{3}n^3 + \mathcal{O}(n^2)$ multiplicaciones y divisiones. *Sugerencia:* si $X^{(j)}$ y $B^{(j)}$ son las j -ésimas columnas de X y B , respectivamente, entonces $AX^{(j)} = B^{(j)}$.

- 16.** Sea X una matriz cuadrada que tiene la forma

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

donde A y D son matrices cuadradas y A^{-1} existe. Se sabe que X^{-1} existe si y sólo si también existe $(D - CA^{-1}B)^{-1}$. Compruebe que X^{-1} está dada por

$$X^{-1} = \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & (D - CA^{-1}B)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix}$$

Como una aplicación, calcule la inversa de lo siguiente:

$$\text{a. } X = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{array} \right] \quad \text{b. } X = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 2 \end{array} \right]$$

- 17.** Sea A una matriz compleja de $n \times n$ tal que A^{-1} existe. Compruebe que

$$\begin{bmatrix} A & \overline{A} \\ -Ai & -\overline{Ai} \end{bmatrix}^{-1} = \frac{1}{2} \begin{bmatrix} A^{-1} & A^{-1}i \\ \overline{A}^{-1} & -\overline{A}^{-1}i \end{bmatrix}$$

donde \overline{A} denota la conjugada compleja de A ; si $A = (a_{ij})$, entonces $\overline{A} = (\overline{a}_{ij})$. Recuerde que para un número complejo $z = a + bi$, donde a y b son reales y $\overline{z} = a - bi$.

- 18.** Encuentre la factorización LU de esta matriz:

$$A = \begin{bmatrix} 2 & 2 & 1 \\ 4 & 7 & 2 \\ 2 & 11 & 5 \end{bmatrix}$$

- 19.** a. Demuestre que el producto de dos matrices triangulares inferiores es triangular inferior.
 b. Demuestre que el producto de dos matrices triangulares inferiores unitarias es triangular inferior unitario.
 c. Demuestre que la inversa de una matriz triangular inferior unitaria es triangular inferior unitaria.
 d. Usando la operación de transposición, demuestre que todos los resultados anteriores son verdaderos para matrices triangulares superiores.

20. Sean L triangular inferior, U triangular superior y D diagonal.

- a. Si L y U son ambas triangulares unitarias y LDU es diagonal, ¿entonces L y U son diagonales?
 b. Si LDU es no singular y diagonal, ¿entonces L y U son diagonales?
 c. Si L y U son ambas triangulares unitarias y si LDU es diagonal, ¿entonces $L = U = I$?

21. Determine la factorización LDL^T para la siguiente matriz:

$$A = \begin{bmatrix} 1 & 2 & -1 & 1 \\ 2 & 3 & -4 & 3 \\ -1 & -4 & -1 & 3 \\ 1 & 3 & 3 & 0 \end{bmatrix}$$

22. Encuentre la factorización de Cholesky de

$$A = \begin{bmatrix} 4 & 6 & 10 \\ 6 & 25 & 19 \\ 10 & 19 & 62 \end{bmatrix}$$

23. Considere el sistema

$$\begin{bmatrix} A & \mathbf{0} \\ B & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ d \end{bmatrix}$$

Muestre cómo resolver el sistema de forma más económica utilizando submatrices en lugar de usar todo el sistema. Dé una estimación del costo computacional tanto con el método nuevo como con el viejo. Este problema ilustra la solución de un sistema de bloque lineal con una estructura especial.

24. Determine la factorización LDL^T de la matriz

$$A = \begin{bmatrix} 5 & 35 & -20 & 65 \\ 35 & 244 & -143 & 461 \\ -20 & -143 & 73 & -232 \\ 65 & 461 & -232 & 856 \end{bmatrix}$$

¿Puede encontrar la factorización de Cholesky?

25. (Factorizaciones dispersas) Considere las siguientes matrices simétricas dispersas con el patrón diferente de cero que se muestra donde las entradas diferentes de cero en la matriz están señaladas con el símbolo \times y las entradas iguales a cero están en blanco. Muestre el patrón diferente de cero en la matriz L para la factorización de Cholesky usando el símbolo $+$ para llenar una entrada cero por una entrada diferente de cero.

$$\mathbf{a. } \mathbf{A} = \left[\begin{array}{ccc|c|c} \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ \hline \times & & & \times & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ \hline & & \times & \times & \\ & & \times & \times & \\ & & \times & \times & \\ \end{array} \right]$$

$$\mathbf{b. } \mathbf{A} = \left[\begin{array}{ccc|c|c} \times & & \times & \times & \\ & \times & \times & \times & \\ & & \times & \times & \\ \hline & & \times & \times & \\ \times & \times & \times & \times & \\ \times & \times & \times & \times & \\ \hline \times & \times & \times & \times & \\ \times & & \times & \times & \\ \times & & \times & \times & \\ \end{array} \right]$$

$$\mathbf{c. } \mathbf{A} = \left[\begin{array}{ccc|c|c} \times & \times & & \times & \\ & \times & \times & & \\ & & \times & & \\ \hline & & \times & \times & \\ & & \times & \times & \\ & & \times & \times & \\ \hline \times & & \times & \times & \\ \times & & \times & \times & \\ \times & & \times & \times & \\ \end{array} \right]$$

Problemas de cómputo 8.1

- Escriba y pruebe un procedimiento para la implementación de los algoritmos del problema 8.1.14.
- La factorización $\mathbf{A} = \mathbf{LU}$ de $n \times n$, donde $\mathbf{L} = (\ell_{ij})$ es triangular inferior y $\mathbf{U} = (u_{ij})$ es triangular superior, se puede calcular directamente con el algoritmo siguiente (siempre que no haya divisiones entre cero): especifique ya sea ℓ_{11} o u_{11} y calcule la otra tal que $\ell_{11}u_{11} = a_{11}$. Calcule la primera columna de \mathbf{L} con

$$\ell_{i1} = \frac{a_{i1}}{u_{11}} \quad (1 \leq i \leq n)$$

y calcule el primer renglón en \mathbf{U} con

$$u_{1j} = \frac{a_{1j}}{\ell_{11}} \quad (1 \leq j \leq n)$$

Ahora suponga que las columnas $1, 2, \dots, k-1$ se han calculado en \mathbf{L} y que los renglones $1, 2, \dots, k-1$ se han calculado en \mathbf{U} . En el k -ésimo paso, especifique ya sea ℓ_{kk} o u_{kk} y calcule la otra tal que

$$\ell_{kk}u_{kk} = a_{kk} - \sum_{m=1}^{k-1} \ell_{km}u_{mk}$$

Calcule la k -ésima columna en \mathbf{L} con

$$\ell_{ik} = \frac{1}{u_{kk}} \left(a_{ik} - \sum_{m=1}^{k-1} \ell_{im}u_{mk} \right) \quad (k \leq i \leq n)$$

y calcule el k ésmo renglón en \mathbf{U} con

$$u_{kj} = \frac{1}{\ell_{kk}} \left(a_{kj} - \sum_{m=1}^{k-1} \ell_{km} u_{mj} \right) \quad (k \leq j \leq n)$$

Este algoritmo se continúa hasta que todos los elementos de \mathbf{U} y \mathbf{L} están completamente determinados. Cuando $\ell_{ii} = 1$ ($1 \leq i \leq n$), este procedimiento se llama **factorización de Doolittle**, y cuando $u_{jj} = 1$ ($1 \leq j \leq n$), se conoce como **factorización de Crout**.

Defina la matriz de prueba

$$\mathbf{A} = \begin{bmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{bmatrix}$$

Utilizando el algoritmo anterior, calcule e imprima las factorizaciones de modo que los elementos diagonales de \mathbf{L} y \mathbf{U} sean de las siguientes formas:

$\text{diag}(\mathbf{L})$	$\text{diag}(\mathbf{U})$	
[1, 1, 1, 1]	[?, ?, ?, ?]	Doolittle
[?, ?, ?, ?]	[1, 1, 1, 1]	Crout
[1, ?, 1, ?]	[?, 1, ?, 1]	
[?, 1, ?, 1]	[1, ?, 1, ?]	
[?, ?, 7, 9]	[3, 5, ?, ?]	

Aquí el signo de interrogación significa que se va a calcular la entrada. Escriba el código para comprobar los resultados multiplicando a \mathbf{L} por \mathbf{U} .

3. Escriba

procedure $\text{Poly}(n, (a_{ij}), (c_i), k, (y_{ij}))$

para calcular la matriz de $n \times n$, $p_k(\mathbf{A})$ guardada en el arreglo (y_{ij}) :

$$y_k = p_k(\mathbf{A}) = c_0 \mathbf{I} + c_1 \mathbf{A} + c_2 \mathbf{A}^2 + \cdots + c_k \mathbf{A}^k$$

donde \mathbf{A} es una matriz de $n \times n$ y p_k es un polinomio de grado k . Aquí (c_i) son constantes reales para $0 \leq i \leq k$. Utilice la multiplicación anidada y escriba un código eficiente. Pruebe el procedimiento Poly en los siguientes datos:

Caso 1.

$$\mathbf{A} = \mathbf{I}_5, \quad p_3(x) = 1 - 5x + 10x^3$$

Caso 2.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad p_2(x) = 1 - 2x + x^2$$

Caso 3.

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 4 \\ 0 & 0 & 8 \\ 0 & 0 & 0 \end{bmatrix}, \quad p_3(x) = 1 + 3x - 3x^2 + x^3$$

"Caso 4.

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}, \quad p_5(x) = 10 + x - 2x^2 + 3x^3 - 4x^4 + 5x^5$$

Caso 5.

$$\mathbf{A} = \begin{bmatrix} -20 & -15 & -10 & -5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad p_4(x) = 5 + 10x + 15x^2 + 20x^3 + x^4$$

Caso 6.

$$\mathbf{A} = \begin{bmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{bmatrix}, \quad p_4(x) = 1 - 100x + 146x^2 - 35x^3 + x^4$$

4. Escriba y pruebe un procedimiento para determinar \mathbf{A}^{-1} para una matriz cuadrada \mathbf{A} de orden n dada. Su procedimiento debe utilizar los procedimientos *Gauss* y *Solve*.
5. Escriba y pruebe un procedimiento para resolver el sistema $\mathbf{AX} = \mathbf{B}$ en el que \mathbf{A} , \mathbf{X} y \mathbf{B} son matrices de orden $n \times n$, $n \times m$ y $n \times m$, respectivamente. Compruebe que el procedimiento funciona en varios casos de prueba, uno de los cuales es $\mathbf{B} = \mathbf{I}$, por lo que la solución \mathbf{X} es la inversa de \mathbf{A} . *Sugerencia:* consulte el problema 8.1.15.
6. Escriba y pruebe un procedimiento para calcular directamente la inversa de una matriz tridiagonal. Suponga que no se necesita pivotear.
7. (Continuación) Pruebe el procedimiento del problema de cómputo anterior en la matriz tridiagonal simétrica \mathbf{A} de orden 10:

$$\mathbf{A} = \begin{bmatrix} -2 & 1 & & & & & & & & \\ 1 & -2 & 1 & & & & & & & \\ & 1 & -2 & 1 & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & \\ & & & 1 & -2 & 1 & & & & \\ & & & & 1 & -2 & 1 & & & \\ & & & & & 1 & -2 & 1 & & \\ & & & & & & 1 & -2 & 1 & \\ & & & & & & & 1 & -2 & \\ & & & & & & & & 1 & -2 \end{bmatrix}$$

Se sabe que la inversa de esta matriz es

$$(\mathbf{A}^{-1})_{ij} = (\mathbf{A}^{-1})_{ji} = \frac{-i(n+1-j)}{(n+1)} \quad (i \leq j)$$

8. Investigue las dificultades numéricas al invertir la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} -0.0001 & 5.096 & 5.101 & 1.853 \\ 0. & 3.737 & 3.740 & 3.392 \\ 0. & 0. & 0.006 & 5.254 \\ 0. & 0. & 0. & 4.567 \end{bmatrix}$$

9. Considere las siguientes dos matrices de prueba:

$$\mathbf{A} = \begin{bmatrix} 4 & 6 & 10 \\ 6 & 25 & 19 \\ 10 & 19 & 62 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 4 & 6 & 10 \\ 6 & 13 & 19 \\ 10 & 19 & 62 \end{bmatrix}$$

Muestre que la primera factorización de Cholesky tiene todos los enteros en la solución, mientras que la segunda tiene todos los enteros hasta el último paso, donde hay una raíz cuadrada.

- a. Programe el algoritmo de Cholesky.
 - b. Use Matlab, Maple o Mathematica para encontrar las factorizaciones de Cholesky.
10. Sea \mathbf{A} real, simétrica y definida positiva. ¿También es cierto lo mismo para la matriz que se obtiene eliminando el primer renglón y la primera columna de \mathbf{A} ?
11. Diseñe un código para invertir una matriz triangular inferior unitaria. Pruebelo con la matriz siguiente:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 5 & 2 & 1 & 0 \\ 7 & 4 & -3 & 1 \end{bmatrix}$$

12. Compruebe el ejemplo 1 usando Matlab, Maple o Mathematica.
13. En el ejemplo 3, compruebe las factorizaciones de la matriz \mathbf{A} usando Matlab, Maple y Mathematica.
14. Encuentre la factorización $\mathbf{PA} = \mathbf{LU}$ de esta matriz:

$$\mathbf{A} = \begin{bmatrix} -0.05811 & -0.11696 & 0.51004 & -0.31330 \\ -0.04291 & 0.56850 & 0.07041 & 0.68747 \\ -0.01652 & 0.38953 & 0.01203 & -0.52927 \\ -0.06140 & 0.32179 & -0.22094 & 0.42448 \end{bmatrix}$$

que fue estudiada por Wilkinson [1965, p. 640].

8.2 Soluciones iterativas de sistemas lineales

En esta sección se analiza una estrategia completamente diferente para resolver un sistema lineal singular

$$\mathbf{Ax} = \mathbf{b} \tag{1}$$

Este método alternativo se utiliza a menudo en problemas enormes que surgen en la solución numérica de ecuaciones diferenciales parciales. En este tema, los sistemas con cientos de miles de ecuaciones se presentan de forma rutinaria.

Normas de vector y matriz

Para empezar, presentamos una breve visión general de las normas de vector y de matriz, ya que son útiles en el análisis de errores y en los criterios de paro de los métodos iterativos. Las normas se pueden definir en cualquier espacio vectorial, pero por lo general se usa \mathbb{R}^n o \mathbb{C}^n . Una norma vec-

torial $\|x\|$ se puede pensar como el **tamaño** o **magnitud** de un vector $x \in \mathbb{R}^n$. Una **norma vectorial** es cualquier mapeo de \mathbb{R}^n a \mathbb{R} que obedece estas tres propiedades:

$$\begin{aligned}\|x\| &> 0 \text{ si } x \neq \mathbf{0} \\ \|\alpha x\| &= |\alpha| \|x\| \\ \|x + y\| &\leq \|x\| + \|y\| \quad (\text{desigualdad del triángulo})\end{aligned}$$

para vectores $x, y \in \mathbb{R}^n$ y escalares $\alpha \in \mathbb{R}$. Algunos ejemplos de normas vectoriales para el vector $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ son

$$\begin{aligned}\|x\|_1 &= \sum_{i=1}^n |x_i| && \text{norma vectorial } \ell_1 \\ \|x\|_2 &= \left(\sum_{i=1}^n x_i^2 \right)^{1/2} && \text{euclíadiana / norma vectorial } \ell_2 \\ \|x\|_\infty &= \max_{1 \leq i \leq n} |x_i| && \text{norma vectorial } \ell_\infty\end{aligned}$$

Para matrices de $n \times n$, podemos también tener **normas matriciales**, sujetas a los mismos requisitos:

$$\begin{aligned}\|A\| &> 0 \text{ si } A \neq 0 \\ \|\alpha A\| &= |\alpha| \|A\| \\ \|A + B\| &\leq \|A\| + \|B\| \quad (\text{desigualdad del triángulo})\end{aligned}$$

para matrices A, B y escalares α .

En general se prefieren normas matriciales que estén relacionadas con una norma vectorial.

Para una norma vectorial $\|\cdot\|$, la **norma matricial subordinada** se define por

$$\|A\| \equiv \sup \{ \|Ax\| : x \in \mathbb{R}^n \text{ y } \|x\| = 1 \}$$

Aquí, A es una matriz de $n \times n$. Para una norma matricial subordinada, algunas propiedades adicionales son

$$\begin{aligned}\|I\| &= 1 \\ \|Ax\| &\leq \|A\| \|x\| \\ \|AB\| &\leq \|A\| \|B\|\end{aligned}$$

Hay dos significados asociados con la notación $\|\cdot\|_p$, uno para los vectores y otro para las matrices. El contexto determina cuál de ellas se intenta. Ejemplos de normas matriciales subordinadas de una matriz A de $n \times n$ son

$$\begin{aligned}\|A\|_1 &= \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| && \text{norma matricial } \ell_1 \\ \|A\|_2 &= \max_{1 \leq i \leq n} \sqrt{\sigma_{\max}} && \text{espectral / norma matricial } \ell_2 \\ \|A\|_\infty &= \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| && \text{norma matricial } \ell_\infty\end{aligned}$$

Aquí, σ_i son los valores propios $A^T A$, que se llaman los **valores singulares** de A . El σ_{\max} más grande en valor absoluto se denomina **radiopectral** de A . (Véase en la sección 8.3 un análisis de valores singulares.)

Número de condición y mal condicionado

Una cantidad importante que tiene cierta influencia en la solución numérica de un sistema lineal $\mathbf{A}\mathbf{x} = \mathbf{b}$ es el **número de condición**, que se define como

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2$$

Resulta que no es necesario calcular la inversa de \mathbf{A} para obtener una estimación del número de condición. Además, se puede demostrar que el número de condición $\kappa(\mathbf{A})$ mide la transferencia de error de la matriz \mathbf{A} y el vector \mathbf{b} a la solución \mathbf{x} . La regla de oro es que si $\kappa(\mathbf{A}) = 10^k$, entonces se puede esperar perder al menos k dígitos de precisión en la solución del sistema $\mathbf{A}\mathbf{x} = \mathbf{b}$. Si el sistema lineal es sensible a las perturbaciones en los elementos de \mathbf{A} , o a las perturbaciones de los componentes de \mathbf{b} , entonces este hecho se refleja en \mathbf{A} , que tendrá un gran número de condición. En tal caso, se dice que la matriz \mathbf{A} está **mal condicionada**. En pocas palabras, cuanto mayor sea el número de condición, más mal condicionado está el sistema.

Supongamos que queremos resolver un sistema lineal invertible de ecuaciones $\mathbf{A}\mathbf{x} = \mathbf{b}$ para una matriz de coeficientes dada \mathbf{A} y del lado derecho \mathbf{b} pero puede haber perturbaciones de los datos debido a la incertidumbre en las mediciones y a los errores de redondeo en los cálculos. Supongamos que el lado derecho es perturbado por una cantidad asignada al símbolo $\delta\mathbf{b}$ y la solución correspondiente es perturbada por la cantidad denotada por el símbolo $\delta\mathbf{x}$. Entonces tenemos

$$\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{A}\delta\mathbf{x} = \mathbf{b} + \delta\mathbf{b}$$

donde

$$\mathbf{A}\delta\mathbf{x} = \delta\mathbf{b}$$

Del sistema lineal original $\mathbf{A}\mathbf{x} = \mathbf{x}$ y las normas, tenemos

$$\|\mathbf{b}\| = \|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\| \|\mathbf{x}\|$$

lo que nos da

$$\frac{1}{\|\mathbf{x}\|} \leq \frac{\|\mathbf{A}\|}{\|\mathbf{b}\|}$$

Del sistema lineal perturbado $\mathbf{A}\delta\mathbf{x} = \delta\mathbf{b}$, obtenemos $\delta\mathbf{x} = \mathbf{A}^{-1}\delta\mathbf{b}$ y

$$\|\delta\mathbf{x}\| \leq \|\mathbf{A}^{-1}\| \|\delta\mathbf{b}\|$$

Combinando las dos desigualdades anteriores obtenemos

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \kappa(\mathbf{A}) \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}$$

que contiene el número de condición de la matriz original \mathbf{A} .

Como un ejemplo de matriz mal condicionada considere la matriz de Hilbert

$$H_3 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

Podemos usar comandos de Matlab para generar la matriz y luego calcular tanto el número de condición utilizando la norma 2 y el determinante de la matriz. Encontramos el número de condición 524.0568 y el determinante 4.6296×10^{-4} . En la solución de sistemas lineales, el número de

condición de la matriz de coeficientes mide la sensibilidad del sistema a los errores en los datos. Cuando el número de condición es grande, ¡la solución calculada del sistema puede correr peligro de error! Se deben hacer más verificaciones antes de aceptar la solución como correcta. Los valores del número de condición, cercanos a 1 indican una matriz bien condicionada, mientras que los valores grandes indican una matriz mal condicionada. Utilizar el determinante para comprobar la singularidad es apropiado solamente para matrices de tamaño pequeño. Usando software matemático se puede calcular el número de condición para comprobar la singularidad o casi singularidad de las matrices.

Un objetivo en el estudio de los métodos numéricos es adquirir conciencia de que un resultado numérico puede ser confiable o puede ser sospechoso (y por tanto la necesidad de análisis más profundo). El número de condición proporciona cierta evidencia sobre este tema. Con la introducción de complejos sistemas de software matemático como Matlab y otros, con frecuencia está disponible una estimación del número de condición, junto con una solución aproximada para que se pueda juzgar la credibilidad de los resultados. De hecho, algunos procedimientos de solución implican características avanzadas que dependen de una estimación y puede cambiar las técnicas de solución basadas en ésta. Por ejemplo, este criterio puede dar como resultado un cambio de la técnica de solución de una variante de la eliminación gaussiana a una solución de mínimos cuadrados para un sistema mal condicionado. Los usuarios desprevenidos pueden no darse cuenta de que esto ha sucedido, a menos que vean todos los resultados, incluida la estimación del número de condición. (Los números de condición también pueden estar asociados con otros problemas numéricos, tales como la localización de raíces de ecuaciones.)

Métodos iterativos básicos

La estrategia del método iterativo produce una sucesión de vectores solución aproximados $x^{(0)}, x^{(1)}, x^{(2)}, \dots$ para el sistema $Ax = b$. El procedimiento numérico está diseñado de manera que, en principio, la sucesión de vectores converge a la solución real. El proceso se puede detener cuando se ha alcanzado la suficiente precisión. Esto está en contraste con el algoritmo de eliminación gaussiana, que no tiene ninguna disposición para detenerse a medio camino y ofrecer una solución aproximada. Un algoritmo iterativo general para resolver el sistema (1) es el siguiente. Seleccione una matriz no singular Q y elija un vector inicial arbitrario $x^{(0)}$, genere los vectores $x^{(1)}, x^{(2)}, \dots$ recursivamente de la ecuación

$$Qx^{(k)} = (Q - A)x^{(k-1)} + b \quad (k = 1, 2, \dots) \quad (2)$$

Para ver que esto es sensible, suponga que la sucesión $x^{(k)}$ converge a un vector x^* . Luego, obteniendo el límite cuando $k \rightarrow \infty$ en el sistema (2) obtenemos

$$Qx^* = (Q - A)x^* + b$$

Esto conduce a $Ax^* = b$. Así, si la sucesión converge, su límite es una solución del sistema original (1). Por ejemplo, la **iteración de Richardson** utiliza $Q = I$.

Una descripción del pseudocódigo para realizar el procedimiento iterativo general (2) es la siguiente:

```

integer k, kmax
real array (x(0))1:n, (b)1:n, (c)1:n, (x)1:n, (y)1:n, (A)1:n×1:n, (Q)1:n×1:n
x ← x(0)
for k = 1 to kmax do
```

```

 $y \leftarrow x$ 
 $c \leftarrow (Q - A)x + b$ 
 $\text{solve } Qx = c$ 
 $\text{output } k, x$ 
 $\text{if } \|x - y\| < \varepsilon \text{ then}$ 
     $\text{output "convergencia"}$ 
     $\text{stop}$ 
 $\text{end if}$ 
 $\text{end for}$ 
 $\text{output "se alcanzó la iteración máxima"}$ 

```

Al elegir la matriz no singular Q estamos influenciados por las siguientes consideraciones:

- El sistema (2) debe ser *fácil* de resolver para $x^{(k)}$, cuando se conoce el lado derecho.
- La matriz Q se debe elegir de manera que garantice que la sucesión $x^{(k)}$ converge, sin importar qué vector inicial se utilice. Idealmente, esta convergencia será rápida.

Uno no debe creer en la necesidad de calcular la inversa de Q para realizar un procedimiento iterativo. En los sistemas pequeños se puede calcular fácilmente la inversa de Q , pero en general, esto definitivamente no se hace! Queremos resolver un sistema lineal en el que Q sea la matriz de coeficientes. Como se mencionó anteriormente, queremos elegir Q de manera que un sistema lineal con Q como la matriz de coeficientes sea fácil de resolver. Ejemplos de estas matrices son diagonal, tridiagonal, en banda, triangular inferior y triangular superior.

Ahora, vamos a ver el sistema (1) en su forma detallada

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (1 \leq i \leq n) \quad (3)$$

Resolviendo la i -ésima ecuación para el i -ésimo término, obtenemos una ecuación que describe el **método de Jacobi**:

$$x_i^{(k)} = \left[- \sum_{\substack{j=1 \\ j \neq i}}^n (a_{ij}/a_{ii})x_j^{(k-1)} + (b_i/a_{ii}) \right] \quad (1 \leq i \leq n) \quad (4)$$

En este caso, suponemos que todos los elementos de la diagonal son distintos de cero. (si no es así, en general se pueden reordenar las ecuaciones de modo que lo sea).

En el método de Jacobi anterior, las ecuaciones se resuelven en orden. Los componentes de $x_j^{(k-1)}$ y los correspondientes nuevos valores $x_j^{(k)}$ se pueden utilizar inmediatamente en su lugar. Si se hace esto, tenemos el **método de Gauss-Seidel**:

$$x_i^{(k)} = \left[- \sum_{\substack{j=1 \\ j < i}}^n (a_{ij}/a_{ii})x_j^{(k)} - \sum_{\substack{j=1 \\ j > i}}^n (a_{ij}/a_{ii})x_j^{(k-1)} + (b_i/a_{ii}) \right] \quad (5)$$

Si $x^{(k-1)}$ no se ha guardado, entonces podemos prescindir de los exponentes en el pseudocódigo como sigue:

```

integer i, j, k, kmax, n; real array (aij)1:n × 1:n, (bi)1:n, (xi)1:n
for k = 1 to kmax do
    for i = 1 to n do
        xi  $\leftarrow \left[ b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j \right] / a_{ii}$ 
    end for
end for

```

Una aceleración del método de Gauss-Seidel es posible por la introducción de un factor de relajación ω , lo que da como resultado el **método de sobrerrelajación sucesiva (SOR)**:

$$x_i^{(k)} = \omega \left\{ \left[- \sum_{\substack{j=1 \\ j < i}}^n (a_{ij}/a_{ii}) x_j^{(k)} - \sum_{\substack{j=1 \\ j > i}}^n (a_{ij}/a_{ii}) x_j^{(k-1)} + (b_i/a_{ii}) \right] \right\} + (1 - \omega) x_i^{(k-1)} \quad (6)$$

El método SOR con $\omega = 1$ se reduce al método de Gauss-Seidel.

Ahora consideraremos ejemplos numéricos usando los métodos iterativos asociados con los nombres Jacobi, Gauss-Seidel y sobrerrelajación sucesiva.

EJEMPLO 2 (Iteración de Jacobi) Sea

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$$

Realice una serie de iteraciones de la iteración de Jacobi, comenzando con el vector inicial cero.

Solución Reescribiendo las ecuaciones, tenemos el método de Jacobi:

$$\begin{aligned} x_1^{(k)} &= \frac{1}{2} x_2^{(k-1)} + \frac{1}{2} \\ x_2^{(k)} &= \frac{1}{3} x_1^{(k-1)} + \frac{1}{3} x_3^{(k-1)} + \frac{8}{3} \\ x_3^{(k)} &= \frac{1}{2} x_2^{(k-1)} - \frac{5}{2} \end{aligned}$$

Tomando el vector inicial para $x^{(0)} = [0, 0, 0]^T$, encontramos (con la ayuda de un programa de computadora o de una calculadora programable) que

$$\begin{aligned} x^{(0)} &= [0, 0, 0]^T \\ x^{(1)} &= [0.5000, 2.6667, -2.5000]^T \\ x^{(2)} &= [1.8333, 2.0000, -1.1667]^T \\ &\vdots \\ x^{(21)} &= [2.0000, 3.0000, -1.0000]^T \end{aligned}$$

Se obtiene la solución real (a cuatro decimales redondeados). ■

En la iteración de Jacobi, \mathbf{Q} se toma como la diagonal de \mathbf{A} :

$$\mathbf{Q} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Ahora

$$\mathbf{Q}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}, \quad \mathbf{Q}^{-1}\mathbf{A} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}$$

La matriz iterativa de Jacobi y el vector constante son

$$\mathbf{B} = \mathbf{I} - \mathbf{Q}^{-1}\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 \end{bmatrix}, \quad \mathbf{h} = \mathbf{Q}^{-1}\mathbf{b} = \begin{bmatrix} \frac{1}{2} \\ \frac{8}{3} \\ -\frac{5}{2} \end{bmatrix}$$

Se puede ver que \mathbf{Q} es cercana a \mathbf{A} , $\mathbf{Q}^{-1}\mathbf{A}$ está cerca de \mathbf{I} e $\mathbf{I} = \mathbf{Q}^{-1}\mathbf{A}$ es pequeña. Escribimos el método de Jacobi como

$$\mathbf{x}^{(k)} = \mathbf{B}\mathbf{x}^{(k-1)} + \mathbf{h}$$

EJEMPLO 2 (**Iteración de Gauss-Seidel**) Repita el ejemplo anterior utilizando la iteración de *Gauss-Seidel*.

Solución La idea de la iteración de *Gauss-Seidel* es simplemente acelerar la convergencia mediante la incorporación de cada vector en cuanto se ha calculado. Obviamente, sería más eficiente en el método de Jacobi para utilizar el valor actualizado $x_1^{(k)}$ en la segunda ecuación en lugar del valor viejo $x_1^{(k-1)}$. Del mismo modo, $x_2^{(k)}$ se podría utilizar en la tercera ecuación, en lugar de $x_2^{(k-1)}$. Utilizando la nueva iteración en cuanto estén disponibles, tenemos el método de *Gauss-Seidel*:

$$\begin{aligned} x_1^{(k)} &= \frac{1}{2}x_2^{(k-1)} + \frac{1}{2} \\ x_2^{(k)} &= \frac{1}{3}x_1^{(k)} + \frac{1}{3}x_3^{(k-1)} + \frac{8}{3} \\ x_3^{(k)} &= \frac{1}{2}x_2^{(k)} - \frac{5}{2} \end{aligned}$$

Comenzando con el vector cero inicial, algunas de las iteraciones son

$$\begin{aligned} \mathbf{x}^{(0)} &= [0, 0, 0]^T \\ \mathbf{x}^{(1)} &= [0.5000, 2.8333, -1.0833]^T \\ \mathbf{x}^{(2)} &= [1.9167, 2.9444, -1.0278]^T \\ &\vdots \\ \mathbf{x}^{(9)} &= [2.0000, 3.0000, -1.0000]^T \end{aligned}$$

En este ejemplo, la convergencia del método de *Gauss-Seidel* es aproximadamente del doble de rápida que la del método de Jacobi.

En el algoritmo iterativo que lleva el nombre de *Gauss-Seidel*, \mathbf{Q} es elegida como la parte triangular inferior de \mathbf{A} , incluida la diagonal. Utilizando los datos del ejemplo anterior, ahora en-

contramos que

$$\mathbf{Q} = \begin{bmatrix} 2 & 0 & 0 \\ -1 & 3 & 0 \\ 0 & -1 & 2 \end{bmatrix}$$

Las operaciones usuales de renglón nos dan

$$\mathbf{Q}^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{2} \end{bmatrix}, \quad \mathbf{Q}^{-1} \mathbf{A} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & \frac{5}{6} & -\frac{1}{3} \\ 0 & -\frac{1}{12} & \frac{5}{6} \end{bmatrix}$$

Una vez más, enfatizamos que en un problema práctico no se podría calcular \mathbf{Q}^{-1} . La matriz iterativa de Gauss-Seidel y el vector constante son

$$\mathcal{L} = \mathbf{I} - \mathbf{Q}^{-1} \mathbf{A} = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} \\ 0 & \frac{1}{12} & \frac{1}{6} \end{bmatrix}, \quad \mathbf{h} = \mathbf{Q}^{-1} \mathbf{b} = \begin{bmatrix} \frac{1}{2} \\ \frac{17}{6} \\ -\frac{13}{12} \end{bmatrix}$$

Escribimos el método de Gauss-Seidel como

$$\mathbf{x}^{(k)} = \mathcal{L} \mathbf{x}^{(k-1)} + \mathbf{h}$$

EJEMPLO 3 (Iteración SOR)

Repita el ejemplo anterior usando la iteración SOR con $\omega = 1.1$.

Solución Introduciendo un factor de relajación ω en el método de Gauss-Seidel, tenemos el método SOR:

$$\begin{aligned} x_1^{(k)} &= \omega \left[\frac{1}{2} x_2^{(k-1)} + \frac{1}{2} \right] + (1 - \omega) x_1^{(k-1)} \\ x_2^{(k)} &= \omega \left[\frac{1}{3} x_1^{(k)} + \frac{1}{3} x_3^{(k-1)} + \frac{8}{3} \right] + (1 - \omega) x_2^{(k-1)} \\ x_3^{(k)} &= \omega \left[\frac{1}{2} x_2^{(k)} - \frac{5}{2} \right] + (1 - \omega) x_3^{(k-1)} \end{aligned}$$

Comenzando con el vector inicial de ceros y con $\omega = 1.1$, algunas iteraciones son

$$\begin{aligned} \mathbf{x}^{(0)} &= [0, 0, 0]^T \\ \mathbf{x}^{(1)} &= [0.5500, 3.1350, -1.0257]^T \\ \mathbf{x}^{(2)} &= [2.2193, 3.0574, -0.9658]^T \\ &\vdots \\ \mathbf{x}^{(7)} &= [2.0000, 3.0000, -1.0000]^T \end{aligned}$$

En este ejemplo, la convergencia del método SOR es más rápida que la del método de Gauss-Seidel.

En el algoritmo iterativo que lleva el nombre de sobrerrelajación sucesiva (SOR), \mathbf{Q} se elige como la parte triangular inferior de \mathbf{A} , incluida la diagonal, pero cada elemento diagonal a_{ij} se sustituye por a_{ij}/ω , donde ω es el así llamado **factor de relajación**. (El trabajo inicial sobre el método SOR fue realizado por Southwell [1946] y por Young [1950].) Del ejemplo anterior,

esto significa que

$$Q = \begin{bmatrix} \frac{20}{11} & 0 & 0 \\ -1 & \frac{30}{11} & 0 \\ 0 & -1 & \frac{20}{11} \end{bmatrix}$$

Ahora

$$Q^{-1} = \begin{bmatrix} \frac{11}{20} & 0 & 0 \\ \frac{121}{600} & \frac{11}{30} & 0 \\ \frac{1331}{12000} & \frac{121}{600} & \frac{11}{20} \end{bmatrix}, \quad Q^{-1} A = \begin{bmatrix} \frac{11}{10} & -\frac{11}{20} & 0 \\ \frac{11}{300} & \frac{539}{600} & -\frac{11}{30} \\ \frac{121}{6000} & \frac{671}{12000} & \frac{539}{600} \end{bmatrix}$$

La matriz iterativa SOR y el vector constante son

$$\mathcal{L}_\omega = I - Q^{-1} A = \begin{bmatrix} -\frac{1}{10} & \frac{11}{20} & 0 \\ -\frac{11}{300} & \frac{61}{600} & \frac{11}{30} \\ -\frac{121}{6000} & -\frac{671}{12000} & \frac{61}{600} \end{bmatrix}, \quad h = Q^{-1} b = \begin{bmatrix} \frac{11}{20} \\ \frac{627}{200} \\ -\frac{4103}{4000} \end{bmatrix}$$

Escribimos el método SOR como

$$x^{(k)} = \mathcal{L}_\omega x^{(k-1)} + h$$

Seudocódigo

Podemos escribir el seudocódigo para los métodos de Jacobi, Gauss-Seidel y SOR suponiendo que el sistema lineal (1) se almacena en forma de matriz-vector:

```

procedure Jacobi(A, b, x)
real kmax ← 100, δ ← 10-10, ε ← ½ × 10-4
integer i, j, k, kmax, n; real diag, sum
real array (A)1:n×1:n, (b)1:n, (x)1:n, (y)1:n
n ← size(A)
for k = 1 to kmax do
    y ← x
    for i = 1 to n do
        sum ← bi
        diag ← aii
        if |diag| < δ then
            output "elemento de la diagonal muy chico"
            return
        end if
        for j = 1 to n do
            if j ≠ i then
                sum ← sum - aij yj
            end if
        end for
        xi ← sum / diag
    end for
    output k, x
end procedure

```

```

if  $\|x - y\| < \varepsilon$  then
    output  $k, x$ 
    return
end if
end for
output "se alcanzó la iteración máxima"
return
end Jacobi

```

En este caso, el vector y contiene los valores iterados viejos y el vector x contiene los actualizados. Los valores de $kmax$, δ y ε se fijan ya sea en una declaración de parámetros o como variables globales.

El seudocódigo para el procedimiento $Gauss_Seidel(\mathbf{A}, \mathbf{B}, \mathbf{x})$ sería igual que el seudocódigo de Jacobi anterior, salvo el ciclo j más interno se sustituye por lo siguiente:

```

for  $j = 1$  to  $i - 1$  do
     $sum \leftarrow sum - a_{ij}x_j$ 
en or
for  $j = i + 1$  to  $n$  do
     $\leftarrow -a_{ij}x_j$ 
end for

```

El seudocódigo para el procedimiento $SOR(\mathbf{A}, \mathbf{B}, \mathbf{x}, \omega)$ sería igual que el de Gauss-Seidel sustituyendo lo siguiente en el ciclo j :

$$\begin{aligned} x_i &\leftarrow sum / diag \\ x_i &\leftarrow \omega x_i + (1 - \omega)y_i \end{aligned}$$

En la solución de ecuaciones diferenciales parciales, con frecuencia se utilizan los métodos iterativos para resolver grandes sistemas lineales dispersos, que a menudo tienen estructuras especiales. Las derivadas parciales se aproximan mediante plantillas compuestas con relativamente pocos elementos, como 5, 7 o 9. Esto nos lleva a tan sólo unos pocos elementos diferentes de cero por renglón en el sistema lineal. En tales sistemas, la matriz de coeficientes \mathbf{A} generalmente no se almacena, ya que el producto matriz-vector se puede escribir directamente en el código. Véase el capítulo 15 para más detalles acerca de esto y cómo se relaciona con la solución de ecuaciones diferenciales parciales elípticas.

Teoremas de convergencia

Del análisis del método descrito por el sistema (2), escribimos

$$\mathbf{x}^{(k)} = \mathbf{Q}^{-1} [(\mathbf{Q} - \mathbf{A})\mathbf{x}^{(k-1)} + \mathbf{b}]$$

o

$$\mathbf{x}^{(k)} = \mathcal{G}\mathbf{x}^{(k-1)} + \mathbf{h} \quad (7)$$

donde la matriz y el vector de iteración son

$$\mathcal{G} = \mathbf{I} - \mathbf{Q}^{-1}\mathbf{A}, \quad \mathbf{h} = \mathbf{Q}^{-1}\mathbf{b}$$

Observe que en el seudocódigo, *no* se calcula \mathbf{Q}^{-1} . La matriz \mathbf{Q}^{-1} se utiliza para facilitar el análisis. Ahora sea \mathbf{x} la solución de sistema (1). Puesto que \mathbf{A} es no singular, \mathbf{x} existe y es única. Tenemos, de la ecuación (7)

$$\begin{aligned}\mathbf{x}^{(k)} - \mathbf{x} &= (\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A})\mathbf{x}^{(k-1)} - \mathbf{x} + \mathbf{Q}^{-1}\mathbf{b} \\ &= (\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A})\mathbf{x}^{(k-1)} - (\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A})\mathbf{x} \\ &= (\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A})(\mathbf{x}^{(k-1)} - \mathbf{x})\end{aligned}$$

Podemos interpretar $\mathbf{e}^{(k)} \equiv \mathbf{x}^{(k)} - \mathbf{x}$ como el **vector de error** actual. Por lo tanto, tenemos

$$\mathbf{e}^{(k)} = (\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A})\mathbf{e}^{(k-1)} \quad (8)$$

Queremos que $\mathbf{e}^{(k)}$ sea *más pequeño* conforme k aumenta. La ecuación (8) muestra que $\mathbf{e}^{(k)}$ será menor que $\mathbf{e}^{(k-1)}$ si $\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A}$ es *pequeña*, en algún sentido. A su vez, esto significa que $\mathbf{Q}^{-1}\mathbf{A}$ debe estar *cerca de \mathbf{A}* . Así, \mathbf{Q} debe estar *cerca de \mathbf{A}* . (Las normas se pueden utilizar para que *pequeño* y *cercano* sean exactos).

■ TEOREMA 1

Teorema del radiopectral

Con el fin de que la sucesión generada por $\mathbf{Q}\mathbf{x}^{(k)} = (\mathbf{Q} - \mathbf{A})\mathbf{x}^{(k-1)} + \mathbf{b}$ converja, sin importa qué punto de inicio $\mathbf{x}^{(0)}$ se seleccione, es necesario y suficiente que todos los valores propios de $\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A}$ se encuentren en el disco unitario abierto, $|z| < 1$, en el plano complejo.

La conclusión de este teorema también se puede escribir como

$$\rho(\mathbf{I} - \mathbf{Q}^{-1}\mathbf{A}) < 1$$

donde ρ es la función de **radiopectral**: para cualquier matriz \mathbf{G} de $n \times n$, con valores propios λ_i , $\rho(\mathbf{G}) = \max_{1 \leq i \leq n} |\lambda_i|$.

EJEMPLO 4 Determine si los métodos de Jacobi, Gauss-Seidel y SOR (con $\omega = 1.1$) de los ejemplos anteriores convergen para todas las iteraciones iniciales.

Solución Con el método de Jacobi podemos fácilmente calcular los valores propios de la importante matriz \mathbf{B} . Los pasos son

$$\det(\mathbf{B} - \lambda\mathbf{I}) = \det \begin{bmatrix} -\lambda & \frac{1}{2} & 0 \\ \frac{1}{3} & -\lambda & \frac{1}{3} \\ 0 & \frac{1}{2} & -\lambda \end{bmatrix} = -\lambda^3 + \frac{1}{6}\lambda + \frac{1}{6}\lambda = 0$$

Los valores propios son $\lambda = 0, \pm\sqrt{1/3} \approx \pm 0.5774$. Así, por el teorema anterior, la iteración de Jacobi tiene éxito con cualquier vector inicial en este ejemplo.

Por el método de Gauss-Seidel, los valores propios de la iteración de la matriz \mathcal{L} se determinan de

$$\det(\mathcal{L} - \lambda I) = \det \begin{bmatrix} -\lambda & \frac{11}{20} & 0 \\ 0 & \frac{1}{6} - \lambda & \frac{1}{3} \\ 0 & \frac{1}{12} & \frac{1}{6} - \lambda \end{bmatrix} = -\lambda \left(\frac{1}{6} - \lambda \right)^2 + \frac{1}{36}\lambda = 0$$

Los valores propios son $\lambda = 0, 0, \frac{1}{3} \approx 0.333$. Por lo tanto, la iteración de Gauss-Seidel también tendrá éxito para cualquier vector inicial en este ejemplo.

Para el método de SOR con $\omega = 1.1$, los valores propios de la matriz de iteración \mathcal{L}_ω se determinan a partir de

$$\begin{aligned} \det(\mathcal{L}_\omega - \lambda I) &= \det \begin{bmatrix} -\frac{1}{10} - \lambda & \frac{11}{20} & 0 \\ -\frac{11}{300} & \frac{61}{600} - \lambda & \frac{11}{30} \\ -\frac{121}{6000} & \frac{671}{12000} & \frac{61}{600} - \lambda \end{bmatrix} \\ &= \left(-\frac{1}{10} - \lambda \right) \left(\frac{61}{600} - \lambda \right)^2 - \frac{121}{6000} \frac{11}{30} \frac{11}{20} \\ &\quad + \frac{11}{20} \frac{11}{300} \left(\frac{61}{600} - \lambda \right) - \left(-\frac{1}{10} - \lambda \right) \frac{671}{12000} \frac{11}{30} \\ &= -\frac{1}{1000} + \frac{31}{3000} \lambda + \frac{31}{3000} \lambda^2 - \lambda^3 = 0 \end{aligned}$$

Los valores propios son $\lambda \approx 0.1200, 0.0833, -0.1000$. Por lo tanto, la iteración SOR también tendrá éxito para cualquier vector inicial en este ejemplo. ■

Una condición que es más fácil de verificar que la desigualdad $\rho(I - Q^{-1}A) < 1$ es el dominio de los elementos de la diagonal sobre los otros elementos en el mismo renglón. Tal como se define en la sección 7.3, podemos usar la propiedad de **dominio diagonal**

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$$

para determinar si los métodos de Jacobi y de Gauss-Seidel convergen mediante el siguiente teorema.

■ TEOREMA 2

Teorema de convergencia de Jacobi y de Gauss-Seidel

Si A tiene dominio diagonal, entonces los métodos de Jacobi y de Gauss-Seidel convergen para cualquier vector inicial $x^{(0)}$.

Observe que esta es una condición suficiente, pero no una condición necesaria. De hecho, hay matrices que no tienen dominio diagonal para las que convergen estos métodos.

Otra propiedad importante es la siguiente:

■ DEFINICIÓN 1

Simétrica positiva definida

La matriz A es **simétrica positiva definida (SPD)** si $A = A^T$ y $x^T A x > 0$ para todos los vectores reales x distintos de cero.

Para que una matriz A sea SPD, es necesario y suficiente que $A = A^T$ y que todos los valores propios de A sean positivos.

■ TEOREMA 3

Teorema de convergencia SOR

Suponga que la matriz A tiene elementos positivos en diagonal y que $0 < \omega < 2$. El método SOR converge para cualquier vector inicial $x^{(0)}$ si y sólo si A es simétrica y definida positiva.

Formulación matricial

En la teoría formal de los métodos iterativos, se divide la matriz A en la suma de una matriz diagonal D distinta de cero, una matriz estrictamente triangular inferior C_L y una matriz estrictamente triangular superior C_U tal que

$$A = D - C_L - C_U$$

En este caso, $D = \text{diag}(A)$, $C_L = (-a_{ij})_{i>j}$ y $C_U = (-a_{ij})_{i<j}$. El sistema lineal (3) se puede escribir como

$$(D - C_L - C_U)x = b$$

De la ecuación (4), el **método de Jacobi** en forma matriz-vector es

$$Dx^{(k)} = (C_L + C_U)x^{(k-1)} + b$$

Esto corresponde a la ecuación (2) con $Q = \text{diag}(A) = D$. De la ecuación (5), el método de **Gauss-Seidel** será

$$(D - C_L)x^{(k)} = C_Ux^{(k-1)} + b$$

Esto corresponde a la ecuación (2) con $Q = \text{diag}(A) + \text{triangular inferior}(A) = D - C_L$. De la ecuación (6), el **método SOR** se puede escribir como

$$(D - \omega C_L)x^{(k)} = [\omega C_U + (1 - \omega)D]x^{(k-1)} + \omega b$$

Esto corresponde a la ecuación (2) con $Q = (1/\omega)\text{diag}(A) + \text{triangular inferior}(A) = (1/\omega)D - C_L$.

En resumen, la matriz de iteración y el vector constante para los tres métodos iterativos básicos (Jacobi, Gauss-Seidel y SOR) se pueden escribir en términos de esta división. Para el **método de Jacobi**, tenemos $Q = D$, de modo que

$$\begin{aligned} B &= I - Q^{-1}A = D^{-1}(C_L + C_U) \\ h &= Q^{-1}b = D^{-1}b \end{aligned}$$

Para el **método de Gauss-Seidel**, tenemos $Q = D - C_L$, así

$$\begin{aligned} L &= I - Q^{-1}A = (D - C_L)^{-1}C_U \\ h &= Q^{-1}b = (D - C_L)^{-1}b \end{aligned}$$

Para el **método SOR**, tenemos $Q = 1/\omega(D - \omega C_L)$, de

$$\begin{aligned}\mathcal{L}_\omega &= I - Q^{-1}A = (D - \omega C_L)^{-1}[\omega C_U + (1 - \omega)D] \\ h &= Q^{-1}b = \omega(D - \omega C_L)^{-1}b\end{aligned}$$

Otra visión de la sobrerrelajación

En algunos casos, la razón de convergencia del esquema iterativo básico (2) se puede mejorar introduciendo un vector auxiliar y un *parámetro de la aceleración* ω de la siguiente manera:

$$\begin{aligned}Qz^{(k)} &= (Q - A)x^{(k-1)} + b \\ x^{(k)} &= \omega z^{(k)} + (1 - \omega)x^{(k-1)}\end{aligned}$$

o

$$x^{(k)} = \omega\{(I - Q^{-1}A)x^{(k-1)} + Q^{-1}b\} + (1 - \omega)x^{(k-1)}$$

El parámetro ω da una ponderación en favor de los valores actualizados. Cuando $\omega = 1$, este procedimiento se reduce al método iterativo básico, y cuando $1 < \omega < 2$, la razón de convergencia puede mejorar, lo que se llama **sobrerrelajación**. Cuando $Q = D$, tenemos el **método de sobrerrelajación de Jacobi (SRJ)**:

$$x^{(k)} = \omega\{Bx^{(k-1)} + h\} + (1 - \omega)x^{(k-1)}$$

La sobrerrelajación ofrece ventajas especiales cuando se utiliza con el método de Gauss-Seidel de una manera ligeramente diferente:

$$\begin{aligned}Dz^{(k)} &= C_Lx^{(k)} + C_Ux^{(k-1)} + b \\ x^{(k)} &= \omega z^{(k)} + (1 - \omega)x^{(k-1)}\end{aligned}$$

y tenemos el **método SOR**:

$$x^{(k)} = \mathcal{L}_\omega x^{(k-1)} + h$$

Método del gradiente conjugado

El método del gradiente conjugado es uno de los métodos iterativos más populares para resolver sistemas de ecuaciones lineales dispersos. Es particularmente válido para sistemas que surgen en las soluciones numéricas de las ecuaciones diferenciales parciales.

Comenzamos con una breve presentación de definiciones y de la notación asociada. (algo de lo que se presenta con más detalle en el capítulo 16). Suponga que la matriz real A de $n \times n$ es **simétrica**, lo que significa que $A^T = A$. El **producto interno** de dos vectores $u = (u_1, u_2, \dots, u_n)$ y $v = (v_1, v_2, \dots, v_n)$ se puede escribir como $\langle u, v \rangle = u^T v = \sum_{i=1}^n u_i v_i$, que es la suma escalar. Tenga en cuenta que $\langle u, v \rangle = \langle v, u \rangle$. Si u y v son ortogonales entre sí, entonces $\langle u, v \rangle = 0$. Un **producto interno escalar de A** de dos vectores u y v se define como

$$\langle u, v \rangle_A = \langle Au, v \rangle = u^T A^T v$$

Dos vectores distintos de cero u y v son **conjugados** de A si $\langle u, v \rangle_A = 0$. Una matriz de $n \times n$ es **definida positiva** si

$$\langle x, x \rangle_A > 0$$

para todos los vectores distintos de cero $\mathbf{x} \in \mathbb{R}^n$. En general, expresiones tales como $\langle \mathbf{u}, \mathbf{v} \rangle$ y $\langle \mathbf{u}, \mathbf{v} \rangle_A$ se reducen a matrices de 1×1 y se tratan como valores escalares. Una **forma cuadrática** es una función cuadrática escalar de un vector de la forma

$$f(\mathbf{x}) = \frac{1}{2} \langle \mathbf{x}, \mathbf{x} \rangle_A - \langle \mathbf{b}, \mathbf{x} \rangle + c$$

Aquí, A es una matriz, \mathbf{x} y \mathbf{b} son vectores y c es una constante escalar. El **gradiente** de una forma cuadrática

$$\mathbf{f}'(\mathbf{x}) = [\partial f(\mathbf{x})/\partial x_1, \quad \partial f(\mathbf{x})/\partial x_2, \quad \dots, \quad \partial f(\mathbf{x})/\partial x_n]^T$$

Podemos deducir lo siguiente:

$$\mathbf{f}'(\mathbf{x}) = \frac{1}{2} A^T \mathbf{x} + \frac{1}{2} A\mathbf{x} - \mathbf{b}$$

Si A es simétrica, esto se reduce a

$$\mathbf{f}'(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$$

Haciendo el gradiente igual a cero se obtiene el sistema lineal por resolver, $A\mathbf{x} = \mathbf{b}$. Por lo tanto, la solución de $A\mathbf{x} = \mathbf{b}$ es un punto crítico de $f(\mathbf{x})$. Si A es simétrica y definida positiva, entonces $f(\mathbf{x})$ se minimiza por la solución de $A\mathbf{x} = \mathbf{b}$. Por ende, una forma alternativa de resolver el sistema lineal $A\mathbf{x} = \mathbf{b}$ es buscar una \mathbf{x} que minimice a $f(\mathbf{x})$.

Queremos resolver el sistema lineal

$$A\mathbf{x} = \mathbf{b}$$

donde la matriz A de $n \times n$ es simétrica y positiva definida.

Suponga que $\{\mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(k)}\}$ es un conjunto que tiene una sucesión de **vectores de dirección** mutuamente conjugados. Entonces se forma una base para el espacio \mathbb{R}^n . Por lo tanto, podemos expandir el verdadero vector solución \mathbf{x}^* de $A\mathbf{x} = \mathbf{b}$ en una combinación lineal de estos vectores de la base:

$$\mathbf{x}^* = \alpha_1 \mathbf{p}^{(1)} + \alpha_2 \mathbf{p}^{(2)} + \dots + \alpha^{(k)} \mathbf{p}^{(k)} + \dots + \alpha_n \mathbf{p}^{(n)}$$

donde los coeficientes están dados por

$$\alpha_k = \langle \mathbf{p}^{(k)}, \mathbf{b} \rangle / \langle \mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle_A$$

Esto se puede ver como un método directo para resolver el sistema lineal $A\mathbf{x} = \mathbf{b}$: primero encontramos la sucesión de n vectores de dirección conjugada $\mathbf{p}^{(k)}$ y luego calculamos los coeficientes α_k . Sin embargo, en la práctica, este método es poco práctico porque tomaría demasiado tiempo de máquina y almacenamiento.

Por otra parte, si consideramos al método del gradiente conjugado como un método iterativo, entonces podríamos resolver grandes sistemas lineales dispersos en una cantidad razonable de tiempo y de almacenamiento. La clave está en elegir cuidadosamente un pequeño conjunto de los vectores de dirección conjugada $\mathbf{p}^{(k)}$ de modo que no los necesitamos todos para obtener una buena aproximación al vector solución verdadero.

Se comienza con una conjectura inicial $\mathbf{x}^{(0)}$ sobre la verdadera solución \mathbf{x}^* . Podemos suponer sin perder generalidad que $\mathbf{x}^{(0)}$ es el vector cero. La verdadera solución \mathbf{x}^* es también el único minimizador de

$$f(\mathbf{x}) = \frac{1}{2} \langle \mathbf{x}, \mathbf{x} \rangle_A - \langle \mathbf{x}, \mathbf{x} \rangle = \frac{1}{2} \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{x}$$

para $\mathbf{x} \in \mathbb{R}^n$. Esto sugiere tomar el primer vector base $\mathbf{p}(1)$ como el gradiente de f en $\mathbf{x} = \mathbf{x}^{(0)}$, lo que es igual a $-\mathbf{b}$. Los otros vectores en la base son ahora conjugados del gradiente; de aquí el nombre

del *método del gradiente conjugado*. El k -ésimo vector residual es

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$$

El método del gradiente de descenso se mueve en la dirección $\mathbf{r}^{(k)}$. Tome la dirección más cercana al vector gradiente $\mathbf{r}^{(k)}$, insistiendo en que los vectores de dirección $\mathbf{p}^{(k)}$ se conjuguen entre sí. Uniendo todo esto, obtenemos la expresión

$$\mathbf{p}^{(k+1)} = \mathbf{r}^{(k)} - [\langle \mathbf{p}^{(k)}, \mathbf{r}^{(k)} \rangle_A / \langle \mathbf{p}^{(k)}, \mathbf{p}^{(k)} \rangle_A] \mathbf{p}_k$$

Después de algunas simplificaciones se obtiene el algoritmo para resolver el sistema lineal $\mathbf{Ax} = \mathbf{b}$, donde la matriz de coeficientes \mathbf{A} es real, simétrica y definida positiva. El vector de entrada $\mathbf{x}^{(0)}$ es una aproximación inicial de la solución o del vector cero.

En teoría, el método iterativo del gradiente conjugado resuelve un sistema de n ecuaciones lineales a lo más en n pasos, si la matriz \mathbf{A} es simétrica y definida positiva. Por otra parte, el n -ésimo vector iterativo $\mathbf{x}(n)$ es el único que minimiza la función cuadrática $q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}$. Cuando el método del gradiente conjugado fue introducido por Hestenes y Stiefel [1952], el interés inicial se desvaneció una vez que se descubrió que la propiedad de terminación finita no se obtuvo en la práctica. Pero el interés por este método se renovó dos décadas más tarde, cuando se vio como un proceso iterativo por Reid [1971] y otros. En la práctica, la solución de un sistema de ecuaciones lineales se puede encontrar con frecuencia con una precisión satisfactoria en un número de pasos considerablemente menor que el orden del sistema.

Aquí se presenta un pseudocódigo para el **algoritmo del gradiente conjugado**:

```

 $k \leftarrow 0; \mathbf{x} \leftarrow \mathbf{0}; \mathbf{r} \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}; \delta \leftarrow \langle \mathbf{r}, \mathbf{r} \rangle$ 
while ( $\sqrt{\delta} > \varepsilon \sqrt{\langle \mathbf{b}, \mathbf{b} \rangle}$ ) and ( $k < k_{\text{máx}}$ )
     $k \leftarrow k + 1$ 
    if  $k = 1$  then
         $\mathbf{p} \leftarrow \mathbf{r}$ 
    else
         $\beta \leftarrow \delta / \delta_{\text{viejo}}$ 
         $\mathbf{p} \leftarrow \mathbf{r} + \beta \mathbf{p}$ 
    end if
     $\mathbf{w} \leftarrow \mathbf{A}\mathbf{p}$ 
     $\alpha \leftarrow \delta / \langle \mathbf{p}, \mathbf{w} \rangle$ 
     $\mathbf{x} \leftarrow \mathbf{x} + \alpha \mathbf{p}$ 
     $\mathbf{r} \leftarrow \mathbf{r} - \alpha \mathbf{w}$ 
     $\delta_{\text{viejo}} \leftarrow \delta$ 
     $\delta \leftarrow \langle \mathbf{r}, \mathbf{r} \rangle$ 
end while
```

Aquí, ε es un parámetro utilizado en el criterio de convergencia (por ejemplo, $\varepsilon = 10^{-5}$) y $k_{\text{máx}}$ es el número máximo de iteraciones permitido. Por lo general, el número de iteraciones necesarias es mucho menor que el tamaño del sistema lineal. Guardamos el valor anterior de δ en la variable δ_{viejo} . Si se conoce una buena estimación para el vector solución \mathbf{x} , entonces se debe utilizar como un vector inicial en lugar de cero. La variable ε es la tolerancia de la convergencia deseada. El algoritmo produce no sólo una sucesión de vectores $\mathbf{x}^{(i)}$ que converge a la solución, sino también una sucesión ortogonal de vectores residuales $\mathbf{r}^{(i)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(i)}$ y una sucesión de \mathbf{A} ortogonal de vectores de dirección

de búsqueda $\mathbf{p}^{(i)}$, a saber, $\langle \mathbf{r}^{(i)}, \mathbf{r}^{(j)} \rangle = 0$ si $i \neq j$ y $\langle \mathbf{p}^{(i)}, \mathbf{A} \mathbf{p}^{(j)} \rangle = 0$ si $i \neq j$. Las principales características de cálculo del algoritmo del gradiente conjugado son complicadas de deducir, pero la conclusión final es que en cada paso sólo se requiere *una* multiplicación matriz-vector y sólo se calculan unos pocos productos punto. Estos son atributos muy deseables en la solución de sistemas lineales grandes y dispersos. Además, a diferencia de la eliminación gaussiana, no se llenan, de modo que sólo se necesitan almacenar los elementos diferentes de cero en \mathbf{A} en la memoria de la computadora. Para algunos problemas de ecuaciones diferenciales parciales, las ecuaciones del sistema lineal se pueden representar mediante plantillas de símbolos que describen el factor de estructura diferente de cero dentro de la matriz de coeficientes. A veces, estas plantillas se utilizan en un programa de cómputo en lugar de guardar los elementos diferentes de cero en la matriz de coeficientes.

EJEMPLO 5 Use el método del gradiente conjugado para resolver este sistema lineal:

$$\begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ -5 \end{bmatrix}$$

Solución Programando el seudocódigo, obtenemos las iteraciones

$$\begin{aligned} \mathbf{x}^{(0)} &= [0.00000, 0.00000, 0.00000]^T \\ \mathbf{x}^{(1)} &= [0.29221, 2.33766, -1.46108]^T \\ \mathbf{x}^{(2)} &= [1.82254, 2.60772, -1.55106]^T \\ \mathbf{x}^{(3)} &= [2.00000, 3.00000, -1.00000]^T \end{aligned}$$

Con sólo tres iteraciones, tenemos la respuesta exacta con precisión total de máquina, lo que muestra la propiedad de terminación finita. La matriz \mathbf{A} es simétrica definida positiva y los valores propios de \mathbf{A} son 1, 2, 4. Este ejemplo simple puede ser un poco engañoso, porque uno no puede esperar una convergencia tan rápida en aplicaciones realistas. (La razón de convergencia depende de varias propiedades del sistema lineal.) De hecho, el ejemplo anterior es muy pequeño para mostrar la potencia de los avanzados métodos iterativos en sistemas muy grandes y dispersos. ■

El método del gradiente conjugado puede converger lentamente cuando la matriz \mathbf{A} está mal condicionada; sin embargo, la convergencia se puede acelerar con una técnica llamada de **precondicionamiento**. Implica una matriz \mathbf{M}^{-1} que aproxima a \mathbf{A} de tal forma que $\mathbf{M}^{-1}\mathbf{A}$ está bien condicionada y $\mathbf{M}\mathbf{x} = \mathbf{y}$ es fácil de resolver. Para muchos sistemas lineales muy grandes y dispersos, los métodos del gradiente conjugado precondicionados ¡se han convertido en los métodos iterativos de elección! Para más detalles, véase Golub y Van Loan [1996], así como muchos otros libros académicos comunes y sus referencias.

Resumen

(1) Para el sistema lineal

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

la forma general de un método iterativo es

$$\mathbf{x}^{(k)} = \mathcal{G}\mathbf{x}^{(k-1)} + \mathbf{h}$$

donde la matriz y el vector de iteración son

$$\mathcal{G} = \mathbf{I} - \mathbf{Q}^{-1} \mathbf{A} \quad \mathbf{h} = \mathbf{Q}^{-1} \mathbf{b}$$

El vector de error es

$$\mathbf{e}^{(k)} = (\mathbf{I} - \mathbf{Q}^{-1} \mathbf{A}) \mathbf{e}^{(k-1)}$$

(2) En concreto, consideramos el sistema lineal en forma

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad (1 \leq i \leq n)$$

El método de Jacobi es

$$x_i^{(k)} = \sum_{\substack{j=1 \\ j \neq i}}^n (-a_{ij}/a_{ii}) x_j^{(k-1)} - (b_i/a_{ii}) \quad (1 \leq i \leq n)$$

suponiendo que $a_{ii} \neq 0$. El método de Gauss-Seidel es

$$x_i^{(k)} = \sum_{\substack{j=1 \\ j < i}}^n (-a_{ij}/a_{ii}) x_i^{(k)} + \sum_{\substack{j=1 \\ j > i}}^n (-a_{ij}/a_{ii}) x_j^{(k-1)} - (b_i/a_{ii})$$

El método SOR es

$$x_i^{(k)} = \omega \left\{ \sum_{\substack{j=1 \\ j < i}}^n (-a_{ij}/a_{ii}) x_i^{(k)} + \sum_{\substack{j=1 \\ j > i}}^n (-a_{ij}/a_{ii}) x_j^{(k-1)} - (b_i/a_{ii}) \right\} + (1 - \omega) x_i^{(k-1)}$$

El método SOR se reduce al método de Gauss-Seidel, cuando $\omega = 1$.

(3) Para una formulación matricial, dividimos la matriz \mathbf{A} :

$$\mathbf{A} = \mathbf{D} - \mathbf{C}_L - \mathbf{C}_U$$

donde \mathbf{D} es una matriz diagonal diferente de cero, \mathbf{C}_L es una matriz estrictamente triangular inferior y \mathbf{C}_U es una matriz estrictamente triangular superior. En este caso, $\mathbf{D} = \text{diag}(\mathbf{A})$, $\mathbf{C}_L = (-a_{ij})_{i>j}$ y $\mathbf{C}_U = (-a_{ij})_{i<j}$. El **método de Jacobi** en una forma matriz-vector es

$$\mathbf{D}\mathbf{x}^{(k)} = (\mathbf{C}_L + \mathbf{C}_U)\mathbf{x}^{(k-1)} + \mathbf{b}$$

puesto que $\mathbf{Q} = \mathbf{D}$. El **método de Gauss-Seidel** es

$$(\mathbf{D} - \mathbf{C}_L)\mathbf{x}^{(k)} = \mathbf{C}_U\mathbf{x}^{(k-1)} + \mathbf{b}$$

ya que $\mathbf{Q} = \mathbf{D} - \mathbf{C}_L$. El **método SOR** es

$$(\mathbf{D} - \omega \mathbf{C}_L)\mathbf{x}^{(k)} = [\omega \mathbf{C}_U + (1 - \omega) \mathbf{D}]\mathbf{x}^{(k-1)} + \omega \mathbf{b}$$

puesto que $\mathbf{Q} = (1/\omega)\mathbf{D} - \mathbf{C}_L$. Las matrices divididas, las matrices de iteración y los vectores constantes son los siguientes. Para el **método de Jacobi**, tenemos

$$\begin{aligned} \mathbf{Q} &= \mathbf{D} \\ \mathbf{B} &= \mathbf{D}^{-1}(\mathbf{C}_L + \mathbf{C}_U) \\ \mathbf{h} &= \mathbf{D}^{-1}\mathbf{b} \end{aligned}$$

Para el **método de Gauss-Seidel**, tenemos

$$\begin{aligned} Q &= D - C_L \\ \mathcal{L} &= (D - C_L)^{-1} C_U \\ h &= (D - C_L)^{-1} b \end{aligned}$$

Para el **método SOR**, tenemos

$$\begin{aligned} Q &= \frac{1}{\omega}(D - \omega C_L) \\ \mathcal{L}_\omega &= (D - \omega C_L)^{-1} [\omega C_U + (1 - \omega)D] \\ h &= \omega(D - \omega C_L)^{-1} b \end{aligned}$$

(4) Un método iterativo converge para una matriz específica A si y sólo si

$$\rho(I - Q^{-1}A) < 1$$

Si A tiene dominio diagonal, entonces los métodos de Jacobi y de Gauss-Seidel convergen para cualquier $x^{(0)}$. El método SOR converge para $0 < \omega < 2$ y cualquier $x^{(0)}$, si y sólo si A es simétrica y definida positiva con elementos diagonales positivos.

Problemas 8.2

1. Dé una solución alternativa del ejemplo 4.
2. Escriba la fórmula matricial del método de sobrerrelajación de Gauss-Seidel.
3. (Opción múltiple) En la solución de un sistema de ecuaciones $Ax = b$, con frecuencia es conveniente utilizar un método iterativo que genere una sucesión de vectores x , que deben converger a una solución. El proceso se detiene cuando se ha alcanzado la suficiente precisión. Un procedimiento general es obtener $x^{(k)}$ al resolver $Qx^{(k)} = (Q - A)x^{(k-1)} + b$. En este caso, Q es cierta matriz que suele estar de alguna manera conectada a A . El proceso se repite, iniciando con cualquier $x^{(0)}$ supuesta disponible. ¿Qué hipótesis garantiza que el método funciona sin importar qué punto inicial se seleccione?
 - a. $\|Q\| < 1$
 - b. $\|Q\| < 1$
 - c. $\|I - Q\| < 1$
 - d. $\|I - Q^{-1}\| < 1$
 - e. Ninguna de estos.

Sugerencia: el radio espectral es menor o igual que la norma.
4. (Opción múltiple) A partir de la norma de un vector podemos crear una norma de matriz subordinada. ¿Qué relación satisface cada norma de matriz subordinada?
 - a. $\|Ax\| \geq \|A\| \|x\|$
 - b. $\|I\| = 1$
 - c. $\|AB\| \geq \|A\| \|B\|$
 - d. $\|A + B\| \geq \|A\| + \|B\|$
 - e. Ninguna de estos.
5. (Opción múltiple) La condición de dominio diagonal de una matriz A es:
 - a. $|a_{ii}| < \sum_{j=1, j \neq i}^n |a_{ij}|$
 - b. $|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|$
 - c. $|a_{ii}| < \sum_{j=1}^n |a_{ij}|$
 - d. $|a_{ii}| > \sum_{j=1}^n |a_{ij}|$
 - e. Ninguno de estos.

6. (Opción múltiple) Una condición necesaria y suficiente para que la fórmula de iteración estándar $\mathbf{x}^{(k)} = \mathcal{G}\mathbf{x}^{(k-1)} + \mathbf{h}$ para producir una sucesión $\mathbf{x}^{(k)}$ que converja a una solución de la ecuación $(\mathbf{I} - \mathcal{G})\mathbf{x} = \mathbf{h}$ es que:
- El radio espectral de \mathcal{G} sea mayor que 1.
 - La matriz \mathcal{G} tenga dominio diagonal.
 - El radio espectral de \mathcal{G} sea menor que 1.
 - \mathcal{G} sea no singular.
 - Ninguna de estas.
7. (Opción múltiple) Una condición suficiente para que el método de Jacobi converja para el sistema lineal $\mathbf{A}\mathbf{x} = \mathbf{b}$.
- $\mathbf{A} - \mathbf{I}$ tiene dominio diagonal.
 - \mathbf{A} tiene dominio diagonal.
 - \mathcal{G} es no singular.
 - El radio espectral de \mathcal{G} es menor que 1.
 - Ninguna de estas.
8. (Opción múltiple) Una condición suficiente para que el método de Gauss-Seidel funcione en el sistema lineal $\mathbf{A}\mathbf{x} = \mathbf{b}$.
- \mathbf{A} tiene dominio diagonal.
 - $\mathbf{A} - \mathbf{I}$ tiene dominio diagonal.
 - El radio espectral de \mathbf{A} es menor que 1.
 - \mathcal{G} es no singular.
 - Ninguna de estas.
9. (Opción múltiple) Las condiciones necesarias y suficientes para que el método SOR funcione, donde $0 < \omega < 2$, en el sistema lineal $\mathbf{A}\mathbf{x} = \mathbf{b}$ son
- \mathbf{A} tiene dominio diagonal.
 - $\rho(\mathbf{A}) < 1$.
 - \mathbf{A} es simétrica definida positiva.
 - $\mathbf{x}^{(0)} = \mathbf{0}$.
 - Ninguna de estas.

10. La **norma de Frobenius** está dada por $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2}$ que se usa con frecuencia porque es fácil de calcular. Encuentre el valor de esta norma para estas matrices:

$$\text{a. } \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 4 \\ 2 & 1 & 3 \end{bmatrix} \quad \text{b. } \begin{bmatrix} 0 & 0 & 1 & 2 \\ 3 & 0 & 5 & 4 \\ 1 & 1 & 1 & 2 \\ 1 & 3 & 2 & 2 \end{bmatrix} \quad \text{c. } \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 2 & 3 & 4 & 5 & 6 \\ 0 & 1 & 0 & 1 & 0 \\ 3 & 4 & 3 & 4 & 3 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

11. Determine los números de condición $k(\mathbf{A})$ de estas matrices:

$$\text{a. } \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix} \quad \text{b. } \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

c. $\begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

d. $\begin{bmatrix} -2 & -1 & 2 & -1 \\ 1 & 2 & 1 & -2 \\ 2 & -1 & 2 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}$

Problemas de cómputo 8.2

1. Repita varios o todos los ejemplos del 1 al 5 usando el sistema lineal que implique uno de los siguientes pares de matrices de coeficientes y vectores del lado derecho:

a. $A = \begin{bmatrix} 5 & -1 \\ -1 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$

b. $A = \begin{bmatrix} 5 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 7 \\ 4 \\ 5 \end{bmatrix}$

c. $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 6 & -2 \\ 4 & -3 & 8 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 3 \\ 9 \end{bmatrix}$

d. $A = \begin{bmatrix} 7 & 3 & -1 \\ 3 & 8 & 1 \\ -1 & 1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ -4 \\ 2 \end{bmatrix}$

2. Usando los métodos iterativos de Jacobi, de Gauss-Seidel y SOR ($\omega = 1.1$), escriba y ejecute un programa de cómputo para resolver el sistema lineal siguiente (redondeado) con cuatro decimales de exactitud:

$$\begin{bmatrix} 7 & 1 & -1 & 2 & x_1 \\ 1 & 8 & 0 & -2 & x_2 \\ -1 & 0 & 4 & -1 & x_3 \\ 2 & -2 & -1 & 6 & x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ -5 \\ 4 \\ -3 \end{bmatrix}$$

Compare el número de iteraciones necesario en cada caso. *Sugerencia:* la solución exacta es $x = (1, -1, 1, -1)^T$.

3. Usando los métodos iterativos de Jacobi, de Gauss-Seidel y SOR ($\omega = 1.4$), escriba y ejecute un código para resolver el siguiente sistema lineal con cuatro decimales de exactitud:

$$\begin{bmatrix} 7 & 3 & -1 & 2 & x_1 \\ 3 & 8 & 1 & -4 & x_2 \\ -1 & 1 & 4 & -1 & x_3 \\ 2 & -4 & -1 & 6 & x_4 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ -3 \\ 1 \end{bmatrix}$$

Compare el número de iteraciones en cada caso. *Sugerencia:* en este caso, la solución exacta es $x = (-1, 1, -1, 1)^T$.

4. (Continuación) Resuelva el sistema utilizando el método iterativo SOR con valores de $\omega = 1(0.1), 2$. Trace la gráfica del número de iteraciones para la convergencia contra los valores de ω . ¿Qué valor de ω da como resultado la convergencia más rápida?

5. Programe y ejecute los métodos de Jacobi, de Gauss-Seidel y SOR para el sistema del ejemplo 1 usando
- ecuaciones que impliquen la matriz dividida Q .
 - las formulaciones de ecuaciones del ejemplo 4.
 - el seudocódigo que implica la multiplicación matriz-vector.
6. (Continuación) Seleccione uno o más de los sistemas del problema de cómputo 1 y vuelva a ejecutar estos programas.
7. Considere el sistema lineal

$$\begin{bmatrix} 9 & -3 \\ -2 & 8 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ -4 \end{bmatrix}$$

Usando Maple o Matlab, compare la solución usando el método de Jacobi y el método de Gauss-Seidel iniciando con $\mathbf{x}^{(0)} = (0, 0)^T$.

8. (Continuación)
- Cambie la entrada (1, 1) de 9 a 1 de tal forma que la matriz de coeficientes ya no tenga más dominio diagonal y vea si el método de Gauss-Seidel aún funciona. Explique por qué sí o por qué no.
 - Después cambie la entrada (2, 2) de 8 a 1 y también pruébelo. De nuevo explique los resultados.
9. Use el método del gradiente conjugado para resolver este sistema lineal:

$$\begin{bmatrix} 2.0 & -0.3 & -0.2 \\ -0.3 & 2.0 & -0.1 \\ -0.2 & -0.1 & 2.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 5 \\ 3 \end{bmatrix}$$

10. (**Viga de Euler-Bernoulli**) Un modelo simple de una viga flexionada implica la ecuación diferencial de Euler-Bernoulli. Una discretización en diferencias finitas la convierte en un sistema de ecuaciones lineales. Conforme se reduce el tamaño de la discretización, el sistema lineal se hace más grande y más malcondicionado.

- a. Para una viga articulada en ambos extremos obtenemos el siguiente sistema en banda de ecuaciones lineales con un ancho de banda de cinco:

$$\begin{bmatrix} 12 & -6 & \frac{4}{3} & & \\ -4 & 6 & -4 & 1 & \\ 1 & -4 & 6 & -4 & 1 \\ 1 & -4 & 6 & -4 & 1 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ & 1 & -4 & 6 & -4 & 1 \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 6 & -4 & 1 \\ & & & & 1 & -4 & 6 & -4 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_{n-3} \\ y_{n-2} \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_{n-3} \\ b_{n-2} \\ b_{n-1} \\ b_n \end{bmatrix}$$

El lado derecho representa las fuerzas sobre la viga. Establezca el lado derecho de modo que haya una solución conocida, como una flecha en el centro de la viga. Utilizando un proceso

con método iterativo, en varias ocasiones resuelva el sistema permitiendo que n aumente. ¿El error en la solución aumenta conforme se incrementa n ? Use software matemático para calcular el número de condición de la matriz de coeficientes para explicar lo que está sucediendo.

- b.** El sistema lineal de ecuaciones para una **viga en voladizo** con una condición de borde libre en un solo extremo es

$$\left[\begin{array}{cccccc|c} 12 & -6 & \frac{4}{3} & & & & y_1 \\ -4 & 6 & -4 & 1 & & & b_1 \\ 1 & -4 & 6 & -4 & 1 & & b_2 \\ 1 & -4 & 6 & -4 & 1 & & b_3 \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 1 & -4 & 6 & -4 & 1 & & y_{n-3} \\ 1 & -4 & 6 & -4 & 1 & & b_{n-3} \\ 1 & -4 & \frac{93}{25} & \frac{111}{25} & -\frac{43}{25} & 1 & b_{n-2} \\ \frac{12}{25} & \frac{24}{25} & \frac{12}{25} & & & & b_{n-1} \\ \hline & & & & & & y_n \\ & & & & & & b_n \end{array} \right]$$

Repita el experimento numérico para este sistema. Véase Sauer [2006] para más detalles.

- 11.** Considere este sistema lineal disperso:

$$\left[\begin{array}{cccccc|c} 3 & -1 & & & & & \frac{1}{2} & x_1 \\ -1 & 3 & -1 & & & & & x_2 \\ -1 & 3 & -1 & & & & & x_3 \\ \ddots & \ddots & \ddots & \ddots & & & \vdots & \vdots \\ -1 & 3 & -1 & & & & \vdots & 1.0 \\ \ddots & \ddots & \ddots & \ddots & & & \vdots & \vdots \\ \frac{1}{2} & \frac{1}{2} & -1 & 3 & -1 & & x_{n-2} & 1.5 \\ \frac{1}{2} & -1 & 3 & -1 & & & x_{n-1} & 1.5 \\ \hline & & -1 & 3 & -1 & & x_n & 1.5 \end{array} \right]$$

La solución verdadera es $x = [1, 1, 1, \dots, 1, 1, 1]^T$. Use un método iterativo para resolver este sistema para valores de n más grandes.

- 12.** Considere la muestra del sistema lineal bidimensional $\mathbf{A}\mathbf{x} = \mathbf{b}$, donde $\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} 2 \\ -8 \end{bmatrix}$ y $\mathbf{c} = 0$. Trace las gráficas para mostrar lo siguiente:

- a.** La solución se encuentra en la intersección de dos rectas.
- b.** Trace la gráfica de la forma cuadrática $F(\mathbf{x}) = \mathbf{c} + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$ mostrando que el punto mínimo de esta superficie es la solución de $\mathbf{A}\mathbf{x} = \mathbf{b}$.
- c.** Los contornos de la forma cuadrática son tales que cada curva elíptoidal tiene un valor constante.
- d.** El gradiente $\nabla F(\mathbf{x})$ de la forma cuadrática. Muestre que para cada \mathbf{x} , los puntos del gradiente en la dirección de aumento más pronunciado de $F(\mathbf{x})$ y es ortogonal en las líneas de contorno (véase la sección 16.2).

8.3 Valores propios y vectores propios

Sea A una matriz de $n \times n$. Nos formulamos la siguiente pregunta natural acerca de A : ¿hay vectores v distintos de cero para los que Av sea un múltiplo escalar de v ? A pesar de que nos hemos planteado esta cuestión por pura curiosidad, hay muchas situaciones en el cálculo científico en las que surge esta pregunta.

La respuesta a nuestra pregunta es un *sí!* Debemos estar dispuestos a considerar escalares complejos, así como vectores con componentes complejas. Con esa ampliación de nuestro punto de vista, esos vectores siempre existen. He aquí dos ejemplos. En el primero, no tenemos que usar los números complejos para mostrar la situación, mientras que en el segundo, los vectores y factores escalares deben ser complejos.

EJEMPLO 1 Sea $A = \begin{bmatrix} 3 & 2 \\ 7 & -2 \end{bmatrix}$. Encuentre un vector v diferente de cero para el cual Av sea un múltiplo de v .

Solución Se puede fácilmente comprobar que

$$\begin{aligned} A \begin{bmatrix} 1 \\ 1 \end{bmatrix} &= \begin{bmatrix} 5 \\ 5 \end{bmatrix} = 5 \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ A \begin{bmatrix} 2 \\ -7 \end{bmatrix} &= \begin{bmatrix} -8 \\ 28 \end{bmatrix} = -4 \begin{bmatrix} 2 \\ -7 \end{bmatrix} \end{aligned}$$

Tenemos dos respuestas diferentes (pero no revelamos cómo se encuentran). ■

EJEMPLO 2 Repita el ejemplo anterior con la matriz $A = \begin{bmatrix} 1 & 1 \\ -2 & 3 \end{bmatrix}$.

Solución Como en el ejemplo 1, se puede comprobar que

$$\begin{aligned} A \begin{bmatrix} 1 \\ 1+i \end{bmatrix} &= (2+i) \begin{bmatrix} 1 \\ 1+i \end{bmatrix} \\ A \begin{bmatrix} 1 \\ 1-i \end{bmatrix} &= (2-i) \begin{bmatrix} 1 \\ 1-i \end{bmatrix} \end{aligned}$$

En estas ecuaciones, $i = \sqrt{-1}$. Sorprendentemente, encontramos respuestas que implican números complejos ¡a pesar de que la matriz no tiene ningún elemento complejo! ■

Cuando la ecuación $Ax = \lambda x$ es válida y x no es cero, decimos que λ es un valor propio de A y x es un vector propio asociado. Así, en el ejemplo 1, la matriz tiene 5 como un valor propio asociado con el vector propio $[1, 1]^T$ y -4 es otro vector propio con valor propio asociado $[2, -7]^T$. El ejemplo 2 pone de manifiesto que una matriz real puede tener valores propios complejos y vectores propios complejos. Observe que una ecuación $A\mathbf{0} = \lambda\mathbf{0}$ y una ecuación $A\mathbf{0} = \mathbf{0}x$ no dicen nada útil acerca de los valores y vectores propios de A .

Muchos de los problemas en ciencia conducen a problemas de valores propios en los que en general la cuestión principal es: ¿cuáles son los valores propios de una matriz dada y cuáles son los vectores propios asociados? Un uso sobresaliente de esta teoría es en sistemas de ecuaciones diferenciales lineales, de los que se hablará más adelante.

Observe que si $Ax = \lambda x$ y $x \neq 0$, entonces cada múltiplo distinto de cero de x es un vector propio (con el mismo valor propio). Si λ es un valor propio de una matriz A de $n \times n$, entonces el conjunto $\{x: Ax = \lambda x\}$ es un subespacio de \mathbb{R}^n llamado el **espacio propio**, que necesariamente es de dimensión al menos 1.

Cálculo de valores propios y vectores propios

Dada una matriz cuadrada A , ¿cómo encontrar sus valores propios? Comenzamos por observar que la ecuación $Ax = \lambda x$ es equivalente a $(A - \lambda I)x = \mathbf{0}$. Puesto que estamos interesados en las soluciones distintas de cero a esta ecuación, la matriz $A - \lambda I$ debe ser singular (no invertible) y, por lo tanto, $\text{Det}(A - \lambda I) = \mathbf{0}$. Esto es como (en principio) podemos encontrar todos los valores propios de A . En concreto, forme la función p con la definición $p(\lambda) = \text{Det}(A - \lambda I)$ y halle los ceros de p . Resulta que p es un polinomio de grado n y debe tener n ceros, siempre que permitamos ceros complejos y contamos cada cero un número de veces igual a su multiplicidad. Incluso si la matriz A es real, debemos estar preparados para valores propios complejos. El polinomio que acabamos de describir se llama el **polinomio característico** de la matriz A . Si el polinomio tiene un factor repetido, como $(\lambda - 3)^k$, entonces decimos que 3 es una raíz de multiplicidad k . Esas raíces aún son valores propios, pero pueden ser problemáticas cuando $k > 1$.

Para mostrar el cálculo de los valores propios vamos a utilizar la matriz del ejemplo 1, a saber,

$$A = \begin{bmatrix} 3 & 2 \\ 7 & -2 \end{bmatrix}$$

El polinomio característico es

$$\begin{aligned} p(\lambda) &= \text{Det}(A - \lambda I) = \text{Det} \begin{bmatrix} 3 - \lambda & 2 \\ 7 & -2 - \lambda \end{bmatrix} = (3 - \lambda)(-2 - \lambda) - 14 \\ &= \lambda^2 - \lambda - 20 = (\lambda - 5)(\lambda + 4) \end{aligned}$$

Los valores propios son 5 y -4.

Podemos realizar este cálculo con una o dos instrucciones de Matlab, Maple o Mathematica. Podemos determinar el polinomio característico y, posteriormente, calcular sus ceros. Esto nos da dos raíces del polinomio característico, que son los valores propios 5 y -4. Estos sistemas de software matemático también tienen instrucciones sólo para generar una lista de valores propios, que se calcula de la mejor manera posible, ¡que usualmente *no* determinan el polinomio característico y el subsecuente cálculo de sus ceros!

En general, una matriz de $n \times n$ tendrá un polinomio característico de grado n y sus raíces son los valores propios de A . Puesto que el cálculo de ceros de un polinomio es numéricamente desafiante si no es inestable, este sencillo procedimiento no es recomendable (véase el problema de cómputo 8.3.2 para un experimento con respecto a esta situación). Sin embargo, para valores pequeños de n , puede ser muy satisfactorio. A éste se le llama el **método directo** para el cálculo de valores propios.

Una vez que se ha determinado un valor propio λ para una matriz A , se puede calcular un vector propio resolviendo el sistema $(A - \lambda I)x = \mathbf{0}$. Así, en el ejemplo 1 debemos resolver $(A - 5I)x = \mathbf{0}$, o

$$\begin{bmatrix} -2 & 2 \\ 7 & -7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Por supuesto, esta matriz es singular y la ecuación homogénea tiene soluciones no triviales, tales como $[1, 1]^T$. El otro valor propio se trata de la misma manera, dando lugar a un vector propio $[2, -7]^T$. Cualquier múltiplo escalar de un vector propio es también un vector propio.

Este trabajo se puede realizar mediante el uso de software matemático para encontrar un vector propio de cada valor propio λ a través del espacio nulo de la matriz $A - \lambda I$. Además, podemos utilizar una instrucción única para calcular todos los valores propios directamente o solicitar el cálculo de todos los valores y vectores propios de una vez. La instrucción de Matlab $[V, D] = \text{eig}(A)$ produce dos matrices, V y D . La matriz V tiene vectores propios de A como sus columnas y la matriz D contiene todos los valores propios de A en su diagonal. El programa devuelve un vector de longitud unitaria como $[0.7071, 0.7071]^T$. Este vector por sí mismo proporciona la base para un espacio nulo de $A - 5I$.

Observe que el problema valor propio-vector propio no es lineal. La ecuación $Ax = \lambda x$ tiene dos incógnitas, λ y x . Se presentan en la ecuación multiplicadas entre sí. Si x o λ fueran conocidas, la búsqueda de la otra sería un problema lineal muy fácil.

Software matemático

Un uso mundial, típico del software matemático como Matlab podría ser calcular los valores propios y los valores propios de una matriz con una instrucción tal como $[V, D] = \text{eig}(A)$ para la matriz

$$A = \begin{bmatrix} 1 & 3 & -7 \\ -3 & 4 & 1 \\ 2 & -5 & 3 \end{bmatrix}$$

Matlab responde instantáneamente con los vectores propios en la matriz V y los valores propios en la diagonal de la matriz D . El valor propio real es 0.0214 y el par de valores propios complejos son $3.9893 \pm 5.5601i$. En el fondo, se pueden estar realizando cálculos mucho más complicados. El procedimiento general tiene los siguientes componentes. Primero, por medio de las transformaciones de semejanza, A se escribe en la forma inferior de Hessenberg. Esto significa que todos los elementos debajo de la primera subdiagonal son cero. Así, la nueva $A = (a_{ij})$ satisface $a_{ij} = 0$ cuando $i > j + 1$. Las transformaciones de semejanza garantizan que los valores propios no son perturbados. Si A es real, transformaciones de semejanza posteriores colocan a A cerca de una forma diagonal en la que cada elemento diagonal es ya sea un número real o una matriz real de 2×2 cuyos valores propios son un par de números complejos conjugados. La creación de los ceros adicionales justo debajo de la diagonal requiere cierto proceso iterativo, porque después de todo, estamos en efecto calculando los ceros de un polinomio. El proceso iterativo es una reminiscencia del método de potencias que se describe en la sección 8.4.

Maple se puede utilizar para calcular los valores propios y vectores propios. Las cantidades se calculan en aritmética exacta y luego se convierte en punto flotante. En algunas versiones de Maple y Matlab, se pueden utilizar algunas de las instrucciones de uno de estos paquetes en el otro. En Mathematica, podemos usar las instrucciones para obtener resultados similares.

El mejor consejo para quien se enfrenta a un desafiante problema de valores propios es utilizar el software del paquete LAPACK. Hay algoritmos especiales de valores propios para los distintos tipos de matrices. Por ejemplo, si la matriz en cuestión es real y simétrica, se debe utilizar un algoritmo a la medida para ese caso. Hay alrededor de una docena de categorías disponibles para elegir en LAPACK. Matlab emplea algunos de los programas en LAPACK.

Propiedades de los valores propios

Un teorema que resume las propiedades especiales de una matriz que inciden en el cálculo de sus valores propios es el siguiente.

■ TEOREMA 1

Propiedades de la matriz de valores propios

Las siguientes afirmaciones son verdaderas para cualquier matriz cuadrada A :

1. Si λ es un valor propio de A , entonces $p(\lambda)$ es un valor propio de $p(A)$, para cualquier polinomio p . En particular, λ^k es un valor propio de A^k .
2. Si A es no singular y λ es un valor propio de A , entonces $p(1/\lambda)$ es un valor propio de $p(A^{-1})$, para cualquier polinomio p . En particular, λ^{-1} es un valor propio de A^{-1} .
3. Si A es real y simétrica, entonces sus valores propios son reales.
4. Si A es compleja y hermitiana, sus valores propios son reales.
5. Si A es hermitiana y positiva definida, entonces sus valores propios son positivos.
6. Si P es no singular, entonces A y PAP^{-1} tienen el mismo polinomio característico (y los mismos valores propios).

Recuerde que una matriz A es **simétrica** si $A = A^T$, donde $A^T = (a_{ji})$ es la **transpuesta** de $A = (a_{ij})$. Por otro lado, una matriz compleja A es **hermitiana** si $A = A^*$, donde $A^* = \bar{A}^T = (\bar{a}_{ji})$. Aquí, A^* es la transpuesta conjugada de la matriz A . Usando la sintaxis de programación se puede escribir $A^T(i, j) = A(j, i)$ y $A^* = (i, j) = \bar{A}(j, i)$. Recuerde también que A es **definida positiva** si $x^T Ax > 0$ para todos los vectores x distintos de cero.

Dos matrices A y B son **semejantes** entre sí si existe una matriz P no singular tal que $B = PAP^{-1}$. Las matrices semejantes tienen el mismo polinomio característico

$$\begin{aligned}\text{Det}(B - \lambda I) &= \text{Det}(PAP^{-1} - \lambda I) \\ &= \text{Det}(P(A - \lambda I)P^{-1}) \\ &= \text{Det}(P) \cdot \text{Det}(A - \lambda I) \cdot \text{Det}(P^{-1}) \\ &= \text{Det}(A - \lambda I)\end{aligned}$$

Por ello, tenemos un teorema importante.

■ TEOREMA 2

Valores propios de matrices semejantes

Las matrices semejantes tienen los mismos valores propios.

Este teorema sugiere una estrategia para encontrar valores propios de A . Transformar la matriz A a una matriz B mediante una transformación de semejanza $B = PAP^{-1}$ en la que B tiene una estructura especial y luego encontrar los valores propios de la matriz B . En concreto, si B es triangular o diagonal, los valores propios de B (y de A) son simplemente los elementos de la diagonal de B .

Se dice que las matrices A y B son **unitariamente semejantes** entre sí si $B = U^*AU$ para alguna matriz unitaria U . Recuerde que una matriz U es **unitaria** si $UU^* = I$. Esto nos lleva naturalmente a otro teorema importante y dos corolarios.

TEOREMA 3**Teorema de Schur**

Cada matriz cuadrada es unitariamente semejante a una matriz triangular.

En este teorema, se da una matriz arbitraria compleja A de $n \times n$ y se afirma que existe matriz unitaria U tal que:

$$U A U^* = T$$

donde $UU^* = I$ y T es una matriz triangular.

La demostración del teorema de Schur se puede encontrar en Kincaid y Cheney [2002] y en Golub y Van Loan [1996].

COROLARIO 1**Matriz semejante a una matriz triangular**

Cada matriz real cuadrada es semejante a una matriz triangular.

Por tanto, la factorización

$$PAP^{-1} = T$$

es posible, donde T es triangular, P es invertible y A es real.

EJEMPLO 3 Ejemplificamos el teorema de Schur encontrando la descomposición de esta matriz de 2×2 :

$$A = \begin{bmatrix} 3 & -2 \\ 8 & 3 \end{bmatrix}$$

Solución De la ecuación característica $\det(A - \lambda I) = \lambda^2 - 6\lambda + 25 = 0$, los valores propios son de $3 \pm 4i$. Al resolver $A - \lambda I = 0$ con cada uno de estos valores propios, los correspondientes vectores propios son $v_1 = [i, 2]^T$ y $v_2 = [-i, 2]^T$. Utilizando el proceso de ortogonalización de Gram-Schmidt, obtenemos $u_1 = v_1$ y $u_2 = v_2 - [v_2^* u_1 / v_1^* u_1] v_1 = [-2, -i]^T$. Despues de la normalización de estos vectores, se obtiene la matriz unitaria

$$U = \frac{1}{\sqrt{5}} \begin{bmatrix} i & -2 \\ 2 & -i \end{bmatrix}$$

que satisface la propiedad $UU^* = I$. Por último, se obtiene la forma de Schur

$$U A U^* = \begin{bmatrix} 3 + 4i & -6 \\ 0 & 3 - 4i \end{bmatrix}$$

que es una matriz triangular superior con los valores propios en la diagonal. ■

COROLARIO 2**Matriz hermitiana unitariamente semejante a una matriz diagonal**

Cada matriz cuadrada hermitiana es unitariamente semejante a una matriz diagonal.

En el segundo corolario, una matriz hermitiana, A se factoriza como

$$A = U^* D U$$

donde D es diagonal y U es unitaria.

Además, $U^* A U = T$ y $U^* A^* U = T^*$ y $A = A^*$, por lo que $T = T^*$, lo que debe ser una matriz diagonal.

La mayoría de los métodos numéricos para encontrar los valores propios de una matriz A de $n \times n$ proceden determinando estas transformaciones de semejanza. Entonces, se calcula un valor propio a la vez, digamos, λ , y se utiliza un **proceso de deflación** para producir una matriz \tilde{A} de $(n-1) \times (n-1)$ cuyos valores propios son los mismos que los de A , excepto por λ . Dicho procedimiento se puede repetir con la matriz \tilde{A} para encontrar tantos valores propios de la matriz A como se desee. En la práctica, se debe utilizar esta estrategia con precaución, ya que los valores propios sucesivos pueden estar infectados con errores de redondeo.

Teorema de Gershgorin

A veces es necesario determinar en una forma burda donde se sitúan en el plano complejo \mathbb{C} los valores propios de la matriz. El más famoso de los teoremas llamados **teoremas de localización** es el siguiente.

■ TEOREMA 4

Teorema de Gershgorin

Todos los valores propios de una matriz $A = (a_{ij})$ de $n \times n$ están contenidos en la unión de los n discos $C_i = C_i(a_{ii}, r_i)$ en el plano complejo con centro en a_{ii} y radios r_i dada por la suma de las magnitudes de las entradas fuera de la diagonal en el i -ésimo renglón.

La matriz A puede tener enteros ya sean reales o complejos. La región que contiene los valores propios de A se puede escribir como

$$\bigcup_{i=1}^n C_i = \bigcup_{i=1}^n \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i\}$$

donde los radios son $r_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$.

Los valores propios de A y A^T son iguales porque la ecuación característica implica el determinante, que es igual para una matriz y su transpuesta. Por lo tanto, podemos aplicar el teorema de Gershgorin a A^T y obtener el siguiente resultado útil .

■ COROLARIO 3

Más discos de Gershgorin

Todos los valores propios de una matriz $A = (a_{ii})$ de $n \times n$, se encuentran en la unión de los n discos $D_i = D_i(a_{ii}, s_i)$ en el plano complejo con centro en a_{ii} y radios s_i dada por la suma de las magnitudes de las columnas de A .

En consecuencia, la región que contiene los valores propios de A se puede escribir como

$$\bigcup_{i=1}^n D_i = \bigcup_{i=1}^n \{z \in \mathbb{C} : |z - a_{ii}| \leq s_i\}$$

donde los radios son $s_i = \sum_{\substack{i=j \\ i \neq j}}^n |a_{ij}|$. Por último, la región que contiene los valores propios de A es

$$\left(\bigcup_{i=1}^n C_i \right) \cap \left(\bigcup_{i=1}^n D_i \right)$$

Esta puede contener en algunos casos límites más estrictos en los valores propios. Además, un resultado útil de localización es

COROLARIO 4

Para una matriz A , la unión de los k discos de Gershgorin que no intersecan a los restantes $n - k$ círculos contiene exactamente k (contando multiplicidades) valores propios de A .

Para una matriz con estricto dominio diagonal, el cero no se puede encontrar en cualquiera de sus discos de Gershgorin, por lo que debe ser invertible. En consecuencia, se obtiene el siguiente resultado.

COROLARIO 5

Cada matriz con estricto dominio diagonal es no singular.

EJEMPLO 4

Considere la matriz

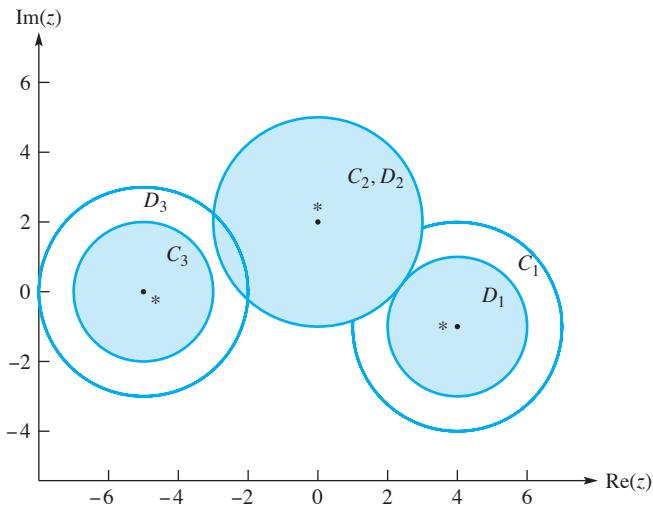
$$A = \begin{bmatrix} 4-i & 2 & i \\ -1 & 2i & 2 \\ 1 & -1 & -5 \end{bmatrix}$$

Dibuje los discos de Gershgorin.

Solución Usando los renglones de A , encontramos que los discos de Gershgorin son $C_1(4 - i, 3)$, $C_2(2i, 3)$ y $C_3(-5, 2)$. Usando las columnas de A , se obtiene más discos de Gershgorin: $D_1(4 - i, 2)$, $D_2(2i, 3)$ y $D_3(-5, 3)$. En consecuencia, todos los valores propios de A están en los tres discos D_1 , C_2 y C_3 , como se muestra en la figura 8.1. Por otros medios, calculamos los valores propios de A como $\lambda_1 = 3.7208 - 1.05461i$, $\lambda_2 = 4.5602 + -0.2849i$ y $\lambda_3 = -0.1605 + 2.3395i$. En la figura 8.1, el centro de los discos está denotado por puntos • y los valores propios por *.

Descomposición en valor singular

En esta subsección se requiere que usted tenga un conocimiento adicional de álgebra lineal, en particular, la diagonalización de matrices simétricas, valores propios, vectores propios, rango, espacio



de columna y normas. Consulte el apéndice D para una breve revisión de estos temas. (En el análisis que sigue, se supone que se está utilizando la norma euclírdiana.)

La descomposición de valor singular es una herramienta de uso general que tiene muchas aplicaciones, en particular en problemas de mínimos cuadrados (capítulo 12). Se puede aplicar a cualquier matriz, sea o no cuadrada. Comenzamos diciendo que los **valores singulares** de una matriz A son las raíces cuadradas no negativas de los valores propios de $A^T A$.

■ TEOREMA 5

Teorema espectral de matrices

Sea A de $m \times n$. Entonces $A^T A$ es una matriz simétrica de $n \times n$ y se puede diagonalizar con una matriz ortogonal, digamos, Q :

$$A^T A = Q D Q^{-1}$$

donde $QQ^T = Q^T Q = I$ y D es una matriz diagonal de $n \times n$.

Además, la matriz diagonal D contiene los valores propios de $A^T A$ en su diagonal. Esto se deduce del hecho de que en $A^T A Q = Q D$, por lo que las columnas de Q son los vectores propios de $A^T A$. Si λ es un valor propio de A y si x es un vector propio correspondiente, entonces $A^T A x = \lambda x$ donde

$$\|Ax\|^2 = (Ax)^T (Ax) = x^T A^T A x = x^T \lambda x = \lambda \|x\|^2$$

Esta ecuación muestra que λ es real y no negativa. Podemos ordenar los valores propios como $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$. (Reordenar los valores propios requiere reordenar las columnas de Q .) Los números $\sigma_j = +\sqrt{\lambda_j}$ son los **valores singulares** de A .

Puesto que \mathbf{Q} es una matriz ortogonal, sus columnas forman una base ortonormal para \mathbb{R}^n . Estas son vectores propios unitarios de $\mathbf{A}^T \mathbf{A}$, así que si \mathbf{v}_j es la j -ésima columna de \mathbf{Q} , entonces $\mathbf{A}^T \mathbf{A} \mathbf{v}_j = \lambda_j \mathbf{V}_j$. Algunos de los valores propios de $\mathbf{A}^T \mathbf{A}$ pueden ser cero. Se define r por la condición

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0 = \lambda_{r+1} = \cdots = \lambda_n$$

Para una revisión de conceptos tales como rango, base ortogonal, base ortonormal, espacio columna, espacio nulo, etcétera, consulte el apéndice D.

■ TEOREMA 6

Teorema de bases ortogonales

Si el rango de \mathbf{A} es r , entonces una base ortogonal para el espacio columna de \mathbf{A} es $\{\mathbf{A}\mathbf{v}_j : 1 \leq j \leq r\}$.

Demostración Observe que

$$(\mathbf{A}\mathbf{v}_k)^T (\mathbf{A}\mathbf{v}_j) = \mathbf{v}_k^T \mathbf{A}^T \mathbf{A} \mathbf{v}_j = \mathbf{v}_k^T \lambda_j \mathbf{v}_j = \lambda_j \delta_{kj}$$

Esto establece la ortogonalidad del conjunto $\{\mathbf{A}\mathbf{v}_j : 1 \leq j \leq n\}$. Haciendo $k=j$, obtenemos $\|\mathbf{A}\mathbf{v}_j\|^2 = \lambda_j$. Por lo tanto, $\mathbf{A}\mathbf{v}_j \neq 0$ si y sólo si $1 \leq j \leq r$. Si \mathbf{w} es cualquier vector en el espacio columna de \mathbf{A} , entonces $\mathbf{w} = \mathbf{Ax}$ para alguna \mathbf{x} en \mathbb{R}^n . Haciendo $\mathbf{x} = \sum_{j=1}^r c_j \mathbf{v}_j$, obtenemos

$$\mathbf{w} = \mathbf{Ax} = \sum_{j=1}^r c_j \mathbf{A} \mathbf{v}_j = \sum_{j=1}^r c_j \mathbf{A} \mathbf{v}_j$$

y por lo tanto, \mathbf{w} está en el espacio de $\{\mathbf{A}\mathbf{v}_1, \mathbf{A}\mathbf{v}_2, \dots, \mathbf{A}\mathbf{v}_r\}$. ■

El teorema anterior da una forma razonable de calcular el rango de una matriz numérica. Primero, se calcula su valor singular. Cualesquiera que sean muy pequeños se puede suponer que son cero. Los que quedan son fuertemente positivos y si hay r de ellos, tomamos r para calcular numéricamente el rango de \mathbf{A} .

Una **descomposición de valor singular** de una matriz \mathbf{A} de $m \times n$ es cualquier representación de \mathbf{A} en la forma

$$\mathbf{A} = \mathbf{UDV}^T$$

donde \mathbf{U} y \mathbf{V} son matrices ortogonales y \mathbf{D} es una matriz *diagonal* de $m \times n$ que tiene elementos de la diagonal no negativos que se ordenan $d_{11} \geq d_{22} \geq \dots \geq 0$. Entonces, del problema 4, se deduce que los elementos diagonales d_{ii} son necesariamente los valores singulares de \mathbf{A} . Observe que la matriz \mathbf{U} es de $m \times m$ y \mathbf{V} es de $n \times n$. Sin embargo, una matriz no cuadrada \mathbf{D} se dice que es **diagonal** si los únicos elementos que son diferentes de cero están entre aquellos cuyos dos índices son iguales.

Una descomposición de valor singular de \mathbf{A} (hay muchas de ellas) se puede obtener del trabajo descrito anteriormente. Comience con los vectores $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r$. Normalice los vectores $\mathbf{A}\mathbf{v}_j$ para obtener los vectores \mathbf{u}_j . Por lo tanto, tenemos

$$\mathbf{u}_j = \mathbf{A}\mathbf{v}_j / \|\mathbf{A}\mathbf{v}_j\| \quad (1 \leq j \leq r)$$

Extendemos este conjunto a una base ortonormal para \mathbb{R}^m . Sea \mathbf{U} la matriz de $m \times m$ cuyas columnas son $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m$. Se define \mathbf{D} como la matriz de $m \times n$ que consta de ceros excepto para $\sigma_1, \sigma_2, \dots, \sigma_r$ en su diagonal. Sea $\mathbf{V} = \mathbf{Q}$, donde \mathbf{Q} es como se definió anteriormente.

Para comprobar la ecuación $\mathbf{A} = \mathbf{UDV}^T$, primero observe que $\sigma_j = \|\mathbf{Av}_j\|_2$ y que $\sigma_j \mathbf{u}_j = \mathbf{Av}_j$. Después calcule \mathbf{UD} . Puesto que \mathbf{D} es diagonal, esto es fácil. Obtenemos

$$\begin{aligned}\mathbf{UD} &= [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m] \mathbf{D} = [\sigma_1 \mathbf{u}_1, \sigma_2 \mathbf{u}_2, \dots, \sigma_r \mathbf{u}_r, 0, \dots, 0] \\ &= [\mathbf{Av}_1, \mathbf{Av}_2, \dots, \mathbf{Av}_r, \dots, \mathbf{Av}_n] = \mathbf{AQ} = \mathbf{AV}\end{aligned}$$

Esto implica que

$$\mathbf{A} = \mathbf{UDV}^T$$

El número de condición de una matriz se puede expresar en términos de sus valores singulares

$$\kappa(\mathbf{A}) = \sqrt{\frac{\sigma_{\max}}{\sigma_{\min}}}$$

puesto que $\|\mathbf{A}\|_2^2 = \rho(\mathbf{A}^T \mathbf{A}) = \sigma_{\max}(\mathbf{A})$ y $\|\mathbf{A}^{-1}\|_2^2 = \rho(\mathbf{A}^{-T} \mathbf{A}^{-1}) = \sigma_{\min}(\mathbf{A})$.

Ejemplos numéricos de descomposición en valor singular

La determinación numérica de la descomposición en valor singular es mejor dejársela al software disponible de alta calidad. Estos programas se pueden encontrar en Matlab, Maple, LAPACK y otros paquetes de software. Los programas de alta calidad *no* forman una $\mathbf{A}^T \mathbf{A}$ y buscan su valores propios. Se desea evitar el uso de $\mathbf{A}^T \mathbf{A}$ en el cálculo numérico porque su **número de condición** puede ser mucho peor que el de \mathbf{A} . Este fenómeno se ejemplifica fácilmente con las matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ \varepsilon & 0 & 0 \\ 0 & \varepsilon & 0 \\ 0 & 0 & \varepsilon \end{bmatrix}, \quad \mathbf{A}^T \mathbf{A} = \begin{bmatrix} 1 + \varepsilon^2 & 1 & 1 \\ 1 & 1 + \varepsilon^2 & 1 \\ 1 & 1 & 1 + \varepsilon^2 \end{bmatrix}$$

Habrá algunos valores pequeños de ε para los que \mathbf{A} tenga rango 3 y $\mathbf{A}^T \mathbf{A}$ tenga rango 1 (en la computadora).

EJEMPLO 5 En un ejemplo de la sección 1.1 (p. 4) encontramos esta matriz:

$$\mathbf{A} = \begin{bmatrix} 0.1036 & 0.2122 \\ 0.2081 & 0.4247 \end{bmatrix}$$

Determine sus valores propios, valores singulares y el número de condición.

Solución Utilizando el software matemático es fácil encontrar los valores propios $\lambda_1(\mathbf{A}) \approx -0.0003$ y $\lambda_2(\mathbf{A}) \approx 0.5286$. Podemos formar la matriz

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} 0.0540 & 0.1104 \\ 0.1104 & 0.2254 \end{bmatrix}$$

y encontrar sus propios valores $\lambda_1(\mathbf{A}^T \mathbf{A}) \approx 0.3025 \times 10^{-4}$ y $\lambda_2(\mathbf{A}^T \mathbf{A}) \approx 0.2794$. Por lo tanto, los valores singulares son $\sigma_1(\mathbf{A}) = \sqrt{|\lambda_1(\mathbf{A}^T \mathbf{A})|} \approx 0.0003$ y $\sigma_2(\mathbf{A}) = \sqrt{|\lambda_2(\mathbf{A}^T \mathbf{A})|} \approx 0.5286$. Además, podemos obtener los valores singulares directamente como $\sigma_1 \approx 0.0003$ y $\sigma_2 \approx 0.5286$ usando software matemático. En consecuencia, el número de condición es $\kappa(\mathbf{A}) = \sigma_2/\sigma_1 \approx 1747.6$. Debido a este gran número de condición, ahora entendemos por qué ¡había dificultad en la solución de un sistema de ecuaciones lineales con esta matriz de coeficientes!

EJEMPLO 6 Calcule la descomposición en valor singular de la matriz

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (1)$$

Solución En este caso, la matriz \mathbf{A} es de $m \times n$ y $m = 3$ y $n = 2$. Primero, encontramos que los valores propios de la matriz

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

arreglados en orden descendente son $\lambda_1 = 3$ y $\lambda_2 = 1$. El número de valores propios distintos de cero de la matriz $\mathbf{A}^T \mathbf{A}$ es 2. Ahora, determinamos que los vectores propios de la matriz $\mathbf{A}^T \mathbf{A}$ son $[1, 1]^T$ para $\lambda_1 = 3$ y $[1, -1]^T$ para $\lambda_2 = 1$. Por lo tanto, el conjunto ortonormal de vectores propios de $\mathbf{A}^T \mathbf{A}$ son $\left[\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2}\right]^T$ para $\lambda_1 = 3$ y $\left[\frac{1}{2}\sqrt{2}, -\frac{1}{2}\sqrt{2}\right]^T$. Despues los arreglamos en el mismo orden que los valores propios para forma los vectores columna de la matriz \mathbf{V} de $n \times n$:

$$\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2] = \begin{bmatrix} \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \end{bmatrix}$$

Ahora formamos una matriz diagonal \mathbf{D} , colocando en la diagonal que lleva los valores singulares: $\sigma_i = \sqrt{\lambda_i}$. Puesto que $\sigma_1 = \sqrt{3}$ y $\sigma_2 = 1$, la matriz de valor singular de $m \times n$ es

$$\mathbf{D} = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{1} \\ 0 & 0 \end{bmatrix}$$

Aquí, en la diagonal principal están las raíces cuadradas de los valores propios de $\mathbf{A}^T \mathbf{A}$ en orden ascendente y el resto de las entradas de la matriz \mathbf{D} son ceros. A continuación, calculamos vectores $\mathbf{u}_i = \sigma_i^{-1} \mathbf{A} \mathbf{v}_i$ para $i = 1$ y formamos los vectores columna de la matriz \mathbf{U} de $m \times m$. En este caso, encontramos

$$\mathbf{u}_1 = \sigma_1^{-1} \mathbf{A} \mathbf{v}_1 = \frac{1}{3}\sqrt{3} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\sqrt{6} \\ \frac{1}{6}\sqrt{6} \\ \frac{1}{6}\sqrt{6} \end{bmatrix}$$

y

$$\mathbf{u}_2 = \sigma_2^{-1} \mathbf{A} \mathbf{v}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\sqrt{2} \\ -\frac{1}{2}\sqrt{2} \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \end{bmatrix}$$

Por último, agregamos a la matriz \mathbf{U} , el resto de los $m - r$ vectores mediante el proceso de ortogonalización de Gram-Schmidt. Por tanto, hacemos al vector \mathbf{u}_3 perpendicular a \mathbf{u}_1 y \mathbf{u}_2 :

$$\tilde{\mathbf{u}}_3 = \mathbf{e}_1 - (\mathbf{u}_1^T \mathbf{e}_1) \mathbf{u}_1 - (\mathbf{u}_2^T \mathbf{e}_1) \mathbf{u}_2 = \begin{bmatrix} \frac{1}{3} \\ -\frac{1}{3} \\ -\frac{1}{3} \end{bmatrix}$$

Normalizando al vector \mathbf{u}_3 , obtenemos

$$\mathbf{u}_3 = \begin{bmatrix} \frac{1}{3}\sqrt{3} \\ -\frac{1}{3}\sqrt{3} \\ -\frac{1}{3}\sqrt{3} \end{bmatrix}$$

Así, tenemos la matriz

$$\mathbf{U} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \mathbf{u}'_3] = \begin{bmatrix} \frac{1}{3}\sqrt{6} & 0 & \frac{1}{3}\sqrt{3} \\ \frac{1}{6}\sqrt{6} & \frac{1}{2}\sqrt{2} & -\frac{1}{3}\sqrt{3} \\ \frac{1}{6}\sqrt{6} & -\frac{1}{2}\sqrt{2} & -\frac{1}{3}\sqrt{3} \end{bmatrix}$$

La descomposición en valor singular de la matriz \mathbf{A} es

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \overset{\mathbf{A} = \mathbf{UDV}^T}{=} \begin{bmatrix} \frac{1}{3}\sqrt{6} & 0 & \frac{1}{3}\sqrt{3} \\ \frac{1}{6}\sqrt{6} & \frac{1}{2}\sqrt{2} & -\frac{1}{3}\sqrt{3} \\ \frac{1}{6}\sqrt{6} & -\frac{1}{2}\sqrt{2} & -\frac{1}{3}\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{1} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \end{bmatrix}$$

Así que ¡ahí la tenemos! Afortunadamente, ¡hay software matemático que hace todo esto al instante! Podemos verificar los resultados calculando la matriz diagonal y la matriz \mathbf{A} de la factorización. ■

Consulte los capítulos 12 y 16 para algunas aplicaciones importantes de la descomposición en valor singular. Ahí se dan otros ejemplos y en los problemas de esos capítulos.

Aplicación: ecuaciones diferenciales lineales

Aquí se explica brevemente la aplicación de la teoría de los valores propios para sistemas de ecuaciones diferenciales lineales. Comencemos con una sola ecuación diferencial lineal con una *variable dependiente* x . La variable *independiente* es t y con frecuencia representa el tiempo. Escribimos $x' = ax$ o con más detalle $(d/dt)x(t) = ax(t)$. Hay una familia de soluciones, a saber, $x(t) = ce^{at}$, donde c es un parámetro real arbitrario. Si se da un valor inicial $x(0)$, tendremos el parámetro c para obtener el valor inicial de la derecha.

Un par de **ecuaciones diferenciales lineales** con dos variables dependientes, x_1 y x_2 se verá así:

$$\begin{cases} x'_1 = a_{11}x_1 + a_{12}x_2 \\ x'_2 = a_{21}x_1 + a_{22}x_2 \end{cases}$$

La forma general de un sistema de n ecuaciones diferenciales lineales de primer orden, con coeficientes constantes, es simplemente $\mathbf{x}' = \mathbf{Ax}$. Aquí, \mathbf{A} es una matriz numérica de $n \times n$ y el vector \mathbf{x} tiene n componentes, x_j , siendo cada uno, una función de t . La derivada es con respecto a t . Para resolver esto, nos guiamos con el sencillo caso de $n = 1$, analizado anteriormente. Aquí, probamos $\mathbf{x}(t) = e^{\lambda t}\mathbf{v}$, donde \mathbf{v} es un vector constante. Obteniendo la derivada de \mathbf{x} , tenemos $\mathbf{x}' = \lambda e^{\lambda t}\mathbf{v}$. Ahora el sistema de ecuaciones se ha convertido en $\lambda e^{\lambda t}\mathbf{v} = \mathbf{A}e^{\lambda t}\mathbf{v}$, o $\lambda\mathbf{v} = \mathbf{Av}$. Así es como los valores propios entran en el proceso. Hemos probado el siguiente resultado.

TEOREMA 7**Ecuaciones diferenciales lineales**

Si λ es un valor propio de la matriz A y si v es un vector propio asociado, entonces una solución de la ecuación diferencial $x' = Ax$ es $x(t) = e^{\lambda t}v$.

Aplicación: un problema de vibración

El análisis de valores y vectores propios se puede utilizar en muchas ecuaciones diferenciales. Consideré el sistema de dos masas y tres resortes que se muestra en la figura 8.2. Aquí, las masas se ven obligadas a moverse sólo en la dirección horizontal.

FIGURA 8.2
Problema de
dos masas
vibrando



A partir de esta situación, escribimos las ecuaciones de movimiento en la forma matriz-vector:

$$\begin{bmatrix} x_1'' \\ x_2'' \end{bmatrix} = \begin{bmatrix} -\beta & \alpha \\ \alpha & -\beta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad x'' = Ax$$

Suponiendo que la solución es puramente oscilatoria (no amortiguada), tenemos

$$x = ve^{i\omega t}$$

En forma matricial, obtenemos

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} e^{i\omega t}$$

Derivando, obtenemos

$$x'' = -\omega^2 ve^{i\omega t} = -\omega^2 x$$

y

$$\begin{bmatrix} -\beta & \alpha \\ \alpha & -\beta \end{bmatrix} x = -\omega^2 x$$

Este es el problema de valores propios

$$Ax = \lambda x$$

donde $\lambda = -\omega^2$. Los valores propios se pueden encontrar de la ecuación característica:

$$\det(A + \omega^2 I) = \det \begin{bmatrix} \omega^2 - \beta & \alpha \\ \alpha & \omega^2 - \beta \end{bmatrix} = 0$$

Esto es $(\omega^2 - \beta)^2 - \alpha^2 = \omega^4 - 2\beta\omega^2 + (\beta^2 - \alpha^2) = 0$, y

$$\omega^2 = \frac{1}{2} [2\beta \pm \sqrt{4\beta^2 - 4(\beta^2 - \alpha^2)}] = \beta \pm \alpha$$

Por simplicidad, ahora suponemos masa unitarias y resortes unitarios, por lo que $\beta = 2$ y $\alpha = 1$. Entonces obtenemos

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}$$

Entonces, las raíces de las ecuaciones características son $\omega_1^2 = \beta + \alpha = 3$ y $\omega_2^2 = \beta - \alpha = 1$. A continuación podemos encontrar los vectores propios. Para el primer valor propio, obtenemos

$$(A + \omega_1^2 I) \mathbf{v}_1 = \mathbf{0} \quad \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \mathbf{0}$$

Puesto que $v_{11} = -v_{12}$, obtenemos el primer vector propio

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Para el segundo vector propio, tenemos

$$(A + \omega_2^2 I) \mathbf{v}_2 = \mathbf{0} \quad \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \mathbf{0}$$

Puesto que $v_{21} = -v_{22}$, obtenemos el segundo vector propio

$$\mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

La solución general para las ecuaciones de movimiento para el sistema de dos masas es

$$\mathbf{x}(t) = c_1 \mathbf{v}_1 e^{i\omega_1 t} + c_2 \mathbf{v}_1 e^{-i\omega_1 t} + c_3 \mathbf{v}_2 e^{i\omega_2 t} + c_4 \mathbf{v}_2 e^{-i\omega_2 t}$$

Debido a que la solución era el cuadrado de la frecuencia, cada frecuencia se utiliza dos veces (una positiva y otra negativa). Podemos utilizar las condiciones iniciales para encontrar los coeficientes desconocidos.

Resumen

(1) El valor propio λ y el vector propio \mathbf{x} satisfacen la ecuación $A\mathbf{x} = \lambda\mathbf{x}$. El método directo para calcular los valores propios es encontrar las raíces de la ecuación característica $p(\lambda) = \det(A - \lambda I) = 0$. Entonces, para cada uno de los valores propios λ , los vectores propios se encuentran resolviendo el sistema homogéneo $(A - \lambda I)\mathbf{x} = 0$. Hay paquetes de software para encontrar los pares valor propio-vector propio con métodos más complejos.

(2) Hay muchas propiedades útiles para las matrices que influyen en sus valores propios. Por ejemplo, los valores propios son reales cuando A es simétrica o hermitiana. Los valores propios son positivos cuando A es simétrica o hermitiana definida positiva.

(3) Muchos procedimientos de valores propios implican transformaciones de semejanza o unitarias para producir matrices triangulares o diagonales.

(4) Los discos de Gershgorin se pueden utilizar para localizar valores propios, encontrando estimaciones burdas de ellos.

(5) La **descomposición en valor singular** de una matriz A de $m \times n$ es

$$A = \mathbf{U} \mathbf{D} \mathbf{V}^T$$

donde \mathbf{D} es una matriz diagonal de $m \times n$ cuyos elementos diagonales son los valores singulares, \mathbf{U} es una matriz ortogonal de $m \times m$ y \mathbf{V} es una matriz ortogonal de $n \times n$. Los valores singulares de A son las raíces cuadradas no negativas de los valores propios de $A^T A$.

Problemas 8.3

1. ¿Son $[i, -1 + i]^T$ y $[-i, -1 - i]^T$ vectores propios de la matriz del ejemplo 2?
2. Demuestre que si λ es un valor propio de una matriz real con vector propio \mathbf{x} , entonces $\bar{\lambda}$ es también un valor propio con vector propio $\bar{\mathbf{x}}$. (Para un número complejo $z = x + iy$, el conjugado se define como $\bar{z} = x - iy$.)
3. Sea

$$\mathbf{A} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

Consideré el hecho de que la matriz \mathbf{A} tiene el efecto de los vectores de rotación en sentido contrario al reloj a través de un ángulo θ y por ende no se puede mapear cualquier vector en un múltiplo de sí mismo.

4. Sea \mathbf{A} una matriz de $m \times n$ tal que $\mathbf{A} = \mathbf{UDV}^T$, donde \mathbf{U} y \mathbf{V} son ortogonales y \mathbf{D} es diagonal y no negativa. Demuestre que los elementos diagonales de \mathbf{D} son los valores singulares de \mathbf{A} .
5. Sean \mathbf{A} , \mathbf{U} , \mathbf{D} y \mathbf{V} como en la descomposición en valor singular: $\mathbf{A} = \mathbf{UDV}^T$. Sea r tal como se describe en el libro. Se define \mathbf{U}_r consistiendo de las primeras r columnas de \mathbf{U} . Sea \mathbf{V}_r que consta de las primeras r columnas de \mathbf{V} , y sea \mathbf{D}_r una matriz de $r \times r$ que tiene la misma diagonal que \mathbf{D} . Demuestre que $\mathbf{A} = \mathbf{U}_r \mathbf{D}_r \mathbf{V}_r^T$. (Esta factorización se llama la **versión económica** de la descomposición en valor singular.)
6. Un mapeo lineal P es una **proyección** si $P^2 = P$. Podemos utilizar la misma terminología para una matriz de $n \times n$: $\mathbf{A}^2 = \mathbf{A}$ es la propiedad de la proyección. Utilice la **descomposición de Pierce**, $I = \mathbf{A} + (I - \mathbf{A})$, para mostrar que cada punto en \mathbb{R}^n es la suma de un vector en el rango de \mathbf{A} y un vector en el espacio nulo de \mathbf{A} . ¿Cuáles son los valores propios de una proyección?
7. Encuentre todos los discos de Gershgorin para las siguientes matrices. Indique la(s) región(es) más pequeña(s) que contiene(n) todos los valores propios:

$$\text{a. } \begin{bmatrix} 3 & -1 & 1 \\ 2 & 4 & -2 \\ 3 & -1 & 9 \end{bmatrix} \quad \text{b. } \begin{bmatrix} 3 & 1 & 2 \\ -1 & 4 & -1 \\ 1 & -2 & 9 \end{bmatrix} \quad \text{c. } \begin{bmatrix} 1-i & 1 & i \\ 0 & 2i & 2 \\ 1 & 0 & 2 \end{bmatrix}$$

8. (Opción múltiple) Sea \mathbf{A} una matriz invertible de $n \times n$ (no singular). Sea \mathbf{x} un vector distinto de cero. Supongamos que $\mathbf{Ax} = \lambda \mathbf{x}$. ¿Qué ecuación *no* se deduce de estas hipótesis?

- a. $\mathbf{A}^k \mathbf{x} = \lambda^k \mathbf{x}$
- b. $\lambda^{-k} \mathbf{x} = (\mathbf{A}^{-1})^k \mathbf{x}$ para $k \geq 0$
- c. $p(\mathbf{A})\mathbf{x} = p(\lambda)\mathbf{x}$ para cualquier polinomio p
- d. $\mathbf{A}^k \mathbf{x} = (1 - \lambda)^k \mathbf{x}$
- e. Ninguna de estas.

9. (Opción múltiple) ¿Para qué valores de s será la matriz $\mathbf{I} - s\mathbf{v}\mathbf{v}^*$ unitaria, donde \mathbf{v} es un vector columna de longitud unitaria?

- a. 0, 1
- b. 0, 2
- c. 1, 2
- d. 0, $\sqrt{2}$
- e. Ninguna de estas

10. (Opción múltiple) Sean \mathbf{U} y \mathbf{V} matrices unitarias de $n \times n$, posiblemente complejas. ¿Qué conclusión *no* se justifica?

- a. $U + V$ es unitaria. b. U^* es unitaria. c. UV es unitaria.
d. $U - vv^*$ es unitaria cuando $\|v\| = \sqrt{2}$ y v es un vector columna. e. Ninguna de estas.

11. (Opción múltiple) ¿Cuál enunciado es verdadero?

- a. Toda matriz de $n \times n$ tiene n distintos valores propios.
b. Los valores propios de una matriz real son reales.
c. Si U es una matriz unitaria, entonces $U^* = U^T$
d. Una matriz cuadrada y su transpuesta tienen los mismos valores propios.
e. Ninguna de estas.

12. (Opción múltiple) Considere la matriz simétrica

$$A = \begin{bmatrix} 1 & 3 & 4 & -1 \\ 3 & 7 & -6 & 1 \\ 4 & -6 & 3 & 0 \\ -1 & 1 & 0 & 5 \end{bmatrix}$$

¿Cuál es el más pequeño intervalo deducido del teorema de Gershgorin para el que todos los valores propios de la matriz A se encuentran en ese intervalo?

- a. $[-7, 9]$ b. $[-7, 13]$ c. $[3, 7]$ d. $[-3, 17]$ e. Ninguna de estas.

13. (Falso o verdadero) El teorema de Gershgorin afirma que todo valor propio de una matriz A de $n \times n$ deberá cumplir una de estas desigualdades:

$$|\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad \text{para } 1 \leq i \leq n.$$

14. (Falso o verdadero) Una consecuencia del teorema de Schur es que toda matriz cuadrada A puede factorizarse como $A = PTP^{-1}$, donde P es una matriz no singular y T es triangular superior.

15. (Falso o verdadero) Una consecuencia del teorema de Schur es que toda matriz (real) simétrica puede factorizarse en la forma $A = PDP^{-1}$, donde P es unitaria y D es diagonal.

16. Explique por qué $\|UB\|_2 = \|B\|_2$ para cualquier matriz B cuando $U^TU = I$.

17. Considere la matriz $A = \begin{bmatrix} 4 & -\frac{1}{2} & 0 \\ \frac{3}{5} & 5 & -\frac{3}{5} \\ 0 & \frac{1}{2} & 3 \end{bmatrix}$. Dibuje los discos de Gershgorin en el plano complejo para A y A^T e indique las posiciones de los valores propios.

18. (Continuación) Sea B la matriz obtenida al cambiar las entradas negativas en A por números positivos. Repita el proceso para B .

19. (Continuación) Repita para $C = \begin{bmatrix} 4 & 0 & -2 \\ 1 & 2 & 0 \\ 1 & 1 & 9 \end{bmatrix}$

20. Encuentre la descomposición de Schur de $A = \begin{bmatrix} 5 & 7 \\ -2 & -4 \end{bmatrix}$.

Problemas de cómputo 8.3

1. Use Matlab, Maple, Mathematica u otros programas de computadora que tenga disponibles para calcular los valores propios y los vectores propios de estas matrices:

a. $A = \begin{bmatrix} 1 & 7 \\ 2 & -5 \end{bmatrix}$

b. $\begin{bmatrix} 4 & -7 & 3 & 2 & 3 \\ 1 & 6 & 11 & -1 & 2 \\ 5 & -5 & -2 & -4 & 1 \\ 9 & -3 & 1 & 6 & 5 \\ 3 & 2 & 5 & -5 & 1 \end{bmatrix}$

c. Sea $n = 12$, $a_{ij} = i/j$ cuando $i \leq j$ y $a_{ij} = j/i$ cuando $i > j$. Encuentre los valores propios.

d. Construya una matriz de $n \times n$ con estructura tridiagonal y elementos distintos de cero ($-1, 2, -1$) en cada renglón. Para $n = 5$ y 20 , determine todos los valores propios y compruebe que son $2 - 2\cos(j\pi/(n + 1))$.

e. Para cualquier entero positivo n , forme la matriz simétrica cuya parte triangular superior está dada por

$$\begin{bmatrix} n & n-1 & n-2 & n-3 & \cdots & 2 & 1 \\ n-1 & n-2 & n-3 & \cdots & 2 & 1 & \\ n-2 & n-3 & \cdots & 2 & 1 & & \\ \ddots & \ddots & \vdots & \vdots & & & \\ & \ddots & 2 & 1 & & & \\ & & 2 & 1 & & & \\ & & & 1 & & & \end{bmatrix}$$

Los valores propios de A son $1/\{2 - 2\cos[(2i - \lambda)\pi/(2n + 1)]\}$. (Véase Frank [1958] y Gregory y Karney [1969].) Numéricamente compruebe este resultado para $n = 30$.

2. Use Matlab para calcular los valores propios de una matriz aleatoria de 100×100 usando directamente la instrucción `eig` y usando las instrucciones `poly` y `roots`. Use las funciones cronometradas para determinar el tiempo de CPU para cada una.
3. Sea p el polinomio de grado 20 cuyas raíces son los enteros $1, 2, \dots, 20$. Encuentre la forma de potencias usual de este polinomio tal que $p(t) = t^{20} + a_{19}t^{19} + a_{18}t^{18} + \dots + a_0$. Después, la forma llamada **matriz acompañante**, que es de 20×20 y tiene ceros en todas las posiciones excepto todos los 1 en la superdiagonal y los coeficientes $-a_0, -a_1, \dots, -a_{19}$ así como su renglón inferior. Encuentre los valores propios de esta matriz y explique cualquier dificultad encontrada.
4. **(Proyecto de investigación estudiantil)** Investigue algunos métodos modernos para el cálculo de valores propios y vectores propios. Para el caso simétrico, consulte el libro de Parlett [1997]. También lea la guía del usuario de LAPACK. (Véase Anderson y colaboradores [1999].)
5. **(Proyecto de investigación estudiantil)** Experimente con el teorema de Cayley-Hamilton, que afirma que toda matriz cuadrada satisface su propia ecuación característica. Verifique esto

numéricamente usando Matlab o algún otro software de sistema matemático. Use matrices de tamaño 3, 6, 9, 12 y 15 y explique cualquier sorpresa. Si puede usar alta precisión aritmética, hágallo-Matlab trabaja con 15 dígitos de precisión.

- 6. (Proyecto de investigación estudiantil)** Experimente con el algoritmo *QR* y la descomposición en valor singular de matrices, por ejemplo, usando Matlab. Pruebe ejemplos con cuatro tipos de ecuaciones $Ax = b$, a saber, (a) el sistema tiene una única solución, (b) el sistema tiene muchas soluciones, (c) el sistema es inconsistente pero tiene una única solución de mínimos cuadrados (d) el sistema es inconsistente y tiene muchas soluciones de mínimos cuadrados.

- 7.** Usando software matemático como Matlab, Maple o Mathematica en cada de las siguientes matrices, calcule los valores propios usando el polinomio característico, calcule los vectores propios usando el espacio nulo de la matriz y calcule los valores propios y vectores propios directamente:

$$\text{a. } \begin{bmatrix} 3 & 2 \\ 7 & -1 \end{bmatrix} \quad \text{b. } \begin{bmatrix} 1 & 3 & -7 \\ -3 & 4 & 1 \\ 2 & -5 & 3 \end{bmatrix}$$

- 8.** Usando software matemático como Matlab, Maple o Mathematica, determine el tiempo de ejecución para el cálculo de todos los valores propios de una matriz de 1000×1000 con entradas aleatorias.

- 9.** Usando software matemático como Matlab, Maple o Mathematica, calcule la factorización de Schur de estas matrices complejas y verifique los resultados de acuerdo con el teorema de Schur y sus corolarios:

$$\text{a. } \begin{bmatrix} 3-i & 2-i \\ 2+i & 3+i \end{bmatrix} \quad \text{b. } \begin{bmatrix} 2+i & 3+i \\ 3-i & 2-i \end{bmatrix} \quad \text{c. } \begin{bmatrix} 2-i & 2+i \\ 3-i & 3+i \end{bmatrix}$$

- 10.** Usando software matemático como Matlab, Maple o Mathematica, calcule la descomposición de valor singular de estas matrices y compruebe que cada resultado satisface la ecuación $A = UDV^T$:

$$\text{a. } \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{b. } \begin{bmatrix} 1 & 3 & -2 \\ 2 & 7 & 5 \\ -2 & -3 & 4 \\ 5 & -3 & -2 \end{bmatrix}$$

Construya la matriz diagonal $D = U^TAV$ para comprobar los resultados (se recomienda siempre). Uno puede ver que los efectos de los errores de redondeo en estos cálculos para los elementos fuera de la diagonal en D son teóricamente cero.

- “11.** Considere $A = \begin{bmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{bmatrix}$. Encuentre los valores propios y los vectores asociados de

esta matriz, de Gregory y Karney [1969], sin usar software. *Sugerencia:* las respuestas pueden ser enteros.

- 12.** Encuentre la descomposición de valor singular en estas matrices:

$$\text{a. } \begin{bmatrix} 2 & 1 & -2 \end{bmatrix} \quad \text{b. } \begin{bmatrix} 3 \\ 4 \end{bmatrix} \quad \text{c. } \begin{bmatrix} -\frac{5}{2} + 3\sqrt{3} & \frac{5}{2}\sqrt{3} + 3 \end{bmatrix}$$

d.
$$\begin{bmatrix} 2 & 2 & 2 & 2 \\ \frac{17}{10} & \frac{1}{10} & -\frac{17}{10} & -\frac{1}{10} \\ \frac{3}{5} & \frac{9}{5} & -\frac{3}{5} & -\frac{9}{5} \end{bmatrix}$$
 e.
$$\begin{bmatrix} \frac{7}{2} - \frac{13}{6}\sqrt{6} & \frac{7}{2} + \frac{13}{6}\sqrt{6} \\ -\frac{7}{2} - \frac{13}{6}\sqrt{6} & -\frac{7}{2} + \frac{13}{6}\sqrt{6} \\ -\frac{13}{6}\sqrt{6} & \frac{13}{6}\sqrt{6} \end{bmatrix}$$

13. Considere $B = \begin{bmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{bmatrix}$. Encuentre los valores propios, los valores singulares y el número de condición de la matriz B .

8.4 Método de potencias

Un procedimiento llamado **método de potencias** se puede emplear para calcular valores propios. Este es un ejemplo de un proceso iterativo que, bajo las circunstancias adecuadas, produce una sucesión que converge a un valor propio de una matriz dada.

Supongamos que A es una matriz de $n \times n$ y que sus valores propios (que no se conocen) tienen la siguiente propiedad:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

Observe la estricta desigualdad en esta hipótesis. Salvo por eso, estamos simplemente ordenando los valores propios de acuerdo con cómo disminuye el valor absoluto. (Esto es sólo por notación.) Cada valor propio tiene un vector propio diferente de cero $\mathbf{u}^{(i)}$ y

$$A\mathbf{u}^{(i)} = \lambda_i \mathbf{u}^{(i)} \quad (i = 1, 2, \dots, n) \quad (1)$$

Suponemos que hay un conjunto linealmente independiente de n vectores propios $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(n)}\}$. Es necesariamente una base para \mathbb{C}^n .

Queremos calcular el *único* valor propio de un módulo máximo (el valor propio *dominante*) y el vector propio asociado. Seleccionamos un vector inicial arbitrario, $\mathbf{x}^{(0)} \in \mathbb{C}^n$ y se expresa como una combinación lineal de $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(n)}\}$:

$$\mathbf{x}^{(0)} = c_1 \mathbf{u}^{(1)} + c_2 \mathbf{u}^{(2)} + \cdots + c_n \mathbf{u}^{(n)}$$

En esta ecuación, debemos suponer que $c_1 \neq 0$. Puesto que los coeficientes pueden ser absorbidos por los vectores $\mathbf{u}^{(i)}$ no hay pérdida de generalidad en el supuesto de que

$$\mathbf{x}^{(0)} = \mathbf{u}^{(1)} + \mathbf{u}^{(2)} + \cdots + \mathbf{u}^{(n)} \quad (2)$$

Luego, realizando repetidamente la multiplicación de matriz-vector, usando la matriz A para producir una sucesión de vectores. En concreto, tenemos

$$\left\{ \begin{array}{l} \mathbf{x}^{(1)} = A\mathbf{x}^{(0)} \\ \mathbf{x}^{(2)} = A\mathbf{x}^{(1)} = A^2\mathbf{x}^{(0)} \\ \mathbf{x}^{(3)} = A\mathbf{x}^{(2)} = A^3\mathbf{x}^{(0)} \\ \vdots \\ \mathbf{x}^{(k)} = A\mathbf{x}^{(k-1)} = A^k\mathbf{x}^{(0)} \\ \vdots \end{array} \right.$$

En general, tenemos

$$\mathbf{x}^{(k)} = \mathbf{A}^k \mathbf{x}^{(0)} \quad (k = 1, 2, 3, \dots)$$

Sustituyendo $\mathbf{x}^{(0)}$ en la ecuación (2), obtenemos

$$\begin{aligned}\mathbf{x}^{(k)} &= \mathbf{A}^k \mathbf{x}^{(0)} \\ &= \mathbf{A}^k \mathbf{u}^{(1)} + \mathbf{A}^k \mathbf{u}^{(2)} + \mathbf{A}^k \mathbf{u}^{(3)} + \cdots + \mathbf{A}^k \mathbf{u}^{(n)} \\ &= \lambda_1^k \mathbf{u}^{(1)} + \lambda_2^k \mathbf{u}^{(2)} + \lambda_3^k \mathbf{u}^{(3)} + \cdots + \lambda_n^k \mathbf{u}^{(n)}\end{aligned}$$

usando la ecuación (1). Esto se puede escribir en la forma

$$\mathbf{x}^{(k)} = \lambda_1^k \left[\mathbf{u}^{(1)} + \left(\frac{\lambda_2}{\lambda_1} \right)^k \mathbf{u}^{(2)} + \left(\frac{\lambda_3}{\lambda_1} \right)^k \mathbf{u}^{(3)} + \cdots + \left(\frac{\lambda_n}{\lambda_1} \right)^k \mathbf{u}^{(n)} \right]$$

Puesto que $|\lambda_1| > |\lambda_j|$ para $j > 1$, tenemos $|\lambda_j/\lambda_1| < 1$ y $(\lambda_j/\lambda_1)^k \rightarrow 0$ cuando $k \rightarrow \infty$. Para simplificar la notación, escribimos la ecuación anterior en la forma

$$\mathbf{x}^{(k)} = \lambda_1^k [\mathbf{u}^{(1)} + \varepsilon^{(k)}] \quad (3)$$

donde $\varepsilon^{(k)} \rightarrow 0$ cuando $k \rightarrow \infty$. Sea φ cualquier funcional lineal evaluada compleja en \mathbb{C}^n tal que $\varphi(\mathbf{u}^{(1)}) \neq 0$. Recuerde que φ es un *funcional lineal* si $\varphi(ax + by) = a\varphi(\mathbf{x}) + b\varphi(\mathbf{y})$ para los escalares a y b y los vectores \mathbf{x} y \mathbf{y} . Por ejemplo, $\varphi(\mathbf{x}) = x_j$ para alguna j fija ($1 \leq j \leq n$) es un **funcional lineal**. Ahora, viendo hacia atrás en la ecuación (3), aplicamos φ a ésta:

$$\varphi(\mathbf{x}^{(k)}) = \lambda_1^k [\varphi(\mathbf{u}^{(1)}) + \varphi(\varepsilon^{(k)})]$$

Después, formamos los cocientes r_1, r_2, \dots como sigue:

$$r_k \equiv \frac{\varphi(\mathbf{x}^{(k+1)})}{\varphi(\mathbf{x}^{(k)})} = \lambda_1 \left[\frac{\varphi(\mathbf{u}^{(1)}) + \varphi(\varepsilon^{(k+1)})}{\varphi(\mathbf{u}^{(1)}) + \varphi(\varepsilon^{(k)})} \right] \rightarrow \lambda_1 \quad \text{cuando } k \rightarrow \infty$$

Por lo tanto, podemos calcular el valor propio dominante λ_1 como el límite de la sucesión $\{r_k\}$. Con un poco más de cuidado, podemos obtener un vector propio asociado. En la definición de los vectores $\mathbf{x}^{(k)}$ en la ecuación (2), vemos que nada impide que los vectores crezcan o converjan a cero. La normalización arreglará este problema, como en uno de los seudocódigos que se muestran a continuación.

Algoritmos del método de potencias

Aquí presentamos un seudocódigo para calcular el valor propio dominante y el vector propio asociado para una matriz dada \mathbf{A} . En cada algoritmo, φ es un funcional lineal elegido por el usuario. Por ejemplo, se puede usar $\varphi(\mathbf{x}) = x_1$ (el primer componente del vector).

Algoritmo del método de potencias

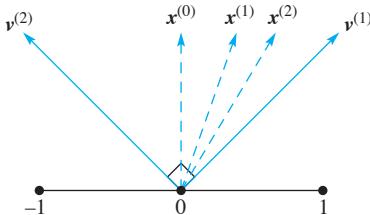
```

integer k, kmax, n; real r
real array (A)1:n×1:n, (x)1:n, (y)1:n
external function φ
output 0, x
for k = 1 to kmax do
    y ← Ax
    r ← φ(y)/φ(x)
    x ← y
    output k, x, r
end do

```

Usamos una matriz simple de 2×2 tal que $A = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$, para dar un ejemplo geométrico del método de potencias como se muestra en la figura 8.3. Claramente, los valores propios son $\lambda_1 = 2$ y $\lambda_2 = 4$ con vectores propios $v^{(1)} = [-1, 1]^T$ y $v^{(2)} = [1, 1]^T$, respectivamente. Iniciando con $x^{(0)} = [0, 1]^T$, el método de potencias repetidamente multiplica la matriz A por un vector. Esto produce una sucesión de vectores $x^{(1)}, x^{(2)}$, etcétera, que se mueven en la dirección del vector propio $v^{(2)}$, que corresponde al valor propio dominante $\lambda_2 = 4$.

FIGURA 8.3
Ejemplo del
método de
potencias
en dos
dimensiones



Podemos modificar fácilmente este algoritmo para producir vectores propios normalizados usando la norma infinita del vector $\|x\|_\infty = \max_{1 \leq j \leq n} |x_j|$, como en el código siguiente:

Algoritmo modificado del método de potencias con normalización

```

integer k, kmax, n; real r
real array (A)1:n×1:n, (x)1:n, (y)1:n
external function φ
output 0, x
for k = 1 to kmax do
    y ← Ax
    r ← φ(y)/φ(x)
    x ← y / \|y\|_∞
    output k, x, r
end do

```

Aceleración de Aitken

A partir de una sucesión $\{r_k\}$, podemos construir otra sucesión $\{s_k\}$ por medio de la fórmula de la aceleración de Aitken

$$s_k = r_k - \frac{(r_k - r_{k-1})^2}{r_k - 2r_{k-1} + r_{k-2}} \quad (k \geq 3)$$

Si la sucesión original $\{r_k\}$ converge a r y si se cumplen ciertas condiciones, entonces, la nueva sucesión $\{s_k\}$ converge a r más rápido que la original. (Para más detalles, véase Kincaid y Cheney [2002].) La eliminación por sustracción puede llegar a arruinar los resultados, por lo que el proceso de aceleración de Aitken se debe suspender inmediatamente después de que los valores aparentemente no cambian.

EJEMPLO 1 Utilice el algoritmo modificado del método de potencias y la aceleración de Aitken para encontrar el valor propio dominante y un vector propio de la matriz dada A , con el vector $x^{(0)}$ y $\varphi(x)$ dados como se muestra:

$$A = \begin{bmatrix} 6 & 5 & -5 \\ 2 & 6 & -2 \\ 2 & 5 & -1 \end{bmatrix}, \quad x^{(0)} = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}, \quad \varphi(x) = x_2$$

Solución Después de codificar y ejecutar el algoritmo modificado del método de potencias con la aceleración de Aitken, obtenemos los siguientes resultados:

$$\begin{aligned} x^{(0)} &= [-1.0000, 1.0000, 1.0000]^T \\ x^{(1)} &= [-1.0000, 0.3333, 0.3333]^T & r_0 &= 2.0000 \\ x^{(2)} &= [-1.0000, -0.1111, -0.1111]^T & r_1 &= -2.0000 \\ x^{(3)} &= [-1.0000, -0.4074, -0.4074]^T & r_2 &= 22.0000 \\ x^{(4)} &= [-1.0000, -0.6049, -0.6049]^T & r_3 &= 8.9091 & s_3 &= 13.5294 \\ x^{(5)} &= [-1.0000, -0.7366, -0.7366]^T & r_4 &= 7.3061 & s_4 &= 7.0825 \\ x^{(6)} &= [-1.0000, -0.8244, -0.8244]^T & r_5 &= 6.7151 & s_5 &= 6.3699 \\ &\vdots &&\vdots &&\vdots \\ x^{(14)} &= [-1.0000, -0.9931, -0.9931]^T & r_{13} &= 6.0208 & s_{13} &= 6.0005 \end{aligned}$$

La sucesión acelerada de Aitken, s_k , converge mucho más rápido que la sucesión $\{r_k\}$. El valor propio dominante real y un vector propio asociado son

$$\lambda_1 = 6 \quad u^{(1)} = [1, 1, 1]^T$$

La codificación del método de potencias modificado es muy simple y dejamos la implementación real como ejercicio. También usamos la *norma infinita* simple para normalizar vectores. Los vectores finales y las estimaciones de los valores propios se presentan con 15 dígitos decimales.

En un problema de este tipo, siempre se debe buscar una verificación independiente de la respuesta pretendida. En este caso, simplemente calculamos Ax para ver si coincide con $s_{14}x$. Unas cuantas de las últimas instrucciones en el código hacen esta comprobación burdamente, tomando s_{14} como probablemente la mejor estimación del valor propio y el último vector x como la mejor estimación de un vector propio. Los resultados después de 14 pasos no son muy precisos. ¡Para mayor precisión, tome 80 pasos!

Método de potencias inverso

Es posible calcular otros valores propios de una matriz mediante modificaciones del método de potencias. Por ejemplo, si A es invertible podemos calcular su valor propio de menor magnitud observando esta equivalencia lógica:

$$Ax = \lambda x \iff x = A^{-1}(\lambda x) \iff A^{-1}x = \frac{1}{\lambda}x$$

Así, el menor valor propio de A en magnitud es la inversa del mayor valor propio de A^{-1} . Lo calculamos al aplicar el método de potencias a A^{-1} y tomando el recíproco del resultado.

Supongamos que sólo hay un único valor propio más pequeño de A . Con nuestro orden normal, este será λ_n :

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_{n-1}| > |\lambda_n| > 0$$

Se deduce que A es invertible. (¿Por qué?) Los valores propios de A^{-1} son λ_j^{-1} para $1 \leq j \leq n$. Por lo tanto, tenemos

$$|\lambda_n^{-1}| > |\lambda_{n-1}^{-1}| \geq \dots \geq |\lambda_1^{-1}| > 0$$

Podemos utilizar el método de potencias de la matriz A^{-1} para calcular su valor propio dominante λ_n^{-1} . El recíproco de este es el valor propio de A que buscamos. Observe que no es necesario calcular A^{-1} , porque la ecuación

$$\mathbf{x}^{(k+1)} = A^{-1} \mathbf{x}^{(k)}$$

es equivalente a la ecuación

$$A\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$$

y el vector $\mathbf{x}^{(k+1)}$ puede ser más fácil de calcular resolviendo el último sistema lineal. Para hacer esto, primero se encuentra la factorización LU de A , es decir, $A = LU$. Luego, repetidamente se actualiza el lado derecho y se resuelve hacia atrás:

$$U\mathbf{x}^{(k+1)} = L^{-1} \mathbf{x}^{(k)}$$

para obtener $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots$

EJEMPLO 2 Calcule el menor valor propio y un vector propio asociado de la matriz siguiente:

$$A = \frac{1}{3} \begin{bmatrix} -154 & 528 & 407 \\ 55 & -144 & -121 \\ -132 & 396 & 318 \end{bmatrix}$$

usando el siguiente vector inicial y la función lineal:

$$\mathbf{x}^{(0)} = [1, 2, 3]^T, \quad \varphi(\mathbf{x}) = x_2$$

Solución Decidimos tomar el camino fácil y usar la inversa de A para la producción de los vectores \mathbf{x} sucesivos. Dejamos la implementación real como un ejercicio. Los cocientes r_k se guardan y una vez que se haya completado, se calculan los valores acelerados de Aitken, s_k . Observe que, al final, queremos el recíproco del cociente limitante. Por lo tanto, es más fácil usar recíprocos en cada paso en el código. Por lo tanto, se ve $r_k = x_2/y_2$ en lugar de y_2/x_2 y estos cocientes deben converger al valor

propio más pequeño de A . Los resultados finales después de 80 pasos son estos:

$$\begin{aligned} \mathbf{x} &= [0.26726101285547, -0.53452256017715, 0.80178375118802]^T \\ s_{80} &= 3.3333333343344 \end{aligned}$$

Podemos dividir cada entrada de \mathbf{x} entre el primer componente y llegar a

$$\mathbf{x} = [1.0, -2.00000199979120, 3.00000266638827]^T$$

El valor propio es realmente $\frac{10}{3}$ y el vector propio debe ser $[1, -2, 3]^T$. La diferencia entre $A\mathbf{x}$ y $s_{80}\mathbf{x}$ es aproximadamente 2.6×10^{-6} . █

Ejemplos con software: método de potencias inverso

Usando software matemático en un pequeño ejemplo,

$$A = \begin{bmatrix} 6 & 5 & -5 \\ 2 & 6 & 2 \\ 2 & 5 & -1 \end{bmatrix} \quad (4)$$

primero podemos obtener A^{-1} y luego usar el método de potencias. (Tenemos que cambiar una entrada en la matriz A del ejemplo 1 para resolver un problema diferente.) Dejamos la implementación del código como un ejercicio. En el código, r es el recíproco de la cantidad r en el método de potencias original. Así, al final del cálculo, r debe ser el valor propio de A que tiene el menor valor absoluto. Después de los 30 pasos dados, encontramos que $r = 0.214$ y $\mathbf{x} = [0.7916, 0.5137, 0.03308]^T$. Como de costumbre, se puede comprobar el resultado calculando de forma independiente $A\mathbf{x}$ y $r\mathbf{x}$, que deben ser iguales. El método que acabamos de exemplificar se llama **método de potencias inverso**. En los grandes ejemplos, los vectores sucesivos no se deben calcular a través de A^{-1} , sino más bien resolviendo la ecuación $A\mathbf{y} = \mathbf{x}$ para \mathbf{y} . En los sistemas de software matemático como Matlab, Maple y Mathematica, esto se puede hacer con una sola instrucción. Alternativamente, se puede obtener la factorización LU de A y resolver $L\mathbf{z} = \mathbf{x}$ y $U\mathbf{y} = \mathbf{z}$.

En este ejemplo, dos valores propios son complejos. Puesto que la matriz es real, deben ser pares conjugados de la forma $\alpha + bi$ y $\alpha - bi$. Tienen la misma magnitud; así, se viola la hipótesis $|\lambda_1| > |\lambda_2|$ necesaria en la prueba de convergencia del método de potencias. ¿Qué sucede cuando el método de potencias se aplica a A ? Los valores de r para $k = 26$ a 30 son $0.76, -53.27, 8.86, 2.69$ y -9.42 . Dejamos la implementación del código como un problema de cómputo.

Método de potencias (inverso) desplazado

Otros valores propios de una matriz (además del más grande y del más pequeño) se pueden calcular usando las equivalencias lógicas siguientes:

$$A\mathbf{x} = \lambda\mathbf{x} \iff (A - \mu I)\mathbf{x} = (\lambda - \mu)\mathbf{x} \iff (A - \mu I)^{-1}\mathbf{x} = \frac{1}{\lambda - \mu}\mathbf{x}$$

Si queremos calcular un valor propio de A que esté cerca de un número dado de m , podemos aplicar el método de potencias inverso a $A - \mu I$ y tomar el recíproco del valor límite de r . Esto debe ser $\lambda - \mu$.

También se puede calcular un valor propio de A que sea el *más alejado* de un número dado μ . Supongamos que para algún valor propio λ_j de la matriz A , tenemos

$$|\lambda_j - \mu| > \varepsilon \quad \text{y} \quad 0 < |\lambda_i - \mu| < \varepsilon \quad \text{para toda } i \neq j$$

Considera la matriz *desplazada* $A - \mu I$. Aplicando el método de potencias a la matriz desplazada $A - \mu I$, calculamos los cocientes r_k que convergen a $\lambda_j - \mu$. Este procedimiento se llama **método de potencias desplazado**.

Si queremos calcular el valor propio de A que sea el *más cercano* a un número dado μ , es necesario hacer una variación de este procedimiento. Supongamos que λ_j es un valor propio de A tal que

$$0 < |\lambda_j - \mu| < \varepsilon \quad \text{y} \quad |\lambda_i - \mu| > \varepsilon \quad \text{para toda } i \neq j$$

Consideremos la matriz *desplazada* $A - \mu I$. Los valores propios de esta matriz son $\lambda_i - \mu$. Aplicando el método de potencias inverso a $A - \mu I$ se obtiene un valor aproximado de $(\lambda_j - \mu)^{-1}$. Podemos usar la inversa explícita de $A - \mu I$ o la factorización LU , $A - \mu I = LU$. Ahora repetidamente resolvemos las ecuaciones

$$(A - \mu I)x^{(k+1)} = x^{(k)}$$

resolviendo en su lugar $Ux^{(k+1)} = L^{-1}x^{(k)}$. Puesto que los cocientes r_k convergen a $(\lambda_j - \mu)^{-1}$, tenemos

$$\lambda_j = \mu + \left(\lim_{k \rightarrow \infty} r_k \right)^{-1} = \mu + \lim_{k \rightarrow \infty} \frac{1}{r_k}$$

Este algoritmo se llama **método de potencias inverso desplazado**.

Ejemplo: método de potencias inverso desplazado

Para ejemplificar el método de potencias inverso desplazado, considere la siguiente matriz:

$$A = \begin{bmatrix} 1 & 3 & 7 \\ 2 & -4 & 5 \\ 3 & 4 & -6 \end{bmatrix}$$

y use software matemático para calcular el valor propio más cercano a -6 . El código que usamos tiene cocientes de y_2/x_2 y por eso estamos esperando la convergencia de estos cocientes a $\lambda + 6$. Después de ocho pasos, tenemos $r = 0.9590$ y $x = [-0.7081, 0.6145, 0.3478]^T$. Por lo tanto, el valor propio debe ser $\lambda = 0.9590 - 6 = -5.0410$. Podemos pedir a Matlab que confirme el valor propio y el vector propio calculando tanto Ax como λx que sean aproximadamente $[3.57, -3.10, -1.75]^T$.

Resumen

(1) Hemos considerado los siguientes métodos para el cálculo de valores propios de una matriz. En el **método de potencias**, nos aproximamos al mayor valor propio λ_1 mediante la generación de una sucesión de puntos mediante la fórmula

$$x^{(k+1)} = Ax^{(k)}$$

y luego formando una sucesión de $r_k = \varphi(x^{(k+1)})/\varphi(x^{(k)})$, donde φ es un funcional lineal. En las circunstancias adecuadas, esta sucesión, r_k , convergerá al mayor valor propio de A .

(2) En el **método de potencias inverso** encontramos el valor propio más pequeño usando el proceso anterior en la inversa de la matriz. El recíproco del mayor valor propio de A^{-1} es el menor valor propio de A . También se puede describir este proceso como uno de cálculo de la sucesión de modo que

$$Ax^{(k+1)} = x^{(k)}$$

(3) En el **método de potencias desplazado** encontramos el valor propio más alejado de un número dado m buscando el mayor valor propio de $A - \mu I$. Esto implica una iteración para producir una sucesión de

$$x^{(k+1)} = (A - \mu I)x^{(k)}$$

(4) En el **método de potencias inverso desplazado** encontramos el valor propio más cercano de m aplicando el método de potencias inverso a $A - \mu I$. Este requiere resolver la ecuación

$$(A - \mu I)x^{(k+1)} = x^{(k)} \quad (A - \mu I = LU)$$

Referencias adicionales

Para lectura suplementaria y estudio, véase Anderson Bai, Bischof, Blackford, Demmel, Dongarra, Du Croz, Greenbaum, Hammarling y McKenney [1999]; Axelsson [1994]; Bai, Demmel, Dongarra, Ruhe y Van der Vorst [2000]; Barrett, Berry, Chan, Demmel, Donato, Dongarra, Eijkhout, Pozo, Romine y Van der Vorst [1994]; Davis [2006]; Dekker y Hoffmann [1989]; Dekker, Hoffmann y Potma [1997]; Demmel [1997]; Dongarra et al. [1990]; Elman, Silvester y Wathen [2004]; Fox [1967]; Gautschi [1997]; Greenbaum [1997]; Hageman y Young [1981]; Heroux, Raghavan y Simon [2006]; Jennings [1977]; Kincaid y Young [1979, 2000]; Lynch [2004]; Meurant [2006]; Noble y Daniel [1988]; Ortega [1990b]; Parlett [2000]; Saad [2003]; Schewchuck [1994]; Southwell [1946]; Stewart [1973]; Trefethen y Bau [1997]; Van der Vorst [2003]; Watkins [1991]; Wilkinson [1988]; y Young [1971].

Problemas 8.4

1. Sea $A = \begin{bmatrix} 5 & 2 \\ 4 & 7 \end{bmatrix}$. El método de potencias se ha aplicado a la matriz A . El resultado es una larga lista de vectores que parecen establecer un vector de la forma $[h, 1]^T$, donde $|h| < 1$. ¿Cuál es el valor propio más grande, aproximadamente, en términos de ese número h ?

- a. $4h + 7$ b. $5h + 2$ c. $1/h$ d. $5h + 4$ e. Ninguno de estos.

2. ¿Cuál es la salida esperada del siguiente seudocódigo?

```
integer n, kmax;  real r
real array (A-1)1:n x 1:n, (x)1:n, (y)1:n
for k = 1 to 30 do
    y ← A-1x
    r ← y1/x1  (primeras componentes de y y x)
    x ← y / ||y||
    output r, x
end do
```

- a. r es el valor propio de A de magnitud más grande y x es un vector propio asociado.
 b. $r = 1/\lambda$, donde λ es el valor propio más pequeño de A y x es tal que $Ax = \lambda x$.
 c. Un vector x tal que $Ax = rx$, donde r es el valor propio de A que tiene la magnitud más pequeña.
 d. r es el más grande (en magnitud) valor propio de A y x es un correspondiente vector propio de A .
 e. Ninguna de estas.

3. Describa brevemente cómo calcular lo siguiente:

- a. El valor propio dominante y el vector propio asociado.
 b. El siguiente valor propio dominante y el vector propio asociado.
 c. El menor valor propio dominante y el vector propio asociado.
 d. Un valor propio distinto del dominante o el menor valor propio dominante y sus vectores propios asociados.

4. Sea $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$. Realice varias iteraciones del método de potencias, iniciando con $x^{(0)} = (1, 1, 1)$. ¿Cuál es el objetivo de este procedimiento?

5. Sea $B = A - 4I = \begin{bmatrix} -2 & -1 & 0 \\ -1 & -2 & -1 \\ 0 & -1 & -2 \end{bmatrix}$. Realice varias iteraciones del método de potencias aplicado a B , iniciando con $x^{(0)} = (1, 1, 1)$. ¿Cuál es el objetivo de este procedimiento?

6. Sea $C = A^{-1} = \frac{1}{4} \begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix}$. Realice pocas iteraciones del método de potencias aplicado a C , iniciando con $x^{(0)} = (1, 1, 1)$. ¿Cuál es el objetivo de este procedimiento?

7. El cociente de Rayleigh es la expresión $\langle x, x \rangle_A / \langle x, x \rangle = x^T Ax / x^T x$. ¿Cómo se puede usar el cociente de Rayleigh cuando $Ax = \lambda x$?

Problemas de cómputo 8.4

1. Use el método de potencias, el método de potencias inverso y sus formas desplazadas, así como la aceleración de Aitken para encontrar algún o todos los valores propios de las matrices siguientes:

a. $\begin{bmatrix} 5 & 4 & 1 & 1 \\ 4 & 5 & 1 & 1 \\ 1 & 1 & 4 & 2 \\ 1 & 1 & 2 & 4 \end{bmatrix}$

b. $\begin{bmatrix} 2 & 3 & 4 \\ 7 & -1 & 3 \\ 1 & -1 & 5 \end{bmatrix}$

c. $\begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}$

2. Repita los ejemplos en esta sección utilizando Matlab, Maple o Mathematica.
3. Modifique y pruebe el seudocódigo para el método de potencias para normalizar el vector de manera que la componente más grande sea siempre 1 en la norma infinita. Este procedimiento da el vector y el valor propios sin tener que calcular un funcional lineal.
4. Encuentre los valores propios de la matriz

$$\mathbf{A} = \begin{bmatrix} -57 & 192 & 148 \\ 20 & -53 & -44 \\ -48 & 144 & 115 \end{bmatrix}$$

que están cerca de $-4, 2$ y 8 usando el método de potencias inverso.

5. Usando software matemático como Matlab, Maple o Mathematica, escriba y ejecute el código para la implementación de los métodos en la sección 8.4. Compruebe que los resultados son coherentes con los descritos en el libro.
- a. El ejemplo 1 usando el método de potencias modificado.
 - b. El ejemplo 2 usando el método de potencias inverso con aceleración de Aitken.
 - c. La matriz (4) usando el método de potencias inverso.
 - d. La matriz (5) usando el método de potencias desplazado.

6. Considere la matriz $\mathbf{A} = \begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 1 & 1 & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{4} & 2 \end{bmatrix}$

- a. Use el método de potencias normalizado iniciando con $\mathbf{x}^{(0)} = [1, 1, 1]^T$ y determine el valor propio dominante y el vector propio de la matriz \mathbf{A} .
- b. Repita, empezando con el valor inicial $\mathbf{x}^{(0)} = [-0.64966116, 0, 74822116, 0]^T$. Explique los resultados. Véase Ralston [1965, pp. 475-476].

7. Sea $\mathbf{A} = \begin{bmatrix} -4 & 14 & 0 \\ -5 & 13 & 0 \\ -1 & 0 & 2 \end{bmatrix}$. Codifique y aplique cada uno de lo siguiente:

- a. El algoritmo modificado de potencias iniciando con $\mathbf{x}^{(0)} = [1, 1, 1]^T$, así como el proceso de aceleración de Aitken.
- b. El algoritmo de potencias inverso.
- c. El algoritmo de potencias desplazado.
- d. El algoritmo de potencias inverso desplazado.

8. (Continuación) Sea $\mathbf{B} = \begin{bmatrix} 4 & -1 & 1 \\ -1 & 3 & -2 \\ 1 & -2 & 3 \end{bmatrix}$. Repita el problema anterior iniciando con $\mathbf{x}^{(0)} = [1, 0, 0]^T$.

9. (Continuación) Sea $\mathbf{C} = \begin{bmatrix} -8 & -5 & 8 \\ 6 & 3 & -8 \\ -3 & 1 & 9 \end{bmatrix}$. Use $\mathbf{x}^{(0)} = [1, 1, 1]^T$. Repita el problema anterior iniciando con $\mathbf{x}^{(0)} = [1, 0, 0]^T$.

- 10.** Por medio del método de potencias, encuentre un valor propio y un vector propio asociado con estas matrices de los libros históricos de Fox [1957] y Wilkinson [1965]. Verifique los resultados mediante el uso de software matemático como Matlab, Maple o Mathematica.

a. $\begin{bmatrix} 0.9901 & 0.002 \\ -0.0001 & 0.9904 \end{bmatrix}$ iniciando con $x^{(0)} = [1, 0.9]^T$

b. $\begin{bmatrix} 8 & -1 & -5 \\ -4 & 4 & -2 \\ 18 & -5 & -7 \end{bmatrix}$ iniciando con $x^{(0)} = [1, 0.8, 1]^T$

c. $\begin{bmatrix} 1 & 1 & 3 \\ 1 & -2 & 1 \\ 3 & 1 & 3 \end{bmatrix}$ iniciando con $x^{(0)} = [1, 1, 1]^T$

d. $\begin{bmatrix} -2 & -1 & 4 \\ 2 & 1 & -2 \\ -1 & -1 & 3 \end{bmatrix}$ iniciando con $x^{(0)} = [3, 1, 2]^T$ sin normalización y con normalización

- 11.** Encuentre todos los valores propios y los vectores propios asociados de estas matrices de Fox [1957] y Wilkinson [1965] por medio del método de potencias y las variaciones de este. Compruebe sus resultados usando software matemático como Matlab, Maple o Mathematica.

a. $\begin{bmatrix} 2 & 1 \\ 4 & 2 \end{bmatrix}$

b. $\begin{bmatrix} 0.4812 & 0.0023 \\ -0.0024 & 0.4810 \end{bmatrix}$

c. $\begin{bmatrix} 1 & 1 & 0 \\ -1 + 10^{-8} & 3 & 0 \\ 0 & 1 & 1 \end{bmatrix}$

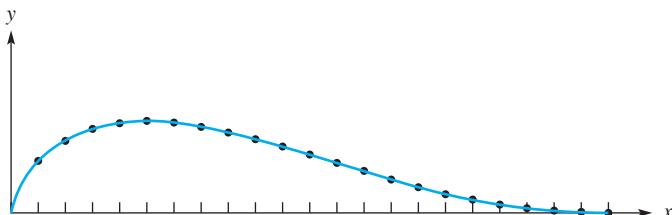
d. $\begin{bmatrix} 5 & -1 & -2 \\ -1 & 3 & -2 \\ -2 & -2 & 5 \end{bmatrix}$

e. $\begin{bmatrix} 0.987 & 0.400 & -0.487 \\ -0.079 & 0.500 & -0.479 \\ 0.082 & 0.400 & 0.418 \end{bmatrix}$

Aproximación por funciones spline

Experimentando en un túnel de viento, con ensayo y error se construye un perfil aerodinámico para que tenga ciertas características deseadas. La sección transversal de la superficie de sustentación se dibuja como una curva en papel cuadriculado (figura 9.1). Para estudiar este perfil aerodinámico con métodos analíticos o para fabricarlo, es esencial tener una fórmula para esta curva. Para obtenerla, primero obtenga las coordenadas de un conjunto finito de puntos en la curva. Entonces se puede construir una curva suave llamada *spline de interpolación cúbica* para que coincida con estos datos. Este capítulo trata de funciones de spline polinomiales generales y cómo se pueden utilizar en diversos problemas numéricos, tales como los del problema de ajuste de datos que acabamos de describir.

FIGURA 9.1
Sección
transversal
de la
superficie de
sustentación



9.1 Splines de primer y segundo grado

La historia de las funciones spline se basa en el trabajo de los dibujantes, quienes con frecuencia deben trazar una curva suave entre los puntos de inflexión en un dibujo. Este proceso se denomina *carenado* y se puede lograr con una serie de dispositivos con fines específicos, como la *curva francesa*, hecha de plástico y que presenta una serie de curvas de curvatura diferente para que el dibujante seleccione. También se utilizan largas tiras de madera, que se hacen pasar a través de los *puntos de control* con pesas colocadas horizontalmente sobre la mesa del dibujante y unidas a las tiras. Las pesas se llamaban *ducks* y las tiras de madera se llamaban *splines*, aún en 1891. La naturaleza elástica de las tiras de madera les permitía doblarse sólo un poco sin dejar de pasar por los puntos dados. La madera, en efecto, resolvía una ecuación diferencial y minimizaba la energía de deformación. Esto último se conoce como una función simple de la curvatura. La teoría matemática de las curvas debe mucho a los primeros investigadores, Isaac Schoenberg en particular en los años 1940 y 1950. Otros nombres importantes asociados con los primeros desarrollos del tema (es decir, antes de 1964) son Garrett Birkhoff, C. de Boor, J. H. Ahlberg, E. N. Nilson,

H. Garabedian, R. S. Johnson, F. Landis, A. Whitney, J. L. Walsh y J. C. Holladay. El primer libro que presentó una exposición sistemática de la teoría de splines fue el de Ahlberg, Nilson y Walsh [1967].

Spline de primer grado

Una **función spline** es una función que consta de partes de un polinomio unidas con ciertas condiciones de suavidad. Un ejemplo sencillo es la función **poligonal** (o spline de grado 1), cuyas piezas son polinomios lineales que se unen para lograr la continuidad, como se muestra en la figura 9.2. Los puntos, t_0, t_1, \dots, t_n , en los que la función cambia su carácter, se denominan **nudos** en la teoría de splines. Así, en la figura 9.2 se muestra una función spline con ocho nudos.

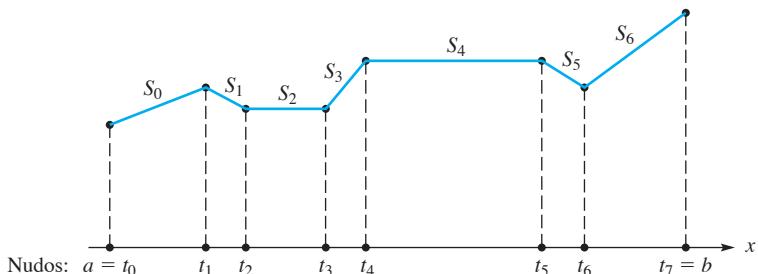


FIGURA 9.2
Función spline
de primer grado

Esta función parece algo complicada cuando se define en términos explícitos. Nos vemos obligados a escribir

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases} \quad (1)$$

donde

$$S_i(x) = a_i x + b_i \quad (2)$$

debido a que cada pieza de $S(x)$ es un polinomio lineal. Tal función $S(x)$ es **lineal por partes**. Si los nudos t_0, t_1, \dots, t_n eran dados y si los coeficientes, $a_0, b_0, a_1, b_1, \dots, a_{n-1}, b_{n-1}$ eran también conocidos, entonces para evaluar $S(x)$ en una x dada se procedía a determinar el intervalo que contenía a x y después a usar la función lineal apropiada para ese intervalo.

Si la función S definida por la ecuación (1) es continua, la llamamos un **spline de primer grado**. Se caracteriza por los siguientes tres propiedades.

DEFINICIÓN 1

Spline de primer grado

Una función S se llama un **spline de primer grado** si:

1. El dominio de S es un intervalo $[a, b]$.
2. S es continua en $[a, b]$.
3. Hay una partición del intervalo $a = t_0 < t_1 < \dots < t_n = b$ tal que S es un polinomio lineal en cada subintervalo $[t_i, t_{i+1}]$.

Fuera del intervalo $[a, b]$, $S(x)$ se define generalmente como la misma función a la izquierda de a porque está en el subintervalo del extremo izquierdo $[t_0, t_1]$ e igual a la derecha de b pues, está en el extremo derecho en el subintervalo $[t_{n-1}, t_n]$, a saber, $S(x) = S_0(x)$ cuando $x < a$ y $S(x) = S_{n-1}(x)$ cuando $x > b$.

La **continuidad** de una función f en un punto s se puede definir por la condición

$$\lim_{x \rightarrow s^+} f(x) = \lim_{x \rightarrow s^-} f(x) = f(s)$$

Aquí, $\lim_{x \rightarrow s^+}$ significa que el límite toma valores de x que convergen a s por arriba de s , es decir, $(x - s)$ es positivo para todos los valores de x . Del mismo modo, $\lim_{x \rightarrow s^-}$ significa que los valores de x convergen a s por debajo de s .

EJEMPLO 1 Determine si esta función es una función spline de primer grado:

$$S(x) = \begin{cases} x & x \in [-1, 0] \\ 1 - x & x \in (0, 1) \\ 2x - 2 & x \in [1, 2] \end{cases}$$

Solución La función es obviamente lineal por partes, pero no es un spline de primer grado, ya que es discontinua en $x = 0$. Observe que $\lim_{x \rightarrow 0^+} S(x) = \lim_{x \rightarrow 0} (1 - x) = 1$, mientras que $\lim_{x \rightarrow 0^-} S(x) = \lim_{x \rightarrow 0} x = 0$. ■

Las funciones spline de primer grado se pueden usar en interpolación. Suponga que se da la siguiente tabla de valores de una función:

x	t_0	t_1	\cdots	t_n
y	y_0	y_1	\cdots	y_n

No hay pérdida de la generalidad al suponer que $t_0 < t_1 < \cdots < t_n$ porque esto es sólo una cuestión de etiquetar los nudos.

La tabla se puede representar con un conjunto de $n + 1$ puntos en el plano, $(t_0, y_0), (t_1, y_1), \dots, (t_n, y_n)$ y estos puntos tienen abscisas distintas. Por lo tanto, podemos trazar una línea poligonal a través de los puntos sin tener que dibujar un segmento *vertical*. Esta línea poligonal es la gráfica de una función, la cual obviamente es una función spline de primer grado. ¿Cuáles son las ecuaciones de cada uno los segmentos que componen este gráfico?

Con referencia a la figura 9.3 y utilizando la forma punto-pendiente de una recta se obtiene

$$S_i(x) = y_i + m_i(x - t_i) \quad (3)$$

en el intervalo $[t_i, t_{i+1}]$, donde m_i es la pendiente de la recta y está por lo tanto dada por la fórmula

$$m_i = \frac{y_{i+1} - y_i}{t_{i+1} - t_i}$$

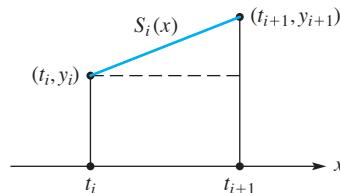


FIGURA 9.3
Spline de primer grado: lineal
 $S_i(x)$

Observe que la función S que estamos construyendo tiene $2n$ parámetros: los n coeficientes a_i y las n constantes b_i en la ecuación (2). Por otro lado, exactamente se están imponiendo $2n$ condiciones, ya que cada función constituyente S_i debe interpolar los datos en los extremos de su subintervalo. Así, el número de parámetros es igual al número de condiciones. Para splines de grado superior, encontraremos un incongruencia entre estos dos números; el spline de grado k tendrá $k - 1$ parámetros libres para que los utilicemos como queramos en el problema de interpolación en los nudos.

La forma de la ecuación (3) es mejor que la de la ecuación (2) para la evaluación práctica de $S(x)$ porque algunas de las cantidades $x - t_i$ se deben calcular en cualquier caso, simplemente para determinar qué subintervalo contiene a x . Si $t_0 \leq x \leq t_n$ entonces el intervalo de $[t_i, t_{i+1}]$ que contiene a x se caracteriza por el hecho de que $x - t_i$ es la primera de las cantidades $x - t_{n-1}, x - t_{n-2}, \dots, x - t_0$ que es no negativa.

La siguiente es un procedimiento de función que utiliza $n + 1$ valores de tabla (t_i, y_i) en arreglos lineales (t_i) y (y_i) , suponiendo que $a = t_0 < t_1 < \dots < t_n = b$. Dado un valor x , la rutina regresa $S(x)$ usando las ecuaciones (1) y (3). Si $x < t_0$, entonces $S(x) = y_0 + m_0(x - t_0)$; si $x > t_n$, entonces $S(x) = y_{n-1} + m_{n-1}(x - t_{n-1})$.

```

real function SplineI(n,(ti),(yi),x)
integer i,n; real x; real array (ti)0:n,(yi)0:n
for i = n - 1 to 0 step -1 do
    if x - ti  $\geqq 0$  then exit loop
end for
SplineI  $\leftarrow y_i + (x - t_i)[(y_{i+1} - y_i)/(t_{i+1} - t_i)]$ 
end function SplineI

```

Módulo de continuidad

Para evaluar la **bondad del ajuste** cuando interpolemos una función con un spline de primer grado, es conveniente disponer de algo que se llama *módulo de continuidad* de una función f . Supongamos que f se define en un intervalo $[a, b]$. El **módulo de continuidad** de f es

$$\omega(f; h) = \sup\{|f(u) - f(v)| : a \leq u \leq v \leq b, |u - v| \leq h\}$$

Aquí, sup es el **supremo**, que es el límite superior mínimo del conjunto dado de números reales. La cantidad $\omega(f; h)$ mide cuánto puede cambiar f en un pequeño intervalo de ancho h . Si f es continua en $[a, b]$, entonces es uniformemente continua, y $\omega(f; h)$ tenderá a cero cuando h tiende a cero. Si f no es continua, $\omega(f; h)$ no tenderá a cero. Si f es derivable en (a, b) (además de ser continua en $[a, b]$) y si $f'(x)$ está limitada en (a, b) , entonces el teorema del valor medio se puede utilizar para obtener una estimación del módulo de continuidad: si u y v se describen en la definición de $\omega(f; h)$, entonces

$$|f(u) - f(v)| = |f'(c)(u - v)| \leq M_1|u - v| \leq M_1h$$

Aquí, M_1 denota el máximo de $|f'(x)|$ cuando x corre sobre (a, b) . Por ejemplo, si $f(x) = x^3$ y $[a, b] = [1, 4]$, entonces encontramos que $\omega(f; h) \leq 48h$.

TEOREMA 1**Teorema de exactitud del polinomio de primer grado**

Si p es el polinomio de primer grado que interpola una función f en los extremos de un intervalo $[a, b]$, entonces con $h = b - a$, tenemos

$$|f(x) - p(x)| \leq \omega(f; h) \quad (a \leq x \leq b)$$

Demostración La función lineal p está dada explícitamente por la fórmula

$$p(x) = \left(\frac{x - a}{b - a} \right) f(b) + \left(\frac{b - x}{b - a} \right) f(a)$$

Por tanto,

$$f(x) - p(x) = \left(\frac{x - a}{b - a} \right) [f(x) - f(b)] + \left(\frac{b - x}{b - a} \right) [f(x) - f(a)]$$

Entonces tenemos

$$\begin{aligned} |f(x) - p(x)| &\leq \left(\frac{x - a}{b - a} \right) |f(x) - f(b)| + \left(\frac{b - x}{b - a} \right) |f(x) - f(a)| \\ &\leq \left(\frac{x - a}{b - a} \right) \omega(f; h) + \left(\frac{b - x}{b - a} \right) \omega(f; h) \\ &= \left[\left(\frac{x - a}{b - a} \right) + \left(\frac{b - x}{b - a} \right) \right] \omega(f; h) = \omega(f; h) \end{aligned}$$

A partir de este resultado básico es fácil demostrar el siguiente teorema, simplemente aplicando la desigualdad básica a cada subintervalo.

TEOREMA 2**Teorema de exactitud del spline de primer grado**

Sea p un spline de primer grado con nudos $a = x_0 < x_1 < \dots < x_n = b$. Si p interpola una función f en estos nudos, entonces con $h = \max_i (x_i - x_{i-1})$ tenemos

$$|f(x) - p(x)| \leq \omega(f; h) \quad (a \leq x \leq b)$$

Si f' o f'' existen y son continuas, entonces se puede decir más, a saber,

$$\begin{aligned} |f(x) - p(x)| &\leq M_1 \frac{h}{2} \quad (a \leq x \leq b) \\ |f(x) - p(x)| &\leq M_2 \frac{h^2}{8} \quad (a \leq x \leq b) \end{aligned}$$

En estos cálculos, M_1 es el valor máximo de $|f'(x)|$ en el intervalo y M_2 es el máximo de $|f''(x)|$.

El primer teorema nos dice que si se insertan más nudos de tal manera que la separación máxima h tiende a cero, entonces el correspondiente spline de primer grado converge uniformemente a f . Recordemos que este tipo de resultado hace notablemente falta en la teoría de interpolación polinomial. En esta situación, elevar el grado y tomar nodos para llenar el intervalo no garantiza necesariamente que se produzca la convergencia de una función continua arbitraria (véase la sección 4.2).

Splines de segundo grado

Los splines de grado mayor que 1 son más complicados. Ahora abordamos el spline cuadrático. Vamos a usar la letra Q para recordar que estamos considerando funciones cuadráticas por partes. Una función Q es un **spline de segundo grado** si tiene las siguientes propiedades.

DEFINICIÓN 2

Spline de segundo grado

Una función Q se llama un **spline de segundo grado** si:

1. El dominio de Q es un intervalo $[a, b]$.
2. Q y Q' son continuas en $[a, b]$.
3. Hay t_i puntos (llamados **nudos**) tales que $a = t_0 < t_1 < \dots < t_n = b$ y Q es un polinomio de grado a lo más 2 en cada subintervalo $[t_i, t_{i+1}]$.

En resumen, un spline cuadrático es una función cuadrática por partes continuamente derivable, donde *cuadrática* incluye todas las combinaciones lineales de las funciones base $x \mapsto 1, x, x^2$.

EJEMPLO 2 Determine si la siguiente función es un spline cuadrático:

$$Q(x) = \begin{cases} x^2 & (-10 \leq x \leq 0) \\ -x^2 & (0 \leq x \leq 1) \\ 1 - 2x & (1 \leq x \leq 20) \end{cases}$$

Solución La función es obviamente cuadrática por partes. Si Q y Q' son continuas en los nudos interiores se pueden determinar en la forma siguiente:

$$\begin{array}{lll} \lim_{x \rightarrow 0^-} Q(x) = \lim_{x \rightarrow 0^-} x^2 = 0 & \lim_{x \rightarrow 0^+} Q(x) = \lim_{x \rightarrow 0^+} (-x^2) = 0 \\ \lim_{x \rightarrow 1^-} Q(x) = \lim_{x \rightarrow 1^-} (-x^2) = -1 & \lim_{x \rightarrow 1^+} Q(x) = \lim_{x \rightarrow 1^+} (1 - 2x) = -1 \\ \lim_{x \rightarrow 0^-} Q'(x) = \lim_{x \rightarrow 0^-} 2x = 0 & \lim_{x \rightarrow 0^+} Q'(x) = \lim_{x \rightarrow 0^+} (-2x) = 0 \\ \lim_{x \rightarrow 1^-} Q'(x) = \lim_{x \rightarrow 1^-} (-2x) = -2 & \lim_{x \rightarrow 1^+} Q'(x) = \lim_{x \rightarrow 1^+} (-2) = -2 \end{array}$$

En consecuencia, $Q(x)$ es un spline cuadrático. ■

Interpolación del spline cuadrático $Q(x)$

Los splines cuadráticos no se utilizan en aplicaciones con tanta frecuencia como los splines cúbicos naturales, que se desarrollan en la siguiente sección. Sin embargo, las deducciones de la interpolación de splines cuadráticos y cúbicos son tan similares que entender la teoría de los sencillos splines de segundo grado permite crear fácilmente la teoría más complicada de los splines de tercer grado. Queremos hacer hincapié en que los splines cuadráticos rara vez se utilizan en interpolación y el análisis que aquí se presenta es sólo como preparación para el estudio de splines de orden superior, que se utilizan en muchas aplicaciones.

Continuando ahora con el problema de interpolación, suponga que se ha dado una tabla de valores:

x	t_0	t_1	t_2	\cdots	t_n
y	y_0	y_1	y_2	\cdots	y_n

Suponemos que los puntos t_0, t_1, \dots, t_n , que imaginamos como **nodos** en el problema de interpolación, son también los nudos para construir la función spline. Más tarde se analizará otra función de interpolación cuadrática en la que los nodos de interpolación son diferentes de los nudos.

Un spline cuadrático, como acabamos de describir, consta de n funciones cuadráticas separadas $x \mapsto a_i x^2 + b_i x + c_i$, una para cada subintervalo creado por los $n + 1$ nudos. Así pues, comenzamos con $3n$ coeficientes. En cada subintervalo $[t_i, t_{i+1}]$, la función spline cuadrática Q_i debe satisfacer las condiciones de interpolación $Q_i(t_i) = y_i$ y $Q_i(t_{i+1}) = y_{i+1}$. Puesto que hay n subintervalos de estos, se imponen $2n$ condiciones. La continuidad de Q no agrega condiciones adicionales. (¿Por qué?) Sin embargo, la continuidad de Q' en cada uno de los nudos interiores agrega $n - 1$ condiciones. Por lo tanto, tenemos que $2n + n - 1 = 3n - 1$ condiciones o una condición por debajo de las $3n$ condiciones requeridas. Hay muchas maneras de imponer esta condición adicional, por ejemplo, $Q'(t_0) = 0$ o $Q''_0 = 0$.

Ahora se deducen las ecuaciones para la interpolación del spline cuadrático, $Q(x)$. El valor de $Q'(t_0)$ se establece como condición adicional. Buscamos una función cuadrática por partes

$$Q(x) = \begin{cases} Q_0(x) & (t_0 \leq x \leq t_1) \\ Q_1(x) & (t_1 \leq x \leq t_2) \\ \vdots & \vdots \\ Q_{n-1}(x) & (t_{n-1} \leq x \leq t_n) \end{cases} \quad (4)$$

que es continuamente derivable en todo el intervalo $[t_0, t_n]$ y que interpola la tabla; es decir, $Q(t_i) = y_i$ para $0 \leq i \leq n$.

Puesto que Q' es continua, podemos poner $z_i \equiv Q'(t_i)$. En este momento, no sabemos los valores correctos de z_i ; sin embargo, el siguiente debe ser la fórmula para Q_i :

$$Q_i(x) = \frac{z_{i+1} - z_i}{2(t_{i+1} - t_i)}(x - t_i)^2 + z_i(x - t_i) + y_i \quad (5)$$

Para ver que esto es correcto, compruebe que $Q_i(t_i) = y_i$, $Q'_i(t_i) = z_i$, y $Q''_i(t_{i+1}) = z_{i+1}$. Estas tres condiciones definen la función Q únicamente en $[t_i, t_{i+1}]$, como la da la ecuación (5).

Ahora, para que la función de interpolación del spline cuadrático Q sea continua e interpole la tabla de datos es necesario y suficiente que $Q_i(t_{i+1}) = y_{i+1}$ para $i = 0, 1, \dots, n - 1$ en la ecuación (5). Cuando esta ecuación se escribe en detalle y se simplifica, el resultado es

$$z_{i+1} = -z_i + 2 \left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right) \quad (0 \leq i \leq n - 1) \quad (6)$$

Esta ecuación se puede utilizar para obtener al vector $[z_0, z_1, \dots, z_n]^T$, empezando con un valor arbitrario para z_0 . Resumimos con un algoritmo:

■ ALGORITMO 1 Interpolación de spline cuadrático en los nudos

- Determine $[z_0, z_1, \dots, z_n]^T$ seleccionando z_0 arbitrariamente y calculando z_1, z_2, \dots, z_n recursivamente mediante la fórmula (6).
- La función de interpolación del spline cuadrático Q está dada por las fórmulas (4) y (5).

EJEMPLO 3 Para los cinco puntos $(0, 8), (1, 12), (3, 2), (4, 6), (8, 0)$, construya el spline lineal S y el spline cuadrático Q .

Solución La figura 9.4 muestra gráficamente estas dos curvas spline de bajo orden. Se adaptan mejor que los polinomios de interpolación de la figura 4.6 (p. 154) sin considerar la reducción en las oscilaciones.

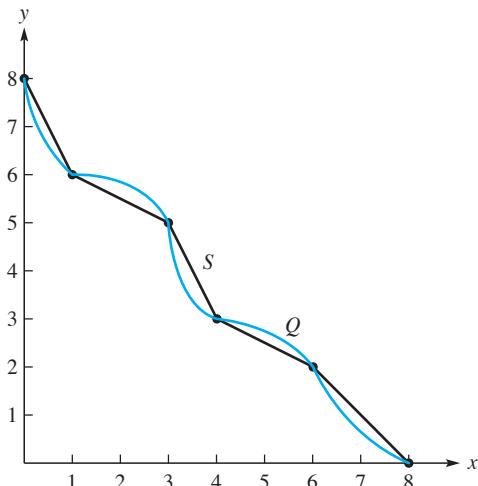


FIGURA 9.4
Funciones spline de primer y segundo grado

Spline cuadrático de Subbotin

Un proceso de aproximación útil, propuesto primeramente por Subbotin [1967], consiste en la interpolación con splines *cuadráticos*, donde los nudos de interpolación se eligen como los primeros y los últimos nudos y los puntos medios entre los nudos. Recuerde que los **nudos** se definen como los puntos donde se permite a la función spline cambiar de forma de un polinomio a otro. Los **nudos** son los puntos donde se especifican los valores del spline. En la función de spline cuadrático de Subbotin, hay $n + 2$ condiciones de interpolación y $2(n - 1)$ las condiciones de continuidad de Q y Q' . Por tanto, tenemos el número exacto de condiciones necesarias, $3n$, para definir la función spline cuadrático completamente.

Aquí esbozamos la teoría, dejando los detalles para que usted los complete. Supongamos que los nudos $a = t_0 < t_1 < \dots < t_n = b$ se han dado; hagamos que los nudos sean los puntos

$$\begin{cases} \tau_0 = t_0 & \tau_{n+1} = t_n \\ \tau_i = \frac{1}{2}(t_i + t_{i-1}) & (1 \leq i \leq n) \end{cases}$$

Buscamos una función de spline cuadrático Q que tenga los nudos dados y que tome los valores dados en los nudos:

$$Q(\tau_i) = y_i \quad (0 \leq i \leq n + 1)$$

como en la figura 9.5. Los nudos crean n subintervalos y en cada uno de ellos Q puede ser un polinomio diferente de segundo grado. Digamos que en $[t_i, t_{i+1}]$, Q es igual al polinomio de segundo grado Q_i . Como Q es un spline cuadrático, éste y su primera derivada deben ser continuos. Por lo tanto, $z_i \equiv Q'(t_i)$ está bien definido, aunque todavía no conocemos sus valores. Es fácil ver que

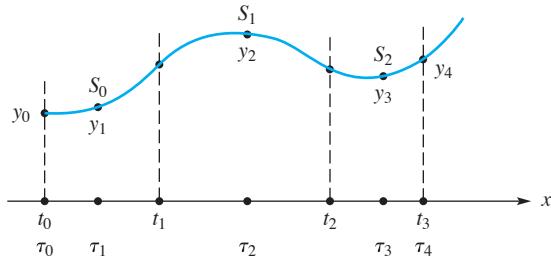


FIGURA 9.5
Splines cuadráticos de Subbotin ($t_0 = \tau_0, t_3 = \tau_4$)

en $[t_i, t_{i+1}]$, nuestro polinomio cuadrático se puede representar en la forma

$$Q_i(x) = y_{i+1} + \frac{1}{2}(z_{i+1} + z_i)(x - \tau_{i+1}) + \frac{1}{2h_i}(z_{i+1} - z_i)(x - \tau_{i+1}) \quad (7)$$

en el que $h_i = t_{i+1} - t_i$. Para verificar la exactitud de la ecuación (7), debemos comprobar que $Q_i(\tau_{i+1}) = y_{i+1}$, $Q'_i(t_i) = z_i$ y $Q'_i(t_{i+1}) = z_{i+1}$. Cuando las partes de polinomio Q_0, Q_1, \dots, Q_{n-1} se unen para formar Q , el resultado puede ser discontinuo. Por lo tanto, imponemos las condiciones de continuidad en los nudos interiores:

$$\lim_{x \rightarrow t_i^-} Q_{i-1}(x) = \lim_{x \rightarrow t_i^+} Q_i(x) \quad (1 \leq i \leq n-1)$$

Usted debe realizar este análisis, que conduce a

$$h_{i-1}z_{i-1} + 3(h_{i-1} + h_i)z_i + h_iz_{i+1} = 8(y_{i+1} - y_i) \quad (1 \leq i \leq n-1) \quad (8)$$

Se deben tambien imponer la primera y la última condición de interpolación:

$$Q(\tau_0) = y_0 \quad Q(\tau_{n+1}) = y_{n+1}$$

Estas dos ecuaciones conducen a

$$\begin{aligned} 3h_0z_0 + h_0z_1 &= 8(y_1 - y_0) \\ h_{n-1}z_{n-1} + 3h_{n-1}z_n &= 8(y_{n+1} - y_n) \end{aligned}$$

El sistema de ecuaciones que gobiernan al vector $z = [z_0, z_1, \dots, z_n]^T$ se puede escribir en forma matricial

$$\begin{aligned} &\left[\begin{array}{cccccc|c} 3h_0 & h_0 & & & & & z_0 \\ h_0 & 3(h_0 + h_1) & h_1 & & & & z_1 \\ & h_1 & 3(h_1 + h_2) & h_2 & & & z_2 \\ & & \ddots & \ddots & \ddots & & \vdots \\ & & & h_{n-2} & 3(h_{n-2} + h_{n-1}) & h_{n-1} & z_{n-1} \\ & & & & h_{n-1} & 3h_{n-1} & z_n \end{array} \right] \\ &= 8 \begin{bmatrix} y_1 - y_0 \\ y_2 - y_1 \\ y_3 - y_2 \\ \vdots \\ y_n - y_{n-1} \\ y_{n+1} - y_n \end{bmatrix} \end{aligned}$$

Este sistema de $n + 1$ ecuaciones con $n + 1$ incógnitas puede ser convenientemente resuelto con el procedimiento *Tri* del capítulo 7. Después de que se ha obtenido el vector \mathbf{z} , los valores de $Q(x)$ se pueden calcular a partir de la ecuación (7). La escritura de un código adecuado para realizar este método de interpolación se deja como un proyecto de programación.

Resumen

(1) Se nos han dado $n + 1$ pares de puntos (t_i, y_i) , con nudos distintos $a = t_0 < t_1 < \dots < t_{n-1} < t_n = b$ en el intervalo $[a, b]$. Una función spline de primer grado S es un polinomio lineal definido por partes en el intervalo $[a, b]$ de modo que es continua. Tiene la forma

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

donde

$$S_i(x) = y_i + \left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right) (x - t_i)$$

en el intervalo $[t_i, t_{i+1}]$. Obviamente, $S(x)$ es continua, ya que $S_{i-1}(t_i) = S_i(t_i) = y_i$ para $1 \leq i \leq n$.

(2) Una **función spline de segundo grado** Q es un polinomio cuadrático definido por partes con Q y Q' continuas en el intervalo $[a, b]$. Este tiene la forma

$$Q(x) = \begin{cases} Q_0(x) & x \in [t_0, t_1] \\ Q_1(x) & x \in [t_1, t_2] \\ \vdots & \vdots \\ Q_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

donde

$$Q_i(x) = \left(\frac{z_{i+1} - z_i}{2(t_{i+1} - t_i)} \right) (x - t_i)^2 + z_i(x - t_i) + y_i$$

en el intervalo $[t_i, t_{i+1}]$. Los coeficientes z_0, z_1, \dots, z_n se obtienen seleccionando z_0 y después utilizando la relación de recurrencia

$$z_{i+1} = -z_i + 2 \left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i} \right) \quad (0 \leq i \leq n - 1)$$

(3) Una **función spline cuadrático de Subbotin** Q es un polinomio de segundo grado por partes Q y Q' continuas en el intervalo $[a, b]$ y con la condición de interpolación en los extremos del intervalo $[a, b]$ y en los puntos medios de los subintervalos, es decir, $Q(\tau_i) = y_i$ para una $0 \leq i \leq n + 1$, donde

$$\tau_0 = t_0, \quad \tau_i = \frac{1}{2}(t_i + t_{i-1}) \quad (1 \leq i \leq n), \quad \tau_{n+1} = t_n$$

Este tiene la forma

$$Q_i(x) = y_{i+1} + \frac{1}{2}(z_{i+1} + z_i)(x - \tau_{i+1}) + \frac{1}{2h_i}(z_{i+1} - z_i)(x - \tau_{i+1})^2$$

donde $h_i = t_{i+1} - t_i$. Los coeficientes z_i se encuentran al resolver el sistema tridiagonal

$$\begin{cases} 3h_0 z_0 + h_0 z_1 = 8(y_1 - y_0) \\ h_{i-1} z_{i-1} + 3(h_{i-1} + h_i) z_i + h_i z_{i+1} = 8(y_{i+1} - y_i) \quad (1 \leq i \leq n-1) \\ h_{n-1} z_{n-1} + 3h_{n-1} z_n = 8(y_{n+1} - y_n) \end{cases}$$

como se analiza en la sección 7.3.

Problemas 9.1

- 1.** Determine si esta función es un spline de primer grado:

$$S(x) = \begin{cases} x & (-1 \leq x \leq 0.5) \\ 0.5 + 2(x - 0.5) & (0.5 \leq x \leq 2) \\ x + 1.5 & (2 \leq x \leq 4) \end{cases}$$

- 2.** El tipo más simple de la función spline es la función constante por partes, que podría definirse como

$$S(x) = \begin{cases} c_0 & (t_0 \leq x < t_1) \\ c_1 & (t_1 \leq x < t_2) \\ \vdots & \vdots \\ c_{n-1} & (t_{n-1} \leq x \leq t_n) \end{cases}$$

Demuestre que la integral indefinida de esta función es una función poligonal. ¿Cuál es la relación entre las funciones constantes por partes y la regla del rectángulo de integración numérica? (Véase el problema 5.2.29.)

- 3.** Demuestre que $f(x) - p(x) = \frac{1}{2}f'(\xi)(x-a)(x-b)$ para alguna ξ en el intervalo (a, b) , donde p es un polinomio lineal que interpola f en a y b . *Sugerencia:* utilice un resultado de la sección 4.2.
- 4.** (Continuación) Demuestre que $|f(x) - p(x)| \leq \frac{1}{8}M\ell^2$, donde $\ell = b - a$, si $|f''(x)| \leq M$ en el intervalo (a, b) .

- 5.** (Continuación) Demuestre que

$$f(x) - p(x) = \frac{(x-a)(x-b)}{b-a} \left[\frac{f(x) - f(b)}{x-b} - \frac{f(x) - f(a)}{x-a} \right]$$

- 6.** (Continuación) Si $|f'(x)| \leq C$ en (a, b) , demuestre que $|f(x) - p(x)| \leq C\ell/2$. *Sugerencia:* use el teorema del valor medio en el resultado del problema anterior.
- 7.** (Continuación) Sea S una función spline de primer grado que interpola f en t_0, t_1, \dots, t_n . Sea $t_0 < t_1 < \dots < t_n$ y sea $\delta = \max_{0 \leq i \leq n-1} (t_{i+1} - t_i)$. Entonces $|f(x) - S(x)| \leq C\delta/2$, donde C es un límite superior de $|f'(x)|$ en (t_0, t_n) .
- 8.** Sea f continua en $[a, b]$. Para una $\varepsilon > 0$ dada, sea que δ tenga la propiedad de que $|f(x) - f(y)| < \varepsilon$ siempre que $|x - y| < \delta$ (principio de continuidad uniforme). Sea $n > 1 + (b - a)/\delta$. Demuestre que hay un spline de primer grado S que tiene n nudos tal que $|f(x) - S(x)| < \varepsilon$ en $[a, b]$. *Sugerencia:* use el problema 5.

- ^a9. Si la función $f(x) = \operatorname{sen}(100x)$ se aproxima en el intervalo $[0, \pi]$ mediante un spline de interpolación de primer grado, ¿cuántos nudos se necesitan para asegurar que $|S(x) - f(x)| < 10^{-8}$? *Sugerencia:* use el problema 7.

- ^a10. Sea $t_0 < t_1 < \dots < t_n$. Construya funciones spline de primer grado G_0, G_1, \dots, G_n , requiriendo que los G_i se anulen en $t_0, t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$ pero que $G'_i(t_i) = 1$. Demuestre que la función spline de primer grado que interpola a f en t_0, t_1, \dots, t_n es $\sum_{i=0}^n f(t_i)G_i(x)$.

11. Demuestre que la regla del trapecio para integración numérica (sección 5.2) resulta de la aproximación de f con un spline de primer grado S y después usando

$$\int_a^b f(x) dx \approx \int_a^b S(x) dx$$

- ^a12. Demuestre que la derivada de un spline cuadrático es un spline de primer grado.

13. Si los nudos t_i son los enteros $0, 1, \dots, n$, encuentre una buena manera de determinar el índice i para el que $t_i \leq x < t_{i+1}$. (*Nota:* este problema es engañoso, porque a la palabra *buena* se le puede dar diferentes significados.)

14. Demuestre que la integral indefinida de un spline de primer grado es un spline de segundo grado.

15. Defina $f(x) = 0$ si $x < 0$ y $f(x) = x^2$ si $x \geq 0$. Demuestre que f y f' son continuas. Demuestre que cualquier spline cuadrático con nudos, t_0, t_1, \dots, t_n es de la forma

$$ax^2 + bx + c + \sum_{i=1}^{n-1} d_i f(x - t_i)$$

16. Se define una función g con la ecuación

$$g(x) = \begin{cases} 0 & (t_0 \leq x \leq 0) \\ x & (0 \leq x \leq t_n) \end{cases}$$

Demuestre que cada función spline de primer grado que tiene nudos t_0, t_1, \dots, t_n se puede escribir en la forma

$$ax + b + \sum_{i=1}^{n-1} c_i g(x - t_i)$$

- ^a17. Encuentre un spline cuadrático para interpolar estos datos:

x	-1	0	$\frac{1}{2}$	1	2	$\frac{5}{2}$
y	2	1	0	1	2	3

Suponga que $z_0 = 0$.

18. (Continuación) Demuestre que ningún spline cuadrático Q interpola la tabla del problema anterior y satisface que $Q'(t_0) = Q'(t_5)$.

- ^a19. ¿Qué ecuaciones se deben resolver si una función spline cuadrática Q que tiene nudos, t_0, t_1, \dots, t_n se le obliga a adoptar los valores dados en los puntos $\frac{1}{2}(t_i + t_{i+1})$ para $0 \leq i \leq n-1$?

20. ¿Son estas funciones splines cuadráticas? Explique por qué sí o por qué no.

$$\text{a. } Q(x) = \begin{cases} 0.1x^2 & (0 \leq x \leq 1) \\ 9.3x^2 - 18.4x + 9.2 & (1 \leq x \leq 1.3) \end{cases}$$

$$\text{b. } Q(x) = \begin{cases} -x^2 & (-100 \leq x \leq 0) \\ x & (0 \leq x \leq 100) \end{cases}$$

$$\text{c. } Q(x) = \begin{cases} x & (-50 \leq x \leq 1) \\ x^2 & (1 \leq x \leq 2) \\ 4 & (2 \leq x \leq 50) \end{cases}$$

21. ¿Es $S(x) = |x|$ un spline de primer grado? Por qué sí o por qué no.

22. Compruebe que la fórmula (5) tiene las tres propiedades $Q_i(t_i) = y_i$, $Q'_i(t_i) = z_i$ y $Q'_i(t_{i+1}) = z_{i+1}$.

23. (Continuación) Imponga la condición de continuidad en Q y deduzca el sistema de ecuaciones (6).

24. Demuestre por inducción que la fórmula recurrente (6), junto con la ecuación (5) produce una función de interpolación de spline cuadrático.

25. Verifique la exactitud de las ecuaciones del libro que se refieren al proceso de interpolación de spline de Subbotin.

26. Analice el método de interpolación de Subbotin de esta manera alternativa. Primero, sea $v_i = Q(t_i)$. Demuestre que

$$Q_i(x) = A_i(x - t_i)^2 + B_i(x - t_{i+1})^2 + C_i$$

donde

$$C_i = 2y_i - \frac{1}{2}v_i - \frac{1}{2}v_{i+1}, \quad B_i = \frac{v_i - C_i}{h_i^2}$$

$$A_i = \frac{v_{i+1} - C_i}{h_i^2} \quad h_i = t_{i+1} - t_i$$

Sugerencia: demuestre que $Q_i(t_i) = v_i$, $Q_i(t_{i+1}) = v_{i+1}$ y $Q_i(\tau_i) = y_i$.

27. (Continuación) Cuando las condiciones de continuidad en Q' se imponen, demuestre que el resultado es la siguiente ecuación, en la que $i = 1, 2, \dots, n-1$:

$$h_i v_{i-1} + 3(h_i + h_{i+1})v_i + h_{i-1}v_{i+1} = 4h_{i-1}y_i + 4h_i y_{i+1}$$

28. (Proyecto de investigación estudiantil) Es comúnmente aceptado que el artículo de Schoenberg [1946] es la primera referencia matemática en la que se utiliza la palabra *spline* en relación con suavizar aproximaciones polinomiales por partes. Sin embargo, la palabra *spline* como una fina tira de madera utilizada por un dibujante se remonta al menos a la década de 1890. Muchas de las ideas utilizadas en la teoría de splines tienen sus raíces en el trabajo realizado en diversas industrias como la construcción de aviones, automóviles y barcos en las que se utilizan ampliamente los splines. Investigue y escriba un artículo sobre la historia de los splines. (Consulte libros de historia de las matemáticas. Para un análisis de la historia de los splines en la industria automotriz, véase la *NA Digest*, volumen 98, núm. 26, julio 19, 1998).

Problemas de cómputo 9.1

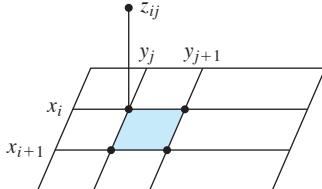
- Reescriba el procedimiento *Spline1* para que se consideren los subintervalos ascendentes en lugar de los descendentes. Pruebe el código en una tabla de 15 puntos de datos espaciados desigualmente.
- Reescriba el procedimiento *Spline1* para que se utilice una **búsqueda binaria** para encontrar el intervalo deseado. Pruebe el código revisado. ¿Cuáles son las ventajas y las desventajas de una búsqueda binaria en comparación con el procedimiento del libro? Una búsqueda binaria es similar al método de bisección en el que elegimos t_k con $k = (i + j)/2$ o $k = (i + j + 1)/2$ y determine si x está en $[t_i, t_k]$ o en $[t_k, t_j]$.
- Un **polinomio bilineal por partes** que interpola puntos (x, y) especificados en una malla rectangular está dado por

$$p(x, y) = \frac{(\ell_{ij}z_{i+1,j+1} + \ell_{i+1,j+1}z_{ij}) - (\ell_{i+1,j}z_{i,j+1} + \ell_{i,j+1}z_{i+1,j})}{(x_{i+1} - x_i)(y_{j+1} - y_j)}$$

donde $\ell_{ij} = (x_i - x)(y_j - y)$. Aquí $x_i \leq x \leq x_{i+1}$ y $y_j \leq y \leq y_{j+1}$. La malla dada (x_i, y_j) está especificada por estrictos arreglos crecientes (x_i) y (y_j) de longitud n y m , respectivamente. Los valores dados z_{ij} en los puntos de la malla (x_i, y_j) están contenidos en el arreglo (z_{ij}) de $n \times m$, que se muestra en la siguiente figura. Escriba

real function *Bi_Lineal*((x_i), n , (y_j), m , (z_{ij}), x , y)

para calcular el valor de $p(x, y)$. Pruebe esta rutina en un conjunto de 5×10 puntos de datos desigualmente espaciados. Evalúe *Bi_Lineal* en cuatro puntos de la malla y cinco puntos que no están en la malla.



- Escriba un procedimiento adaptado de interpolación con spline. La entrada debe ser una función f , un intervalo $[a, b]$ y una tolerancia ε . El resultado debería ser un conjunto de nudos $a = t_0 < t_1 < \dots < t_n = b$ y un conjunto de valores de la función $y_i = f(t_i)$ tal que la función de interpolación S con spline de primer grado satisface $|S(x) - f(x)| \leq \varepsilon$ cuando x es cualquier punto $x_j = t_i + j(t_{i+1} - t_i)/10$ para $0 \leq i \leq n-1$ y $0 \leq j \leq 9$.

- Escriba

procedure *Spline2_Coeff*(n , t , (y_i), (z_i))

que calcula el arreglo (z_i) en el proceso de interpolación de spline cuadrático (interpolación en los nudos). Después escriba

real function *Spline2_Eval*(n , (t_i), (y_i), (z_i), x)

que calcula los valores de $Q(x)$.

- Realice el proyecto de programación del problema de cómputo anterior para el spline cuadrático de Subbotin.

9.2 Splines cúbicos naturales

Introducción

Los splines de primer y segundo grado analizados en la sección anterior, aunque son útiles en ciertas aplicaciones, tienen una imperfección evidente: sus derivadas de bajo orden son discontinuas. En el caso del spline de primer grado (o línea poligonal), esta falta de uniformidad es evidente de inmediato, porque la pendiente del spline puede cambiar abruptamente de un valor a otro en cada nudo. Para el spline cuadrático, la discontinuidad está en la segunda derivada y por lo tanto no es tan evidente. Sin embargo, la *curvatura* del spline cuadrático cambia abruptamente en cada nudo y la curva puede no ser agradable a la vista.

La definición general de las funciones spline de grado arbitrario es la siguiente.

■ DEFINICIÓN 1

Spline de grado k

Una función S se llama un **spline de grado k** si:

1. El dominio de S es un intervalo $[a, b]$.
2. $S, S', S'', \dots, S^{(k-1)}$ son todas funciones continuas en $[a, b]$.
3. Hay puntos t_i (los nudos de S) tales que $a = t_0 < t_1 < \dots < t_n = b$ y tal que S es un polinomio a lo más de grado k en cada subintervalo $[t_i, t_{i+1}]$.

Observe que no se ha mencionado la interpolación en la definición de una función spline. De hecho, los splines son funciones tan versátiles que tienen muchos usos de la interpolación.

Los splines de grado superior se utilizan cada vez para suavizar más si se necesita en la función de aproximación. De la definición de una función spline de grado k , vemos que esta función será continua y tienen derivadas continuas $S, S', S'', \dots, S^{(k-1)}$. Si queremos que el spline de aproximación tenga una derivada m -ésima continua, se selecciona al menos un spline de grado $m + 1$. Para ver por qué, consideremos una situación en la que se han dado los nudos $t_0 < t_1 < \dots < t_n$. Supongamos que se define un polinomio por partes de grado m , con sus piezas unidas por los nudos de tal manera que resulta un spline S que tiene m derivadas continuas. En un nudo interior típico t , tenemos las siguientes circunstancias: a la izquierda de t , $S(x) = p(x)$, a la derecha de t , $S(x) = q(x)$, donde p y q son polinomios de grado m -ésimo. La continuidad de la m -ésima derivada $S^{(m)}$ implica la continuidad de las derivadas de orden menor $S^{(m-1)}, S^{(m-2)}, \dots, S'$, S . Por lo tanto, en el nudo t ,

$$\lim_{x \rightarrow t^-} S^{(k)}(x) = \lim_{x \rightarrow t^+} S^{(k)}(x) \quad (0 \leq k \leq m)$$

de lo que concluimos que

$$\lim_{x \rightarrow t^-} p^{(k)}(x) = \lim_{x \rightarrow t^+} q^{(k)}(x) \quad (0 \leq k \leq m) \tag{1}$$

Puesto que p y q son polinomios, sus derivadas de todos los órdenes son continuas y así la ecuación (1) es el igual a

$$p^{(k)}(t) = q^{(k)}(t) \quad (0 \leq k \leq m)$$

Esta condición obliga a que p y q sean *el mismo* polinomio, ya que por el teorema de Taylor,

$$p(x) = \sum_{k=0}^m \frac{1}{k!} p^{(k)}(t)(x-t)^k = \sum_{k=0}^m \frac{1}{k!} q^{(k)}(t)(x-t)^k = q(x)$$

Este argumento se puede aplicar en cada uno de los nudos interiores t_1, t_2, \dots, t_{n-1} y vemos que S es simplemente un polinomio en todo el intervalo de t_0 a t_n . Así, necesitamos un polinomio por partes de grado $m+1$ con al menos m derivadas continuas que tengan una función spline que no sea sólo un polinomio único en todo el intervalo. (Ya sabemos que los polinomios ordinarios no suelen servir muy bien en el ajuste de la curva. Véase la sección 4.2).

La elección de grado más utilizada para una función spline es 3. Los splines resultantes se denominan **splines cúbicos**. En este caso, unimos polinomios cúbicos de tal manera que la función spline resultante tiene dos derivadas continuas en todas partes. En cada nudo, se impondrán tres condiciones de continuidad. Puesto que S, S' y S'' son continuas, la gráfica de la función parecerá uniforme a la vista. Por supuesto, las discontinuidades se producirán en la tercera derivada, pero no se pueden detectar fácilmente a golpe de vista, lo cual es una de las razones para la elección de grado 3. La experiencia ha mostrado, además, que usando splines de grado superior a 3 rara vez se produce alguna ventaja. Por razones técnicas, los splines de grado impar se comportan mejor que los splines de grado par (cuando se interpola en los nudos). Por último, un teorema muy elegante, que se demostrará después, muestra que en un sentido exacto dado, la función spline cúbico de interpolación es la mejor función de interpolación disponible. Por tanto, nuestro interés en los splines cúbicos está bien justificado.

Spline cúbico natural

Pasamos ahora a la interpolación de una determinada tabla de valores mediante una función con un spline cúbico cuyos nudos coinciden con los valores de la variable independiente en la tabla. Como se dijo anteriormente, empezamos con la tabla:

x	t_0	t_1	\cdots	t_n
y	y_0	y_1	\cdots	y_n

Los t_i son los nudos y se supone que están arreglados en orden ascendente.

La función S que deseamos construir consta de n partes de polinomios cúbicos:

$$S(x) = \begin{cases} S_0(x) & (t_0 \leq x \leq t_1) \\ S_1(x) & (t_1 \leq x \leq t_2) \\ \vdots & \vdots \\ S_{n-1}(x) & (t_{n-1} \leq x \leq t_n) \end{cases}$$

En esta fórmula, S_i denota el polinomio cúbico que se usará en el subintervalo $[t_i, t_{i+1}]$. Las condiciones de interpolación son

$$S(t_i) = y_i \quad (0 \leq i \leq n)$$

Las condiciones de continuidad se imponen sólo en los nudos *interiores* t_1, t_2, \dots, t_{n-1} . (¿Por qué?) Estas condiciones se escriben como

$$\lim_{x \rightarrow t_i^-} S^{(k)}(t_i) = \lim_{x \rightarrow t_i^+} S^{(k)}(t_i) \quad (k = 0, 1, 2)$$

Resulta que se deben imponer dos condiciones más para utilizar todos los grados de libertad disponibles. La elección que hacemos de estas dos condiciones extra es

$$S''(t_0) = S''(t_n) = 0 \quad (2)$$

La función spline resultante se denomina **spline cúbico natural**. Otras formas de cerrar el sistema de ecuaciones para los coeficientes de los splines son los **splines cúbicos periódicos** y los **splines cúbicos sujetos**. Un spline sujeto es una curva spline cuya pendiente está fija en los dos extremos: $S'(t_0) = d_0$ y $S'(t_n) = d_n$. Un spline cúbico periódico tiene $S(t_0) = S(t_n)$, $S'(t_0) = S'(t_n)$ y $S''(t_0) = S''(t_n)$. Para todas las funciones diferenciables continuas, los splines sujetos y naturales cúbicos producen menos oscilaciones sobre la función f que interpolan.

Ahora comprobamos que el número de condiciones impuestas es igual al número de coeficientes disponibles. Hay $n + 1$ nudos y por lo tanto, n subintervalos. En cada uno de estos subintervalos tendremos un polinomio cúbico diferente. Dado que un polinomio cúbico tiene cuatro coeficientes, hay disponible un total de $4n$ coeficientes. En cuanto a las condiciones impuestas, hemos especificado que dentro de cada intervalo el polinomio de interpolación debe pasar por dos puntos, lo que da $2n$ condiciones. La continuidad no agrega condiciones adicionales. La primera y segunda derivadas deben ser continuas en los $n - 1$ puntos interiores, para $2(n - 1)$ más condiciones. La segunda derivada debe anularse en los dos extremos de un total de $2n + 2(n - 1) + 2 = 4n$ condiciones.

EJEMPLO 1 Deduzca las ecuaciones del spline cúbico natural de interpolación para la tabla siguiente:

x	−1	0	1
y	1	2	−1

Solución Nuestro método consiste en determinar los parámetros a, b, c, d, e, f, g y h de modo que $S(x)$ sea un spline cúbico natural, donde

$$S(x) = \begin{cases} S_0(s) = ax^3 + bx^2 + cx + d & x \in [-1, 0] \\ S_1(s) = ex^3 + fx^2 + gx + h & x \in [0, 1] \end{cases}$$

donde los dos polinomios cúbicos son $S_0(x)$ y $S_1(x)$. A partir de estas condiciones de interpolación, tenemos las condiciones de interpolación $S(-1) = S_0(-1) = -a + b - c + d = 1$, $S(0) = S_0(0) = d = 2$, $S(0) = S_1(0) = h = 2$ y $S(1) = S_1(1) = e + f + g + h = -1$. Tomando las primeras derivadas obtenemos

$$S'(x) = \begin{cases} S'_0(x) = 3ax^2 + 2bx + c & x \in [-1, 0] \\ S'_1(x) = 3ex^2 + 2fx + g & x \in [0, 1] \end{cases}$$

De la condición de continuidad de S' tenemos $S'_0(0) = S'_1(0)$, y hacemos $c = g$. Entonces, tomando las segundas derivadas, obtenemos

$$S''(x) = \begin{cases} S''_0(x) = 6ax + 2b & x \in [-1, 0] \\ S''_1(x) = 6ex + 2f & x \in [0, 1] \end{cases}$$

De la condición de continuidad de S'' , tenemos $S''_0(0) = S''_1(0)$, y hacemos $b = f$. Para S un spline cúbico natural, debemos tener $S''_0(-1) = 0$ y $S''_1(1) = 0$, y se obtiene $3a = b$ y $3e = -f$. De todas estas ecuaciones, obtenemos $a = -1$, $b = -3$, $c = -1$, $d = 2$, $e = 1$, $f = -3$, $g = -1$ y $h = 2$. ■

Algoritmo para el spline cúbico natural

Del ejemplo anterior, es evidente que necesitamos desarrollar un procedimiento sistemático para determinar la fórmula de un spline cúbico natural a partir de una tabla de valores de interpolación. Este es nuestro objetivo en el material de las páginas siguientes.

Puesto que S'' es continua, los números

$$z_i \equiv S''(t_i) \quad (0 \leq i \leq n)$$

están inequívocamente definidos. No conocemos aún los valores de z_1, z_2, \dots, z_{n-1} , pero, por supuesto, $z_0 = z_n = 0$ por la ecuación (2).

Si se conocieran los z_i podríamos construir S como se describe a continuación. En el intervalo $[t_i, t_{i+1}]$, S'' es un polinomio lineal que toma los valores z_i y z_{i+1} en los extremos. Por lo tanto,

$$S''_i(x) = \frac{z_{i+1}}{h_i}(x - t_i) + \frac{z_i}{h_i}(t_{i+1} - x) \quad (3)$$

con $h_i = t_{i+1} - t_i$ para $0 \leq i \leq n-1$. Con el fin de comprobar que la ecuación (3) es correcta, observe que $S''_i(t_i) = z_i$, $S''_i(t_{i+1}) = z_{i+1}$ y S'' es lineal en x . Si esto se integra dos veces, se obtiene S_i misma:

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + cx + d$$

donde c y d son constantes de integración. Ajustando las constantes de integración, se obtiene una forma para S_i con la que es más fácil trabajar, a saber,

$$S_i(x) = \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 + C_i(x - t_i) + D_i(t_{i+1} - x) \quad (4)$$

donde C_i y D_i son constantes. Si derivamos la ecuación (4) dos veces obtenemos la ecuación (3).

Las condiciones de interpolación $S_i(t_i) = y_i$ y $S_i(t_{i+1}) = z_{i+1}$ ahora se pueden imponer para determinar los valores apropiados de C_i y D_i . Usted debe hacerlo (problema 9.2.27) y comprobar que el resultado es

$$\begin{aligned} S_i(x) &= \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 \\ &\quad + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1} \right)(x - t_i) + \left(\frac{y_i}{h_i} - \frac{h_i}{6}z_i \right)(t_{i+1} - x) \end{aligned} \quad (5)$$

Cuando los valores, z_0, z_1, \dots, z_n se han determinado se obtiene la función spline $S(x)$ a partir de las ecuaciones de esta forma para $S_0(x), S_1(x), \dots, S_{n-1}(x)$.

Ahora mostramos cómo determinar los z_i . Una condición restante debe imponerse, a saber, la continuidad de S' . En los nudos interiores t_i para $1 \leq i \leq n-1$, debemos tener $S_{i-1}'(t_i) = S_i'(t_i)$ como puede verse en la figura 9.6.

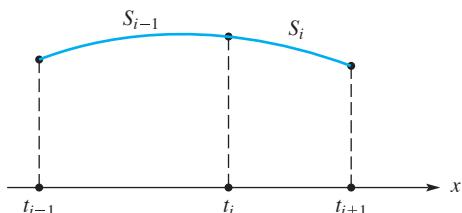


FIGURA 9.6
Spline cúbico: piezas adyacentes S_{i-1} y S_i

Tenemos, de la ecuación (5),

$$S'_i(x) = \frac{z_{i+1}}{2h_i}(x - t_i)^2 - \frac{z_i}{2h_i}(t_{i+1} - x)^2 + \frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{h_i}{6}z_i$$

Esto da

$$S'_i(t_i) = -\frac{h_i}{6}z_{i+1} - \frac{h_i}{3}z_i + b_i \quad (6)$$

donde

$$b_i = \frac{1}{h_i}(y_{i+1} - y_i) \quad (7)$$

Análogamente, tenemos

$$S'_{i-1}(t_i) = \frac{h_{i-1}}{6}z_{i-1} + \frac{h_{i-1}}{3}z_i + b_{i-1}$$

Cuando se igualan estas ecuaciones, la ecuación resultante se puede rearrugar como

$$h_{i-1}z_{i-1} + 2(h_{i-1} + h_i)z_i + h_iz_{i+1} = 6(b_i - b_{i-1})$$

para $1 \leq i \leq n-1$. Haciendo

$$\begin{aligned} u_i &= 2(h_{i-1} + h_i) \\ v_i &= 6(b_i - b_{i-1}) \end{aligned} \quad (8)$$

obtenemos un sistema tridiagonal de ecuaciones:

$$\begin{cases} z_0 = 0 \\ h_{i-1}z_{i-1} + u_iz_i + h_iz_{i+1} = v_i & (1 \leq i \leq n-1) \\ z_n = 0 \end{cases} \quad (9)$$

que se resuelve para las z_i . La simplicidad de la primera y de la última ecuación es un resultado de las condiciones del spline cúbico natural $S''(t_0) = S''(t_n) = 0$.

EJEMPLO 2 Repita el ejemplo 1 construyendo el spline cúbico natural que pasa por los puntos $(-1, 1)$, $(0, 2)$ y $(1, -1)$. También dibuje los resultados para visualizar la curva spline.

Solución De los valores dados, tenemos $t_0 = -1$, $t_1 = 0$, $t_2 = 1$, $y_0 = 1$, $y_1 = 1$ y $y_2 = 1$. En consecuencia, obtenemos $h_0 = t_1 - t_0 = 1$, $h_1 = t_2 - t_1 = 1$, $b_0 = (y_1 - y_0)/h_0 = 1$, $b_1 = (y_2 - y_1)/h_1 = -3$, $u_1 = 2(h_0 + h_1) = 4$ y $v_1 = 6(b_1 - b_0) = -24$. Entonces, el sistema tridiagonal de ecuaciones (9) es

$$\begin{cases} z_0 = 0 \\ z_0 + 4z_1 + z_2 = -24 \\ z_2 = 0 \end{cases}$$

Evidentemente, obtenemos la solución $z_0 = 0$, $z_1 = -6$ y $z_2 = 0$. De la ecuación (5), tenemos

$$S(x) = \begin{cases} S_0(x) = -(x+1)^3 + 3(x+1) - x & x \in [-1, 0] \\ S_1(x) = -(1-x)^3 - x + 3(1-x) & x \in [0, 1] \end{cases}$$

o

$$S(x) = \begin{cases} S_0(x) = -x^3 - 3x^2 - x + 2 & x \in [-1, 0] \\ S_1(x) = x^3 - 3x^2 - x + 2 & x \in [0, 1] \end{cases}$$

Esto concuerda con los resultados del ejemplo 1. El resultado de la curva spline natural que pasa por los puntos dados se muestra en la figura 9.7.

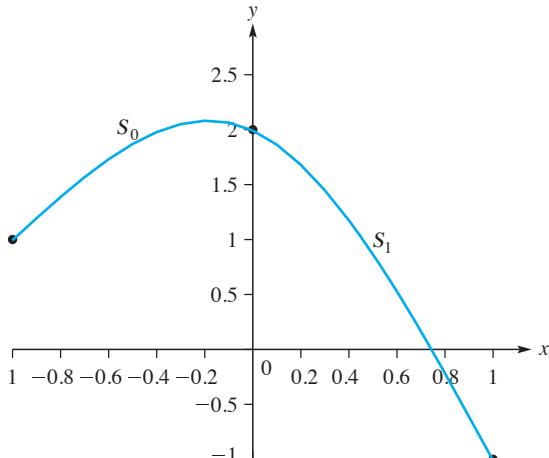


FIGURA 9.7
Spline cúbico
natural para los
ejemplos 1 y 2

Ahora considere el sistema (9) en forma matricial:

$$\begin{bmatrix} 1 & 0 & & \\ h_0 & u_1 & h_1 & \\ & h_1 & u_2 & h_2 \\ & \ddots & \ddots & \ddots \\ & & u_{n-1} & h_{n-1} \\ & & 0 & 1 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ \vdots \\ z_{n-1} \\ z_n \end{bmatrix} = \begin{bmatrix} 0 \\ v_1 \\ v_2 \\ \vdots \\ v_{n-1} \\ 0 \end{bmatrix}$$

Eliminando la primera y la última ecuación tenemos

$$\begin{bmatrix} u_1 & h_1 & & \\ h_1 & u_2 & h_2 & \\ & \ddots & \ddots & \ddots \\ & & u_{n-2} & h_{n-2} \\ & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & h_{n-2} & u_{n-1} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{n-2} \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{bmatrix} \quad (10)$$

que es un sistema tridiagonal simétrico de orden $n - 1$. Podríamos utilizar el procedimiento *Tri* desarrollado en la sección 7.3 para resolver este sistema. Sin embargo, podemos diseñar un algoritmo específico para este (basado en las ideas de la sección 7.3). En la eliminación gaussiana *sin pivoteo*, la fase de eliminación hacia adelante modificaría las u_i y las v_i de la manera siguiente:

$$\begin{cases} u_i \leftarrow u_i - \frac{h_{i-1}^2}{u_{i-1}} \\ v_i \leftarrow v_i - \frac{h_{i-1} v_{i-1}}{u_{i-1}} \end{cases} \quad (i = 2, 3, \dots, n-1)$$

Con la sustitución hacia atrás se obtiene

$$\begin{cases} z_{n-1} \leftarrow \frac{v_{n-1}}{u_{n-1}} \\ z_i \leftarrow \frac{v_i - h_i z_{i+1}}{u_i} \end{cases} \quad (i = n-2, n-3, \dots, 1)$$

Al reunir todo lo anterior se llega al algoritmo siguiente, diseñado especialmente para el sistema triagonal (10).

■ ALGORITMO 1 *Solución directa del sistema triagonal del spline cúbico natural*

Dados los puntos de interpolación (t_i, y_i) para $i = 0, 1, \dots, n$:

1. Calcule para $i = 0, 1, \dots, n - 1$:

$$\begin{cases} h_i = t_{i+1} - t_i \\ b_i = \frac{1}{h_i}(y_{i+1} - y_i) \end{cases}$$

2. Haga

$$\begin{cases} u_1 = 2(h_0 + h_1) \\ v_1 = 6(b_1 - b_0) \end{cases}$$

y calcule por inducción para $i = 2, 3, \dots, n - 1$:

$$\begin{cases} u_i = 2(h_i + h_{i-1}) - \frac{h_{i-1}^2}{u_{i-1}} \\ v_i = 6(b_i - b_{i-1}) - \frac{h_{i-1}v_{i-1}}{u_{i-1}} \end{cases}$$

3. Haga

$$\begin{cases} z_n = 0 \\ z_0 = 0 \end{cases}$$

y calcule por inducción para $i = n - 1, n - 2, \dots, 1$:

$$z_i = \frac{v_i - h_i z_{i+1}}{u_i}$$

Este algoritmo posiblemente podría fallar debido a la división entre cero en los pasos 2 y 3. Por lo tanto, vamos a demostrar que $u_i \neq 0$ para todo i . Es evidente que $u_1 > h_1 > 0$. Si $u_{i-1} > h_{i-1}$, entonces $u_i > h_i$, porque

$$u_i = 2(h_i + h_{i-1}) - \frac{h_{i-1}^2}{u_{i-1}} > 2(h_i + h_{i-1}) - h_{i-1} > h_i$$

Entonces, por inducción, $u_i > 0$ para $i = 1, 2, \dots, n - 1$.

La ecuación (5) no es la mejor forma de cálculo para evaluar los polinomios cúbicos $S_i(x)$. Preferimos tenerlo en la forma

$$S_i(x) = A_i + B_i(x - t_i) + C_i(x - t_i)^2 + D_i(x - t_i)^3 \quad (11)$$

ya que se puede utilizar la multiplicación anidada.

Observe que la ecuación (11) es el desarrollo de Taylor de S_i con respecto al punto t_i . Por tanto,

$$A_i = S_i(t_i), \quad B_i = S'_i(t_i), \quad C_i = \frac{1}{2}S''_i(t_i), \quad D_i = \frac{1}{6}S'''_i(t_i)$$

Por lo tanto, $A_i = y_i$ y $C_i = z_i/2$. El coeficiente de x^3 en la ecuación (11) es D_i , mientras que el coeficiente de x^3 en la ecuación (5) es $(z_{i+1} - z_i)/6h_i$. Por tanto,

$$D_i = \frac{1}{6h_i}(z_{i+1} - z_i)$$

Por último, la ecuación (6) da el valor de $S_i(t_i)$, que es

$$B_i = -\frac{h_i}{6}z_{i+1} - \frac{h_i}{3}z_i + \frac{1}{h_i}(y_{i+1} - y_i)$$

Así, la forma anidada de $S_i(x)$ es

$$S_i(x) = y_i + (x - t_i) \left(B_i + (x - t_i) \left(\frac{z_i}{2} + \frac{1}{6h_i}(x - t_i)(z_{i+1} - z_i) \right) \right) \quad (12)$$

Seudocódigo para splines cúbicos naturales

Ahora escribimos rutinas para determinar un spline cúbico natural con base en una tabla de valores y para evaluar esta función en un valor dado. Primero, usamos el algoritmo 1 para resolver directamente el sistema tridiagonal (10). Este procedimiento, llamado *Spline3_Coef*, toma $n + 1$ valores de tabla (t_i, y_i) en arreglos (t_i) y (y_i) y calcula los z_i y los guarda en el arreglo (z_i) . Se necesitan arreglos intermedios (de trabajo) (h_i) , (b_i) , (u_i) y (v_i) .

```

procedure Spline3_Coef( $n, (t_i), (y_i), (z_i)$ )
integer  $i, n;$  real array  $(t_i)_{0:n}, (y_i)_{0:n}, (z_i)_{0:n}$ 
allocate real array  $(h_i)_{0:n-1}, (b_i)_{0:n-1}, (u_i)_{1:n-1}, (v_i)_{1:n-1}$ 
for  $i = 0$  to  $n - 1$  do
     $h_i \leftarrow t_{i+1} - t_i$ 
     $b_i \leftarrow (y_{i+1} - y_i)/ h_i$ 
end for
     $u_1 \leftarrow 2(h_0 + h_1)$ 
     $v_1 \leftarrow 6(b_1 - b_0)$ 
for  $i = 2$  to  $n - 1$  do
     $u_i \leftarrow 2(h_i + h_{i-1}) - h_{i-1}^2/u_{i-1}$ 
     $v_i \leftarrow 6(b_i - b_{i-1}) - h_{i-1}v_{i-1}/u_{i-1}$ 
end for
     $z_n \leftarrow 0$ 
for  $i = n - 1$  to  $1$  step  $-1$  do
     $z_i \leftarrow (v_i - h_iz_{i+1})/ u_i$ 
end for
     $z_0 \leftarrow 0$ 
deallocate array  $(h_i), (b_i), (u_i), (v_i)$ 
end procedure Spline3_Coef

```

Ahora, se escribe un procedimiento llamado *Spline3_Eval* para evaluar la ecuación (12), la función spline cúbico natural $S(x)$, para un valor de x dado. El procedimiento *Spline3_Eval* primero determina el intervalo $[t_i, t_{i+1}]$ que contiene a x y después evalúa $S(x)$, utilizando una forma anidada de este polinomio cúbico:

```

real function Spline3_Eval( $n, (t_i), (y_i), (z_i), x$ )
integer  $i;$  real  $h, tmp$ 
real array  $(t_i)_{0:n}, (y_i)_{0:n}, (z_i)_{0:n}$ 
for  $i = n - 1$  to  $0$  step  $-1$  do
    if  $x - t_i \geq 0$  then exit loop

```

```

end for
 $h \leftarrow t_{i+1} - t_i$ 
 $tmp \leftarrow (z_i / 2) + (x - t_i)(z_{i+1} - z_i)/(6h)$ 
 $tmp \leftarrow -(h/6)(z_{i+1} + 2z_i) + (y_{i+1} - y_i)/h + (x - t_i)(tmp)$ 
 $Spline3_Eval \leftarrow y_i + (x - t_i)(tmp)$ 
end function Spline3_Eval

```

La función *Spline3_Eval* se puede utilizar varias veces con diferentes valores de x después de una llamada al procedimiento *Spline3_Coef*. Por ejemplo, este sería el procedimiento al trazar la curva del spline cúbico natural. Puesto que el procedimiento *Spline3_Coef* guarda la solución del sistema triagonal correspondiente a una función spline particular en el arreglo (z_i) , los n argumentos, (t_i) , (y_i) y (z_i) no se deben alterar entre los usos repetidos de *Spline3_Eval*.

Uso de seudocódigo para interpolar y ajustar curvas

Para ilustrar el uso de las rutinas de splines cúbicos naturales *Spline3_Coef* y *Spline3_Eval*, trabajamos de nuevo con un ejemplo de la sección 4.1.

EJEMPLO 3 Escriba un seudocódigo para un programa que determine el spline cúbico natural para $\sin x$ en diez nudos equidistantes en el intervalo $[0, 1.6875]$. En el mismo intervalo, subdivida cada subintervalo en cuatro partes igualmente espaciadas y encuentre el punto donde el valor de $|\sin x - S(x)|$ es grande.

Solución Aquí se presenta seudocódigo adecuado para un programa principal, que llama a los procedimientos *Spline3_Coef* y *Spline3_Eval*:

```

procedure Test_Spline3
integer i; real e, h, x
real array (ti)0:n, (yi)0:n, (zi)0:n
integer n ← 9
real a ← 0, b ← 1.6875
h ← (b - a)/n
for i = 0 to n do
    ti ← a + i h
    yi ← sin(ti)
end for
call Spline3_Coef(n, (ti), (yi), (zi))
temp ← 0
for j = 0 to 4n do
    x ← a + jh/4
    e ← |sin(x) - Spline3_Eval(n, (ti), (yi), (zi), x)|
    if e > temp then temp ← e
    output j, x, e
end for
end Test_Spline3

```

De la computadora, la salida es $j = 19$, $x = 0.890625$ y $d = 0.930 \times 10^{-5}$.

Podemos usar software matemático como el de Matlab para trazar la curva spline cúbico para estos datos, pero la rutina `spline` de Matlab **sin condición de nudo en un extremo**, que es diferente de la condición final natural. Tal condición establece que S''' sea una sola constante en los primeros dos subintervalos y otra única constante en los últimos dos subintervalos. Primero, se generan los datos originales. A continuación, se hace una subdivisión más fina del intervalo $[a, b]$ en el eje x , y los valores correspondientes y se obtienen del procedimiento `spline`. Por último, se trazan los puntos de los datos originales y la curva spline.

Ahora mostramos el uso de funciones spline en el ajuste de una curva a un conjunto de datos. Considere la siguiente tabla:

x	0.0	0.6	1.5	1.7	1.9	2.1	2.3	2.6	2.8	3.0
y	-0.8	-0.34	0.59	0.59	0.23	0.1	0.28	1.03	1.5	1.44
3.6	4.7	5.2	5.7	5.8	6.0	6.4	6.9	7.6	8.0	
0.74	-0.82	-1.27	-0.92	-0.92	-1.04	-0.79	-0.06	1.0	0.0	

Estos 20 puntos se seleccionaron de una curva dibujada a mano alzada en papel cuadriculado. Intencionalmente hemos seleccionado más puntos donde la curva se inclina más fuertemente y se trató de reproducir la curva utilizando un trazador automático. Una curva es siempre agradable a la vista usando las rutinas de spline cúbico `Spline3_Coef` y `Spline3_Eval`. La figura 9.8 muestra el resultado natural de la curva de spline cúbico.

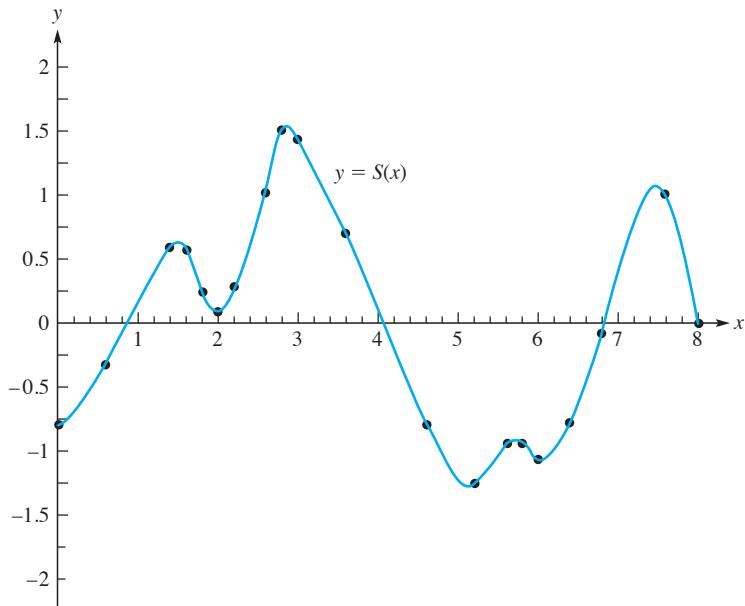


FIGURA 9.8
Curva del
spline cúbico
natural

Alternativamente, podemos usar software matemático como Matlab, Maple o Mathematica para trazar la gráfica de la función spline cúbico para esta tabla.

Curvas espaciales

En dos dimensiones, se pueden utilizar dos funciones spline cúbico para formar una **representación paramétrica** de una curva complicada que se volteá y se tuerce. Seleccione puntos de la curva y

etiquételos $t = 0, 1, \dots, n$. Para cada valor de t , lea las coordenadas x y y del punto, lo que resulta en la tabla:

t	0	1	\cdots	n
x	x_0	x_1	\cdots	x_n
y	y_0	y_1	\cdots	y_n

Entonces ajuste $x = S(t)$ y $y = \bar{S}(t)$, donde S y \bar{S} son splines cúbicos naturales de interpolación. Las dos funciones S y \bar{S} dan una representación paramétrica de la curva (véase el problema de cómputo 9.2.6).

EJEMPLO 4 Seleccione 13 puntos de la bien conocida **curva serpentina** dada por

$$y = \frac{x}{1/4 + x^2}$$

Puesto que los nudos no estarán igualmente espaciados, escribiendo la curva en forma paramétrica:

$$\begin{cases} x = \frac{1}{2} \tan \theta \\ y = \sin 2\theta \end{cases}$$

y tome $\theta = i(\pi/12)$, donde $i = -6, -5, \dots, 5, 6$. Trace la curva de spline cúbico natural y la del polinomio de interpolación para compararlas.

Solución Este es un ejemplo de ajuste de curvas utilizando las rutinas de interpolación polinómica *Coef* y *Eval* del capítulo 4 y las rutinas de spline cúbico *Spline3_Coef* y *Spline3_Eval*. La figura 9.9 muestra la curva resultante del spline cúbico y la curva polinómica de alto grado (línea discontinua) de un trazador automático. El polinomio se vuelve extremadamente irregular después del cuarto nudo a partir del origen y oscila fuertemente, mientras que el spline es un ajuste casi perfecto.

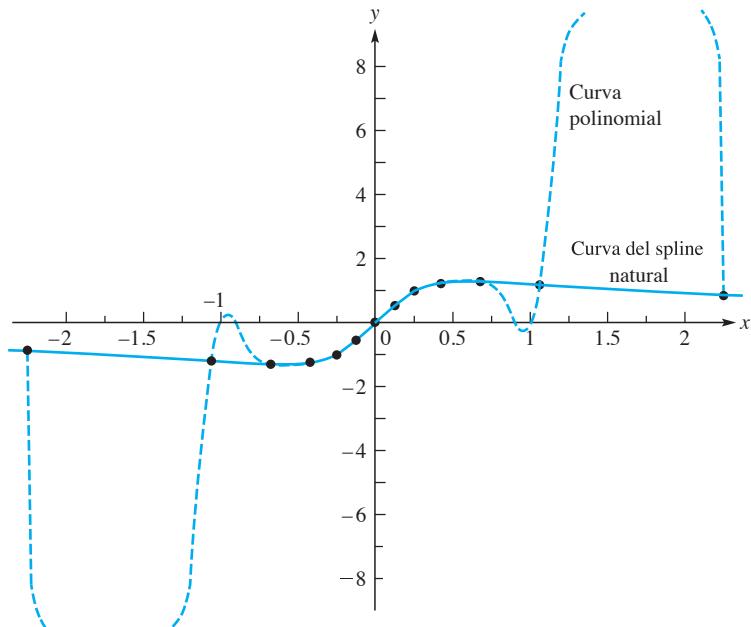


FIGURA 9.9
Curva
serpentina

EJEMPLO 5 Use funciones de spline cúbico para obtener la curva para los datos siguientes:

t	0	1	2	3	4	5	6	7
y	1.0	1.5	1.6	1.5	0.9	2.2	2.8	3.1

Se sabe que la curva es continua pero su pendiente no lo es.

Solución Un spline cúbico no es conveniente. En su lugar, podemos utilizar dos splines cúbicos de interpolación, el primero con 0, 1, 2, 3, 4 nudos y el segundo con 4, 5, 6, 7 nudos. Realizando dos procedimientos independientes de spline de interpolación obtenemos dos curvas de spline cúbico que se encuentran en el punto $(4, 0.9)$. En este punto, las dos curvas tienen pendientes distintas. La curva resultante se muestra en la figura 9.10.

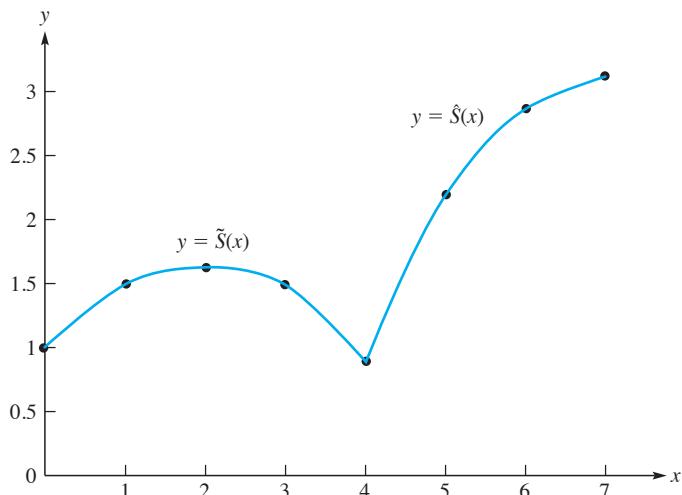


FIGURA 9.10
Dos splines cúbicos



Propiedad de suavidad

¿Por qué las funciones spline satisfacen las necesidades del ajuste de datos mejor que los polinomios comunes? Para responder esto, se debe entender que la interpolación mediante polinomios de grado alto con frecuencia no es satisfactoria, ya que los polinomios pueden presentar fuertes *oscilaciones*. Los polinomios son suaves en el sentido técnico, pues tienen derivadas continuas de todos los órdenes, mientras que en este sentido, las funciones spline *no* son suaves.

Las fuertes oscilaciones en una función se pueden atribuir a que sus derivadas son muy grandes. Considere la función cuya gráfica se muestra en la figura 9.11. La pendiente de la cuerda que

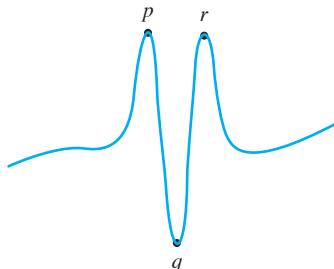


FIGURA 9.11
Función oscilando fuertemente

une los puntos p y q tiene magnitud muy grande. Por el teorema del valor medio, la pendiente de dicha cuerda es igual al valor de la derivada en algún punto entre p y q . Por lo tanto, la derivada debe tener valores grandes. De hecho, en algún lugar de la curva entre p y q hay un punto donde $f'(x)$ es grande y negativa. Del mismo modo, entre q y r hay un punto en el que $f'(x)$ es grande y positiva. Por tanto, hay un punto en la curva entre p y r donde $f''(x)$ es grande. Este razonamiento se puede continuar a derivadas más altas si hay más oscilaciones. Este es el comportamiento que no presentan las funciones spline. De hecho, el siguiente resultado muestra que, desde cierto punto de vista, los splines cúbicos naturales son las *mejores* funciones por utilizar para el ajuste de curvas.

■ TEOREMA 1

Teorema de suavidad del spline cúbico

Si S es la función spline cúbico natural que interpola una función dos veces derivable f en los nudos $a = t_0 < t_1 < \dots < t_n = b$, entonces

$$\int_a^b [S''(x)]^2 dx \leq \int_a^b [f''(x)]^2 dx$$

Demostración Para comprobar el enunciado acerca de $[S''(x)]^2$, hacemos

$$g(x) = f(x) - S(x)$$

de modo que $g(t_i) = 0$ para $0 \leq i \leq n$ y

$$f'' = S'' + g''$$

Ahora

$$\int_a^b (f'')^2 dx = \int_a^b (S'')^2 dx + \int_a^b (g'')^2 dx + 2 \int_a^b S'' g'' dx$$

Si la última integral fuese 0, podríamos terminar porque entonces

$$\int_a^b (f'')^2 dx = \int_a^b (S'')^2 dx + \int_a^b (g'')^2 dx \geq \int_a^b (S'')^2 dx$$

Aplicando la técnica de integración por partes a la integral mostramos que esta es 0.* Tenemos

$$\int_a^b S'' g'' dx = S'' g' \Big|_a^b - \int_a^b S''' g' dx = - \int_a^b S''' g' dx$$

* La fórmula de **integración por partes** es

$$\int u dv = uv - \int v du$$

Aquí, se ha usado el hecho de que S es un spline cúbico *natural*, es decir, $S'''(a) = 0$ y $S'''(b) = 0$. Continuando, tenemos

$$\int_a^b S'''g' dx = \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} S'''g' dx$$

Puesto que S es un polinomio cúbico en cada intervalo $[t_i, t_{i+1}]$, su tercera derivada es una constante, digamos c_i . Así

$$\int_a^b S'''g' dx = \sum_{i=0}^{n-1} c_i \int_{t_i}^{t_{i+1}} g' dx = \sum_{i=0}^{n-1} c_i [g(t_{i+1}) - g(t_i)] = 0$$

ya que g es igual a cero en cada nudo. ■

La interpretación de la desigualdad de las integrales en el teorema es que el valor promedio de $[S'''(x)]^2$ en el intervalo $[a, b]$ nunca es mayor que el valor promedio de esta expresión con cualesquiera dos funciones continuas f que concuerden con S en los nudos. La cantidad $[f''(x)]^2$ está muy relacionada con la curvatura de la función f .

Resumen

(1) Nos dan $n + 1$ pares de puntos (t_i, y_i) con distintos **nudos** $a = t_0 < t_1 < \dots < t_{n-1} < t_n = b$ sobre el intervalo $[a, b]$. Una **función spline de grado k** es una función polinomial por partes tal que $S, S', S'', \dots, S^{(k-1)}$ son todas funciones continuas en $[a, b]$ y S es un polinomio de grado a lo más k en cada subintervalo $[t_i, t_{i+1}]$.

(2) Una **función spline cúbico natural** S es un polinomio cúbico por partes definido en el intervalo $[a, b]$ tal que S, S', S'' son continuas y $S'''(t_0) = S'''(t_n) = 0$. Esto se puede escribir en la forma

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

donde en el intervalo $[t_i, t_{i+1}]$

$$\begin{aligned} S_i(x) &= \frac{z_{i+1}}{6h_i}(x - t_i)^3 + \frac{z_i}{6h_i}(t_{i+1} - x)^3 \\ &\quad + \left(\frac{y_{i+1}}{h_i} - \frac{h_i}{6}z_{i+1} \right)(x - t_i) + \left(\frac{y_i}{h_i} - \frac{h_i}{6}z_i \right)(t_{i+1} - x) \end{aligned}$$

y donde $h_i = t_{i+1} - t_i$. Obviamente, $S(x)$ es continua, ya que $S_{i-1}(t_i) = S_i(t_i) = y_i$ para $1 \leq i \leq n$. Se puede demostrar que $S'_{i-1}(t_i) = S'_i(t_i)$ y $S''_{i-1}(t_i) = S''_i(t_i) = z_i$ para $1 \leq i \leq n$. Para evaluar eficientemente, se utiliza la forma anidada de $S_i(x)$, que es

$$S_i(x) = y_i + (x - t_i) \left(B_i + (x - t_i) \left(\frac{z_i}{2} + \frac{1}{6h_i}(x - t_i)(z_{i+1} - z_i) \right) \right)$$

donde $B_i = -(h_i/6)z_{i+1} - (h_i/3)z_i + (y_{i+1} - y_i)/h_i$. Los coeficientes z_0, z_1, \dots, z_n se encuentran haciendo $b_i = (y_{i+1} - y_i)/h_i$, $u_i = 2(h_{i-1} + h_i)$, $v_i = 6(b_i - b_{i-1})$, y después resolviendo

el sistema tridiagonal de ecuaciones

$$\begin{cases} z_0 = 0 \\ h_{i-1}z_{i-1} + u_i z_i + h_i z_{i+1} = v_i & (1 \leq i \leq n-1) \\ z_n = 0 \end{cases}$$

Esto se puede hacer eficientemente usando sustitución hacia adelante:

$$\begin{cases} u_i \leftarrow u_i - \frac{h_{i-1}^2}{u_{i-1}} \\ v_i \leftarrow v_i - \frac{h_{i-1}v_{i-1}}{u_{i-1}} & (i = 2, 3, \dots, n-1) \end{cases}$$

y sustitución hacia atrás:

$$\begin{cases} z_{n-1} \leftarrow \frac{v_{n-1}}{u_{n-1}} \\ z_i \leftarrow \frac{v_i - h_i z_{i+1}}{u_i} & (i = n-2, n-3, \dots, 1) \end{cases}$$

Problemas 9.2

1. ¿Existen a, b, c y d tal que la función

$$S(x) = \begin{cases} ax^3 + x^2 + cx & (-1 \leq x \leq 0) \\ bx^3 + x^2 + dx & (0 \leq x \leq 1) \end{cases}$$

sea una función spline cúbico natural que concuerde con el valor absoluto de la función $|x|$ en los nudos $-1, 0, 1$?

2. ¿Existen a, b, c y d tal que la función

$$S(x) = \begin{cases} -x & (-10 \leq x \leq -1) \\ ax^3 + bx^2 + cx + d & (-1 \leq x \leq 1) \\ x & (1 \leq x \leq 10) \end{cases}$$

sea una función spline cúbico natural?

3. Determine el spline cúbico natural que interpola la función $f(x) = x^6$ sobre el intervalo $[0, 2]$ usando los nudos $0, 1$ y 2 .

4. Determine los parámetros a, b, c, d y e tales que S sea un spline cúbico natural:

$$S(x) = \begin{cases} a + b(x-1) + c(x-1)^2 + d(x-1)^3 & (x \in [0, 1]) \\ (x-1)^3 + ex^2 - 1 & (x \in [1, 2]) \end{cases}$$

5. Determine los valores de a, b, c y d tales que f sea un spline cúbico y tales que $\int_0^2 [f''(x)]^2 dx$ sea un mínimo:

$$f(x) = \begin{cases} 3 + x - 9x^3 & (0 \leq x \leq 1) \\ a + b(x-1) + c(x-1)^2 + d(x-1)^3 & (1 \leq x \leq 2) \end{cases}$$

“6. Determine si f es un spline cúbico con nudos $-1, 0, 1$ y 2 :

$$f(x) = \begin{cases} 1 + 2(x+1) + (x+1)^3 & (-1 \leq x \leq 0) \\ 3 + 5x + 3x^2 & (0 \leq x \leq 1) \\ 11 + (x-1) + 3(x-1)^2 + (x-1)^3 & (1 \leq x \leq 2) \end{cases}$$

7. Liste todas las formas en las que las funciones siguientes fallan en ser splines cúbicos naturales:

a. $S(x) = \begin{cases} x+1 & (-2 \leq x \leq -1) \\ x^3 - 2x + 1 & (-1 \leq x \leq 1) \\ x-1 & (1 \leq x \leq 2) \end{cases}$

b. $f(x) = \begin{cases} x^3 + x - 1 & (-1 \leq x \leq 0) \\ x^3 - x - 1 & (0 \leq x \leq 1) \end{cases}$

8. Suponga que $S(x)$ es una función spline de interpolación de m -ésimo grado sobre el intervalo $[a, b]$ con $n+1$ nudos $a = t_0 < t_1 < \dots < t_n = b$.

a. ¿Cuántas condiciones se necesitan para definir $S(x)$ únicamente sobre $[a, b]$?

b. ¿Cuántas condiciones se definen con las condiciones de interpolación en los nudos?

c. ¿Cuántas condiciones se definen por la continuidad de las derivadas?

d. ¿Cuántas condiciones adicionales se necesitan para que el total iguale al número del inciso a?

9. Demuestre que

$$S(x) = \begin{cases} 28 + 25x + 9x^2 + x^3 & (-3 \leq x \leq -1) \\ 26 + 19x + 3x^2 - x^3 & (-1 \leq x \leq 0) \\ 26 + 19x + 3x^2 - 2x^3 & (0 \leq x \leq 3) \\ -163 + 208x - 60x^2 + 5x^3 & (3 \leq x \leq 4) \end{cases}$$

es una función spline cúbico natural.

“10. Dé un ejemplo de un spline cúbico con nudos $0, 1, 2$ y 3 que es cuadrático en $[0, 1]$, cúbico en $[1, 2]$ y cuadrático en $[2, 3]$.

11. De un ejemplo de una función spline cúbico S con nudos $0, 1, 2$ y 3 tal que S es lineal en $[0, 1]$ pero de grado 3 en los otros dos intervalos.

“12. Determine a, b y c tal que S es una función spline cúbico:

$$S(x) = \begin{cases} x^3 & (0 \leq x \leq 1) \\ \frac{1}{2}(x-1)^3 + a(x-1)^2 + b(x-1) + c & (1 \leq x \leq 3) \end{cases}$$

“13. ¿Hay una elección de coeficientes para la cual la siguiente función sea un spline cúbico natural? ¿Por qué sí o por qué no?

$$f(x) = \begin{cases} x+1 & (-2 \leq x \leq -1) \\ ax^3 + bx^2 + cx + d & (-1 \leq x \leq 1) \\ x-1 & (1 \leq x \leq 2) \end{cases}$$

14. Determine los coeficientes en la función

$$S(x) = \begin{cases} x^3 - 1 & (-9 \leq x \leq 0) \\ ax^3 + bx^2 + cx + d & (0 \leq x \leq 5) \end{cases}$$

tal que esta sea un spline cúbico que toma el valor 2 cuando $x = 1$.

- 15.** Determine los coeficientes tales que la función

$$S(x) = \begin{cases} x^2 + x^3 & (0 \leq x \leq 1) \\ a + bx + cx^2 + dx^3 & (1 \leq x \leq 2) \end{cases}$$

sea un spline cúbico y tenga la propiedad $S'''(x) = 12$.

- 16.** Suponga que $a = x_0 < x_1 < \dots < x_m = b$. Describa la función f que interpola una tabla de valores (x_i, y_i) , donde $0 \leq i \leq m$, y que minimiza la expresión $\int_a^b |f'(x)| dx$.

- 17.** ¿Cuántas condiciones adicionales son necesarias para especificar de manera única un spline de grado 4 en n nudos?

- 18.** Sean los nudos $t_0 < t_1 < \dots < t_n$ y sean los números y_i y z_i dados. Determine las fórmulas para una función cúbica por partes f que tiene los nudos dados tales que $f(t_i) = y_i$ ($0 \leq i \leq n$), $\lim_{x \rightarrow t_i^+} f''(x) = z_i$ ($0 \leq i \leq n-1$) y $\lim_{x \rightarrow t_i^-} f''(x) = z_i$ ($1 \leq i \leq n$). ¿Por qué f no es generalmente un spline cúbico?

- 19.** Defina una función f por

$$f(x) = \begin{cases} x^3 + x - 1 & (-1 \leq x \leq 0) \\ x^3 - x - 1 & (0 \leq x \leq 1) \end{cases}$$

Demuestre que $\lim_{x \rightarrow 0^+} f(x) = \lim_{x \rightarrow 0^-} f(x)$ y que $\lim_{x \rightarrow 0^+} f''(x) = \lim_{x \rightarrow 0^-} f''(x)$. ¿Son f y f'' continuas? ¿Se deduce que f es un spline cúbico? Explique.

- 20** Demuestre que hay un único spline cúbico S con nudos $t_0 < t_1 < \dots < t_n$, que interpola datos $S(t_i) = y_i$ ($0 \leq i \leq n$) y que satisface las dos condiciones de los extremos $S'(t_0) = S'(t_n) = 0$.

- 21.** Describa explícitamente el spline cúbico natural que interpola una tabla con sólo dos entradas:

x	t_0	t_1
y	y_0	y_1

Dé una fórmula para este. En este caso, t_0 y t_1 son los nudos.

- 22.** Suponga que $f(0) = 0$, $f(1) = 1.1752$, $f'(0) = 1$ y $f'(1) = 1.5431$. Determine el polinomio cúbico de interpolación $p_3(x)$ para estos datos. ¿Es este un spline cúbico natural?

- 23.** Un **spline cúbico periódico** que tiene los nudos t_0, t_1, \dots, t_n está definido como una función spline cúbico $S(x)$ tal que $S(t_0) = S(t_n)$, $S'(t_0) = S'(t_n)$ y $S''(t_0) = S''(t_n)$. Estos se utilizarán para ajustar los datos que se sabe que son periódicos. Realice los análisis necesarios para obtener un spline cúbico periódico de interpolación para la tabla

x	t_0	t_1	\cdots	t_n
y	y_0	y_1	\cdots	y_n

suponiendo que $y_n = y_0$.

- 24.** Las derivadas e integrales de los polinomios son polinomios. Establezca y demuestre un resultado similar de las funciones spline.

- 25.** Dada una función derivable f y los nudos $t_0 < t_1 < \dots < t_n$, muestre cómo obtener un spline cúbico S que interpole f en los nudos y que satisfaga las condiciones de los extremos $S'(t_0) = f'(t_0)$ y $S'(t_n) = f'(t_n)$. *Nota:* este procedimiento resulta en un mejor ajuste de f cuando es aplicable. Si f' no se conoce, se pueden usar las aproximaciones por diferencias finitas a $f'(t_0)$ y $f'(t_n)$.

- 26.** Sea S un spline cúbico que tiene los nudos $t_0 < t_1 < \dots < t_n$. Suponga que en los dos intervalos $[t_0, t_1]$ y $[t_2, t_3]$, S se reduce a polinomios lineales. ¿Qué se puede decir de S en $[t_1, t_2]$?

- 27.** En la construcción de un spline cúbico de interpolación, realice la evaluación de las constantes C_i y D_i , y así justifique la ecuación (5).

- 28.** Muestre que S_i también se puede escribir en la forma

$$S_i(x) = y_i + A_i(x - t_i) + \frac{1}{2}z_i(x - t_i)^2 + \frac{z_{i+1} - z_i}{6h_i}(x - t_i)^3$$

con

$$A_i = -\frac{h_i}{3}z_i - \frac{h_i}{6}z_{i+1} - \frac{y_i}{h_i} + \frac{y_{i+1}}{h_i}$$

- 29.** Realice los detalles de la deducción de la ecuación (9), iniciando con la ecuación (5).

- 30.** Verifique que el algoritmo para calcular el arreglo (z_i) es correcto demostrando que si (z_i) satisface la ecuación (9), entonces satisface la ecuación en el paso 3 del algoritmo.

- 31.** Establezca que $u_i > 2h_i + \frac{3}{2}h_{i-1}$ en el algoritmo para determinar el spline cúbico de interpolación.

- 32.** Con un cálculo manual, determine el spline cúbico natural de interpolación para esta tabla:

x	1	2	3	4	5
y	0	1	0	1	0

- 33.** Determine un spline cúbico en los nudos $-1, 0$ y 1 tal que se satisfacen las siguientes condiciones: $S'''(-1) = S''(1) = 0$, $S(-1) = S(1) = 0$ y $S(0) = 1$.

- 34.** Este problema y los dos siguientes conducen a un algoritmo más eficiente para un spline cúbico natural de interpolación en el caso de nudos igualmente espaciados. Sea $h_i = h$ en la ecuación (5) y sustituya los parámetros z_i por $q_i = h^2 z_i / 6$. Demuestre que la nueva forma de la ecuación (5) es entonces

$$\begin{aligned} S_i(x) = q_{i+1} \left(\frac{x - t_i}{h} \right)^3 + q_i \left(\frac{t_{i+1} - x}{h} \right)^3 + (y_{i+1} - q_{i+1}) \left(\frac{x - t_i}{h} \right) \\ + (y_i - q_i) \left(\frac{t_{i+1} - x}{h} \right) \end{aligned}$$

- 35.** (Continuación) Establezca las nuevas condiciones de continuidad:

$$q_0 = q_n = 0 \quad q_{i-1} + 4q_i + q_{i+1} = y_{i+1} - 2y_i + y_{i-1} \quad (1 \leq i \leq n-1)$$

- 36.** (Continuación) Demuestre que los parámetros q_i se pueden determinar con recursión hacia atrás como sigue:

$$q_n = 0 \quad q_{n-1} = \beta_{n-1} \quad q_i = \alpha_i q_{i+1} + \beta_i \quad (i = n-2, n-3, \dots, 0)$$

donde los coeficientes α_i y β_i se generan por recursión ascendente de las fórmulas

$$\begin{aligned} \alpha_0 &= 0 & \alpha_i &= -(a_{i-1} + 4)^{-1} & (1 \leq i \leq n) \\ \beta_0 &= 0 & \beta_i &= -\alpha_i(y_{i+1} - 2y_i + y_{i-1} - \beta_{i-1}) & (1 \leq i \leq n) \end{aligned}$$

(Este algoritmo estable y eficiente se debe a MacLeod [1973].)

37. Demuestre que si $S(x)$ es un spline de grado k en $[a, b]$, entonces $S'(x)$ es un spline de grado $k - 1$.

38. ¿Cuántos coeficientes se necesitan para definir una función cuártica por partes (cuarto-grado) con $n + 1$ nudos? ¿Cuántas condiciones se impondrán si la función cuártica por partes es un spline cuártico? Justifique sus respuestas.

39. Determine si esta función es un spline cúbico natural:

$$S(x) = \begin{cases} x^3 + 3x^2 + 7x - 5 & (-1 \leq x \leq 0) \\ -x^3 + 3x^2 + 7x - 5 & (0 \leq x \leq 1) \end{cases}$$

40. Determine si esta función es o *no* es un spline cúbico natural que tiene nudos en 0, 1 y 2:

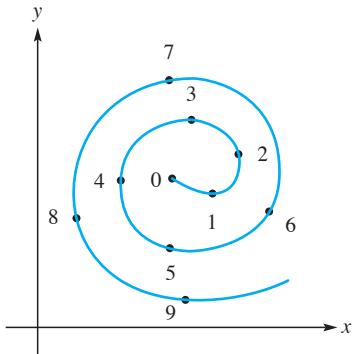
$$f(x) = \begin{cases} x^3 + x - 1 & (0 \leq x \leq 1) \\ -(x - 1)^3 + 3(x - 1)^2 + 4(x - 1) + 1 & (1 \leq x \leq 2) \end{cases}$$

41. Demuestre que el spline cúbico natural que pasa por los puntos $(0, 1), (1, 2), (2, 3), (3, 4)$ y $(4, 5)$ debe ser $y = x + 1$. (El spline cúbico natural que interpola el conjunto de puntos dados es único, ya que la matriz en la ecuación (10) es diagonalmente dominante y no singular, como se demostró en la sección 7.3.)

Problemas de cómputo 9.2

- Reescriba y pruebe el procedimiento *Spline3_Coef* usando el procedimiento *Tri* del capítulo 7. Use la simetría del sistema tridiagonal de $(n - 1) \times (n - 1)$.
- El almacenamiento extra que se requiere en el paso 1 del algoritmo para resolver el sistema tridiagonal del spline cúbico natural se puede eliminar directamente a cambio de una pequeña cantidad de cálculo extra, a saber, calculando los h_i y los b_i directamente de los t_i y y_i en la fase de eliminación hacia adelante (paso 2) y en la fase de sustitución hacia atrás (paso 3). Reescriba y pruebe el procedimiento *Spline3_Coef* utilizando esta idea.
- Usando a lo más 20 nudos y las rutinas de spline cúbico *Spline3_Coef* y *Spline3_Eval*, trace la gráfica en un graficador para bosquejar su:
 - mascota de la escuela.
 - firma.
 - perfil.
- Sea S la función spline cúbico que interpola $f(x) = (x^2 + 1)^{-1}$ en 41 nudos igualmente espaciados en el intervalo $[-5, 5]$. Evalúe $S(x) - f(x)$ en 101 puntos igualmente espaciados en el intervalo $[0, 5]$.
- Dibuje una curva de forma libre en papel cuadriculado, asegurándose de que la curva es la gráfica de una función. Entonces,lea los valores de su función en un número razonable de puntos, digamos, 10–50, y calcule la función spline cúbico que tiene esos valores. Compare la curva dibujada libremente con la gráfica del spline cúbico.

6. Dibuje un espiral (u otra curva que no sea una función) y reproduzcala por medio de funciones spline paramétricas (véase la figura a continuación).



7. Escriba y pruebe los procedimientos que sean *tan simples como sea posible* para realizar interpolación con spline cúbico natural con nudos igualmente espaciados. *Sugerencia:* vea los problemas 9.3.34–9.3.36.
8. Escriba un programa para calcular $\int_a^b f(x) dx$, suponiendo que conoce los valores de f en sólo determinados nudos $a = t_0 < t_1 < \dots < t_n = b$. Aproxime f primero mediante un spline cúbico de interpolación y después calcule la integral de este usando la ecuación (5).
9. Escriba un procedimiento para calcular $f'(x)$ para cualquier x en $[a, b]$, suponiendo que sólo conocemos los valores en los nudos $a = t_0 < t_1 < \dots < t_n = b$.
10. Usando la función de Runge $f(x) = 1/(1 + x^2)$ de la sección 4.2 con un número mayor de nudos igualmente espaciados, vea que el ajuste de la curva con el spline cúbico natural mejora, mientras que con el polinomio de interpolación empeora.
11. Use software matemático como Matlab, Maple o Mathematica para generar y trazar la función spline del ejemplo 2.
12. Use software matemático como Matlab, Maple o Mathematica para trazar la gráfica de las funciones de spline cúbico correspondientes a la
- a. figura 9.8. b. figura 9.9. c. figura 9.10.

9.3 Splines B: Interpolación y aproximación

En esta sección damos una introducción a la teoría de *splines B*, que son funciones spline especiales que se adaptan bien a las tareas numéricas y se utilizan cada vez con mayor frecuencia en la producción de programas tipo para la aproximación de datos. Así, el usuario inteligente de la biblioteca de códigos debe tener cierta familiaridad con ellos. Los splines B se llaman así porque forman una base para el conjunto de todos los splines. (Nosotros preferimos el nombre más romántico de *splines campana* que procede de su forma característica.)

En esta sección, se supone que un conjunto infinito de nudos $\{t_i\}$ se ha dado en forma tal que

$$\begin{cases} \cdots < t_{-2} < t_{-1} < t_0 < t_1 < t_2 < \cdots \\ \lim_{i \rightarrow \infty} t_i = \infty = -\lim_{i \rightarrow -\infty} t_{-i} \end{cases} \quad (1)$$

Los splines B que se definen ahora dependen de este conjunto de nudos, aunque la notación no muestre dicha dependencia. Los **splines B de grado 0** se definen como

$$B_i^0(x) = \begin{cases} 1 & t_i \leq x < t_{i+1} \\ 0 & \text{de otra manera} \end{cases} \quad (2)$$

La gráfica de B_i^0 se muestra en la figura 9.12.

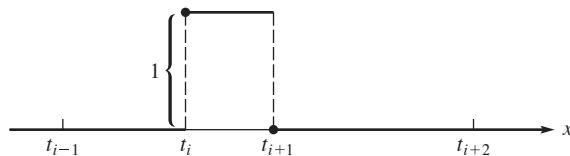


FIGURA 9.12
Spline B_i^0

Obviamente, B_i^0 es discontinua. Sin embargo, es continua desde la derecha a todos los puntos, aun donde ocurren los saltos. Así,

$$\lim_{x \rightarrow t_i^+} B_i^0(x) = 1 = B_i^0(t_i) \quad \text{y} \quad \lim_{x \rightarrow t_{i+1}^-} B_i^0(x) = 0 = B_i^0(t_{i+1})$$

Si el **apoyo** de una función f se define como el conjunto de puntos x donde $f(x) \neq 0$, entonces podemos decir que el apoyo de B_i^0 es el intervalo semiabierto $[t_i, t_{i+1})$. Puesto que B_i^0 es una función constante por partes, es un spline de grado 0.

Se pueden hacer dos observaciones:

$$\begin{aligned} B_i^0(x) &\geq 0 && \text{para toda } x \text{ y para toda } i \\ \sum_{i=-\infty}^{\infty} B_i^0(x) &= 1 && \text{para toda } x \end{aligned}$$

Aunque el segundo de estos enunciados contiene una serie infinita, no hay duda de la convergencia, porque de todos los términos de x sólo uno de la serie es diferente de 0. En efecto, para x fija, hay un único número entero m que tal $t_m \leq x < t_{m+1}$ y entonces

$$\sum_{i=-\infty}^{\infty} B_i^0(x) = B_m^0(x) = 1$$

Usted debe ahora ver la razón para haber definido B_i^0 en la forma de la ecuación (2).

Una última observación sobre estos splines B de grado 0. Cualquier spline de grado 0 que es continuo por la derecha y se basa en los nudos (1) puede expresarse como una combinación lineal de los splines B y B_i^0 . En efecto, si S es una función, entonces se puede especificar con una regla como

$$S(x) = b_i \quad \text{si } t_i \leq x < t_{i+1} \quad (i = 0, \pm 1, \pm 2, \dots)$$

Entonces S se puede escribir como

$$S = \sum_{i=-\infty}^{\infty} b_i B_i^0$$

Con las funciones B_i^0 como un punto inicial, ahora generaremos todos los splines B de grado alto con una simple definición *recursiva*:

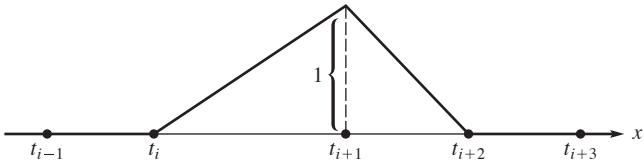
$$B_i^k(x) = \left(\frac{x - t_i}{t_{i+k} - t_i} \right) B_i^{k-1}(x) + \left(\frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x) \quad (k \geq 1) \quad (3)$$

Aquí $k = 1, 2, \dots$, e $i = 0, \pm 1, \pm 2, \dots$

Para ilustrar la ecuación (3), permítanos determinar B_i^1 en una forma alternativa:

$$\begin{aligned} B_i^1(x) &= \left(\frac{x - t_i}{t_{i+1} - t_i} \right) B_i^0(x) + \left(\frac{t_{i+2} - x}{t_{i+2} - t_{i+1}} \right) B_{i+1}^0(x) \\ &= \begin{cases} 0 & (x \geq t_{i+2} \text{ o } x \leq t_i) \\ \frac{x - t_i}{t_{i+1} - t_i} & (t_i \leq x < t_{i+1}) \\ \frac{t_{i+2} - x}{t_{i+2} - t_{i+1}} & (t_{i+1} \leq x < t_{i+2}) \end{cases} \end{aligned}$$

FIG 9.13
Spline B_i^1



La gráfica de B_i^1 se muestra en la figura 9.13. Estas algunas veces se llaman **funciones sombrero** o **funciones chapeau** (sombrero, en francés), ya que parecen un sombrero triangular que se puede hacer con un periódico. El apoyo de B_i^1 es el intervalo abierto (t_i, t_{i+2}) . Es cierto, pero quizás no es tan obvio, que

$$\sum_{i=-\infty}^{\infty} B_i^1(x) = 1 \quad \text{para toda } x$$

y que cada spline de grado 1 basado en los nudos (1) es una combinación lineal de B_i^1 .

Las funciones B_i^k como se definen con la ecuación (3) se llaman **splines B de grado k** . Ya que cada B_i^k se obtiene al aplicar factores lineales a B_i^{k-1} y B_{i+1}^{k-1} , vemos que los grados en realidad aumentan en 1 en cada paso. Por tanto, B_i^1 es lineal por partes, B_i^2 es cuadrática por partes y así sucesivamente.

Es también fácil mostrar por inducción que

$$B_i^k(x) = 0 \quad x \notin [t_i, t_{i+k+1}] \quad (k \geq 0)$$

Para determinar esto, empezamos por observar que es cierto cuando $k = 0$ por la definición (2). Si bien es cierto para el índice $k - 1$, entonces es cierto para el índice k por el siguiente razonamiento. La hipótesis de inducción nos dice que $B_i^{k-1}(x) = 0$ si x se encuentra fuera de $[t_i, t_{i+k}]$ y que $B_{i+1}^{k-1}(x) = 0$ si x está afuera de $[t_{i+1}, t_{i+k+1}]$. Si x está afuera de ambos intervalos, se encuentra fuera de su unión, $[t_i, t_{i+k+1}]$; entonces los dos términos en el miembro derecho de la ecuación (3) son 0. Así $B_i^k(x) = 0$ fuera de $[t_i, t_{i+k+1}]$. Que $B_i^k(t_i) = 0$ se sigue directamente de la ecuación (3), nos permite saber que $B_i^k(x) = 0$ para toda x fuera de (t_i, t_{i+k+1}) si $k \geq 1$.

Como complemento a la propiedad que acabamos de establecer, podemos demostrar, una vez más por inducción, que

$$B_i^k(x) > 0 \quad x \in (t_i, t_{i+k+1}) \quad (k \geq 0)$$

Por la ecuación (2), este enunciado es verdadero cuando $k = 0$. Si es cierto para el índice $k - 1$, entonces $B_i^{k-1}(x) > 0$ en (t_i, t_{i+k+1}) y $B_{i+1}^{k-1}(x) > 0$ en (t_{i+1}, t_{i+k+1}) . En la ecuación (3), los factores que se multiplican por $B_i^{k-1}(x)$ y $B_{i+1}^{k-1}(x)$ son positivos cuando $t_i < x < t_{i+k+1}$. Por lo tanto, $B_i^k(x) > 0$ en este intervalo.

La figura 9.14 muestra los primeros cuatro splines B trazados en los mismos ejes.

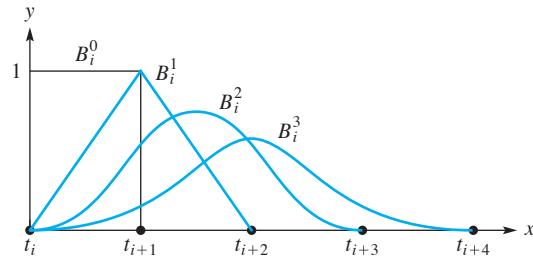


FIGURA 9.14
Primeros cuatro
splines B

El uso principal de los splines B $B_i^k (i = 0, \pm 1, \pm 2, \dots)$ es como una base para el conjunto de todos los splines de k ésmo grado que tengan la misma sucesión de nudos. Así, las combinaciones lineales

$$\sum_{i=-\infty}^{\infty} c_i B_i^k$$

son importantes objetos de estudio. (Usamos c_i para una k fija y C_i^k para enfatizar el grado k de los splines B correspondientes). Nuestra primera tarea es desarrollar un método eficiente para evaluar una función de la forma

$$f(x) = \sum_{i=-\infty}^{\infty} C_i^k B_i^k(x) \quad (4)$$

suponiendo que los coeficientes C_i^k están dados (así como también la sucesión de nudos). Partiendo de la definición (3) y con algunas simples manipulaciones de la serie, tenemos

$$\begin{aligned} f(x) &= \sum_{i=-\infty}^{\infty} C_i^k \left[\left(\frac{x - t_i}{t_{i+k} - t_i} \right) B_i^{k-1}(x) + \left(\frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x) \right] \\ &= \sum_{i=-\infty}^{\infty} \left[C_i^k \left(\frac{x - t_i}{t_{i+k} - t_i} \right) + C_{i-1}^k \left(\frac{t_{i+k} - x}{t_{i+k} - t_i} \right) \right] B_i^{k-1}(x) \\ &= \sum_{i=-\infty}^{\infty} C_i^{k-1} B_i^{k-1}(x) \end{aligned} \quad (5)$$

donde C_i^{k-1} está definido como el coeficiente adecuado de la ecuación anterior (5).

Esta manipulación algebraica muestra cómo una combinación lineal de $B_i^k(x)$ se puede expresar como una combinación lineal de $B_i^{k-1}(x)$. Repitiendo este proceso $k - 1$ veces, finalmente expresamos $f(x)$ en la forma

$$f(x) = \sum_{i=-\infty}^{\infty} C_i^0 B_i^0(x) \quad (6)$$

Si $t_m \leq x < t_{m+1}$, entonces $f(x) = C_m^0$. La fórmula con la que se obtienen los coeficientes C_i^{j-1} es

$$C_i^{j-1} = C_i^j \left(\frac{x - t_i}{t_{i+j} - t_i} \right) + C_{i-1}^j \left(\frac{t_{i+j} - x}{t_{i+j} - t_i} \right) \quad (7)$$

Una buena característica de la ecuación (4) es que sólo los coeficientes $k+1$ $C_m^k, C_{m-1}^k, \dots, C_{m-k}^k$ se necesitan para calcular $f(x)$ si $t_m \leq x < t_{m+1}$ (véase el problema 9.3.6). Por lo tanto, si f se define por la ecuación (4) y queremos calcular $f(x)$, utilizamos la ecuación (7) para el cálculo de las entradas de la matriz triangular siguiente:

$$\begin{matrix} C_m^k & C_m^{k-1} & \cdots & C_m^0 \\ C_{m-1}^k & C_{m-1}^{k-1} & \ddots & \\ \vdots & \ddots & & \\ C_{m-k}^k & & & \end{matrix}$$

Aunque nuestra notación no lo muestra, los coeficientes de la ecuación (4) son independientes de x , mientras que los C_i^{j-1} calculados posteriormente por la ecuación (7) no dependen de x .

Ahora es un asunto sencillo establecer que

$$\sum_{i=-\infty}^{\infty} B_i^k(x) = 1 \quad \text{para toda } x \text{ y para toda } k \geq 0$$

Si $k=0$, ya sabemos esto. Si $k>0$, utilizamos la ecuación (4) con $C_i^k=1$ para toda i . Por la ecuación (7), todos los coeficientes siguientes $C_i^k, C_i^{k-1}, C_i^{k-2}, \dots, C_i^0$ también son iguales a 1 (¡aquí se necesita inducción!). Así, al final, la ecuación (6) es verdadera con $C_i^0=1$, y así $f(x)=1$. Por lo tanto, de la ecuación (4), la suma de todos los splines B de grado k es la unidad.

La suavidad de los splines B, B_i^k aumenta con el índice k . De hecho, podemos demostrar por inducción que B_i^k tiene una $(k-1)$ ésima derivada continua.

Los splines B se pueden usar como sustitutos para funciones complicadas en muchas situaciones matemáticas. La derivación y la integración son importantes ejemplos. Un resultado básico acerca de las **derivadas de los splines B** es

$$\frac{d}{dx} B_i^k(x) = \left(\frac{k}{t_{i+k} - t_i} \right) B_i^{k-1}(x) - \left(\frac{k}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x) \quad (8)$$

Esta ecuación se puede demostrar por inducción usando la fórmula recursiva (3). Una vez que se ha establecido la ecuación (8), obtenemos la útil fórmula

$$\frac{d}{dx} \sum_{i=-\infty}^{\infty} c_i B_i^k(x) = \sum_{i=-\infty}^{\infty} d_i B_i^{k-1}(x) \quad (9)$$

donde

$$d_i = k \left(\frac{c_i - c_{i-1}}{t_{i+k} - t_i} \right)$$

La comprobación es la siguiente. Por la ecuación (8),

$$\begin{aligned}
 & \frac{d}{dx} \sum_{i=-\infty}^{\infty} c_i B_i^k(x) \\
 &= \sum_{i=-\infty}^{\infty} c_i \frac{d}{dx} B_i^k(x) \\
 &= \sum_{i=-\infty}^{\infty} c_i \left[\left(\frac{k}{t_{i+k} - t_i} \right) B_i^{k-1}(x) - \left(\frac{k}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x) \right] \\
 &= \sum_{i=-\infty}^{\infty} \left[\left(\frac{c_i k}{t_{i+k} - t_i} \right) - \left(\frac{c_{i-1} k}{t_{i+k} - t_i} \right) \right] B_i^{k-1}(x) \\
 &= \sum_{i=-\infty}^{\infty} d_i B_i^{k-1}(x)
 \end{aligned}$$

Por integración numérica, los splines B son también recomendables, especialmente para integración indefinida. Este es el resultado básico necesario para la integración:

$$\int_{-\infty}^x B_i^k(s) ds = \left(\frac{t_{i+k+1} - t_i}{k + 1} \right) \sum_{j=i}^{\infty} B_j^{k+1}(x) \quad (10)$$

Esta ecuación se puede comprobar derivando ambos miembros con respecto a x y simplificar usando la ecuación (9). Para asegurarnos de que los dos lados de la ecuación (10) no difieren por una constante, observamos que para cualquier $x < t_i$, ambos se reducen a cero.

El resultado básico (10) produce esta útil fórmula:

$$\int_{-\infty}^x \sum_{i=-\infty}^{\infty} c_i B_i^k(s) ds = \sum_{i=-\infty}^{\infty} e_i B_i^{k+1}(x) \quad (11)$$

donde

$$e_i = \frac{1}{k + 1} \sum_{j=-\infty}^i c_j (t_{j+k+1} - t_j)$$

Cabe destacar que esta fórmula da una integral indefinida (primitiva) de cualquier función expresada como una combinación lineal de splines B. Cualquier integral definida se puede obtener seleccionando un valor específico de x . Por ejemplo, si x es un nudo, digamos, $x = t_m$, entonces

$$\int_{-\infty}^{t_m} \sum_{i=-\infty}^{\infty} c_i B_i^k(s) ds = \sum_{i=-\infty}^{\infty} e_i B_i^{k+1}(t_m) = \sum_{i=m-k-1}^m e_i B_i^{k+1}(t_m)$$

Matlab tiene una caja de herramientas *Spline*, desarrollada por Carl de Boor, que se puede usar para muchas tareas que impliquen a los splines. Por ejemplo, las rutinas de interpolación de datos con splines con diversas condiciones de extremo y rutinas para el ajuste con mínimos cuadrados de los datos. Hay muchas rutinas de demostración en esa caja de herramientas que presentan gráficas y proporciona modelos para la programación con archivos M de Matlab. Estas demostraciones son muy instructivas para la visualización y el aprendizaje de los conceptos de la teoría de splines, especialmente de los splines B.

Maple tiene el paquete *Bspline* para la construcción de las funciones base de splines B de grado k de una lista de nudos dada, que pueden incluir nudos múltiples. Se basa en una implemen-

tación de diferencia dividida que se encuentra en Bartels, Beatty y Barskey [1987]. Se puede bajar del Centro de aplicaciones de Maple en www.maplesoft.com.

Interpolación y aproximación con splines B

Hemos desarrollado una serie de propiedades de los splines B y mostrado cómo se utilizan en las tareas numéricas. El problema de obtener una representación de spline B de una función dada no se analizó. Aquí, consideramos el problema de la interpolación de una tabla de datos; más tarde, se describe un método de aproximación de no interpolación.

Una cuestión básica es cómo determinar los coeficientes en la expresión

$$S(x) = \sum_{i=-\infty}^{\infty} A_i B_{i-k}^k(x) \quad (12)$$

de manera que la función spline resultante interpola una tabla dada:

x	t_0	t_1	\cdots	t_n
y	y_0	y_1	\cdots	y_n

Se entiende por *interpolar* que

$$S(t_i) = y_i \quad (0 \leq i \leq n) \quad (13)$$

El punto de partida natural es con los splines simples, correspondientes a $k = 0$. Dado que

$$B_i^0(t_j) = \delta_{ij} = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases}$$

la solución al problema es inmediata: se establece $A_i = y_i$ para $0 \leq i \leq n$. Todos los demás coeficientes en la ecuación (12) son arbitrarios. En particular, pueden ser cero. Llegamos entonces a este resultado: el spline B de grado cero

$$S(x) = \sum_{i=0}^n y_i B_i^0(x)$$

tiene la propiedad de interpolación (13).

El siguiente caso, $k = 1$, también tiene una solución simple. Usamos el hecho que

$$B_{i-1}^1(t_j) = \delta_{ij}$$

Por lo tanto, se cumple lo siguiente: el spline B de primer grado

$$S(x) = \sum_{i=0}^n y_i B_{i-1}^1(x)$$

tiene la propiedad de interpolación (13). De modo que de nuevo $A_i = y_i$.

Si la tabla tiene cuatro entradas ($n = 3$), por ejemplo, usamos B_{-1}^1, B_0^1, B_1^1 y B_2^1 . Estos, a su vez, requieren para su definición a los nudos $t_{-1}, t_0, t_1, \dots, t_4$. Los nudos t_{-1} y t_4 pueden ser arbitrarios. En la figura 9.15 se muestran las gráficas de los cuatro splines B^1 . En tal problema, si t_{-1} y t_4 no están dados, es natural que se definan de manera que t_0 sea el punto medio del intervalo $[t_{-1}, t_1]$ y t_3 sea el punto medio de $[t_2, t_4]$.

En ambos casos elementales considerados, los coeficientes desconocidos A_0, A_1, \dots, A_n en la ecuación (12) fueron determinados únicamente por las condiciones de interpolación (13). Si los

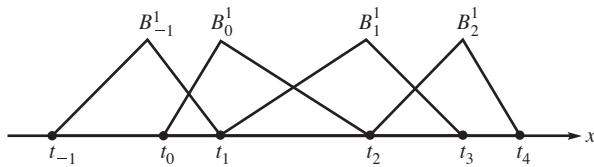


FIGURA 9.15
Splines B_i^1

términos presentes en la ecuación (12) corresponden a valores de i fuera del rango $\{0, 1, \dots, n\}$, entonces no tendrían influencia en los valores de $S(x)$ en t_0, t_1, \dots, t_n .

Para splines de mayor grado, veremos que existe cierta arbitrariedad en la elección de coeficientes. De hecho, *ninguno* de los coeficientes está únicamente determinado por las condiciones de interpolación. Este hecho puede ser ventajoso si se desean otras propiedades de la solución. En el caso cuadrático, comenzamos con la ecuación

$$\sum_{i=-\infty}^{\infty} A_i B_{i-2}^2(t_j) = \frac{1}{t_{j+1} - t_{j-1}} [A_j(t_{j+1} - t_j) + A_{j+1}(t_j - t_{j-1})] \quad (14)$$

Su justificación se deja en el problema 9.3.26. Si ahora se imponen las condiciones de interpolación (13), obtenemos el siguiente sistema de ecuaciones, que da las condiciones necesarias y suficientes sobre los coeficientes

$$A_j(t_{j+1} - t_j) + A_{j+1}(t_j - t_{j-1}) = y_j(t_{j+1} - t_{j-1}) \quad (0 \leq j \leq n) \quad (15)$$

Este es un sistema de $n + 1$ ecuaciones lineales con $n + 2$ incógnitas A_0, A_1, \dots, A_{n+1} .

Una forma de resolver la ecuación (15) es asignar cualquier valor a A_0 y después usar la ecuación (15) para calcular a A_1, A_2, \dots, A_{n+1} , recursivamente. Con este propósito, las ecuaciones se podrían reescribir como

$$A_{j+1} = a_j + \beta_j A_j \quad (0 \leq j \leq n) \quad (16)$$

donde se han usado estas abreviaturas:

$$\begin{cases} a_j = y_j \left(\frac{t_{j+1} - t_{j-1}}{t_j - t_{j-1}} \right) \\ \beta_j = \frac{t_j - t_{j+1}}{t_j - t_{j-1}} \end{cases} \quad (0 \leq j \leq n)$$

Para conservar los coeficientes con magnitud pequeña, recomendamos seleccionar A_0 tal que la expresión

$$\Phi = \sum_{i=0}^{n+1} A_i^2$$

será un mínimo. Para determinar este valor de A_0 , procedemos en la forma siguiente. Por sustitución sucesiva usando la ecuación (16) podemos demostrar que

$$A_{j+1} = \gamma_j + \delta_j A_0 \quad (0 \leq j \leq n) \quad (17)$$

donde los coeficientes γ_i y δ_j se obtienen recursivamente con este algoritmo:

$$\begin{cases} \gamma_0 = \alpha_0 & \delta_0 = \beta_0 \\ \gamma_j = \alpha_j + \beta_j \gamma_{j-1} & \delta_j = \beta_j \delta_{j-1} \end{cases} \quad (1 \leq j \leq n) \quad (18)$$

Entonces Φ es una función cuadrática de A_0 como sigue:

$$\begin{aligned} \Phi &= A_0^2 + A_1^2 + \cdots + A_{n+1}^2 \\ &= A_0^2 + (\gamma_0 + \delta_0 A_0)^2 + (\gamma_1 + \delta_1 A_0)^2 + \cdots + (\gamma_n + \delta_n A_0)^2 \end{aligned}$$

Para encontrar el mínimo de Φ , tomamos su derivada con respecto a A_0 y la hacemos igual a cero:

$$\frac{d\Phi}{dA_0} = 2A_0 + 2(\gamma_0 + \delta_0 A_0)\delta_0 + 2(\gamma_1 + \delta_1 A_0)\delta_1 + \cdots + 2(\gamma_n + \delta_n A_0)\delta_n = 0$$

Esto equivale a $qA_0 + p = 0$, donde

$$\begin{cases} q = 1 + \delta_0^2 + \delta_1^2 + \cdots + \delta_n^2 \\ p = \gamma_0 \delta_0 + \gamma_1 \delta_1 + \cdots + \gamma_n \delta_n \end{cases}$$

Seudocódigo y ejemplo de un ajuste de curva

Ahora se presenta un procedimiento que calcula los coeficientes A_0, A_1, \dots, A_{n+1} en la forma indicada anteriormente. En su secuencia de llamada, $(t_i)_{0:n}$ es el arreglo de nudos, $(y_i)_{0:n}$ es el arreglo de puntos de abscisas, $(a_i)_{0:n+1}$ es el arreglo de coeficientes A_i y $(h_i)_{0:n+1}$ es el arreglo que contiene $h_i = t_i - t_{i-1}$. Sólo $n, (t_i)$ y (y_i) son valores de entrada. Están disponibles sin cambios cuando se termina la rutina. Los arreglos (a_i) y (h_i) se calculan y están disponibles como salida.

```

procedure BSpline2_Coef(n,(ti),(yi),(ai),(hi))
integer i,n; real δ,γ,p,q
real array (ai)0:n+1,(hi)0:n+1,(ti)0:n,(yi)0:n
for i = 1 to n do
    hi ← ti - ti-1
end for
h0 ← h1
hn+1 ← hn
δ ← -1
γ ← 2y0
p ← δγ
q ← 2
for i = 1 to n do
    r ← hi+1/ hi
    δ ← -rδ
    γ ← -rγ + (r + 1)yi
    p ← p + γδ
    q ← q + δ2
end for

```

```

 $a_0 \leftarrow -p/q$ 
for  $i = 1$  to  $n + 1$  do
     $a_i \leftarrow [(h_{i-1} + h_i)y_{i-1} - h_i a_{i-1}]/h_{i-1}$ 
end for
end procedure BSpline2_Coef

```

A continuación se presenta una función de procedimiento *BSpline2_Eval* para calcular los valores del spline cuadrático dado por $S(x) = \sum_{i=0}^{n+1} A_i B_{i-2}^2(x)$. Su secuencia de llamada tiene algunas de las mismas variables que en el seudocódigo anterior. La variable de entrada x es un número real único que debe quedar entre t_0 y t_n . Se usa el resultado del problema 9.3.26.

```

real function BSpline2_Eval( $n, (t_i), (a_i), (h_i), x$ )
integer  $i, n$ ; real  $d, e, x$ ; real array  $(a_i)_{0:n+1}, (h_i)_{0:n+1}, (t_i)_{0:n}$ 
for  $i = n - 1$  to  $0$  step  $-1$  do
    if  $x - t_i \geq 0$  then exit loop
end for
 $i \leftarrow i + 1$ 
 $d \leftarrow [a_{i+1}(x - t_{i-1}) + a_i(t_i - x + h_{i+1})]/(h_i + h_{i+1})$ 
 $e \leftarrow [a_i(x - t_{i-1} + h_{i-1}) + a_{i-1}(t_{i-1} - x + h_i)]/(h_{i-1} + h_i)$ 
 $BSpline2_Eval \leftarrow [d(x - t_{i-1}) + e(t_i - x)]/h_i$ 
end function BSpline2_Eval

```

Usando la tabla de 20 puntos de la sección 9.2, podemos comparar la curva resultante del spline cúbico natural con el spline cuadrático producido por los procedimientos *BSpline2_Coef* y *BSpline2_Eval*. La primera de estas curvas se muestra en la figura 9.8 y la segunda está en la figura 9.16. La última es razonable, pero quizás no tan agradable como la primera. Estas curvas muestran una vez más que las funciones de splines cúbicos naturales son elegantes y simples para el ajuste de curvas.

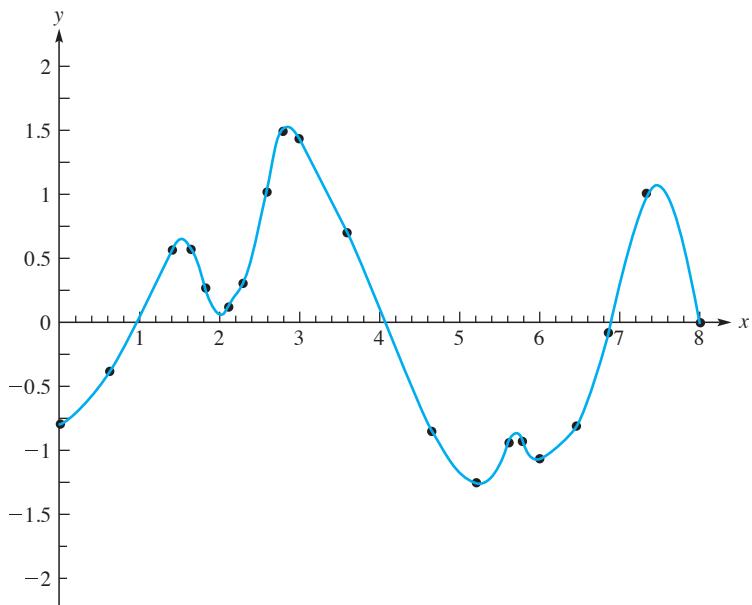


FIGURA 9.16
Spline cuadrático
de interpolación

Proceso de Schoenberg

Un proceso eficiente debido a Schoenberg [1967] también se puede utilizar para obtener aproximaciones del spline B para una función dada. Su versión cuadrática está definida por

$$S(x) = \sum_{i=-\infty}^{\infty} f(\tau_i) B_i^2(x) \quad \text{donde} \quad \tau_i = \frac{1}{2}(t_{i+1} + t_{i+2})$$

En este caso, por supuesto, los nudos son $\{t_i\}_{i=-\infty}^{\infty}$ y los puntos donde se debe evaluar f son los puntos medios entre los nudos.

La ecuación (19) es útil en la producción de una función spline cuadrático que se approxima a f . Las características sobresalientes de este proceso son los siguientes:

1. Si $f(x) = ax + b$, entonces $S(x) = f(x)$.
2. Si $f(x) \geq 0$ en todas partes, entonces $S(x) \geq 0$ en todas partes.
3. $\max_x |S(x)| \leq \max_x |f(x)|$.
4. Si f es continua en $[a, b]$, si $\delta = \max_i |t_{i+1} - t_i|$ y si $\delta < b - a$, entonces para x en $[a, b]$,

$$|S(x) - f(x)| \leq \frac{3}{2} \max_{a \leq u \leq v \leq u+\delta \leq b} |f(u) - f(v)|$$

5. La gráfica de S no corta ninguna línea en el plano un número de veces mayor de lo que lo hace la gráfica de f .

Algunas de estas propiedades son elementales, mientras que otras son más abstractas. La propiedad 1 se describe en el problema 9.3.29. La propiedad 2 es evidente, porque $B_i^2(x) \geq 0$ para toda x . La propiedad 3 se deduce fácilmente de la ecuación (19), porque si $|f(x)| \leq M$, entonces

$$|S(x)| \leq \left| \sum_{i=-\infty}^{\infty} f(\tau_i) B_i^2(x) \right| \leq \sum_{i=-\infty}^{\infty} |f(\tau_i)| B_i^2(x) \leq M \sum_{i=-\infty}^{\infty} B_i^2(x) = M$$

Las propiedades 4 y 5 se aceptarán sin demostración. Sin embargo, su importancia no debe pasarse por alto. Por la propiedad 4, podemos hacer que la función S sea cercana a una función continua f simplemente haciendo que el *tamaño de la malla* δ sea pequeño. Esto se debe a que $|f(u) - f(v)|$ puede ser tan pequeño como queramos, simplemente imponiendo la desigualdad $|u - v| \leq \delta$ (propiedad de continuidad uniforme). La propiedad 5 se puede interpretar como una forma de conservación de atributo del proceso de aproximación. En una interpretación cruda, S no debería mostrar más ondulaciones que f .

Seudocódigo

Aquí se desarrolla un seudocódigo para obtener una aproximación spline por medio del proceso de Schoenberg. Suponga que f está definida en un intervalo $[a, b]$ y que se quiere la aproximación spline de la ecuación (19) en el mismo intervalo. Después definimos los *nudos* $\tau_i = a + ih$, donde $h = (b - a)/n$. Aquí, i puede ser cualquier número entero, pero los nudos en $[a, b]$ son sólo $\tau_0, \tau_1, \dots, \tau_n$. Para que $\tau_i = \frac{1}{2}(t_{i+1} + t_{i+2})$, definimos los *nudos* $t_i = a + (i - \frac{3}{2})h$. En la ecuación (19), sólo los splines B_i^2 que están *activos* en $[a, b]$ son $B_{-1}^2, B_0^2, \dots, B_{n+1}^2$. Por lo tanto, para nuestros propósitos, la ecuación (19) se convierte en

$$S(x) = \sum_{i=-1}^{n+1} f(\tau_i) B_i^2(x) \tag{20}$$

Por lo tanto, se requieren los valores de f en $\tau_{-1}, \tau_0, \dots, \tau_{n+1}$. Dos de estos nodos se encuentran fuera del intervalo $[a, b]$; por lo tanto, proporcionamos valores extrapolados linealmente en el código al definir

$$\begin{aligned}f(\tau_{-1}) &= 2f(\tau_0) - f(\tau_1) \\f(\tau_{n+1}) &= 2f(\tau_n) - f(\tau_{n-1})\end{aligned}$$

Para usar las fórmulas del problema 9.3.26, escribimos

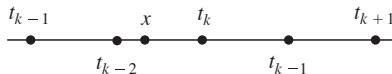
$$S(x) = \sum_{i=1}^{n+3} D_i B_{i-2}^2(x) \quad [D_i = f(\tau_{i-2})]$$

Ahora se presenta un seudocódigo para calcular D_1, D_2, \dots, D_{n+3} . En la secuencia de llamado para el procedimiento *Schoenberg_Coef*, f es una función externa. Después de la ejecución, los $n+3$ coeficientes deseados están en el arreglo (d_i) .

```
procedure Schoenberg_Coef( $f, a, b, n, (d_i)$ )
integer  $i$ ; real  $a, b, h$ ; real array  $(d_i)_{1:n+3}$ 
external function  $f$ 
 $h \leftarrow (b - a)/n$ 
for  $i = 2$  to  $n + 2$  do
     $d_i \leftarrow f(a + (i - 2)h)$ 
end for
 $d_1 \leftarrow 2d_2 - d_3$ 
 $d_{n+3} \leftarrow 2d_{n+2} - d_{n+1}$ 
end procedure Schoenberg_Coef
```

Después de que se han obtenido los coeficientes D_i con el procedimiento que se acaba de dar, se pueden recuperar valores del spline $S(x)$ en la ecuación (20). Aquí, nosotros usamos el algoritmo del problema 9.3.26. Dado un x , primero necesitamos saber dónde está con respecto a los nudos. Para determinar k tal que $t_{k-1} \leq x \leq t_k$, observamos que k debe ser el mayor número entero tal que $t_{k-1} \leq x$. Esta desigualdad es equivalente a la desigualdad $k \leq \frac{s}{2} + (x - a)/h$, como se comprueba fácilmente. Esto explica los cálculos de k en el seudocódigo. La ubicación de x se indica en la figura 9.17. En la secuencia de llamada para la función de *Schoenberg_Eval*, a y b son los extremos del intervalo y x es un punto donde se desea el valor de $S(x)$. El procedimiento determina los nudos t_i de tal manera que los puntos igualmente espaciados τ_i en el procedimiento anterior satisfacen $\tau_i = \frac{1}{2}(t_{i+1} + t_{i+2})$.

FIGURA 9.17
Localización
de x



```
real function Schoenberg_Eval( $a, b, n, (d_i), x$ )
integer  $k$ ; real  $c, h, p, w$ ; real array  $(d_i)_{1:n+3}$ 
 $h \leftarrow (b - a)/n$ 
 $k \leftarrow \text{integer}[(x - a)/h + 5/2]$ 
 $p \leftarrow x - a - (k - 5/2)h$ 
 $c \leftarrow [d_{k+1}p + d_k(2h - p)]/(2h)$ 
 $e \leftarrow [d_k(p + h) + d_{k-1}(h - p)]/(2h)$ 
 $Schoenberg\_Eval \leftarrow [cp + e(h - p)]/h$ 
end function Schoenberg_Eval
```

Curvas de Bézier

En el diseño asistido por computadora es útil disponer de un procedimiento para la producción de una curva que pasa por (o cerca de) algunos **puntos de control**, o una curva que se puede manipular fácilmente para dar una forma deseada. La interpolación polinomial de alto grado generalmente no es adecuada para este tipo de tarea, como se podría adivinar por los anteriores comentarios negativos acerca de ella. La experiencia demuestra que si se especifica un número de puntos de control por los que debe pasar el polinomio, la forma general de la curva resultante puede ser muy decepcionante!

Sin embargo, los polinomios se pueden utilizar de una manera diferente, que conduce a las **curvas de Bézier**. Las curvas de Bézier se utilizan como base para el espacio Π_n (todos los polinomios de grado menor que n) un conjunto especial de polinomios que se prestan a la tarea que se encara. Estandarizamos el intervalo $[0, 1]$ y fijamos un valor n . Entonces, se definen las funciones base del polinomio

$$\varphi_{ni}(x) = \binom{n}{i} x^i (1-x)^{n-i} \quad (0 \leq i \leq n)$$

Los polinomios φ_{ni} constituyen los **polinomios de Bernstein**. Para una función continua f definida en $[0, 1]$, Bernstein, en 1912, demostró que la sucesión de polinomios

$$p_n(x) = \sum_{i=0}^n f\left(\frac{i}{n}\right) \varphi_{ni}(x) \quad (n \geq 1)$$

converge uniformemente a f , con lo que se proporciona una demostración muy atractiva del teorema de aproximación de Weierstrass.

En la figura 9.18 se muestran las gráficas de algunos polinomios φ_{ni} , donde usamos $n = 7$ e $i = 0, 1, 5$. Por ejemplo, los polinomios base de Bernstein se encuentran en sistemas de software matemático como Maple o Mathematica.

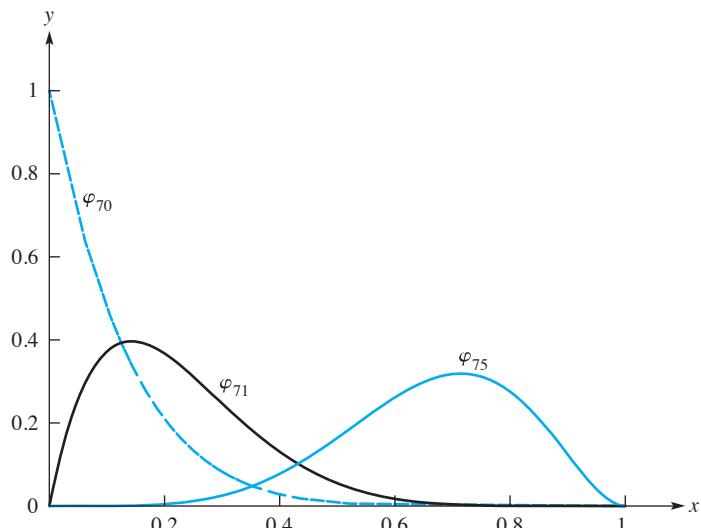


FIGURA 9.18

Algunos de los primeros polinomios base de Bernstein

Los polinomios de Bernstein tienen dos importantes propiedades

■ PROPIEDADES

Para toda x que satisface $0 \leq x \leq 1$,

1. $\varphi_{ni}(x) \geq 0$
2. $\sum_{i=0}^n \varphi_{ni}(x) = 1$

Cualquier conjunto de funciones que tenga estas dos propiedades es llamado **partición de la unidad** en el intervalo $[0, 1]$. Observe que la segunda ecuación anterior es realmente válida para todas las x . El conjunto $\{\varphi_{n0}, \varphi_{n1}, \dots, \varphi_{nn}\}$ es una base para el espacio Π_n . Por lo tanto, todo polinomio de grado a lo más n tiene una representación

$$\sum_{i=0}^n a_i \varphi_{ni}(x)$$

Si queremos crear un polinomio que se aproxime a los valores de interpolación $(i/n, y_i)$ para $0 \leq i \leq n$, podemos utilizar $\sum_{i=0}^n y_i \varphi_{ni}$ para comenzar y entonces, después de examinar la curva resultante, ajustar los coeficientes para cambiar la forma de ella. Este es un procedimiento que se puede usar en diseño asistido por computadora. Al cambiar el valor de y_i cambiará la curva, principalmente en la vecindad de i/n debido a la naturaleza local de los polinomios base φ_{ni} .

Otra forma en la que se pueden utilizar estos polinomios es en la creación de curvas que no son simplemente las gráficas de una función f . Ahora, pasamos a una forma vectorial del procedimiento sugerido antes. Si se dan $n + 1$ vectores v_0, v_1, \dots, v_n , digamos, en \mathbb{R}^2 o \mathbb{R}^3 , la expresión

$$u(t) = \sum_{i=0}^n \varphi_{ni}(t) v_i \quad (0 \leq t \leq 1)$$

tiene sentido, ya que el miembro derecho es (para cada t) una combinación lineal de los vectores v_i . Conforme t corre por el intervalo $[0, 1]$, el vector $u(t)$ describe una curva en el espacio donde se encuentran los vectores v_i . Esta curva se encuentra en la **envoltura convexa** de los vectores v_i , ya que $u(t)$ es una combinación lineal **convexa** de los v_i . Esto requiere las dos propiedades de φ_{ni} antes mencionadas.

Para mostrar este procedimiento hemos seleccionado siete puntos en el plano y se ha dibujado la curva cerrada generada por la ecuación anterior, es decir, por el vector $u(t)$. La figura 9.19 muestra la curva resultante, así como los puntos de control. En la Figura 9.19, los puntos de control

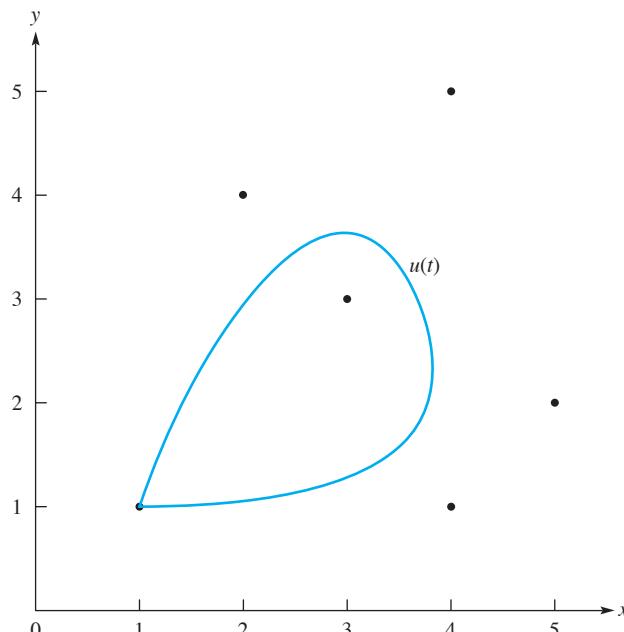


FIGURA 9.19
Curva usando
los puntos de
control

son los vértices del polígono y la curva es la que resulta en la forma descrita. Para hacer esto, se pueden utilizar sistemas de software matemático como Maple y Mathematica.

Una mirada a la figura 9.18 sugiere al lector que tal vez se puedan utilizar los splines B en el papel de las funciones de Bernstein φ_{ni} . De hecho, este es el caso y los splines B se han hecho cargo en la mayoría de los programas de diseño asistido por computadora. Así, para obtener una curva que esté cerca de un conjunto de puntos (t_i, y_i) , podemos establecer un sistema de splines B (por ejemplo, splines cúbicos B) con nudos t_i . Entonces la combinación lineal $\sum_{i=0}^n y_i B_i^3$ se puede examinar para ver si tiene la forma deseada. Aquí, por supuesto, B_i^3 denota un spline B cúbico, cuyo apoyo es el intervalo (t_i, t_{i+4}) .

El caso vectorial es parecido al descrito anteriormente, excepto que las funciones φ_{ni} se sustituyen por B_i^3 . Además, es más fácil tomar los nudos como enteros y dejar correr a t de 0 a n . Las propiedades 1 y 2 antes presentadas de las φ_{ni} son también compartidas por los splines B.

Resumen

(1) El spline B de grado 0 es

$$B_i^0(x) = \begin{cases} 1 & (t_i \leq x < t_{i+1}) \\ 0 & (\text{de otra forma}) \end{cases}$$

Los splines B de grado superior se definen recursivamente:

$$B_i^k(x) = \left(\frac{x - t_i}{t_{i+k} - t_i} \right) B_i^{k-1}(x) + \left(\frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x)$$

donde $k = 1, 2, \dots$; $i = 0, \pm 1, \pm 2, \dots$

(2) Algunas propiedades son

$$\begin{aligned} B_i^k(x) &= 0 & x \notin [t_i, t_{i+k+1}] \\ B_i^k(x) &> 0 & x \in (t_i, t_{i+k+1}) \end{aligned}$$

Un método eficiente de evaluar una función de la forma

$$f(x) = \sum_{i=-\infty}^{\infty} C_i^k B_i^k(x)$$

es usar

$$C_i^{j-1} = C_i^j \left(\frac{x - t_i}{t_{i+j} - t_i} \right) + C_{i-1}^j \left(\frac{t_{i+j} - x}{t_{i+j} - t_{i+1}} \right)$$

(3) La derivada de los splines B es

$$\frac{d}{dx} B_i^k(x) = \left(\frac{k}{t_{i+k} - t_i} \right) B_i^{k-1}(x) - \left(\frac{k}{t_{i+k+1} - t_{i+1}} \right) B_{i+1}^{k-1}(x)$$

Una fórmula útil es

$$\frac{d}{dx} \sum_{i=-\infty}^{\infty} c_i B_i^k(x) = \sum_{i=-\infty}^{\infty} d_i B_i^{k-1}(x)$$

donde $d_i = k(c_i - c_{i-1})/(t_{i+k} - t_i)$. Un resultado básico necesario para la integración es

$$\int_{-\infty}^x B_i^k(s) ds = \left(\frac{t_{i+k+1} - t_i}{k+1} \right) \sum_{j=i}^{\infty} B_j^{k+1}(x)$$

Una fórmula resultante útil es

$$\int_{-\infty}^x \sum_{i=-\infty}^{\infty} c_i B_i^k(s) ds = \sum_{i=-\infty}^{\infty} e_i B_i^{k+1}(x)$$

donde $e_i = 1/(k+1) \sum_{j=-\infty}^i c_j (t_{j+k+1} - t_j)$.

(4) Para determinar los coeficientes en la expresión

$$S(x) = \sum_{i=-\infty}^{\infty} A_i B_{i-k}^2(x)$$

y para que la función spline resultante interpole una tabla dada, usamos la condición

$$A_j(t_{j+1} - t_j) + A_{j+1}(t_j - t_{j-1}) = y_j(t_{j+1} - t_{j-1}) \quad (0 \leq j \leq n)$$

Este es un sistema de $n+1$ ecuaciones lineales con $n+2$ incógnitas A_0, A_1, \dots, A_{n+1} que se puede resolver recursivamente.

(5) El proceso de Schoenberg es eficiente para obtener las aproximaciones de spline B para una función dada. Por ejemplo, su versión cuadrática está definida por

$$S(x) = \sum_{i=-\infty}^{\infty} f(\tau_i) B_i^2(x)$$

donde $\tau_i = \frac{1}{2}(t_{i+1} + t_{i+2})$ y los nudos son $\{t_i\}_{i=-\infty}^{\infty}$. Los puntos τ_i , donde f se debe evaluar, son los puntos medios entre los nudos.

(6) Las curvas de Bézier se usan en diseño asistido por computadora para producir una curva que pasa por (o cerca de) los *puntos de control*, o una curva que se puede manejar fácilmente para dar una forma deseada. Las curvas de Bézier usan *polinomios de Bernstein*. Para una función continua f definida en $[0, 1]$, la sucesión de los polinomios de Bernstein

$$p_n(x) = \sum_{i=0}^n f\left(\frac{i}{n}\right) \varphi_{ni}(x) \quad (n \geq 1)$$

converge uniformemente a f . Los polinomios φ_{ni} son

$$\varphi_{ni}(x) = \binom{n}{i} x^i (1-x)^{n-i} \quad (0 \leq i \leq n)$$

Referencias adicionales

Véase Ahlberg *et al.* [1967], DeBoor [1978], Farin [1990], MacLeod [1973], Schoenberg [1946, 1967], Schultz [1973], Schumaker [1981], Subbotin [1967] y Yamaguchi [1988].

Problemas 9.3

- 1.** Demuestre que las funciones $f_n(x) = \cos nx$ son generadas por esta definición recursiva:

$$\begin{cases} f_0(x) = 1, & f_1(x) = \cos x \\ f_{n+1}(x) = 2f_1(x)f_n(x) - f_{n-1}(x) & (n \geq 1) \end{cases}$$

- 2.** ¿Qué funciones se generan por la siguiente definición recursiva?

$$\begin{cases} f_0(x) = 1, & f_1(x) = x \\ f_{n+1}(x) = 2xf_n(x) - f_{n-1}(x) & (n \geq 1) \end{cases}$$

- 3.** Encuentre una expresión para $B_i^2(x)$ y compruebe que es cuadrática por partes. Demuestre que $B_i^2(x)$ es cero en todos los nudos, excepto

$$B_i^2(t_{i+1}) = \frac{t_{i+1} - t_i}{t_{i+2} - t_i} \quad \text{y} \quad B_i^2(t_{i+2}) = \frac{t_{i+3} - t_{i+2}}{t_{i+3} - t_{i+1}}$$

- 4.** Compruebe la ecuación (5).

- 5.** Establezca que $\sum_{i=-\infty}^{\infty} f(t_i) B_{i-1}^1(x)$ es un spline de primer grado que interpola f en cada nudo. ¿Cuál es el spline de grado cero que hace esto?

- 6.** Demuestre que si $t_m \leq x < t_{m+1}$, entonces

$$\sum_{i=-\infty}^{\infty} c_i B_i^k(x) = \sum_{i=m-k}^m c_i B_i^k(x)$$

- 7.** Sea $h_i = t_{i+1} - t_i$. Demuestre que si

$$S(x) = \sum_{i=-\infty}^{\infty} c_i B_i^2(x) \quad \text{y si} \quad c_{i-1} h_{i-1} + c_{i-2} h_i = y_i(h_i + h_{i-1})$$

para toda i , entonces $S(t_m) = y_m$ para toda m . *Sugerencia:* use el problema 3.

- 8.** Demuestre que los coeficientes C_i^{j-1} generados por la ecuación (7) satisfacen la condición $\min_i C_i^{j-1} \leq f(x) \leq \max_i C_i^{j-1}$.

- 9.** Para nudos igualmente espaciados, demuestre que $k(k+1)^{-1} B_i^k(x)$ se encuentra en el intervalo con puntos extremos $B_i^{k-1}(x)$ y $B_{i+1}^{k-1}(x)$.

- 10.** Demuestre que $B_i^k(x) = B_0^k(x - t_i)$ si los nudos son los enteros en la recta real ($t_i = i$).

- 11.** Demuestre que

$$\int_{-\infty}^{\infty} B_i^k(x) dx = \frac{t_{i+k+1} - t_i}{k+1}$$

- 12.** Demuestre que la clase de todas las funciones spline de grado m que tienen nudos x_0, x_1, \dots, x_n incluye la clase de polinomios de grado m .

- 13.** Establezca la ecuación (8) por inducción.

- 14.** ¿Cuáles splines B, B_i^k tienen un valor distinto de cero en el intervalo (t_n, t_m) ? Explique.

15. Demuestre que en $[t_i, t_{i+1}]$ tenemos

$$B_i^k(x) = \frac{(x - t_i)^k}{(t_{i+1} - t_i)(t_{i+2} - t_i) \cdots (t_{i+k} - t_i)}$$

16. ¿Un spline de la forma $S(x) = \sum_{i=-\infty}^{\infty} c_i B_i^k(x)$ está determinado únicamente por un conjunto finito de condiciones de interpolación $S(t_i) = y_i$ ($0 \leq i \leq n$)? ¿Por qué sí o por qué no?

17. Si la función spline $S(x) = \sum_{i=-\infty}^{\infty} c_i B_i^k(x)$ es igual a cero en cada nudo t , debe ser idénticamente cero? Por qué sí o por qué no.

18. ¿Cuál es la condición necesaria y suficiente en los coeficientes para que $\sum_{i=-\infty}^{\infty} c_i B_i^k = 0$? Establezca y pruebe.

19. Desarrolle la función $f(x) = x$ en una serie infinita $\sum_{i=-\infty}^{\infty} B_i^k$.

20. Establezca que $\sum_{i=-\infty}^{\infty} B_i^k$ es una función constante usando la ecuación (9).

21. Demuestre que si $k \geq 2$, entonces

$$\begin{aligned} \frac{d^2}{dx^2} \sum_{i=-\infty}^{\infty} c_i B_i^k &= k(k-1) \sum_{i=-\infty}^{\infty} \left[\frac{c_i - c_{i-1}}{(t_{i+k} - t_i)(t_{i+k-1} - t_i)} \right. \\ &\quad \left. - \frac{c_{i-1} - c_{i-2}}{(t_{i+k-1} - t_{i-1})(t_{i+k-1} - t_i)} \right] B_i^{k-2} \end{aligned}$$

22. Demuestre que si los nudos se toman como enteros, entonces

$$B_{-1}^1(x) = \max\{0, 1 - |x|\}.$$

23. Haciendo que los nudos sean enteros, demuestre que

$$B_0^2(x) = \begin{cases} 0 & (x < 0) \\ \frac{1}{2}x^2 & (0 \leq x < 1) \\ \frac{1}{2}(6x - 3 - 2x^2) & (1 \leq x < 2) \\ \frac{1}{2}(3 - x)^2 & (2 \leq x < 3) \\ 0 & (x \geq 3) \end{cases}$$

24. Deduzca las fórmulas

$$B_{i-1}^2(t_i) = \frac{t_i - t_{i-1}}{t_{i+1} - t_{i-1}} = \frac{h_{i-1}}{h_i + h_{i-1}}$$

$$B_{i-2}^2(t_i) = \frac{t_{i+1} - t_i}{t_{i+1} - t_{i-1}} = \frac{h_i}{h_i + h_{i-1}}$$

donde $h_i = t_{i+1} - t_i$.

25. Demuestre por inducción que si

$$A_j = \frac{1}{t_{j-1} - t_{j-2}} [y_{j-1}(t_j - t_{j-2}) - A_{j-1}(t_j - t_{j-1})]$$

para $j = 2, 3, \dots, n+1$, entonces

$$\sum_{i=0}^{n+1} A_i B_{i-2}^2(t_j) = y_j \quad (0 \leq j \leq n)$$

26. Demuestre que si $S(x) = \sum_{i=-\infty}^{\infty} A_i B_{i-2}^2(x)$ y $t_{j-1} \leq x \leq t_j$, entonces

$$S(x) = \frac{1}{t_j - t_{j-1}} [d(x - t_{j-1}) + e(t_j - x)]$$

con

$$d = \frac{1}{t_{j+1} - t_{j-1}} [A_{j+1}(x - t_{j-1}) + A_j(t_{j+1} - x)]$$

y

$$e = \frac{1}{t_j - t_{j-2}} [A_j(x - t_{j-2}) + A_{j-1}(t_j - x)]$$

27. Compruebe las ecuaciones (17) y (18) por inducción, usando la ecuación (16).

28. Si los puntos $\tau_0 < \tau_1 < \dots < \tau_n$ están dados, ¿podemos siempre determinar los puntos t_i tales que $t_i < t_{i+1}$ y $\tau_i = \frac{1}{2}(t_{i+1} + t_{i+2})$? Por qué si o por qué no.

29. Demuestre que si $f(x) = x$, entonces el proceso de Schoenberg produce $S(x) = x$.

30. Demuestre que $x^2 = \sum_{i=-\infty}^{\infty} t_{i+1} t_{i+2} B_i^2(x)$.

31. Sea $f(x) = x^2$. Suponga que $t_{i+1} - t_i \leq \delta$ para toda i . Demuestre que la aproximación de spline cuadrático a f dada por la ecuación (19) difiere de f por no más de $\delta^2/4$. *Sugerencia:* use el problema anterior y el hecho de que $\sum_{i=-\infty}^{\infty} B_i^2 \equiv 1$.

32. Compruebe (para $k > 0$) que $B_i^k(t_j) = 0$ si y sólo si $j \leq i$ o $j \geq i + k + 1$.

33. ¿Cuál es el máximo valor de B_i^2 y dónde ocurre?

34. Haga que los nudos sean enteros y demuestre que

$$B_0^3(x) = \begin{cases} 0 & (x < 0) \\ \frac{1}{6}x^3 & (0 \leq x < 1) \\ \frac{1}{6}(4 - 3x(x - 2)^2) & (1 \leq x < 2) \\ \frac{1}{6}(4 + 3(x - 4)(x - 2)^2) & (2 \leq x < 3) \\ \frac{1}{6}(4 - x)^3 & (3 \leq x < 4) \\ 0 & (x \geq 4) \end{cases}$$

35. En la teoría de las curvas de Bézier, usando los polinomios base de Bernstein, demuestre que la curva pasa por el primer punto v_0 .

36. Demuestre que un **spline lineal B** con nudos enteros se puede escribir en forma matricial como

$$S(x) = [x \quad 1] \begin{bmatrix} -1 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_0 \end{bmatrix} = b_{10}c_0 + b_{11}c_1$$

donde

$$B_0^1(x) = \begin{cases} b_{10} = x & (0 \leq x < 1) \\ b_{11} = 2 - x & (1 \leq x < 2) \\ 0 & (\text{de otra forma}) \end{cases}$$

- 37.** Demuestre que el **spline cuadrático B** con nudos enteros se puede escribir en forma matricial como

$$S(x) = \frac{1}{2}[x^2 \quad x \quad 1] \begin{bmatrix} 1 & -2 & 1 \\ -6 & 6 & 0 \\ 9 & -3 & 0 \end{bmatrix} \begin{bmatrix} c_2 \\ c_1 \\ c_0 \end{bmatrix} = b_{20}c_0 + b_{21}c_1 + b_{22}c_2$$

donde

$$B_0^2(x) = \begin{cases} b_{20} & (0 \leq x < 1) \\ b_{21} & (1 \leq x < 2) \\ b_{22} & (2 \leq x < 3) \\ 0 & (\text{de otra manera}) \end{cases}$$

Sugerencia: véase el problema 9.3.23.

- 38.** Demuestre que el **spline cúbico B** con nudos enteros se puede escribir como

$$\begin{aligned} S(x) &= \frac{1}{6}[x^3 \quad x^2 \quad x \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 12 & -24 & 12 & 0 \\ -48 & 60 & -12 & 0 \\ 64 & -44 & 4 & 0 \end{bmatrix} \begin{bmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} \\ &= b_{30}c_0 + b_{31}c_1 + b_{32}c_2 + b_{33}c_3 \end{aligned}$$

donde

$$B_0^3(x) = \begin{cases} b_{30} & (0 \leq x < 1) \\ b_{31} & (1 \leq x < 2) \\ b_{32} & (2 \leq x < 3) \\ b_{33} & (3 \leq x < 4) \\ 0 & (\text{de otra manera}) \end{cases}$$

Sugerencia: véase el problema 9.3.34.

Problemas de cómputo 9.3

- Usando un graficador automático, trace la gráfica de B_0^k para $k = 0, 1, 2, 3, 4$. Use nudos enteros $t_i = i$ en el intervalo $[0, 5]$.
- Sea $t_i = i$ (por lo que los nudos son los puntos enteros en la recta real). Imprima una tabla de 100 valores de la función $3B_7^1 + 6B_8^1 - 4B_9^1 + 2B_{10}^1$ en el intervalo $[6, 14]$. Usando un graficador, construya la gráfica de esta función en el intervalo dado.
- (Continuación) Repita para la función $3B_7^2 + 6B_8^2 - 4B_9^2 + 2B_{10}^2$.
- Suponiendo que $S(x) = \sum_{i=0}^n c_i B_i^k(x)$, escriba un procedimiento para evaluar $S'(x)$ en una x dada. La entrada es $n, k, x, t_0, \dots, t_{n+k+1}$ y c_0, c_1, \dots, c_n .
- Escriba un procedimiento para evaluar a $\int_a^b S(x) dx$, usando la suposición de que $S(x) = \sum_{i=0}^n c_i B_i^k(x)$. La entrada será $n, k, a, b, c_0, c_1, \dots, c_n, t_0, \dots, t_{n+k+1}$.

6. (**Paso de los splines B**) Producza gráficas de varios splines B del mismo grado que *pasan* a por el eje x . Use un graficador automático o un paquete de cómputo con capacidades gráficas en pantalla como Matlab.

7. Los historiadores han calculado el tamaño del ejército de Flanders como sigue:

Fecha	Sept. 1572	Dic. 1573	Mar. 1574	Ene. 1575	May 1576
Número	67 259	62 280	62 350	59 250	51 457
Feb. 1578	Sept. 1580	Oct. 1582	Abr. 1588	Nov. 1591	Mar. 1607
27 603	45 435	61 162	63 455	62 164	41 471

Ajuste la tabla con un spline B cuadrático y utilícelo para encontrar el tamaño promedio del ejército durante el periodo indicado. (El *promedio* se define con una integral.)

8. Reescriba los procedimientos *BSpline2_Coef* y *BSpline_Eval* para que el arreglo (h_i) no se use.
9. Reescriba los procedimientos *BSpline2_Coef* y *BSpline2_Eval* para el caso especial de nudos igualmente espaciados, simplificando el código donde sea posible.
10. Escriba un procedimiento para producir una aproximación de spline para $F(x) = \int_a^x f(t) dt$. Suponga que $a \leq x \leq b$. Empiece por encontrar un spline cuadrático de interpolación para f en los n puntos $t_i = a + i(b - a)/n$. Pruebe su programa con las funciones siguientes:
- $f(x) = \sin x$ $(0 \leq x \leq \pi)$
 - $f(x) = e^x$ $(0 \leq x \leq 4)$
 - $f(x) = (x^2 + 1)^{-1}$ $(0 \leq x \leq 2)$
11. Escriba un procedimiento para producir una función spline que aproxime a $f'(x)$ para una f dada en un intervalo $[a, b]$. Empiece por encontrar un spline cuadrático de interpolación a f en los $n + 1$ puntos espaciados uniformemente en $[a, b]$, incluidos los extremos. Pruebe su procedimiento con las funciones propuestas en el problema de cómputo anterior.
12. Defina f en $[0, 6]$ como una línea poligonal que une los puntos $(0, 0), (1, 2), (3, 3), (5, 3)$ y $(6, 0)$. Determine las aproximaciones de spline para f , usando el proceso de Schoenberg y tomando 7, 13, 19, 25 y 31 nudos.
13. Escriba un código adecuado para calcular $\sum_{i=-\infty}^{\infty} f(s_i) B_i^2(x)$ con $s_i = \frac{1}{3}(t_{i+1} + t_{i+2})$. Suponga que f está definida en $[a, b]$ y que x se encontrará en $[a, b]$. Suponga también que $t_1 < a < t_2$ y $t_{n+1} < b < t_{n+2}$. (No haga suposiciones acerca del espacio entre los nudos.)
14. Escriba un procedimiento para realizar este esquema de aproximación:

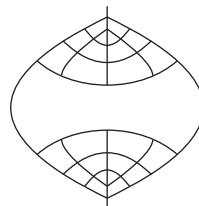
$$S(x) = \sum_{i=-\infty}^{\infty} f(\tau_i) B_i^3(x) \quad \tau_i = \frac{1}{3}(t_{i+1} + t_{i+2} + t_{i+3})$$

Suponga que f está definida en $[a, b]$ y que $\tau_i = a + ih$ para $0 \leq i \leq n$, donde $h = (b - a)/n$.

15. Usando un sistema de software matemático como Matlab con rutinas de spline B, calcule y trae la gráfica de la curva spline de la figura 9.16 con base en los 20 puntos de datos de la sección 9.2. Cambie el grado de los splines B de 0, 1, 2, 3 y por 4 y observe las curvas resultantes.

16. Usando los splines B, escriba un programa para desarrollar una interpolación de spline cúbico natural en los nudos $t_0 < t_1 < \dots < t_n$.

17. El sistema de preparación de documentos L^AT_EX está muy disponible y tiene facilidades para dibujar algunas curvas simples como las de Bézier. Use este sistema para reproducir la figura siguiente.



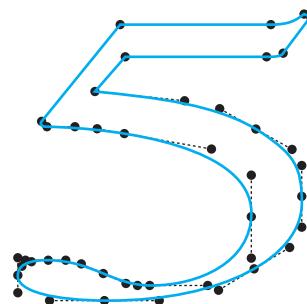
18. Use software matemático como el que se encuentra en Matlab, Maple o Mathematica para trazar las funciones correspondientes a la

a. Figura 9.17.

b. Figura 9.18.

c. Figura 9.19.

19. **(Diseño geométrico asistido por computadora)** Use software matemático para dibujar curvas de splines de Bézier bidimensionales y trace la gráfica del número cinco en letra de imprenta que se muestra, usando puntos de spline y puntos de control. Véase Farin [1990], Sauer [2006] y Yamaguchi [1988] para más detalles.



Ecuaciones diferenciales ordinarias

En un circuito eléctrico simple, la corriente \mathcal{I} en amperes es una función de tiempo: $\mathcal{I}(t)$. La función $\mathcal{I}(t)$ satisface una ecuación diferencial ordinaria de la forma

$$\frac{d\mathcal{I}}{dt} = f(t, \mathcal{I})$$

En este caso, el miembro derecho es una función de t e \mathcal{I} que depende del circuito y de la naturaleza de la fuerza electromotriz suministrada al circuito. Usando los métodos desarrollados en este capítulo, podemos resolver la ecuación diferencial numéricamente para obtener una tabla de \mathcal{I} como una función de t .

10.1 Métodos de series de Taylor

Primero presentamos un análisis general de las ecuaciones diferenciales ordinarias y sus soluciones.

Problema con valor inicial: solución analítica contra numérica

Una **ecuación diferencial ordinaria** (EDO) es una ecuación que implica una o más derivadas de una función desconocida. Una **solución** de una ecuación diferencial es una función específica que satisface la ecuación. Aquí se presentan algunos ejemplos de ecuaciones diferenciales con sus soluciones. En cada caso, t es la variable independiente y x es la variable dependiente. Por tanto, x es el nombre de la función desconocida de la variable independiente t :

Ecuación	Solución
$x' - x = e^t$	$x(t) = te^t + ce^t$
$x'' + 9x = e^t$	$x(t) = c_1 \operatorname{sen} 3t + c_2 \cos 3t$
$x' + \frac{1}{2x} = 0$	$x(t) = \sqrt{c - t}$

En estos tres ejemplos, la letra c denota una constante arbitraria. El hecho de que esas constantes se presenten en las soluciones es una indicación de que, en general, una ecuación diferencial, no determina una función solución única. Cuando en un problema científico se presenta, una ecuación diferencial generalmente se acompaña de condiciones auxiliares que (junto con la ecuación diferencial) determinan la función desconocida exactamente.

En este capítulo nos dedicamos a un tipo de ecuación diferencial y a un tipo de condición auxiliar: el **problema con valor inicial** para una ecuación diferencial de primer orden. La forma estándar que se ha adoptado es

$$\begin{cases} x' = f(t, x) \\ x(a) \text{ está dada} \end{cases} \quad (1)$$

Se sobrentiende que x es una función de t , por lo que la ecuación diferencial escrita con más detalle está dada por:

$$\frac{dx(t)}{dt} = f(t, x(t))$$

El problema (1) se llama un problema con valor inicial porque t se puede interpretar como el tiempo y $t = a$ se puede pensar como el instante inicial en el tiempo. Queremos poder determinar el valor de x en cualquier tiempo t antes o después de a .

Aquí se presentan algunos ejemplos de problemas con valores iniciales, junto con sus soluciones:

Ecuación	Valor inicial	Solución
$x' = x + 1$	$x(0) = 0$	$x = e^t - 1$
$x' = 6t - 1$	$x(1) = 6$	$x = 3t^2 - t + 4$
$x' = \frac{t}{x+1}$	$x(0) = 0$	$x = \sqrt{t^2 + 1} - 1$

Aunque existen muchos métodos para obtener soluciones analíticas de ecuaciones diferenciales, están limitados principalmente a ecuaciones diferenciales especiales. Cuando son aplicables, se obtiene una solución en forma de una fórmula, como se muestra en los ejemplos anteriores. Sin embargo, con frecuencia, en problemas prácticos, una ecuación diferencial no tiene solución por métodos especiales y se debe buscar una solución numérica. Incluso cuando se puede obtener una solución formal puede ser preferible una solución numérica, especialmente si la solución formal es muy complicada. Una solución numérica de una ecuación diferencial se obtiene generalmente en forma de una tabla, la forma funcional de la solución no se conoce al grado de contar con una fórmula específica.

La forma que adopte la ecuación diferencial en este caso permite que la función f dependa de t y de x . Si f no implica a x , como en el segundo ejemplo anterior, entonces la ecuación diferencial se puede resolver con un proceso directo de integración indefinida. Como ejemplo, considere el problema con valor inicial

$$\begin{cases} x' = 3t^2 - 4t^{-1} + (1+t^2)^{-1} \\ x(5) = 17 \end{cases} \quad (2)$$

La ecuación diferencial se puede integrar, con lo que se obtiene

$$x(t) = t^3 - 4 \ln t + \arctan t + C$$

La constante C entonces se puede elegir de manera que $x(5) = 17$. Podemos usar un sistema de software matemático como Maple o Mathematica para resolver esta ecuación diferencial explícitamente y en consecuencia encontrar el valor de esta constante como $C = 4 \ln(5) - \arctan(5) - 108$.

Con frecuencia queremos una solución numérica para una ecuación diferencial porque (a) la solución de *forma cerrada* puede ser muy complicada y difícil de evaluar o (b) no hay otra opción; es decir, no se puede encontrar una solución de *forma cerrada*. Considere, por ejemplo, la ecuación diferencial

$$x' = e^{-\sqrt{t^2 - \sin t}} + \ln |\sin t + \tanh t^3| \quad (3)$$

La solución se obtiene integrando o tomando la antiderivada del miembro derecho. Esto se puede hacer en principio pero no es en la práctica. En otras palabras, existe una función x para la cual dx/dt es el miembro derecho de la ecuación (3), pero no es posible escribir $x(t)$ en términos de funciones conocidas.

Resolver ecuaciones diferenciales ordinarias en una computadora puede requerir un gran número de pasos pequeños, por lo que se puede acumular una cantidad significativa de errores de redondeo. En consecuencia, puede ser necesario hacer cálculos de precisión múltiple en computadoras con longitud de palabra pequeña.

Ejemplo de un problema práctico

Muchos problemas prácticos de dinámica implican las tres **leyes de movimiento de Newton**, en particular la segunda ley, que se enuncia simbólicamente como $F = ma$, donde F es la fuerza que actúa sobre un cuerpo de masa m y a es la aceleración resultante de este cuerpo. Esta ley es una ecuación diferencial disfrazada porque la aceleración es la derivada de la velocidad y la velocidad es, a su vez, la derivada de la posición. Ejemplificamos con un modelo simplificado de un cohete que se lanzó al tiempo $t = 0$. Su movimiento es vertical hacia arriba y medimos su altura con la variable x . La fuerza de propulsión tiene un valor constante, a saber, 5370. (Las unidades se eligen coherentes entre sí.) Hay fuerza negativa debido a la resistencia del aire, cuya magnitud es $v^{3/2}/\ln(2 + v)$, donde v es la velocidad del cohete. La masa está disminuyendo a un ritmo constante debido a la quema de combustible y se toma de $321 - 24t$. La variable independiente es el tiempo, t . El combustible se consume totalmente al tiempo $t = 10$. Hay una fuerza hacia abajo debido a la gravedad, de magnitud 981. Reuniendo todos estos términos en la ecuación $F = ma$, tenemos

$$5370 - 981 - v^{3/2} / \ln(2 + v) = (321 - 24t)v' \quad (4)$$

La condición inicial es $v = 0$ en $t = 0$.

En las secciones siguientes desarrollaremos métodos para resolver ecuaciones diferenciales. Por otra parte, también se puede utilizar un sistema de software matemático para resolver este problema.

Un código de computadora para resolver ecuaciones diferenciales ordinarias produce una tabla con valores discretos, mientras que la solución matemática es una función continua. Se pueden necesitar valores adicionales dentro de un intervalo para diversos fines, como el trazo de gráficas. Se pueden utilizar procedimientos de interpolación para obtener todos los valores de la solución aproximada numéricamente dentro de un intervalo dado. Por ejemplo, un sistema de interpolación por partes con un polinomio puede producir una solución numérica que es continua y tiene una primera derivada continua que concuerda con la derivada de la solución. Al usar cualquier solucionador de EDO, existe una aproximación a $x'(t)$ en el hecho de que $x'(t) = f(t, x)$. Los paquetes matemáticos para resolver EDO pueden incluir capacidades de trazado automático, ya que la mejor manera de dar sentido a la gran cantidad de datos que se pueden obtener como la *solución* es presentar las curvas solución en una pantalla gráfica o trazadas en papel.

Resolución de ecuaciones diferenciales e integración

Existe una conexión cercana entre resolver ecuaciones diferenciales e integrar. Considere la ecuación diferencial

$$\begin{cases} \frac{dx}{dr} = f(r, x) \\ x(a) = s \end{cases}$$

Integrando de t a $t + h$, tenemos

$$\int_t^{t+h} dx = \int_t^{t+h} f(r, x(r)) dr$$

Por tanto,

$$x(t + h) = x(t) + \int_t^{t+h} f(r, x(r)) dr$$

Sustituyendo la integral por una de las reglas de integración numérica del capítulo 5 se obtiene una fórmula para resolver la ecuación diferencial. Por ejemplo, el método de Euler, ecuación (6) (véase la pág. 432), se obtiene de la aproximación de rectángulo izquierdo (véase el problema 5.2.28):

$$\int_t^{t+h} f(r, x(r)) dr \approx hf(t, x(t))$$

La regla del trapecio

$$\int_t^{t+h} f(r, x(r)) dr \approx \frac{h}{2}[f(t, x(t)) + f(t + h, x(t + h))]$$

da la fórmula

$$x(t + h) = x(t) + \frac{h}{2}[f(t, x(t)) + f(t + h, x(t + h))]$$

Puesto que $x(t + h)$ se presenta en ambos miembros de esta ecuación, se llama **fórmula implícita**. Si el método de Euler

$$x(t + h) = x(t) + hf(t, x(t))$$

se usa para el $x(t + h)$ del miembro derecho, entonces obtenemos la fórmula de Runge-Kutta de orden 2, a saber, la ecuación (10) de la sección 10.2.

Usando el teorema fundamental del cálculo podemos fácilmente demostrar que un valor numérico aproximado para la integral

$$\int_a^b f(r, x(r)) dr$$

se puede calcular resolviendo el siguiente problema con valor inicial para $x(b)$:

$$\begin{cases} \frac{dx}{dr} = f(r, x) \\ x(a) = 0 \end{cases}$$

Campos vectoriales

Considere una ecuación diferencial general de primer orden con condición inicial dada:

$$\begin{cases} x'(t) = f(t, x(t)) \\ x(a) = b \end{cases}$$

Antes de resolver numéricamente este problema con valor inicial es útil pensar el significado intuitivo de la ecuación. La función f proporciona la pendiente de la función solución en el plano tx . En cada punto donde $f(t, x)$ está definida, podemos imaginar una recta corta que pasa por ese punto y que tiene la pendiente dada. No podemos trazar la gráfica de *todas* las rectas cortas, pero podemos dibujar tantas como queramos, con la esperanza de entender cómo la función solución $x(t)$ traza

su camino a través de este bosque de segmentos de recta, manteniendo su pendiente en cada punto igual a la pendiente del segmento de recta trazado en ese punto. El diagrama de segmentos de recta ilustra discretamente el llamado **campo vectorial** de la ecuación diferencial.

Por ejemplo, consideremos la ecuación

$$x' = \operatorname{sen}(x + t^2)$$

con valor inicial $x(0) = 0$. En el rectángulo descrito por las desigualdades $-4 \leq x \leq 4$ y $-4 \leq t \leq 4$, podemos indicar al software matemático como Matlab que, proporcione una imagen del campo vectorial generado por nuestra ecuación diferencial. Usando instrucciones en el entorno de ventanas, nos abrirá una ventana con la ecuación diferencial mostrada en un rectángulo. Entonces detrás de las figuras, el software matemático realizará los cálculos inmensos para presentar el campo vectorial de esta ecuación diferencial y lo presentará correctamente etiquetado. Para ver qué solución pasa a través de cualquier punto en el diagrama, sólo es necesario utilizar el ratón para colocar el puntero sobre ese punto y haciendo clic en el botón izquierdo del ratón, el software mostrará la solución buscada. Usando este software de herramientas se puede ver inmediatamente el efecto de cambiar las condiciones iniciales. En la figura 10.1 se muestran varias curvas solución para el problema que estamos considerando (correspondientes a diferentes valores iniciales).

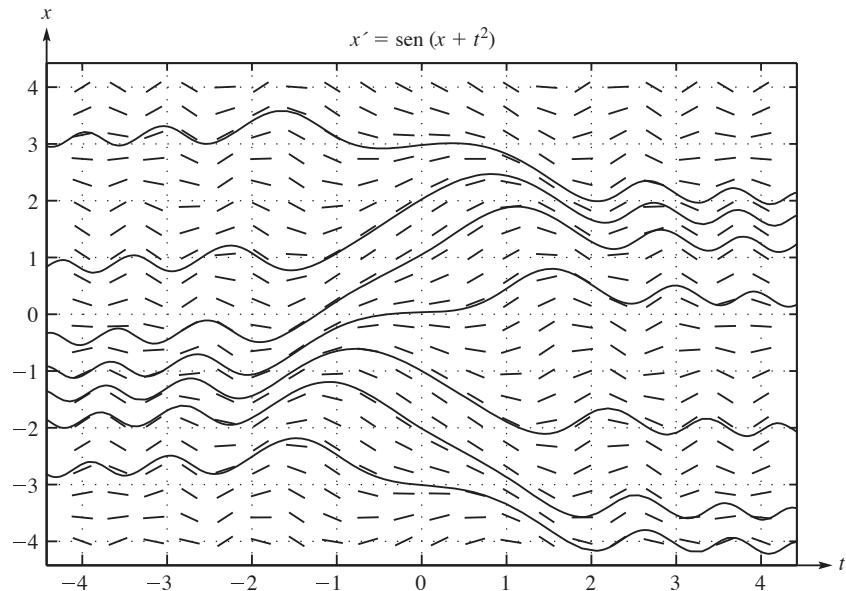


FIGURA 10.1
Campo
vectorial y
algunas curvas
solución para
 $x' = \operatorname{sen}(x + t^2)$

Otro ejemplo, tratado de la misma forma, es la ecuación diferencial

$$x' = x^2 - t$$

La figura 10.2 muestra un campo vectorial de esta ecuación y algunas de sus soluciones. Observe el fenómeno de que se trata de muchas curvas muy diferentes y todas parecen surgir de la misma condición inicial. ¿Qué está pasando en este caso? Este es un ejemplo extremo de una ecuación diferencial cuyas soluciones son sumamente sensibles a la condición inicial! Se pueden esperar problemas en la solución de esta ecuación diferencial con un valor inicial dado en $t = -2$.

¿Cómo sabemos que la ecuación diferencial $x' = x^2 - t$, junto con un valor inicial, $x(t_0) = x_0$, tiene una solución única? Hay muchos teoremas de ecuaciones diferenciales que tratan los temas de existencia y unicidad. Uno de los más fáciles de usar es el siguiente.

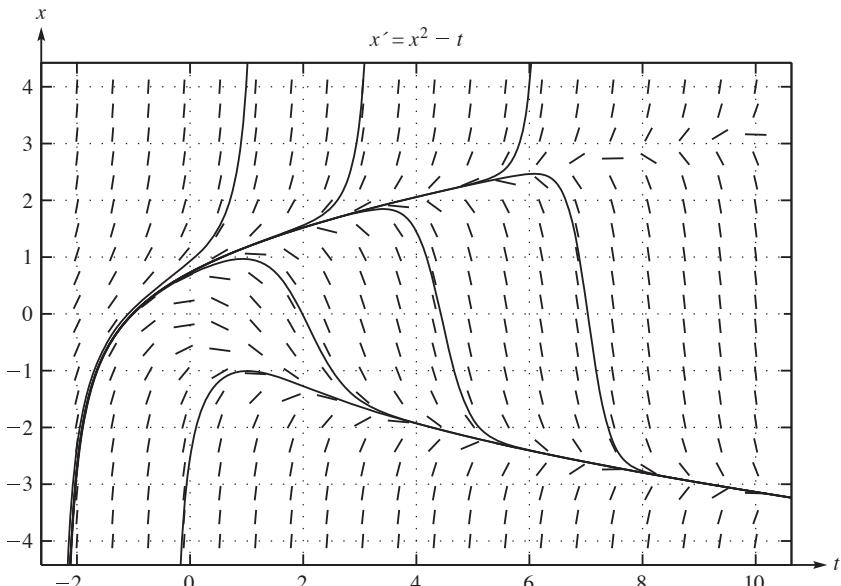


FIGURA 10.2
Campo vectorial y algunas curvas solución para $x' = x^2 - t$

■ TEOREMA 1

Unicidad del problema con valores iniciales

Si f y $\partial f / \partial y$ son continuas en el rectángulo definido por $|t - t_0| < \alpha$ y $|x - x_0| < \beta$, entonces el problema con valor inicial $x' = f(t, x)$, $x(t_0) = x_0$ tiene una única solución continua en algún intervalo $|t - t_0| < \epsilon$.

Del teorema que acabamos de citar, no podemos concluir que la solución en cuestión se defina por $|t - t_0| < \beta$. Sin embargo, el valor de ϵ en el teorema es, al menos, β/M , donde M es un límite superior para $|f(t, x)|$ en el rectángulo original.

Métodos de series de Taylor

El método numéricico descrito en esta sección no tiene la mayor generalidad, pero es natural y capaz de tener alta precisión. Su principio es representar la solución de una ecuación diferencial localmente con pocos términos de su serie de Taylor.

En lo que sigue, supondremos que nuestra función solución x está representada por su serie de Taylor*

$$\begin{aligned} x(t+h) = & x(t) + h x'(t) + \frac{1}{2!} h^2 x''(t) + \frac{1}{3!} h^3 x'''(t) \\ & + \frac{1}{4!} h^4 x^{(4)}(t) + \cdots + \frac{1}{m!} h^m x^{(m)}(t) + \cdots \end{aligned} \quad (5)$$

* Recuerde que algunas funciones tales como e^{-1/x^2} son suaves, pero *no* se representan con una serie de Taylor en 0.

Por razones numéricas, la serie de Taylor truncada después de $m + 1$ términos nos permite calcular $x(t + h)$ con bastante precisión si h es pequeño y si $x(t), x'(t), x''(t), \dots, x^{(m)}(t)$ son conocidos. Cuando sólo se incluyen términos $h^m x^{(m)}(t)/m!$ en la serie de Taylor, el método que resulta se llama **método de la serie de Taylor de orden m** . Empezamos con el caso $m = 1$.

Seudocódigo del método de Euler

El método de la serie de Taylor de orden 1 se conoce como el **método de Euler**. Para encontrar valores aproximados de las soluciones al problema con valor inicial

$$\begin{cases} x' = f(t, x(t)) \\ x(a) = x_a \end{cases}$$

en el intervalo $[a, b]$, se utilizan los primeros dos términos de la serie de Taylor (5):

$$x(t + h) \approx x(t) + h x'(t)$$

Por lo tanto, la fórmula

$$x(t + h) = x(t) + h f(t, x(t)) \quad (6)$$

se puede utilizar para el paso de $t = a$ a $t = b$, con n pasos de tamaño $h = (b - a)/n$. El seudocódigo para el método de Euler se puede escribir como sigue, donde se utilizan algunos valores dados para n, a, b y x_a :

```
program Euler
integer k; real h, t; integer n ← 100
external function f
real a ← 1, b ← 2, x ← -4
h ← (b - a)/n
t ← a
output 0, t, x
for k = 1 to n do
    x ← x + hf(t, x)
    t ← t + h
    output k, t, x
end for
end program Euler
```

Para utilizar este programa se necesita un código para $f(t, x)$, como se muestra en el ejemplo 1.

EJEMPLO 1 Usando el método de Euler, calcule un valor aproximado para $x(2)$ para la ecuación diferencial $x' = 1 + x^2 + t^3$ con el valor inicial $x(1) = -4$ usando 100 pasos.

Solución Utilizando el seudocódigo anterior con el valor inicial dado y combinando con la función siguiente:

```
real function f(t, x)
real t, x
f ← 1 + x2 + t3
end function
```

El valor calculado es $x(2) \approx 4.23585$.

Podemos escribir un programa de cómputo para ejecutar el método de Euler en este sencillo problema:

$$\begin{cases} x'(t) = x \\ x(0) = 1 \end{cases}$$

Obtenemos los resultados $x(2) \approx 7.3891$. En la figura 10.3 se muestra la gráfica obtenida con el código. La solución, $x(t) = e^t$, es la curva sólida y los puntos obtenidos con el método de Euler se muestran con puntos. ¿Puede entender por qué los puntos siempre están debajo de la curva?

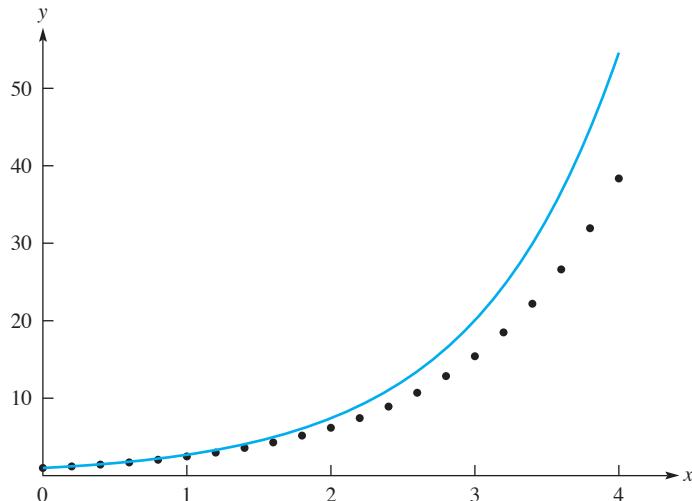


FIGURA 10.3
Curvas del
método de
Euler

Antes de aceptar estos resultados y continuar, hay que plantearse algunas preguntas tales como: *¿cuán exactas son las respuestas? ¿Son necesarios los métodos de la serie de Taylor de orden superior?* Lamentablemente, el método de Euler no es muy preciso porque sólo se utilizan dos términos de la serie de Taylor (5); por lo tanto, el error de truncamiento es $\mathcal{O}(h^2)$.

Método de la serie de Taylor de orden superior

El ejemplo 1 se puede utilizar para explicar el método de la serie de Taylor de orden superior. Consideremos de nuevo el problema con valor inicial

$$\begin{cases} x' = 1 + x^2 + t^3 \\ x(1) = -4 \end{cases}$$

Si las funciones en la ecuación diferencial se derivan varias veces con respecto a t , los resultados son los siguientes (recuerde que una función de x se debe derivar con respecto a t usando la *regla de la cadena*)

$$\begin{aligned} x' &= 1 + x^2 + t^3 \\ x'' &= 2xx' + 3t^2 \\ x''' &= 2xx'' + 2x'x' + 6t \\ x^{(4)} &= 2xx''' + 6x'x'' + 6 \end{aligned}$$

Si los valores numéricos de t y de $x(t)$ se conocen, estas cuatro fórmulas, aplicadas en orden, producen $x'(t)$, $x''(t)$, $x'''(t)$ y $x^{(4)}(t)$. Así, a partir de este trabajo es posible usar los primeros cinco términos en la serie de Taylor, ecuación (5). Puesto que $x(1) = -4$, tenemos un punto inicial adecuado y seleccionamos $n = 100$, que determina h . A continuación, podemos calcular una aproximación a $x(a + h)$ a partir de las fórmulas (5) y (8). El mismo proceso se puede repetir para calcular $x(a + 2h)$ usando $x(a + h)$, $x'(a + h)$, ..., $x^{(4)}(a + h)$. Aquí se presenta el seudocódigo:

```

program Taylor
integer k; real h, t, x, x', x'', x'''; x(4)
integer n ← 100
real a ← 1, b ← 2, x ← -4
h ← (b - a)/n
t ← a
output 0, t, x
for k = 1 to n do
    x' ← 1 + x2 + t3
    x'' ← 2xx' + 3t2
    x''' ← 2xx'' + 2(x)2 + 6t
    x(4) ← 2xx''' + 6x'x'' + 6
    x ← x + h [x' +  $\frac{1}{2}$ h [x'' +  $\frac{1}{3}$ h [x''' +  $\frac{1}{4}$ h [x(4)]]]]
    t ← a + kh
    output k, t, x
end for
end program Taylor

```

Unas cuantas palabras de explicación pueden ser útiles en este caso. Antes de escribir el seudocódigo, determinamos el intervalo en el que desea calcular la solución de la ecuación diferencial. En el ejemplo, este intervalo es elegido como $a = 1 \leq t \leq 2 = b$ y se utilizan 100 pasos. En cada paso, el valor actual de t es un múltiplo entero de tamaño de paso h . Los enunciados de asignación que definen x' , x'' , x''' y $x^{(4)}$ son simplemente para realizar los cálculos de las derivadas de acuerdo con la ecuación (8). El cálculo final realiza la evaluación de la serie de Taylor en la ecuación (5) usando cinco términos. Puesto que esta ecuación es un polinomio en h , se evalúa de manera más eficaz con la multiplicación anidada, lo que explica la fórmula para x en el seudocódigo. El cálculo de $t \leftarrow t + h$ puede producir una pequeña cantidad de error de redondeo para acumularse en el valor de t . Esto se evita usando $t \leftarrow a + kh$.

Como era de esperarse, los resultados de usar sólo dos términos en la serie de Taylor (método de Euler) no son tan exactos como cuando se utilizan cinco términos:

Método de Euler
 $x(2) \approx 4.23585\,41$

Método de la serie de Taylor(orden 4)
 $x(2) \approx 4.37120\,96$

Mediante un análisis adicional, se puede probar que el valor correcto con más cifras significativas es $x(2) \approx 4.37122\,1866$. En este caso, los cálculos se realizaron con mayor precisión sólo para demostrar que la falta de precisión no fue un factor contribuyente.

Tipos de errores

Cuando se programa el seudocódigo descrito anteriormente y se ejecuta en un equipo, ¿qué clase de precisión podemos esperar? Son exactos todos los dígitos impresos por la máquina para la variable x ? ¡Por supuesto que no! Por otra parte, no es fácil decir cuántos dígitos *son* confiables. Esta es una evaluación burda. Puesto que los términos hasta $\frac{1}{24}h^4x^{(4)}(t)$ están incluidos, el primer término que *no* está incluido en la serie de Taylor es $\frac{1}{120}h^5x^{(5)}(t)$. El error puede ser mayor que este, pero el factor $h^5 = (10^{-2})^5 \approx 10^{-10}$ sólo afecta la décima posición decimal. La solución impresa es tal vez de ocho cifras decimales. Los puentes o aviones no se deben construir en tales análisis de mala calidad, pero por el momento, nuestra atención se centra en la forma general del procedimiento.

Realmente, hay que considerar dos tipos de errores. En cada paso, si se conoce $x(t)$ y se calcula $x(t+h)$ de algunos de los primeros términos de la serie de Taylor, ocurre un error porque hemos truncado la serie de Taylor. Entonces, este error se llama el **error de truncamiento** o, para ser más precisos, **error de truncamiento local**. En el ejemplo anterior, este es burdamente $\frac{1}{120}h^5x^{(5)}(\xi)$. En esta situación, digamos que el error de truncamiento local es del *orden* h^5 , que se abrevia con $\mathcal{O}(h^5)$.

El segundo tipo de error presente obviamente se debe a los efectos de acumulación de todos los errores locales de truncamiento. En realidad, el valor calculado de $x(t+h)$ tiene error porque $x(t)$ ya está equivocado (debido a los errores de truncamiento anteriores) y porque se produce otro error de truncamiento local en el cálculo de $x(t+h)$ por medio de la serie de Taylor.

Se deben considerar otras fuentes de error en una teoría completa. Una de ellas es el **error de redondeo**. Aunque no es grave en cualquier paso del procedimiento de solución, después de cientos o miles de pasos es posible que se acumulen y contaminen fuertemente la solución calculada. Recuerde que un error que se hace en un determinado paso se realiza hacia adelante en todos los pasos sucesivos. Dependiendo de la ecuación diferencial y del método que se utiliza para resolvérila, este tipo de errores se pueden amplificar por pasos sucesivos.

Método de la serie de Taylor usando cálculos simbólicos

Varias rutinas de cálculos matemáticos, tanto de carácter no numérico como de tipo numérico, que incluyen derivación e integración de las expresiones, incluso de las más complicadas, ahora se pueden dejar a la computadora. Por supuesto, esto sólo se aplica a una clase de funciones, pero esta clase es lo suficientemente amplia como para incluir todas las funciones que uno encuentra en un típico libro de texto de cálculo. Con el uso de estos programas simbólicos de cálculo, el método de la serie de Taylor de orden superior se puede realizar sin dificultad. Usando las potencialidades de manejo algebraico del software matemático como Maple o Mathematica, podemos escribir el código para resolver el problema con valor inicial (7). El resultado final es $x(2) \approx 4.37121\ 00522\ 49692\ 27234\ 569$.

Resumen

(1) Deseamos resolver el problema con valor inicial de primer orden

$$\begin{cases} x'(t) = f(t, x(t)) \\ x(a) = x_a \end{cases}$$

sobre el intervalo $[a, b]$ con tamaño de paso $h = (b - a)/n$. El **método de la serie de Taylor de orden m** es

$$\begin{aligned}x(t + h) &= x(t) + hx'(t) + \frac{1}{2!}h^2x''(t) + \frac{1}{3!}h^3x'''(t) \\&\quad + \frac{1}{4!}h^4x^{(4)}(t) + \cdots + \frac{1}{m!}h^mx^{(m)}(t)\end{aligned}$$

donde todas las derivadas $x'', x''', \dots, x^{(m)}$ se han determinado analíticamente.

(2) El **método de Euler** es el método de la serie de Taylor de orden 1 y se puede escribir como

$$x(t + h) = x(t) + hf(t, x(t))$$

Debido a que sólo se utilizan dos términos en la serie de Taylor, el error de truncamiento es grande y los resultados no se pueden calcular con mucha precisión. Por consiguiente, los métodos de la serie de Taylor de orden superior se utilizan con más frecuencia. Por supuesto, requieren que se determinen más derivadas, con más posibilidades de errores matemáticos.

Problemas 10.1

1. Dé las soluciones de estas ecuaciones diferenciales:

- | | |
|--|---|
| ^a a. $x' = t^3 + 7t^2 - t^{1/2}$ | ^a b. $x' = x$ |
| c. $x' = -x$ | d. $x'' = -x$ |
| ^a e. $x'' = x$ | f. $x'' + x' - 2x = 0$ Sugerencia: pruebe con $x = e^{at}$. |

^a2. Dé las soluciones de estos problemas con valores iniciales:

- | | |
|--|---------------------------------|
| ^a a. $x' = t^2 + t^{1/3}$ $x(0) = 7$ | b. $x' = 2x$ $x(0) = 15$ |
| c. $x'' = -x$ $x(\pi) = 0$ $x'(\pi) = 3$ | |

3. Resuelva las siguientes ecuaciones diferenciales:

- | |
|---|
| a. $x' = 1 + x^2$ Sugerencia: $1 + \tan^2 t = \sec^2 t$ |
| b. $x' = \sqrt{1 - x^2}$ Sugerencia: $\sin^2 t + \cos^2 t = 1$ |
| ^a c. $x' = t^{-1} \operatorname{sen} t$ Sugerencia: véase el problema de cómputo 5.1.2. |
| ^a d. $x' + tx = t^2$ Sugerencia: multiplique la ecuación por $f(t) = \exp(t^2/2)$. El miembro izquierdo será $(xf)'$ |

^a4. Resuelva el problema 3b sustituyendo una serie de potencias $x(t) = \sum_{n=0}^{\infty} a_n t^n$ y después determine los valores adecuados de los coeficientes.

5. Determine x'' cuando $x' = xt^2 + x^3 + e^x t$.

^a6. Encuentre un polinomio p con la propiedad $p - p' = t^3 + t^2 - 2t$.

7. La ecuación diferencial lineal de primer orden general es $x' + px + q = 0$, donde p y q son funciones de t . Muestre que la solución es $x = -y^{-1}(z + c)$, donde y y z son funciones que se obtienen de la forma siguiente. Sea u una antiderivada de p . Haga $y = e^u$, y sea z una antiderivada de yq .

8. Aquí se presenta un problema con valor inicial que tiene dos soluciones: $x' = x^{1/3}$, $x(0) = 0$. Compruebe que las dos soluciones son $x_1(t) = 0$ y $x_2(t) = \left(\frac{2}{3}t\right)^{3/2}$ para $t \geq 0$. Si se aplica el método de la serie de Taylor, ¿qué pasa?

9. Considere el problema $x' = x$. Si la condición inicial es $x(0) = c$, entonces la solución es $x(t) = ce^t$. Si se produce un error de redondeo ε en la lectura del valor de c en la computadora, ¿qué efecto existe en la solución en el punto $t = 10$? ¿En $t = 20$? Haga lo mismo para $x' = -x$.

10. Si la serie de Taylor es el método utilizado en el problema con valor inicial $x' = t^2 + x^3$, $x(0) = 0$ y si tenemos la intención de utilizar las derivadas de x hasta $x^{(4)}$ inclusive, ¿cuáles son las cinco ecuaciones principales que se deben programar?

11. En la resolución de las siguientes ecuaciones diferenciales con el método de la serie de Taylor de orden n , ¿cuáles son las ecuaciones principales en el algoritmo?

a. $x' = x + e^x \quad n = 4$ **b.** $x' = x^2 - \cos x \quad n = 5$

12. Calcule un valor aproximado de $x(0.1)$ usando un paso del método de la serie de Taylor de orden 3 en la ecuación diferencial ordinaria

$$\begin{cases} x'' = x^2 e^t + x' \\ x(0) = 1 \quad x'(0) = 2 \end{cases}$$

13. Supongamos que una ecuación diferencial se resuelve numéricamente en un intervalo $[a, b]$ y que el error de truncamiento local es de ch^p . Demuestre que si todos los errores de truncamiento tienen el mismo signo (el peor caso posible), entonces el error de truncamiento total es $(b - a) ch^{p-1}$, donde $h = (b - a)/n$.

14. Si va a utilizar el método de la serie de Taylor con términos hasta h^{20} , ¿cómo debería realizarse el cálculo de $\sum_{n=0}^{20} x^{(n)}(t) h^n / n!$? Suponga que $x(t)$, $x^{(1)}(t)$, $x^{(2)}(t)$, ..., y $x^{(20)}(t)$ están disponibles. Sugerencia: con algunas declaraciones es suficiente.

15. Explique cómo utilizar el método de las EDO, que se basa en la regla del trapecio:

$$\begin{aligned} \hat{x}(t+h) &= x(t) + hf(t, x(t)) \\ x(t+h) &= x(t) + \frac{h}{2}[f(t, x(t)) + f(t+h, \hat{x}(t+h))] \end{aligned}$$

Este se llama **método de Euler mejorado** o **método de Heun**. En este caso, $\hat{x}(t+h)$ se calcula usando el método de Euler.

16. (Continuación) Use el método de Euler mejorado para resolver la siguiente ecuación diferencial en el intervalo $[0, 1]$ con tamaño de paso $h = 0.1$:

$$\begin{cases} x' = -x + t + \frac{1}{2} \\ x(0) = 1 \end{cases}$$

17. Considere el problema con valor inicial

$$\begin{cases} x' = -100x^2 \\ x(0) = 1 \end{cases}$$

En el método de Euler mejorado, sustituimos $\hat{x}(t+h)$ con $x(t+h)$ y tratamos de resolver con un tamaño de paso $h = 0.1$. Explique lo que sucede. Encuentre la solución en forma cerrada mediante la sustitución de $x = (a + bt)^c$ y determine a, b, c .

Problemas de cómputo 10.1

- ^a1. Escriba y pruebe un programa para aplicar el método de la serie de Taylor al problema con valor inicial

$$\begin{cases} x' = x + x^2 \\ x(1) = \frac{e}{16 - e} = 0.20466\ 34172\ 89155\ 26943 \end{cases}$$

Para generar la solución en el intervalo $[1, 2.77]$. Use derivadas hasta $x^{(5)}$ en la serie de Taylor. Use $h = 1/100$. Imprima por comparar los valores de la solución exacta $x(t) = e^t/(16 - e^t)$. Compruebe que esta es la solución exacta.

2. Escriba un programa para resolver cada problema en los intervalos indicados. Use el método de la serie de Taylor con $h = 1/100$ e incluya términos hasta h^3 . Explique cualquier dificultad.

^ab. $\begin{cases} x' = t + x^2 & \text{en } [0, 0.9] \\ x(0) = 1 \end{cases}$

^ac. $\begin{cases} x' = x - t & \text{en } [1, 1.75] \\ x(1) = 1 \end{cases}$

^ac. $\begin{cases} x' = tx + t^2x^2 & \text{en } [2, 5] \\ x(2) = -0.63966\ 25333 \end{cases}$

- ^a3. Resuelva la ecuación diferencial $x' = x$ con valor inicial $x(0) = 1$ con el método de la serie de Taylor en el intervalo $[0, 10]$. Compare el resultado con la solución exacta $x(t) = e^t$. Use derivadas hasta el décimo inclusive. Use tamaño de paso $h = 1/100$.

4. Resuelva para $x(1)$:

^aa. $x' = 1 + x^2, \quad x(0) = 0 \quad \text{b. } x' = (1 + t)^{-1}x, \quad x(0) = 1$

Use el método de la serie de Taylor de orden 5 con $h = 1/100$ y compare con las soluciones exactas, que son $\tan t$ y $1 + t$, respectivamente.

- ^a5. Resuelva el problema con valor inicial $x' = t + x + x^2$ en el intervalo $[0, 1]$ con condición inicial $x(1) = 1$. Use el método de la serie de Taylor de orden 5.

6. Resuelva el problema con valor inicial $x' = (x + t)^2$ con $x(0) = -1$ en el intervalo $[0, 1]$ usando el método de la serie de Taylor con derivadas hasta la cuarta inclusive. Compare esto con los métodos de la serie de Taylor de órdenes 1, 2 y 3.

- ^a7. Escriba un programa para resolver en el intervalo $[0, 1]$ el problema con valor inicial

$$\begin{cases} x' = tx \\ x(0) = 1 \end{cases}$$

usando el método de la serie de Taylor de orden 20; es decir, incluya términos en la serie de Taylor hasta h^{20} inclusive. Observe que una simple fórmula recursiva se puede usar para obtener $x^{(n)}$ para $n = 1, 2, \dots, 20$.

8. Escriba un programa para resolver el problema con valor inicial $x' = \sin x + \cos t$, usando el método de la serie de Taylor. Continúe la solución de $t = 2$ a $t = 5$, iniciando con $x(2) = 0.32$. Incluya términos hasta h^3 inclusive.

- ^a9. Escriba un programa que resuelva el problema con valor inicial $x' = e^t x$ con $x(2) = 1$ en el intervalo $0 \leq t \leq 2$ usando el método de la serie de Taylor. Incluya términos hasta h^4 .

10. Escriba un programa que resuelva el problema $x' = tx + t^4$ en el intervalo $0 \leq t \leq 5$ con $x(5) = 3$. Use el método de la serie de Taylor con términos hasta h^4 .

11. Escriba un programa que resuelva el problema con valor inicial del ejemplo de esta sección en el intervalo $[1, 3]$. Explique.

12. Calcule una tabla, de 101 puntos igualmente espaciados en el intervalo $[0, 2]$, de la **integral de Dawson**

$$f(x) = \exp(-x^2) \int_0^x \exp(t^2) dt$$

presolviendo numéricamente, con el método de la serie de Taylor de orden adecuado, un problema con valor inicial en el que f es la solución. Haga la tabla exacta a ocho lugares decimales e imprima sólo ocho lugares decimales. *Sugerencia:* encuentre la relación entre $f'(x)$ y $xf(x)$. El teorema fundamental del cálculo es útil. *Compruebe los valores:* $f(1) = 0.53807\ 95069$ y $f(2) = 0.30134\ 03889$.

13. Resuelva el problema con valor inicial $x' = t^3 + e^x$ con $x(3) = 7.4$ en el intervalo $0 \leq t \leq 3$ usando el método de la serie de Taylor de cuarto orden.

14. Use un paquete de manejo simbólico como Maple para resolver las ecuaciones diferenciales del ejemplo 1 con el método de la serie de Taylor de cuarto orden de alta precisión, con 24 dígitos decimales.

15. Programe los seudocódigos de Euler y de Taylor y compare los resultados numéricos con los dados en el libro.

16. (Continuación) Repita llamando directamente una rutina de solución de ecuación diferencial ordinaria dentro de un sistema de software matemático como Matlab, Maple o Mathematica.

17. Use un software matemático como Matlab, Maple o Mathematica para encontrar soluciones analíticas o numéricas de las ecuaciones diferenciales ordinarias del inicio de esta sección:

a. (2)

b. (3)

c. (4)

18. Escriba programas de cómputo para reproducir las figuras siguientes:

a. Figura 10.1 b. Figure 10.2 c. Figura 10.3

10.2 Métodos de Runge-Kutta

Los métodos nombrados en honor de Carl Runge y Wilhelm Kutta están diseñados para imitar el método de la serie de Taylor sin requerir derivación analítica de la ecuación diferencial original. Recuerde que al usar el método de la serie de Taylor en el problema con valor inicial

$$\begin{cases} x' = f(t, x) \\ x(a) = x_a \end{cases} \quad (1)$$

necesitamos obtener x'', x''', \dots derivando la función f . Este requisito puede ser un serio obstáculo para aplicar el método. El usuario de este método debe realizar un trabajo analítico

preliminar antes de escribir un programa de cómputo. Idealmente, un método para resolver la ecuación (1) debería implicar nada más escribir un código para evaluar f . Los métodos de Runge-Kutta logran esto.

Para propósitos de la exposición, se presenta el método de Runge-Kutta de orden 2, aunque su precisión es baja y en general impide su uso real en cálculos científicos. Más tarde se presenta el método de Runge-Kutta de orden 4 sin una deducción. Es de uso común. El procedimiento de Runge-Kutta de orden 2 encuentra aplicación en cálculos en tiempo real en equipos pequeños. Por ejemplo, se utiliza en algunos aviones en la minicomputadora de a bordo.

El corazón de cualquier método para la resolución de un problema con valor inicial es un procedimiento para avanzar en la función solución dando un paso a la vez, es decir, una fórmula que se debe dar para $x(t+h)$ en términos de cantidades conocidas. Como ejemplos de cantidades conocidas podemos citar $x(t), x(t-h), x(t-2h), \dots$ si el proceso de resolución ha pasado por una serie de pasos. Al principio, sólo se conoce $x(a)$. Por supuesto, suponemos que $f(t, x)$ se puede calcular para cualquier punto (t, x) .

Serie de Taylor para $f(x, y)$

Antes de explicar el método de Runge-Kutta de orden 2, vamos a presentar la serie de Taylor en dos variables. La serie infinita es

$$f(x + h, y + k) = \sum_{i=0}^{\infty} \frac{1}{i!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^i f(x, y) \quad (2)$$

Esta serie es análoga a la serie de Taylor de una variable dada con la ecuación (11) en la sección 1.2. Los términos que parecen misteriosos en la ecuación (2) se interpretan como sigue:

$$\begin{aligned} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^0 f(x, y) &= f \\ \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^1 f(x, y) &= h \frac{\partial f}{\partial x} + k \frac{\partial f}{\partial y} \\ \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^2 f(x, y) &= h^2 \frac{\partial^2 f}{\partial x^2} + 2hk \frac{\partial^2 f}{\partial x \partial y} + k^2 \frac{\partial^2 f}{\partial y^2} \\ &\vdots \end{aligned}$$

donde f y todas las derivadas parciales están evaluadas en (x, y) . Como en el caso de una variable, si se trunca la serie de Taylor, se necesita un término de error o un término de residuo para restablecer la igualdad. En este caso esta la ecuación apropiada:

$$f(x + h, y + k) = \sum_{i=0}^{n-1} \frac{1}{i!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^i f(x, y) + \frac{1}{n!} \left(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y} \right)^n f(\bar{x}, \bar{y}) \quad (3)$$

El punto (\bar{x}, \bar{y}) se encuentra en el segmento de recta que une a (x, y) con $(x + h, y + k)$ en el plano.

En la aplicación de la serie de Taylor se utilizan subíndices para denotar derivadas parciales. Así, por ejemplo,

$$f_x = \frac{\partial f}{\partial x} \quad f_t = \frac{\partial f}{\partial t} \quad f_{xx} = \frac{\partial^2 f}{\partial x^2} \quad f_{xt} = \frac{\partial^2 f}{\partial t \partial x} \quad (4)$$

Se trata de funciones para las que el orden de estos subíndices no es importante; por ejemplo, $f_{xt} = f_{tx}$. Así, tenemos

$$\begin{aligned} f(x + h, y + k) &= f + (hf_x + kf_y) \\ &\quad + \frac{1}{2!} (h^2 f_{xx} + 2hk f_{xy} + k^2 f_{yy}) \\ &\quad + \frac{1}{3!} (h^3 f_{xxx} + 3h^2 k f_{xxy} + 3hk^2 f_{xyy} + k^3 f_{yyy}) \\ &\quad + \dots \end{aligned}$$

Como casos especiales, vemos que

$$\begin{aligned} f(x + h, y) &= f + hf_x + \frac{h^2}{2!} f_{xx} + \frac{h^3}{3!} f_{xxx} + \dots \\ f(x, y + k) &= f + kf_y + \frac{k^2}{2!} f_{yy} + \frac{k^3}{3!} f_{yyy} + \dots \end{aligned}$$

Método de Runge-Kutta de orden 2

En el método de Runge-Kutta de orden 2 se adopta una fórmula que tiene dos evaluaciones de la función en la forma especial

$$\begin{cases} K_1 = hf(t, x) \\ K_2 = hf(t + \alpha h, x + \beta K_1) \end{cases}$$

y una combinación lineal de estas se agrega al valor de x en t para obtener el valor en $t + h$:

$$x(t + h) = x(t) + w_1 K_1 + w_2 K_2$$

o, equivalentemente,

$$x(t + h) = x(t) + w_1 hf(t, x) + w_2 hf(t + \alpha h, x + \beta hf(t, x)) \quad (5)$$

El objetivo es determinar las constantes w_1 , w_2 , α y β de modo que la ecuación (5) sea tan exacta como sea posible. Explícitamente, queremos reproducir tantos términos como sea posible en la serie de Taylor

$$x(t + h) = x(t) + hx'(t) + \frac{1}{2!}h^2x''(t) + \frac{1}{3!}h^3x'''(t) + \dots \quad (6)$$

Ahora, comparemos la ecuación (5) con la ecuación (6). Una manera de forzarlas a concordar con el término en h es hacer $w_1 = 1$ y $w_2 = 0$ porque $x' = f$. Sin embargo, esto simplemente reproduce el método de Euler (descrito en la sección anterior) y su orden de precisión es de sólo 1. Concordar con el término h^2 es posible con una elección más inteligente de los parámetros. Para ver cómo hacerlo, se aplica la forma de dos variables de la serie de Taylor en el término final de la ecuación (5). Usamos $n = 2$ en la serie de Taylor de dos variables dada por la fórmula (3), con t , αh , x y βhf en lugar de x , h y k , respectivamente:

$$f(t + \alpha h, x + \beta hf) = f + \alpha hf_t + \beta hff_x + \frac{1}{2} \left(\alpha h \frac{\partial}{\partial t} + \beta hf \frac{\partial}{\partial x} \right)^2 f(\bar{x}, \bar{y})$$

Usando la ecuación anterior se obtiene en una nueva forma de la ecuación (5). Tenemos

$$x(t + h) = x(t) + (w_1 + w_2)hf + \alpha w_2 h^2 f_t + \beta w_2 h^2 ff_x + \mathcal{O}(h^3) \quad (7)$$

La ecuación (6) también está dando una nueva forma de utilizar la ecuación diferencial (1). Puesto que $x' = f$, tenemos

$$x'' = \frac{dx'}{dt} = \frac{df(t, x)}{dt} = \left(\frac{\partial f}{\partial t} \right) \left(\frac{dt}{dt} \right) + \left(\frac{\partial f}{\partial x} \right) \left(\frac{dx}{dt} \right) = f_t + f_x f$$

Así, la ecuación (6) implica que

$$x(t+h) = x(t) + hf + \frac{1}{2}h^2 f_t + \frac{1}{2}h^2 f f_x + \mathcal{O}(h^3) \quad (8)$$

La concordancia entre las ecuaciones (7) y (8) se logra estableciendo que

$$w_1 + w_2 = 1 \quad \alpha w_2 = \frac{1}{2} \quad \beta w_2 = \frac{1}{2} \quad (9)$$

Una adecuada solución de estas ecuaciones es

$$\alpha = 1 \quad \beta = 1 \quad w_1 = \frac{1}{2} \quad w_2 = \frac{1}{2}$$

El **método de Runge-Kutta de segundo orden** resultante es, entonces, de la ecuación (5),

$$x(t+h) = x(t) + \frac{h}{2}f(t, x) + \frac{h}{2}f(t+h, x+hf(t, x))$$

o, equivalentemente,

$$x(t+h) = x(t) + \frac{1}{2}(K_1 + K_2) \quad (10)$$

dónde

$$\begin{cases} K_1 = hf(t, x) \\ K_2 = hf(t+h, x+K_1) \end{cases}$$

La fórmula (10) muestra que la función solución en $t+h$ se calcula a expensas de las dos evaluaciones de la función f .

Observe que son posibles otras soluciones para el sistema no lineal (9). Por ejemplo, α puede ser arbitraria y entonces

$$\beta = \alpha \quad w_1 = 1 - \frac{1}{2\alpha} \quad w_2 = \frac{1}{2\alpha}$$

Se puede demostrar (véase el problema 10.2.10) que el término de error en los métodos de Runge-Kutta de orden 2 es

$$\frac{h^3}{4} \left(\frac{2}{3} - \alpha \right) \left(\frac{\partial}{\partial t} + f \frac{\partial}{\partial x} \right)^2 f + \frac{h^3}{6} f_x \left(\frac{\partial}{\partial t} + f \frac{\partial}{\partial x} \right) f \quad (11)$$

Observe que el método con $\alpha = \frac{2}{3}$ es especialmente interesante. Sin embargo, ninguno de los métodos de Runge-Kutta de segundo orden es muy usado en computadoras grandes, ya que el error es de sólo $\mathcal{O}(h^3)$.

Método de Runge-Kutta de orden 4

Un algoritmo de uso común para el problema con valor inicial (1) es el **método de Runge-Kutta de cuarto orden** clásico. Sus fórmulas son las siguientes:

$$x(t+h) = x(t) + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad (12)$$

donde

$$\begin{aligned} K_1 &= hf(t, x) \\ K_2 &= hf\left(t + \frac{1}{2}h, x + \frac{1}{2}K_1\right) \\ K_3 &= hf\left(t + \frac{1}{2}h, x + \frac{1}{2}K_2\right) \\ K_4 &= hf(t + h, x + K_3) \end{aligned}$$

La deducción de las fórmulas de Runge-Kutta de orden 4 es tediosa. Muy pocos textos dan los detalles. Dos excepciones son los libros de Henrici [1962] y Ralston [1965]. Existen fórmulas de Runge-Kutta de orden más alto y aún son más tediosas de deducir. Sin embargo, los paquetes de software de manejo simbólico como Maple o Mathematica se pueden utilizar para desarrollar las fórmulas.

Como puede verse, la solución en $x(t + h)$ se obtiene a expensas de la evaluación de la función f cuatro veces. La fórmula final concuerda con el desarrollo de Taylor hasta el término en h^4 inclusive. Por lo tanto, el error contiene h^5 pero no potencias menores de h . Sin conocer el coeficiente de h^5 en el error no podemos ser precisos acerca del error de truncamiento local. En los tratados dedicados a esta materia se exploran más estos temas. Véase, por ejemplo, Butcher [1987] o Gear [1971].

Seudocódigo

Aquí se presenta un seudocódigo para implementar el método de Runge-Kutta de orden 4 clásico:

```

procedure RK4(f, t, x, h, n)
integer j, n;      real K1, K2, K3, K4, h, t, ta, x
external function f
output 0, t, x
ta ← t
for j = 1 to n do
    K1 ← hf(t, x)
    K2 ← hf(t + 1/2h, x + 1/2K1)
    K3 ← hf(t + 1/2h, x + 1/2K2)
    K4 ← hf(t + h, x + K3)
    x ← x + 1/6(K1 + 2K2 + 2K3 + K4)
    t ← ta + jh
    output j, t, x
end for
end procedure RK4

```

Para ejemplificar el uso del seudocódigo anterior, considere el problema con valor inicial

$$\begin{cases} x' = 2 + (x - t - 1)^2 \\ x(1) = 2 \end{cases} \quad (13)$$

cuya solución exacta es $x(t) = 1 + t + \tan(t - 1)$. Un seudocódigo que resuelve este problema en el intervalo $[1, 1.5625]$ con el procedimiento de Runge-Kutta es el siguiente. El tamaño de paso necesario se calcula al dividir la longitud del intervalo entre el número de pasos, digamos, $n = 72$.

```
program Test_RK4
real h, t;      external function f
integer n ← 72
real a ← 1, b ← 1.5625, x ← 2
h ← (b - a)/ n
t ← a
call RK4(f, t, x, h, n)
end program Test_RK4
```

```
real function f(t, x)
real t, x
f ← 2 + (x - t - 1)2
end function f
```

Incluimos un enunciado de la función externa, tanto en el programa principal como en el procedimiento *RK4*, ya que el procedimiento *f* se pasa en la lista de argumentos de *RK4*. El valor final de la solución numérica calculada es $x(1.5625) = 3.19293\ 7699$.

Las rutinas de propósito general incorporadas en el algoritmo de Runge-Kutta normalmente incluyen más programación para controlar el error de truncamiento y hacer los ajustes necesarios en el tamaño de paso conforme la solución avanza. En términos generales, el tamaño de paso puede ser grande cuando la solución varía lentamente, pero debe ser pequeño cuando varía con rapidez. Este programa se presenta en la sección 10.3.

Resumen

(1) El **método de Runge-Kutta de segundo orden** es

$$x(t + h) = x(t) + \frac{1}{2}(K_1 + K_2)$$

donde

$$\begin{cases} K_1 = hf(t, x) \\ K_2 = hf(t + h, x + K_1) \end{cases}$$

Este método requiere dos evaluaciones de la función *f* por paso. Es equivalente a un método de Taylor de orden 2.

(2) Uno de los más populares métodos de un solo paso para la solución de EDO es el **método de Runge-Kutta de cuarto orden**

$$x(t + h) = x(t) + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

donde

$$\begin{cases} K_1 = hf(t, x) \\ K_2 = hf\left(t + \frac{1}{2}h, x + \frac{1}{2}K_1\right) \\ K_3 = hf\left(t + \frac{1}{2}h, x + \frac{1}{2}K_2\right) \\ K_4 = hf(t + h, x + K_3) \end{cases}$$

Se necesitan cuatro evaluaciones de la función f por paso. Puesto que equivale al método de la serie de Taylor de orden 4, tiene error de truncamiento de orden $\mathcal{O}(h^5)$. El pequeño número de evaluaciones de la función y el error de truncamiento de alto orden contribuyen a su popularidad.

Problemas 10.2

- Deduzca las ecuaciones necesarias para aplicar el método de la serie de Taylor de cuarto orden a la ecuación diferencial $x' = tx^2 + x - 2t$. Compárelo en complejidad con las ecuaciones necesarias para el método Runge-Kutta de cuarto orden.
- Ponga estas ecuaciones diferenciales en una forma adecuada para solución numérica por el método de Runge-Kutta.
 - $x + 2xx' - x' = 0$
 - $\log x' = t^2 - x^2$
 - $(x')^2(1 - t^2) = x$

- ^a3. Resuelva la ecuación diferencial

$$\begin{cases} \frac{dx}{dt} = -tx^2 \\ x(0) = 2 \end{cases}$$

en $t = 0.2$, correcta a dos decimales, usando un paso del método de la serie de Taylor de orden 2 y un paso del método de Runge-Kutta de orden 2.

4. Considere la ecuación diferencial ordinaria

$$\begin{cases} x' = (tx)^3 - (x/t)^2 \\ x(1) = 1 \end{cases}$$

Tome un paso del método de la serie de Taylor de orden 2, con $h = 0.1$, y después use el método de Runge-Kutta de orden 2 para volver a calcular $x(1.1)$. Compare las respuestas.

- Al resolver las siguientes ecuaciones diferenciales usando un procedimiento de Runge-Kutta es necesario escribir el código para una función $f(t, x)$. Hágalo para cada una de las siguientes ecuaciones:
 - $x' = t^2 + tx' - 2xx'$
 - $x' = e^t + x'\cos x + t^2$
- Considere la ecuación diferencial ordinaria $x' = t^3x^2 - 2x^3/t^2$ con $x(1) = 0$. Determine las ecuaciones que se utilizan en la aplicación del método de la serie de Taylor de orden 3 y del método de Runge-Kutta de orden 4.
- Considere el **método de Runge-Kutta de tercer orden**:

$$x(t + h) = x(t) + \frac{1}{9}(2K_1 + 3K_2 + 4K_3)$$

donde

$$\begin{cases} K_1 = hf(t, x) \\ K_2 = hf\left(t + \frac{1}{2}h, x + \frac{1}{2}K_1\right) \\ K_3 = hf\left(t + \frac{3}{4}h, x + \frac{3}{4}K_2\right) \end{cases}$$

- a. Muestre que esto concuerda con el método de la serie de Taylor del mismo orden para la ecuación diferencial $x' = x + t$.
 - b. Pruebe que este método de Runge-Kutta de tercer orden reproduce la serie de Taylor de la solución hasta términos en h^3 inclusive para *cualquier* ecuación diferencial.
- ^a8. Describa cómo el método de Runge-Kutta de cuarto orden se puede usar para generar una tabla de valores para la función

$$f(x) = \int_0^x e^{-t^2} dt$$

en 100 puntos igualmente espaciados en el intervalo unitario. *Sugerencia:* determine un problema con valor inicial adecuado cuya solución es f .

9. Demuestre que la fórmula de Runge-Kutta de cuarto orden se reduce a una forma simple cuando se aplica a una ecuación diferencial ordinaria de la forma

$$x' = f(t)$$

- ^a10. Establezca el término de error (11) para los métodos de Runge-Kutta de orden 2.

- ^a11. En determinada computadora, se encontró que cuando el método de Runge-Kutta de cuarto orden fue usado en un intervalo $[a, b]$ con $h = (b - a)/n$, el error total debido al redondeo fue aproximadamente de $36n2^{-50}$ y el error de truncamiento total fue de $9nh^5$, donde n es el número de pasos y h es el tamaño de paso. ¿Cuál es un óptimo valor de h ? *Sugerencia:* minimice el error total: error de redondeo más error de truncamiento.

- ^a12. ¿Cómo resolvería el problema con valor inicial

$$\begin{cases} x' = \sin x + \sin t \\ x(0) = 0 \end{cases}$$

en el intervalo $[0, 1]$ si se necesitan diez lugares decimales de exactitud? Suponga que tiene una computadora en la que el error de redondeo es $\frac{1}{2} \times 10^{-14}$ y suponga que el método de Runge-Kutta de cuarto orden implicará errores de truncamiento local de magnitud $100h^5$.

13. Un importante teorema de cálculo establece que la ecuación $f_{xx} = f_{xt}$ es verdadera siempre que al menos una de estas dos derivadas parciales exista y sea continua. Pruebe esta ecuación en algunas funciones, tales como $f(t, x) = xt^2 + x^2t + x^3t^4$, $\log(x - t^{-1})$ y $e^x \operatorname{senh}(t + x) + \cos(2x - 3t)$.

14. a. Si $x' = f(x, t)$, entonces

$$x'' = Df, \quad x''' = D^2f + f_x Dff$$

donde

$$D = \frac{\partial}{\partial t} + f \frac{\partial}{\partial x}, \quad D^2 = \frac{\partial^2}{\partial t^2} + 2f \frac{\partial^2}{\partial x \partial t} + f^2 \frac{\partial^2}{\partial x^2}$$

Compruebe estas ecuaciones.

- b. Determine $x^{(4)}$ en una forma similar.

- 15.** Deduzca la forma de dos variables de la serie de Taylor a partir de la forma de una variable considerando la función de una variable $\phi(t) = f(x + th, y + tk)$ y desarrollela con el teorema de Taylor.

- 16.** El desarrollo de la serie de Taylor con respecto al punto (a, b) en términos de dos variables x y y está dado por

$$f(x, y) = \sum_{i=0}^{\infty} \frac{1}{i!} \left((x - a) \frac{\partial}{\partial x} + (y - b) \frac{\partial}{\partial y} \right)^i f(a, b)$$

Demuestre que la fórmula (2) se puede obtener a partir de esta forma con un cambio de variables.

- 17.** (Continuación) Usando la forma dada en el problema anterior, determine los primeros cuatro términos diferentes de cero en la serie de Taylor para $f(x, y) = \sin x + \cos y$ con respecto al punto $(0, 0)$. Compare el resultado con las series conocidas para $\sin x$ y $\cos y$. Haga una conjectura acerca de la serie de Taylor para las funciones que tengan la forma especial $f(x, y) = g(x) + h(y)$.

- 18.** Para la función $f(x, y) = y^2 - 3 \ln x$, escriba los primeros seis términos en la serie de Taylor de $f(1 + h, 0 + k)$.

- 19.** Usando la serie de Taylor truncada con respecto a $(1, 1)$, dé una aproximación de tres términos a $e^{(1-xy)}$ *Sugerencia:* use el problema 10.2.16.

- 20.** La función $f(x, y) = xe^y$ se puede aproximar con la serie de Taylor en dos variables por medio de $f(x + h, y + k) \approx (Ax + B)e^y$. Determine A y B cuando se usan los términos con segundas derivadas parciales en la serie.

- 21.** Para $f(x, y) = (y - x)^{-1}$, la serie de Taylor se puede escribir como

$$f(x + h, y + k) = Af + Bf^2 + Cf^3 + \dots$$

donde $f = f(x, y)$. Determine los coeficientes A , B y C .

- 22.** Considere la función e^{x^2+y} . Determine su serie de Taylor con respecto al punto $(0, 1)$ con los términos de segundas derivadas parciales. Use este resultado para obtener un valor aproximado para $f(0.001, 0.998)$.

- 23.** Demuestre que el método de Euler mejorado es un método de Runge-Kutta de orden 2.

Problemas de cómputo 10.2

- Ejecute el ejemplo de seudocódigo que se presentó en el libro para la ecuación diferencial (13) al ejemplificar el método de Runge-Kutta.
- Resuelva el problema con valor inicial $x' = x/t + t \sec(x/t)$ con $x(0) = 0$ con el método de Runge-Kutta de cuarto orden. Continúe la solución para $t = 1$ usando un tamaño de paso $h = 2^{-7}$. Compare la solución numérica con la solución exacta, que es $x(t) = t \arcsen t$. Defina $f(0, 0) = 0$, donde $f(t, x) = x/t + t \sec(x/t)$.
- Seleccione uno de los siguientes problemas con valores iniciales y compare las soluciones numéricas obtenidas con las fórmulas de Runge-Kutta de cuarto orden y con la serie de Taylor de cuarto orden.

Use diferentes valores de $h = 2^{-n}$, para $n = 2, 3, \dots, 7$ para calcular la solución en el intervalo $[1, 2]$.

a. $x' = 1 + x/t \quad x(1) = 1$

"b. $x' = 1/x^2 - xt \quad x(1) = 1$

"c. $x' = 1/t^2 - x/t - x^2 \quad x(1) = -1$

- 4.** Seleccione una rutina de Runge-Kutta de una biblioteca de programación y pruébela en el problema con valor inicial $x' = (2-t)x$ con $x(2) = 1$. Compare con la solución exacta, $[-(\frac{1}{2})(t-2)^2]$.
- 5. (EDO mal-condicionada)** Resuelva la ecuación diferencial ordinaria $x' = 10x + 11t - 5t^2 - 1$ con valor inicial $x(0) = 0$. Continúe la solución de $t = 0$ a $t = 3$, usando el método de Runge-Kutta de cuarto orden con $h = 2^{-8}$. Imprima la solución numérica y la solución exacta ($t^2/2 - t$) en cada décimo paso y trace la gráfica de las dos soluciones. Compruebe que la solución de la misma ecuación diferencial con valor inicial $x(0) = \epsilon$ es $\epsilon e^{10t} + t^2/2 - t$ y por ello explique la diferencia entre las soluciones numérica y exacta del problema original.
- 6.** Resuelva el problema con valor inicial $x' = x\sqrt{x^2 - 1}$ con $x(0) = 1$ por el método de Runge-Kutta en el intervalo $0 \leq t \leq 1.6$ y explique cualquier dificultad. Después, usando h negativa, resuelva la misma ecuación diferencial en el mismo intervalo con valor inicial $x(1.6) = 1.0$.
- 7.** El siguiente ejemplo patológico fue dado por Dahlquist y Björck [1974]. Considere la ecuación diferencial $x' = 100(\sin t - x)$ con valor inicial $x(0) = 0$. Intégrela con el método de Runge-Kutta de cuarto orden en el intervalo $[0, 3]$, usando tamaños de paso $h = 0.015, 0.020, 0.025, 0.030$. ¡Observe la inestabilidad numérica!
- 8.** Considere la ecuación diferencial

$$\begin{cases} x' = \begin{cases} x + t & -1 \leq t \leq 0 \\ x - t & 0 \leq t \leq 1 \end{cases} \\ x(-1) = 1 \end{cases}$$

Usando el procedimiento de Runge-Kutta *RK4* con tamaño de paso $h = 0.1$, resuelva este problema en el intervalo $[-1, 1]$. Ahora resuélvalo usando $h = 0.09$. ¿Cuál solución numérica es más exacta y por qué? *Sugerencia:* la solución verdadera está dada por $x = e^{(t+1)} - (t+1)$ si $t \leq 0$ y $x = e^{(t+1)} - 2e^t + (t+1)$ si $t \geq 0$.

- 9.** Resuelva $t - x' + 2xt = 0$ con $x(0) = 0$ en el intervalo $[0, 10]$ usando las fórmulas de Runge-Kutta con $h = 0.1$. Compare con la solución verdadera: $\frac{1}{2}(e^{t^2} - 1)$. Dibuje una gráfica o consiga una creada con un trazador automático. Después trace la gráfica del logaritmo de la solución.
- 10.** Escriba un programa que resuelva el problema $x' = \sin(xt) + \arctan t$ en $1 \leq t \leq 7$ con $x(2) = 4$ usando el procedimiento de Runge-Kutta *RK4*.
- 11.** La forma general de los métodos de Runge-Kutta de orden 2 está dada por las ecuaciones (5) y (10). Escriba y pruebe el procedimiento *RK2(f, t, x, h, α, n)* para realizar n pasos con tamaño de paso h y las condiciones iniciales t y x para varios valores de $α$ dados.

12. Queremos resolver

$$\begin{cases} x' = e^t x^2 + e^3 \\ x(2) = 4 \end{cases}$$

en $x(5)$ con tamaño de paso 0.5. Resuélvala de las dos formas siguientes.

- a. Codifique la función $f(t, x)$ que se necesita y use el procedimiento *RK4*.
- b. Escriba un programa corto que use el método de la serie de Taylor incluidos los términos hasta h^4 .

13. Dibuje la solución para la ecuación diferencial (13).

14. Seleccione una ecuación diferencial con una solución conocida y compare el método de Runge-Kutta de cuarto orden con uno o ambos de los siguientes. Imprima los errores en cada paso. ¿Es el cociente de los dos errores una constante en cada paso? ¿Cuáles son las ventajas o desventajas de cada método?

- a. Un método Runge-Kutta de cuarto orden similar al clásico está dado por

$$x(t+h) = x(t) + \frac{1}{6}(K_1 + 4K_3 + K_4)$$

donde

$$\begin{cases} K_1 = hf(t, x) \\ K_2 = hf\left(t + \frac{1}{2}h, x + \frac{1}{2}K_1\right) \\ K_3 = hf\left(t + \frac{1}{2}h, x + \frac{1}{4}K_1 + \frac{1}{4}K_2\right) \\ K_4 = hf(t + h, x - K_2 + 2K_3) \end{cases}$$

Véase England [1969] o Shampine, Alien y Pruess [1997].

- b. Otro método de Runge-Kutta de cuarto orden esta dado por

$$x(t+h) = x(t) + w_1 K_1 + w_2 K_2 + w_3 K_3 + w_4 K_4$$

donde

$$\begin{cases} K_1 = hf(t, x) \\ K_2 = hf\left(t + \frac{2}{5}h, x + \frac{2}{5}K_1\right) \\ K_3 = hf\left(t + \frac{1}{16}(14 - 3\sqrt{5})h, x + c_{31}K_1 + c_{32}K_2\right) \\ K_4 = hf(t + h, x + c_{41}K_1 + c_{42}K_2 + c_{43}K_3) \end{cases}$$

En este caso las constantes apropiadas son

$$c_{31} = \frac{3(-963 + 476\sqrt{5})}{1024} \quad c_{32} = \frac{5(757 - 324\sqrt{5})}{1024}$$

$$c_{41} = \frac{-3365 + 2094\sqrt{5}}{6040} \quad c_{42} = \frac{-975 - 3046\sqrt{5}}{2552}$$

$$c_{43} = \frac{32(14595 + 6374\sqrt{5})}{240845}$$

$$w_1 = \frac{263 + 24\sqrt{5}}{1812} \quad w_2 = \frac{125(1 - 8\sqrt{5})}{3828}$$

$$w_3 = \frac{1024(3346 + 1623\sqrt{5})}{5924787} \quad w_4 = \frac{2(15 - 2\sqrt{5})}{123}$$

Nota: hay cualquier número de métodos de Runge-Kutta de cualquier orden. Cuanto mayor sea el orden, más complicadas son las fórmulas. Puesto que la dada por la ecuación (12) tiene error $\mathcal{O}(h^5)$ y es bastante simple, es el más popular método de Runge-Kutta de cuarto orden. El término de error para el método del inciso b de este problema también es $\mathcal{O}(h^5)$ y es óptimo en cierto sentido (véase Ralston [1965] para más detalles).

15. Un método de Runge-Kutta de quinto orden está dado por

$$x(t+h) = x(t) + \frac{1}{24}K_1 + \frac{5}{48}K_4 + \frac{27}{56}K_5 + \frac{125}{336}K_6$$

donde

$$\left\{ \begin{array}{l} K_1 = hf(t, x) \\ K_2 = hf\left(t + \frac{1}{2}h, x + \frac{1}{2}K_1\right) \\ K_3 = hf\left(t + \frac{1}{2}h, x + \frac{1}{4}K_1 + \frac{1}{4}K_2\right) \\ K_4 = hf(t+h, x - K_2 + 2K_3) \\ K_5 = hf\left(t + \frac{2}{3}h, x + \frac{7}{27}K_1 + \frac{10}{27}K_2 + \frac{1}{27}K_4\right) \\ K_6 = hf\left(t + \frac{1}{5}h, x + \frac{28}{625}K_1 - \frac{1}{5}K_2 + \frac{546}{625}K_3 + \frac{54}{625}K_4 - \frac{378}{625}K_5\right) \end{array} \right.$$

Escriba y pruebe un procedimiento que use esta fórmula.

16. a. Use un paquete de manejo simbólico como Maple o Mathematica para encontrar el método de Runge-Kutta general de orden 2.
 b. Repita para orden 3.
17. (**Ecuación diferencial ordinaria retrasada**) Investigue procedimientos para determinar la solución numérica de una ecuación diferencial ordinaria con un retraso constante tal como

$$x'(t) = -x(t) + x(t-20) + \frac{1}{20} \cos\left(\frac{1}{20}t\right) + \sin\left(\frac{1}{20}t\right) - \sin\left(\frac{1}{20}(t-20)\right)$$

en el intervalo $0 \leq t \leq 1000$, donde $x(t) = \sin\left(\frac{1}{20}t\right)$ para $t \leq 0$. Use un tamaño de paso menor o igual a 20 de modo que no haya traslape. Compare con la solución exacta $x(t) = \sin\left(\frac{1}{20}t\right)$.

18. Escriba un software para el programa *Test_RK4* y la rutina *RK4* y compruebe los resultados numéricos dados en el libro.

10.3 Estabilidad y adaptación de los métodos de Runge-Kutta y de multipaso

Un método adaptado de Runge-Kutta-Fehlberg

En situaciones reales que implican la resolución numérica de problemas con valores iniciales, siempre hay necesidad de estimar la precisión lograda en el cálculo. En general, se da una tolerancia del error y la solución numérica no se debe desviar de la solución verdadera

más allá de esta tolerancia. Una vez que se ha seleccionado un método, la tolerancia del error determina el tamaño de paso máximo permitido. Incluso si consideramos solamente el error de truncamiento local, la determinación de un tamaño de paso adecuado puede ser difícil. Además, a menudo se necesita un tamaño de paso pequeño en una parte de la curva solución, mientras que uno más grande puede ser suficiente en otras partes.

Por las razones dadas, se han desarrollado distintos métodos que ajustan *automáticamente* el tamaño de paso en los algoritmos para el problema con valor inicial. Ahora describiremos un procedimiento simple. Considere el método clásico de Runge-Kutta de cuarto orden analizado en la sección 10.2. Para avanzar la curva solución de t a $t + h$ podemos tomar un paso de tamaño h usando las fórmulas de Runge-Kutta. Pero podemos también tomar *dos* pasos de tamaño $h/2$ para llegar a $t + h$. Si no hay error de truncamiento, el valor de la solución numérica $x(t + h)$ sería el mismo con ambos procedimientos. La diferencia en los resultados numéricos se puede tomar como una estimación del error de truncamiento local. Por ello, en la práctica, si esta diferencia está dentro de la tolerancia dada, el tamaño de paso h actual es satisfactorio. Si esta diferencia excede la tolerancia, el tamaño de paso se reducirá a la mitad. Si la diferencia es mucho menor que la tolerancia, se duplica el tamaño de paso.

El procedimiento que acabamos de exponer es fácil de programar, pero desperdicia tiempo de máquina y no es recomendable. Un método más refinado fue desarrollado por Fehlberg [1969]. El **método de Fehlberg de orden 4** es del tipo de Runge-Kutta y utiliza estas fórmulas

$$x(t + h) = x(t) + \frac{25}{216}K_1 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5$$

donde

$$\left\{ \begin{array}{l} K_1 = hf(t, x) \\ K_2 = hf\left(t + \frac{1}{4}h, x + \frac{1}{4}K_1\right) \\ K_3 = hf\left(t + \frac{3}{8}h, x + \frac{3}{32}K_1 + \frac{9}{32}K_2\right) \\ K_4 = hf\left(t + \frac{12}{13}h, x + \frac{1932}{2197}K_1 - \frac{7200}{2197}K_2 + \frac{7296}{2197}K_3\right) \\ K_5 = hf\left(t + h, x + \frac{439}{216}K_1 - 8K_2 + \frac{3680}{513}K_3 - \frac{845}{4104}K_4\right) \end{array} \right.$$

Puesto que este método necesita una evaluación más de la función que el método clásico de Runge-Kutta de orden 4, su valor por sí solo es cuestionable. Sin embargo, con una evaluación adicional de la función

$$K_6 = hf\left(t + \frac{1}{2}h, x - \frac{8}{27}K_1 + 2K_2 - \frac{3544}{2565}K_3 + \frac{1859}{4104}K_4 - \frac{11}{40}K_5\right)$$

podemos obtener un **método de Runge-Kutta de quinto orden**, a saber,

$$x(t + h) = x(t) + \frac{16}{135}K_1 + \frac{6656}{12825}K_3 + \frac{28561}{56430}K_4 - \frac{9}{50}K_5 + \frac{2}{55}K_6$$

La diferencia entre los valores de $x(t + h)$ obtenidos de los procedimientos de cuarto y quinto orden es una estimación del error de truncamiento local en el procedimiento de cuarto orden. ¡Por ello, con seis evaluaciones de la función se obtiene una aproximación de quinto orden junto con una estimación del error!

En el procedimiento RK45 se da un seudocódigo para el método de Runge-Kutta-Fehlberg:

```

procedure RK45(f, t, x, h, ε)
real ε, K1, K2, K3, K4, K5, K6, h, t, x, x4
external function f
real c20 ← 0.25, c21 ← 0.25
real c30 ← 0.375, c31 ← 0.09375, c32 ← 0.28125
real c40 ← 12./13., c41 ← 1932./2197.
real c42 ← -7200./2197., c43 ← 7296./2197.
real c51 ← 439./216., c52 ← -8.
real c53 ← 3680./513., c54 ← -845./4104.
real c60 ← 0.5, c61 ← -8./27., c62 ← 2.
real c63 ← -3544./2565., c64 ← 1859./4104.
real c65 ← -0.275
real a1 ← 25./216., a2 ← 0., a3 ← 1408./2565.
real a4 ← 2197./4104., a5 ← -0.2
real b1 ← 16./135., b2 ← 0., b3 ← 6656./12825.
real b4 ← 28561./56430., b5 ← -0.18
real b6 ← 2./55.
K1 ← hf(t, x)
K2 ← hf(t + c20h, x + c21K1)
K3 ← hf(t + c30h, x + c31K1 + c32K2)
K4 ← hf(t + c40h, x + c41K1 + c42K2 + c43K3)
K5 ← hf(t + h, x + c51K1 + c52K2 + c53K3 + c54K4)
K6 ← hf(t + c60h, x + c61K1 + c62K2 + c63K3 + c64K4 + c65K5)
x4 ← x + a1K1 + a3K3 + a4K4 + a5K5
x ← x + b1K1 + b3K3 + b4K4 + b5K5 + b6K6
t ← t + h
ε ← |x - x4|
end procedure RK45

```

Por supuesto, el programador podría considerar diversas técnicas de optimización tales como la asignación de valores numéricos a los coeficientes de expansión decimal correspondientes la precisión del equipo que se utiliza para que las fracciones no tengan que calcularse de nuevo en cada llamada del procedimiento.

Podemos utilizar el procedimiento *RK45* en una forma no adaptada como en el siguiente programa:

```

program Prueba_RK45
integer k; real t, h, ε; external function f
integer n ← 72
real a ← 1.0, b ← 1.5625, x ← 2.0
h ← (b - a) / n
t ← a
output 0, t, x
for k = 1 to n do
    call RK45(f, t, x, h, ε)
    output k, t, x, ε
end for
end program Prueba_RK45

```

```

real function f(t, x)
real t, x
    f ← 2.0 + (x - t - 1.0)2
end function f

```

Aquí, imprimimos la estimación del error en cada paso. Sin embargo, podemos usarla en un procedimiento adaptado, ya que la estimación del error ε puede decírnos cuándo ajustar el tamaño de paso para controlar el *error de un solo paso*.

Ahora describiremos un procedimiento adaptado simple. En el procedimiento *RK45*, las aproximaciones de cuarto y quinto orden para $x(t+h)$, digamos, x_4 y x_5 , se calculan a partir de seis evaluaciones de la función, y se conoce la estimación del error $\varepsilon = |x_4 - x_5|$. De los límites especificados por el usuario en la estimación del error admisible ($\varepsilon_{\min} \leq \varepsilon \leq \varepsilon_{\max}$), el tamaño de paso h se duplica o se reduce a la mitad, según se necesite, para conservar ε dentro de estos límites. Un rango permisible para el tamaño de paso h es especificado por el usuario ($h_{\min} \leq |h| \leq h_{\max}$). Evidentemente, el usuario debe definir con cuidado los límites ($\varepsilon_{\min}, \varepsilon_{\max}, h_{\min}, h_{\max}$) para que el procedimiento adaptado no quede atrapado en un ciclo, tratando repetidas veces de dividir a la mitad y de duplicar el tamaño de paso desde el mismo punto para cumplir con los límites de error que son muy restrictivos para la ecuación diferencial dada.

Básicamente, nuestro proceso adaptado es el siguiente:

■ ALGORITMO 1 Panorama de un proceso adaptado

1. Dado el tamaño de paso h y un valor inicial $x(t)$, la rutina *RK45* calcula el valor $x(t+h)$ y una estimación del error ε .
2. Si $\varepsilon_{\min} \leq \varepsilon \leq \varepsilon_{\max}$, entonces el tamaño de paso h no se cambia y el paso siguiente se toma repitiendo el paso 1 con valor inicial $x(t+h)$.
3. Si $\varepsilon < \varepsilon_{\min}$, entonces h se sustituye por $2h$, siempre que $|2h| \leq h_{\max}$.
4. Si $\varepsilon > \varepsilon_{\max}$, entonces h se sustituye por $h/2$, siempre que $|h/2| \geq h_{\min}$.
5. Si $h_{\min} \leq |h| \leq h_{\max}$, entonces el paso se repite volviendo al paso 1 con $x(t)$ y el valor h nuevo.

El procedimiento de este método adaptado es *Adaptativo_RK45*. En la lista de parámetros del seudocódigo, f es la función $f(t, x)$ de la ecuación diferencial, t y x contienen los valores iniciales, h es el tamaño de paso inicial, t_b es el valor final de t , $itmax$ es el número máximo de pasos por seguir para ir de $a = t_a$ a $b = t_b$, ε_{\min} y ε_{\max} son los límites inferior y superior de la estimación del error permitido ε , h_{\min} y h_{\max} son los límites en el tamaño de paso h , e *iflag* es un indicador de error que devuelve uno de los valores siguientes:

iflag	Significado
0	Corre con éxito de t_a a t_b
1	Número máximo de iteraciones alcanzada

A su regreso, t y x son los valores de salida y h es el último valor de tamaño de paso considerado o utilizado:

```

procedure Adaptativo_RK45(f, t, x, h, tb, itmax, εmax, εmin, hmin, hmax, iflag)
integer iflag, itmax, n; external function f
real ε, εmax, εmin, d, h, hmin, hmax, t, tb, x, xsave, tsave
real δ ←  $\frac{1}{2} \times 10^{-5}$ 

```

```

output  $0, h, t, x$ 
 $iflag \leftarrow 1$ 
 $k \leftarrow 0$ 
while  $k \leq itmax$ 
     $k \leftarrow k + 1$ 
    if  $|h| < h_{\min}$  then  $h \leftarrow \text{sign}(h)h_{\min}$ 
    if  $|h| > h_{\max}$  then  $h \leftarrow \text{sign}(h)h_{\max}$ 
     $d \leftarrow |t_b - t|$ 
    if  $d \leq |h|$  then
         $iflag \leftarrow 0$ 
        if  $d \leq \delta \cdot \max\{|t_b|, |t|\}$  then exit loop
         $h \leftarrow \text{sign}(h)d$ 
    end if
     $x_{\text{save}} \leftarrow x$ 
     $t_{\text{save}} \leftarrow t$ 
    call RK45( $f, t, x, h, \varepsilon$ )
    output  $n, h, t, x, \varepsilon$ 
    if  $iflag = 0$  then exit loop
    if  $\varepsilon < \varepsilon_{\min}$  then  $h \leftarrow 2h$ 
    if  $\varepsilon > \varepsilon_{\max}$  then
         $h \leftarrow h / 2$ 
         $x \leftarrow x_{\text{save}}$ 
         $t \leftarrow t_{\text{save}}$ 
         $k \leftarrow k - 1$ 
    end if
end while
end procedure Adaptativo_RK45

```

En el seudocódigo, observe que se deben revisar varias condiciones para determinar el tamaño del paso final, puesto que está implicada la aritmética de punto flotante y el tamaño de paso varía.

Como un ejemplo, usted debe repetir el ejemplo de computadora de la sección anterior usando *Adaptativo_RK45*, que permite que el tamaño de paso sea variable, en lugar de *RK4*. Compare la exactitud de estas dos soluciones calculadas.

Un ejemplo industrial

Una ecuación diferencial de primer orden que surgió en el modelado de un proceso industrial químico es el siguiente:

$$x' = a + b \operatorname{sen} t + cx \quad x(0) = 0 \quad (1)$$

en la que $a = 3$, $b = 5$ y $c = 0.2$ son constantes. Esta ecuación es responsable de la solución de las técnicas de cálculo, en particular, la utilización de un factor de integración. Sin embargo, la solución analítica es complicada y puede ser preferible una solución numérica.

Para resolver este problema numéricamente usando las fórmulas adaptadas del método de Runge-Kutta, sólo hay que identificar (y programar) la función f que aparece en la descripción general. En este problema, esta es $f(t, x) = 3 + 5 \operatorname{sen} t + 0.2x$. Aquí se presenta un seudocódigo breve

para resolver la ecuación en el intervalo $[0, 10]$ con valores particulares asignados a los parámetros en la rutina *Adaptativo_RK45*:

```

program Prueba_Adaptativa_RK45
integer iflag; real t, x, h, tb; external function f
integer itmax ← 1000
real εmax ← 10-5, εmin ← 10-8, hmin ← 10-6, hmax ← 1.0
t ← 0.0; x ← 0.0; h ← 0.01; tb ← 10.0
call RK45_Adaptive(f, t, x, h, tb, itmax, εmax, εmin, hmin, hmax, iflag)
output itmax, iflag
end program Prueba_Adaptativa_RK45

real function f(t, x)
real t, x
f ← 3 + 5 sin(t) + 0.2x
end function f

```

Se obtiene la aproximación $x(10) \approx 135.917$. La salida del código es una tabla con valores que se pueden enviar a una rutina de trazado de gráficas. La gráfica resultante ayuda al usuario a visualizar la curva solución.

Fórmulas de Adams-Bashforth-Moulton

Ahora introducimos una estrategia en la que se utilizan las fórmulas de cuadratura numérica para resolver una ecuación diferencial ordinaria de primer orden simple. La ecuación del modelo es

$$x'(t) = f(t, x(t))$$

y suponemos que los valores de la función desconocida se han calculado en varios puntos a la izquierda de t , a saber, $t, t - h, t - 2h, \dots, t - (n - 1)h$. Queremos calcular $x(t + h)$. Por los teoremas del cálculo podemos escribir

$$\begin{aligned} x(t + h) &= x(t) + \int_t^{t+h} x'(s) ds \\ &= x(t) + \int_t^{t+h} f(s, x(s)) ds \\ &\approx x(t) + \sum_{j=1}^n c_j f_j \end{aligned}$$

donde se ha usado la abreviatura $f_j = f(t - (j - 1)h, x(t - (j - 1)h))$. En el último renglón de la ecuación anterior tenemos una fórmula de integración numérica adecuada. El caso más simple de dicha fórmula estará en el intervalo $[0, 1]$ y se usarán los valores del integrando en los puntos $0, -1, -2, \dots, 1 - n$ en el caso de una **fórmula de Adams-Bashforth**. Una vez que tenemos una regla básica, con un cambio de variable se obtendrá la regla para cualquier otro intervalo con cualquier otro espaciamiento uniforme.

Vamos a encontrar una regla de la forma

$$\int_0^1 F(r) dr \approx c_1 F(0) + c_2 F(-1) + \cdots + c_n F(1 - n)$$

Hay n coeficientes c_j a nuestra disposición. Sabemos de la teoría de interpolación que la fórmula se puede hacer exacta para todos los polinomios de grado $n - 1$. Basta con insistir en integrar cada función $1, r, r^2, \dots, r^{n-1}$ exactamente. Por tanto, escribimos a continuación la ecuación adecuada:

$$\int_0^1 r^{i-1} dt = \sum_{j=1}^n c_j (1-j)^{i-1} \quad (1 \leq i \leq n)$$

Este es un sistema $A\mathbf{u} = \mathbf{b}$ de n ecuaciones con n incógnitas. Los elementos de la matriz A son $A_{ij} = (1-j)^{i-1}$ y el miembro derecho es $b_i = 1/i$.

Cuando se corre este programa, la salida es el vector de coeficientes $(\frac{55}{24}, -\frac{59}{24}, \frac{37}{24}, -\frac{3}{8})$. Por supuesto, las fórmulas de orden superior se obtienen al cambiar el valor de n en el código. Para obtener las **fórmulas de Adams-Moulton**, iniciamos con una regla de cuadratura de la forma

$$\int_0^1 G(r) dr \approx \sum_{j=1}^n C_j G(2-j)$$

Un programa similar al anterior produce los coeficientes $(\frac{9}{24}, \frac{19}{24}, -\frac{5}{24}, \frac{1}{24})$. La diferencia entre las dos reglas de cuadratura es que una implica el valor del integrando en 1 y la otra no.

¿Cómo se llega a las fórmulas para $\int_t^{t+h} g(s) ds$ del trabajo ya hecho? Utilizando el cambio de variable de s a σ dado por $s = h\sigma - t$. Con estas consideraciones, se piensa t como una constante. La nueva integral será $h \int_0^1 g(h\sigma + t) d\sigma$, que se puede tratar con cualquiera de las dos fórmulas ya diseñadas para el intervalo $[0, 1]$. Por ejemplo,

$$\begin{aligned} \int_t^{t+h} F(r) dr &\approx \frac{h}{24} [55F(t) - 59F(t-h) + 37F(t-2h) - 9F(t-3h)] \\ \int_t^{t+h} G(r) dr &\approx \frac{h}{24} [9G(t+h) + 19G(t) - 5G(t-h) + G(t-2h)] \end{aligned}$$

El método de los coeficientes indeterminados que se uso aquí para obtener las fórmulas de cuadratura, por sí mismo, no proporciona los términos error que nos gustaría tener. Se puede hacer una evaluación de los errores de la teoría de interpolación, ya que los métodos considerados en este caso provienen de la integración de un polinomio de interpolación. Los detalles se pueden encontrar en libros más avanzados. Puede experimentar con algunas de las fórmulas de Adams-Bashforth-Moulton en los problemas de cómputo 10.3.2–10.3.4. Estos métodos se consideran de nuevo en la sección 11.3.

Análisis de estabilidad

Vamos ahora a resumir el análisis de los errores que inevitablemente se producen en la solución numérica de un problema con valor inicial

$$\begin{cases} x' = f(t, x) \\ x(a) = s \end{cases} \tag{2}$$

La solución exacta es una función $x(t)$. Depende del valor inicial s , y para demostrar esto, escribimos $x(t, s)$. La ecuación diferencial por tanto da lugar a una familia de curvas solución, cada una de las cuales corresponde a un valor del parámetro s . Por ejemplo, la ecuación diferencial

$$\begin{cases} x' = x \\ x(a) = s \end{cases}$$

da lugar a la familia de curvas solución $x = se^{(t-a)}$ que difieren en sus valores iniciales $x(a) = s$. En la figura 10.4 se muestran algunas de estas curvas. El hecho de que las curvas divergen unas de otras conforme t aumenta tiene un significado numérico importante. Supongamos, por ejemplo, que el valor inicial s se lee en la computadora con algún error de redondeo. Entonces, aun si todos los cálculos posteriores son precisos y *no ocurren errores de truncamiento*, la solución estará equivocada. Un error cometido al principio tiene el efecto de seleccionar la *curva equivocada* de la familia de todas las curvas de solución. Puesto que estas curvas divergen entre sí, cualquier pequeño error al inicio es responsable de una eventual pérdida total de precisión. Este fenómeno no se limita a los errores cometidos en el primer paso, porque cada punto en la solución numérica se puede interpretar como el valor inicial para los futuros puntos.

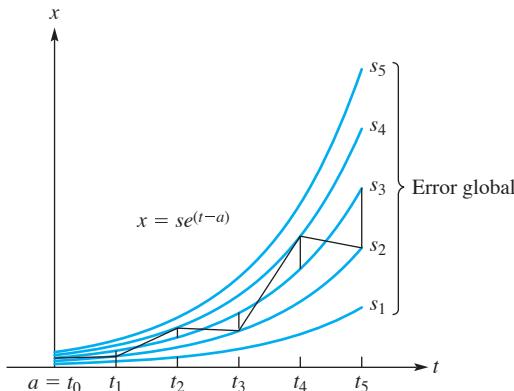


FIGURA 10.4
Curvas solución
a $x' = x$ con
 $x(a) = s$

Para un ejemplo en el que no surge esta dificultad, considere

$$\begin{cases} x' = -x \\ x(a) = s \end{cases}$$

Sus soluciones son $x = se^{-(t-a)}$. Conforme t aumenta, estas curvas se acercan más entre sí, como se muestra en la figura 10.5. Así, los errores cometidos en la solución numérica aún dan como resultado la selección de la curva equivocada, pero el efecto no es tan grave porque las curvas se unen.

En un paso dado, el error global de una solución aproximada de una ecuación diferencial ordinaria contiene tanto el error local en ese paso como el efecto acumulativo de todos los errores locales de todos los pasos anteriores. Para las curvas solución divergentes, los errores locales en cada paso se magnifican con el tiempo y el error global puede ser mayor que la suma de todos los errores locales. En las figuras 10.4 y 10.5 se indican los pasos de la solución numérica con puntos conectados por líneas oscuras. También, los errores locales se indican mediante pequeñas barras verticales y el error global con una barra vertical en el extremo derecho de las curvas.

Para las curvas solución convergentes, los errores locales en cada paso se reducen con el tiempo y el error global puede ser menor que la suma de todos los errores locales. Para la ecuación

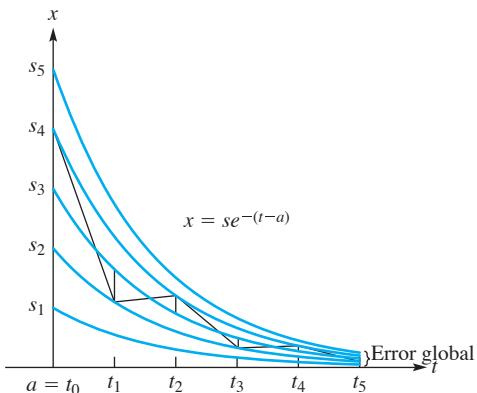


FIGURA 10.5
Curvas solución para $x' = -x$ con $x(a) = s$

diferencial general (2), ¿cómo pueden distinguirse los dos modos de comportamiento que acabamos de analizar? Es simple. Si $f_x > \delta$ para alguna δ positiva, las curvas **divergen**. Sin embargo, si $f_x < -\delta$, **convergen**. Para ver por qué, considere dos curvas solución cercanas que corresponden a valores iniciales s y $s + h$. Por la serie de Taylor, tenemos

$$x(t, s + h) = x(t, s) + h \frac{\partial}{\partial s} x(t, s) + \frac{1}{2} h^2 \frac{\partial^2}{\partial s^2} x(t, s) + \dots$$

donde

$$x(t, s + h) - x(t, s) \approx h \frac{\partial}{\partial s} x(t, s)$$

Así, la divergencia de las curvas significa que

$$\lim_{t \rightarrow \infty} |x(t, s + h) - x(t, s)| = \infty$$

y se puede escribir como

$$\lim_{t \rightarrow \infty} \left| \frac{\partial}{\partial s} x(t, s) \right| = \infty$$

Para calcular esta derivada parcial, comencemos con la ecuación diferencial satisfecha por $x(t, s)$:

$$\frac{\partial}{\partial t} x(t, s) = f(t, x(t, s))$$

y derivamos parcialmente con respecto a s :

$$\frac{\partial}{\partial s} \frac{\partial}{\partial t} x(t, s) = \frac{\partial}{\partial s} f(t, x(t, s))$$

Por tanto,

$$\frac{\partial}{\partial t} \frac{\partial}{\partial s} x(t, s) = f_x(t, x(t, s)) \frac{\partial}{\partial s} x(t, s) + f_t(t, x(t, s)) \frac{\partial t}{\partial s} \quad (3)$$

Pero s y t son variables independientes (un cambio en s no produce ningún cambio en t), de modo que $\partial t / \partial s = 0$. Si s está ahora fija y si hacemos $u(t) = (\partial / \partial s)x(t, s)$ y $q(t) = f_x(t, x(t, s))$, entonces la ecuación (3) se convierte en

$$u' = qu \quad (4)$$

Esta es una ecuación diferencial lineal con solución $u(t) = ce^{Q(t)}$, donde Q es la integral indefinida (primitiva) de q . La condición $\lim_{t \rightarrow \infty} |u(t)| = \infty$ se cumple si $\lim_{t \rightarrow \infty} Q(t) = \infty$.

Esta situación, a su vez, se produce si $q(t)$ es positiva y su límite está lejos de cero, porque entonces

$$Q(t) = \int_a^t q(\theta) d\theta > \int_a^t \delta d\theta = \delta(t - a) \rightarrow \infty$$

cuando $t \rightarrow \infty$ si $f_x = q = d > 0$.

Para ilustrar, considere la ecuación diferencial $x' = t + \tan x$. Las curvas solución divergen entre sí cuando $t \rightarrow \infty$ porque $f_x(t, x) = \sec^2 x > 1$.

Resumen

(1) El método de Runge-Kutta-Fehlberg es

$$\begin{aligned}\tilde{x}(t) &= x(t) + \frac{25}{216}K_1 + \frac{1408}{2565}K_3 + \frac{2197}{4104}K_4 - \frac{1}{5}K_5 \\ x(t+h) &= x(t) + \frac{16}{135}K_1 + \frac{6656}{12825}K_3 + \frac{28561}{56430}K_4 - \frac{9}{50}K_5 + \frac{2}{55}K_6\end{aligned}$$

donde

$$\left\{ \begin{array}{l} K_1 = hf(t, x) \\ K_2 = hf\left(t + \frac{1}{4}h, x + \frac{1}{4}K_1\right) \\ K_3 = hf\left(t + \frac{3}{8}h, x + \frac{3}{32}K_1 + \frac{9}{32}K_2\right) \\ K_4 = hf\left(t + \frac{12}{13}h, x + \frac{1932}{2197}K_1 - \frac{7200}{2197}K_2 + \frac{7296}{2197}K_3\right) \\ K_5 = hf\left(t + h, x + \frac{439}{216}K_1 - 8K_2 + \frac{3680}{513}K_3 - \frac{845}{4104}K_4\right) \\ K_6 = hf\left(t + \frac{1}{2}h, x - \frac{8}{27}K_1 + 2K_2 - \frac{3544}{2565}K_3 + \frac{1859}{4104}K_4 - \frac{11}{40}K_5\right) \end{array} \right.$$

La cantidad $\varepsilon = |x(t+h) - \tilde{x}|$ se puede utilizar en un tamaño de paso del procedimiento adaptado.

(2) Un método de múltiples pasos de cuarto orden es el método de Adams-Bashforth-Moulton:

$$\begin{aligned}\tilde{x}(t+h) &= x(t) + \frac{h}{24} [55f(t, x(t)) - 59f(t-h, x(t-h)) \\ &\quad + 37f(t-2h, x(t-2h)) - 9f(t-3h, x(t-3h))] \\ x(t+h) &= x(t) + \frac{h}{24} [9f(t+h, \tilde{x}(t+h)) + 19f(t, x(t)) \\ &\quad - 5f(t-h, x(t-h)) + f(t-2h, x(t-2h))]\end{aligned}$$

El valor $\tilde{x}(t+h)$ es el **valor predicho** y $x(t+h)$ es el **valor corregido**. Los errores de truncamiento de estas dos fórmulas son $\mathcal{O}(h^5)$. Puesto que el valor de $x(a)$ está dado, los valores de $x(a+h)$, $x(a+2h)$, $x(a+3h)$, $x(a+4h)$ se calculan con algunos métodos de un solo paso como el método de Runge-Kutta de cuarto orden.

Referencias adicionales

Véase Aiken [1985], Butcher [1987], Dekker y Verwer [1984], England [1969], Fehlberg [1969], Henrici [1962], Hundsdorfer [1985], Lambert [1973], Lapidus y Seinfeld [1971], Miranker [1981], Moulton [1930], Shampine y Gordon [1975] y Stetter [1973].

Problemas 10.3

- ^a1. Resuelva el problema

$$\begin{cases} x' = -x \\ x(0) = 1 \end{cases}$$

usando la regla del trapecio, como se analizó al comienzo de este capítulo. Compare la solución verdadera en $t = 1$ con la solución aproximada obtenida con n pasos. Muestre, por ejemplo, que para $n = 5$, el error es de 0.00123.

- ^a2. Deduzca una fórmula de pasos múltiples implícita basada en la regla de Simpson, que implique puntos espaciados uniformemente $x(t-h), x(t), x(t+h)$, para resolver numéricamente la ecuación diferencial ordinaria $x' = f$.
3. Un estudiante atento se da cuenta de que los coeficientes de la fórmula de Adams-Bashforth suman 1. ¿Por qué es así?

- ^a4. Deduzca una fórmula de la forma

$$x(t+h) = ax(t) + bx(t-h) + h[cx'(t+h) + dx''(t) + ex'''(t-h)]$$

que es exacta para polinomios de un grado tan alto como sea posible. *Sugerencia:* utilice polinomios $1, t, t^2$ y así sucesivamente.

- ^a5. Determine los coeficientes de un método de EDO implícito y de un solo paso de la forma

$$x(t+h) = ax(t) + bx'(t) + cx'(t+h)$$

de modo que sea exacta para polinomios de un grado tan alto como sea posible. ¿Cuál es el orden del término de error?

6. La ecuación diferencial que se utiliza para ilustrar el programa del método de Runge–Kutta adaptado se puede resolver con un factor de integración. Hágalo.
7. Establezca la ecuación (4).

- ^a8. El problema con valor inicial $x' = (1+t^2)x$, con $x(0) = 1$ debe resolverse en el intervalo $[0, 9]$. ¿Cuán sensible es $x(9)$ a las perturbaciones del valor inicial $x(0)$?

9. Para cada ecuación diferencial, determine las regiones en las que las curvas solución tienden a separarse unas de otras cuando t aumenta:

^aa. $x' = \sin t + e^x$

b. $x' = x + te^{-t}$

^ac. $x' = xt$

d. $x' = x^3(t^2 + 1)$

^ae. $x' = \cos t - e^x$

f. $x' = (1-x^3)(1+t^2)$

- “10.** Para la ecuación diferencial $x' = t(x^3 - 6x^2 + 15x)$, determine si las curvas solución divergen entre sí cuando $t \rightarrow \infty$.
- “11.** Determine si las curvas solución de $x' = (1 + t^2)^{-1}x$ divergen entre sí cuando $t \rightarrow \infty$.

Problemas de cómputo 10.3

- 1.** Use software matemático para resolver sistemas de ecuaciones lineales cuyas soluciones son
- a.** coeficientes de Adams-Bashforth **b.** coeficientes de Adams-Moulton

- 2.** El **método de Adams-Bashforth-Moulton** de segundo orden está dado por

$$\begin{aligned}\tilde{x}(t+h) &= x(t) + \frac{h}{2}[3f(t, x(t)) - f(t-h, x(t-h))] \\ x(t+h) &= x(t) + \frac{h}{2}[f(t+h, \tilde{x}(t+h)) + f(t, x(t))]\end{aligned}$$

El error aproximado de un solo paso es $\varepsilon \equiv K|x(t+h) - \tilde{x}(t+h)|$, donde $K = \frac{1}{6}$. Usando ε para vigilar la convergencia, escriba y pruebe un procedimiento adaptado para resolver una EDO de su elección usando estas fórmulas.

- 3.** (Continuación) Realice las instrucciones del problema de cómputo anterior para el **método de Adams-Bashforth-Moulton de tercer orden**:

$$\begin{aligned}\tilde{x}(t+h) &= x(t) + \frac{h}{12}[23f(t, x(t)) - 16f(t-h, x(t-h)) \\ &\quad + 5f(t-2h, x(t-2h))] \\ x(t+h) &= x(t) + \frac{h}{12}[5f(t+h, \tilde{x}(t+h)) + 8f(t, x(t)) \\ &\quad - f(t-h, x(t-h))]\end{aligned}$$

donde $K = \frac{1}{10}$ en la expresión para el error aproximado de un solo paso.

- 4.** (Esquema predictor-corrector) Usando el **método de Adams-Bashforth-Moulton de cuarto orden**, deduzca el esquema predictor-corrector dado por las ecuaciones siguientes:

$$\begin{aligned}\tilde{x}(t+h) &= x(t) + \frac{h}{24}[55f(t, x(t)) - 59f(t-h, x(t-h)) \\ &\quad + 37f(t-2h, x(t-2h)) - 9f(t-3h, x(t-3h))] \\ x(t+h) &= x(t) + \frac{h}{24}[9f(t+h, \tilde{x}(t+h)) + 19f(t, x(t)) \\ &\quad - 5f(t-h, x(t-h)) + f(t-2h, x(t-2h))]\end{aligned}$$

Escriba y pruebe un procedimiento para el método de Adams-Bashforth-Moulton. Nota: este es un proceso de múltiples pasos, porque se usan los valores de x en $t, t-h, t-2h$ y $t-3h$ para determinar el valor *predicho* $\tilde{x}(t+h)$, que, a su vez, se utiliza con los valores de x en $t, t-h$ y $t-2h$, para obtener el valor *corregido* $x(t+h)$. Los términos de error para estas fórmulas son $(251/720)h^5 f^{(4)}(\xi)$ y $-(19/720)h^5 f^{(4)}(\eta)$, respectivamente. (Véase la sección 9.3 para más información sobre estos métodos.)

"5. Resuelva

$$\begin{cases} x' = \frac{3x}{t} + \frac{9}{2}t - 13 \\ x(3) = 6 \end{cases}$$

en $x\left(\frac{1}{2}\right)$ usando el procedimiento *Adaptativo_RK45* para obtener la solución deseada con nueve decimales. Compare con la solución verdadera:

$$x = t^3 - \frac{9}{2}t^2 + \frac{13}{2}t$$

"6. (Continuación) Repita el problema anterior para $x\left(-\frac{1}{2}\right)$.

- 7.** Se sabe que el método de Runge-Kutta de cuarto orden descrito en la ecuación (12) de la sección 10.2 tiene un error de truncamiento local que es $\mathcal{O}(h^5)$. Diseñe y lleve a cabo un experimento numérico para probar esto. *Sugerencias:* tome sólo un paso en la solución numérica de una ecuación diferencial no trivial cuya solución es conocida de antemano. Sin embargo, use diferentes valores para h , como 2^{-n} , donde $1 \leq n \leq 24$. Pruebe si el cociente de errores para h^5 permanece acotado cuando $h \rightarrow 0$. Puede ser necesario un cálculo de precisión múltiple. Imprima los cocientes indicados.

- 8.** Calcule la solución numérica de

$$\begin{cases} x' = -x \\ x(0) = 1 \end{cases}$$

usando el **método del punto medio**

$$x_{n+1} = x_{n-1} + 2hx_n'$$

con $x_0 = 1$ y $x_1 = -h + \sqrt{1 + h^2}$. ¿Existen dificultades al usar este método para este problema? Realice un análisis de estabilidad de este método. *Sugerencia:* considere h fijo y suponga $x_n = \lambda^n$.

- 9.** Tabule y trace la gráfica de la función $[1 - \ln v(x)]v(x)$ en $[0, e]$, donde $v(x)$ es la solución del problema con valor inicial $(dv/dx)[\ln v(x)] = 2x$, $v(0) = 1$. *Valor de comprobación:* $v(1) = e$.

- 10.** Determine el valor numérico de

$$2\pi \int_4^5 \frac{e^s}{s} ds$$

de tres maneras: resolviendo la integral, una ecuación diferencial ordinaria y usando la fórmula exacta.

- 11.** Calcule e imprima una tabla de la función

$$f(\phi) = \int_0^\phi \sqrt{1 - \frac{1}{4} \sin^2 \theta} d\theta$$

resolviendo un adecuado problema con valor inicial. Cubra el intervalo $[0, 90^\circ]$, con pasos de 1° y use el método de Runge-Kutta de orden 4. *Valores de comprobación:* use $f(30^\circ) = 0.51788193$ y $f(90^\circ) = 1.46746221$. *Nota:* este es un ejemplo de una integral elíptica de segunda clase. Surge en la búsqueda de la longitud de arco de una elipse y en muchos problemas de ingeniería.

“12. Resolviendo un adecuado problema con valor inicial, haga una tabla de la función

$$f(x) = \int_{1/x}^{\infty} \frac{dt}{te^t}$$

en el intervalo $[0, 1]$. Determine cuán bien f se aproxima mediante $xe^{-1/x}$. Sugerencia: sea $t = -\ln s$.

“13. Resolviendo un adecuado problema con valor inicial, haga una tabla de la función

$$f(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

en el intervalo $0 \leq x \leq 2$. Determine con qué precisión es $f(x)$ aproximada en este intervalo mediante la función

$$g(x) = 1 - (ay + by^2 + cy^3) \frac{2}{\sqrt{\pi}} e^{-x^2}$$

donde

$$\begin{cases} a = 0.3084284 & b = -0.0849713 \\ c = 0.6627698 & y = (1 + 0.47047x)^{-1} \end{cases}$$

14. Utilice el método de Runge-Kutta para calcular $\int_0^1 \sqrt{1 + s^3} ds$.

“15. Escriba y ejecute un programa para imprimir una tabla exacta de la **integral seno**

$$\text{Si}(x) = \int_0^x \frac{\sin r}{r} dr$$

La tabla debe cubrir el intervalo $0 \leq x \leq 1$ en pasos de tamaño 0.01. [Use $\sin(0)/0 = 1$. Véase el problema de cómputo 5.1.2]

16. Calcule una tabla de la función

$$\text{Shi}(x) = \int_0^x \frac{\sinh t}{t} dt$$

hallando un problema con valor inicial que esta cumpla y después resolviendo el problema con valor inicial. Su tabla debe tener precisión cercana a la precisión de la máquina. [Use $\sinh(0)/0 = 1$].

17. Diseñe y realice un experimento numérico para comprobar que una pequeña perturbación en un problema con valor inicial puede causar errores catastróficos en la solución numérica. *Nota:* un **problema con valor inicial** es una ecuación diferencial ordinaria con condiciones dadas sólo en el punto inicial. (Compare esto con el *problema con valor en la frontera* que se presenta en el capítulo 12.)

18. Ejecute los programas de ejemplo para resolver la ecuación (1) del ejemplo industrial, compare las soluciones y construya las gráficas.

19. England [1969] desarrolló otro método adaptado de Runge-Kutta. El **método de Runge-Kutta-England** es similar al método de Runge-Kutta-Fehlberg en que combina una fórmula de Runge-Kutta de cuarto orden y una de quinto orden. Para reducir el número de evaluaciones de la función, las fórmulas se deducen de manera que algunas de las evaluaciones de la misma función se utilizan en cada par de fórmulas. (Una fórmula de Runge-Kutta de cuarto orden re-

quiere al menos de cuatro evaluaciones de la función y uno de quinto orden requiere al menos seis). El método de Runge-Kutta-England utiliza los métodos de Runge-Kutta de cuarto orden en el problema de cómputo 10.2.14a y toma los dos medios pasos de la siguiente manera:

$$\left(t + \frac{1}{2}h \right) = x(t) + \frac{1}{6}(K_1 + 4K_3 + K_4)$$

donde

$$\begin{cases} K_1 = \frac{1}{2}hf(t, x(t)) \\ K_2 = \frac{1}{2}hf\left(t + \frac{1}{4}h, x(t) + \frac{1}{2}K_1\right) \\ K_3 = \frac{1}{2}hf\left(t + \frac{1}{4}h, x(t) + \frac{1}{4}K_1 + \frac{1}{4}K_2\right) \\ K_4 = \frac{1}{2}hf\left(t + \frac{1}{2}h, x(t) - K_2 + 2K_3\right) \end{cases}$$

y

$$x(t+h) = x\left(t + \frac{1}{2}h\right) + \frac{1}{6}(K_5 + 4K_7 + K_8)$$

donde

$$\begin{cases} K_5 = \frac{1}{2}hf\left(t + \frac{1}{2}h, x\left(t + \frac{1}{2}h\right)\right) \\ K_6 = \frac{1}{2}hf\left(t + \frac{3}{4}h, x\left(t + \frac{1}{2}h\right) + \frac{1}{2}K_5\right) \\ K_7 = \frac{1}{2}hf\left(t + \frac{3}{4}h, x\left(t + \frac{1}{2}h\right) + \frac{1}{4}K_5 + \frac{1}{4}K_6\right) \\ K_8 = \frac{1}{2}hf\left(t + h, x\left(t + \frac{1}{2}h\right) - K_6 + 2K_7\right) \end{cases}$$

Con estos dos medios pasos, hay evaluaciones suficientes de la función de modo que sólo se necesita una más

$$K_9 = \frac{1}{2}hf\left(t + h, x(t) - \frac{1}{12}(K_1 + 96K_2 - 92K_3 + 121K_4 - 144K_5 - 6K_6 + 12K_7)\right)$$

para obtener un método Runge-Kutta de quinto orden:

$$\hat{x}(t+h) = x(t) + \frac{1}{90}(14K_1 + 64K_3 + 32K_5 - 8K_7 + 64K_9 + 15K_8 - K_9)$$

Se puede desarrollar un procedimiento adaptado usando una estimación de error basada en los dos valores $x(t+h)$ y $\hat{x}(t+h)$. Programe y pruebe dicho procedimiento. (Véase, por ejemplo, Shampine, Allen y Pruess [1997].)

- 20.** Investigue la solución numérica del problema con valor inicial

$$\begin{cases} x' = -\sqrt{1-x^2} \\ x(0) = 1 \end{cases}$$

Este problema está mal condicionado, puesto que $x(t) = \cos t$ es una solución y $x(t) = 1$ también lo es. Para obtener más información acerca de este y otros problemas de prueba, véase Cash [2003] o www.ma.ic.ac.uk/~jcash/.

- 21. (Proyecto de investigación estudiantil)** Aprenda acerca de ecuaciones diferenciales algebraicas.

- 22.** Escriba un software para implementar los seudocódigos siguientes y compruebe los resultados numéricos dados en el libro:

- a. *Test_RK45* y *RK45* b. *Prueba_Adapatativa_RK45* y *Adapatativa_RK45*

Sistemas de ecuaciones diferenciales ordinarias

Un modelo simple para explicar cómo dos especies animales a veces interactúan es el *modelo depredador-presa*. Si $u(t)$ es el número de individuos de las especies de depredadores y $v(t)$ el número de individuos en las especies presa, entonces, bajo suposiciones adecuadas de simplificación y con constantes apropiadas a , b , c y d ,

$$\begin{cases} \frac{du}{dt} = a(v + b)u \\ \frac{dv}{dt} = c(u + d)v \end{cases}$$

Este es un par de ecuaciones diferenciales ordinarias (EDO) no lineales que regulan las poblaciones de las dos especies (en función del tiempo t). En este capítulo se desarrollan procedimientos numéricos para resolver problemas de este tipo.

11.1 Métodos para sistemas de primer orden

En el capítulo 10 se consideraron ecuaciones diferenciales ordinarias en el contexto más simple, es decir, limitamos nuestra atención a una sola ecuación diferencial de primer orden acompañada de una condición auxiliar. Sin embargo, los problemas científicos y tecnológicos conducen a menudo a situaciones más complicadas. El siguiente grado de complicación ocurre con *sistemas* de varias ecuaciones de primer orden.

Sistemas desacoplados y acoplados

El Sol y los nueve planetas forman un sistema de *partículas* que se mueven bajo la jurisdicción de la ley de la gravitación de Newton. Los vectores de posición de los planetas constituyen un sistema de 27 funciones y las leyes de Newton del movimiento se pueden escribir, entonces, como un sistema de 54 ecuaciones diferenciales ordinarias de primer orden. En principio, las posiciones pasadas y futuras de los planetas se pueden obtener mediante la solución numérica de estas ecuaciones.

Tomando un ejemplo de alcance más modesto, se consideran dos ecuaciones con dos condiciones auxiliares. Sean x e y dos funciones de t sujetas al sistema

$$\begin{cases} x'(t) = x(t) - y(t) + 2t - t^2 - t^3 \\ y'(t) = x(t) + y(t) - 4t^2 + t^3 \end{cases} \quad (1)$$

con condiciones iniciales

$$\begin{cases} x(0) = 1 \\ y(0) = 0 \end{cases}$$

Este es un ejemplo de un problema con valor inicial que implica un sistema de dos ecuaciones diferenciales de primer orden. Observe que en el ejemplo dado no es posible resolver ninguna de las dos ecuaciones diferenciales por sí misma porque la primera ecuación que gobierna a x' implica la función desconocida y y la segunda ecuación que gobierna a y' implica la función desconocida x . En esta situación, decimos que las dos ecuaciones diferenciales están **acopladas**.

Se le invita a usted a comprobar que la solución analítica es

$$\begin{cases} x(t) = e^t \cos(t) + t^2 = \cos(t)[\cosh(t) + \operatorname{senh}(t)] + t^2 \\ y(t) = e^t \operatorname{sen}(t) - t^3 = \operatorname{sen}(t)[\cosh(t) + \operatorname{senh}(t)] - t^3 \end{cases}$$

Veamos otro ejemplo que es superficialmente similar al primero pero en realidad es más simple:

$$\begin{cases} x'(t) = x(t) + 2t - t^2 - t^3 \\ y'(t) = y(t) - 4t^2 + t^3 \end{cases} \quad (2)$$

con condiciones iniciales

$$\begin{cases} x(0) = 1 \\ y(0) = 0 \end{cases}$$

Estas dos ecuaciones *no* están acopladas y se pueden resolver por separado como dos problemas con valor inicial independientes (usando, por ejemplo, los métodos del capítulo 10). Naturalmente, nuestra preocupación en este caso es con los sistemas que están acoplados, aunque los métodos que resuelven los sistemas acoplados también sirven para los que no lo son. Los procedimientos analizados en el capítulo 10 se extienden a los sistemas acoplados o desacoplados.

Método de la serie de Taylor

Se ilustra el método de la serie de Taylor para el sistema (1), que comienza por derivar las ecuaciones que lo constituyen:

$$\begin{aligned} & \begin{cases} x' = x - y + 2t - t^2 - t^3 \\ y' = x + y - 4t^2 + t^3 \end{cases} \\ & \begin{cases} x'' = x' - y' + 2 - 2t - 3t^2 \\ y'' = x' + y' - 8t + 3t^2 \end{cases} \\ & \begin{cases} x''' = x'' - y'' - 2 - 6t \\ y''' = x'' + y'' - 8 + 6t \end{cases} \\ & \begin{cases} x^{(4)} = x''' - y''' - 6 \\ y^{(4)} = x''' + y''' + 6 \end{cases} \\ & \text{etc.} \end{aligned}$$

Un programa para pasar de $x(t)$ a $x(t + h)$ y de $y(t)$ a $y(t + h)$ se escribe fácilmente usando algunos términos de la serie de Taylor:

$$\begin{aligned}x(t + h) &= x + hx' + \frac{h^2}{2}x'' + \frac{h^3}{6}x''' + \frac{h^4}{24}x^{(4)} + \dots \\y(t + h) &= y + hy' + \frac{h^2}{2}y'' + \frac{h^3}{6}y''' + \frac{h^4}{24}y^{(4)} + \dots\end{aligned}$$

junto con las ecuaciones para las diferentes derivadas. En este caso, x y y y todas sus derivadas son funciones de t , es decir, $x = x(t)$, $y = y(t)$, $x' = x'(t)$, $y' = y'(t)$ y así sucesivamente.

Un pseudocódigo que genera e imprime una solución numérica de 0 a 1 en 100 pasos es el siguiente. Se han utilizado términos hasta h^4 en la serie de Taylor.

```
program Sistema_TaylorI
integer k; real h, t, x, y, x', y', x'', y'', x''', y''', x(4), y(4)
integer nsteps ← 100; real a ← 0, b ← 1
x ← 1; y ← 0; t ← a
output 0, t, x, y
h ← (b - a)/nsteps
for k = 1 to nsteps do
    x' ← x - y + t(2 - t(1 + t))
    y' ← x + y + t2(-4 + t)
    x'' ← x' - y' + 2 - t(2 + 3t)
    y'' ← x' + y' + t(-8 + 3t)
    x''' ← x'' - y'' - 2 - 6t
    y''' ← x'' + y'' - 8 + 6t
    x(4) ← x''' - y''' - 6
    y(4) ← x''' + y''' + 6
    x ← x + h [x' +  $\frac{1}{2}h$  [x'' +  $\frac{1}{3}h$  [x''' +  $\frac{1}{4}h$  [x(4)]]]]
    y ← y + h [y' +  $\frac{1}{2}h$  [y'' +  $\frac{1}{3}h$  [y''' +  $\frac{1}{4}h$  [y(4)]]]]
    t ← t + h
    output k, t, x, y
end for
end program Sistema_TaylorI
```

Notación vectorial

Observe que el sistema (1) se puede escribir en notación vectorial como

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x - y + 2t - t^2 - t^3 \\ x + y - 4t^2 + t^3 \end{bmatrix} \quad (3)$$

con condiciones iniciales

$$\begin{bmatrix} x(0) \\ y(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Este es un caso especial de un problema más general que se puede escribir como

$$\begin{cases} X' = \mathbf{F}(t, X) \\ X(a) = \mathbf{S}, \text{ dada} \end{cases} \quad (4)$$

donde

$$\mathbf{X} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \mathbf{X}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

y \mathbf{F} es el vector cuyas dos componentes están dadas por los miembros derechos en la ecuación (1). Puesto que \mathbf{F} depende de t y de X , se escribe $\mathbf{F}(t, X)$.

Sistemas de EDO

Podemos continuar con esta idea con el propósito de manejar un sistema de n ecuaciones diferenciales de primer orden. Primero, se escribe como

$$\begin{cases} x_1' = f_1(t, x_1, x_2, \dots, x_n) \\ x_2' = f_2(t, x_1, x_2, \dots, x_n) \\ \vdots \\ x_n' = f_n(t, x_1, x_2, \dots, x_n) \\ x_1(a) = s_1, x_2(a) = s_2, \dots, x_n(a) = s_n \quad \text{todas dadas} \end{cases}$$

Entonces hacemos

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{X}' = \begin{bmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$$

y obtenemos la ecuación (4), que es una ecuación diferencial ordinaria escrita en notación vectorial.

Método de series de Taylor: notación vectorial

El **método de la serie de Taylor de orden m** debería escribirse como

$$\mathbf{X}(t + h) = \mathbf{X} + h\mathbf{X}' + \frac{h^2}{2}\mathbf{X}'' + \cdots + \frac{h^m}{m!}\mathbf{X}^{(m)} \quad (5)$$

donde $\mathbf{X} = \mathbf{X}(t)$, $\mathbf{X}' = \mathbf{X}'(t)$, $\mathbf{X}'' = \mathbf{X}''(t)$ y así sucesivamente.

Un seudocódigo del método de la serie de Taylor de orden 4 aplicado al problema anterior se puede reescribir fácilmente con un simple cambio de variables y la introducción de un arreglo y de un ciclo interno.

```
program Sistema_Taylor2
integer i, k; real h, t; real array (x_i)_{1:n}, (d_{ij})_{1:n \times 1:4}
integer n ← 2, nsteps ← 100
real a ← 0, b ← 1
t ← 0; (x_i) ← (1, 0)
output 0, t, (x_i)
h ← (b - a)/nsteps
```

```

for  $k = 1$  to  $nsteps$  do
     $d_{11} \leftarrow x_1 - x_2 + t(2 - t(1 + t))$ 
     $d_{21} \leftarrow x_1 + x_2 + t^2(-4 + t)$ 
     $d_{12} \leftarrow d_{11} - d_{21} + 2 - t(2 + 3t)$ 
     $d_{22} \leftarrow d_{11} + d_{21} + t(-8 + 3t)$ 
     $d_{13} \leftarrow d_{12} - d_{22} - 2 - 6t$ 
     $d_{23} \leftarrow d_{12} + d_{22} - 8 + 6t$ 
     $d_{14} \leftarrow d_{13} - d_{23} - 6$ 
     $d_{24} \leftarrow d_{13} + d_{23} + 6$ 
    for  $i = 1$  to  $n$  do
         $x_i \leftarrow x_i + h [d_{i1} + \frac{1}{2}h [d_{i2} + \frac{1}{3}h [d_{i3} + \frac{1}{4}h [d_{i4}]]]]$ 
    end for
     $t \leftarrow t + h$ 
    output  $k, t, (x_i)$ 
end for
end program Sistema_Taylor2

```

Aquí se utiliza un arreglo bidimensional en lugar de todas las derivadas de las diferentes variables, es decir, $d_{ij} \leftrightarrow x_i^{(j)}$. De hecho, este y otros métodos en este capítulo se vuelven particularmente fáciles de programar si el lenguaje de programación soporta operaciones vectoriales.

Método de Runge-Kutta

Los métodos de Runge-Kutta del capítulo 10 también se extienden a sistemas de ecuaciones diferenciales. El método clásico de Runge-Kutta de cuarto orden para el sistema (4) utiliza estas fórmulas:

$$\mathbf{X}(t + h) = \mathbf{X} + \frac{h}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4) \quad (6)$$

donde

$$\begin{cases} \mathbf{K}_1 = \mathbf{F}(t, \mathbf{X}) \\ \mathbf{K}_2 = \mathbf{F}\left(t + \frac{1}{2}h, \mathbf{x} + \frac{1}{2}h\mathbf{k}_1\right) \\ \mathbf{K}_3 = \mathbf{F}\left(t + \frac{1}{2}h, \mathbf{x} + \frac{1}{2}h\mathbf{k}_2\right) \\ \mathbf{K}_4 = \mathbf{F}(t + h, \mathbf{X} + h\mathbf{K}_3) \end{cases}$$

En este caso, $\mathbf{X} = \mathbf{X}(t)$ y todas las cantidades son vectores con n componentes, excepto las variables t y h .

A continuación se presenta un procedimiento para realizar el procedimiento de Runge-Kutta. Se supone que el sistema que hay que resolver está en la forma de la ecuación (4) y que hay n ecuaciones en el sistema. El usuario proporciona el valor inicial de t , el valor inicial de \mathbf{X} , el tamaño de paso h y el número de pasos por seguir, $nsteps$. Además, se necesita el procedimiento XP_System ($n, t, (\mathbf{x}_i), (f_i)$), que evalúa el miembro derecho de la ecuación (4) para un valor determinado del arreglo (\mathbf{x}_i) y almacena el resultado en el arreglo (f_i) . (El nombre $XP_System2$ se elige como una abreviatura de X' para un sistema.)

```

procedure Sistema1_RK4(n, h, t, (xi), nsteps)
integer i, j, n; real h, t; real array (xi)1:n
allocate real array (yi)1:n, (Ki,j)1:n×1:4
output 0, t, (xi)
for j = 1 to nsteps do
    call XP_System(n, t, (xi), (Ki,1))
    for i = 1 to n do
        yi  $\leftarrow$  xi +  $\frac{1}{2}hK_{i,1}$ 
    end for
    call XP_System(n, t + h / 2, (yi), (Ki,2))
    for i = 1 to n do
        yi  $\leftarrow$  xi +  $\frac{1}{2}hK_{i,2}$ 
    end for
    call XP_System(n, t + h / 2, (yi), (Ki,3))
    for i = 1 to n do
        yi  $\leftarrow$  xi +  $hK_{i,3}$ 
    end for
    call XP_System(n, t + h, (yi), (Ki,4))
    for i = 1 to n do
        xi  $\leftarrow$  xi +  $\frac{1}{6}h[K_{i,1} + 2K_{i,2} + 2K_{i,3} + K_{i,4}]$ 
    end for
    t  $\leftarrow$  t + h
    output j, t, (xi)
end for
deallocate array (yi), (Ki,j)
end procedure Sistema1_RK4

```

Para ilustrar el uso de este procedimiento empleamos una vez más el sistema (1) para nuestro ejemplo. Por supuesto, se debe reescribir en la forma de la ecuación (4). Un programa principal adecuado y un procedimiento para calcular el miembro derecho de la ecuación (4) es el siguiente:

```

program Prueba_Sistema1_RK4
integer n  $\leftarrow$  2, nsteps  $\leftarrow$  100
real a  $\leftarrow$  0, b  $\leftarrow$  1
real h, t; real array (xi)1:n
t  $\leftarrow$  0
(xi)  $\leftarrow$  (1, 0)
h  $\leftarrow$  (b - a)/nsteps
call RK4_System1(n, h, t, (xi), nsteps)
end program Test_RK4_System1

procedure XP_System(n, t, (xi), (fi))
real array (xi)1:n, (fi)1:n
integer n

```

```

real t
f1 ← x1 − x2 + t(2 − t(1 + t))
f2 ← x1 + x2 − t2(4 − t)
end procedure Sistema_XP

```

Un experimento numérico para comparar los resultados del método de la serie de Taylor y el método de Runge-Kutta con la solución analítica del sistema (1) se sugiere en el problema de cómputo 11.1.1. En el punto $t = 1.0$, los resultados son los siguientes:

Serie de Taylor	Runge-Kutta	Solución analítica
$x(1.0) \approx 2.46869\ 40$	2.46869 42	2.46869 39399
$y(1.0) \approx 1.28735\ 46$	1.28735 61	1.28735 52872

Podemos utilizar las rutinas de software matemático de Matlab, Maple o Mathematica para obtener la solución numérica del sistema de ecuaciones diferenciales (1). Para t en el intervalo $[0, 1]$, usamos a un procedimiento de EDO para ir de $t = 0$ en el que $x(0) = 1$ y $y(0) = 0$ para $t = 1$ en la que $x(1) = 2.468693912$ y $y(1) = 1.287355325$.

Para obtener la solución numérica de la ecuación diferencial ordinaria definida para t en el intervalo $[1, 1.5]$, usamos un procedimiento de resolución de una ecuación diferencial ordinaria para ir de $t = 0$ en el que $x(1) = 2$ y de $y(1) = -2$ a $t = 1.5$ en el que $x(1.5) \approx 15.5028$ y $y(1.5) \approx 6.15486$.

EDO autónoma

Cuando escribimos el sistema de ecuaciones diferenciales en forma vectorial

$$\mathbf{X}' = \mathbf{F}(t, \mathbf{X})$$

supusimos que la variable t fue explícitamente separada de las otras variables y tratada de manera diferente. No es necesario hacer esto. En realidad, podemos introducir una nueva variable x_0 que es t disfrazada y agregar una nueva ecuación diferencial $x'_0 = 1$. También se debe proporcionar una nueva condición inicial $x_0(a) = a$. De esta manera, se aumenta el número de ecuaciones diferenciales de n a $n + 1$ y se obtiene un sistema escrito en forma vectorial más elegante

$$\begin{cases} \mathbf{X}' = \mathbf{F}(\mathbf{X}) \\ \mathbf{X}(a) = \mathbf{S}, \text{ dada} \end{cases}$$

Considere el sistema de dos ecuaciones dado por la ecuación (1). Se escribe como un sistema con tres variables haciendo

$$x_0 = t, \quad x_1 = x, \quad x_2 = y$$

Así, tenemos

$$\begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} 1 \\ x_1 - x_2 + 2x_0 - x_0^2 - x_0^3 \\ x_1 + x_2 - 4x_0^2 + x_0^3 \end{bmatrix}$$

La condición de auxiliar para el vector \mathbf{X} es $\mathbf{X}(0) = [0, 1, 0]^T$.

Como resultado de las observaciones anteriores, no sacrificamos generalidad en la consideración de un sistema de $n + 1$ ecuaciones diferenciales de primer orden escrito como

$$\left\{ \begin{array}{l} x'_0 = f_0(x_0, x_1, x_2, \dots, x_n) \\ x'_1 = f_1(x_0, x_1, x_2, \dots, x_n) \\ x'_2 = f_2(x_0, x_1, x_2, \dots, x_n) \\ \vdots \\ x'_n = f_n(x_0, x_1, x_2, \dots, x_n) \\ x_0(a) = s_0, x_1(a) = s_1, x_2(a) = s_2, \dots, x_n(a) = s_n \end{array} \right. \text{ todas dadas}$$

Podemos escribir este sistema en notación vectorial general

$$\left\{ \begin{array}{l} \mathbf{X}' = \mathbf{F}(\mathbf{X}) \\ \mathbf{X}(a) = \mathbf{S}, \quad \text{dado} \end{array} \right. \quad (7)$$

donde

$$\mathbf{X} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{X}' = \begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \quad \mathbf{S} = \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$$

Un sistema de ecuaciones diferenciales sin la variable t presente de forma explícita se dice que es **autónomo**. Los métodos numéricos que se analicen no requieren que $x_0 = t$ o $f_0 = 1$ o $s_0 = a$.

Para un sistema autónomo, el **método clásico de Runge-Kutta de cuarto orden** para el sistema (6) utiliza estas fórmulas:

$$\mathbf{X}(t + h) = \mathbf{X} + \frac{h}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4) \quad (8)$$

donde

$$\left\{ \begin{array}{l} \mathbf{K}_1 = \mathbf{F}(\mathbf{X}) \\ \mathbf{K}_2 = \mathbf{F}\left(\mathbf{X} + \frac{1}{2}h\mathbf{K}_1\right) \\ \mathbf{K}_3 = \mathbf{F}\left(\mathbf{X} + \frac{1}{2}h\mathbf{K}_2\right) \\ \mathbf{K}_4 = \mathbf{F}(\mathbf{X} + h\mathbf{K}_3) \end{array} \right.$$

En este caso, $\mathbf{X} = \mathbf{X}(t)$ y todas las cantidades son vectores con $n + 1$ componentes excepto las variables h .

En el ejemplo anterior, el procedimiento *RK4_System1* tendría que ser modificado iniciando los arreglos en 0 en vez de 1 y omitiendo de la variable t . (Lo llamamos *RK4_System2* y lo dejamos como el problema de cómputo 11.1.4.) Entonces la llamada de los programas sería el siguiente:

```
program Prueba_Sistema2.RK4
real h, t;  real array (x_i)_{0:n}
integer n ← 2, nsteps ← 100
real a ← 0, b ← 1
(x_i) ← (0, 1, 0)
h ← (b - a)/nsteps
call RK4_System2(n, h, (x_i), nsteps)
end program Test_RK4_System2
```

```

procedure XP_System( $n, (x_i), (f_i)$ )
real array ( $x_i$ ) $_{0:n}, (f_i)$  $_{0:n}$ 
integer  $n$ 
 $f_0 \leftarrow 1$ 
 $f_1 \leftarrow x_1 - x_2 + x_0(2 - x_0(1 + x_0))$ 
 $f_2 \leftarrow x_1 + x_2 - x_0^2(4 - x_0)$ 
end procedure Sistema_XP

```

Es usual en solucionadores de ecuaciones diferenciales ordinarias como los que se encuentran en las bibliotecas de software matemático que el usuario interactúe con ellos escribiendo un subprograma en un formato no autónomo. En otras palabras, el solucionador de la ecuación diferencial ordinaria toma como entrada tanto la variable independiente como la dependiente y devuelve los valores para el miembro derecho de la ecuación diferencial ordinaria. En consecuencia, la convención de programación no autónoma puede parecer más natural para los que están usando estos paquetes de software.

Es un ejercicio útil encontrar una aplicación física en su campo de estudio o profesión que implique la solución de una ecuación diferencial ordinaria. Es instructivo analizar y resolver el problema físico determinando el método numérico adecuado y traduciendo el problema al formato que sea compatible con el software disponible.

Resumen

(1) Un sistema de ecuaciones diferenciales ordinarias

$$\begin{cases} x'_1 = f_1(t, x_1, x_2, \dots, x_n) \\ x'_2 = f_2(t, x_1, x_2, \dots, x_n) \\ \vdots \\ x'_n = f_n(t, x_1, x_2, \dots, x_n) \\ x_1(a) = s_1, x_2(a) = s_2, \dots, x_n(a) = s_n, \quad \text{todas dadas} \end{cases}$$

se puede escribir en notación vectorial como

$$\begin{cases} X' = \mathbf{F}(t, X) \\ X(a) = S, \quad \text{dada} \end{cases}$$

donde se definen los siguientes n vectores componentes

$$\begin{cases} X = [x_1, x_2, \dots, x_n]^T \\ X' = [x'_1, x'_2, \dots, x'_n]^T \\ \mathbf{F} = [f_1, f_2, \dots, f_n]^T \\ X(a) = [x_1(a), x_2(a), \dots, x_n(a)]^T \end{cases}$$

(2) El **método de la serie de Taylor de orden m** es

$$X(t+h) = X + hX' + \frac{h^2}{2}X'' + \cdots + \frac{h^m}{m!}X^{(m)}$$

donde $X = X(t)$, $X' = X'(t)$, $X'' = X''(t)$ y así sucesivamente.

(3) El **método de Runge-Kutta de orden 4** es

$$X(t + h) = X + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

donde

$$\begin{cases} K_1 = F(t, X) \\ K_2 = F\left(t + \frac{1}{2}h, X + \frac{1}{2}hK_1\right) \\ K_3 = F\left(t + \frac{1}{2}h, X + \frac{1}{2}hK_2\right) \\ K_4 = F(t + h, X + hK_3) \end{cases}$$

En este caso, $X = X(t)$ y todas las cantidades son vectores con n componentes, excepto las variables t y h .

(4) Podemos absorber la variable t en el vector haciendo $x_0 = t$ y después escribiendo la forma autónoma para el sistema de ecuaciones diferenciales ordinarias en notación vectorial como

$$\begin{cases} X' = F(X) \\ X(a) = S, \text{ dada} \end{cases}$$

donde los vectores se definen a tener $n + 1$ componentes. Entonces

$$\begin{cases} X = [x_0, x_1, x_2, \dots, x_n]^T \\ X' = [x'_0, x'_1, x'_2, \dots, x'_n]^T \\ F = [1, f_1, f_2, \dots, f_n]^T \\ X(a) = [a, x_1(a), x_2(a), \dots, x_n(a)]^T \end{cases}$$

(5) El **método de Runge-Kutta de orden 4** para el sistema de ecuaciones diferenciales ordinarias en forma autónoma es

$$X(t + h) = X + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

donde

$$\begin{cases} K_1 = F(X) \\ K_2 = F\left(X + \frac{1}{2}hK_1\right) \\ K_3 = F\left(X + \frac{1}{2}hK_2\right) \\ K_4 = F(X + hK_3) \end{cases}$$

En este caso, $X = X(t)$ y todas las cantidades de F y K_i son vectores con $n + 1$ componentes, excepto las variables t y h .

Problemas 11.1

1. Considere

$$\begin{cases} x' = y \\ y' = x \end{cases} \quad \text{con} \quad \begin{cases} x(0) = -1 \\ y(0) = 0 \end{cases}$$

Escriba las ecuaciones, sin derivadas, para utilizarse en el método de la serie de Taylor de orden 5.

“2. ¿Cómo resolvería este sistema de ecuaciones diferenciales numéricamente?

$$\begin{cases} x'_1 = x_1^2 + e^t - t^2 \\ x'_2 = x_2 - \cos t \\ x_1(0) = 0 \quad x_2(1) = 0 \end{cases}$$

“3. ¿Cómo resolvería el problema con valor inicial

$$\begin{cases} x'_1(t) = x_1(t)e^t + \sin t - t^2 \\ x'_2(t) = [x_2(t)]^2 - e^t + x_2(t) \\ x_1(1) = 2 \quad x_2(1) = 4 \end{cases}$$

si tuviera un programa de computadora para resolver un problema con valor inicial de la forma $x' = f(t, x)$ que implica una única función incógnita $x = x(t)$?

“4. Escriba un sistema equivalente de ecuaciones diferenciales de primer orden sin que aparezca t en el miembro derecho:

$$\begin{cases} x' = x^2 + \log(y) + t^2 \\ y' = e^y - \cos(x) + \sin(tx) - (xy)^7 \\ x(0) = 1 \quad y(0) = 3 \end{cases}$$

Problemas de cómputo 11.1

“1. Resuelva el sistema de ecuaciones diferenciales (1) usando los dos métodos que se presentan en esta sección y compare los resultados con la solución analítica.

“2. Resuelva el problema con valor inicial

$$\begin{cases} x' = t + x^2 - y \\ y' = t^2 - x + y^2 \\ x(0) = 3 \quad y(0) = 2 \end{cases}$$

por medio del método de la serie de Taylor usando $h = 1/128$ en el intervalo $[0, 0.38]$. Incluya términos que impliquen tres derivadas en x y y . ¿Cuán exactos son los valores de la función calculados?

3. Escriba el procedimiento de Runge-Kutta para resolver

$$\begin{cases} x'_1 = -3x_2 \\ x'_2 = \frac{1}{3}x_1 \\ x_1(0) = 0 \quad x_2(0) = 1 \end{cases}$$

en el intervalo $0 \leq t \leq 4$. Trace la gráfica de la solución.

“4. Escriba el procedimiento *RK4_System2* y un programa controlador para resolver el sistema de ecuaciones diferenciales ordinarias dado por la ecuación (2). Use $h = -10^{-2}$ e imprima los valores de x_0 , x_1 y x_2 , junto con la verdadera solución en el intervalo $[-1, 0]$. Compruebe que la verdadera solución es $x(t) = e^t + 6 + 6t + 4t^2 + t^3$ y $y(t) = e^t - t^3 + t^2 + 2t + 2$.

- “5.** Usando el procedimiento de Runge-Kutta, resuelva el siguiente problema con valor inicial en el intervalo $0 \leq t \leq 2\pi$. Trace las gráficas de las curvas resultantes $(x_1(t), x_2(t))$ y $(x_3(t), x_4(t))$. Deben ser círculos.

$$\begin{cases} X' = \begin{bmatrix} x_3 \\ x_4 \\ -x_1(x_1^2 + x_2^2)^{-3/2} \\ -x_2(x_1^2 + x_2^2)^{-3/2} \end{bmatrix} \\ X(0) = [1, 0, 0, 1]^T \end{cases}$$

- 6.** Resuelva el problema

$$\begin{cases} x_0' = 1 \\ x_1' = -x_2 + \cos x_0 \\ x_2' = x_1 + \sin x_0 \\ x_0(1) = 1 \quad x_1(1) = 0 \quad x_2(1) = -1 \end{cases}$$

Utilice el método de Runge-Kutta y el intervalo $-1 \leq t \leq 2$.

- “7.** Escriba y pruebe un programa, usando el método de la serie de Taylor de orden 5, para resolver el sistema de

$$\begin{cases} x' = tx - y^2 + 3t \\ y' = x^2 - ty - t^2 \\ x(5) = 2 \quad y(5) = 3 \end{cases}$$

en el intervalo $[5, 6]$ usando $h = 10^{-3}$. Imprima valores de x y y en pasos de 0.1.

- 8.** Imprima una tabla de $\sin t$ y de $\cos t$ en el intervalo $[0, \pi/2]$ al resolver numéricamente el sistema

$$\begin{cases} x' = y \\ y' = -x \\ x(0) = 0 \quad y(0) = 1 \end{cases}$$

- 9.** Escriba un programa para usar el método de la serie de Taylor de orden 3 para resolver el sistema

$$\begin{cases} x' = tx + y' - t^2 \\ y' = ty + 3t \\ z' = tz - y' + 6t^3 \\ x(0) = 1 \quad y(0) = 2 \quad z(0) = 3 \end{cases}$$

en el intervalo $[0, 0.75]$ usando $h = 0.01$.

- 10.** Escriba y pruebe un programa corto para resolver el sistema de ecuaciones diferenciales

$$\begin{cases} y' = x^3 - t^2y - t^2 \\ x' = tx^2 - y^4 + 3t \\ y(2) = 5 \quad x(2) = 3 \end{cases}$$

en el intervalo $[2, 5]$, con $h = 0.25$. Utilice el método de la serie de Taylor de orden 4.

- 11.** Recodifique y pruebe el procedimiento *Sistema2_RK4* usando un lenguaje de computadora que soporte operaciones vectoriales

12. Compruebe los resultados numéricos dados en el libro para el sistema de ecuaciones diferenciales (1) de los programas *Prueba_Sistema1_RK4* y *Sistema2_RK4*.
13. (Continuación) Usando el software matemático como Matlab, Maple o Mathematica que contienen capacidades de manejo simbólico para comprobar la solución analítica para el sistema de ecuaciones diferenciales (1).
14. (Continuación) Use rutinas de software matemático como las que se encuentran en Matlab, Maple o Mathematica para comprobar las soluciones numéricas dadas en el libro. Trace la curva solución resultante. Compare con los resultados de los programas *Prueba_Sistema1_RK4* y *Prueba_Sistema2_RK4*.

11.2 Ecuaciones de orden superior y sistemas

Considere el problema con valor inicial para ecuaciones diferenciales ordinarias de orden mayor a 1. Una ecuación diferencial de orden n normalmente se acompaña de n condiciones auxiliares. Se necesitan varias condiciones iniciales para especificar la solución de la ecuación diferencial con precisión (suponiendo que ciertas condiciones de suavidad están presentes). Tomemos, por ejemplo, un problema con valor inicial de segundo orden dado

$$\begin{cases} x''(t) = -3 \cos^2(t) + 2 \\ x(0) = 0 \quad x'(0) = 0 \end{cases} \quad (1)$$

Sin las condiciones auxiliares, la solución analítica general es

$$x(t) = \frac{1}{4}t^2 + \frac{3}{8} \cos(2t) + c_1 t + c_2$$

donde c_1 y c_2 son constantes arbitrarias. Para seleccionar una solución específica se deben fijar c_1 y c_2 y establecer dos condiciones iniciales. De hecho, con $x(0) = 0$ se obtiene $c_2 = -\frac{3}{8}$, y con $x'(0) = 0$ se obtiene $c_1 = 0$.

Ecuaciones diferenciales de orden superior

En general, los problemas de mayor orden pueden ser mucho más complicados que este simple ejemplo porque el sistema (1) tiene la característica especial de que la función en el miembro derecho de la ecuación diferencial no implica a x . La forma más general de una ecuación diferencial ordinaria con condiciones iniciales que vamos a considerar es

$$\begin{cases} x^{(n)} = f(t, x, x', x'', \dots, x^{(n-1)}) \\ x(a), x'(a), x''(a), \dots, x^{(n-1)}(a) \quad \text{todas dadas} \end{cases} \quad (2)$$

Esta se puede resolver numéricamente al convertirla en un sistema de ecuaciones diferenciales de *primer orden*. Para hacerlo se definen nuevas variables x_1, x_2, \dots, x_n como sigue:

$$x_1 = x \quad x_2 = x' \quad x_3 = x'' \quad \dots \quad x_{n-1} = x^{(n-2)} \quad x_n = x^{(n-1)}$$

En consecuencia, el problema original con valor inicial (2) es equivalente a

$$\begin{cases} x'_1 = x_2 \\ x'_2 = x_3 \\ \vdots \\ x'_{n-1} = x_n \\ x'_n = f(t, x_1, x_2, \dots, x_n) \\ x_1(a), x_2(a), \dots, x_n(a) \text{ todas dadas} \end{cases}$$

o, en notación vectorial,

$$\begin{cases} \mathbf{X}' = \mathbf{F}(t, \mathbf{X}) \\ \mathbf{X}(a) = \mathbf{S}, \text{ dada} \end{cases} \quad (3)$$

donde

$$\begin{aligned} \mathbf{X} &= [x_1, x_2, \dots, x_n]^T \\ \mathbf{X}' &= [x'_1, x'_2, \dots, x'_n]^T \\ \mathbf{F} &= [x_2, x_3, x_4, \dots, x_n, f]^T \end{aligned}$$

y

$$\mathbf{X}(a) = [x_1(a), x_2(a), \dots, x_n(a)]$$

Cada vez que un problema se deba transformar introduciendo nuevas variables se recomienda que se proporcione un *diccionario* para mostrar la relación entre variables nuevas y viejas. Al mismo tiempo, esta información, junto con las ecuaciones diferenciales y los valores iniciales, se puede presentar en una tabla. Este archivo sistemático puede ser útil en una situación complicada.

Como ejemplo, transformemos el problema con valor inicial

$$\begin{cases} x''' = \cos x + \operatorname{sen} x' - e^{x''} + t^2 \\ x(0) = 3 \quad x'(0) = 7 \quad x''(0) = 13 \end{cases} \quad (4)$$

en una forma adecuada para su solución mediante el procedimiento de Runge-Kutta. La tabla que resume el problema transformado es la siguiente:

Variable vieja	Variable nueva	Valor inicial	Ecuación diferencial
x	x_1	3	$x'_1 = x_2$
x'	x_2	7	$x'_2 = x_3$
x''	x_3	13	$x'_3 = \cos x_1 + \operatorname{sen} x_2 - e^{x_3} + t^2$

Así, el correspondiente sistema de primer orden es

$$\mathbf{X}' = \begin{bmatrix} x_2 \\ x_3 \\ \cos x_1 + \operatorname{sen} x_2 - e^{x_3} + t^2 \end{bmatrix}$$

y $\mathbf{X}(0) = [3, 7, 13]^T$.

Sistemas de ecuaciones diferenciales de orden superior

Al introducir sistemáticamente variables nuevas, podemos transformar un sistema de ecuaciones diferenciales de diferentes órdenes en un sistema más grande de ecuaciones de primer orden. Por ejemplo, el sistema de

$$\begin{cases} x'' = x - y - (3x')^2 + (y')^3 + 6y'' + 2t \\ y''' = y'' - x' + e^x - t \\ x(1) = 2 \quad x'(1) = -4 \quad y(1) = -2 \quad y'(1) = 7 \quad y''(1) = 6 \end{cases} \quad (5)$$

se puede resolver con el procedimiento de Runge-Kutta si primero se transforma de acuerdo con la siguiente tabla:

Variable vieja	Variable nueva	Valor inicial	Ecuación diferencial
x	x_1	2	$x'_1 = x_2$
x'	x_2	-4	$x'_2 = x_1 - x_3 - 9x_2^2 + x_4^3 + 6x_5 + 2t$
y	x_3	-2	$x'_3 = x_4$
y'	x_4	7	$x'_4 = x_5$
y''	x_5	6	$x'_5 = x_5 - x_2 + e^{x_1} - t$

Por lo tanto, tenemos

$$\mathbf{X}' = \begin{bmatrix} x_2 \\ x_1 - x_3 - 9x_2^2 + x_4^3 + 6x_5 + 2t \\ x_4 \\ x_5 \\ x_5 - x_2 + e^{x_1} - t \end{bmatrix}$$

$$\text{y } X(1) = [2, -4, -2, 7, 6]^T.$$

Sistemas de EDO autónomas

Nos damos cuenta de que t está presente en el miembro derecho de la ecuación (3) y que por tanto se pueden introducir las ecuaciones $x_0 = t$ y $x'_0 = 1$ para formar un sistema autónomo de ecuaciones diferenciales ordinarias en notación vectorial. Es fácil demostrar que un sistema de ecuaciones diferenciales de orden superior que tiene la forma de la ecuación (2) se puede escribir en notación vectorial como

$$\begin{cases} \mathbf{X}' = \mathbf{F}(\mathbf{X}) \\ \mathbf{X}(a) = \mathbf{S}, \quad \text{dada} \end{cases}$$

donde

$$\begin{aligned} \mathbf{X} &= [x_0, x_1, x_2, \dots, x_n]^T \\ \mathbf{X}' &= [x'_0, x'_1, x'_2, \dots, x'_n]^T \\ \mathbf{F} &= [1, x_2, x_3, x_4, \dots, x_n, f]^T \end{aligned}$$

y

$$\mathbf{X}(a) = [a, x_1(a), x_2(a), \dots, x_n(a)]$$

Como ejemplo, el sistema de ecuaciones diferenciales ordinarias de la ecuación (4) se puede escribir en forma autónoma como

$$\mathbf{X}' = \begin{bmatrix} 1 \\ x_2 \\ x_1 - x_3 - 9x_2^2 + x_4^3 + 6x_5 + 2x_0 \\ x_4 \\ x_5 \\ x_5 - x_2 + e^{x_1} - x_0 \end{bmatrix}$$

y $\mathbf{X}(0) = [1, 2, -4, -2, 7, 6]^T$.

Resumen

(1) Una única ecuación diferencial ordinaria de n -ésimo orden con valores iniciales tiene la forma

$$\begin{cases} x^{(n)} = f(t, x, x', x'', \dots, x^{(n-1)}) \\ x(a), x'(a), x''(a), \dots, x^{(n-1)}(a), \text{ todas dadas} \end{cases}$$

Puede convertirse en un sistema de ecuaciones de primer orden de la forma

$$\begin{cases} \mathbf{X}' = \mathbf{F}(t, \mathbf{X}) \\ \mathbf{X}(a) = \mathbf{S}, \text{ dada} \end{cases}$$

dónde

$$\begin{cases} \mathbf{X} = [x_1, x_2, \dots, x_n]^T \\ \mathbf{X}' = [x'_1, x'_2, \dots, x'_n]^T \\ \mathbf{F} = [x_2, x_3, x_4, \dots, x_n, f]^T \\ \mathbf{X}(a) = [x_1(a), x_2(a), \dots, x_n(a)]^T \end{cases}$$

(2) Podemos absorber la variable t en la notación vectorial haciendo $x_0 = t$ y extendiendo los vectores de longitud $n + 1$. Así, una única ecuación ordinaria diferencial de orden n se puede escribir como

$$\begin{cases} \mathbf{X}' = \mathbf{F}(\mathbf{X}) \\ \mathbf{X}(a) = \mathbf{S}, \text{ dada} \end{cases}$$

dónde

$$\begin{cases} \mathbf{X} = [x_0, x_1, x_2, \dots, x_n]^T \\ \mathbf{X}' = [x'_0, x'_1, x'_2, \dots, x'_n]^T \\ \mathbf{F} = [1, x_2, x_3, x_4, \dots, x_n, f]^T \\ \mathbf{X}(a) = [a, x_1(a), x_2(a), \dots, x_n(a)]^T \end{cases}$$

Problemas 11.2

- 1** Convierta esta ecuación diferencial en un sistema de ecuaciones de primer orden adecuado para la aplicación del método de Runge-Kutta:

$$\begin{cases} x''' = 2x' + \log(x'') + \cos(x) \\ x(0) = 1 \quad x'(0) = -3 \quad x''(0) = 5 \end{cases}$$

- 2. a.** Suponiendo que se cuenta con un programa para resolver problemas con valor iniciales de la forma de la ecuación (3), ¿cómo se puede utilizar para resolver la siguiente ecuación diferencial?

$$\begin{cases} x''' = t + x + 2x' + 3x'' \\ x(1) = 3 \quad x'(1) = -7 \quad x''(1) = 4 \end{cases}$$

- b.** ¿Cómo puede resolverse este problema si las condiciones iniciales son $x(1) = 3$, $x'(1) = -7$, y $x'''(1) = 0$?

- 3.** ¿Cómo resolvería numéricamente este problema de ecuaciones diferenciales?

$$\begin{cases} x_1'' = x_1' + x_1^2 - \sin t \\ x_2'' = x_2 - (x_2')^{1/2} + t \\ x_1(0) = 1 \quad x_2(1) = 3 \quad x_1'(0) = 0 \quad x_2'(1) = -2 \end{cases}$$

- 4.** Convierta estas ecuaciones orbitales en un sistema de primer orden

$$\begin{cases} x'' + x(x^2 + y^2)^{-3/2} = 0 \\ y'' + y(x^2 + y^2)^{-3/2} = 0 \end{cases}$$

con condiciones iniciales

$$x(0) = 0.5 \quad x'(0) = 0.75 \quad y(0) = 0.25 \quad y'(0) = 1.0$$

- 5.** Reescriba la ecuación siguiente como un sistema de ecuaciones diferenciales de primer orden sin que aparezca t en el miembro derecho:

$$\begin{cases} x^{(4)} = (x''')^2 + \cos(x'') - \sin(tx) + \log\left(\frac{x}{t}\right) \\ x(0) = 1 \quad x'(0) = 3 \quad x''(0) = 4 \quad x'''(0) = 5 \end{cases}$$

- 6.** Exprese el sistema de ecuaciones diferenciales ordinarias

$$\begin{cases} \frac{d^2z}{dt^2} - 2t \frac{dz}{dt} = 2te^{xz} \\ \frac{d^2x}{dt^2} - 2xz \frac{dx}{dt} = 3x^2yt^2 \\ \frac{d^2y}{dt^2} - e^y \frac{dy}{dt} = 4xt^2z \\ z(1) = x''(1) = y'(1) = 2 \quad z(1) = x(1) = y(1) = 3 \end{cases}$$

como un sistema de ecuaciones diferenciales ordinarias de primer orden.

- 7.** Determine un sistema de ecuaciones de primer orden equivalente a cada de los siguientes:

- a.** $x''' + x'' \operatorname{sen} x + tx' + x = 0$ **b.** $x^{(4)} + x'' \cos x' + txx' = 0$
c. $\begin{cases} x'' = 3x^2 - 7y^2 + \operatorname{sen} t + \cos(x'y') \\ y''' = y + x^2 - \cos t - \operatorname{sen}(xy'') \end{cases}$

- 8.** Considere

$$\begin{cases} x'' = x' - x \\ x(0) = 0 \quad x'(0) = 1 \end{cases}$$

Determine el sistema asociado de primer orden y sus condiciones iniciales auxiliares.

- 9.** Escriba el problema

$$\begin{cases} x''(t) = x + y - 2x' + 3y' + \log t \\ y''(t) = 2x - 3y + 5x' + ty' - \sin t \\ x(0) = 1 \quad x'(0) = 2 \\ y(0) = 3 \quad y'(0) = 4 \end{cases}$$

en la forma de un sistema autónomo de cinco ecuaciones de primer orden. Presente el sistema resultante y los valores iniciales adecuados.

- 10.** Escriba el procedimiento *XP_System* para usarlo con la rutina de Runge-Kutta de cuarto orden *RK4_System1* para la siguiente ecuación diferencial:

$$\begin{cases} x''' = 10e^{x''} - x'''\sin(x'x) - (xt)^{10} \\ x(2) = 6.5 \quad x'(2) = 4.1 \quad x''(2) = 3.2 \end{cases}$$

- 11.** Si se va a resolver el problema con valor inicial

$$\begin{cases} x''' = x' - tx'' + x + \ln t \\ x(1) = x'(1) = x''(1) = 1 \end{cases}$$

usando las fórmulas de Runge-Kutta, ¿cómo se debe transformar el problema?

- 12.** Convierta este problema que implica ecuaciones diferenciales en un sistema autónomo de ecuaciones de primer orden (con valores iniciales):

$$\begin{cases} 3x' + \tan x'' - x^2 = \sqrt{t^2 + 1} + y^2 + (y')^2 \\ -3y' + \cot y'' + y^2 = t^2 + (x + 1)^{1/2} + 4x' \\ x(1) = 2 \quad x'(1) = -2 \quad y(1) = 7 \quad y'(1) = 3 \end{cases}$$

- 13.** Siga en este ejemplo las instrucciones del problema anterior:

$$\begin{cases} txyz + x'y'/t = tx^2 + x/y'' + z \\ t^2x/z + y'z't = y^2 - (z'')^2x + x'y' \\ tyz - x'z'y' = z^2 - zx'' - (yz)' \\ x(3) = 1 \quad y(3) = 2 \quad z(3) = 4 \quad x'(3) = 5 \quad y'(3) = 6 \quad z'(3) = 7 \end{cases}$$

- 14.** Convierta este par de ecuaciones diferenciales en una ecuación diferencial de segundo orden que implique sólo a x :

$$\begin{cases} x' = -x + axy \\ y' = 3y - xy \end{cases}$$

Problemas de cómputo 11.2

- 1.** Utilice *RK4_System1* para resolver cada de los siguientes sistemas de $0 \leq t \leq 1$. Use $h = 2^{-k}$, con $k = 5, 6$ y 7 , y compare resultados.

a. $\begin{cases} x'' = 2(e^{2t} - x^2)^{1/2} \\ x(0) = 0 \quad x'(0) = 1 \end{cases}$

b. $\begin{cases} x'' = x^2 - y + e^t \\ y'' = x - y^2 - e^t \\ x(0) = 0 \quad x'(0) = 0 \\ y(0) = 1 \quad y'(0) = -2 \end{cases}$

2. Resuelva la ecuación diferencial de Airy

$$\begin{cases} x'' = tx \\ x(0) = 0.35502\ 80538\ 87817 \\ x'(0) = -0.25881\ 94037\ 92807 \end{cases}$$

en el intervalo $[0, 4.5]$ usando el método de Runge-Kutta. *Valor de comprobación:* el valor $x(4.5) = 0.00033\ 02503$ es correcto.

3. Resuelva

$$\begin{cases} x'' + x' + x^2 - 2t = 0 \\ x(0) = 0 \quad x'(0) = 0.1 \end{cases}$$

en $[0, 3]$ usando cualquier método conveniente. Si se dispone de un graficador, trace la gráfica de la solución.

4. Resuelva

$$\begin{cases} x'' = 2x' - 5x \\ x(0) = 0 \quad x'(0) = 0.4 \end{cases}$$

en el intervalo $[-2, 0]$.

- 5.** Escriba programas de computadora basados en el seudocódigo del libro para encontrar la solución numérica de estos sistemas de ecuaciones diferenciales ordinarias:

a. (1)

b. (4)

c. (5)

- 6.** (Continuación) Utilice software matemático como Matlab, Maple o Mathematica con capacidades de manejo simbólico para encontrar sus soluciones analíticas.

- 7.** (Continuación) Use rutinas de software matemático como las que se encuentran en Matlab, Maple o Mathematica para comprobar las soluciones numéricas de estos sistemas de ecuaciones diferenciales ordinarias. Trace la curva solución resultante.

11.3 Métodos de Adams-Bashforth-Moulton

Un esquema predictor-corrector

Los procedimientos explicados hasta ahora han resuelto el problema con valor inicial

$$\begin{cases} X' = F(X) \\ X(a) = S, \quad \text{dada} \end{cases} \tag{1}$$

por medio de métodos numéricos de un **solo paso**. En otras palabras, si se conoce la solución $X(t)$ en un punto t en particular, entonces se calcula $X(t+h)$ sin conocer la solución en los puntos anteriores a t . Los métodos de Runge-Kutta y de la serie de Taylor calculan $X(t+h)$ en términos de $X(t)$ y diferentes valores de F .

Se pueden concebir métodos más eficientes si se utilizan varios valores $X(t), X(t-h), X(t-2h), \dots$ en el cálculo de $X(t+h)$. Estos métodos se llaman métodos de **multipaso**. Tienen el inconveniente obvio de que al comienzo de la solución numérica no hay valores anteriores de X . Por ello es normal iniciar una solución numérica con métodos de un solo paso, tales como el procedimiento de Runge-Kutta, y trasladarse por eficiencia a un procedimiento multipaso en cuanto se hayan calculado suficientes valores iniciales.

Un ejemplo de una fórmula multipaso se conoce como el **método de Adams-Bashforth** (véase la sección 10.3 y el problema relacionado). Este es

$$\begin{aligned}\tilde{X}(t+h) = X(t) + \frac{h}{24} &\{55F[X(t)] - 59F[X(t-h)] + 37F[X(t-2h)] \\ &- 9F[X(t-3h)]\}\end{aligned}\quad (2)$$

En este caso, $\tilde{X}(t+h)$ es el valor predicho de $X(t+h)$ calculado usando la fórmula (2). Si se ha calculado la solución X en los cuatro puntos $t, t-h, t-2h$ y $t-3h$, entonces se puede usar la fórmula (2) para calcular $\tilde{X}(t+h)$. Si esto se hace sistemáticamente, entonces sólo se requiere *una* evaluación de F para cada paso. Esto representa un ahorro considerable comparado con el procedimiento de Runge-Kutta de cuarto orden, que requiere *cuatro* evaluaciones de F por paso. (Por supuesto, una consideración de error de truncamiento y estabilidad podría permitir un mayor tamaño de paso en el método de Runge-Kutta y esto lo hace mucho más competitivo.)

En la práctica, la fórmula (2) nunca se utiliza por sí misma. En su lugar, se utiliza como un *predictor* y entonces otra fórmula se emplea como un *corrector*. El corrector que se usa generalmente con la fórmula (2) es la **fórmula de Adams-Moulton**:

$$\begin{aligned}X(t+h) = X(t) + \frac{h}{24} &\{9F[\tilde{X}(t+h)] + 19F[X(t)] - 5F[X(t-h)] \\ &+ F[X(t-2h)]\}\end{aligned}\quad (3)$$

Así, la ecuación (2) predice un valor provisional de $X(t+h)$ y la ecuación (3) calcula este valor X con más precisión. La combinación de las dos fórmulas da como resultado un **esquema corrector-predictor**.

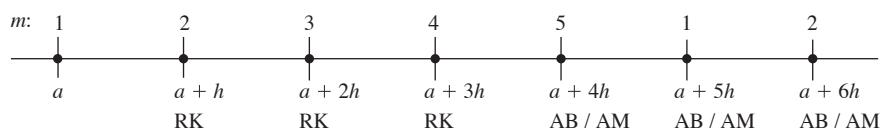
Con valores iniciales de X especificados en a , se pueden realizar tres pasos del método de Runge-Kutta para determinar suficientes valores de X como para que el procedimiento de Adams-Bashforth-Moulton pueda comenzar. Las fórmulas de cuarto orden de Adams-Bashforth y de Adams-Moulton, que inician con el método de Runge-Kutta de cuarto orden, se conocen como el **método de Adams-Moulton**. Las fórmulas predictor y corrector del mismo orden se utilizan para que sólo se necesite una aplicación de la fórmula corrector. Algunos sugieren iterar la fórmula corrector, pero la experiencia ha demostrado que el mejor método global es sólo *una* aplicación por paso.

Seudocódigo

El almacenamiento de la solución aproximada en pasos previos en el método de Adams-Moulton se maneja generalmente ya sea mediante el almacenamiento de un arreglo de dimensión mayor que el número total de pasos por seguir o bien, mediante el cambio físico de los datos después de cada paso (descartando los datos más antiguos y almacenando en su lugar los más nuevos). Si se utiliza un proceso adaptado, el número total de pasos no se puede determinar de antemano. El desplazamiento físico de los datos se puede eliminar por la interacción de los índices de un arreglo de almacenamiento de dimensión fija. Por el método de Adams-Moulton, los datos x_i para $X(t)$ se almacenan en un arreglo bidimensional con entradas z_{im} en los lugares $m = 1, 2, 3, 4, 5, 1, 2, \dots$ para $t = a, a+h, a+2h, a+3h, a+4h, a+5h, a+6h, \dots$, respectivamente. El dibujo de la figura 11.1 muestra los primeros diferentes valores de t con los correspondientes valores de m y las abreviaturas de las fórmulas utilizadas.

Se puede realizar un análisis de error después de cada paso del método de Adams-Moulton. Si $x_i^{(p)}$ es la aproximación numérica de la i -ésima ecuación del sistema (1) en $t+h$ obtenido por

FIGURA 11.1
Valores iniciales para las aplicaciones de RK y métodos AB/AM



la fórmula predictor (2) y x_i se obtiene con la fórmula corrector (3) en $t + h$, entonces se puede demostrar que el error de un solo paso para el i -ésimo componente en $t + h$ está dado aproximadamente por

$$\varepsilon_i = \frac{19}{270} \frac{|x_i - x_i^{(p)}|}{|x_i|}$$

Así, calculamos

$$\text{est} = \max_{1 \leq i \leq n} |\varepsilon_i|$$

con el procedimiento de Adams-Moulton, *System_AM* para obtener una estimación del error máximo de un solo paso en $t + h$.

Se necesita un procedimiento de control que llame al procedimiento de Runge-Kutta tres veces y después llame al esquema predictor-corrector de Adams-Moulton para calcular los pasos restantes. El procedimiento para hacer *nsteps* con pasos con un tamaño fijo de paso h es el siguiente:

```

procedure AMRK( $n, h, (x_i), nsteps$ )
integer  $i, k, m, n;$  real est, h; real array ( $x_i$ ) $_{0:n}$ 
allocate real array ( $f_{ij}$ ) $_{0:n \times 0:4}, (z_{ij})_{0:n \times 0:4}$ 
 $m \leftarrow 0$ 
output h
output 0, ( $x_i$ )
for  $i = 0$  to  $n$  do
     $z_{im} \leftarrow x_i$ 
end for
for  $k = 1$  to 3 do
    call RK_System( $m, n, h, (z_{ij}), (f_{ij})$ )
    output  $k, (z_{im})$ 
end for
for  $k = 4$  to nsteps do
    call AM_System( $m, n, h, est, (z_{ij}), (f_{ij})$ )
    output  $k, (z_{im})$ 
    output est
end for
for  $i = 0$  to  $n$  do
     $x_i \leftarrow z_{im}$ 
end for
deallocate array ( $f, z$ )
end procedure AMRK

```

El método de Adams-Moulton para un sistema y el cálculo del error de un solo paso se realizan con el siguiente seudocódigo:

```

procedure Sistema_AM(m, n, h, est, (zij), (fij))
integer i, j, k, m, mp1; real d, dmax, est, h
real array (zij)0:n×0:4, (fij)0:n×0:4
allocate real array (si)0:n, (yi)0:n
real array (ai)1:4 ← (55, -59, 37, -9)
real array (bi)1:4 ← (9, 19, -5, 1)
mp1 ← (1 + m) mod 5
call XP_System(n, (zim), (fim))
for i = 0 to n do
    si ← 0
end for
for k = 1 to 4 do
    j ← (m - k + 6) mod 5
    for i = 0 to n do
        si ← si + ak fij
    end for
end for
for i = 0 to n do
    yi ← zim + h si / 24
end for
call XP_System(n, (yi), (fi,mp1))
for i = 0 to n do
    si ← 0
end for
for k = 1 to 4 do
    j ← (mp1 - k + 6) mod 5
    for i = 0 to n do
        si ← si + bk fij
    end for
end for
for i = 0 to n do
    zi,mp1 ← zim + h si / 24
end for
m ← mp1
dmáx ← 0
for i = 0 to n do
    d ← |zim - yi| / |zim|
    if d > dmax then
        dmax ← d
        j ← i
    end if
end for
est ← 19dmax / 270
deallocate array (s, y)
end procedure Sistema_AM

```

Aquí, las evaluaciones de la función se almacenan cíclicamente en f_{im} para usarse con las fórmulas (2) y (3). Son posibles diversas técnicas de optimización en este seudocódigo. Por ejemplo, el programador puede desear mover el cálculo $\frac{1}{24}h$ fuera de los ciclos.

Se necesita un procedimiento compañero de Runge-Kutta, que es una modificación del procedimiento de *Sistema2_RK4* de la sección 11.1:

```

procedure Sistema_RK( $m, n, h, (z_{ij}), (f_{ij})$ )
integer  $i, m, mp1, n; \text{ real } h; \text{ real array } (z_{ij})_{0:n \times 0:4}, (f_{ij})_{0:n \times 0:4}$ 
allocate real array ( $g_{ij})_{0:n \times 0:3}, (y_i)_{0:n}$ 
 $mp1 \leftarrow (1 + m) \bmod 5$ 
call XP_System ( $n, (z_{im}), (f_{im})$ )
for  $i = 0$  to  $n$  do
     $y_i \leftarrow z_{im} + \frac{1}{2}h f_{im}$ 
end for
call XP_System ( $n, (y_i), (g_{i,1})$ )
for  $i = 0$  to  $n$  do
     $y_i \leftarrow z_{im} + \frac{1}{2}h g_{i,1}$ 
end for
call XP_System ( $n, (y_i), (g_{i,2})$ )
for  $i = 0$  to  $n$  do
     $y_i \leftarrow z_{im} + h g_{i,2}$ 
end for
call XP_System ( $n, (y_i), (g_{i,3})$ )
for  $i = 0$  to  $n$  do
     $z_{i,mp1} \leftarrow z_{im} + h[f_{im} + 2g_{i,1} + 2g_{i,2} + g_{i,3}]/6$ 
end for
 $m \leftarrow mp1$ 
deallocate array ( $g_{ij}), (y_i)$ 
end procedure Sistema_RK

```

Como antes, el programador podrá querer mover $\frac{1}{6}h$ fuera del ciclo.

Para utilizar el seudocódigo de Adams-Moulton, se suministra el procedimiento *Sistema_XP* que define el sistema de ecuaciones diferenciales ordinarias y se escribe un programa controlador con una llamada al procedimiento *AMRK*. El programa completo se compone de las siguientes cinco partes: el programa principal y los procedimientos *Sistema_XP*, *AMRK*, *Sistema_RK* y *Sistema_AM*.

Como ejemplo, el seudocódigo para el último ejemplo de la sección 11.2 (p. 479) es el siguiente:

```

program Prueba_AMRK
real  $h; \text{ real array } (x_i)_{0:n}$ 
integer  $n \leftarrow 5, nsteps \leftarrow 100$ 
real  $a \leftarrow 0, b \leftarrow 1$ 
 $(x_i) \leftarrow (1, 2, -4, -2, 7, 6)$ 
 $h \leftarrow (b - a)/nsteps$ 
call AMRK( $n, h, (x_i), nsteps$ )
end program Prueba_AMRK

```

```

procedure Sistema_XP (n,(xi),(fi))
integer n; real array (xi)0:n,(fi)0:n
f0  $\leftarrow$  1
f1  $\leftarrow$  x2
f2  $\leftarrow$  x1  $-$  x3  $-$  9x22  $+$  x43  $+$  6x5  $+$  2x0
f3  $\leftarrow$  x4
f4  $\leftarrow$  x5
f5  $\leftarrow$  x5  $-$  x2  $+$   $e^{x_1}$   $-$  x0
end procedure Sistema_XP

```

En este caso, hemos programado este procedimiento para un sistema autónomo de ecuaciones diferenciales ordinarias.

Un esquema adaptado

Puesto que hay una estimación del error en el método de Adams-Moulton, es natural remplazar el procedimiento *AMRK* con uno que emplea un esquema adaptado, es decir, uno que cambie el tamaño de paso. Aquí se describe un procedimiento similar al utilizado en la sección 10.3. El método de Runge-Kutta se utiliza para calcular los primeros tres pasos y luego se emplea el método de Adams-Moulton. Si la prueba de error determina que es necesario reducir a la mitad o duplicar el tamaño de paso en el primer paso usando el método de Adams-Moulton, entonces el tamaño de paso se reduce a la mitad o se duplica y todo el proceso comienza de nuevo con los valores iniciales, por lo que al menos se debe realizar un paso del método de Adams-Moulton. Si durante este proceso la prueba de error indica que se necesita reducir a la mitad en algún punto del intervalo $[a, b]$, entonces el tamaño de paso se reduce a la mitad. Se retira el valor calculado previamente y después de que se han calculado tres pasos de Runge-Kutta, el proceso continúa, usando de nuevo el método de Adams-Moulton, pero con el nuevo tamaño de paso. En otras palabras, el punto en el que el error fue demasiado grande se debe calcular con el método de Adams-Moulton, no con el de Runge-Kutta. Duplicar el tamaño de paso se maneja de una manera análoga. Duplicar el tamaño de paso sólo requiere guardar un número adecuado de valores anteriores; sin embargo, este proceso se puede simplificar (ya sea reducir a la mitad o duplicar el tamaño de paso) siempre guardando copias de dos pasos con el *viejo* tamaño de paso y luego usando esto como el punto inicial de un *nuevo* problema con valor inicial con el *nuevo* tamaño de paso. Se pueden diseñar otros procedimientos más complicados y pueden ser objeto de experimentación numérica (véase el problema de cómputo 11.3.3).

Un ejemplo de ingeniería

En ingeniería química, una compleja actividad de producción puede implicar varios reactores conectados con tuberías de entrada y salida. La concentración de un determinado producto químico en el *i*-ésimo reactor es una cantidad desconocida, *x_i*. Cada *x_i* es una función del tiempo. Si hay *n* reactores, todo el proceso se rige por un sistema de *n* ecuaciones diferenciales de la forma

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{V} \\ \mathbf{X}(0) &= \mathbf{S}, \quad \text{dada} \end{aligned}$$

donde \mathbf{X} es el vector que contiene las cantidades desconocidas x_i , \mathbf{A} es una matriz de $n \times n$ y \mathbf{V} es un vector constante. Las entradas en \mathbf{A} dependen de las tasas de flujo permitidas entre los diferentes reactores del sistema.

Hay varios métodos para resolver este problema. Uno de ellos es diagonalizar la matriz A encontrando una matriz P no singular para la que $P^{-1}AP$ es diagonal y después usar la función exponencial matricial para resolver el sistema en forma analítica. Esta es una tarea que puede manejar el software matemático. Por otra parte, podemos simplemente cambiar el problema a un *solucionador de EDO* y obtener la solución numérica. Una parte de la información que siempre se quiere en este tipo de problemas es una descripción del **estado estable** del sistema. Esto significa los valores de todas las variables en $t = \infty$. Cada función x_i debe ser una combinación lineal de funciones exponentiales de la forma $t \mapsto e^{\lambda t}$, en la que $\lambda < 0$. Aquí se presenta un ejemplo simple que puede ilustrar todo esto:

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} -8/3 & -4/3 & 1 \\ -17/3 & -4/3 & 1 \\ -35/3 & 14/3 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 12 \\ 29 \\ 48 \end{bmatrix} \quad (4)$$

Usando software matemático tal como Matlab, Maple o Mathematica podemos obtener una solución en forma cerrada:

$$\begin{aligned} x(t) &= \frac{1}{6}e^{-3t}(6 - 50e^t + 10e^{2t} + 34e^{3t}) \\ y(t) &= \frac{1}{6}e^{-3t}(12 - 125e^t + 40e^{2t} + 73e^{3t}) \\ z(t) &= \frac{1}{6}e^{-3t}(14 - 200e^t + 70e^{2t} + 116e^{3t}) \end{aligned}$$

Para un sistema de ecuaciones diferenciales ordinarias con un gran número de variables, puede ser más conveniente representarlo en un programa de computadora con un arreglo como \mathbf{x} (i , t) más que por nombres separados de las variables. Para ver el valor numérico de la solución analítica en un solo punto, por ejemplo, $t = 2.5$, obtenemos $x(2.5) \approx 5.74788$, $y(2.5) \approx 12.5746$, $z(2.5) \approx 20.0677$. También, podemos producir una representación gráfica de la solución analítica del problema.

Por último, los programas presentados en esta sección se pueden utilizar para generar una solución numérica en un intervalo dado con un número determinado de puntos.

Algunas observaciones acerca de las ecuaciones rígidas

En muchas aplicaciones de ecuaciones diferenciales hay varias funciones que se pueda *rastrear* juntas como funciones del tiempo. Se puede utilizar un sistema de ecuaciones diferenciales ordinarias para modelar los fenómenos físicos. En tal situación, puede ocurrir que diferentes funciones solución (o diferentes componentes de una solución única) tengan un comportamiento muy dispares que hace problemática la selección del tamaño de paso en la solución numérica. Por ejemplo, una componente de una función puede requerir un paso pequeño en la solución numérica, ya que está variando rápidamente, mientras que otra de las componentes puede variar lentamente y no requiere un pequeño tamaño de paso para su cálculo. Este sistema se dice que es **rígido**. La figura 11.2 ilustra una solución que varía lentamente rodeada de otras soluciones transitorias que decaen rápidamente.

Un ejemplo ilustrará esta posibilidad. Considere un sistema de dos ecuaciones diferenciales con condiciones iniciales:

$$\begin{cases} x' = -20x - 19y & x(0) = 2 \\ y' = -19x - 20y & y(0) = 0 \end{cases} \quad (5)$$

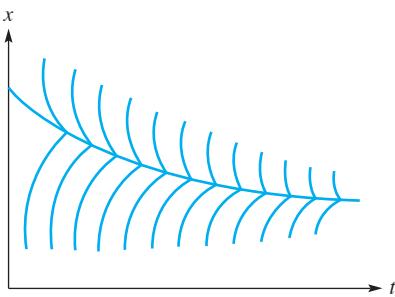


FIGURA 11.2
Curvas solución
para un ode
rígido

Se ve fácilmente que la solución es

$$\begin{aligned}x(t) &= e^{-39t} + e^{-t} \\y(t) &= e^{-39t} - e^{-t}\end{aligned}$$

La componente e^{-39t} decrece rápidamente conforme aumenta t , comenzando en 0. Entonces la solución está aproximadamente dada por $x(t) = -y(t) = e^{-t}$, y esta función es suave y decrece a 0. Parece ser que en casi cualquier solución numérica se podría utilizar un gran tamaño de paso. Sin embargo, vamos a examinar el más simple de los procedimientos numéricos: el método de Euler. Este genera la solución usando las siguientes ecuaciones:

$$\begin{aligned}x_{n+1} &= x_n + h(-20x_n - 19y_n) & x_0 &= 2 \\y_{n+1} &= y_n + h(-19x_n - 20y_n) & y_0 &= 0\end{aligned}$$

Estas ecuaciones en diferencias se puede resolver en forma cerrada y tenemos

$$\begin{aligned}x_n &= (1 - 39h)^n + (1 - h)^n \\y_n &= (1 - 39h)^n - (1 - h)^n\end{aligned}$$

Para que la solución numérica converja a 0 (y así imite a la solución real), es necesario que $h < \frac{2}{39}$. Si tuviéramos que resolver sólo la ecuación diferencial $x' = -x$ para obtener la solución $x(t) = e^{-t}$, el tamaño de paso podría ser tan grande como $h = 2$ para obtener el comportamiento correcto conforme t aumenta (véase el problema 11.3.2).

Para ver que el éxito numérico (en el sentido de ser capaz de usar un tamaño de paso razonable) depende del método utilizado, vamos a considerar el método de Euler implícito. Para una sola ecuación diferencial, este emplea la fórmula

$$x_{n+1} = x_n + hf(t_{n+1}, x_{n+1})$$

Puesto que x_{n+1} aparece en ambos lados de esta ecuación, se debe resolver la ecuación para x_{n+1} . En el ejemplo que se está considerando, las ecuaciones de Euler son

$$\begin{aligned}x_{n+1} &= x_n + h(-20x_{n+1} - 19y_{n+1}) \\y_{n+1} &= y_n + h(-19x_{n+1} - 20y_{n+1})\end{aligned}$$

Este par de ecuaciones tiene la forma $X_{n+1} = X_n + A\bar{X}_{n+1}$, donde A es la matriz de 2×2 del par de ecuaciones anteriores y \bar{X}_n es el vector, con componentes x_n y y_n . Esta ecuación matricial se puede escribir $(I - A)\bar{X}_{n+1} = \bar{X}_n$ o $\bar{X}_{n+1} = (I - A)^{-1}\bar{X}_n$. Una consecuencia es que la solución explícita es $\bar{X}_n = (I - A)^{-n}\bar{X}_0$. En este punto, es necesario recurrir a un resultado acerca de procesos iterativos. Para que \bar{X}_n converja a 0 para cualquier elección del vector inicial \bar{X}_0 es necesario y suficiente que

todos los valores propios de $(I - A)^{-1}$ sean de magnitud menor que uno (véase Kincaid y Cheney [2002]). De manera equivalente, los valores propios de $I - A$ deben ser de magnitud mayor que 1. Un cálculo sencillo muestra que para h positiva se cumple esta condición, sin hipótesis adicionales. Así, el método de Euler implícito se puede utilizar con cualquier tamaño de paso *razonable* en este problema. En los libros sobre ecuaciones rígidas se puede encontrar mucha más información y hay obras que abordan este tema a fondo. Algunas referencias esenciales son Dekker y Verwer [1984], Gear [1971], Miranker [1981] y Shampine y Gordon [1975].

En general, las ecuaciones diferenciales ordinarias rígidas son bastante difíciles de resolver. Ello se debe a que en la mayoría de los casos no se sabe de antemano si una ecuación diferencial ordinaria que se está tratando de resolver numéricamente es rígida. Los paquetes de software suelen tener solucionadores de ecuaciones diferenciales ordinarias diseñados específicamente para manejar ecuaciones diferenciales ordinarias rígidas. Algunos de estos procedimientos pueden variar tanto el tamaño de paso como el orden del método. En este tipo de algoritmos, la matriz jacobiana $\partial F/\partial X_y$ puede tener importancia. La solución de un sistema lineal asociado que implica la matriz jacobiana es fundamental para la confiabilidad y la eficiencia del código. La matriz jacobiana puede ser escasa, una indicación de que la función F no depende de algunas de las variables del problema.

Para los lectores que están interesados en la historia del análisis numérico le recomendamos el libro de Goldstine [1977]. El libro de texto de ecuaciones diferenciales de Moulton [1930] da una idea de los métodos numéricos utilizados antes de la llegada de las computadoras de alta velocidad. También (página 224) presenta algo de historia ¡que se remonta a Newton! El cálculo de órbitas de la mecánica celeste siempre ha sido un estímulo para la invención de métodos numéricos, como también lo ha sido las necesidades de la ciencia balística. Moulton menciona que el retraso de un proyectil debido a la resistencia del aire es una función de la velocidad muy complicada que necesita solución numérica de las demás ecuaciones simples de balística.

Resumen

(1) Por la forma autónoma de un sistema de ecuaciones diferenciales ordinarias en el notación vectorial

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}) \\ \mathbf{X}(a) = \mathbf{S}, \quad \text{dada} \end{cases}$$

el **método de Adams-Bashforth-Moulton de cuarto orden** es

$$\begin{aligned} \tilde{\mathbf{X}}(t+h) &= \mathbf{X}(t) + \frac{h}{24} \left\{ 55\mathbf{F}[\mathbf{X}(t)] - 59\mathbf{F}[\mathbf{X}(t-h)] + 37\mathbf{F}[\mathbf{X}(t-2h)] \right. \\ &\quad \left. - 9\mathbf{F}[\mathbf{X}(t-3h)] \right\} \\ \mathbf{X}(t+h) &= \mathbf{X}(t) + \frac{h}{24} \left\{ 9\mathbf{F}[\tilde{\mathbf{X}}(t+h)] + 19\mathbf{F}[\mathbf{X}(t)] - 5\mathbf{F}[\mathbf{X}(t-h)] \right. \\ &\quad \left. + \mathbf{F}[\mathbf{X}(t-2h)] \right\} \end{aligned}$$

En este caso, $\tilde{\mathbf{X}}(t+h)$ es el **predictor** y $\mathbf{X}(t+h)$ es el **corrector**. El método de Adams-Bashforth-Moulton necesita *cinco* evaluaciones de \mathbf{F} por paso. Con el vector inicial $\mathbf{X}(a)$ dado, los valores de $\mathbf{X}(a+h), \mathbf{X}(a+2h), \mathbf{X}(a+3h)$ se calcula por el método de Runge-Kutta de cuarto orden. Entonces, el método de Adams-Bashforth-Moulton se puede utilizar repetidamente. El valor predicho $\tilde{\mathbf{X}}$ se calcula a partir de los cuatro valores de \mathbf{X} en $t, t-h, t-2h$ y $t-3h$ y después el valor corregido de $\mathbf{X}(t+h)$ puede calcularse usando el valor de predicción $\tilde{\mathbf{X}}(t+h)$ y los valores evaluados previamente de \mathbf{F} en $t, t-h$ y $t-2h$.

Referencias adicionales

Véase Aiken [1985], Ascher y Petzold [1998], Boyce y DiPrima [2003], Butcher [1987], Carrier y Pearson [1991], Chicone [2006], Collatz [1966], Dekker y Verwer [1984], Edwards y Penny [2004], England [1969], Enright [2006], Fehlberg [1969], Gear [1971], Golub y Ortega [1992], Henrici [1962], Hull et al. [1972], Hundsdorfer [1985], Lambert [1973, 1991], Lapidus y Seinfeld [1971], Miranker [1981], Moulton [1930] y Shampine y Gordon [1975].

Problemas 11.3

1. Encuentre la solución general de este sistema al convertirlo en un sistema de cuatro ecuaciones de primer orden:

$$\begin{cases} x'' = \alpha y \\ y'' = \beta x \end{cases}$$

2. Compruebe las afirmaciones hechas sobre el tamaño de paso h en el análisis de ecuaciones rígidas.

Problemas de cómputo 11.2

1. Pruebe el procedimiento *AMRK* en el sistema dado en el problema de cómputo 11.2.2.
2. El error de un solo paso se controla cercanamente usando fórmulas de cuarto orden; sin embargo, el error de redondeo en la realización de los cálculos de las ecuaciones (3) y (4) puede ser grande. Es lógico realizarlos en lo que se conoce como aritmética de **doble precisión parcial**. La función \mathbf{F} sería evaluada con precisión simple en los puntos deseados $X(t + ih)$, pero la combinación lineal $\sum_i c_i \mathbf{F}(X(t + ih))$ sería acumulada en doble precisión. También, la adición de $X(t)$ a este resultado se realiza con doble precisión. Recodifique el método de Adams-Moulton de manera que se utilice aritmética de doble precisión parcial. Compare este código con el del libro para un sistema con una solución conocida. ¿Cómo se comparan con respecto al redondeo de error en cada paso?
3. Escriba y pruebe un proceso similar adaptado a *Adaptativo_RK45* en la sección 10.3 con la secuencia de llamada

```
procedure Adaptativo_AMRK( $n, h, t_a, t_b, (x_i), itmax, \varepsilon_{\min}, \varepsilon_{\max}, h_{\min}, h_{\max}, iflag$ )
```

Esta rutina debe realizar el procedimiento adaptado descrito en esta sección y se utilizará en lugar del procedimiento *AMRK*.

4. Resuelva el problema depredador-presa del ejemplo al principio de este capítulo con $a = -10^{-2}$, $b = -\frac{1}{4} \times 10^2$, $c = 10^{-2}$ y $d = -10^2$ y con valores iniciales $u(0) = 80$, $v(0) = 30$. Trace la gráfica u (la presa) y v (el depredador) como funciones del tiempo t .
5. Resuelva y grafique la solución numérica del sistema de ecuaciones diferenciales ordinarias dado por la ecuación (4) usando software matemático como Matlab, Maple o Mathematica.

6. (Continuación) Repita para la ecuación (5) usando una rutina diseñada específicamente para manejar ecuaciones diferenciales ordinarias rígidas.
7. Resuelva los siguientes problemas de prueba y trace la gráfica de sus curvas solución.

- a. Este problema corresponde a una órbita estable descubierta recientemente que surge en el problema restringido de tres cuerpos en el que las órbitas son coplanares. Las dos coordenadas espaciales del j ésmo cuerpo son x_{ij} y x_{2j} para $j = 1, 2, 3$. Cada una de las seis coordenadas satisface una ecuación diferencial de segundo orden:

$$x_{ij}'' = \sum_{\substack{k=1 \\ k \neq j}}^3 m_k (x_{ik} - x_{ij}) / d_{jk}^3$$

donde $d_{jk}^2 = \sum_{i=1}^2 (x_{ij} - x_{ik})^2$ para $k, j = 1, 2, 3$. Suponga que los cuerpos tienen la misma masa, por ejemplo, $m_1 = m_2 = m_3 = 1$ y con las condiciones iniciales adecuadas, que seguirán la misma órbita con figura de ocho como una solución periódica de estado estable. Cuando el sistema se reescribe como un sistema de primer orden, la dimensión del problema es 12 y las condiciones iniciales en $t = 0$ están dadas por

$$\begin{cases} x_{11} = -0.97000436 & x'_{11} = 0.466203685 \\ x_{21} = 0.24308753 & x'_{21} = 0.43236573 \\ x_{12} = 0.0 & x'_{12} = -0.93240737 \\ x_{22} = 0.0 & x'_{22} = -0.86473146 \\ x_{13} = 0.97000436 & x'_{13} = 0.466203685 \\ x_{23} = -0.24308753 & x'_{23} = 0.43236573 \end{cases}$$

Resuelva el problema para $t \in [0, 20]$.

- b. El problema de Lorenz es bien conocido y surge en el estudio de sistemas dinámicos:

$$\begin{cases} x'_1 = 10(x_2 - x_1) \\ x'_2 = x_1(28 - x_3) - x_2 \\ x'_3 = x_1x_2 - \frac{8}{3}x_3 \\ x_1(0) = 15, x_2(0) = 15, x_3(0) = 36 \end{cases}$$

Resuelva el problema para $t \in [0, 20]$. Se sabe que tienen soluciones que están potencialmente mal condicionadas.

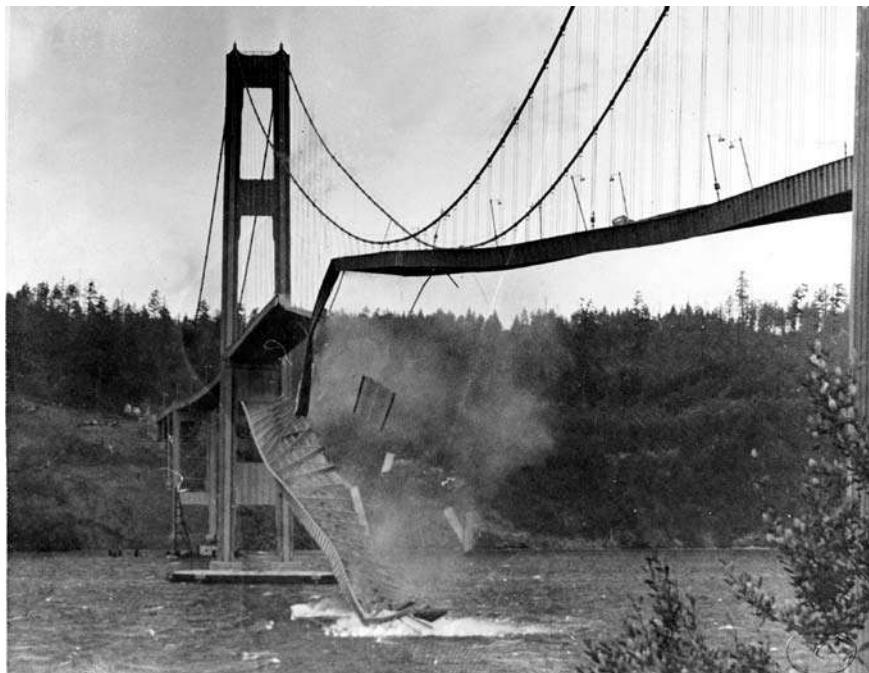
Para detalles adicionales sobre estos problemas, véase Enright [2006].

8. Escriba un programa de cómputo basado en el seudocódigo *Test_AMRK* para encontrar la solución numérica de sistemas de ecuaciones diferenciales ordinarias y compare los resultados con los que se obtienen al usar una rutina incorporada como la que se puede encontrar en Matlab, Maple o Mathematica. Trace la curva solución resultante.
9. (**Proyecto del puente Tacoma Narrows**) En 1940, el tercer puente colgante más largo del mundo se derrumbó con un fuerte viento. El siguiente sistema de ecuaciones diferenciales es un modelo matemático que trata de explicar cómo las oscilaciones de torsión pueden ser

amplificadas y causar tal calamidad:

$$\begin{cases} y'' = -y'd - [K/(ma)] [e^{a(y-\ell \sin \theta)} - 1 + e^{a(y+\ell \sin \theta)} - 1] + 0.2W \sin \omega t \\ \theta' = -\theta y'd + (3 \cos \theta/\ell) [K/(ma)] [e^{a(y-\ell \sin \theta)} - e^{a(y+\ell \sin \theta)}] \end{cases}$$

El último término en la ecuación y es el término que obliga a que el viento W , agregue una oscilación estrechamente vertical del puente. En este caso, la carretera tiene un ancho 2ℓ colgando entre dos cables de suspensión, y es la distancia actual del centro de la carretera que cuelga por debajo de su punto de equilibrio y θ es el ángulo que hace la carretera con la horizontal. También, se utiliza la ley de Newton $F = ma$ y la constante de Hooke K . Explore cómo se usan los solucionadores de EDO para generar trayectorias numéricas para la configuración de diferentes parámetros. Ilustre los diferentes tipos de fenómenos que hay en este modelo. Para más detalles, véase McKenna y Tuama [2001] y Sauer [2006].



Suavizado de datos y el método de mínimos cuadrados

La tensión superficial S en un líquido es una función lineal de la temperatura T . Para un líquido especial, se han hecho mediciones de la tensión superficial para determinadas temperaturas. Los resultados fueron los siguientes:

T	0	10	20	30	40	80	90	95
S	68.0	67.1	66.4	65.6	64.6	61.8	61.0	60.0

¿Cómo se pueden determinar los valores más probables de las constantes en la ecuación siguiente?

$$S = aT + b$$

Los métodos para la resolución de estos problemas se desarrollaran en este capítulo.

12.1 Método de mínimos cuadrados

Recta de mínimos cuadrados

En las ciencias experimentales, sociales y de la conducta, un experimento o una encuesta con frecuencia produce una gran cantidad de datos. Para interpretar los datos, el investigador puede recurrir a métodos gráficos. Por ejemplo, un experimento en física puede producir una tabla numérica de la forma

$$\begin{array}{c|c|c|c|c|c} x & x_0 & x_1 & \cdots & x_m \\ \hline y & y_0 & y_1 & \cdots & y_m \end{array} \quad (1)$$

y de esta, se pueden ubicar $m + 1$ puntos en una gráfica. Supongamos que la gráfica resultante se parece a la de la figura 12.1. Una razonable conclusión tentativa es que la función subyacente es *lineal* y que la falla de los puntos de encontrarse *precisamente* en una línea recta se debe al error experimental. Si se procede en este supuesto o si existen razones teóricas para creer que la función es en realidad lineal, lo siguiente es determinar la función correcta. Suponiendo que

$$y = ax + b$$

¿cuáles son los coeficientes a y b ? Pensando geométricamente, nos preguntamos: *¿qué recta pasa más cerca de los ocho puntos graficados?*

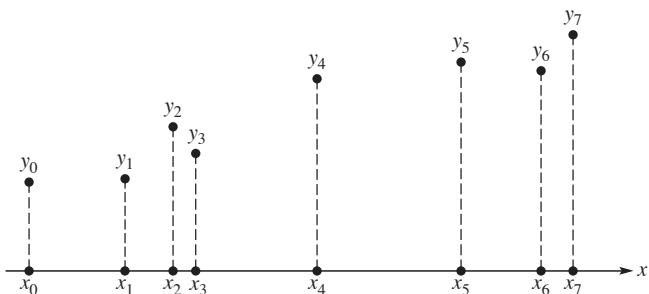


FIGURA 12.1
Datos experimentales

Para responder a esta pregunta, suponga que se realiza una hipótesis acerca de los valores correctos de a y b . Esto equivale a decidir sobre una recta específica para representar los datos. En general, los puntos de los datos no se encuentran en la recta $y = ax + b$. Si por casualidad el k -ésimo dato se encuentra en la recta, entonces

$$ax_k + b - y_k = 0$$

Si no es así, entonces hay una discrepancia o *error* de magnitud

$$|ax_k + b - y_k|$$

El error total absoluto para todos los $m + 1$ puntos es, por tanto

$$\sum_{k=0}^m |ax_k + b - y_k|$$

Esta es una función de a y b , y sería razonable elegir a y b de modo que la función tome su valor mínimo. Este problema es un ejemplo de **aproximación ℓ_1** y se puede resolver con las técnicas de programación lineal, un tema que se trata en el capítulo 17 (los métodos del cálculo no funcionan en esta función porque generalmente no es derivable).

En la práctica, es común minimizar una función de error diferente de a y de b :

$$\varphi(a, b) = \sum_{k=0}^m (ax_k + b - y_k)^2 \quad (2)$$

Esta función es adecuada debido a consideraciones estadísticas. De forma explícita, si los errores siguen una *distribución de probabilidad normal*, entonces la minimización de φ produce una mejor estimación de a y b . Esto se llama una **aproximación ℓ_2** . Otra ventaja es que los métodos de cálculo se pueden usar en la ecuación (2).

Las aproximaciones ℓ_1 y ℓ_2 están relacionadas con casos específicos de la **norma ℓ_p** definida por

$$\|x\|_p = \left\{ \sum_{i=1}^n |x_i|^p \right\}^{1/p} \quad (1 \leq p < \infty)$$

para el vector $x = [x_1, x_2, \dots, x_n]^T$.

Vamos a tratar de hacer $\varphi(a, b)$ un mínimo. Con cálculo, las condiciones

$$\frac{\partial \varphi}{\partial a} = 0 \quad \frac{\partial \varphi}{\partial b} = 0$$

(derivadas parciales de φ con respecto a a y b , respectivamente) son *necesarias* en el mínimo. Obteniendo las derivadas de la ecuación (2), queda

$$\begin{cases} \sum_{k=0}^m 2(ax_k + b - y_k)x_k = 0 \\ \sum_{k=0}^m 2(ax_k + b - y_k) = 0 \end{cases}$$

Este es un par de ecuaciones lineales simultáneas con incógnitas a y b . Se les llama **ecuaciones normales** y se pueden escribir como

$$\begin{cases} \left(\sum_{k=0}^m x_k^2 \right)a + \left(\sum_{k=0}^m x_k \right)b = \sum_{k=0}^m y_k x_k \\ \left(\sum_{k=0}^m x_k \right)a + (m+1)b = \sum_{k=0}^m y_k \end{cases} \quad (3)$$

Aquí, por supuesto, $\sum_{k=0}^m 1 = m+1$, que es el número de puntos de datos. Para simplificar la notación, hacemos

$$p = \sum_{k=0}^n x_k \quad q = \sum_{k=0}^n y_k \quad r = \sum_{k=0}^n x_k y_k \quad s = \sum_{k=0}^n x_k^2$$

El sistema de ecuaciones (3) es ahora

$$\begin{bmatrix} s & p \\ p & m+1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} r \\ q \end{bmatrix}$$

Hemos resuelto este par de ecuaciones con eliminación gaussiana y obtenido el siguiente algoritmo. Como alternativa, puesto que se trata de un sistema lineal de 2×2 , podemos usar la regla de Cramer* para resolverlo. El determinante de la matriz de coeficientes es

$$d = \text{Det} \begin{bmatrix} s & p \\ p & m+1 \end{bmatrix} = (m+1)s - p^2$$

Además, se obtiene

$$\begin{aligned} a &= \frac{1}{d} \text{Det} \begin{bmatrix} r & p \\ q & m+1 \end{bmatrix} = \frac{1}{d} [(m+1)r - pq] \\ b &= \frac{1}{d} \text{Det} \begin{bmatrix} s & r \\ p & q \end{bmatrix} = \frac{1}{d} [sq - pr] \end{aligned}$$

Podemos escribir esto como un algoritmo.

■ ALGORITMO 1 Recta de mínimos cuadrados

Los coeficientes de la recta de mínimos cuadrados $y = ax + b$ que pasa por el conjunto de $m+1$ puntos de datos (x_k, y_k) para $k = 0, 1, 2, \dots, m$ se calculan (en orden) como sigue:

1. $p = \sum_{k=0}^m x_k$
2. $q = \sum_{k=0}^m y_k$

*La regla de Cramer se presenta en el apéndice D.

- 3.** $r = \sum_{k=0}^m x_k y_k$
4. $s = \sum_{k=0}^m x_k^2$
5. $d = (m + 1)s - p^2$
6. $a = [(m + 1)r - pq]/d$
7. $b = [sq - pr]/d$

Otra forma de este resultado es

$$\begin{aligned} a &= \frac{1}{d} \left[(m + 1) \left(\sum_{k=0}^m x_k y_k \right) - \left(\sum_{k=0}^m x_k \right) \left(\sum_{k=0}^m y_k \right) \right] \\ b &= \frac{1}{d} \left[\left(\sum_{k=0}^m x_k^2 \right) \left(\sum_{k=0}^m y_k \right) - \left(\sum_{k=0}^m x_k \right) \left(\sum_{k=0}^m x_k y_k \right) \right] \end{aligned} \quad (4)$$

donde

$$d = (m + 1) \left(\sum_{k=0}^m x_k^2 \right) - \left(\sum_{k=0}^m x_k \right)^2$$

Ejemplo lineal

El análisis anterior muestra el procedimiento de **mínimos cuadrados** en el caso lineal simple.

EJEMPLO 1 Como un ejemplo específico, encuentre la solución lineal de mínimos cuadrados para la siguiente tabla de valores:

x	4	7	11	13	17
y	2	0	2	6	7

Trace los puntos originales y la recta usando un conjunto más fino de puntos de cuadricula.

Solución Las ecuaciones en el algoritmo 1 conducen a este sistema de dos ecuaciones:

$$\begin{cases} 644a + 52b = 227 \\ 52a + 5b = 17 \end{cases}$$

cuyas soluciones son $a = 0.4864$ y $b = -1.6589$. Por medio de la ecuación (3) se obtiene el valor $\varphi(a, b) = 10.7810$. En la figura 12.2 se muestra una gráfica de los datos dados y la recta de los mínimos cuadrados lineales.

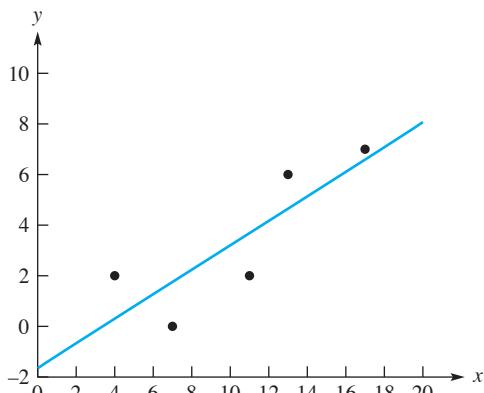


FIGURA 12.2
Recta de
mínimos
cuadrados

Podemos utilizar software matemático como Matlab, Maple o Mathematica para ajustar a los datos a un polinomio lineal de mínimos cuadrados y para comprobar el valor de φ (véase el problema de cómputo 12.1.5).

Para entender lo que está sucediendo en este caso, queremos determinar la ecuación de una recta de la forma $y = ax + b$ que se ajusta mejor a los datos en el sentido de los mínimos cuadrados. Con cuatro puntos de datos (x_i, y_i) , tenemos cuatro ecuaciones $y_i = ax_i + b$ para $i = 1, 2, 3, 4$ que se pueden escribir como

$$\mathbf{Ax} = \mathbf{y}$$

donde

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

En general, queremos resolver un sistema lineal

$$\mathbf{Ax} = \mathbf{b}$$

donde A es una matriz de $m \times n$ y $m > n$. La solución coincide con la solución de las **ecuaciones normales**

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}$$

Esto corresponde a minimizar $\|\mathbf{Ax} - \mathbf{b}\|_2^2$.

Ejemplo no polinomial

El método de los mínimos cuadrados no se limita a polinomios lineales (de primer grado) ni a ninguna forma funcional específica. Supongamos, por ejemplo, que queremos ajustar una tabla de valores (x_k, y_k) , donde $k = 0, 1, \dots, m$, mediante una función de la forma

$$y = a \ln x + b \cos x + ce^x$$

en el sentido de mínimos cuadrados. Las incógnitas en este problema son los tres coeficientes a , b y c . Consideraremos la función

$$\varphi(a, b, c) = \sum_{k=0}^m (a \ln x_k + b \cos x_k + ce^{x_k} - y_k)^2$$

y hacemos $\partial\varphi/\partial a = 0$, $\partial\varphi/\partial b = 0$ y $\partial\varphi/\partial c = 0$. Esto se traduce en las siguientes tres ecuaciones normales:

$$\left\{ \begin{array}{lcl} a \sum_{k=0}^m (\ln x_k)^2 + b \sum_{k=0}^m (\ln x_k)(\cos x_k) + c \sum_{k=0}^m (\ln x_k)e^{x_k} & = & \sum_{k=0}^m y_k \ln x_k \\ a \sum_{k=0}^m (\ln x_k)(\cos x_k) + b \sum_{k=0}^m (\cos x_k)^2 + c \sum_{k=0}^m (\cos x_k)e^{x_k} & = & \sum_{k=0}^m y_k \cos x_k \\ a \sum_{k=0}^m (\ln x_k)e^{x_k} + b \sum_{k=0}^m (\cos x_k)e^{x_k} + c \sum_{k=0}^m (e^{x_k})^2 & = & \sum_{k=0}^m y_k e^{x_k} \end{array} \right.$$

EJEMPLO 2 Ajuste una función de la forma $y = a \ln x + b \cos x + ce^x$ a los valores de la siguiente tabla:

x	0.24	0.65	0.95	1.24	1.73	2.01	2.23	2.52	2.77	2.99
y	0.23	-0.26	-1.10	-0.45	0.27	0.10	-0.29	0.24	0.56	1.00

Solución Usando la tabla y las ecuaciones anteriores obtenemos el sistema de 3×3

$$\begin{cases} 6.79410a - 5.34749b + 63.25889c = 1.61627 \\ -5.34749a + 5.10842b - 49.00859c = -2.38271 \\ 63.25889a - 49.00859b + 1002.50650c = 26.77277 \end{cases}$$

Tiene la solución $a = -1.04103$, $b = -1.26132$ y $c = 0.03073$. Por tanto, la curva

$$y = -1.04103 \ln x - 1.26132 \cos x + 0.03073e^x$$

tiene la forma requerida y se ajusta a la tabla en el sentido de mínimos cuadrados. El valor de $\varphi(a, b, c)$ es 0.92557. La figura 12.3 es un gráfica de los datos dados y de la curva no polinomial de mínimos cuadrados.

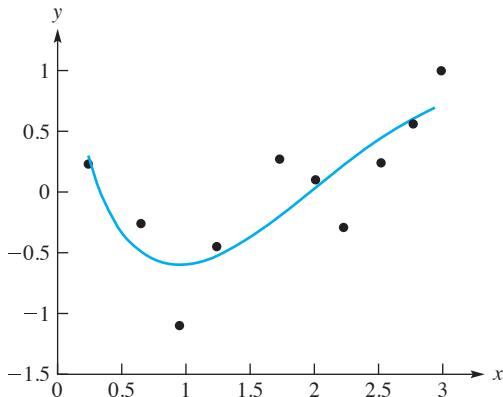


FIGURA 12.3
Mínimos
cuadrados no
polinomial



Podemos utilizar software matemático como Matlab, Maple o Mathematica para verificar estos resultados y para trazar la curva solución (véase el problema de cómputo 12.1.6).

Funciones base $\{g_0, g_1, \dots, g_n\}$

El principio de los mínimos cuadrados, que se ilustra en estos dos casos sencillos, se puede extender a familias de funciones lineales en general sin implicar cualquier idea nueva. Supongamos que los datos de la ecuación (1) obedecen una relación tal como

$$y = \sum_{j=0}^n c_j g_j(x) \tag{5}$$

en el que las funciones, g_0, g_1, \dots, g_n (llamadas **funciones base**) se conocen y se conservan fijas. Se determinarán los coeficientes c_0, c_1, \dots, c_n de acuerdo con el principio de mínimos cuadrados

En otras palabras, se define la expresión

$$\varphi(c_0, c_1, \dots, c_n) = \sum_{k=0}^m \left[\sum_{j=0}^n c_j g_j(x_k) - y_k \right]^2 \quad (6)$$

y se seleccionan los coeficientes para que quede lo más pequeña posible. Por supuesto, la expresión $\varphi(c_0, c_1, \dots, c_n)$ es la suma de los cuadrados de los errores asociados con cada entrada (x_k, y_k) en la tabla dada.

Procediendo como antes, escribimos como condiciones necesarias para minimizar las n ecuaciones

$$\frac{\partial \varphi}{\partial c_i} = 0 \quad (0 \leq i \leq n)$$

Estas derivadas parciales se obtienen de la ecuación (7). En realidad,

$$\frac{\partial \varphi}{\partial c_i} = \sum_{k=0}^m 2 \left[\sum_{j=0}^n c_j g_j(x_k) - y_k \right] g_i(x_k) \quad (0 \leq i \leq n)$$

Cuando se hacen igual a cero, las ecuaciones resultantes se puede rearreglar como

$$\sum_{j=0}^n \left[\sum_{k=0}^m g_i(x_k) g_j(x_k) \right] c_j = \sum_{k=0}^m y_k g_i(x_k) \quad (0 \leq i \leq n) \quad (7)$$

Estas son las **ecuaciones normales** en esta situación y sirven para determinar los mejores valores de los parámetros c_0, c_1, \dots, c_n . Las ecuaciones normales son lineales en c_i ; así, en principio, se pueden determinar por el método de eliminación gaussiana (véase el capítulo 7).

En la práctica, las ecuaciones normales pueden ser difíciles de resolver si no se tiene cuidado con la elección de las funciones base g_0, g_1, \dots, g_n . Primero, el conjunto $\{g_0, g_1, \dots, g_n\}$ debe ser **linealmente independiente**. Esto significa que ninguna combinación lineal $\sum_{i=0}^n c_i g_i$ puede ser la función cero (excepto en el caso trivial cuando $c_0 = c_1 = \dots = c_n = 0$). En segundo lugar, las funciones g_0, g_1, \dots, g_n deben ser *adecuadas* para el problema de que se trata. Por último, se debe optar por un conjunto de funciones base que esté *bien condicionado* para el trabajo numérico. Trabajaremos en este aspecto del problema en la siguiente sección.

Resumen

(1) Queremos encontrar una recta $y = ax + b$ que pase lo más cerca posible de los $m + 1$ pares de puntos (x_i, y_i) para $0 \leq i \leq m$. Un ejemplo de la **aproximación ℓ_1** es elegir a y b de modo que el error absoluto total de todos estos puntos se minimice:

$$\sum_{k=0}^m |ax_k + b - y_k|$$

Esto se puede resolver con las técnicas de programación lineal.

(2) Una **aproximación ℓ_2** minimiza una función de error diferente de a y b :

$$\varphi(a, b) = \sum_{k=0}^m (ax_k + b - y_k)^2$$

La minimización de φ produce una mejor estimación de a y b en el sentido de mínimos cuadrados. Se resuelven las **ecuaciones normales**

$$\begin{cases} \left(\sum_{k=0}^m x_k^2 \right) a + \left(\sum_{k=0}^m x_k \right) b = \sum_{k=0}^m y_k x_k \\ \left(\sum_{k=0}^m x_k \right) a + (m+1)b = \sum_{k=0}^m y_k \end{cases}$$

(3) En un caso más general, los puntos se ajustan a una relación como

$$y = \sum_{j=0}^n c_j g_j(x)$$

en la que las **funciones base** g_0, g_1, \dots, g_n se conocen y se conservan fijas. Los coeficientes c_0, c_1, \dots, c_n , se determinarán de acuerdo con el principio de los mínimos cuadrados. Las **ecuaciones normales** en esta situación son

$$\sum_{j=0}^n \left[\sum_{k=0}^m g_i(x_k) g_j(x_k) \right] c_j = \sum_{k=0}^m y_k g_i(x_k) \quad (0 \leq i \leq n)$$

y se pueden resolver, en principio, por el método de eliminación gaussiana para determinar los mejores valores de los parámetros c_0, c_1, \dots, c_n .

Problemas 12.1

- 1.** Usando el método de mínimos cuadrados, encuentre la función constante que mejor se ajuste a los siguientes datos:

x	-1	2	3
y	$\frac{5}{4}$	$\frac{4}{3}$	$\frac{5}{12}$

- 2.** Determine la función *constante* c que se produce mediante la teoría de mínimos cuadrados aplicada a la tabla de la página 495. ¿La fórmula resultante implica los puntos x_k de alguna manera? Aplique su fórmula general para el problema anterior.
- 3.** Encuentre una ecuación de la forma $y = ae^{x^2} + bx^3$ que mejor se ajuste a los puntos $(-1, 0)$, $(0, 1)$ y $(1, 2)$ en el sentido de mínimos cuadrados.
- 4.** Supongamos que los puntos x en la tabla (1) están situados simétricamente respecto a 0 en el eje x . En este caso, hay una fórmula especialmente simple de la recta que mejor se adapta a los puntos. Encuéntrela.
- 5.** Encuentre la ecuación de una parábola de la forma $y = ax^2 + b$ que mejor representa los datos siguientes. Utilice el método de mínimos cuadrados.

x	-1	0	1
y	3.1	0.9	2.9

- 6.** Supongamos que se sabe que la tabla (1) se ajusta a una función similar a $y = x^2 - x + c$. ¿Qué valor de c se obtiene por la teoría de mínimos cuadrados?

^a7. Supongamos que la tabla (1) se representa por medio de una función $y = c \log x$. Si es así, ¿qué valor de c surge de la teoría de mínimos cuadrados?

8. Muestre que la ecuación (4) es la solución de la ecuación (3).

9. (Continuación) ¿Cómo sabemos que el divisor d no es cero? De hecho, demuestre que d es positivo para $m \geq 1$. *Sugerencia:* demuestre que

$$d = \sum_{k=0}^m \sum_{l=0}^{k-1} (x_k - x_l)^2$$

por inducción sobre m . La desigualdad de Cauchy-Schwarz también se puede utilizar para demostrar que $d > 0$.

10. (Continuación) Demuestre que a y b también se pueden calcular como sigue:

$$\begin{aligned}\hat{x} &= \frac{1}{m+1} \sum_{k=0}^m x_k & \hat{y} &= \frac{1}{m+1} \sum_{k=0}^m y_k \\ c &= \sum_{k=0}^m (x_k - \hat{x})^2 & a &= \frac{1}{c} \sum_{k=0}^m (x_k - \hat{x})(y_k - \hat{y}) & b &= \hat{y} - a\hat{x}\end{aligned}$$

Sugerencia: demuestre que $d = (m+1)c$.

^a11. ¿Cómo sabemos que los coeficientes c_0, c_1, \dots, c_n que satisfacen las ecuaciones normales (7) no conducen a un máximo en la función definida por la ecuación (6)?

^a12. Si se piensa que la tabla (1) se ajusta a una relación $y = \log(cx)$, ¿cuál es el valor de c obtenido por el método de mínimos cuadrados?

^a13. ¿Cuál es la línea recta que mejor se ajusta a los siguientes datos

x	1	2	3	4
y	0	1	1	2

en el sentido de mínimos cuadrados?

14. En geometría analítica, aprendimos que la distancia de un punto (x_0, y_0) a una recta representada por la ecuación $ax + by = c$ es $(ax_0 + by_0 - c)(a^2 + b^2)^{-1/2}$. Determine una línea recta que se ajusta a una tabla de puntos de datos (x_i, y_i) , para $0 \leq i \leq m$, de tal manera que se minimice la suma de los cuadrados de las distancias de los puntos a la recta.

15. Demuestre que si una línea recta se ajusta a una tabla (x_i, y_i) por el método de los mínimos cuadrados, la recta pasará por el punto (x^*, y^*) , donde x^* y y^* corresponden a los promedios de las x_i y y_i , respectivamente.

^a16. La viscosidad V de un líquido varía con la temperatura de acuerdo con una ley cuadrática $V = a + bT + cT^2$. Encuentre los mejores valores de a , b y c para la siguiente tabla:

T	1	2	3	4	5	6	7
V	2.31	2.01	1.80	1.66	1.55	1.47	1.41

17. Un experimento consta de dos variables independientes x y y y una variable dependiente z . ¿Cómo puede una función $z = a + bx + cy$ ajustarse a la tabla de puntos (x_k, y_k, z_k) ? Presente las ecuaciones normales.

18. Encuentre la mejor función (en el sentido de mínimos cuadrados) que se ajuste a los puntos de datos siguientes y sea de la forma $f(x) = a \operatorname{sen} \pi x + b \cos \pi x$:

x	-1	$-\frac{1}{2}$	0	$\frac{1}{2}$	1
y	-1	0	1	2	1

19. Encuentre el polinomio cuadrático que mejor se ajusta a los datos siguientes en el sentido de los mínimos cuadrados:

x	-2	-1	0	1	2
y	2	1	1	1	2

20. ¿Qué recta representa mejor los datos siguientes en el sentido de mínimos cuadrados?

x	0	1	2
y	5	-6	7

21. ¿Qué constante c hace que la expresión

$$\sum_{k=0}^m [f(x_k) - ce^{x_k}]^2$$

sea lo más pequeña posible?

22. Demuestre que la fórmula para la recta que mejor ajusta los datos (k, y_k) en los enteros k de $1 \leq k \leq n$ es

$$y = ax + b$$

donde

$$a = \frac{6}{n(n^2 - 1)} \left[2 \sum_{k=1}^n ky_k - (n + 1) \sum_{k=1}^n y_k \right]$$

$$b = \frac{2}{n(n - 1)} \left[(2n + 1) \sum_{k=1}^n y_k - 3 \sum_{k=1}^n ky_k \right]$$

23. Establezca las ecuaciones normales y compruebe los resultados del ejemplo 1.

24. Se afirma que un vector v es la solución de mínimos cuadrados de un sistema inconsistente $\mathbf{A}x = \mathbf{b}$. ¿Cómo podemos probar v sin pasar por todo el procedimiento de mínimos cuadrados?

25. Encuentre las ecuaciones normales para los siguientes puntos de datos:

x	1.0	2.0	2.5	3.0
y	3.7	4.1	4.3	5.0

Determine la línea recta que mejor se ajusta a los datos en el sentido de mínimos cuadrados. Trace la gráfica de los puntos y de la recta de mínimos cuadrados.

26. Para el caso $n = 4$, demuestre directamente que al formar las ecuaciones normales a partir de los puntos de datos (x_i, y_i) se obtienen los resultados del teorema 1.

Problemas de cómputo 12.1

- Escriba un procedimiento que establezca las ecuaciones normales (7). Usando este procedimiento y otras rutinas como *Gauss* y *Solve* del capítulo 7, compruebe la solución dada para el problema que implica $\ln x$, $\cos x$ y e^x en la subsección titulada “Ejemplo no polinomial”.
- Escriba un procedimiento que ajuste una línea recta a la tabla (1). Luego úselo para encontrar las constantes en la ecuación $S = at + b$ para la tabla del ejemplo de inicio de este capítulo. También compruebe los resultados obtenidos para el problema en la subsección titulada “Ejemplo lineal”.
- Escriba y pruebe un programa que tome $m + 1$ puntos en el plano (x_i, y_i) , donde $0 \leq i \leq m$, con $x_0 < x_1 < \dots < x_m$, y calcule el mejor ajuste lineal con el método de mínimos cuadrados. Después, el programa debe crear una gráfica de puntos y de la mejor recta determinada por el método de mínimos cuadrados.
- El Servicio de rentas internas (IRS) publica la siguiente tabla de valores que se tienen que hacer con las distribuciones mínimas de planes de pensiones:

x	1	2	3	4	5	6	7	8	
y	29.9	29.0	28.1	27.1	26.2	25.3	24.4	23.6	
	9	10	11	12	13	14	15	16	
	22.7	21.8	21.0	20.1	19.3	18.5	17.7	16.9	

¿Qué función simple representa los datos? Utilice la ecuación (5) y trace la gráfica de los datos y de los resultados ya sea usando software de trazado tal como gnuplot o algún sistema de software de matemáticas como Maple, Matlab o Mathematica.

- Usando software matemático como Matlab, Maple o Mathematica, ajuste un polinomio lineal con mínimos cuadrados a los datos del ejemplo 1. Después trace la gráfica de los datos originales y del polinomio usando un fino conjunto de puntos de cuadrícula.
- (Continuación) Compruebe los resultados del ejemplo 2 y trace la curva.

12.2 Sistemas ortogonales y polinomios de Chebyshev

Funciones base ortonormales $\{g_0, g_1, \dots, g_n\}$

Una vez que se han elegido las funciones g_0, g_1, \dots, g_n de la ecuación (5) en la sección 12.1, el problema de mínimos cuadrados se puede interpretar como sigue. El conjunto de todas las funciones g que se pueden expresar como combinaciones lineales de g_0, g_1, \dots, g_n es un espacio vectorial \mathcal{G} (en este caso, un conocimiento de espacios vectoriales no es esencial para entender el análisis). Simbólicamente, tenemos

$$\mathcal{G} = \left\{ g : \text{existen } c_0, c_1, \dots, c_n \text{ tales que } g(x) = \sum_{j=0}^n c_j g_j(x) \right\}$$

La función que se busca en el problema de mínimos cuadrados es un elemento del espacio vectorial \mathcal{G} . Puesto que las funciones g_0, g_1, \dots, g_n forman una **base** de \mathcal{G} , el conjunto no es linealmente dependiente. Sin embargo, un espacio vectorial dado tiene muchas bases y pueden variar drásticamente en sus propiedades numéricas.

Volvamos nuestra atención de la base dada $\{g_0, g_1, \dots, g_n\}$ al espacio vectorial \mathcal{G} generado por esa base. Sin cambiar \mathcal{G} , nos preguntamos: *¿qué base para \mathcal{G} se debe elegir para el trabajo numérico?* En el presente problema, la principal tarea numérica es resolver las ecuaciones normales, es decir, la ecuación (7) de la sección 12.1:

$$\sum_{j=0}^n \left[\sum_{k=0}^m g_i(x_k) g_j(x_k) \right] c_j = \sum_{k=0}^m y_k g_i(x_k) \quad (0 \leq i \leq n) \quad (1)$$

La naturaleza de este sistema depende, obviamente, de la base $\{g_0, g_1, \dots, g_n\}$. Queremos que estas ecuaciones se resuelvan *fácilmente* o que se puedan resolver *exactamente*. La situación ideal ocurre cuando la matriz de coeficientes en la ecuación (1) es la matriz identidad. Esto ocurre si la base $\{g_0, g_1, \dots, g_n\}$ tiene la propiedad de **ortonormalidad**:

$$\sum_{k=0}^m g_i(x_k) g_j(x_k) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Usando esta propiedad, la ecuación (1) se simplifica drásticamente a

$$c_j = \sum_{k=0}^m y_k g_j(x_k) \quad (0 \leq j \leq n)$$

que ya no es un sistema de ecuaciones que hay que resolver sino más bien una fórmula explícita para los coeficientes c_j .

Bajo condiciones bastante generales, el espacio \mathcal{G} tiene una base que es ortogonal en el sentido que acabamos de describir. Un procedimiento conocido como el **proceso de Gram-Schmidt** se puede utilizar para obtener dicha base. Hay algunas situaciones en las que se justifica el esfuerzo de obtener una base ortonormal, pero procedimientos más simples con frecuencia son suficientes. Ahora presentamos un procedimiento de este tipo.

Recuerde que nuestro objetivo es hacer que la ecuación (1) tenga una buena disposición para solución numérica. Queremos evitar cualquier matriz de coeficientes que implique las dificultades encontradas en relación con la matriz de Hilbert (véase el problema de cómputo 7.2.4). Este objetivo puede alcanzarse si se ha elegido bien la base para el espacio \mathcal{G} .

Ahora consideremos el espacio \mathcal{G} que se compone de todos los polinomios de grado $\leq n$, que es un ejemplo importante de la teoría de mínimos cuadrados. Puede parecer lógico utilizar las siguientes $n + 1$ funciones como una base para \mathcal{G} :

$$g_0(x) = 1 \quad g_1(x) = x \quad g_2(x) = x^2 \quad \dots \quad g_n(x) = x^n$$

Usando esta base, se escribe un elemento típico del espacio \mathcal{G} de la forma

$$g(x) = \sum_{j=0}^n c_j g_j(x) = \sum_{j=0}^n c_j x^j = c_0 + c_1 x + c_2 x^2 + \dots + c_n x^n$$

Esta base, a pesar de ser natural, es casi siempre una *mala* elección para el trabajo numérico. Para muchos propósitos, los polinomios de Chebyshev (debidamente definidos para el intervalo implicado) forman una *buena* base.

La figura 12.4 da una indicación de por qué los monomios x^j no forman una buena base para el cálculo numérico: ¡estas funciones son muy parecidas! Si se da una función g y queremos

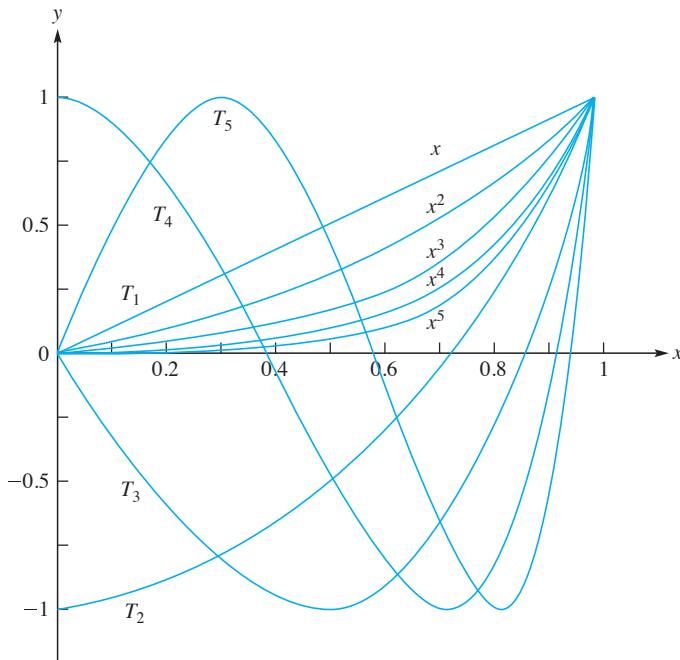


FIGURA 12.4
Polinomios x^k y
polinomios de
Chebyshev T_k

expresarla como una combinación lineal de los monomios, $g(x) = \sum_{j=0}^n c_j x^j$, es difícil determinar los coeficientes de c_j exactamente. La figura 12.4 muestra también algunos de los polinomios de Chebyshev que son muy diferentes unos de otros.

Por simplicidad, supongamos que los puntos en nuestro problema de mínimos cuadrados tienen la propiedad

$$-1 = x_0 < x_1 < \cdots < x_m = 1$$

Entonces se pueden utilizar los **polinomios de Chebyshev** para el intervalo $[-1, 1]$. La notación tradicional es

$$\begin{cases} T_0(x) = 1 & T_1(x) = x & T_2(x) = 2x^2 - 1 \\ T_3(x) = 4x^3 - 3x & T_4(x) = 8x^4 - 8x^2 + 1 & \text{etc.} \end{cases}$$

Una fórmula recursiva para estos polinomios es

$$T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x) \quad (j \geq 2) \quad (2)$$

Esta fórmula, junto con las ecuaciones $T_0(x) = 1$ y $T_1(x) = x$, ofrece una definición formal de los polinomios de Chebyshev. Alternativamente, podemos escribir $T_k(x) = \cos(k \arccos x)$.

Combinaciones lineales de los polinomios de Chebyshev son fáciles de evaluar porque se aplica un algoritmo especial de multiplicación anidada. Para describir este procedimiento, considere una combinación lineal arbitraria de $T_0, T_1, T_2, \dots, T_n$:

$$g(x) = \sum_{j=0}^n c_j T_j(x)$$

Un algoritmo para calcular $g(x)$ para cualquier x dada es el siguiente:

$$\begin{cases} w_{n+2} = w_{n+1} = 0 \\ w_j = c_j + 2xw_{j+1} - w_{j+2} \quad (j = n, n-1, \dots, 0) \\ g(x) = w_0 - xw_1 \end{cases} \quad (3)$$

Para ver que este algoritmo produce realmente $g(x)$, escribimos la serie para g , corremos algunos índices y usamos las fórmulas (2) y (3):

$$\begin{aligned} g(x) &= \sum_{j=0}^n c_j T_j(x) \\ &= \sum_{j=0}^n (w_j - 2xw_{j+1} + w_{j+2}) T_j \\ &= \sum_{j=0}^n w_j T_j - 2x \sum_{j=0}^n w_{j+1} T_j + \sum_{j=0}^n w_{j+2} T_j \\ &= \sum_{j=0}^n w_j T_j - 2x \sum_{j=1}^{n+1} w_j T_{j-1} + \sum_{j=2}^{n+2} w_j T_{j-2} \\ &= \sum_{j=0}^n w_j T_j - 2x \sum_{j=1}^n w_j T_{j-1} + \sum_{j=2}^n w_j T_{j-2} \\ &= w_0 T_0 + w_1 T_1 + \sum_{j=2}^n w_j T_j - 2xw_1 T_0 - 2x \sum_{j=2}^n w_j T_{j-1} + \sum_{j=2}^n w_j T_{j-2} \\ &= w_0 + xw_1 - 2xw_1 + \sum_{j=2}^n w_j (T_j - 2xT_{j-1} + T_{j-2}) \\ &= w_0 - xw_1 \end{aligned}$$

En general, es el mejor arreglar los datos para que todas las abscisas $\{x_k\}$ se encuentren en el intervalo $[-1, 1]$. Entonces, si algunos de los primeros polinomios de Chebyshev se utilizan como base para los polinomios, las ecuaciones normales deben estar razonablemente bien condicionadas. No hemos dado una definición técnica de este término; se puede interpretar de manera informal como que la eliminación gaussiana con pivoteo produce una solución exacta de las ecuaciones normales.

Si los datos originales no satisfacen que $\min\{x_k\} = -1$ y $\max\{x_k\} = 1$ pero se encuentran; en cambio, en un intervalo $[a, b]$, entonces el cambio de variable

$$x = \frac{1}{2}(b-a)z + \frac{1}{2}(a+b)$$

produce una variable z que atraviesa $[-1, 1]$ conforme x atraviesa $[a, b]$.

Diseño de algoritmo

Aquí se presenta un diseño de un procedimiento, basado en el análisis anterior, que produce un polinomio de grado $\leq (n+1)$ que se ajusta mejor a una determinada tabla de valores (x_k, y_k) ($0 \leq k \leq m$). En este caso, m es generalmente mucho mayor que n .

■ **ALGORITMO 1** *Ajuste polinomial*

1. Encuentre el intervalo $[a, b]$ más pequeño que contiene todos los x_k . Así, haciendo $a = \min\{x_k\}$ y $b = \max\{x_k\}$.

2. Realice una trasformación en el intervalo $[-1, 1]$ mediante la definición

$$z_k = \frac{2x_k - a - b}{b - a} \quad (0 \leq k \leq m)$$

3. Decida si se utilizará el valor de n . En esta situación, 8 o 10 sería un valor muy grande de n .

4. Usando polinomios de Chebyshev como base, genere las $(n + 1) \times (n + 1)$ ecuaciones normales

$$\sum_{j=0}^n \left[\sum_{k=0}^m T_i(z_k) T_j(z_k) \right] c_j = \sum_{k=0}^m y_k T_i(z_k) \quad (0 \leq i \leq n) \quad (4)$$

5. Use una rutina de resolución de ecuaciones para resolver las ecuaciones normales para los coeficientes c_0, c_1, \dots, c_n en la función

$$g(x) = \sum_{j=0}^n c_j T_j(x)$$

6. El polinomio que se busca es

$$g\left(\frac{2x - a - b}{b - a}\right)$$

Los detalles del paso 4 son los siguientes. Comience introduciendo una variable de doble subíndice:

$$t_{jk} = T_j(z_k) \quad 0 \leq k \leq m, 0 \leq j \leq n$$

La matriz $\mathbf{T} = (t_{jk})$ se puede calcular de manera eficiente usando la definición recursiva de los polinomios de Chebyshev, ecuación (2), como en el siguiente segmento de pseudocódigo:

```
integer j, k, m;  real array (t_{ij})_{0:n \times 0:m}, (z_i)_{0:n}
for k = 0 to m do
    t_{0k} ← 1
    t_{1k} ← z_k
    for j = 2 to n do
        t_{jk} ← 2z_k t_{j-1,k} - t_{j-2,k}
    end for
end for
```

Las ecuaciones normales tienen una matriz de coeficientes $\mathbf{A} = (a_{ij})_{0:n \times 0:n}$ y el miembro derecho $\mathbf{b} = (b_i)_{0:n}$ dados por

$$a_{ij} = \sum_{k=0}^m T_i(z_k) T_j(z_k) = \sum_{k=0}^m t_{ik} t_{jk} \quad (0 \leq i, j \leq n) \quad (5)$$

$$b_i = \sum_{k=0}^m y_k T_i(z_k) = \sum_{k=0}^m y_k t_{ik} \quad (0 \leq i \leq n)$$

El seudocódigo para calcular A y b es el siguiente:

```

real array ( $a_{ij}$ ) $_{0:n \times 0:n}$ , ( $b_i$ ) $_{0:n}$ , ( $t_{ij}$ ) $_{0:n \times 0:m}$ , ( $y_i$ ) $_{0:n}$ 
integer  $i, j, m, n$ ; real  $s$ 
for  $i = 0$  to  $n$  do
     $s \leftarrow 0$ 
    for  $k = 0$  to  $m$  do
         $s \leftarrow s + y_k t_{ik}$ 
    end for
     $b_i \leftarrow s$ 
    for  $j = i$  to  $n$  do
         $s \leftarrow 0$ 
        for  $k = 0$  to  $m$  do
             $s \leftarrow s + t_{ik} t_{jk}$ 
        end for
         $a_{ij} \leftarrow s$ 
         $a_{ji} \leftarrow s$ 
    end for
end for
```

Para ajustar los datos con los polinomios, existen otros métodos que emplean sistemas de polinomios hechos a la medida para un determinado conjunto de abscisas. El método que hemos delineado es, sin embargo, sencillo y directo.

Suavizado de datos: regresión polinomial

Una de las aplicaciones importantes de este procedimiento de mínimos cuadrados es el suavizado de datos. En este contexto, **suavizar** se refiere a ajustar a una curva “suave” a un conjunto de valores “con ruido” (es decir, valores que contienen errores experimentales). Si uno conoce el tipo de función a la que deben ajustarse los datos, entonces se puede utilizar el procedimiento de mínimos cuadrados para calcular los parámetros desconocidos en la función. Esto se ha ilustrado ampliamente en los ejemplos dados anteriormente. Sin embargo, si se desea simplemente suavizar los datos mediante el ajuste con cualquier función conveniente y después aumentar el grado del polinomio hasta que se obtenga un equilibrio razonable entre un buen ajuste y suavidad.

Esta idea se ilustrará con los datos experimentales que se indican en la tabla, que muestra 20 puntos (x_i, y_i) :

x	-1.0	-0.92	-0.84	-0.8	-0.72	-0.64	-0.56	-0.48	-0.36
y	4.0	1.0	5.0	7.0	6.0	3.0	2.0	2.0	5.0
-0.24	-0.12	0.0	0.12	0.2	0.32	0.4	0.52	0.64	0.76
12.0	13.0	11.0	7.0	4.0	-2.0	-6.0	-8.0	-2.0	4.0
									9.0

Por supuesto, se puede determinar un polinomio de grado 19 que pase por estos puntos *exactamente*. Pero si los puntos están contaminados con errores experimentales, nuestros objetivos se logran mejor con un polinomio de menor grado que ajuste los datos *aproximadamente* con el método de mínimos cuadrados. En el lenguaje estadístico, este es el problema de la *regresión curvilínea*.

Una buena biblioteca de software contendrá códigos para el ajuste polinomial de datos empíricos usando un criterio de mínimos cuadrados. Estos programas determinarán los polinomios de ajuste de grados 0, 1, 2, . . . con un mínimo de esfuerzo computacional y con alta precisión. Por supuesto, se pueden utilizar las técnicas que ya se presentaron en este capítulo, aunque no son del todo eficientes. Así, con los polinomios de Chebyshev como base, podemos plantear y resolver ecuaciones normales para $n = 0, 1, 2, \dots$ y trazar la gráfica de las funciones resultantes. En la figura 12.5 se muestran algunos de los polinomios obtenidos de esta manera para los datos de la tabla.

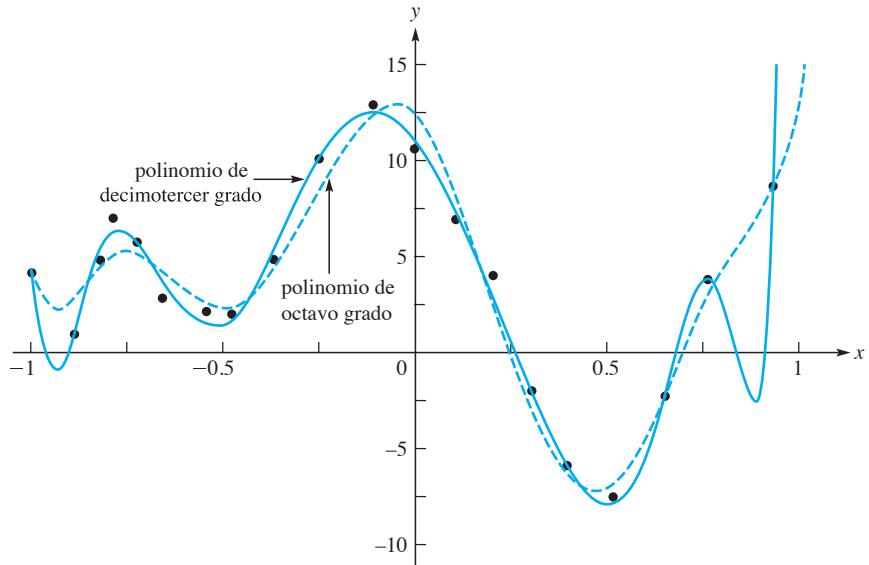


FIGURA 12.5
Polinomio de grado 8 (línea punteada) y polinomio de grado 13 (línea sólida)

Ahora explicaremos un procedimiento eficaz para la regresión polinomial, dado por Forsythe [1957]. Este procedimiento utiliza un sistema de polinomios ortogonales que se hacen a la medida para el problema que se encara. Empezaremos con una tabla de valores experimentales:

x	x_0	x_1	\dots	x_m
y	y_0	y_1	\dots	y_m

El objetivo final es sustituir esta tabla por un polinomio adecuado de grado modesto, eliminando de alguna manera los errores experimentales de la tabla. No sabemos qué grado de polinomio se deba utilizar.

Para efectos estadísticos, una hipótesis razonable es que existe un polinomio

$$p_N(x) = \sum_{i=0}^N a_i x^i$$

que representa la tendencia de la tabla y que los valores dados en ella obedecen la ecuación

$$y_i = p_N(x_i) + \varepsilon_i \quad (0 \leq i \leq m)$$

En esta ecuación, ε_i representa un error de observación que está presente en y_i . Una hipótesis más razonable es que estos errores son variables aleatorias independientes que se distribuyen normalmente.

Para un valor fijo de n , ya nos hemos referido a un procedimiento para determinar p_n por el método de mínimos cuadrados. Así, se puede establecer un sistema de ecuaciones normales para determinar los coeficientes p_n . Una vez que éstos sean conocidos, se puede calcular una cantidad llamada **varianza** a partir de la fórmula

$$\sigma_n^2 = \frac{1}{m-n} \sum_{i=0}^m [y_i - p_n(x_i)]^2 \quad (m > n) \quad (6)$$

La teoría estadística indica que si la tendencia de la tabla es en realidad un polinomio de grado N (pero infectado con ruido), entonces

$$\sigma_0^2 > \sigma_1^2 > \dots > \sigma_N^2 = \sigma_{N+1}^2 = \sigma_{N+2}^2 = \dots = \sigma_{m-1}^2$$

Este hecho sugiere la siguiente estrategia para tratar el caso en que N no se conoce: calcule $\sigma_0^2, \sigma_1^2, \dots$ en sucesión. Mientras estas están disminuyendo significativamente, se continúa con el cálculo. Cuando se alcance un número entero N para el que $\sigma_N^2 \approx \sigma_{N+1}^2 \approx \sigma_{N+2}^2 \approx \dots$, se detiene y se declara p_N como el polinomio buscado.

Si $\sigma_0^2, \sigma_1^2, \dots$ se calculan directamente de la definición de la ecuación (6), entonces se tendrán que determinar cada uno de los polinomios, p_0, p_1, \dots . El procedimiento que se describe a continuación puede evitar la determinación de todos pero obtener el polinomio deseado.

En lo que resta del análisis, se mantienen fijas las abscisas x_i . Se supone que estos puntos son diferentes, aunque la teoría se puede extender para incluir los casos en los que se repiten algunos puntos. Si f y g son dos funciones cuyos dominios incluyen los puntos $\{x_0, x_1, \dots, x_m\}$, entonces se utiliza la siguiente notación:

$$\langle f, g \rangle = \sum_{i=0}^m f(x_i)g(x_i) \quad (7)$$

Esta cantidad se llama el *producto interno* de f y g . Gran parte de nuestro análisis no depende de la forma exacta del producto interno, sino sólo de algunas de sus propiedades. Un **producto interno** $\langle \cdot, \cdot \rangle$ tiene las siguientes propiedades:

■ PROPIEDADES Definición de propiedades de un producto interno

1. $\langle f, g \rangle = \langle g, f \rangle$
2. $\langle f, f \rangle > 0$ a menos que $f(x_i) = 0$ para toda i
3. $\langle af, g \rangle = a\langle f, g \rangle$ donde $a \in \mathbb{R}$
4. $\langle f, g + h \rangle = \langle f, g \rangle + \langle f, h \rangle$

Usted debe comprobar que el producto interno definido en la ecuación (7) tiene las propiedades enumeradas.

Se dice que un conjunto de funciones es **ortogonal** si $\langle f, g \rangle = 0$ para cualesquiera dos funciones diferentes f y g en dicho conjunto. Un conjunto ortogonal de polinomios se puede generar de forma recursiva con las siguientes fórmulas:

$$\begin{cases} q_0(x) = 1 \\ q_1(x) = x - \alpha_0 \\ q_{n+1}(x) = xq_n(x) - \alpha_n q_n(x) - \beta_n q_{n-1}(x) \quad (n \geq 1) \end{cases}$$

donde

$$\begin{cases} \alpha_n = \frac{\langle xq_n, q_n \rangle}{\langle q_n, q_n \rangle} \\ \beta_n = \frac{\langle xq_n, q_{n-1} \rangle}{\langle q_{n-1}, q_{n-1} \rangle} \end{cases}$$

En estas fórmulas, ocurre un ligero abuso de notación donde “ xq_n ” se utiliza para denotar la función cuyo valor en x es $xq_n(x)$.

Para entender cómo esta definición conduce a un sistema ortogonal, vamos a examinar algunos casos. Primero,

$$\langle q_1, q_0 \rangle = \langle x - \alpha_0, q_0 \rangle = \langle xq_0 - \alpha_0 q_0, q_0 \rangle = \langle xq_0, q_0 \rangle - \alpha_0 \langle q_0, q_0 \rangle = 0$$

Observe que aquí se han utilizado algunas de las propiedades de un producto interno enumeradas antes. Además, se utilizó la definición de α_0 . Otro de algunos de los primeros casos es este:

$$\begin{aligned} \langle q_2, q_1 \rangle &= \langle xq_1 - \alpha_1 q_1 - \beta_1 q_0, q_1 \rangle \\ &= \langle xq_1, q_1 \rangle - \alpha_1 \langle q_1, q_1 \rangle - \beta_1 \langle q_0, q_1 \rangle = 0 \end{aligned}$$

Aquí, se ha utilizado la definición de α_1 , así como el hecho (establecido antes) que $\langle q_1, q_0 \rangle = 0$. El paso siguiente en una demostración formal es comprobar que $\langle q_2, q_0 \rangle = 0$. Despues, una demostración por inducción completa el argumento.

Una parte de esta demostración consiste en mostrar que los coeficientes α_n y β_n están bien definidos. Esto significa que los denominadores $\langle q_n, q_n \rangle$ no son cero. Para comprobar que este es el caso, supongamos que $\langle q_n, q_n \rangle = 0$. Entonces $\sum_{i=0}^m [q_n(x_i)]^2 = 0$ y por lo tanto, $q_n(x_i) = 0$ para cada valor de i . Esto significa que el polinomio q_n tiene $m+1$ raíces, x_0, x_1, \dots, x_m . Puesto que el grado n es menor que m , podemos concluir que q_n es el polinomio cero. Sin embargo, esto no es posible porque, obviamente

$$\begin{aligned} q_0(x) &= 1 \\ q_1(x) &= x - \alpha_0 \\ q_2(x) &= x^2 + \text{(términos de orden bajo)} \end{aligned}$$

y así sucesivamente. Obsérvese que este argumento requiere que $n < m$.

El sistema de polinomios ortogonales $\{q_0, q_1, \dots, q_{m-1}\}$ generados por el algoritmo anterior es una base para el espacio vectorial \prod_{m-1} de todos los polinomios de grado a lo más $m-1$. Es claro del algoritmo que cada q_n inicia con el término más alto x^n . Si se desea expresar un polinomio dado p de grado n ($n \leq m-1$) como una combinación lineal de q_0, q_1, \dots, q_n , esto se puede realizar de la siguiente manera. Hacemos

$$p = \sum_{i=0}^n a_i q_i \tag{8}$$

En el miembro derecho sólo un sumando contiene x^n . Es el término $a_n q_n$. En el miembro izquierdo, también hay un término en x^n . Se elige a_n de modo que el $a_n x^n$ de la derecha es igual al término correspondiente en p . Ahora, escribimos

$$p - a_n q_n = \sum_{i=0}^{n-1} a_i q_i$$

En ambos lados de esta ecuación hay polinomios de grado a lo más $n - 1$ (debido a la elección de a_n). Por lo tanto, ahora podemos elegir a_{n-1} en la forma en que se eligió a_n , es decir, se elige a_{n-1} para que los términos en x^{n-1} sean los mismos en ambos lados. Continuando de esta manera, descubrimos los únicos valores que deben tener los coeficientes a_i . Así se establece que $\{q_0, q_1, \dots, q_n\}$ es una base para \prod_n , para $n = 0, 1, \dots, m - 1$.

Otra manera de determinar los coeficientes a_i (¡una vez que sabemos que existen!) es tomar el producto interno de ambos lados de la ecuación (8) con q_j . El resultado es

$$\langle p, q_j \rangle = \sum_{i=0}^n a_i \langle q_i, q_j \rangle \quad (0 \leq j \leq n)$$

Puesto que el conjunto q_0, q_1, \dots, q_n es ortogonal, $\langle q_i, q_j \rangle = 0$ para cada i diferente de j . Por lo tanto, se obtiene

$$\langle p, q_j \rangle = a_j \langle q_j, q_j \rangle$$

Esto da a_j como un cociente de dos productos internos.

Ahora volvemos al problema de mínimos cuadrados. Sea F una función que queremos ajustar a un polinomio p_n de grado n . Vamos a encontrar el polinomio que minimiza la expresión

$$\sum_{i=0}^m [F(x_i) - p_n(x_i)]^2$$

La solución está dada por las fórmulas

$$p_n = \sum_{i=0}^n c_i q_i \quad c_i = \frac{\langle F, q_i \rangle}{\langle q_i, q_i \rangle} \quad (9)$$

Es especialmente notable que c_i no depende de n . Esto implica que los diferentes polinomios p_0, p_1, \dots , que estamos buscando se pueden obtener simplemente truncando *una* serie, a saber, $\sum_{i=0}^{m-1} c_i q_i$. Para demostrar que p_n , tal como se presenta en la ecuación (9), resuelve nuestro problema, volvemos a las ecuaciones normales, ecuación (1). Ahora las funciones básicas que se están utilizando son q_0, q_1, \dots, q_n . Así, las ecuaciones normales son

$$\sum_{j=0}^n \left[\sum_{k=0}^m q_i(x_k) q_j(x_k) \right] c_j = \sum_{k=0}^m y_k q_i(x_k) \quad (0 \leq i \leq n)$$

Usando la notación del producto interno tenemos

$$\sum_{j=0}^n \langle q_i, q_j \rangle c_j = \langle F, q_i \rangle \quad (0 \leq i \leq n)$$

donde F es alguna función tal que $F(x_k) = y_k$ para $0 \leq k \leq m$. A continuación, se aplica la propiedad de ortogonalidad $\langle q_i, q_j \rangle = 0$ cuando $i \neq j$. El resultado es

$$\langle q_i, q_i \rangle c_i = \langle F, q_i \rangle \quad (0 \leq i \leq n) \quad (10)$$

Ahora volvemos a los números de varianza $\sigma_0^2, \sigma_1^2, \dots$ y mostramos cómo se pueden calcular fácilmente. Primero, una observación importante: ¡el conjunto $\{q_0, q_1, \dots, q_n, F - p_n\}$ es ortogonal!

El único hecho nuevo en este caso es que $\langle F - p_n, q_i \rangle = 0$ para $0 \leq i \leq n$. Para comprobar esto, escribimos

$$\begin{aligned}\langle F - p_n, q_i \rangle &= \langle F, q_i \rangle - \langle p_n, q_i \rangle \\&= \langle F, q_i \rangle - \left\langle \sum_{j=0}^n c_j q_j, q_i \right\rangle \\&= \langle F, q_i \rangle - \sum_{j=0}^n c_j \langle q_j, q_i \rangle \\&= \langle F, q_i \rangle - c_i \langle q_i, q_i \rangle = 0\end{aligned}$$

En este cálculo se utilizaron las ecuaciones (9) y (10). Puesto que p_n es una combinación lineal de q_0, q_1, \dots, q_n , se deduce fácilmente que

$$\langle F - p_n, p_n \rangle = 0$$

Recordemos ahora que la varianza σ_n^2 fue definida por

$$\sigma_n^2 = \frac{\rho_n}{m - n} \quad \rho_n = \sum_{i=0}^m [y_i - p_n(x_i)]^2$$

Las cantidades ρ_n se pueden escribir de otra manera:

$$\begin{aligned}\rho_n &= \langle F - p_n, F - p_n \rangle \\&= \langle F - p_n, F \rangle \\&= \langle F, F \rangle - \langle F, p_n \rangle \\&= \langle F, F \rangle - \sum_{i=0}^n c_i \langle F, q_i \rangle \\&= \langle F, F \rangle - \sum_{i=0}^n \frac{\langle F, q_i \rangle^2}{\langle q_i, q_i \rangle}\end{aligned}$$

Así, los números ρ_0, ρ_1, \dots se pueden generar de forma recursiva mediante el algoritmo

$$\begin{cases} \rho_0 = \langle F, F \rangle - \frac{\langle F, q_0 \rangle^2}{\langle q_0, q_0 \rangle} \\ \rho_n = \rho_{n-1} - \frac{\langle F, q_n \rangle^2}{\langle q_n, q_n \rangle} \quad (n \geq 1) \end{cases}$$

Resumen

- (1) Usamos polinomios de Chebyshev $\{T_j\}$ como una base ortogonal que se puede generar recursivamente mediante

$$T_j(x) = 2x T_{j-1}(x) - T_{j-2}(x) \quad (j \geq 2)$$

con $T_0(x) = 1$ y $T_1(x) = x$. La matriz de coeficientes $\mathbf{A} = (a_{ij})_{0:n \times 0:n}$ y el miembro del lado derecho $\mathbf{b} = (b_i)_{0:n}$ de las ecuaciones normales son

$$a_{ij} = \sum_{k=0}^m T_i(z_k) T_j(z_k) \quad (0 \leq i, j \leq n)$$

$$b_i = \sum_{k=0}^m y_k T_i(z_k) \quad (0 \leq i \leq n)$$

Una combinación lineal de los polinomios de Chebyshev

$$g(x) = \sum_{j=0}^n c_j T_j(x)$$

se puede evaluar de forma recursiva:

$$\begin{cases} w_{n+2} = w_{n+1} = 0 \\ w_j = c_j + 2xw_{j+1} - w_{j+2} \quad (j = n, n-1, \dots, 0) \\ g(x) = w_0 - xw_1 \end{cases}$$

(2) Analizamos suavizar los datos mediante regresión polinomial.

Problemas 12.2

1. Sea g_0, g_1, \dots, g_n un conjunto de funciones tales que $\sum_{k=0}^m g_i(x_k) g_j(x_k) = 0$ si $i \neq j$. ¿Qué combinación lineal de estas funciones se ajusta mejor a los datos de la tabla (1) en la sección 12.1?
2. Considere los polinomios g_0, g_1, \dots, g_n definidos por $g_0(x) = 1$, $g_1(x) = x - 1$ y $g_j(x) = 3xg_{j-1}(x) + 2g_{j-2}(x)$. Desarrolle un algoritmo eficiente para el cálculo de valores de la función $f(x) = \sum_{j=0}^n c_j g_j(x)$.
3. Demuestre que $\cos n\theta = 2 \cos \theta \cos ((n-1)\theta) - \cos ((n-2)\theta)$. *Sugerencia:* use la identidad familiar $\cos(A \mp B) = \cos A \cos B \pm \sin A \sin B$.
4. (Continuación) Demuestre que si $f_n(x) = \cos(n \arccos x)$, entonces $f_0(x) = 1$, $f_1(x) = x$ y $f_n(x) = 2x f_{n-1}(x) - f_{n-2}(x)$.
5. (Continuación) Demuestre que una definición alternativa de los polinomios de Chebyshev es $T_n(x) = \cos(n \arccos x)$ para $-1 \leq x \leq 1$.
6. (Continuación) Presente una demostración de un renglón de que $(T_n(T_m(x))) = T_{nm}(x)$.
7. (Continuación) Demuestre que $|T_n(x)| \leq 1$ para x en el intervalo $[-1, 1]$.
8. Defina $g_k(x) = T_k(\frac{1}{2}x + \frac{1}{2})$. ¿Qué relación recursiva deben cumplir estas funciones?
9. Demuestre que T_0, T_2, T_4, \dots son funciones pares y que T_1, T_3, \dots son impares. Recuerde que una función par satisface la ecuación $f(x) = f(-x)$ y una función impar satisface la ecuación $f(x) = -f(-x)$.

“10. Cuente el número de operaciones que intervienen en el algoritmo utilizado para calcular $g(x) = \sum_{j=0}^n c_j T_j(x)$.

11. Demuestre que el algoritmo para calcular $g(x) = \sum_{j=0}^n c_j T_j(x)$ se puede modificar como

$$\begin{cases} w_{n-1} = c_{n-1} + 2x c_n \\ w_k = c_k + 2x w_{k+1} - w_{k-2} & (n-2 \geq k \geq 1) \\ g(x) = c_0 + x w_1 - w_2 \end{cases}$$

por lo que w_{n+2} , w_{n+1} y w_0 son innecesarios.

“12. (Continuación) Cuente las operaciones para el algoritmo del problema anterior.

“13. Determine $T_6(x)$ como un polinomio de x .

14. Compruebe las cuatro propiedades de un producto interno que se presentaron en el libro, usando la definición (7).

15. Compruebe estas fórmulas:

$$p_0(x) = \frac{1}{m+1} \sum_{i=0}^m y_i \quad \beta_n = \frac{\langle q_n, q_n \rangle}{\langle q_{n-1}, q_{n-1} \rangle} \quad c_n = \frac{\rho_{n-1} - \rho_n}{\langle F, q_n \rangle}$$

16. Complete la demostración de que el algoritmo para generar el sistema de polinomios ortogonales funciona.

“17. Existe una función f de la forma

$$f(x) = \alpha x^{12} + \beta x^{13}$$

para la que $f(0.1) = 6 \times 10^{-13}$ y $f(0.9) = 3 \times 10^{-2}$. ¿Cuál es? α y β son sensibles a las perturbaciones en los dos valores dados de $f(x)$?

18. (Opción múltiple) Sea $\mathbf{x}_1 = [2, 2, 1]^T$, $\mathbf{x}_2 = [1, 1, 5]^T$ y $\mathbf{x}_3 = [-3, 2, 1]^T$. Si el proceso de Gram-Schmidt se aplica a este conjunto ordenado de vectores para producir un conjunto ortonormal $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$, ¿qué es \mathbf{u}_1 ?

a. $\left[\frac{2}{3}, \frac{2}{3}, \frac{1}{3}\right]^T$

b. $[2, 2, 1]^T$

c. $\left[\frac{2}{5}, \frac{2}{5}, \frac{1}{5}\right]^T$

d. $[1, 0, 0]^T$

e. Ninguno de estos.

19. (Opción múltiple, continuación) ¿Qué es \mathbf{u}_2 ?

a. $\frac{1}{\sqrt{27}}[1, 1, 5]^T$

b. $\frac{1}{\sqrt{18}}[-1, -1, 4]^T$

c. $[2, 2, 1]^T$

d. $[1, 1, -4]^T$

e. Ninguno de estos.

Problemas de cómputo 12.2

- 1.** Realice un experimento con el suavizado de datos de la siguiente manera. Comience con un polinomio de grado modesto, digamos, 7. Calcule 100 valores de este polinomio en los puntos aleatorios en el intervalo $[-1, 1]$. Perturbe estos valores agregando números aleatorios seleccionados de un pequeño intervalo, por ejemplo, $[-\frac{1}{8}, \frac{1}{8}]$. Trate de recuperar el polinomio a partir de estos valores perturbados usando el método de mínimos cuadrados.

2. Escriba **real function** $Cheb(n, x)$ para evaluar $T_n(x)$. Utilice la fórmula recursiva que satisfacen los polinomios de Chebyshev. No use una variable con subíndice. Pruebe el programa con estos 15 casos: $n = 0, 1, 3, 6, 12$ y $x = 0, -1, 0.5$.
3. Escriba **real function** $Cheb(n, x, (y_i))$ para calcular $T_0(x), T_1(x), \dots, T_n(x)$ y almacene estos números en la matriz (y_i) . Utilice su rutina, junto con las rutinas adecuadas de trazado, para obtener las gráficas de $T_0, T_1, T_2, \dots, T_8$ en $[-1, 1]$.
4. Escriba **real function** $F(n, (c_i), x)$ para evaluar $f(x) = \sum_{j=0}^n c_j T_j(x)$. Pruebe su rutina por medio de la fórmula $\sum_{k=0}^{\infty} t^k T_k(x) = (1-tx) / (1-2tx + t^2)$, válido para $|t| < 1$. Si $|t| \leq \frac{1}{2}$, entonces sólo se necesitan unos cuantos términos de la serie para dar precisión total de máquina. Agregue términos en orden ascendente de magnitud.
5. Obtenga una gráfica de T_n de algún valor razonable de n por medio de la siguiente idea. Genera 100 ángulos θ_i igualmente espaciados en el intervalo $[0, \pi]$. Defina $x_i \cos \theta_i$ y $y_i = T_n(x_i) = \cos(n \arcs x_i) = \cos n\theta_i$. Envíe los puntos (x_i, y_i) a una rutina de trazado adecuado.
6. Escriba el código adecuado para llevar a cabo el procedimiento descrito en el libro para el ajuste de una tabla con una combinación lineal de los polinomios de Chebyshev. Pruebe en la forma del problema de cómputo 12.2.1, primero usando un polinomio sin perturbaciones. Averigüe experimentalmente cuán grande puede ser n en este proceso antes de que los errores de redondeo sean graves.
- ^a7. Defina $x_k = \cos [(2k-1)\pi/(2m)]$. Seleccione valores modestos de n y $m > 2n$. Calcule e imprima la matriz A , cuyos elementos son

$$a_{ij} = \sum_{k=0}^m T_i(x_k) T_j(x_k) \quad (0 \leq i, j \leq n)$$

Interprete los resultados en términos de un problema de ajuste polinomial con mínimos cuadrados.

8. Programe el algoritmo para encontrar $\sigma_0^2, \sigma_1^2, \dots$ en el problema de regresión polinomial.
9. Programe el algoritmo de regresión polinomial completo. La salida debe ser $\alpha_n, \beta_n, \sigma_n^2$ y c_n para $0 \leq n \leq N$, donde N está determinado por la condición $\sigma_{N-1}^2 > \sigma_N^2 \approx \sigma_{N+1}^2$.
10. Usando los polinomios ortogonales, encuentre el polinomio cuadrático que se ajuste a los siguientes datos en el sentido de mínimos cuadrados:

a.	x	-1	$-\frac{1}{2}$	0	$\frac{1}{2}$	1
	y	-1	0	1	2	1

b.	x	-2	-1	0	1	2
	y	2	1	1	1	2

12.3 Otros ejemplos del principio de mínimos cuadrados

El principio de mínimos cuadrados se utiliza también en otras situaciones. En una de ellas, tratamos de *resolver* un sistema inconsistente de ecuaciones lineales de la forma

$$\sum_{j=0}^n a_{kj} x_j = b_k \quad (0 \leq k \leq m) \tag{1}$$

en la que $m > n$. En este caso, hay $m + 1$ ecuaciones, pero sólo $n + 1$ incógnitas. Si una determinada tupla $n + 1$ (x_0, x_1, \dots, x_n) se sustituye por la izquierda, la discrepancia entre los dos lados de la k -ésima ecuación se denomina **k -ésimo residual**. Idealmente, por supuesto, todos los residuos deben ser cero. Si no es posible seleccionar (x_0, x_1, \dots, x_n) para que todos los residuos sean cero, el sistema (1) se dice que es **inconsistente** o **incompatible**. En este caso, una alternativa es reducir al mínimo la suma de los cuadrados de los residuos. Ello nos conduce a reducir al mínimo la expresión

$$\varphi(x_0, x_1, \dots, x_n) = \sum_{k=0}^m \left(\sum_{j=0}^n a_{kj} x_j - b_k \right)^2 \quad (2)$$

al hacer una selección adecuada de (x_0, x_1, \dots, x_n). Procediendo como antes, obtenemos derivadas parciales con respecto a x_i y se hacen igual a cero, para así obtener las ecuaciones normales

$$\sum_{j=0}^n \left(\sum_{k=0}^m a_{ki} a_{kj} \right) x_j = \sum_{k=0}^m b_k a_{ki} \quad (0 \leq i \leq n) \quad (3)$$

Este es un sistema lineal de sólo $n + 1$ ecuaciones que implica las incógnitas x_0, x_1, \dots, x_n . Se puede demostrar que este sistema es consistente, a condición de que los vectores columna en la matriz de coeficientes original sean linealmente independientes. El sistema (3) se puede resolver, por ejemplo, por eliminación gaussiana. La solución del sistema (3) es entonces la mejor solución aproximada de la ecuación (1) en el sentido de mínimos cuadrados.

Se han ideado métodos especiales para el problema que acabamos de analizar. En general, se gana en precisión con el método simple descrito anteriormente. Un algoritmo para la solución del sistema (1),

$$Ax = b$$

se comienza por factorizar

$$A = QR$$

donde la matriz Q es de $(m + 1) \times (n + 1)$ y satisface $Q^T Q = I$ y R es la matriz de $(n + 1) \times (n + 1)$ y se satisface que $r_{ii} > 0$ y $r_{ij} = 0$ para $j < i$. Entonces, la solución de mínimos cuadrados se obtiene mediante un algoritmo llamado *proceso de Gram-Schmidt modificado*.

Un algoritmo más elaborado (y más versátil) depende de la **descomposición de valor singular** de la matriz A . Esta es factorizada, $A = U\Sigma V^T$, en la que $U^T U = I_{m+1}$, $V^T V = I_{n+1}$ y Σ es una matriz diagonal $(m + 1) \times (n + 1)$ que tiene entradas no negativas. Para que estos procedimientos sean más confiables, se remite al lector a los materiales al final de esta sección y a Stewart [1973] y Lawson y Hanson [1995].

Uso de una función de peso $w(x)$

Otro ejemplo importante del principio de los mínimos cuadrados se presenta en el ajuste o aproximación de funciones en *intervalos* más que con conjuntos discretos. Por ejemplo, una función dada f definida en un intervalo $[a, b]$ podría tener que aproximarse mediante una función tal como

$$g(x) = \sum_{j=0}^n c_j g_j(x)$$

Es natural, entonces, tratar de reducir al mínimo la expresión

$$\varphi(c_0, c_1, \dots, c_n) = \int_a^b [g(x) - f(x)]^2 dx \quad (4)$$

eliendo los coeficientes de forma adecuada. En algunas aplicaciones, es conveniente obligar a las funciones g y f en un mejor acuerdo en ciertas partes del intervalo. Para esto, podemos modificar la ecuación (4) incluyendo una **función de peso** positiva $w(x)$, que puede por supuesto ser, $w(x) \equiv 1$ si todas las partes del intervalo se tratan igual. El resultado es

$$\varphi(c_0, c_1, \dots, c_n) = \int_a^b [g(x) - f(x)]^2 w(x) dx$$

El mínimo de φ se busca una vez más derivando con respecto a cada c_i y haciendo las derivadas parciales igual a cero. El resultado es un sistema de ecuaciones normales:

$$\sum_{j=0}^n \left[\int_a^b g_i(x) g_j(x) w(x) dx \right] c_j = \int_a^b f(x) g_i(x) w(x) dx \quad (0 \leq i \leq n) \quad (5)$$

Este es un sistema de $n + 1$ ecuaciones lineales con $n + 1$ incógnitas c_0, c_1, \dots, c_n y se puede resolver con eliminación gaussiana. También en este caso se aplican observaciones anteriores acerca de la elección de una buena base. La situación ideal es tener funciones g_0, g_1, \dots, g_n que tengan la propiedad de ortogonalidad:

$$\int_a^b g_i(x) g_j(x) w(x) dx = 0 \quad (i \neq j) \quad (6)$$

Muchos de esos sistemas ortogonales han sido desarrollados a lo largo de los años. Por ejemplo, los **polinomios de Chebyshev** forman un sistema de este tipo, a saber,

$$\int_{-1}^1 T_i(x) T_j(x) (1 - x^2)^{-1/2} dx = \begin{cases} 0 & i \neq j \\ \frac{\pi}{2} & i = j > 0 \\ \pi & i = j = 0 \end{cases}$$

La función de peso $(1 - x^2)^{-1/2}$ asigna gran peso a los extremos del intervalo $[-1, 1]$.

Si una sucesión de funciones distintas de cero g_0, g_1, \dots, g_n es ortogonal de acuerdo con la ecuación (6), entonces la sucesión $\lambda_0 g_0, \lambda_1 g_1, \dots, \lambda_n g_n$ es ortonormal para adecuados números reales positivos λ_j , a saber,

$$\lambda_j = \left\{ \int_a^b [g_j(x)]^2 w(x) dx \right\}^{-1/2}$$

Ejemplo no lineal

Como otro ejemplo del principio de mínimos cuadrados, aquí presentamos un problema no lineal. Supongamos que una tabla de puntos (x_k, y_k) se ajusta a una función de la forma

$$y = e^{cx}$$

Procediendo como antes se llega al problema de reducir al mínimo la función

$$\varphi(c) = \sum_{k=0}^m (e^{cx_k} - y_k)^2$$

El mínimo se produce para un valor de c tal que

$$0 = \frac{\partial \varphi}{\partial c} = \sum_{k=0}^m 2(e^{cx_k} - y_k)e^{cx_k}x_k$$

Esta ecuación es no lineal en c . Se podría contemplar su resolución por el método de Newton o el método de la secante. Por otra parte, el problema de minimizar $\varphi(c)$ se puede atacar directamente. Puesto que no puede haber múltiples raíces de la ecuación normal y mínimos locales en φ misma, una minimización directa de φ sería más seguro. Este tipo de dificultad es típica de los **problemas de mínimos cuadrados no lineales**. En consecuencia, con frecuencia se prefieren, otros métodos de ajuste de curvas si los parámetros desconocidos no se producen de forma lineal en el problema.

Como alternativa, este ejemplo particular se puede linealizar con un cambio de variable $z = \ln y$ y considerando

$$z = cx$$

El problema de minimizar la función

$$\varphi(c) = \sum_{k=0}^m (cx_k - z_k)^2 \quad z_k = \ln y_k$$

es fácil y conduce a

$$c = \frac{\sum_{k=0}^m z_k x_k}{\sqrt{\sum_{k=0}^m x_k^2}}$$

Este valor de c *no* es la solución del problema original, pero puede ser satisfactoria en algunas aplicaciones.

Ejemplo lineal y no lineal

El ejemplo final contiene elementos de la teoría lineal y no lineal. Supongamos que una tabla (x_k, y_k) está dada con $m + 1$ entradas y que se sospecha una relación funcional tal como

$$y = a \operatorname{sen}(bx)$$

¿Puede el principio de mínimos cuadrados utilizarse para obtener los valores adecuados de los parámetros a y b ?

Observe que el parámetro b entra en esta función de una manera no lineal, creando algunas dificultades, como se verá. De acuerdo con el principio de los mínimos cuadrados, los parámetros se eligen de manera que la expresión

$$\sum_{k=0}^m [a \operatorname{sen}(bx_k) - y_k]^2$$

tenga un valor mínimo. El valor mínimo se busca al derivar esta expresión con respecto a a y b y haciendo estas derivadas parciales igual a cero. Los resultados son

$$\begin{cases} \sum_{k=0}^m 2[a \operatorname{sen}(bx_k) - y_k] \operatorname{sen}(bx_k) = 0 \\ \sum_{k=0}^m 2[a \operatorname{sen}(bx_k) - y_k] ax_k \cos(bx_k) = 0 \end{cases}$$

Si se conociera b , a podría obtenerse de cualquiera de las ecuaciones. El valor correcto de b es el que corresponde a que los dos valores de a sean idénticos. Así, cada una de las ecuaciones anteriores se debe resolver para determinar a , y los resultados igualarse entre sí. Este proceso conduce a la ecuación

$$\frac{\sum_{k=0}^m y_k \operatorname{sen} bx_k}{\sum_{k=0}^m (\operatorname{sen} bx_k)^2} = \frac{\sum_{k=0}^m x_k y_k \cos bx_k}{\sum_{k=0}^m x_k \operatorname{sen} bx_k \cos bx_k}$$

que ahora se pueden resolver para el parámetro b , usando, por ejemplo, el método de la bisección o el método de la secante. Entonces cualquier miembro de esta ecuación se puede evaluar como el valor de a .

Detalles adicionales en DVS

La descomposición de valor singular (DVS) de una matriz es una factorización que puede revelar importantes propiedades de la matriz que de otro modo no podrían detectarse. Por ejemplo, de la DVS de una matriz cuadrada se podría advertir la casi singularidad de la matriz. O en la factorización DVS de una matriz no cuadrada se podría presentar una pérdida inesperada de rango. Puesto que la factorización DVS de una matriz produce una descomposición ortogonal completa, proporciona una técnica para calcular la solución de mínimos cuadrados de un sistema de ecuaciones y, al mismo tiempo, produce la norma del vector de error.

Supongamos que una matriz de $m \times n$ tiene la factorización

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T$$

donde $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$ es una matriz ortogonal de $m \times m$, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$ es una matriz ortogonal de $n \times n$ y la matriz diagonal \mathbf{D} de $m \times n$ contiene los valores singulares de \mathbf{A} en su diagonal, listados en orden decreciente. Los valores singulares de una matriz \mathbf{A} son las raíces cuadradas positivas de los valores propios de $\mathbf{A}^T \mathbf{A}$. Estos se denotan por $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$. En detalle, tenemos

$$\mathbf{U}^T \mathbf{A} \mathbf{V} = \mathbf{D} = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \\ & & & 0 \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix}_{m \times n}$$

donde $\mathbf{U}^T \mathbf{U} = \mathbf{I}_m$ y $\mathbf{V}^T \mathbf{V} = \mathbf{I}_n$. (En la matriz anterior, el espacio en blanco corresponde a entradas cero.) Además, tenemos $\mathbf{A} \mathbf{v}_i = \sigma_i \mathbf{u}_i$ y $\sigma_i = \|\mathbf{A} \mathbf{v}_i\|_2$ donde \mathbf{v}_i es la columna i en \mathbf{V} y

\mathbf{u}_i es la columna i en \mathbf{U} . Puesto que \mathbf{U} es ortogonal, obtenemos

$$\begin{aligned}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 &= \|\mathbf{U}^T(\mathbf{A}\mathbf{x} - \mathbf{b})\|_2^2 = \|\mathbf{U}^T\mathbf{A}\mathbf{x} - \mathbf{U}^T\mathbf{b}\|_2^2 \\ &= \|\mathbf{U}^T\mathbf{A}(\mathbf{V}\mathbf{V}^T)\mathbf{x} - \mathbf{U}^T\mathbf{b}\|_2^2 \\ &= \|(\mathbf{U}^T\mathbf{A}\mathbf{V})(\mathbf{V}^T\mathbf{x}) - \mathbf{U}^T\mathbf{b}\|_2^2 \\ &= \|\mathbf{D}\mathbf{V}^T\mathbf{x} - \mathbf{U}^T\mathbf{b}\|_2^2 = \|\mathbf{D}\mathbf{y} - \mathbf{c}\|_2^2 \\ &= \sum_{i=1}^r (\sigma_i y_i - c_i)^2 + \sum_{i=r+1}^m c_i^2\end{aligned}$$

donde $\mathbf{y} = \mathbf{V}^T\mathbf{x}$ y $\mathbf{c} = \mathbf{U}^T\mathbf{b}$. Aquí, \mathbf{y} se define por $y_i = c_i/\sigma_j$ y \mathbf{x} por $\mathbf{x} = \mathbf{V}\mathbf{y}$. Puesto que $c_i = \mathbf{u}_i^T\mathbf{b}$ y $\mathbf{x} = \mathbf{V}\mathbf{y}$, si $y_i = \sigma_i^{-1}c_i$ para $1 \leq i \leq r$ entonces la solución de mínimos cuadrados es

$$\mathbf{x}_{LS} = \sum_{i=1}^n y_i \mathbf{v}_i = \sum_{i=1}^r \sigma_i^{-1} c_i \mathbf{v}_i = \sum_{i=1}^r \sigma_i^{-1} (\mathbf{u}_i^T \mathbf{b}) \mathbf{v}_i$$

y

$$\|\mathbf{A}\mathbf{x}_{LS} - \mathbf{b}\|_2^2 = \sum_{i=r+1}^m c_i^2 = \sum_{i=r+1}^m (\mathbf{u}_i^T \mathbf{b})^2$$

que es el más pequeño de todos los minimizadores de norma dos. Para más información, consulte Golub y Van Loan [1996].

En conclusión, obtenemos el siguiente teorema.

■ TEOREMA 1

Teorema DVS de mínimos cuadrados

Sea \mathbf{A} una matriz de $m \times n$ de rango r . Sea la factorización DVS $\mathbf{A} = \mathbf{UDV}^T$. La solución de mínimos cuadrados del sistema $\mathbf{Ax} = \mathbf{b}$ es $\mathbf{x}_{LS} = \sum_{i=1}^r (\sigma_i^{-1} c_i) \mathbf{v}_i$ donde $c_i = \mathbf{u}_i^T \mathbf{b}$. Si existen muchas soluciones de mínimos cuadrados del sistema dado, entonces la menor de las normas dos es \mathbf{x} como se acaba de describir.

EJEMPLO 1 Encuentre la solución de mínimos cuadrados de este sistema no cuadrado

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

usando la descomposición de valor singular:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\sqrt{6} & 0 & \frac{1}{3}\sqrt{3} \\ \frac{1}{6}\sqrt{6} & \frac{1}{2}\sqrt{2} & -\frac{1}{3}\sqrt{3} \\ \frac{1}{6}\sqrt{6} & -\frac{1}{2}\sqrt{2} & -\frac{1}{3}\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & -\frac{1}{2}\sqrt{2} \end{bmatrix}$$

Solución Tenemos $r = \text{rango}(\mathbf{A}) = 2$ y los valores singulares $\sigma_1 = \sqrt{3}$ y $\sigma_2 = 1$. Esto conduce a

$$c_1 = \mathbf{u}_1^T \mathbf{b} = \left[\frac{1}{3}\sqrt{6} \quad \frac{1}{6}\sqrt{6} \quad \frac{1}{6}\sqrt{6} \right] \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \frac{1}{3}\sqrt{6}$$

y

$$c_2 = \mathbf{u}_2^T \mathbf{b} = \begin{bmatrix} 0 & -\frac{1}{2}\sqrt{2} & \frac{1}{2}\sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = \sqrt{2}$$

y

$$\begin{aligned} x_{LS} &= (\sigma_1^{-1}c_1)\mathbf{v}_1 + (\sigma_2^{-1}c_2)\mathbf{v}_2 = \frac{1}{\sqrt{3}} \left(\frac{1}{3}\sqrt{6} \right) \begin{bmatrix} \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \end{bmatrix} + \sqrt{2} \begin{bmatrix} \frac{1}{2}\sqrt{2} \\ -\frac{1}{2}\sqrt{2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{4}{3} \\ -\frac{2}{3} \end{bmatrix} \end{aligned}$$

Esta solución es igual a la de las ecuaciones normales.

Uso de la descomposición de valor singular

Este material requiere la teoría de la descomposición de valor singular expuesta en la sección 8.3.

Una aplicación importante de la descomposición de valor singular es en el problema de la matriz de mínimos cuadrados, a la que volvemos ahora. Para cualquier sistema de ecuaciones lineales $Ax = b$, queremos definir una única **solución mínima**. Esto se describe como sigue. Sea A de $m \times n$ y se define

$$\rho = \inf\{\|Ax - b\|_2 : x \in \mathbb{R}^n\}$$

La solución mínima de nuestro sistema se toma como el punto de menor norma en el conjunto $\{x: \|Ax - b\|_2 = \rho\}$. Si el sistema es *consistente*, entonces $\rho = 0$ y simplemente estamos pidiendo el punto de menor norma entre todas las soluciones. Si el sistema es *inconsistente*, queremos que Ax este lo más cerca posible de b , es decir, $\|Ax - b\|_2 = \rho$. Si hay muchos de estos puntos, se elige el más cercano al origen.

La solución mínima se produce usando la *seudo inversa* de A y este objeto, a su vez, se puede calcular a partir de la descomposición de valor singular de A , tal como se explica en la sección 8.3. Primero, consideremos una matriz de $m \times n$ diagonal de la forma siguiente, donde las σ_j son números positivos:

$$\mathbf{D} = \begin{bmatrix} \sigma_1 & & & & \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_r & 0 \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix}_{m \times n}$$

Su seudoinversa \mathbf{D}^+ se define para que sea de la misma forma, excepto que es de $n \times m$ y tiene $1/\sigma_j$ en su diagonal. Por ejemplo,

$$\mathbf{D} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \quad \mathbf{D}^+ = \begin{bmatrix} \frac{1}{5} & 0 \\ 0 & \frac{1}{2} \\ 0 & 0 \end{bmatrix}$$

Si \mathbf{A} es cualquier matriz de $m \times n$ y si \mathbf{UDV}^T es una de sus **descomposiciones de valor singular**, se define la **seudoinversa** de \mathbf{A} como

$$\mathbf{A}^+ = \mathbf{VD}^+ \mathbf{U}^T$$

No nos detendremos a demostrar que la seudoinversa de \mathbf{A} es única si imponemos el orden $\sigma_1 \geq \sigma_2 \geq \dots$

■ TEOREMA 2

Teorema de solución mínima

Considere un sistema de ecuaciones lineales $\mathbf{Ax} = \mathbf{b}$, en el que \mathbf{A} es una matriz de $m \times n$. La solución mínima del sistema es $\mathbf{A}^+ \mathbf{b}$.

Demostración Utilice la notación establecida anteriormente y sea \mathbf{x} cualquier punto de \mathbb{R}^n . Se define $\mathbf{y} = \mathbf{V}^T \mathbf{x}$ y $\mathbf{c} = \mathbf{U}^T \mathbf{b}$. Usando las propiedades de \mathbf{V} y \mathbf{U} , obtenemos

$$\begin{aligned} \rho &= \inf_x \|\mathbf{Ax} - \mathbf{b}\|_2 \\ &= \inf_x \|\mathbf{UDV}^T \mathbf{x} - \mathbf{b}\|_2 \\ &= \inf_x \|\mathbf{U}^T (\mathbf{UDV}^T \mathbf{x} - \mathbf{b})\|_2 \\ &= \inf_x \|\mathbf{DV}^T \mathbf{x} - \mathbf{U}^T \mathbf{b}\|_2 \\ &= \inf_y \|\mathbf{Dy} - \mathbf{c}\|_2 \end{aligned}$$

Aprovechando la naturaleza especial de \mathbf{D} , tenemos

$$\|\mathbf{Dy} - \mathbf{c}\|_2^2 = \sum_{i=1}^r (\sigma_i y_i - c_i)^2 + \sum_{i=r+1}^m c_i^2$$

Para minimizar esta última expresión, definimos $y_i = c_i / \sigma_i$ para $1 \leq i \leq r$. Las otras componentes pueden quedar sin especificar. Pero para obtener la \mathbf{y} de menor norma, debemos hacer $y_i = 0$ para $r+1 \leq i \leq m$. Esta construcción la realiza la seudoinversa \mathbf{D}^+ , por lo que $\mathbf{y} = \mathbf{D}^+ \mathbf{c}$. Por lo tanto, se obtiene

$$\mathbf{x} = \mathbf{V} \mathbf{y} = \mathbf{VD}^+ \mathbf{c} = \mathbf{VD}^+ \mathbf{U}^T \mathbf{b} = \mathbf{A}^+ \mathbf{b}$$

Vamos a expresar la solución mínima en otra forma, aprovechando las componentes cero en el vector \mathbf{y} . Puesto que $y_i = 0$ para $i > r$, se requieren sólo las primeras r componentes de \mathbf{y} . Estas están dadas por $y_i = \sigma_i / s_i$. Ahora bien, es evidente que se necesitan sólo las primeras r componentes de \mathbf{c} . Puesto que $\mathbf{c} = \mathbf{U}^T \mathbf{b}$, c_i es el producto interno del renglón i en \mathbf{U}^T con el vector \mathbf{b} . Ese es el mismo que el producto interno de la i ésima columna de \mathbf{U} con \mathbf{b} . Así,

$$y_i = \mathbf{u}_i^T \mathbf{b} / \sigma_i \quad 1 \leq i \leq r$$

La solución mínima, que se puede denotar por \mathbf{x}^* , es entonces

$$\mathbf{x}^* = \mathbf{V}\mathbf{y} = \sum_{i=1}^r y_i \mathbf{v}_i$$



Un ejemplo de este procedimiento puede realizarse con software matemático como Matlab, Maple o Mathematica. Podemos generar un sistema de 20 ecuaciones con tres incógnitas mediante un proceso aleatorio. Esta técnica se utiliza con frecuencia en las pruebas de software, especialmente en los estudios de medida, en el que un gran número de ejemplos se ejecuta con una programación cuidadosa. El software proporciona las entradas de las matrices aleatorias. Cuando se ejecuta, el programa de cómputo primero presenta una entrada aleatoria. Se presentan los tres valores singulares de la matriz \mathbf{A} . Después se presenta la matriz diagonal \mathbf{D} de 20×3 . Se realiza una comprobación del trabajo numérico mediante el cálculo de \mathbf{UDV}^T , que debe ser igual a \mathbf{A} . Después, se calcula la seudoinversa de \mathbf{D}^+ . A continuación, se calcula la seudoinversa de \mathbf{A}^+ . Se calcula la solución mínima, $\mathbf{x} = \mathbf{A}^+ \mathbf{b}$, así como el vector residual, $\mathbf{r} = \mathbf{A}^+ \mathbf{b} - \mathbf{b}$. Entonces, se ha comprobado la condición de ortogonalidad $\mathbf{A}^T \mathbf{r} = \mathbf{0}$. Por lo tanto, este programa realiza todos los pasos descritos anteriormente para obtener la solución mínima de un sistema de ecuaciones. A continuación se presentará otro ejemplo para mostrar lo que ocurre en el caso de una pérdida de rango (véase el problema de cómputo 12.3.10).

En problemas de este tipo, el usuario debe examinar los valores singulares y decidir si alguno de ellos es lo suficientemente pequeño como para merecer ser igual a cero. La necesidad de este paso se hace clara cuando nos fijamos en la definición de \mathbf{D}^+ . Los inversos de los valores singulares son los principales constituyentes de esta matriz. Por lo tanto, cualquier valor singular muy pequeño, que *no* sea igual a cero, tendrá un efecto perjudicial en los cálculos siguientes. Una regla de oro que se ha recomendado es dejar de lado cualquier magnitud singular cuyo valor sea inferior a σ_1 veces la precisión inherente de la matriz de coeficientes. Por ello, si los datos son exactos con tres decimales y si $\sigma_1 = 5$, entonces cualquier σ_i menor de 0.005 debe hacerse igual a cero.

A continuación se presenta un ejemplo de una matriz pequeña que tiene **deficiencia cercana en rango**. En el programa Maple, ciertos valores singulares son igual a cero, si éstos no cumplen con el criterio de tamaño relativo mencionados en el párrafo anterior. Además, hemos agregado, como una comprobación de los cálculos, la verificación de las siguientes cuatro propiedades de la seudomatriz de Penrose.

■ TEOREMA 3

Propiedades de Penrose de la seudoinversa

La seudoinversa \mathbf{A}^+ de la matriz \mathbf{A} tiene estas cuatro propiedades:

$$\begin{aligned} \mathbf{A} &= \mathbf{A}\mathbf{A}^+ \mathbf{A} & \mathbf{A}^+ &= \mathbf{A}^+ \mathbf{A}\mathbf{A}^+ \\ \mathbf{A}\mathbf{A}^+ &= (\mathbf{A}\mathbf{A}^+)^T & \mathbf{A}^+ \mathbf{A} &= (\mathbf{A}^+ \mathbf{A})^T \end{aligned}$$

Podemos utilizar software matemático como Matlab, Maple o Mathematica para encontrar la seudoinversa de una matriz que tiene una deficiencia en rango. Por ejemplo, considere esta matriz de 5×3 :

$$\mathbf{A} = \begin{bmatrix} -85 & -55 & -115 \\ -35 & 97 & -167 \\ 79 & 56 & 102 \\ 63 & 57 & 69 \\ 45 & -8 & 97.5 \end{bmatrix} \quad (7)$$

Se establece un valor de tolerancia para que en la evaluación de los valores singulares cualquier magnitud que sea menor que la tolerancia se considere como cero. Podemos comprobar las propiedades de Penrose para esta matriz (véase el problema de cómputo 12.3.11).

Resumen

(1) Se intenta resolver un **sistema inconsistente**

$$\sum_{j=0}^n a_{kj}x_j = b_k \quad (0 \leq k \leq m)$$

en el que hay $m + 1$ ecuaciones pero sólo $n + 1$ incógnitas con $m > n$. Minimizamos la suma de los cuadrados de los residuos y esto nos conduce a minimizar la expresión

$$\varphi(x_0, x_1, \dots, x_n) = \sum_{k=0}^m \left(\sum_{j=0}^n a_{kj}x_j - b_k \right)^2$$

Se resuelve el sistema de ecuaciones normales de $(n + 1) \times (n + 1)$

$$\sum_{j=0}^n \left(\sum_{k=0}^m a_{ki}a_{kj} \right) x_j = \sum_{k=0}^m b_k a_{ki} \quad (0 \leq i \leq n)$$

con eliminación gaussiana y el resultado es una mejor solución aproximada del sistema original en el sentido de mínimos cuadrados.

Referencias adicionales

Véase Acton [1959], Björck [1996], Branham [1990], Cheney [1982, 2001], Forsythe [1957], Van Huffel y Vandewalle [1991], Lawson y Hanson [1995], Rice [1971], Rice y White [1964], Rivlin [1990], Späth [1992] y Whittaker y Robinson [1944].

Problemas 12.3

1. Analice el problema de mínimos cuadrados de ajuste de datos mediante una función de la forma $y = x^c$.
2. Demuestre que la **matriz de Hilbert** (problema de cómputo 7.2.4) surge en las ecuaciones normales cuando minimizamos

$$\int_0^1 \left[\sum_{j=0}^n c_j x^j - f(x) \right]^2 dx$$

3. Encuentre una función de la forma $y = e^{cx}$ que mejor se ajuste a esta tabla:

x	0	1
y	$\frac{1}{2}$	1

^a4. (Continuación) Repita el problema anterior para la tabla siguiente:

x	0	1
y	a	b

5. (Continuación) Repita el problema anterior, bajo el supuesto de que b es negativo.

6. Demuestre que la ecuación normal para el problema de ajustar $y = e^{cx}$ a los puntos $(1, -12)$ y $(2, 7.5)$ tiene dos raíces reales: $c = \ln 2$ y $c = 0$. ¿Qué valor es correcto para el problema de ajuste?

7. Considere el sistema inconsistente (1). Supongamos que cada ecuación se ha asociado con un número positivo w_i indicando su importancia relativa o fiabilidad. ¿Cómo deberían las ecuaciones (2) y (3) modificarse para reflejar esta situación?

8. Determine la mejor solución aproximada del sistema inconsistente de ecuaciones lineales

$$\begin{cases} 2x + 3y = 1 \\ x - 4y = -9 \\ 2x - y = -1 \end{cases}$$

en el sentido de mínimos cuadrados.

9. ^aa. Encuentre la constante c para los que cx es la mejor aproximación en el sentido de mínimos cuadrados a la función $\sin x$ en el intervalo $[0, \pi/2]$.

^ab. Haga lo mismo para e^x en $[0, 1]$.

10. Analice el problema de ajuste de una función $y = (c - x)^{-1}$ a una tabla de $m + 1$ puntos.

11. Demuestre que las ecuaciones normales para la solución de mínimos cuadrados $Ax = b$ se puede escribir $(A^T A)x = A^T b$.

12. Deduzca las ecuaciones normales dadas por el sistema (5).

13. Una tabla de valores (x_k, y_k) , donde $k = 0, 1, \dots, m$, se obtiene de un experimento. Cuando se trazan en papel semilogarítmico, los puntos se encuentran casi en una línea recta, lo que implica que $y \approx e^{ax + b}$. Sugiera un procedimiento simple para la obtención de los parámetros a y b .

^a14. Al ajustar una tabla de valores a una función de la forma $a + bx^{-1} + cx^{-2}$, tratamos de hacer que cada punto se encuentre sobre la curva. Esto conduce a $a + bx_k^{-1} + cx_k^{-2} = y_k$ para $0 \leq k \leq m$. Una ecuación equivalente es $ax_k^2 + bx_k + c = y_k x_k^2$ para $0 \leq k \leq m$. ¿Son equivalentes los problemas de mínimos cuadrados para estos sistemas de ecuaciones?

^a15. Una tabla de puntos (x_k, y_k) se dibuja y parece ser una hipérbola de la forma $y = (a + bx)^{-1}$. ¿Cómo puede la teoría *lineal* de los mínimos cuadrados utilizarse para obtener buenas estimaciones de a y b ?

^a16. Considere $f(x) = e^{2x}$ en $[0, \pi]$. Deseamos aproximar la función mediante un polinomio trigonométrico de la forma $p(x) = a + b \cos(x) + c \sin(x)$. Determine el sistema lineal por resolver para determinar el ajuste por mínimos cuadrados de p a f .

^a17. Encuentre la constante c que hace que la expresión $\int_0^1 (e^x - cx)^2$ sea un mínimo.

18. Demuestre que en cada problema matricial de mínimos cuadrados, las ecuaciones normales tienen una matriz de coeficientes simétrica.

- 19.** Compruebe que con los siguientes pasos se obtiene la solución de mínimos cuadrados de $\mathbf{A}\mathbf{x} = \mathbf{b}$.
- Factorice $\mathbf{A} = \mathbf{QR}$, donde \mathbf{Q} y \mathbf{R} tienen las características descritas en el libro.
 - Defina $\mathbf{y} = \mathbf{Q}^T\mathbf{b}$.
 - Resuelva el sistema triangular inferior $\mathbf{Rx} = \mathbf{y}$.
- 20.** ¿Qué valor de c se debe utilizar si una tabla de datos experimentales (x_i, y_i) para $0 \leq i \leq m$ se representa mediante la fórmula $y = c \operatorname{sen} x$? Se necesita una fórmula explícita utilizable para c . Utilice el principio de mínimos cuadrados.
- 21.** Refiérase a las fórmulas que conducen a la solución mínima del sistema $\mathbf{Ax} = \mathbf{b}$. Demuestre que el vector \mathbf{y} está dado por la fórmula: $y_i = \sigma_i^{-2} \mathbf{b}^T \mathbf{Av}_i$ para $1 \leq i \leq r$.
- 22.** Demuestre que la seudoinversa cumple las cuatro ecuaciones de Penrose.
- 23.** Utilice las propiedades de Penrose para encontrar la seudoinversa de la matriz $[a, 0]^T$, donde $a > 0$. Demuestre que la seudoinversa es una función discontinua de a .
- 24.** Utilice la técnica sugerida en el problema anterior para encontrar la seudoinversa de la matriz de $m \times n$ que sólo consta de unos.
- 25.** Use las ecuaciones de Penrose para encontrar la seudoinversa de cualquier matriz de $1 \times n$ y de cualquier matriz de $m \times 1$.
- 26.** (Opción múltiple) Sea $\mathbf{A} = \mathbf{PDQ}$, donde \mathbf{A} es una matriz de $m \times n$, \mathbf{P} es una matriz unitaria de $m \times m$, \mathbf{D} es una matriz diagonal de $m \times n$ y \mathbf{Q} es una matriz unitaria de $n \times n$. ¿Qué ecuación se puede deducir de estas suposiciones?
- $\mathbf{A}^* = \mathbf{P}^* \mathbf{D}^* \mathbf{Q}^*$
 - $\mathbf{A}^{-1} = \mathbf{Q}^* \mathbf{D}^{-1} \mathbf{P}^*$
 - $\mathbf{D} = \mathbf{PAQ}$
 - $\mathbf{A}^* \mathbf{A} = \mathbf{Q}^* \mathbf{D}^* \mathbf{DQ}$
 - Ninguna de estas.
- 27.** (Opción múltiple, continuación) Supongamos las hipótesis del problema anterior. Use la notación ${}^+$ para indicar una seudoinversa. ¿Cuál ecuación es la correcta?
- $\mathbf{A}^+ = \mathbf{PD}^+ \mathbf{Q}$
 - $\mathbf{A}^* = \mathbf{Q}^* \mathbf{D}^{-1} \mathbf{P}^*$
 - $\mathbf{A}^+ = \mathbf{Q}^* \mathbf{D}^+ \mathbf{P}^*$
 - $\mathbf{A}^{-1} = \mathbf{Q}^* \mathbf{D}^+ \mathbf{P}^*$
 - Ninguna de estas.
- 28.** (Opción múltiple) Sea \mathbf{D} una matriz diagonal de $m \times n$ con elementos diagonales $p_1, p_2, \dots, p_r, 0, 0, \dots, 0$. En este caso todos los números p_i , para $1 \leq i \leq r$, son positivos. ¿Qué enunciando *no* es válido?
- \mathbf{D}^+ es la matriz diagonal de $m \times n$ con elementos de la diagonal $(1/p_1, 1/p_2, \dots, 1/p_r, 0, 0, \dots, 0)$
 - \mathbf{D}^+ es la matriz diagonal de $n \times m$ con elementos de la diagonal $(1/p_1, 1/p_2, \dots, 1/p_r, 0, 0, \dots, 0)$
 - $(\mathbf{D}^*)^* = (\mathbf{D}^*)^+$
 - $\mathbf{D}^{++} = \mathbf{D}$
 - Ninguno de estos.
- 29.** (Opción múltiple) Considere un sistema inconsistente de ecuaciones $\mathbf{Ax} = \mathbf{b}$. Sea \mathbf{U} una matriz unitaria y sea $\mathbf{E} = \mathbf{U}^* \mathbf{A}$. Sean \mathbf{v}, \mathbf{w} y \mathbf{z} vectores tal que $\mathbf{Uv} = \mathbf{Eb}$, $\mathbf{Uw} = \mathbf{E}^* \mathbf{b}$, $\mathbf{Ey} = \mathbf{U}^* \mathbf{b}$ y $\mathbf{Ex} = \mathbf{Ub}$. Un vector que resuelve el problema de mínimos cuadrados para el sistema original $\mathbf{Ax} = \mathbf{b}$ es:
- \mathbf{v}
 - \mathbf{w}
 - \mathbf{z}
 - \mathbf{y}
 - Ninguno de estos.

Problemas de cómputo 12.3

- 1.** Usando el método propuesto en el libro, ajuste los datos de la tabla

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
y	0.6	1.1	1.6	1.8	2.0	1.9	1.7	1.3

mediante una función $y = a \sen bx$.

- 2. (Método de Prony, $n = 1$)** Para ajustar una tabla de la forma

x	1	2	\cdots	m
y	y_1	y_2	\cdots	y_m

mediante la función $y = ab^x$, podemos proceder de la siguiente manera. Si y es realmente ab^x , entonces $y_k = ab^k$ y $y_{k+1} = by_k$ para $k = 1, 2, \dots, m-1$. Así, determinamos b al resolver este sistema de ecuaciones usando el método de mínimos cuadrados. Tras hallar b , encontramos a al resolver las ecuaciones $y_k = ab^k$ en el sentido de mínimos cuadrados. Escriba un programa para realizar este procedimiento y pruébelo con un ejemplo artificial.

- 3.** (Continuación) Modifique el procedimiento del problema de cómputo anterior para manejar cualquier caso de puntos igualmente espaciados.

- 4.** Una forma rápida de ajustar una función de la forma

$$f(x) \approx \frac{a + bx}{1 + cx}$$

es aplicar el método de mínimos cuadrados al problema $(1 + cx)f(x) \approx a + bx$. Utilice esta técnica para ajustar los datos de la *población mundial* dada en este caso:

Año	Población (miles de millones)
1000	0.340
1650	0.545
1800	0.907
1900	1.61
1950	2.51
1960	3.15
1970	3.65
1980	4.20
1990	5.30

Determine cuándo será infinita la población mundial.

- 5. (Proyecto de investigación estudiantil)** Explore la cuestión de si el método de mínimos cuadrados se debe utilizar para hacer predicciones. Por ejemplo, estudie las variaciones en el problema anterior para determinar si un polinomio de cualquier grado sería satisfactorio.

- 6.** Escriba un procedimiento que tome como entrada una $(m + 1) \times (n + 1)$ matriz A y un $m + 1$ vector b y regrese la solución de mínimos cuadrados del sistema $Ax = b$.

- 7.** Escriba un programa de Maple para encontrar la solución mínima de cualquier sistema de ecuaciones $Ax = b$.

8. (Continuación) Escriba un programa de Matlab para la tarea en el problema anterior.
9. Investigue algunos de los nuevos métodos para resolver ecuaciones lineales inconsistentes $\mathbf{A}\mathbf{x} = \mathbf{b}$, cuando el criterio es hacer $\mathbf{A}\mathbf{x}$ cercana a \mathbf{b} en una de las otras normas útiles, a saber, la norma máxima $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$ o la norma $\ell_1 \|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$. Use algún software disponible.
10. Usando software matemático como Matlab, Maple o Mathematica genere un sistema de veinte ecuaciones con tres incógnitas con un generador de números aleatorios. Forme la matriz seu-doinversa y compruebe las propiedades del teorema 2.
11. (Continuación.) Repita usando la matriz (7).
12. Escriba un programa de cómputo para realizar los ajustes de mínimos cuadrados usando polinomios de Chebyshev. Pruebe el código con un conjunto de datos adecuados y trace la gráfica de resultados.

Métodos de Monte Carlo y simulación

Un ingeniero de caminos desea simular el flujo del tráfico para una propuesta de diseño de una intersección de autopistas principales. La información que se obtenga entonces se utilizará para determinar la capacidad de los *carriles de almacenamiento* (en los que los automóviles deben frenar para ceder el derecho de paso). La intersección tiene la forma que se muestra en la figura 13.1 y los diversos flujos (vehículos por minuto) se suponen en los puntos donde se dibujan flechas. Al escribir y ejecutar un programa de simulación, el ingeniero puede estudiar el efecto de los diferentes límites de velocidad y determinar los flujos que conducen a la saturación (cuellos de botella) y así sucesivamente. En este capítulo se desarrollan algunas técnicas para construir este tipo de programas.

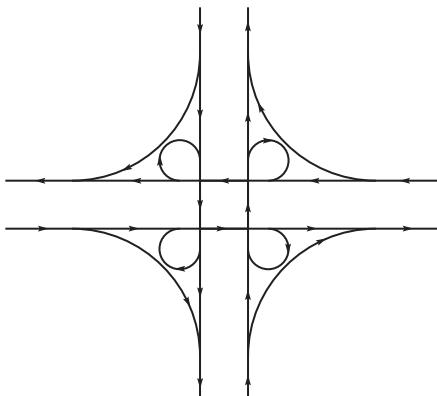


FIGURA 13.1
Flujo de tráfico

13.1 Números aleatorios

Este capítulo difiere de la mayoría de los otros en su punto de vista. En lugar de abordar los problemas con un corte claramente matemático, se intentan desarrollar métodos para la simulación de complicados procesos o fenómenos. Si se puede hacer que la computadora imite un experimento o un proceso, entonces, repitiendo las simulaciones de la computadora con datos diferentes, podemos obtener conclusiones estadísticas. En este método, los resultados pueden carecer de un alto grado de precisión matemática, pero aún son lo suficientemente precisos para que nos permitan comprender el proceso que se está simulando.

Se presta especial atención a los problemas de simulación con computadora que tienen un elemento *aleatorio*. El nombre caprichoso de *métodos de Monte Carlo* lo aplicó hace algunos años

Stanislaw M. Ulam (1909–1984) a esta forma de imitar la realidad con una computadora. Puesto que lo fortuito o lo aleatorio es parte del método, comenzamos con el esquivo concepto de **números aleatorios**.

Consideré una sucesión de números reales x_1, x_2, \dots todos situados en el intervalo unitario $(0, 1)$. Expresado de manera informal, la sucesión es **aleatoria** si los números parecen estar distribuidos al azar en todo el intervalo y si parece no haber un patrón en la progresión de x_1, x_2, \dots . Por ejemplo, si todos los números en forma decimal comienzan con el dígito 3, los números están agrupados en el subintervalo $0.3 \leq x < 0.4$ y no están distribuidos aleatoriamente en $(0, 1)$. Si los números son uniformemente crecientes, no son aleatorios. Si cada x_i se obtiene de su predecesor mediante una función continua, por ejemplo, $x_i = f(x_{i-1})$, entonces la sucesión no es aleatoria (aunque pueda parecer que lo es). Una definición precisa de *aleatoriedad* es muy difícil de formular y el lector interesado puede consultar un artículo de Chaitlin [1975], en el que relaciona la aleatoriedad con la complejidad de los algoritmos de computadora. Así, parece que lo mejor, al menos como introducción, es aceptar de manera intuitiva el concepto de una sucesión aleatoria de números en un intervalo y aceptar algunos algoritmos para la generación de progresiones que sea más o menos aleatorias.

Un libro de consulta recomendable es el de Niederreiter [1992].

Algoritmos y generadores de números aleatorios

La mayoría de los sistemas de cómputo tienen **generadores de números aleatorios**, que son procedimientos que producen un número aleatorio simple o un arreglo de números aleatorios con cada llamada. En este capítulo, llamamos a dicho procedimiento *Aleatorio*. Usted puede utilizar un generador de números aleatorios disponible en su propio sistema de cómputo, uno disponible en el lenguaje de programación utilizado o uno de los generadores que se describen a continuación. Por ejemplo, los generadores de números aleatorios están contenidos en sistemas de software matemático como Matlab, Maple y Mathematica, así como en muchos lenguajes de programación. Estos procedimientos de generación de números aleatorios regresan uno o un arreglo de números seudoaleatorios uniformemente distribuidos en el intervalo unitario $(0, 1)$, dependiendo de si el argumento es una variable escalar o una matriz. Un procedimiento de semillas aleatorias reinicia o consulta el generador de números seudoaleatorios, el cual puede producir cientos de miles de números seudoaleatorios antes de que se repitan, al menos teóricamente.

Para los problemas de este capítulo se debe seleccionar una rutina para proporcionar números aleatorios distribuidos uniformemente en el intervalo $(0, 1)$. Una sucesión de números está **uniformemente distribuida** en el intervalo $(0, 1)$ si ningún subconjunto del intervalo contiene más que su parte de los números. En particular, la probabilidad de que un elemento x procedente de la sucesión esté en el subintervalo $[a, a + h]$ debe ser h y por lo tanto independiente del número a . Del mismo modo, si los puntos $p_i = (x_i, y_i)$ son aleatorios en el plano uniformemente distribuidos en algún rectángulo, entonces el número de estos puntos que caen dentro de un cuadrado pequeño de área k debería depender solamente de k y no del sitio donde el cuadrado está situado dentro del rectángulo.

Los números aleatorios producidos por código de computadora no pueden ser verdaderamente aleatorios, porque la forma en que se producen es totalmente *determinista*, es decir, ningún elemento aleatorio está presente. Sin embargo, las sucesiones que se producen con estas rutinas parecen ser aleatorias y pasan ciertas pruebas de aleatoriedad. Algunos autores prefieren hacer hincapié en este punto, llamando a tales sucesiones números **seudoaleatorios**.

Si usted desea programar un generador de números aleatorios, el siguiente debe ser satisfactorio en una máquina que tiene 32 bits de longitud de la palabra. Este algoritmo genera n números

aleatorios x_1, x_2, \dots, x_n uniformemente distribuidos en el intervalo abierto $(0, 1)$ por medio del algoritmo recursivo siguiente:

```

integer array  $(\ell_i)_{0:n}$ ; real array  $(x_i)_{1:n}$ 
 $\ell_0 \leftarrow$  cualquier entero tal que  $1 < \ell_0 < 2^{31} - 1$ 
for  $i = 1$  to  $n$  do
     $\ell_i \leftarrow$  residuo de dividir  $7^5 \ell_{i-1}$  entre  $2^{31} - 1$ 
     $x_i \leftarrow \ell_i / (2^{31} - 1)$ 
end for
```

Aquí, todos los ℓ_i son enteros en el rango de $1 < \ell_i < 2^{31} - 1$. El entero inicial ℓ_0 se llama la **semilla** de la sucesión y se selecciona como cualquier entero entre 1 y el número primo de Mersenne $2^{31} - 1 = 21474\,83647$.

Para obtener información sobre generadores portátiles de números aleatorios, usted debe consultar el artículo de Schrage [1979]. Un generador de números aleatorios rápido *normal* se puede escribir en sólo unos cuantos renglones de código como se presenta en Leva [1992]. Se basa en la relación de las desviaciones uniformes del método de Kinderman y Monahan [1977].

Un procedimiento de función externa que genera una nueva matriz de números pseudoaleatorios por llamada podría basarse en el siguiente seudocódigo:

```

real procedure Aleatorio( $(x_i)$ )
integer semilla, i, n; real array  $(x_i)_{1:n}$ 
integer k  $\leftarrow 16807$ , j  $\leftarrow 21474\,83647$ 
semilla  $\leftarrow$  selecciona el valor inicial de semilla
n  $\leftarrow$  tamaño( $(x_i)$ )
for i = 1 to n do
    semilla  $\leftarrow$  mod( $k \cdot$  semilla, j)
     $x_i \leftarrow$  real(semilla) / real(j)
end for
end procedure Aleatorio
```

Para permitir una representación adecuada de las cifras implicadas en el procedimiento *Aleatorio*, este se debe escribir utilizando precisión doble o extendida para su uso en una computadora de 32 bits; de lo contrario, no producirá números aleatorios.

Recordemos que aquí y donde sea, $\text{mod}(n, m)$ es el residuo cuando n se divide entre m ; es decir, esto da como resultado $n - [\text{entero}(n/m)] m$, donde $\text{entero}(n/m)$ es el entero que resulta del truncamiento de n/m . Por lo tanto, $\text{mod}(44, 7)$ es 2, $\text{mod}(3, 11)$ es 3 y $\text{mod}(n, m)$ es 0 siempre que m divida a n uniformemente. También observamos que $x \equiv y$ módulo (z) significa que $x - y$ es divisible entre z .

Los bosquejos de los otros dos algoritmos generadores de números aleatorios son los siguientes:

■ ALGORITMO 1 Madre de todos los generadores de números seudoaleatorios

Iniciarizar los cuatro valores de x_0, x_1, x_2, x_3 y c en los valores aleatorios basados en el valor de la semilla. Sea $s = 211111111 x_{n-4} + 1492x_{n-3} + 1776x_{n-2} + 5115x_{n-1} + c$, calcule $x_n = s \bmod(2^{32})$ y $c = [s/2^{32}]$ para $n \geq 4$. Inventado por George Marsaglia (véase www.agnr.org/random/).

■ ALGORITMO 2 *rand () en Unix*

Inicializa la x_0 con un valor aleatorio basado en un valor de la semilla. Calcula $x_{n+1} = (1103515245 x_n + 12345) \text{ mod}(2^{31})$ para $n \geq 1$.

Estos algoritmos son adecuados para algunas aplicaciones, pero no pueden producir la aleatoriedad de alta calidad y pueden no ser adecuados para aplicaciones que requieren estadísticas precisas o en criptografía. En Internet, se pueden encontrar nuevos y perfeccionados generadores de seudonúmeros aleatorios, que están diseñados para generaciones rápidas de números aleatorios de alta calidad, con períodos grandes y con distribuciones especiales (véase, por ejemplo, www.gnu.org/software/gsl/).

Son necesarias algunas advertencias acerca de los generadores de números aleatorios en los sistemas de cómputo. Ya se ha indicado el hecho de que las sucesiones producidas por estos programas no son verdaderamente aleatorias. En algunas simulaciones, la falta de aleatoriedad puede llevar a conclusiones erróneas. En este caso hay tres puntos concretos y ejemplos para recordar:

■ PROPIEDADES

1. Los algoritmos del tipo ilustrado aquí por *Aleatorio* y los anteriores producen sucesiones **periódicas**; es decir, a la larga se repiten. El período para *Aleatorio* es del orden de 2^{30} , que es bastante grande.
2. Si un generador de números aleatorios se utiliza para producir puntos aleatorios en un espacio n -dimensional, estos puntos se encuentran en un número relativamente pequeño de planos o hiperplanos. Como Marsaglia [1968] reporta, los puntos obtenidos de esta manera en espacios tridimensionales se encuentran en un conjunto de sólo 119086 planos para computadoras con almacenamiento entero de 48 bits. En el espacio 10 se encuentran en un conjunto de 126 planos, que es bastante pequeño.
3. Los dígitos individuales que forman números aleatorios generados por las rutinas, como *Aleatorio* no son, en general, dígitos aleatorios independientes. Por ejemplo, podría suceder que el dígito 3 siga al 5 con más (o menos) frecuencia de lo que se esperaría.

Ejemplos

Un ejemplo de un seudocódigo para calcular e imprimir diez números aleatorios usando el procedimiento *Aleatorio* siguiente:

```
program Prueba_Aleatoria
real array (xi)1:n; integer n ← 10
call Random ((xi))
output (xi)
end program Prueba_Aleatoria
```

Los resultados de computadora de una corrida típica son los siguientes:

0.31852 29, 0.53260 59, 0.50676 22, 0.15271 48, 0.67687 93,
0.31067 89, 0.57963 66, 0.95331 68, 0.39584 57, 0.97879 35

Sistemas de software matemático como Matlab, Maple o Mathematica tienen colecciones de generadores de números aleatorios con diversas distribuciones. Por ejemplo, se pueden

generar números seudoaleatorios uniformemente distribuidos en el intervalo $(0, 1)$. Además, son particularmente útiles para el trazado y la visualización de puntos aleatorios generados dentro de las regiones en una, dos y tres dimensiones.

Como una burda comprobación en el generador de números aleatorios, vamos a calcular una larga sucesión de números aleatorios y determinar qué proporción de ellos se encuentra en el intervalo $(0, \frac{1}{2}]$. La respuesta calculada debe ser aproximadamente 50%. Se tabulan los resultados con diferentes longitudes de sucesión. Aquí se presenta el seudocódigo para realizar este experimento:

```
program Comprobación_Burda
integer i, m; real per; real array (ri)1:n
integer n ← 10000
m ← 0
call Random((ri))
for i = 1 to n do
    if ri ≤ 1/2 then m ← m + 1
    if mod(i, 1000) = 0 then
        per ← 100 real(m)/ real(n)
        output i, per
    end if
end for
end program Comprobación_Burda
```

En este seudocódigo se genera una sucesión de 10000 números aleatorios. En el camino, la proporción actual de números menores que $\frac{1}{2}$ se calcula en el 1000 ésmo paso y después en múltiplos de 1000. Algunos de los resultados de la computadora del experimento son 49.5, 50.2, 51.0 y 50.625.

El experimento descrito también puede interpretarse como una simulación con computadora del volado de una moneda. Un solo volado corresponde a la selección de un número aleatorio x en el intervalo $(0, 1)$. Arbitrariamente asociamos caras con $0 < x \leq \frac{1}{2}$ y cruces con el evento $\frac{1}{2} < x < 1$. Mil volados de la moneda corresponden a 1000 opciones de números aleatorios. Los resultados muestran la proporción de caras que resultan de repetidos volados de la moneda. También se pueden utilizar números enteros aleatorios para simular el volado de una moneda.

Observe que (al menos en este experimento) se alcanza una razonable precisión con sólo un número moderado de números aleatorios (4000). Repitiendo el experimento 10000 veces sólo tiene una influencia marginal sobre la precisión. Por supuesto, en teoría, si los números aleatorios fueran verdaderamente aleatorios, el valor límite, así como el número de números aleatorios usados crecería sin límite hasta ser exactamente 50%.

En este seudocódigo y otros en el capítulo, todos los números aleatorios son generados inicialmente, almacenados en un arreglo y se utiliza posteriormente en el programa conforme sea necesario. Esta es una forma eficiente de obtener estos números porque minimiza el número de llamadas de procedimiento, pero tienen el costo de espacio de almacenamiento. Si el espacio de memoria es un lujo, la llamada al generador de números aleatorios se puede mover más cerca de su uso (dentro del ciclo) para que devuelva un número aleatorio simple con cada llamada.

Ahora consideramos algunas cuestiones básicas sobre la generación de puntos aleatorios en diversas configuraciones geométricas. Suponga que el procedimiento *Aleatorio* se utiliza para obtener un número aleatorio r en el intervalo $[0, 1]$. Primero, si se necesitan puntos aleatorios uni-

formemente distribuidos en algún intervalo (a, b) , el enunciado

$$x \leftarrow (b - a)r + a$$

logra esto. Segundo, el seudocódigo

$$i \leftarrow \text{entero}((n + 1)r)$$

produce enteros aleatorios en el conjunto $\{0, 1, \dots, n\}$. Tercero, para enteros aleatorios de j a k ($j \leq k$), utilice el enunciado de asignación

$$i \leftarrow \text{entero}((k - j + 1)r + j)$$

Por último, se pueden utilizar los siguientes enunciados para obtener los primeros cuatro dígitos de un número aleatorio:

```
integer array  $(m_i)_{1:n}$ ; integer  $i$ ; real  $r, x$ 
integer  $n \leftarrow 4$ 
call Aleatorio( $r$ )
for  $i = 1$  to  $n$  do
     $x \leftarrow 10r$ 
     $m_i \leftarrow \text{entero}(x)$ 
     $x \leftarrow x - \text{real}(m_i)$ 
end for
output  $(m_i)$ 
```

Uso del seudocódigo *Aleatorio*

Ahora mostraremos los usos correctos e incorrectos de *Aleatorio* para producir puntos distribuidos uniformemente.

Considere el problema de generación de 1000 puntos aleatorios distribuidos uniformemente dentro de la elipse $x^2 + 4y^2 = 4$.

Una forma de hacerlo es generar puntos aleatorios en el rectángulo $-2 \leq x \leq 2, -1 \leq y \leq 1$ y descartar los que no se encuentran en la elipse (figura 13.2).

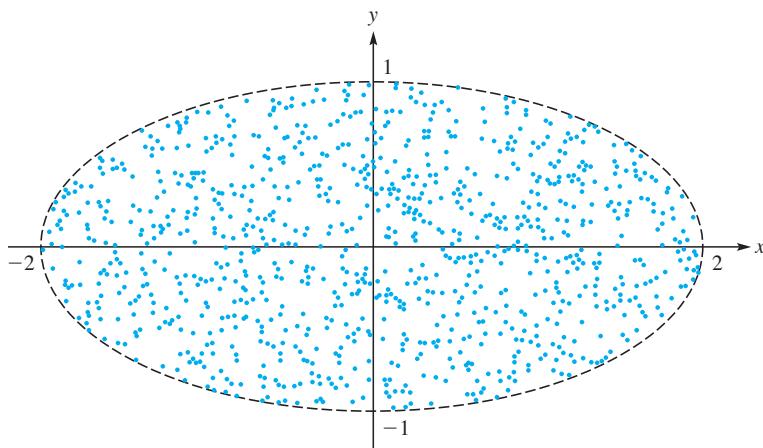


FIGURA 13.2
Puntos
aleatorios
distribuidos
uniformemente
en elipse
 $x^2 + 4y^2 = 4$

```

program Elipse
integer i, j; real u, v; real array (xi)1:n, (yi)1:n, (rij)1:
integer n ← 1000, npts ← 2000
call Aleatorio((rij))
j ← 1
for i = 1 to npts do
    u ← 4ri,1 - 2
    v ← 2ri,2 - 1
    if u2 + 4v2 ≤ 4 then
        xj ← u
        yj ← v
        j ← j + 1
    if j = n then exit loop i
    end if
end for
end program Elipse

```

Para desperdiciar menos, se puede *forzar* al valor $|y|$ a ser menor que $\frac{1}{2}\sqrt{4 - x^2}$, como en el siguiente pseudocódigo, que *produce resultados erróneos* (figura 13.3):

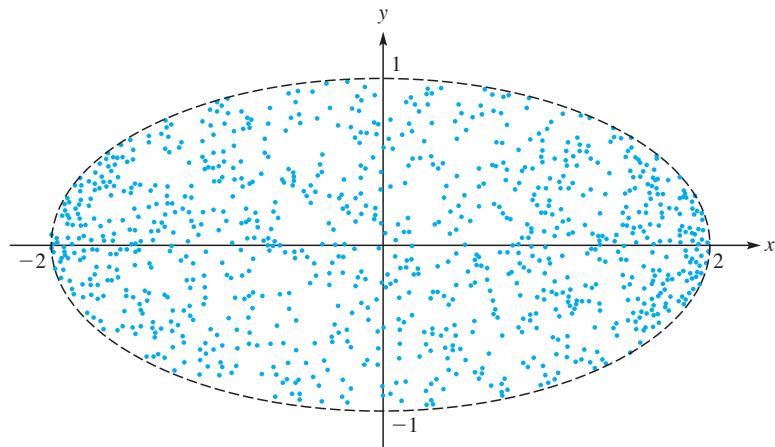


FIGURA 13.3
Puntos aleatorios distribuidos no uniformemente en la elipse $x^2 + 4y^2 = 4$

```

program Elipse_Errónea
integer i; real array (xi)1:n, (yi)1:n, (rij)
integer n ← 1000
call Random((rij))
for i = 1 to n do
    xi ← 4ri,1 - 2
    yi ← [(2ri,2 - 1)/ 2]√(4 - xi2)
end for
end program Elipse_Errónea

```

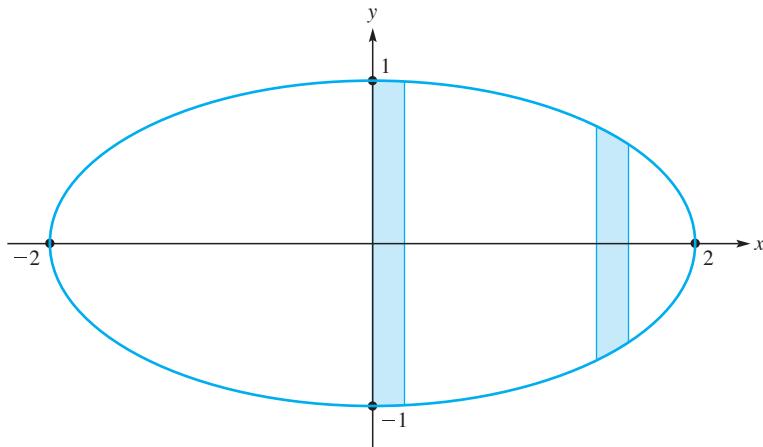


FIGURA 13.4
Las tiras verticales contienen puntos distribuidos de manera no uniforme

Este seudocódigo *no* produce puntos distribuidos uniformemente dentro de la elipse. Para convenirse de esto, considere dos tiras verticales tomadas dentro de la elipse (figura 13.4). Si cada tira tiene un ancho h , entonces aproximadamente $1000(h/4)$ de los puntos se encuentran aleatoriamente en cada tira, ya que la variable aleatoria x está uniformemente distribuida en $(-2, 2)$ y con cada x , el programa genera una y correspondiente de modo que (x, y) está dentro de la elipse. Pero las dos tiras de muestra *no* contienen aproximadamente el mismo número de puntos, ya que no tienen la misma área. Los puntos generados por el segundo programa tienden a estar agrupados en los extremos izquierdo y derecho de la elipse en la figura 13.3.

Por las mismas razones, el siguiente seudocódigo *no* produce puntos aleatorios uniformemente distribuidos en el círculo $x^2 + y^2 = 1$ (figura 13.5):

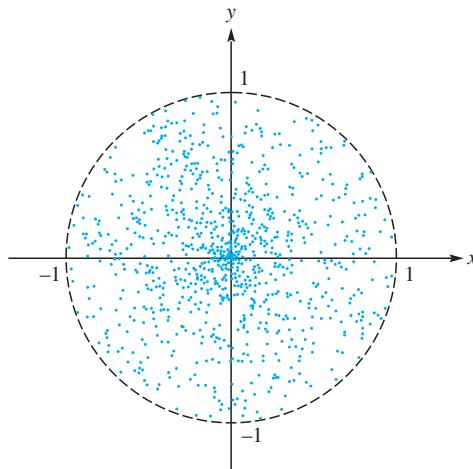


FIGURA 13.5
Puntos aleatorios no uniformemente distribuidos en el círculo $x^2 + y^2 = 1$

```
program Círculo_Erróneo
integer i; real array (xi)1:n, (yi)1:n, (rij)1:n×1:2
integer n ← 1000
```

```

call Random( $(r_{ij})$ )
for  $i = 1$  to  $n$  do
     $x_i \leftarrow r_{i,1} \cos(2\pi r_{i,2})$ 
     $y_i \leftarrow r_{i,1} \sin(2\pi r_{i,2})$ 
end for
end program Círculo Erróneo

```

En este seudocódigo, $2\pi r_{i,2}$ se distribuye uniformemente en $(0, 2\pi)$ y $r_{i,1}$ se distribuye uniformemente en $(0, 1)$. Sin embargo, en el cambio de coordenadas polares a rectangulares con las ecuaciones $x = r_{i,1} \cos(2\pi r_{i,2})$ y $y = r_{i,1} \sin(2\pi r_{i,2})$, se pierde la uniformidad. Los puntos aleatorios están muy agrupados cerca del origen en la figura 13.5.

Un generador de números aleatorios produce una sucesión de números que son aleatorios en el sentido de que están distribuidos uniformemente en un cierto intervalo como $[0, 1]$ y no es posible predecir el siguiente número en la sucesión conociendo los anteriores. Se puede aumentar la aleatoriedad de esta sucesión con una *reorganización* adecuada de ellos. La idea es llenar una matriz con los números consecutivos del generador de números aleatorios y después usar de nuevo el generador para elegir aleatoriamente cuál de los números del arreglo se debe seleccionar como el siguiente número en una nueva sucesión. La esperanza es que la nueva sucesión sea más aleatoria que la original. Por ejemplo, una reorganización puede quitar alguna correlación entre los sucesores cercanos a un número en una sucesión. Véase Flowers [1995] para un procedimiento de reorganización que se puede utilizar con un generador de números aleatorios basado en una congruencia lineal. Esto resulta particularmente útil en computadoras con una longitud de palabra pequeña.

Existen pruebas estadísticas que se pueden realizar en una sucesión de números aleatorios. Si bien estas pruebas no certifican la aleatoriedad de la sucesión, son especialmente importantes en las aplicaciones. Por ejemplo, son útiles en la elección de los diferentes generadores de números aleatorios y es reconfortante saber que el generador de números aleatorios que se utiliza las ha pasado. Hay situaciones en que los generadores de números aleatorios son útiles a pesar de que no pasan pruebas rígidas de aleatoriedad real. Así, si uno está produciendo matrices aleatorias para probar un código de álgebra lineal, entonces la aleatoriedad estricta puede no ser importante. Por otro lado, la aleatoriedad estricta es *fundamental* en la integración de Monte Carlo y otras aplicaciones. En los casos en los que la aleatoriedad estricta es importante, se recomienda que utilice una máquina con un tamaño de palabra grande y un generador de números aleatorios con características estadísticas conocidas (véase el volumen 2 de Knuth [1997] o Flores [1995] para algunas pruebas de aleatoriedad).

Las sucesiones **cuasialeatorias** o de discrepancias bajas se construyen para dar una cobertura uniforme de un área o volumen, manteniendo una apariencia bastante aleatoria a pesar de que en realidad no lo son.

Un **número primo** es un entero mayor que 1, cuyos únicos factores (divisores) son él mismo y el 1. Los números primos son unos de los pilares fundamentales en matemáticas. La búsqueda de números primos grandes tiene una historia larga e interesante. En 1644, Mersenne (un fraile francés) conjectura que $2^n - 1$ era un número primo para $n = 17, 19, 31, 67, 127, 257$ y no para ningún otro en el rango $1 \leq n \leq 257$. En 1876, Lucas demostró que $2^{127} - 1$ era primo. Sin embargo, en 1937, Lehmer demostró que $2^{257} - 1$ no era primo. Hasta 1952, $2^{127} - 1$ era el primo más grande conocido. Entonces se demostró que $2^{521} - 1$ era primo. Como un medio de pruebas de nuevos sistemas de cómputo, la búsqueda de primos mayores que el de Mersenne continúa. De hecho, la búsqueda de números primos cada vez más grandes ha crecido en importancia por su uso en criptografía. En 1992, un superordenador Cray 2 determinó utilizando la prueba de Lucas-Lehmer después de 19 horas de cálculo que el número $2^{756839} - 1$ era primo. ¡Este número tiene 227832 dígitos! El número anterior más grande conocido como el primo de Mersenne fue identificado en 1985 como $2^{216091} - 1$. En 2006, el primo más grande conocido era $2^{32582657} - 1$, con 9.8 millones de dígitos, se descubrió utilizando la facilidad GIMPS (gran búsqueda de números primos de Mersenne por Internet) de Internet. Miles de personas han utilizado la base de datos GIMPS para facilitar su

búsqueda de números primos grandes y la interacción con la base de datos puede hacerse de forma automática sin intervención humana. Para obtener más información acerca de números primos grandes y para encontrar el registro actual para el primo más grande conocido, consulte <http://www.mersenne.org/prime.html> y [www.utm.edu/investigación/primes](http://www.utm.edu/investigacion/primes).

Resumen

(1) Un algoritmo para generar un arreglo (r_i) de números seudoaleatorios es

```
integer  $\ell$ ; real array  $(x_i)_{1:n}$ 
 $\ell \leftarrow$  un entero entre 1 y  $2^{31} - 1$ 
for  $i = 1$  to  $n$  do
     $\ell \leftarrow \text{mod}(7^5\ell, 2^{31} - 1)$ 
     $x_i \leftarrow \ell/(2^{31} - 1)$ 
end for
```

(2) Si (r_i) es un arreglo de números aleatorios, entonces, utilice lo siguiente para generar puntos aleatorios en un intervalo (a, b)

$$x \leftarrow (b - a)r_i + a$$

para producir enteros aleatorios en el conjunto $\{0, 1, \dots, n\}$

$$i \leftarrow \text{entero}((n + 1)r_i)$$

y para obtener números enteros aleatorios de j a k ($j \leq k$)

$$i \leftarrow \text{entero}((k - j + 1)r_i + j)$$

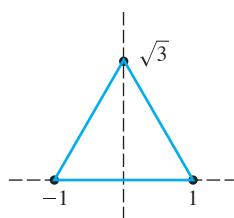
Problemas 13.1

1. Tomando la semilla 123456, calcule a mano los primeros tres números aleatorios producidos por el procedimiento *Random*.
2. Demuestre que si la semilla ℓ es menor o igual a 12777, entonces el primer número aleatorio producido por el procedimiento *Random* es menor que $\frac{1}{10}$.
3. Demuestre que los números producidos mediante el procedimiento *Random* no son aleatorios porque sus productos con $2^{31} - 1$ son enteros.
4. Describa de qué manera este algoritmo de números aleatorios se diferencia del procedimiento *Random*:

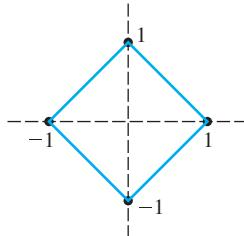
$$\begin{cases} x_0 \text{ arbitraria en } (0, 1) \\ x_i = \text{parte fraccionaria de } 7^5 x_{i-1} & i \geq 1 \end{cases}$$

Problemas de cómputo 13.1

1. Escriba un programa para generar 1000 puntos aleatorios distribuidos uniformemente en la cardioide $r = 2 - \cos \theta$.
2. Use el procedimiento *Random*, para escribir el código del procedimiento *Trapecio_Aleatorio* (x, y) , que genera un punto seudoaleatorio (x, y) en el interior o en el trapecio formado por los puntos $(1, 3), (2, 5), (4, 3)$ y $(3, 5)$.
3. Sin necesidad de utilizar ningún procedimiento, escriba un programa para generar e imprimir 100 números aleatorios uniformemente distribuidos en $(0, 1)$. Ocho enunciados bastan.
4. Pruebe algunos generadores de números aleatorios incluidos en software matemático que hay en la web.
5. Pruebe el generador de números aleatorios en su sistema de cómputo de la siguiente manera: genere 1000 números aleatorios $x_1, x_2, \dots, x_{1000}$.
 - a. En cualquier pequeño intervalo de ancho h , aproximadamente $1000h$ de las x_i deben hallarse ahí. Cuente la cantidad de números aleatorios en cada uno de los diez intervalos de $[(n-1)/10, n/10]$, donde $n = 1, 2, \dots, 10$.
 - b. La desigualdad $x_i < x_{i+1}$ debe ocurrir aproximadamente 500 veces. Cuéntelas en su muestra.
6. Escriba un procedimiento para generar con cada llamada un vector aleatorio de la forma $\mathbf{x} = [x_1, x_2, \dots, x_{20}]^T$, donde cada x_i es un entero de 1 a 100 y no hay dos componentes de \mathbf{x} iguales.
7. Escriba un programa para generar $n = 1000$ puntos aleatorios distribuidos uniformemente en el
 - a. triángulo equilátero de la siguiente figura:



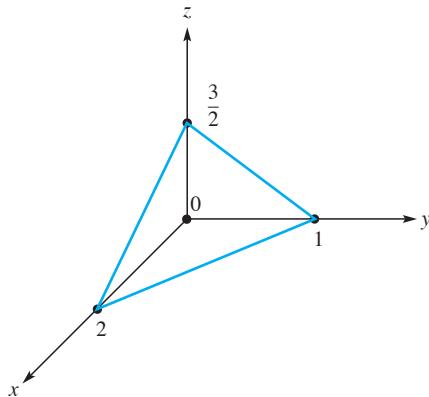
- b. diamante de la siguiente figura:



Almacene los puntos aleatorios (x_i, y_i) en los arreglos $(x_i)_{1:n}$ y $(y_i)_{1:n}$.

- 8.** Si x_1, x_2, \dots , es una sucesión de números aleatorios distribuidos uniformemente en el intervalo $(0, 1)$, ¿qué proporción se espera que satisfaga la desigualdad $40x^2 + 7 > 43x$? Escriba un programa para probar estos 1000 números aleatorios.
- 9.** Escriba un programa para generar e imprimir 1000 puntos distribuidos de manera uniforme y aleatoria en el círculo $(x - 3)^2 + (y + 1)^2 \leq 9$.
- 10.** Genere 1000 números aleatorios x_i de acuerdo con una distribución uniforme en el intervalo $(0, 1)$. Defina una función f en $(0, 1)$ de la siguiente manera: $f(t)$ es la cantidad de números aleatorios $x_1, x_2, \dots, x_{1000}$ menores que t . Calcule $f(t)/1000$ y 200 puntos t distribuidos uniformemente en $(0, 1)$. ¿Qué espera que sea $f(t)/1000$? Esta expectativa es confirmada con el experimento? Si dispone de un graficador, trace $f(t)/1000$.
- 11.** Sea n_i ($1 \leq i \leq 1000$) una sucesión de números enteros que satisface $0 \leq n_i \leq 9$. Escriba un programa para probar su periodicidad. (La sucesión es **periódica** si existe un entero k tal que $n_i = n_{i+k}$ para toda i .)
- 12.** Genere en la computadora 1000 números aleatorios en el intervalo $(0, 1)$. Imprima y examine la evidencia de un comportamiento no aleatorio.
- 13.** Genere 1000 números aleatorios x_i ($1 \leq i \leq 1000$) en su computadora. Sea n_i el octavo dígito decimal en x_i . Cuente cuántos $0, 1, \dots, 9$ hay entre los 1000 números n_i . ¿Cuántos de cada uno se puede esperar? Este código se puede escribir con nueve enunciados.
- 14.** (Continuación) Usando un generador de números aleatorios, produzca 1000 números aleatorios y cuente cuántas veces se presenta el dígito i en el j -ésimo lugar decimal. Imprima una tabla de estos valores, es decir, la frecuencia del dígito contra el lugar decimal. Examinando la tabla, determine qué lugar decimal parece producir la mejor distribución uniforme de dígitos aleatorios. *Sugerencia:* utilice la rutina del problema de cómputo 1.1.7 para calcular la media aritmética, la varianza y la desviación estándar de las entradas de la tabla.
- 15.** Usando números enteros aleatorios, escriba un breve programa para simular cinco personas juegan volados. Imprima el porcentaje de coincidencias (cinco del mismo tipo) después de 125 volados.
- 16.** Escriba un programa para generar 1600 puntos aleatorios distribuidos uniformemente en la esfera definida por $x^2 + y^2 + z^2 = 1$. Cuente el número de puntos aleatorios en el primer octante.
- 17.** Escriba un programa para simular 1000 volados simultáneos de tres monedas. Imprima el número de veces que dos de las tres monedas caen cara.
- 18.** Calcule 1000 triplets de números aleatorios extraídos de una distribución uniforme. Para cada triplet (x, y, z) , calcule el dígito significativo principal del producto xyz (el dígito significativo principal es uno de $1, 2, \dots, 9$). Determine la frecuencia con la que los dígitos de 1 a 9 ocurren en los 1000 casos. Trate de explicar por qué estos dígitos no se presentan con la misma frecuencia (por ejemplo, 1 se produce aproximadamente 7 veces más que 9). Si está intrigado por esto, puede consultar los artículos de Flehinger [1966], Raimi [1969] y Turner [1982].
- 19.** Ejecute los programas de ejemplo de esta sección y vea si se obtienen resultados similares en su sistema de cómputo.

20. Escriba un programa para generar y trazar 1000 puntos seudoaleatorios con la distribución **exponencial** siguiente dentro de la figura a continuación: $x = -\ln(1 - r)/\lambda$, para $r \in [0, 1]$ y $\lambda = 1/30$.



21. Mejore el programa *Coarse_Check* usando diez o cien *cubos* en lugar de dos.
22. (**Proyecto de investigación estudiantil**) Investigue algunos de los últimos desarrollos en generadores de números aleatorios y explore generadores de números aleatorios en paralelo. Los números aleatorios con frecuencia son necesarios para otras distribuciones además de la distribución uniforme, por lo que esto tiene un aspecto estadístico.

13.2 Cálculo de áreas y volúmenes mediante técnicas de Monte Carlo

Integración numérica

Pasemos ahora a las aplicaciones. La primera es la aproximación de una integral definida por el método de Monte Carlo. Si seleccionamos los primeros n elementos x_1, x_2, \dots, x_n de una sucesión aleatoria en el intervalo $(0, 1)$, entonces,

$$\int_0^1 f(x) dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i)$$

En este caso, la integral se aproxima a la media de n números $f(x_1), f(x_2), \dots, f(x_n)$. Cuando esto se realiza, el error es de orden $1/\sqrt{n}$, que no es del todo competitiva con algoritmos buenos, como el método de Romberg. Sin embargo, en dimensiones superiores, el método de Monte Carlo puede ser muy atractivo. Por ejemplo,

$$\int_0^1 \int_0^1 \int_0^1 f(x, y, z) dx dy dz \approx \frac{1}{n} \sum_{i=1}^n f(x_i, y_i, z_i)$$

donde (x_i, y_i, z_i) es una sucesión aleatoria de n puntos en el cubo unitario $0 \leq x \leq 1, 0 \leq y \leq 1$ y $0 \leq z \leq 1$. Para obtener puntos aleatorios en el cubo, se supone que tenemos una sucesión aleatoria

en $(0, 1)$ denotada por $\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \dots$. Para obtener nuestro primer punto aleatorio p_1 en el cubo, sólo sea $p_1 = (\xi_1, \xi_2, \xi_3)$. El segundo es, por supuesto, es $p_2 = (\xi_4, \xi_5, \xi_6)$ y así sucesivamente.

Si el intervalo (en una integral unidimensional) no es de longitud 1, pero, por ejemplo, es el caso general (a, b) , entonces el promedio de f en n puntos aleatorios en (a, b) no es simplemente una aproximación a la integral, sino más bien

$$\frac{1}{b-a} \int_a^b f(x) dx$$

que concuerda con nuestra intención de que la función $f(x) = 1$ tiene un promedio de 1. Del mismo modo, en las dimensiones superiores, el promedio de f en una región se obtiene integrando y dividiendo entre el área, el volumen o una medida de esa región. Por ejemplo,

$$\frac{1}{8} \int_1^3 \int_{-1}^1 \int_0^2 f(x, y, z) dx dy dz$$

es el promedio de f sobre el paralelepípedo descrito por tres estas desigualdades: $0 \leq x \leq 2$, $-1 \leq y \leq 1$, $1 \leq z \leq 3$.

Para conservar los límites de integración, recuerde que

$$\int_a^b \int_c^d f(x, y) dx dy = \int_a^b \left[\int_c^d f(x, y) dx \right] dy$$

y

$$\int_{a_1}^{a_2} \int_{b_1}^{b_2} \int_{c_1}^{c_2} f(x, y, z) dx dy dz = \int_{a_1}^{a_2} \left\{ \int_{b_1}^{b_2} \left[\int_{c_1}^{c_2} f(x, y, z) dx \right] dy \right\} dz$$

Así, si (x_i, y_i) denotan puntos aleatorios con distribución uniforme adecuada, los ejemplos siguientes ilustran técnicas de Monte Carlo:

$$\begin{aligned} \int_0^5 f(x) dx &\approx \frac{5}{n} \sum_{i=1}^n f(x_i) \\ \int_2^5 \int_1^6 f(x, y) dx dy &\approx \frac{15}{n} \sum_{i=1}^n f(x_i, y_i) \end{aligned}$$

En cada caso, los puntos aleatorios deben ser uniformemente distribuidos en las regiones implicadas.

En general, tenemos

$$\int_A f \approx (\text{medida de } A) \times (\text{promedio de } f \text{ en } n \text{ puntos aleatorios en } A)$$

En este caso, estamos usando el hecho de que el promedio de una función en un conjunto es igual a la integral de la función en el conjunto dividido entre la medida del conjunto.

Ejemplo y seudocódigo

Vamos a considerar el problema de obtener el valor numérico de la integral de

$$\iint_{\Omega} \sin \sqrt{\ln(x + y + 1)} dx dy = \iint_{\Omega} f(x, y) dx dy$$

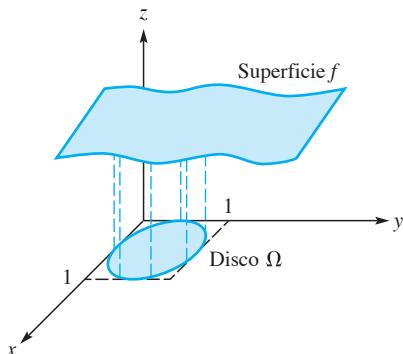


FIGURA 13.6
Dibujo de la superficie f (x, y) sobre el disco Ω

en el disco en el espacio xy , definido por la desigualdad

$$\Omega = \left\{ (x, y) : \left(x - \frac{1}{2} \right)^2 + \left(y - \frac{1}{2} \right)^2 \leq \frac{1}{4} \right\}$$

En la figura 13.6, se muestra un dibujo de este dominio, con una superficie arriba de este. A continuación generamos puntos aleatorios en el cuadrado y descartamos los que no están en el disco. Tomamos $n = 5000$ puntos en el disco. Si los puntos son $p_i = (x_i, y_i)$, entonces la integral se calcula como

$$\begin{aligned} \iint_{\Omega} f(x, y) dx dy &\approx (\text{área del disco } \Omega) \times \left(\begin{array}{l} \text{promedio de altura de } f \\ \text{en } n \text{ puntos aleatorios} \end{array} \right) \\ &= (\pi r^2) \left[\frac{1}{n} \sum_{i=1}^n f(p_i) \right] \\ &= \frac{\pi}{4n} \sum_{i=1}^n f(p_i) \end{aligned}$$

El seudocódigo para este ejemplo es el siguiente. Se imprimen cálculos intermedios de la integral cuando n es un múltiplo de 1000. Esto nos da una idea de cómo el valor correcto se acerca a nuestro proceso de promediado.

```

program Integral_Doble
integer i, j; real sum, vol, x, y; real array (rij)1:n×1:2
integer n ← 5000, iprt ← 1000; external function f
call Random((rij))
j ← 0; sum ← 0
for i = 1 to n do
    x = ri,1; y = ri,2
    if (x - 1/2)2 + (y - 1/2)2 ≤ 1/4 then
        j ← j + 1
        sum ← sum + f(x, y)
        if mod(j, iprt) = 0 then
            vol ← (π/4) sum / real(j)
            output j, vol

```

```

        end if
    end if
end for
vol ← ( $\pi/4$ ) sum / real(j)
output j, vol
end program Integral_Doble

real function f(x, y)
real x, y
f ← sin( $\sqrt{\ln(x + y + 1)}$ )
end function

```

Se obtiene un valor aproximado de 0.57 para la integral.

Cálculo de volúmenes

El volumen de una región complicada en tres dimensiones se puede calcular con una técnica de Monte Carlo. Tomando un caso simple, vamos a determinar el volumen de la región cuyos puntos satisfacen las desigualdades

$$\begin{cases} 0 \leq x \leq 1 & 0 \leq y \leq 1 & 0 \leq z \leq 1 \\ x^2 + \sin y \leq z \\ x - z + e^y \leq 1 \end{cases}$$

El primer renglón define un cubo cuyo volumen es 1. La región definida por *todas* las desigualdades dadas es, por tanto, un subconjunto de este cubo. Si se generan n puntos aleatorios en el cubo y se determina que m de ellos satisfacen las dos últimas desigualdades, entonces el volumen de la región deseada es aproximadamente m/n . Aquí se presenta un seudocódigo que realiza este procedimiento:

```

program Volumen de una Región
integer i, m;  real array (rij)1:n×1:3;  real vol, x, y, z
integer n ← 5000, iprt ← 1000
call Random((rij))
for i = 1 to n do
    x ← ri,1
    y ← ri,2
    z ← ri,3
    if  $x^2 + \sin y \leq z, x - z + e^y \leq 1$  then m ← m + 1
    if mod(i, iprt) = 0 then
        vol ← real(m)/ real(i)
        output i, vol
    end if
end for
end program Volumen de una Región

```

Observe que los cálculos intermedios se imprimen cuando se llega a 1000, 2000, . . . , 5000 puntos. Se determina un valor aproximado de 0.14 para el volumen de la región.

Ejemplo del barquillo de helado

Considere el problema de encontrar el volumen que hay arriba del cono $z^2 = x^2 + y^2$ y dentro de la esfera $x^2 + y^2 + (z - 1)^2 = 1$ como se muestra en la figura 13.7. El volumen está contenido en la caja delimitada por $-1 \leq x \leq 1$, $-1 \leq y \leq 1$ y $0 \leq z \leq 2$, que tiene un volumen de 8. Por lo tanto, queremos generar puntos aleatorios dentro de esta caja y multiplicar por 8 la razón de los que están dentro del volumen deseado al número total generado. Un seudocódigo para hacer esto es el siguiente:

```

program Cono
integer i, m; real vol, x, y, z; real array (rij)1:n×1:3
integer n ← 5000, iprt ← 1000; m ← 0
call Random((rij))
for i = 1 to n do
    x ← 2ri,1 - 1; y ← 2ri,2 - 1; z ← 2ri,3
    if x2 + y2 ≤ z2, x2 + y2 ≤ z(2 - z) then m ← m + 1
    if mod(i, iprt) = 0 then
        vol ← 8 real(m)/ real(i)
        output i, vol
    end if
end for
end program Cono

```

El volumen del cono es de aproximadamente 3.3.

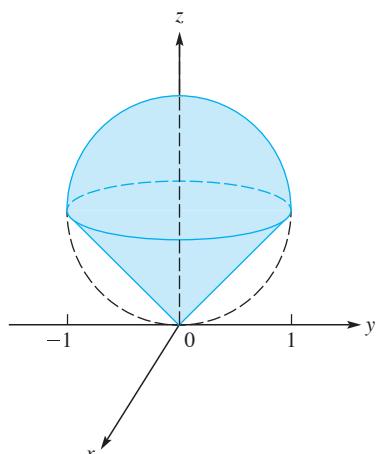


FIGURA 13.7
Región del
barquillo de
helado

Resumen

(1) Analizamos la aproximación de las integrales por el método de Monte Carlo para calcular áreas y volúmenes. Usamos

$$\int_0^1 f(x) dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i)$$

$$\int_0^1 \int_0^1 \int_0^1 f(x, y, z) dx dy dz \approx \frac{1}{n} \sum_{i=1}^n f(x_i, y_i, z_i)$$

donde $\{x_i\}$ es una sucesión de números aleatorios en el intervalo unitario y (x_i, y_i, z_i) es una sucesión aleatoria de n puntos en el cubo unitario.

(2) En general, tenemos

$$\int_A f \approx (\text{medida de } A) \times (\text{promedio de } f \text{ en } n \text{ puntos aleatorios en } A)$$

Problemas 13.2

- 1.** Se propone calcular π usando el método de Monte Carlo. Un círculo de radio 1 está dentro de un cuadrado de lado 2. Contamos cuántos m puntos aleatorios en el cuadrado caen en el círculo. Suponga que el error es $1/\sqrt{m}$. ¿Cuántos puntos se deben tomar para obtener π con tres dígitos de precisión (es decir, 3.142)?

Problemas de cómputo 13.2

- Ejecute los códigos dados en esta sección en su sistema de cómputo y compruebe que producen respuestas razonables.
- Escriba y pruebe un programa para evaluar la integral $\int_0^1 e^x dx$ con el método de Monte Carlo, con $n = 25, 50, 100, 200, 400, 800, 16\,000$ y $32\,000$. Observe que se necesitan $32\,000$ números aleatorios y que el trabajo de cada caso se puede utilizar en el caso siguiente. Imprima la respuesta exacta. Trace la gráfica de los resultados utilizando una escala logarítmica para mostrar la tasa de crecimiento.
- Escriba un programa para comprobar numéricamente que $\pi = \int_0^2 (4 - x^2)^{1/2} dx$. Utilice el método de Monte Carlo y 2500 números aleatorios.
- Utilice el método de Monte Carlo para aproximar la integral

$$\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (x^2 + y^2 + z^2) dx dy dz$$

Compare con la respuesta correcta.

"5. Escriba un programa para calcular

$$\int_0^2 \int_3^6 \int_{-1}^1 (yx^2 + z \log y + e^x) dx dy dz$$

6. Utilizando la técnica de Monte Carlo, escriba un seudocódigo para aproximar la integral

$$\iiint_{\Omega} (e^x \sin y \log z) dx dy dz$$

donde Ω es el cilindro circular que tiene una altura de 3 y una base circular $x^2 + y^2 \leq 4$.

"7. Calcule el área bajo la curva $y = e^{-(x+1)^2}$ y en el interior del triángulo que tiene vértices $(1, 0)$, $(0, 1)$ y $(-1, 0)$ escribiendo y probando un programa corto.

8. Utilizando el método de Monte Carlo, encuentre el área de la figura irregular definida por

$$\begin{cases} 1 \leq x \leq 3 & -1 \leq y \leq 4 \\ x^3 + y^3 \leq 29 \\ y \geq e^x - 2 \end{cases}$$

"9. Utilice el método de Monte Carlo para calcular el volumen de los sólidos, cuyos puntos (x, y, z) satisfacen

$$\begin{cases} 0 \leq x \leq y & 1 \leq y \leq 2 & -1 \leq z \leq 3 \\ e^x \leq y \\ (\sin z)y \geq 0 \end{cases}$$

"10. Utilizando una técnica de Monte Carlo, calcule el área de la región determinada por las desigualdades $0 \leq x \leq 1$, $10 \leq y \leq 13$, $y \geq 12 \cos x$ y $y \geq 10 + x^3$. Imprima las respuestas intermedias.

11. Utilice el método de Monte Carlo para aproximar las integrales siguientes.

a. $\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (x^2 - y^2 - z^2) dx dy dz$

b. $\int_1^4 \int_2^5 (x^2 - y^2 + xy - 3) dx dy$

c. $\int_2^3 \int_{1+y}^{\sqrt{y}} (x^2 y + xy^2) dx dy$ d. $\int_0^1 \int_{y^2}^{\sqrt{y}} \int_0^{y+z} xy dx dy dz$

12. El valor de la integral

$$\int_0^{\pi/4} \int_0^{2 \cos \phi} \int_0^{2\pi} \rho^2 \sin \phi d\theta d\rho d\phi$$

utilizando coordenadas esféricas es el volumen que hay arriba del cono $z^2 = x^2 + y^2$ y dentro de la esfera $x^2 + y^2 + (z - 1)^2 = 1$. Utilice el método de Monte Carlo para aproximar esta integral y compare los resultados con el del ejemplo del libro.

13. Sea que R denote la región en el plano xy definida por las desigualdades

$$\begin{cases} \frac{1}{3} \leq 3x \leq 9 - y \\ \sqrt{x} \leq y \leq 3 \end{cases}$$

Calcule la integral

$$\iint_R (e^x + \cos xy) dx dy$$

- 14.** Utilizando una técnica de Monte Carlo, calcule el área de la región definida por las desigualdades $4x^2 + 9y^2 \leq 36$ y $y \leq \arctan(x+1)$.

- 15.** Escriba un programa para calcular el área de la región definida por las desigualdades

$$\begin{cases} x^2 + y^2 \leq 4 \\ |y| \leq e^x \end{cases}$$

- 16.** Una integral se puede calcular con la fórmula

$$\int_0^1 f(x) dx \approx \frac{1}{n} \sum_{i=1}^n f(x_i)$$

aun si las x_i no son números aleatorios; de hecho, algunas sucesiones no aleatorias se pueden mejorar. Use la sucesión $x_i = (\text{parte fraccionaria de } i\sqrt{2})$ y pruebe el método de integración numérica correspondiente. Pruebe si los cálculos convergen a la razón $1/n$ o $1/\sqrt{n}$ usando algunos ejemplos simples, como $\int_0^1 e^x dx$ y $\int_0^1 (1+x^2)^{-1} dx$.

- 17.** Consideremos el elipsoide

$$\frac{x^2}{4} + \frac{y^2}{16} + \frac{z^2}{4} = 1$$

- a. Escriba un programa para generar y almacenar 5000 puntos aleatorios distribuidos uniformemente en el primer octante de este elipsoide.
 b. Escriba un programa para calcular el volumen de este elipsoide en el primer octante.

- 18.** Un método de Monte Carlo para calcular $\int_a^b f(x) dx$ si $f(x) \geq 0$ es como sigue. Sea $c \geq \max_{a \leq x \leq b} f(x)$. Después de generar n puntos aleatorios (x, y) en el rectángulo $a \leq x \leq b$, $0 \leq y \leq c$. Cuente el número k de estos puntos aleatorios (x, y) que satisfacen $y \leq f(x)$. Entonces $\int_a^b f(x) dx \approx kc(b-a)/n$. Compruebe esto y pruebe el método en $\int_1^2 x^2 dx$, $\int_0^1 (2x^2 - x + 1) dx$ y $\int_0^1 (x^2 + \sin 2x) dx$.

- 19.** (Continuación) Utilice el método del problema de cómputo 13.2.18 para calcular el valor de $\pi = 4 \int_0^1 \sqrt{1-x^2} dx$. Genere puntos aleatorios en $0 \leq x \leq 1$, $0 \leq y \leq 1$. Use $n = 1000, 2000, \dots, 10\,000$ y trate de determinar si el error se comporta como $1/\sqrt{n}$.

- 20.** (Continuación) Modifique el método descrito en el problema de cómputo 13.2.19 para manejar el caso cuando f toma valores positivos y negativos en $[a, b]$. Pruebe el método en $\int_{-1}^1 x^3 dx$.

- 21.** Otro método de Monte Carlo para la evaluación de $\int_a^b f(x) dx$ es el siguiente. Genere un número impar de números aleatorios en (a, b) . Reordene estos puntos de manera que $a < x_1 < x_2 < \dots < x_n < b$. Ahora calcule

$$f(x_1)(x_2 - a) + f(x_3)(x_4 - x_2) + f(x_5)(x_6 - x_4) + \dots + f(x_n)(b - x_{n-1})$$

Pruebe este método en

$$\int_0^1 (1+x^2)^{-1} dx \quad \int_0^1 (1-x^2)^{-1/2} dx \quad \int_0^1 x^{-1} \sin x dx$$

22. ¿Cuál es el valor esperado del volumen de un tetraedro formado por cuatro puntos elegidos aleatoriamente en el interior del tetraedro cuyos vértices son $(0, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ y $(1, 0, 0)$? (La respuesta exacta no se conoce!)

23. Escriba un programa para calcular el área bajo la curva $y = \sin x$ y por encima de la curva $y = \ln(x + 2)$. Utilice el método de Monte Carlo e imprima los resultados intermedios.

24. Calcule la integral

$$\int_{3.2}^{5.9} \left(\frac{e^{\sin x + x^2}}{\ln x} \right) dx$$

por el método de Monte Carlo.

25. Pruebe el generador de números aleatorios que esté a su disposición en la forma siguiente. Comience por crear una lista de N de números aleatorios r_k , uniformemente distribuidos en el intervalo $[0, 1]$. Cree una lista de números enteros aleatorios n_k mediante la extracción de la parte entera de $10 r_k$ para $1 \leq k \leq N$. Calcule los elementos en una matriz (m_{ij}) de 10×10 , donde m_{ij} es el número de veces i seguido por j en la lista (n_k) . Compare estos números con los valores predichos por la teoría de probabilidad elemental. Si es posible, muestre gráficamente los valores de m_{ij} .

26. (Proyecto de investigación estudiantil) Investigue algunos de los últimos avances en los métodos de Monte Carlo para la integración de múltiples variables.

13.3 Simulación

A continuación se ilustra la idea de la **simulación**. Consideremos una situación física en la que un elemento aleatorio está presente e intente imitar la situación en la computadora. Se pueden sacar conclusiones estadísticas si el experimento se realiza muchas veces. Las aplicaciones incluyen la simulación de servidores, clientes y colas que puedan producirse en empresas, tales como bancos o tiendas departamentales.

Problema del dado cargado

En los problemas de simulación, con frecuencia se deben producir variables aleatorias con una distribución determinada. Supongamos, por ejemplo, que queremos simular el lanzamiento de un dado cargado y que las probabilidades de los diferentes resultados se han determinado como se muestra:

Salida	1	2	3	4	5	6
Probabilidad	0.2	0.14	0.22	0.16	0.17	0.11

Si la variable aleatoria x se distribuye uniformemente en el intervalo $(0, 1)$, entonces al descomponer este intervalo en seis subintervalos de las longitudes dadas por la tabla, se puede simular el lanzamiento de este dado cargado. Por ejemplo, estamos de acuerdo en que si x está en $(0, 0.2)$, el dado muestra 1; si x está en $[0.2, 0.34)$, el dado muestra 2, y así sucesivamente. Un pseudocódigo para contar los resultados de 5000 lanzamientos del dado y calcular la probabilidad puede escribirse

como:

```

program Dado_Cargado
integer i, j; real array (yi)1:6, (mi)1:6, (ri)1:n
real n ← 5000
(yi)6 ← (0.2, 0.34, 0.56, 0.72, 0.89, 1.0)
(mi)6 ← (0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
call Random((ri))
for i = 1 to n do
    for j = 1 to 6 do
        if ri < yj then
            mj ← mj + 1
            exit loop j
        end if
    end for
end for
output real(mi)/ real(n)
end program Dado_Cargado

```

Los resultados son 0.2024, 0.1344, 0.2252, 0.1600, 0.1734 y 0.1046, que son aproximaciones razonables a las probabilidades de la tabla.

Problema del cumpleaños

Un problema interesante que se puede resolver usando simulación es el famoso **problema del cumpleaños**. Supongamos que en una habitación hay n personas y cada uno de los 365 días del año tiene la misma probabilidad de ser el cumpleaños de alguien. De la teoría de probabilidad se puede demostrar que, contrariamente a la intuición, sólo 23 personas necesitan estar presentes para que las posibilidades sean cincuenta y cincuenta, de que al menos dos de ellas tengan la misma fecha de cumpleaños! (Siempre es divertido hacer este experimento en una fiesta grande o en la clase para ver cómo funciona en la práctica).

Muchas personas sienten curiosidad por conocer el razonamiento teórico detrás de este resultado, por lo que lo analizaremos brevemente antes de resolver el problema de la simulación. Despues se le pregunta a alguien su cumpleaños, las posibilidades de que la siguiente persona a la que se le pregunte no tenga la misma fecha de cumpleaños es 364/365. Las posibilidades de que el cumpleaños de la tercera persona no coincida con el de las dos primeras personas son 363/365. La posibilidad de que dos eventos sucesivos independientes ocurran es el producto de la probabilidad de los eventos por separado. (La naturaleza secuencial de la explicación no implica que los eventos sean dependientes.) En general, la probabilidad de que la enésima persona a la que se le pregunte tenga un cumpleaños diferente al de cualquiera que ya se le ha preguntado es

$$\left(\frac{365}{365}\right) \left(\frac{364}{365}\right) \left(\frac{363}{365}\right) \cdots \left(\frac{365 - (n - 1)}{365}\right)$$

La probabilidad de que la enésima persona a quien se pregunta será una coincidencia es 1 menos este valor. Una tabla de la cantidad de $1 - (365)(364) \cdots [365 - (n - 1)]/365^n$ muestra que con 23 personas, las posibilidades son 50.7%; con 55 o más personas, las posibilidades son 98.6 % o casi la certeza de que, teóricamente, al menos dos de cada 55 personas tendrá el mismo cumpleaños (véase la tabla 13.1).

Sin utilizar la teoría de probabilidad, podemos escribir una rutina que emplee el generador de números aleatorios para calcular la probabilidad aproximada de los grupos de n personas.

TABLA 13.1 Problema del cumpleaños

<i>n</i>	Teórico	Simulación
5	0.027	0.028
10	0.117	0.110
15	0.253	0.255
20	0.411	0.412
22	0.476	0.462
23	0.507	0.520
25	0.569	0.553
30	0.706	0.692
35	0.814	0.819
40	0.891	0.885
45	0.941	0.936
50	0.970	0.977
55	0.986	0.987

Evidentemente, todo lo que se necesita es seleccionar n enteros aleatorios del conjunto $\{1, 2, 3, \dots, 365\}$ y examinarlos de alguna manera para determinar si hay una coincidencia. Al repetir este experimento un gran número de veces podemos calcular la probabilidad de que al menos hay una coincidencia en cualquier reunión de n personas.

Una forma de escribir una rutina para simular el problema del cumpleaños es la siguiente. En ella se utiliza un método de comprobación de días en un calendario para saber si hay una coincidencia. Por supuesto, hay muchas otras maneras de abordar este problema.

El procedimiento de función *Probably* calcula la probabilidad de cumpleaños repetidos:

```

real function Probably(n, npts)
integer i, npts; logical Birthday; real sum  $\leftarrow$  0
for i = 1 to npts do
    if Birthday (n) then sum  $\leftarrow$  sum + 1
end for
Probably  $\leftarrow$  sum/real(npts)
end function Probably

```

La función lógica *Birthday* genera n números aleatorios y los compara. Devuelve un valor de **true** (verdadero) si estos números tienen al menos uno repetido y **false** (falso) si todos los n números son diferentes.

```

logical function Birthday(n)
integer i, n, number; logical array (daysi)1:365
real array (ri)1:n
call Random((ri))
for i = 1 to 365 do
    days(i)  $\leftarrow$  false
end for

```

```

Birthday ← false
for  $i = 1$  to  $n$  do
     $number \leftarrow$  integer ( $365r_i + 1$ )
    if  $days(number)$  then
         $Birthday \leftarrow$  true
        exit loop  $i$ 
    end if
     $days(number) \leftarrow$  true
end for
end function Birthday

```

Los resultados de los cálculos teóricos y la simulación se presentan en la tabla 13.1.

Problema de la aguja de Buffon

El siguiente ejemplo de una simulación es un problema muy antiguo conocido como **problema de la aguja de Buffon**. Imagine que una aguja de longitud unitaria se ha caído en una hoja de papel cuadriculado por líneas paralelas a 1 unidad de distancia entre si. ¿Cuál es la probabilidad de que la aguja corte una de las rectas?

Para hacer el problema concreto, suponga que el centro de la aguja aterriza entre las rectas en un punto aleatorio. Supongamos también que la orientación angular de la aguja es otra variable aleatoria. Por último, imagine que nuestras variables aleatorias provienen de una distribución uniforme. La figura 13.8 muestra la geometría de la situación.

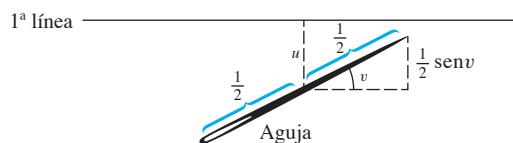


FIGURA 13.8
Problema de la
aguja de Buffon

2^a línea

Sea u la distancia del centro de la aguja a la más cercana de las dos rectas y sea v el ángulo de la horizontal. Aquí, u y v son dos variables aleatorias. La aguja corta una de las líneas si y sólo si $u \leq \frac{1}{2} \operatorname{sen} v$. Realizamos el experimento muchas veces, por ejemplo, 5000. Debido a la simetría del problema, se selecciona u de una distribución aleatoria uniforme en el intervalo $(0, \frac{1}{2})$ y v de una distribución aleatoria uniforme en el intervalo $(0, \pi/2)$, y determinamos el número de veces que $2u \leq \operatorname{sen} v$. Hacemos $w = 2u$ y probamos $w \leq \operatorname{sen} v$, donde w es una variable aleatoria en $(0, 1)$. En este programa, las respuestas intermedias se imprimen de manera que se pueda observar su progresión. También, se imprime la respuesta teórica, $t = 2/\pi \approx 0.63662$, con fines comparativos.

```

program Aguja
integer  $i, m$ ; real  $prob, v, w$ ; real array  $(r_{ij})_{1:n \times 1:2}$ 
integer  $n \leftarrow 5000$ ,  $iprt \leftarrow 1000$ 

```

```

 $m \leftarrow 0$ 
call Random( $(r_{ij})$ )
for  $i = 1$  to  $n$  do
     $w \leftarrow r_{i1}$ 
     $v \leftarrow (\pi/2)r_{i,2}$ 
    if  $w \leq \sin v$  then  $m \leftarrow m + 1$ 
    if mod( $i, iprt$ ) = 0 then
         $prob \leftarrow \text{real}(m)/\text{real}(i)$ 
        output  $i, prob, (2/\pi)$ 
    end if
end for
end program Aguja

```

Problema de dos dados

De nuevo, nuestro siguiente ejemplo tiene una solución analítica. Esto es una ventaja para nosotros porque queremos comparar los resultados de las simulaciones de Monte Carlo con las soluciones teóricas. Considere el experimento de lanzar dos dados. Para un dado (sin cargar), los números 1, 2, 3, 4, 5 y 6 tienen la misma probabilidad de ocurrir. Nos preguntamos: *¿cuál es la probabilidad de lanzar un 12 (es decir, un 6 en cada dado) en 24 tiradas de los dados?*

Hay seis posibles resultados de cada dado de un total de 36 combinaciones posibles. Sólo una de estas combinaciones es un doble 6, de modo que 35 de las 36 combinaciones no son correctas. Con 24 tiradas, tenemos que $(35/36)^{24}$ es la probabilidad de un resultado equivocado. Por tanto, $1 - (35/36)^{24} = 0.49140$ es la respuesta. No todos los problemas de este tipo se pueden analizar como este, por lo que modelamos la situación utilizando un generador de números aleatorios.

Si simulamos este proceso, un solo experimento consiste en lanzar los dados 24 veces y este experimento se debe repetir un gran número de veces, digamos, 1000. Para el resultado de la tirada de un solo dado, necesitamos enteros aleatorios que estén distribuidos uniformemente en el conjunto $\{1, 2, 3, 4, 5, 6\}$. Si x es una variable aleatoria en $(0, 1)$, entonces $6x + 1$ es una variable aleatoria en $(1, 7)$ y la parte entera es un número entero aleatorio en $\{1, 2, 3, 4, 5, 6\}$. Aquí se presenta un seudocódigo:

```

program Dos_Dados
integer  $i, j, i_1, i_2, m$ ; real prob; real array ( $r_{ijk}$ ) $_{1:n \times 1:24 \times 1:2}$ 
integer  $n \leftarrow 5000$ , iprt  $\leftarrow 1000$ 
call Random( $(r_{ijk})$ )
 $m \leftarrow 0$ 
for  $i = 1$  to  $n$  do
    for  $j = 1$  to 24 do
         $i_1 \leftarrow \text{integer}(6r_{i1} + 1)$ 
         $i_2 \leftarrow \text{integer}(6r_{i2} + 1)$ 
        if  $i_1 + i_2 = 12$  then
             $m \leftarrow m + 1$ 
            exit loop  $j$ 
        end if
    end for

```

```

if mod(i, 1000) = 0 then
    prob ← real(m)/ real(i)
    output i, prob
end if
end for
end program Dos_Dados

```

Este programa calcula la probabilidad de sacar un 12 en 24 tiradas de los dados aproximadamente a *ganancias parejas*, es decir, 0.487.

Escudo de neutrones

Nuestro último ejemplo se refiere al escudo de neutrones. Tomamos un modelo simple de neutrones penetrando en una pared de plomo. Se supone que cada neutrón entra en la pared de plomo en un ángulo recto con la pared y recorre una distancia unitaria. Entonces choca con un átomo de plomo y rebota en una dirección aleatoria. Una vez más, viaja una distancia unitaria antes de chocar con otro átomo de plomo. Rebota en una dirección aleatoria y así sucesivamente. Supongamos que después de ocho colisiones, se gasta toda la energía de los neutrones. Supongamos también que la pared de plomo tiene 5 unidades de espesor en la dirección x y para todos los propósitos prácticos espesor infinito en la dirección y . La pregunta es: *¿qué porcentaje de los neutrones se puede esperar que emergan del otro lado de la pared de plomo?* (véase la figura 13.9).

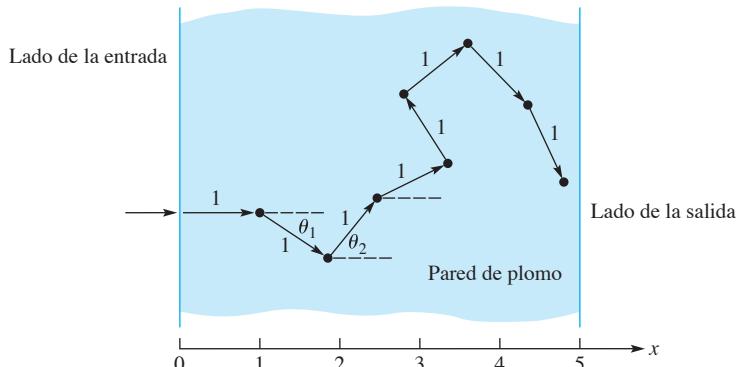


FIGURA 13.9
Experimento
del escudo de
neutrones

Sea x la distancia medida desde la superficie inicial, donde entra el neutrón. De trigonometría, recordamos que en un triángulo rectángulo con hipotenusa 1, un lado es $\cos \theta$. También observe que $\cos \theta \leq 0$ cuando $\pi/2 \leq \theta \leq \pi$ (figura 13.10). La primera colisión se produce en un punto

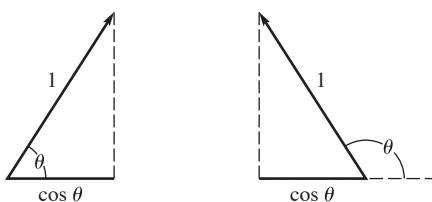


FIGURA 13.10
Triángulos
rectángulos con
hipotenusa 1

donde $x = 1$. La segunda ocurre en un punto cuando $x = 1 + \cos \theta$. La tercera colisión ocurre en un punto donde $x = 1 + \cos \theta_1 + \cos \theta_2$, y así sucesivamente. Si $x \geq 5$, el neutrón ha salido. Si $x < 5$ para todas las ocho colisiones, la pared ha protegido el área de un neutrón particular. Para una simulación de Monte Carlo, por simetría podemos usar ángulos aleatorios θ_i en el intervalo $(0, \pi)$. Entonces el programa de simulación es el siguiente:

```

program Escudo
integer i, j, m; real x, per; real array (rij)1:n×1:7
integer n ← 5000, iprt ← 1000
m ← 0
call Random((rij)) for i = 1 to n do
    x ← 1
    for j = 1 to 7 do
        x ← x + cos(πrij)
        if x ≤ 0 then exit loop j
        if x ≥ 5 then
            m ← m + 1
            exit loop j
        end if
    end for
    if mod(i, iprt) = 0 then
        per ← 100 real(m)/ real(i)
        output i, per
    end if
end for
end program Escudo

```

Después de ejecutar este programa, podemos decir que aproximadamente se puede esperar que el 1.85% de los neutrones emerjan de la pared de plomo.

Resumen

Los generadores de números aleatorios se utilizan en la **simulación** de una situación física en la que está presente un elemento aleatorio. Se pueden obtener conclusiones estadísticas si el experimento numérico se realiza muchas veces.

Referencias adicionales

Véase Bayer y Diaconis [1992], Chaitlin [1975], Evans *et al.* [1967], Flehinger [1966], Gentle [2003], Greenbaum [2002], Hammersley y Handscomb [1964], Hansen *et al.* [1993], Hull y Dobell [1962], Kinderman y Monahan [1977], Leva [1992], Marsaglia [1968], Marsaglia y Tsang [2000], Niederreiter [1978, 1992], Peterson [1997], Raimi [1969], Schrage [1979], Sobol [1994], Steele [1997].

Problemas de cómputo 13.3

1. Un punto (a, b) se elige aleatoriamente en un rectángulo definido por las desigualdades $|a| \leq 1$ y $|b| \leq 2$. ¿Cuál es la probabilidad de que la ecuación cuadrática que resulta $ax^2 + bx + 1 = 0$ tenga raíces reales? Encuentre la respuesta tanto analítica como por el método de Monte Carlo.

2. Calcule la distancia media entre dos puntos en el círculo $x^2 + y^2 = 1$. Para solucionar esto, genere N pares aleatorios de puntos (x_i, y_i) y (v_i, w_i) en el círculo, y calcule

$$N^{-1} \sum_{i=1}^N [(x_i - v_i)^2 + (y_i - w_i)^2]^{1/2}$$

3. (Sistema de ferrocarriles franceses) Defina la distancia entre dos puntos (x_1, y_1) y (x_2, y_2) en el plano es $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ si los puntos están en una línea recta que pasa por el origen, sino $\sqrt{x_1^2 + y_1^2} + \sqrt{x_2^2 + y_2^2}$ en todos los demás casos. Haga un dibujo para ilustrar. Calcule la distancia media entre dos puntos seleccionados aleatoriamente en un círculo unitario con centro en el origen.

4. Considere un círculo de radio 1. Se elige un punto aleatoriamente dentro del círculo y una cuerda que tiene el punto elegido como punto medio. ¿Cuál es la probabilidad de que la cuerda tenga una longitud superior a $\frac{3}{2}$? Resuelva el problema de manera analítica y por el método de Monte Carlo.

5. Se seleccionan dos puntos aleatoriamente en la circunferencia de un círculo. ¿Cuál es la distancia promedio entre el centro del círculo y el centro de gravedad de los dos puntos?

6. Considere la *cardioide* dada por $(x^2 + y^2 + x)^2 = (x^2 + y^2)$. Escriba un programa para encontrar la distancia media, *mantiéndose dentro de la cardioide*, entre dos puntos seleccionados aleatoriamente dentro de la figura. Use 1000 puntos e imprima los cálculos intermedios.

7. Encuentre la longitud de la *lemniscata* cuya ecuación en coordenadas polares viene dada por $r^2 = \cos 2\theta$. *Sugerencia:* en coordenadas polares, $ds^2 = dr^2 + r^2 d\theta^2$.

8. Supongamos que un dado está cargado de tal modo que las seis caras no son igualmente probables de aparecer cuando se rueda el dado. Las probabilidades asociadas con las seis caras son las siguientes:

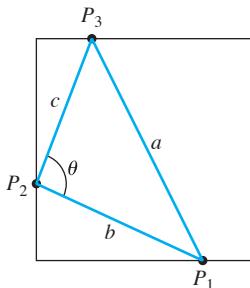
Resultado	1	2	3	4	5	6
Probabilidad	0.15	0.2	0.25	0.15	0.1	0.15

Escriba y ejecute un programa para simular 1500 lanzamientos de dicho dado.

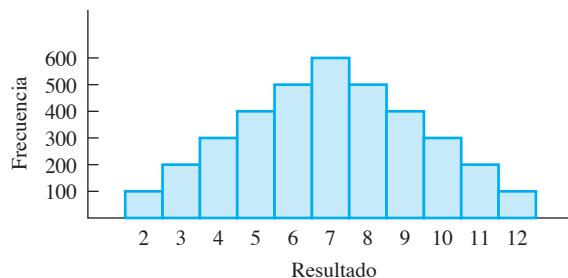
9. Considere un par de dados cargados como se describen en el libro. Mediante simulación de Monte Carlo, determine la probabilidad de sacar un 12 de 25 tiradas de los dados.

10. Considere el problema del escudo de neutrones similar al del libro, pero modificado como sigue. Imagine que el haz de neutrones incide en la pared a 1 unidad por encima de su base. La pared puede ser muy alta. Los neutrones no pueden escapar por la parte superior, pero pueden hacerlo por la parte inferior, así como por el lado de salida. Encuentre el porcentaje de neutrones que escapan.

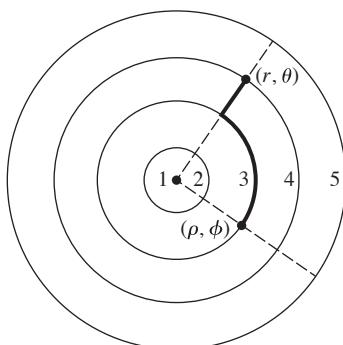
- 11.** Reescriba la rutina(s) para el problema de cumpleaños usando algún otro método para determinar si existe o no una coincidencia.
- 12.** Escriba un programa para calcular la probabilidad de que los tres puntos aleatorios en los bordes de un cuadrado formen un triángulo obtuso (véase la figura). *Sugerencia:* utilice la ley de los cosenos: $\cos \theta = (b^2 + c^2 - a^2)/2bc$.



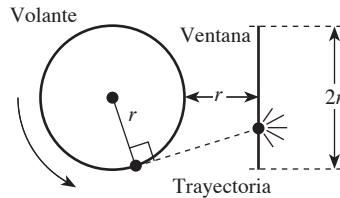
- 13.** Un **histograma** es un dispositivo gráfico para la visualización de frecuencias por medio de rectángulos cuyas alturas son proporcionales a las frecuencias. Por ejemplo, en el lanzamiento de dos dados 3600 veces, las cantidades resultantes 2, 3, . . . , 12 deberían ocurrir con frecuencias similares a las que se muestran en el histograma que aparece a continuación. Por medio de la simulación de Monte Carlo, obtenga un histograma de la frecuencia de los dígitos 0, 1, . . . , 9 que se presentan en 1000 números aleatorios.



- 14.** Considere una ciudad circular de diámetro de 20 kilómetros (véase la siguiente figura). Radiando desde el centro hay 36 caminos rectos, espaciados con un ángulo 10° de separación. Hay también 20 caminos circulares separados 1 kilómetro. ¿Cuál es la distancia promedio, medida a lo largo de los caminos, entre los puntos de intersección de los caminos en la ciudad?



- 15.** Una partícula se desprende de un punto aleatorio de un volante de rotación. Mire la siguiente figura y determine la probabilidad de que la partícula pegue en la ventana. Realice una simulación de Monte Carlo para calcular la probabilidad de una manera experimental.



- 16.** Escriba un programa para simular el siguiente fenómeno. Una partícula se mueve en el plano xy bajo el efecto de una fuerza aleatoria. Inicia en $(0, 0)$. Al final de cada segundo, se mueve 1 unidad en una dirección aleatoria. Queremos registrar en una tabla su posición al final de cada segundo, tomando un total de 1000 segundos.
- 17. (Una caminata aleatoria)** En una noche con viento, un borracho comienza a caminar en el origen de sistema bidimensional de coordenadas. Sus pasos son de 1 unidad de longitud y son aleatorios de la siguiente manera. Con probabilidad $\frac{1}{6}$, da un paso al este, con probabilidad $\frac{1}{4}$, da un paso hacia el norte, con probabilidad $\frac{1}{3}$, da un paso al sur y con probabilidad $\frac{1}{4}$, toma un paso al oeste. ¿Cuál es la probabilidad de que después de 50 pasos esté a más de 20 unidades de distancia del origen? Escriba un programa para simular este problema.
- 18. (Otra caminata aleatoria)** Considere los puntos de la red (puntos con coordenadas enteras) en el cuadrado $0 \leq x \leq 6, 0 \leq y \leq 6$. Una partícula inicia en el punto $(4, 4)$ y se mueve de la siguiente manera. En cada paso, se mueve con igual probabilidad en uno de los cuatro puntos de la red adyacente. ¿Cuál es la probabilidad de que cuando la primera partícula cruce el límite del cuadrado, lo haga la parte de abajo? Utilice la simulación de Monte Carlo.
- 19.** ¿Cuál es la probabilidad de que dentro de 20 generaciones, la familia Kzovck se extinga? Utilice los siguientes datos. En la primera generación, sólo hay un hombre Kzovck. En cada generación, la probabilidad de que un hombre Kzovck tenga exactamente un hijo varón es $\frac{4}{11}$, la probabilidad de que tenga exactamente dos es $\frac{1}{11}$ y la probabilidad de que tenga más de dos es 0.
- 20.** Escriba un programa que simule el barajeado aleatorio de una baraja de 52 cartas.
- 21.** Un carrusel, con un total de 24 caballos, permite que los niños salten hacia adentro en las tres puertas y salten hacia afuera en una sola puerta, mientras que continúa girando lentamente. Si los niños suben y bajan aleatoriamente (a lo más uno por puerta), ¿cuántas revoluciones debe esperar alguien después de una vuelta para subirse? Suponga una probabilidad de $\frac{1}{2}$ de que un niño se suba o se baje.
- 22.** Ejecute los programas que se presentan en esta sección y determine si los resultados son razonables.
- 23.** En el cubo unitario $\{(x, y, z): 0 \leq x \leq 1, 0 \leq y \leq 1, 0 \leq z \leq 1\}$, si los dos puntos son elegidos aleatoriamente, entonces ¿cuál es la distancia esperada entre ellos?
- 24.** Los puntos de la red en el plano se definen como los puntos cuyas coordenadas son enteros. Un círculo de diámetro 1.5 se deja caer en el plano de tal forma que su centro es un punto aleatorio uniformemente distribuido en el cuadrado $0 \leq x \leq 1, 0 \leq y \leq 1$. ¿Cuál es la

probabilidad de que dos o más puntos de la red se encuentran dentro del círculo? Use la simulación de Monte Carlo para calcular una respuesta aproximada.

25. Escriba un programa para simular un problema de flujo de tráfico similar a la del ejemplo con que empieza este capítulo.
26. ¿Puede modificar y ejecutar de nuevo los programas esta sección para que no se utilicen matrices de gran tamaño?
27. (**Proyecto de investigación estudiantil**) En su artículo *Trailing the Dovetail Shuffle to its Lair*, Bayer y Diaconis [1992] demuestran que se necesita siete **barajeados** para aleatorizar un mazo de cartas. Greenbaum [2002] utiliza esto como un ejemplo de aplicación de cascos numéricos de polinomios de diversos grados asociados con la matriz de transición de probabilidad. Este es el fenómeno de corte que se observa a menudo en procesos de Markov.* Mediante **sucesiones crecientes** y modelado matemático, barajar se ilustra en

www.math.washington.edu/~chartier/Shuffle

Investigue algunas de las siguientes preguntas en esta página web. *¿Cuántas veces tenemos que barajar un mazo de cartas antes de que el orden de estas sea suficientemente aleatorio? ¿Existe algún número mínimo de barajeados necesarios para asegurar que la baraja no está ordenada o no es predecible? ¿Hay un punto donde continuar barajeando ya no ayuda a que la baraja sea menos previsible?*

* Las **cadenas de Markov** se pueden utilizar para modelar el comportamiento de un sistema que sólo depende de su estado anterior. Las cadenas de Markov implican una **matriz de transición** $P = (P_{ij})$, donde las entradas son la probabilidad de pasar de un estado j a un estado i .

Problemas con valores en la frontera para ecuaciones diferenciales ordinarias

En el diseño de ejes y cojinetes, el ingeniero mecánico se encuentra con el problema siguiente: la sección transversal de un pivote está determinada por una curva $y = y(x)$ que debe pasar por dos puntos fijos $(0, 1)$ y $(1, a)$, como se muestra en la figura 14.1. Además, para un rendimiento óptimo (principalmente de baja fricción), la función desconocida debe minimizar el valor de una integral dada

$$\int_0^1 [y(y')^2 + b(x)y^2] dx$$

en la que $b(x)$ es una función conocida. Con base en esto, es posible obtener una ecuación diferencial de segundo orden (la así llamada *ecuación de Euler*) para y . La ecuación diferencial con sus valores inicial y final es

$$\begin{cases} -(y')^2 - 2b(x)y + 2yy'' = 0 \\ y(0) = 1 \quad y(1) = a \end{cases}$$

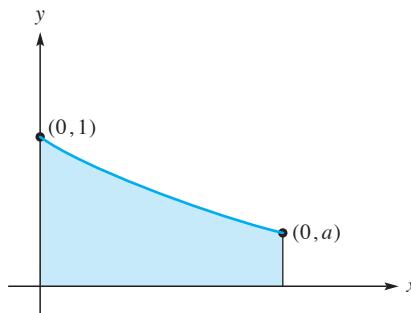


FIGURA 14.1
Sección
transversal del
pivoté

Se trata de un problema no lineal con dos valores en la frontera y los métodos para resolverlo numéricamente se analizan en este capítulo.

14.1 Método de disparo

En los capítulos anteriores tratamos el problema con valor inicial para ecuaciones diferenciales ordinarias, pero ahora consideraremos otro tipo de problema numérico relacionado con las ecuaciones diferenciales ordinarias. Un **problema con valor en la frontera** se ejemplifica con una ecuación diferencial ordinaria de segundo orden, cuya solución se establece en función de los extremos

del intervalo de interés. Un ejemplo de este problema es

$$\begin{cases} x'' = -x \\ x(0) = 1 \quad x\left(\frac{\pi}{2}\right) = -3 \end{cases}$$

En este caso, tenemos una ecuación diferencial cuya solución general implica dos parámetros arbitrarios. Para especificar una solución particular deben darse dos condiciones. Si este fuera un problema con valor inicial, x y x' estarían dados en algún punto inicial. Sin embargo, en este problema, se nos dan dos puntos de la forma $(t, x(t))$ por los cuales pasa la curva solución, a saber, $(0, 1)$ y $(\pi/2, -3)$. La solución general de la ecuación diferencial es $x(t) = c_1 \sin(t) + c_2 \cos(t)$, y las dos condiciones (conocidas como **valores frontera**) nos permiten determinar que $c_1 = -3$ y $c_2 = 1$.

Ahora supongamos que tenemos un problema similar en el que no podemos determinar la solución general como en el anterior. Tomamos como modelo el problema

$$\begin{cases} x''(t) = f(t, x(t), x'(t)) \\ x(a) = \alpha \quad x(b) = \beta \end{cases} \quad (1)$$

Una solución numérica paso a paso del problema (1) mediante los métodos del capítulo 11 requiere dos condiciones iniciales, pero en el problema (1) sólo una condición está presente en $t = a$. Este hecho hace que un problema como (1) sea considerablemente más difícil que un problema con valores iniciales. En este capítulo se consideran varios modos de ataque. Teoremas de existencia y unicidad de soluciones de los problemas de dos puntos con valores en la frontera se pueden encontrar en Keller [1976].

Una forma de proceder para resolver el problema (1) es proponer $x'(a)$, después realizar la solución del problema resultante de valor inicial hasta b y esperar que la solución calculada concuerde con β , es decir, $x(b) = \beta$. Si no es así (lo que es bastante probable), se puede volver atrás y cambiar nuestra suposición de $x'(a)$. Repetir este procedimiento hasta que pegue en el blanco β puede ser un buen método si podemos aprender algo de los diferentes ensayos. Hay formas sistemáticas de usar esta información y el método resultante se conoce con el sobrenombre de **disparo**.

Observamos que el valor final $x(b)$ de la solución de nuestro problema con valor inicial depende de la suposición de que se hizo para $x'(a)$. Todo lo demás permanece fijo en este problema. Así, la ecuación diferencial $x'' = f(t, x, x')$ y el primer valor inicial, $x(a) = \alpha$, no cambian. Si asignamos un valor real z para la condición inicial que falta,

$$x'(a) = z$$

entonces el problema con valor inicial se puede resolver numéricamente. El valor de x en b es ahora una función de z , que denotamos por $\varphi(z)$. En otras palabras, para cada opción de z , se obtiene un nuevo valor de $x(b)$ y φ es el nombre de la función con este comportamiento. Sabemos muy poco acerca de $\varphi(z)$, pero se puede calcular o evaluar. Es, sin embargo, una función *cara* de evaluar, ya que cada valor de $\varphi(z)$ se obtiene sólo después de resolver un problema con valor inicial.

Cabe destacar que el método del disparo combina *cualquier* algoritmo para el problema con valor inicial con *cualquier* algoritmo para encontrar un cero de una función. La elección de estos dos algoritmos debe reflejar la naturaleza del problema por resolver.

La idea básica del método del disparo se ilustra en la figura 14.2. Las curvas solución se muestran, así como dos caminos con diferentes pendientes iniciales. El objetivo es seguir acercándose al objetivo inicial en cada intento.

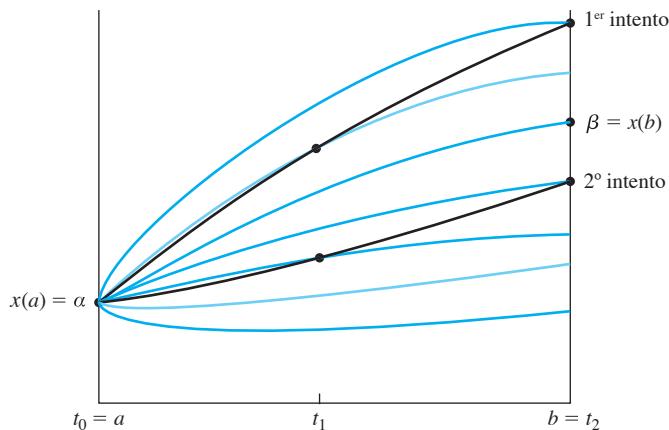


FIGURA 14.2
Ilustración del
método de
disparo

Algoritmo del método de disparo

En resumen, una función $\varphi(z)$ se calcula como sigue. Se resuelve el problema con valor inicial

$$\begin{cases} x'' = f(t, x(t), x'(t)) \\ x(a) = \alpha \quad x'(a) = z \end{cases}$$

en el intervalo $[a, b]$. Sea

$$\varphi(z) = x(b)$$

Nuestro objetivo es ajustar z hasta que encontremos un valor para el que

$$\varphi(z) = \beta$$

Una forma de hacerlo es utilizar la interpolación lineal entre $\varphi(z_1)$ y $\varphi(z_2)$, donde z_1 y z_2 son dos suposiciones para la condición inicial $x'(a)$. Es decir, dados dos valores de φ , se pretende que φ sea una *función lineal* y se determina un valor apropiado de z basado en esta hipótesis. Un boceto de los valores de z contra $\varphi(z)$ puede ser como se muestra en la figura 14.3. La estrategia que acabamos de exponer es el método de la secante para encontrar un cero de $\varphi(z) - \beta$.

Para obtener una fórmula para calcular el siguiente valor z_3 , calculamos $\varphi(z_1)$ y $\varphi(z_2)$, con base en los valores de z_1 y z_2 , respectivamente

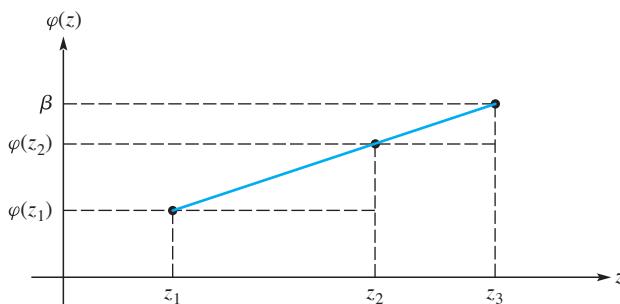


FIGURA 14.3
Función lineal φ

Al considerar triángulos semejantes, tenemos

$$\frac{z_3 - z_2}{\beta - \varphi(z_2)} = \frac{z_2 - z_1}{\varphi(z_2) - \varphi(z_1)}$$

donde

$$z_3 = z_2 + [\beta - \varphi(z_2)] \left[\frac{z_2 - z_1}{\varphi(z_2) - \varphi(z_1)} \right]$$

Podemos repetir este proceso y generar la sucesión

$$z_{n+1} = z_n + [\beta - \varphi(z_n)] \left[\frac{z_n - z_{n-1}}{\varphi(z_n) - \varphi(z_{n-1})} \right] \quad (n \geq 2) \quad (2)$$

todo basado en dos valores iniciales z_1 y z_2 .

Este procedimiento para la resolución del problema con valor en la frontera y dos puntos

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha \quad x(b) = \beta \end{cases} \quad (3)$$

es entonces el siguiente. Se resuelve el problema con valor inicial

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha \quad x'(a) = z \end{cases} \quad (4)$$

de $t = a$ a $t = b$, haciendo que el valor de la solución en b se denote por $\varphi(z)$. Se hace esto dos veces con dos diferentes valores de z , por ejemplo, z_1 y z_2 , y se calcula $\varphi(z_1)$ y $\varphi(z_2)$. Ahora se calcula una nueva z , llamada z_3 , con la fórmula (2). Despues se calcula $\varphi(z_3)$ de nuevo resolviendo (4). Se obtiene z_4 de z_2 y z_3 de la misma manera, y así sucesivamente. Se revisa

$$\varphi(z_{n+1}) - \beta$$

para ver si se están haciendo progresos. Cuando sea satisfactoriamente pequeño hay que detenerse. Este proceso se llama un **método de disparo**. Observe que los valores numéricamente obtenidos $x(t_i)$ por $a \leq t_i \leq b$ se deben guardar hasta que se obtengan mejores (es decir, uno cuyo valor final $x(b)$ esté más cerca de β que el actual), ya que el objetivo en la solución del problema (3) es obtener valores de $x(t)$ para los valores de t entre a y b .

El método del disparo puede durar mucho tiempo si cada solución de los problemas de valor inicial asociados supone un valor pequeño para el tamaño del paso h . Por lo tanto, usamos un valor relativamente grande de h hasta que $|\varphi(z_{n+1}) - \beta|$ sea suficientemente pequeño y luego se reduce h para obtener la precisión requerida.

EJEMPLO 1 ¿Cuál es la función φ para este problema de dos valores en la frontera?

$$\begin{cases} x'' = x \\ x(0) = 1 \quad x(1) = 7 \end{cases}$$

Solución La solución general de la ecuación diferencial es $x(t) = c_1 e^t + c_2 e^{-t}$. La solución del problema con valor inicial

$$\begin{cases} x'' = x \\ x(0) = 1 \quad x'(0) = z \end{cases}$$

es $x(t) = \frac{1}{2}(1+z)e^t + \frac{1}{2}(1-z)e^{-t}$. Por lo tanto, tenemos

$$\varphi(z) = x(1) = \frac{1}{2}(1+z)e + \frac{1}{2}(1-z)e^{-1}$$



Modificaciones y refinamientos

Muchas modificaciones y refinamientos son posibles. Por ejemplo, cuando $\varphi(z_{n+1})$ está cerca de β , se pueden utilizar fórmulas de interpolación de mayor orden para calcular los valores sucesivos de z_i . Supongamos, por ejemplo, que en lugar de utilizar dos valores $\varphi(z_1)$ y $\varphi(z_2)$ para obtener z_3 , usamos los cuatro valores

$$\varphi(z_1) \quad \varphi(z_2) \quad \varphi(z_3) \quad \varphi(z_4)$$

para calcular z_5 . Se podría establecer un polinomio cúbico de interpolación p_3 para los datos

z_1	z_2	z_3	z_4
$\varphi(z_1)$	$\varphi(z_2)$	$\varphi(z_3)$	$\varphi(z_4)$

(5)

y resolver

$$p_3(z_5) = \beta$$

para z_5 . Puesto que p_3 es cúbico, esto implica trabajo adicional. Una mejor manera puede ser establecer un polinomio \hat{p}_3 para interpolar los datos

$\varphi(z_1)$	$\varphi(z_2)$	$\varphi(z_3)$	$\varphi(z_4)$
z_1	z_2	z_3	z_4

(6)

Y después utilizar $\hat{p}_3(\beta)$ como el cálculo de z_5 . Este procedimiento se conoce como **interpolación inversa** (véase la sección 4.1).

Otras observaciones sobre el método de disparo se harán en la siguiente sección, después del análisis de un procedimiento alternativo.

Resumen

(1) Un problema genérico con valores en la frontera y dos puntos en el intervalo $[a, b]$ es

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha \quad x(b) = \beta \end{cases}$$

Hay un problema con valor inicial relacionado

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha \quad x'(a) = z \end{cases}$$

Esperamos encontrar un valor de z de modo que la solución calculada del problema con valor inicial será la solución del problema con valores en la frontera y dos puntos. Definimos una función $\varphi(z)$, cuyo valor es la solución calculada del problema con valor inicial en $t = b$, es decir, $\varphi(z) = x(b)$, donde x resuelve el problema con valor inicial. Repetidamente ajustamos z hasta que encontramos un valor para el que $\varphi(z) = \beta$. Si z_1 y z_2 son dos suposiciones para la condición inicial $x'(a)$, podemos utilizar la interpolación lineal entre $\varphi(z_1)$ y $\varphi(z_2)$ para encontrar un valor mejorado para z . Esto se

hace resolviendo el problema con valor inicial dos veces con z_1 y z_2 y así calculando $\varphi(z_1)$ y $\varphi(z_2)$. Calculamos una nueva z_3 usando

$$z_{n+1} = z_n + [\beta - \varphi(z_n)] \left[\frac{z_n - z_{n-1}}{\varphi(z_n) - \varphi(z_{n-1})} \right] \quad (n \geq 2)$$

y calculando $\varphi(z_{n+1})$ resolviendo de nuevo el problema con valor inicial. Supervisamos $\varphi(z_{n+1}) - \beta$ hasta que sea satisfactoriamente pequeño y entonces se para. Esto se llama el **método del disparo**.

(2) Las mejoras y refinamientos al método de disparo implican el uso de **interpolación polinomial cúbica** o **interpolación inversa**.

Problemas 14.1

1. Compruebe que $x = (2t + 1)e^t$ es la solución a cada uno de los siguientes problemas:

$$\begin{cases} x'' = x + 4e^t \\ x(0) = 1 \quad x\left(\frac{1}{2}\right) = 2e^{1/2} \end{cases} \quad \begin{cases} x'' = x' + x - (2t - 1)e^t \\ x(1) = 3e \quad x(2) = 5e^2 \end{cases}$$

- ^a2. Compruebe que $x = c_1 e^t + c_2 e^{-t}$ resuelve el problema con valor en la frontera

$$\begin{cases} x'' = x \\ x(0) = 1 \quad x(1) = 2 \end{cases}$$

si se eligen los valores adecuados de c_1 y c_2 .

3. Resuelva estos problemas con valores en la frontera mediante el ajuste de la solución general de la ecuación diferencial.

^aa. $x'' = x \quad x(0) = 0 \quad x(\pi) = 1$
^ab. $x'' = t^2 \quad x(0) = 1 \quad x(1) = -1$

4. ^aa. Determine todos los pares (α, β) para los que el problema

$$\begin{cases} x'' = -x \\ x(0) = \alpha \quad x\left(\frac{\pi}{2}\right) = \beta \end{cases}$$

tiene una solución.

- ^ab. Repita el inciso a para $x(0) = \alpha$ y $x(\pi) = \beta$.

5. a. Compruebe el siguiente algoritmo para la técnica de interpolación inversa sugerida en el libro. En este caso tenemos $\varphi_i = \varphi(z_i)$.

$$\begin{aligned} u &= \frac{z_2 - z_1}{\varphi_2 - \varphi_1} & v &= \frac{s - u}{\varphi_3 - \varphi_1} & s &= \frac{z_3 - z_2}{\varphi_3 - \varphi_2} \\ r &= \frac{e - s}{\varphi_4 - \varphi_2} & e &= \frac{z_4 - z_3}{\varphi_4 - \varphi_3} & w &= \frac{r - v}{\varphi_4 - \varphi_1} \\ z_5 &= z_1 + (\beta - \varphi_1)\{u + (\beta - \varphi_2)[v + w(\beta - \varphi_3)]\} \end{aligned}$$

- b. Encuentre fórmulas similares para tres puntos.

- ^a6. Sea $\varphi(z)$ que denota a $x(\pi/2)$, donde x es la solución del problema con valor inicial

$$\begin{cases} x'' = -x \\ x(0) = 0 \quad x'(0) = z \end{cases}$$

¿Qué es $\varphi(z)$?

- 7.** Determine la función φ de forma explícita en el caso de este problema de valores en la frontera y dos puntos.

$$\begin{cases} x'' = -x \\ x(0) = 1 \quad x\left(\frac{\pi}{2}\right) = 3 \end{cases}$$

- 8.** (Continuación) Repita el problema anterior para $x'' = -(x')^2/x$ con $x(1) = 3$ y $x(2) = 5$. Usando su resultado, resuelva el problema con valores en la frontera. *Sugerencia:* la solución general de la ecuación diferencial es $x(t) = c_1\sqrt{c_2 + t}$.

- 9.** Determine la función φ de forma explícita en el caso de este problema con valores en la frontera y dos puntos:

$$\begin{cases} x'' = x \\ x(-1) = e \quad x'(1) = \frac{1}{2}e \end{cases}$$

- 10.** Los problemas con valores en la frontera pueden implicar ecuaciones diferenciales de orden superior a 2. Por ejemplo,

$$\begin{cases} x''' = f(t, x, x', x'') \\ x(a) = \alpha \quad x'(a) = \gamma \quad x(b) = \beta \end{cases}$$

Analice las formas en que se puede resolver este problema utilizando el método de disparo.

- 11.** Resuelva analíticamente este problema con valores en la frontera y tres puntos:

$$\begin{cases} x''' = -e^t + 4(t+1)^{-3} \\ x(0) = -1 \quad x(1) = 3 - e + 2 \ln 2 \quad x(2) = 6 - e^2 + 2 \ln 3 \end{cases}$$

- 12.** Resuelva

$$\begin{cases} x'' = -x \\ x(0) = 2 \quad x(\pi) = 3 \end{cases}$$

analíticamente y analice las dificultades.

- 13.** Muestre que los dos problemas siguientes son equivalentes en el sentido de que a partir de la solución de uno es fácil obtener la del otro:

$$\begin{cases} y''' = f(t, y) \\ y(0) = \alpha \quad y(1) = \beta \end{cases} \quad \begin{cases} z'' = f(t, z + \alpha - \alpha t + \beta t) \\ z(0) = 0 \quad z(1) = 0 \end{cases}$$

- 14.** Analice en términos generales la solución numérica de los siguientes problemas con valores en la frontera y dos puntos. Recomiende métodos específicos para cada uno, asegurándose de aprovechar cualquier estructura especial.

a. $\begin{cases} x'' = \operatorname{sent} + (e^t \sqrt{t^2 + 1})x + (\cos t)x' \\ x(0) = 0 \quad x(1) = 5 \end{cases}$

b. $\begin{cases} x'_1 = x_1^2 + (t-3)x_1 + \operatorname{sent} \\ x'_2 = x_2^3 + \sqrt{t^2 + 1} + (\cos t)x_1 \\ x_1(0) = 1 \quad x_2(2) = 3 \end{cases}$

“15. ¿Qué es $\varphi(z)$ en el caso de este problema con valor en la frontera?

$$\begin{cases} x'' = -x \\ x(0) = 1 \quad x(\pi) = 3 \end{cases}$$

Explique las implicaciones.

- 16.** Encuentre la función φ explícitamente para este problema con valores en la frontera y dos puntos:

$$\begin{cases} x'' = e^{-2t} - 4x - 4x' \\ x(0) = 1 \quad x(2) = 0 \end{cases}$$

¿Cuál es el problema con valor inicial cuya solución resuelve el problema con valor en la frontera? *Sugerencia:* encuentre una solución de la forma $x(t) = q(t)e^{-2t}$, donde q es un polinomio de segundo grado.

Problemas de cómputo 14.1

- 1.** El problema *no lineal* con valores en la frontera y dos puntos

$$\begin{cases} x'' = e^x \\ x(0) = \alpha \quad x(1) = \beta \end{cases}$$

tiene la solución en forma cerrada

$$x = \ln c_1 - 2 \ln \left\{ \cos \left[\left(\frac{1}{2} c_1 \right)^{1/2} t + c_2 \right] \right\}$$

donde c_1 y c_2 son las soluciones de

$$\begin{cases} \alpha = \ln c_1 - 2 \ln \cos c_2 \\ \beta = \ln c_1 - 2 \ln \left\{ \cos \left[\left(\frac{1}{2} c_1 \right)^{1/2} + c_2 \right] \right\} \end{cases}$$

Utilice el método de disparo para resolver este problema con $\alpha = \beta = \ln 8\pi^2$. Inicie con $z_1 = -\frac{25}{2}$ y $z_2 = -\frac{23}{2}$. Determine c_1 y c_2 para que se pueda hacer una comparación con la verdadera solución. *Nota:* el método de discretización correspondiente, como se analiza en la siguiente sección, implica un sistema de ecuaciones no lineales con una solución de forma no cerrada.

- 2.** Escriba un programa para resolver el ejemplo con que empieza este capítulo para a y $b(x)$ dadas, tales que $a = \frac{1}{4}$ y $b(x) = x^2$.

14.2 Un método de discretización

Aproximaciones por diferencias finitas

Pasamos ahora a un método completamente diferente de resolución del problema con valores en la frontera y dos puntos: el basado en una **discretización** directa de la ecuación diferencial.

El problema que queremos resolver es

$$\begin{cases} x'' = f(t, x, x') \\ x(a) = \alpha \quad x(b) = \beta \end{cases} \quad (1)$$

Se selecciona un conjunto de puntos igualmente espaciados t_0, t_1, \dots, t_n en el intervalo $[a, b]$, haciendo

$$t_i = a + ih \quad \text{y} \quad h = \frac{b-a}{n} \quad (0 \leq i \leq n)$$

A continuación, se aproximan las derivadas, utilizando las fórmulas de diferencia central estándares (5) y (20) de la sección 4.3:

$$\begin{aligned} x'(t) &\approx \frac{1}{2h}[x(t+h) - x(t-h)] \\ x''(t) &\approx \frac{1}{h^2}[x(t+h) - 2x(t) + x(t-h)] \end{aligned} \quad (2)$$

El valor aproximado de $x(t_i)$ se denota por x_i . Por lo tanto, el problema se convierte en

$$\begin{cases} x_0 = \alpha \\ \frac{1}{h^2}(x_{i-1} - 2x_i + x_{i+1}) = f\left(t_i, x_i, \frac{1}{2h}(x_{i+1} - x_{i-1})\right) \quad (1 \leq i \leq n-1) \\ x_n = \beta \end{cases} \quad (3)$$

Esto suele ser un sistema de ecuaciones no lineales con $n-1$ incógnitas x_1, x_2, \dots, x_{n-1} porque f generalmente implica las x_i en una forma no lineal. La solución de este sistema no suele ser fácil, pero se podría tratar usando los métodos del capítulo 3.

El caso lineal

En algunos casos, el sistema (3) es lineal. Esta situación se produce exactamente cuando f en la ecuación (1) tiene la forma

$$f(t, x, x') = u(t) + v(t)x + w(t)x' \quad (4)$$

En este caso especial, la ecuación principal del sistema (3) se ve como:

$$\frac{1}{h^2}(x_{i-1} - 2x_i + x_{i+1}) = u(t_i) + v(t_i)x_i + w(t_i)\left[\frac{1}{2h}(x_{i+1} - x_{i-1})\right]$$

o, equivalentemente,

$$-\left(1 + \frac{h}{2}w_i\right)x_{i-1} + (2 + h^2v_i)x_i - \left(1 - \frac{h}{2}w_i\right)x_{i+1} = -h^2u_i \quad (5)$$

donde $u_i = u(t_i)$, $v_i = v(t_i)$ y $w_i = w(t_i)$. Ahora sean

$$\begin{cases} a_i = -\left(1 + \frac{h}{2}w_i\right) \\ d_i = 2 + h^2v_i \\ c_i = -\left(1 - \frac{h}{2}w_i\right) \\ b_i = -h^2u_i \end{cases} \quad (0 \leq i \leq n)$$

Entonces, la ecuación principal (5) se convierte en

$$a_i x_{i-1} + d_i x_i + c_i x_{i+1} = b_i$$

Las ecuaciones correspondientes a $i = 1$ e $i = n - 1$ son diferentes, ya que conocemos x_0 y x_n . El sistema por lo tanto puede escribirse como

$$\begin{cases} d_1 x_1 + c_1 x_2 = b_1 - a_1 \alpha \\ a_i x_{i-1} + d_i x_i + c_i x_{i+1} = b_i \\ a_{n-1} x_{n-2} + d_{n-1} x_{n-1} = b_{n-1} - c_{n-1} \beta \end{cases} \quad (2 \leq i \leq n-2) \quad (6)$$

En forma matricial, el sistema (6) se ve así:

$$\begin{bmatrix} d_1 & c_1 & & & & \\ a_2 & d_2 & c_2 & & & \\ a_3 & d_3 & c_3 & & & \\ \ddots & \ddots & \ddots & & & \\ & a_{n-2} & d_{n-2} & c_{n-2} & & \\ & a_{n-1} & d_{n-1} & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} b_1 - a_1 \alpha \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ b_{n-1} - c_{n-1} \beta \end{bmatrix}$$

Puesto que este sistema es tridiagonal, podemos tratar de resolverlo con el procedimiento especial *Tri* para los sistemas tridiagonales, desarrollados en la sección 7.3. Sin embargo este procedimiento no incluye pivoteo y puede fallar en los casos en que el procedimiento Gauss tendría éxito (véase el problema 14.2.5).

Seudocódigo y ejemplo numérico

Las ideas que acabamos de explicar se utilizan ahora para escribir un programa para un caso de prueba específico. El problema es de la forma (1) con una función *lineal f* como en la ecuación (4):

$$\begin{cases} x'' = e^t - 3\sin(t) + x' - x \\ x(1) = 1.09737491 \quad x(2) = 8.63749661 \end{cases} \quad (7)$$

La solución, conocida de antemano, es $x(t) = e^t - 3\cos(t)$ y se puede utilizar para comprobar la solución informática. Utilizamos la técnica de discretización descrita anteriormente y el procedimiento *Tri* para resolver el sistema lineal resultante.

Primero, decidimos utilizar 100 puntos, incluidos los extremos $a = 1$ y $b = 2$. Así, $n = 99$, $h = \frac{1}{99}$ y $t_i = 1 + ih$ para $0 \leq i \leq 99$. Entonces tenemos $t_0 = 1$, $x_0 = x_0(t_0) = 1.09737491$, $t_{99} = 2$ y $x_{99} = x(t_{99}) = 8.63749661$. Las incógnitas de nuestro problema son los valores restantes de x_i , a saber, x_1, x_2, \dots, x_{98} . En la discretización de las derivadas usando las fórmulas de diferencia central (2), obtenemos un sistema lineal de tipo (3). Nuestra ecuación principal es de la forma (5) y es

$$-\left(1 + \frac{h}{2}\right)x_{i-1} + (2 - h^2)x_i - \left(1 - \frac{h}{2}\right)x_{i+1} = -h^2 [e^{t_i} - 3\sin(t_i)]$$

ya que $u(t) = e^t - 3\sin t$, $v(t) = -1$ y $w(t) = 1$.

Generalizamos el seudocódigo de manera que con sólo unos cuantos cambios puede adaptarse a cualquier problema con dos valores en la frontera del tipo (1) con el miembro derecho de la forma (4). En este caso, $u(x)$, $v(x)$ y $w(x)$ son funciones.

```

program BVP1
integer i; real error,h,t,u,v,w,x
real array (ai)1:n,(bi)1:n,(ci)1:n,(di)1:n,(yi)1:n
integer n ← 99
real ta ← 1, tb ← 2, α ← 1.09737 491, β ← 8.63749 661
u(x) = ex - 3 sin(x)
v(x) = -1
w(x) = 1
h ← (tb - ta)/n
for i = 1 to n - 1 do
    t ← ta + ih
    ai ← -[1 + (h/2)w(t)]
    di ← 2 + h2v(t)
    ci ← -[1 - (h/2)w(t)]
    bi ← -h2u(t)
end for
b1 ← b1 - a1α
bn-1 ← bn-1 - cn-1β
for i = 1 to n - 1 do
    ai ← ai+1
end
call Tri(n - 1,(ai),(di),(ci),(bi),(yi))
error ← eta - 3 cos(ta) - α
output ta,α,error
for i = 1 to n - 1 step 9 do
    t ← ta + ih
    error ← et - 3 cos(t) - yi
    output t,yi,error
end for
error ← etb - 3 cos(tb) - β
output b,β,error
end program BVP1

```

Los resultados de computadora son los siguientes:

Valor de t	Solución	Error
1.000000 00	1.09737 49	0.00
1.090909 91	1.59203 02	-8.83 × 10 ⁻⁵
1.181818 82	2.12274 17	-1.74 × 10 ⁻⁴
1.272727 73	2.68980 86	-2.56 × 10 ⁻⁴
1.363636 64	3.29367 04	-3.28 × 10 ⁻⁴
1.454545 55	3.93494 53	-3.76 × 10 ⁻⁴
1.545454 45	4.61449 10	-4.06 × 10 ⁻⁴
1.636363 36	5.33343 17	-4.13 × 10 ⁻⁴
1.727272 27	6.09319 59	-3.89 × 10 ⁻⁴
1.818181 18	6.89557 22	-3.16 × 10 ⁻⁴
1.909091 10	7.74277 78	-1.88 × 10 ⁻⁴
2.000000 00	8.63749 69	0.00

Método de disparo en el caso lineal

Acabamos de ver que este método de discretización (también llamado **método de diferencias finitas**) es bastante sencillo, en el caso del problema lineal con dos valores en la frontera:

$$\begin{cases} x'' = u(t) + v(t)x + w(t)x' \\ x(a) = \alpha \quad x(b) = \beta \end{cases} \quad (8)$$

El método de disparo también es especialmente sencillo en este caso. Recordemos que este método de disparo nos obliga a resolver un problema con valor inicial:

$$\begin{cases} x'' = u(t) + v(t)x + w(t)x' \\ x(a) = \alpha \quad x'(a) = z \end{cases} \quad (9)$$

e interpretar al valor terminal $x(b)$ como una función de z . Se llama a esa función φ y se busca un valor de z para el que $\varphi(z) = \beta$. Para el problema lineal (9), φ es una función *lineal* de z y así la figura 14.3 de la sección 14.1 es efectivamente real. Por consiguiente, sólo tenemos que resolver el problema (9) con dos valores de z para determinar la función precisamente. Para establecer estos hechos, hagamos un poco más de análisis.

Supongamos que hemos resuelto el problema (9) dos veces con determinados valores z_1 y z_2 . Sean las soluciones así obtenidas denotadas por $x_1(t)$ y $x_2(t)$. Entonces decimos que la función

$$g(t) = \lambda x_1(t) + (1 - \lambda)x_2(t) \quad (10)$$

tiene propiedades

$$\begin{cases} g'' = u + vg + wg' \\ g(a) = \alpha \end{cases}$$

lo que se deja comprobar al lector en el problema 14.2.6. (El valor de λ en este análisis es una constante, pero es totalmente arbitraria.)

La función g casi resuelve el problema de dos valores a la frontera (8) y g contiene un parámetro λ a nuestra disposición. Imponiendo la condición $g(b) = \beta$ obtenemos

$$\lambda x_1(b) + (1 - \lambda)x_2(b) = \beta$$

de la que

$$\lambda = \frac{\beta - x_2(b)}{x_1(b) - x_2(b)}$$

Tal vez la forma más sencilla de poner en práctica estas ideas es resolver dos problemas con valores iniciales

$$\begin{cases} x'' = u(t) + v(t)x + w(t)x' \\ x(a) = \alpha \quad x'(a) = 0 \end{cases}$$

y

$$\begin{cases} y'' = u(t) + v(t)y + w(t)y' \\ y(a) = \alpha \quad y'(a) = 1 \end{cases}$$

Entonces, la solución del problema original con dos valores en la frontera (8) es

$$\lambda x(t) + (1 - \lambda)y(t) \quad \text{con} \quad \lambda = \frac{\beta - y(b)}{x(b) - y(b)} \quad (11)$$

En la realización computacional de este procedimiento debemos guardar las curvas solución x y y completas. Se guardan en arreglos (x_i) y (y_i) .

Seudocódigo y ejemplo numérico

Como un ejemplo de método de disparo, considere el problema de la ecuación (7). Resolvemos los problemas con dos valores iniciales

$$\begin{cases} x'' = e^t - 3 \operatorname{sen}(t) + x' - x \\ x(1) = 1.09737491 \\ x'(1) = 0 \end{cases} \quad \begin{cases} y'' = e^t - 3 \operatorname{sen}(t) + y' - y \\ y(1) = 1.09737491 \\ y'(1) = 1 \end{cases} \quad (12)$$

utilizando el método de Runge-Kutta de cuarto orden. Para ello, se introducen las variables

$$x_0 = t \quad x_1 = x \quad x_2 = x'$$

Entonces el primer problema con valor inicial es

$$\begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} 1 \\ x_2 \\ e^{x_0} - 3 \operatorname{sen}(x_0) + x_2 - x_1 \end{bmatrix} \quad \begin{bmatrix} x_0(1) \\ x_1(1) \\ x_2(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1.09737491 \\ 0 \end{bmatrix}$$

Ahora hacemos

$$y_0 = t \quad y_1 = y \quad y_2 = y'$$

El segundo problema con valor inicial que tenemos que resolver es similar, excepto que modificamos al vector inicial

$$\begin{bmatrix} y'_0 \\ y'_1 \\ y'_2 \end{bmatrix} = \begin{bmatrix} 1 \\ y_2 \\ e^{y_0} - 3 \operatorname{sen}(y_0) + y_2 - y_1 \end{bmatrix} \quad \begin{bmatrix} y_0(1) \\ y_1(1) \\ y_2(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1.09737491 \\ 1 \end{bmatrix}$$

Es más eficiente resolver estos dos problemas juntos como un solo sistema. Introduciendo

$$x_3 = y \quad x_4 = y'$$

en el primer sistema, tenemos

$$\begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix} = \begin{bmatrix} 1 \\ x_2 \\ e^{x_0} - 3 \operatorname{sen}(x_0) + x_2 - x_1 \\ x_4 \\ e^{x_0} - 3 \operatorname{sen}(x_0) + x_4 - x_3 \end{bmatrix} \quad \begin{bmatrix} x_0(1) \\ x_1(1) \\ x_2(1) \\ x_3(1) \\ x_4(1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1.09737491 \\ 0 \\ 1.09737491 \\ 1 \end{bmatrix}$$

Es evidente que las componentes $x_1(t)$ y $x_3(t)$ del vector solución en toda t satisfacen el primer y el segundo problema, respectivamente. En consecuencia, la solución es

$$\lambda x_1(t_i) + (1 - \lambda) x_3(t_i) \quad (1 \leq i \leq n - 1)$$

donde

$$\lambda = \frac{8.63749661 - x_3(2)}{x_1(2) - x_3(2)}$$

Usamos 100 puntos, como antes, de manera que $n = 99$.

```

program BVP2
integer i; real array (xi)0:m,(x1i)0:n,(x3i)0:n; real error,h,p,q,t
integer n ← 99, m ← 4
real a ← 1, b ← 2, α ← 1.09737 491, β ← 8.63749 661
x ← (1,α,0,α,1)
h ← (b-a)/n
for i = 1 to n do
    call RK4_System2(m,h,(xi),1)
    (x1)i ← x1
    (x3)i ← x3
end for
p ← [β - (x3)n]/[(x1)n - (x3)n]
q ← 1 - p
for i = 1 to n do
    (x1)i ← p(x1)i + q(x3)i
end for
error ← ea - 3 cos(a) - α
output a,α,error
for i = 9 to n step 9 do
    t ← a + ih
    error ← et - 3 cos(t) - (x1)i
    output t,(x1)i,error
end for
end program BVP2

procedure Sistema_XP(m,(xi),(fi))
real array (xi)0:m,(fi)0:m
f0 ← 1
f1 ← x2
f2 ← ex0 - 3 sin(x0) + x2 - x1
f3 ← x4
f4 ← ex0 - 3 sin(x0) + x4 - x3
end procedure Sistema_XP

```

Los resultados finales de la computadora son como se muestran:

Valor de t	Solución	Error
1.00000 00	1.09737 49	0.00
1.09090 91	1.59194 09	9.54 × 10 ⁻⁷
1.18181 82	2.12256 57	1.91 × 10 ⁻⁶
1.27272 73	2.68955 09	1.43 × 10 ⁻⁶
1.36363 64	3.29334 26	2.38 × 10 ⁻⁷
1.45454 55	3.93456 79	9.54 × 10 ⁻⁷
1.54545 45	4.61408 57	-4.77 × 10 ⁻⁷
1.63636 36	5.33301 78	4.77 × 10 ⁻⁷
1.72727 27	6.09280 54	1.91 × 10 ⁻⁶
1.81818 18	6.89525 56	9.54 × 10 ⁻⁷
1.90909 10	7.74258 90	9.54 × 10 ⁻⁷
2.00000 00	8.63749 69	0.00

Observe que los errores son menores que los obtenidos en el método de discretización para el mismo problema. (¿Por qué?)

Usando software matemático como el que se encuentra en Matlab, Maple o Mathematica, este problema puede resolverse de varias maneras. En Matlab y Mathematica, se pueden utilizar las rutinas incorporadas para obtener la solución numérica a este problema con valores en la frontera y trazar la curva solución. Por otro lado, Maple puede resolver las dos ecuaciones diferenciales que hay en (12) y combinar las soluciones de la forma descrita anteriormente, con un valor adecuado para λ . También, el código puede evaluar la solución a 1, 1.5 y 2, por ejemplo. Observe que esta es una solución *analítica*. Estos sistemas de software matemático no producen la solución instantáneamente, hay una gran cantidad de cálculo detrás del telón.

En nuestro breve análisis de los problemas con dos valores en la frontera, no hemos tocado la difícil cuestión de la existencia de soluciones. A veces, un problema con valor en la frontera no tiene solución a pesar de tener coeficientes suaves. En el problema 14.1.4b se presenta un ejemplo. Este comportamiento contrasta con el de los problemas con valor inicial. Estas cuestiones están fuera del alcance de este libro, pero se tratan, por ejemplo, en Keller [1976] y en Stoer y Bulirsch [1993].

Resumen

(1) Para los problemas con dos valores en la frontera

$$\begin{cases} x''(t) = f(t, x(t), x'(t)) \\ x(a) = \alpha \quad x(b) = \beta \end{cases}$$

usamos las diferencias finitas en el intervalo $[a, b]$ con $n + 1$ puntos, a saber, $t_i = a + ih$ con $0 \leq i \leq n$ y $h = (b - a)/n$. Obtenemos $x_0 = \alpha$, $x_n = \beta$, y

$$\frac{1}{h^2}(x_{i-1} - 2x_i + x_{i+1}) = f\left(t_i, x_i, \frac{1}{2h}(x_{i+1} - x_{i-1})\right) \quad (1 \leq i \leq n - 1)$$

El caso lineal de este problema se produce cuando el lado derecho es

$$f(t, x, x') = u(t) + v(t)x + w(t)x'$$

En este caso, la ecuación principal se convierte en

$$\frac{1}{h^2}(x_{i-1} - 2x_i + x_{i+1}) = u(t_i) + v(t_i)x_i + w(t_i)\left[\frac{1}{2h}(x_{i+1} - x_{i-1})\right]$$

Entonces la forma de cálculo es

$$-\left(1 + \frac{h}{2}w_i\right)x_{i-1} + (2 + h^2v_i)x_i - \left(1 - \frac{h}{2}w_i\right)x_{i+1} = -h^2u_i$$

donde $u_i = u(t_i)$, $v_i = v(t_i)$ y $w_i = w(t_i)$. Esto conduce a un sistema lineal tridiagonal para resolver.

(2) Considere el problema lineal con dos valores en la frontera

$$\begin{cases} x'' = u(t) + v(t)x + w(t)x' \\ x(a) = \alpha \quad x(b) = \beta \end{cases}$$

y el correspondiente problema con valor inicial

$$\begin{cases} x'' = u(t) + v(t)x + w(t)x' \\ x(a) = \alpha \quad x'(a) = z \end{cases}$$

Supongamos que x_1 y x_2 son dos curvas solución al problema con valor inicial con z_1 y z_2 , respectivamente. La solución del problema con dos valores a la frontera es

$$g(t) = \lambda x_1(t) + (1 - \lambda)x_2(t)$$

con

$$\lambda = \frac{\beta - x_2(b)}{x_1(b) - x_2(b)}$$

Entonces encontramos

$$\begin{cases} g'' = u + vg + wg' \\ g(a) = \alpha \quad g(b) = \lambda x_1(b) + (1 - \lambda)x_2(b) = \beta \end{cases}$$

Una manera sencilla de aplicar este método para resolver problemas con dos valores iniciales:

$$\begin{cases} x'' = u(t) + v(t)x + w(t)x' \\ x(a) = \alpha \quad x'(a) = 0 \end{cases} \quad \begin{cases} y'' = u(t) + v(t)y + w(t)y' \\ y(a) = \alpha \quad y'(a) = 1 \end{cases}$$

Entonces, la solución al problema con dos valores en la frontera original es

$$\lambda x(t) + (1 - \lambda)y(t) \quad \text{con} \quad \lambda = \frac{\beta - y(b)}{x(b) - y(b)}$$

Referencias adicionales

Véase Ascher, Mattheij y Russell [1995], Axelsson y Barker [2001], Keller [1968, 1976] y Stakgold [2000].

Problemas 14.2

^a1. Si las aproximaciones de diferencias finitas estándares a las derivadas se utilizan para resolver un problema con dos valores a la frontera $x'' = t + 2x - x'$, ¿cuál es la ecuación típica en el sistema lineal de ecuaciones resultante?

^a2. Considere el problema con dos valores a la frontera

$$\begin{cases} x'' = -x \\ x(0) = 0 \quad x(1) = 1 \end{cases}$$

Plantee y resuelva el sistema tridiagonal que surge del método de diferencias finitas cuando $h = \frac{1}{4}$. Explique las diferencias de la solución analítica en $x\left(\frac{1}{4}\right) \approx 0.29401$, $x\left(\frac{1}{2}\right) \approx 0.56975$ y $x\left(\frac{3}{4}\right) \approx 0.81006$.

3. Compruebe que la ecuación (11) da la solución del problema con valores en la frontera (8).

^a4. Considere el problema con dos valores en la frontera

$$\begin{cases} x'' = x^2 - t + tx \\ x(0) = 1 \quad x(1) = 3 \end{cases}$$

Supongamos que hemos resuelto el problema con dos valores iniciales

$$\begin{cases} u'' = u^2 - t + tu \\ u(0) = 1 \quad u'(0) = 1 \end{cases} \quad \begin{cases} v'' = v^2 - t + tv \\ v(0) = 1 \quad v'(0) = 2 \end{cases}$$

numéricamente y que se han encontrado valores terminales $u(l) = 2$ y $v(l) = 3.5$. ¿Cuál es un problema con valor inicial razonable para hacer el *siguiente* intento para tratar de resolver el problema original con dos valores?

- 5.** Considere el sistema tridiagonal (6). Demuestre que si $v_i > 0$, entonces existe alguna elección de h para la que la matriz es diagonalmente dominante.

- 6.** Establezca las propiedades enunciadas para la función g en la ecuación (10).

- 7.** Demuestre que para el problema simple

$$\begin{cases} x'' = -x \\ x(a) = \alpha \quad x(b) = \beta \end{cases}$$

el sistema tridiagonal por resolver se puede escribir como

$$\begin{cases} (2 - h^2)x_1 - x_2 = \alpha \\ -x_{i-1} + (2 - h^2)x_i - x_{i+1} = 0 \quad (2 \leq i \leq n-2) \\ -x_{n-2} + (2 - h^2)x_{n-1} = \beta \end{cases}$$

- 8.** Escriba el sistema de ecuaciones $Ax = b$ que resulta de utilizar la habitual aproximación de diferencia central de segundo orden para resolver

$$\begin{cases} x'' = (1+t)x \\ x(0) = 0 \quad x(1) = 1 \end{cases}$$

- 9.** Sea u una solución del problema con valor inicial

$$\begin{cases} u'' = e^t u + t^2 u' \\ u(1) = 0 \quad u'(1) = 1 \end{cases}$$

¿Cómo podemos resolver el siguiente problema con dos valores en la frontera utilizando u ?

$$\begin{cases} x'' = e^t x + t^2 x' \\ x(1) = 0 \quad x(2) = 7 \end{cases}$$

- 10.** ¿Cómo resolver el problema

$$\begin{cases} x' = f(t, x) \\ Ax(a) + Bx(b) = C \end{cases}$$

donde a, b, A, B y C son números reales dados? (Suponga que A y B no son ambos cero.)

- 11.** Utilice el método de disparo en este problema con dos valores en la frontera y explique lo que sucede:

$$\begin{cases} x'' = -x \\ x(0) = 3 \quad x(\pi) = 7 \end{cases}$$

Este problema se debe resolver de forma analítica, no por computadora o calculadora.

Problemas de cómputo 14.2

1. Explique los principales pasos en la creación de un programa para resolver este problema con dos valores en la frontera mediante el método de diferencias finitas.

$$\begin{cases} x'' = x \operatorname{sen} t + x' \cos t - e^t \\ x(0) = 0 \quad x(1) = 1 \end{cases}$$

Muestre cualquier trabajo preliminar que debe hacerse antes de la programación. Aproveche la linealidad de la ecuación diferencial. Programe y compare los resultados cuando se utilizan diferentes valores de n , por ejemplo, $n = 10, 100$ y 1000 .

2. Resuelva numéricamente el problema con dos valores en la frontera. Compare con las soluciones exactas dadas.

a. $\begin{cases} x'' = \frac{(1-t)x + 1}{(1+t)^2} \\ x(0) = 1 \quad x(1) = 0.5 \end{cases}$

b. $\begin{cases} x'' = \frac{1}{3} [(2-t)e^{2x} + (1+t)^{-1}] \\ x(0) = 0 \quad x(1) = -\log 2 \end{cases}$

3. Resuelva el problema con valores en la frontera

$$\begin{cases} x'' = -x + tx' - 2t \cos t + t \\ x(0) = 0 \quad x(\pi) = \pi \end{cases}$$

por discretización. Compare con la solución exacta, que es $x(t) = t + 2 \operatorname{sen} t$.

4. Repita el problema de cómputo 14.1.2, usando un método de discretización.

5. Escriba un programa de cómputo para implementar el

- a. programa BVP1. b. programa BVP2.

6. (Continuación) Usando las rutinas incorporadas en sistemas de software matemático como Matlab, Maple o Mathematica, resuelva y trace la curva solución para el problema con valor en la frontera asociado con el

- a. programa BVP1. b. programa BVP2.

7. Investigue el cálculo de las soluciones numéricas a los siguientes problemas desafiantes, que no son lineales:

a. $\begin{cases} x'' = e^x \\ x(0) = 0, x(1) = 0 \end{cases}$

b. $\begin{cases} \varepsilon x'' + (x')^2 = 1 \\ x(0) = 0, x(1) = 1 \end{cases}$

Varíe $\varepsilon = 10^{-1}, 10^{-2}, 10^{-3}, \dots$. Compare con la solución verdadera

$$x(t) = 1 + \varepsilon \ln \cosh((x - 0.745)/\varepsilon)$$

que tiene un ángulo en $t = 0.745$.

c. **Problema de Troesch:** $\begin{cases} x'' = \mu \operatorname{senh}(\mu x) \\ x(0) = 0, x(1) = 1 \end{cases}$ usando $\mu = 50$.

d. **Problema de Bratu:** $\begin{cases} x'' + \lambda e^x = 0 \\ x(0) = 0, x(1) = 0 \end{cases}$ usando $\lambda = 3.55$.

Si hacemos $\lambda = 3.51383 \dots$, hay dos soluciones cuando $\lambda < \lambda^*$, una solución cuando $\lambda = \lambda^*$ y no hay soluciones cuando $\lambda > \lambda^*$.

e. $\begin{cases} \varepsilon x'' + tx' = 0 \\ x(-1) = 0, x(1) = 2 \end{cases}$ usando $\varepsilon = 10^{-8}$.

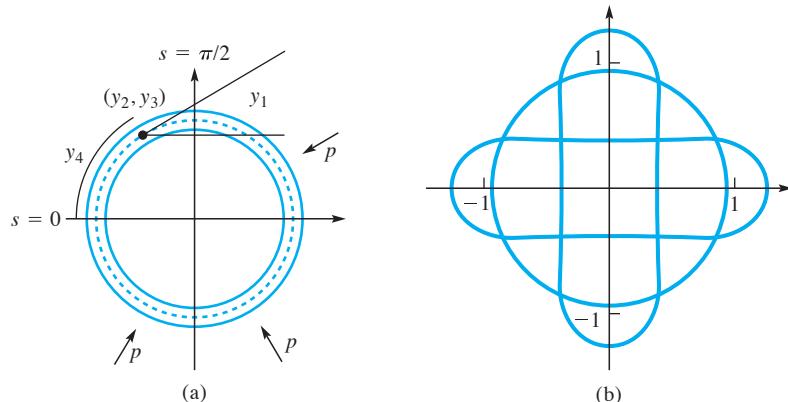
Compare con la verdadera solución $x(t) = 1 + \operatorname{erf}(t/\sqrt{2\varepsilon})/\operatorname{erf}(1/\sqrt{2\varepsilon})$.

Cash [2003] utiliza estos y otros problemas de prueba en su investigación. Para obtener más información sobre ellos, vaya a www.ma.ic.ac.uk/~jcash/

8. **(Proyecto del pandeo de un anillo circular)** Un modelo de un anillo circular con compresibilidad c con presión hidrostática p desde todas las direcciones está dada por el siguiente problema con valor en la frontera que implica un sistema de siete ecuaciones diferenciales:

$$\begin{aligned} y_1' &= -1 - cy_5 + (c+1)y_7, & y_1(0) &= \frac{\pi}{2}, & y_1\left(\frac{\pi}{2}\right) &= 0 \\ y_2' &= [1 + c(y_5 - y_7)] \cos y_1, & & & y_2\left(\frac{\pi}{2}\right) &= 0 \\ y_3' &= [1 + c(y_5 - y_7)] \operatorname{sen} y_1, & y_3(0) &= 0 & & \\ y_4' &= 1 + c(y_5 - y_7), & & & y_4(0) &= 0 \\ y_5' &= y_6[-1 - cy_5 + (c+1)y_7], & & & & \\ y_6' &= y_5y_7 - [1 + c(y_5 - y_7)](y_5 + p), & y_6(0) &= 0, & y_6\left(\frac{\pi}{2}\right) &= 0 \\ y_7' &= [1 + c(y_5 - y_7)]y_6 & & & & \end{aligned}$$

Varias simplificaciones son útiles en el estudio del pandeo o el colapso del anillo circular tal como considerar sólo un cuarto de círculo, por simetría [el dibujo (a) que se muestra a continuación]. A medida que aumenta la presión, el radio del círculo disminuye y puede producirse una bifurcación o un cambio de estado [esquema (b) se muestra a continuación]. El método de disparo, junto con métodos numéricos muchos más avanzados se pueden utilizar para resolver este problema. Explore algunos de ellos. Véase Huddleston [2000] y Sauer [2006] para más detalles.



Ecuaciones diferenciales parciales

En la teoría de la elasticidad, se demuestra que la tensión en una viga cilíndrica bajo torsión se puede deducir de una función $u(x, y)$ que satisface la ecuación de Poisson

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 2 = 0$$

En el caso de una viga cuya sección transversal es el cuadrado definido por $|x| \leq 1, |y| \leq 1$, la función u debe satisfacer la ecuación de Poisson en el *interior* del cuadrado y debe ser cero en cada punto del *perímetro* de éste. Usando los métodos de este capítulo podemos construir una tabla de valores aproximados de $u(x, y)$.

15.1 Problemas parabólicos

Muchos fenómenos físicos se pueden modelar matemáticamente con ecuaciones diferenciales. Cuando la función que se está estudiando implica dos o más variables independientes, la ecuación diferencial es generalmente una ecuación diferencial *parcial*. Puesto que las funciones de varias variables son intrínsecamente más complicadas que las de una variable, las ecuaciones diferenciales parciales puede dar lugar a algunos de los más difíciles problemas numéricos. De hecho, su solución numérica es un tipo de cálculo científico en el que los recursos de los sistemas de computación más rápidos y más caros se convierten fácilmente en insuficientes. Más adelante veremos por qué esto es así.

Algunas ecuaciones diferenciales parciales de problemas de aplicación

Aquí se enumeran algunas ecuaciones diferenciales parciales importantes y los fenómenos físicos que gobiernan:

- La **ecuación de onda** en tres variables espaciales (x, y, z) y el tiempo t es

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$$

La función u representa el desplazamiento en el tiempo t de una partícula cuya posición en reposo es (x, y, z) . Con condiciones en la frontera adecuadas, esta ecuación gobierna las vibraciones de un cuerpo elástico en tres dimensiones.

La **ecuación del calor** es

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$$

La función u representa la temperatura en el tiempo t de un cuerpo físico en el punto que tiene coordenadas (x, y, z) .

- La **ecuación de Laplace** es

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0$$

Regula la distribución de estado estable de calor en un cuerpo o la distribución de estado estable de la carga eléctrica en un cuerpo. La ecuación de Laplace también rige los potenciales gravitacionales, eléctricos, magnéticos y los potenciales de velocidad en flujos irrotacionales de fluidos incompresibles. La forma de la ecuación de Laplace dada anteriormente se aplica en coordenadas rectangulares. En coordenadas cilíndricas y esféricas, toma estas formas respectivas:

$$\begin{aligned} \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \phi^2} + \frac{\partial^2 u}{\partial z^2} &= 0 \\ \frac{1}{r} \frac{\partial^2}{\partial r^2} (ru) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 u}{\partial \phi^2} &= 0 \end{aligned}$$

- La **ecuación biarmónica** es

$$\frac{\partial^4 u}{\partial x^4} + 2 \frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = 0$$

Se presenta en el estudio de la tensión elástica y de su solución en tensiones normales y de corte se pueden deducir para un cuerpo elástico.

- Las **ecuaciones de Navier-Stokes** son

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} &= \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} &= \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \end{aligned}$$

Aquí, u y v son las componentes del vector de velocidad en un flujo de fluidos. La función p es la presión y el fluido se supone que es incompresible, pero viscoso.

En tres dimensiones, los siguientes operadores son útiles para escribir un gran número de ecuaciones diferenciales parciales estándares

$$\begin{aligned} \nabla &= \frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} \\ \nabla^2 &= \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (\text{Operador de Laplace}) \end{aligned}$$

Por ejemplo, tenemos

Ecuación del calor	$\frac{1}{k} \frac{\partial u}{\partial t} = \nabla^2 u$
Ecuación de difusión	$\frac{\partial u}{\partial t} = \nabla \cdot (d \nabla u) + \rho$
Ecuación de onda	$\frac{1}{v^2} \frac{\partial^2 u}{\partial t^2} = \nabla^2 u$
Ecuación de Laplace	$\nabla^2 u = 0$
Ecuación de Poisson	$\nabla^2 u = -4\pi\rho$
Ecuación de Helmholtz	$\nabla^2 u = -k^2 u$

La **ecuación de difusión** con la constante de difusión d tiene la misma estructura que la ecuación del calor porque la transferencia de calor es un proceso de difusión. Algunos autores utilizan una notación alternativa, como $\Delta u = \text{curl}(\text{grad}(u)) = \nabla^2 u$.

También se podrían estudiar otros ejemplos de la mecánica cuántica, el electromagnetismo, la hidrodinámica, la elasticidad, etcétera, pero las cinco ecuaciones diferenciales parciales mostradas ya exhiben una gran diversidad. La ecuación de Navier-Stokes, en particular, ilustra un problema muy complicado: un par de ecuaciones diferenciales parciales no lineales simultáneas.

Para especificar una solución única a una ecuación diferencial parcial, las condiciones adicionales se deben imponer a la función solución. Normalmente, estas condiciones se dan en forma de valores en la frontera que se dan en todo o en parte del perímetro de la región en la que se pide la solución. La naturaleza de la frontera y de los valores en ella suelen ser los factores determinantes en la creación de un sistema numérico apropiado para la obtención de la solución aproximada.

Matlab incluye una caja de herramientas de EDP para ecuaciones diferenciales parciales. Contiene muchas instrucciones para tareas como la que describe el dominio de una ecuación, la generación de mallas, cálculo de soluciones numéricas y la graficación. Dentro de Matlab, la instrucción pdetool llama a una interfaz gráfica de usuario (GUI) que es un ambiente gráfico autocontenido para la resolución de ecuaciones diferenciales parciales. Se traza el dominio y se indica la frontera, se llenan los menús con el problema y las especificaciones de la frontera y se seleccionan los botones para resolver el problema y graficar los resultados. Aunque esta interfaz puede proporcionar un entorno de trabajo cómodo, hay situaciones en las que se necesitan renglones con instrucciones de funciones para obtener flexibilidad adicional. Una serie de demostraciones y archivos de ayuda es útil para encontrar el camino. Por ejemplo, este software puede manejar EDP de los siguientes tipos de

EDP parabólica	$b \frac{\partial u}{\partial t} - \nabla \cdot (c \nabla u) + au = f$
EDP hiperbólica	$b \frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c \nabla u) + au = f$
EDP elíptica	$-\nabla \cdot (c \nabla u) + au = f$

para x y y en el dominio bidimensional Ω del problema. En las fronteras del dominio, se pueden manejar las siguientes condiciones en la frontera:

Dirichlet	$hu = r$
Generalizada de Neumann	$\vec{n} \cdot (c \nabla u) + qu = g$
Mixta	Combinación de Dirichlet/Neumann

Aquí, $\hat{n} = du / dv$ es la derivada de la norma exterior de longitud unitaria. Mientras que la EDP se puede introducir a través de un cuadro de diálogo, tanto las condiciones de frontera como los coeficientes de la EDP a, c, d se pueden introducir en una variedad de maneras. Se puede construir la geometría del dominio mediante la elaboración de objetos sólidos (círculo, polígono, rectángulo y elipse) que se pueden superponer, mover y rotar.

Problema modelo de la ecuación de calor

En esta sección se considera un problema modelo de modesto alcance para introducir algunas de las ideas esenciales. Por motivos técnicos, se dice que el problema es de tipo **parabólico**. En él tenemos la ecuación del calor en una variable espacial acompañado de condiciones en la frontera adecuadas a un fenómeno físico determinado:

$$\begin{cases} \frac{\partial^2}{\partial x^2} u(x, t) = \frac{\partial}{\partial t} u(x, t) \\ u(0, t) = u(1, t) = 0 \\ u(x, 0) = \sin \pi x \end{cases} \quad (1)$$

Estas ecuaciones gobiernan la temperatura $u(x, t)$ en una varilla delgada de longitud 1, cuando los extremos se mantienen a temperatura 0, bajo el supuesto de que la temperatura inicial de la barra está dada por la función $\sin \pi x$ (figura 15.1). En el plano xt , la región en la que se busca la solución se describe por las desigualdades $0 \leq x \leq 1$ y $t \geq 0$. En la frontera de esta región (sombreada en la figura 15.2), los valores de u han sido dados.

FIGURA 15.1
Varilla caliente

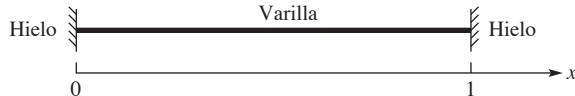
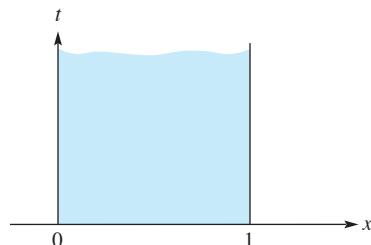


FIGURA 15.2
Ecuación del calor:
plano xt



Método de diferencias finitas

Un enfoque principal para la solución numérica de este problema es el **método de diferencias finitas**. Se procede sustituyendo las derivadas de la ecuación por diferencias finitas. Dos fórmulas

de la sección 4.3 son útiles en este contexto:

$$\begin{aligned}f'(x) &\approx \frac{1}{h}[f(x+h) - f(x)] \\f''(x) &\approx \frac{1}{h^2}[f(x+h) - 2f(x) + f(x-h)]\end{aligned}$$

Si las fórmulas se utilizan en la ecuación diferencial (1), posiblemente con diferentes longitudes de paso h y k , el resultado es

$$\frac{1}{h^2}[u(x+h,t) - 2u(x,t) + u(x-h,t)] = \frac{1}{k}[u(x,t+k) - u(x,t)] \quad (2)$$

Esta ecuación se interpreta ahora como una forma de avanzar en la solución paso a paso en la variable t . Es decir, si se conoce $u(x,t)$ para $0 \leq x \leq 1$ y $0 \leq t \leq t_0$, entonces la ecuación (2) nos permite evaluar la solución para $t = t_0 + k$.

La ecuación (2) se puede reescribir en la forma

$$u(x,t+k) = \sigma u(x+h,t) + (1-2\sigma)u(x,t) + \sigma u(x-h,t) \quad (3)$$

donde

$$\sigma = \frac{k}{h^2}$$

En la figura 15.3 se presenta un dibujo que muestra la ubicación de los cuatro puntos que participan en esta ecuación. Puesto que se conoce la solución en la frontera de la región, se puede calcular una solución aproximada dentro de la región utilizando sistemáticamente la ecuación (3). Esta es, por supuesto, una solución *aproximada*, porque la ecuación (2) es sólo una diferencia finita análoga a la ecuación (1).

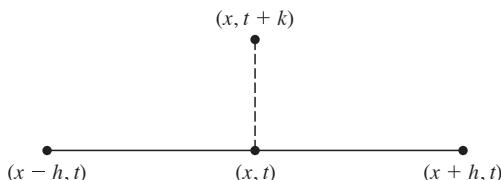


FIGURA 15.3
Ecuación del calor: plantilla explícita

Para obtener una solución aproximada en una computadora, se seleccionan los valores h y k y se usa la ecuación (3). Un análisis de este procedimiento, que está fuera del alcance de este libro, muestra que para la *estabilidad* del cálculo, el coeficiente $1 - 2\sigma$ de la ecuación (3) debe ser no negativo (si esta condición no se cumple, los errores cometidos en un solo paso probablemente serán ampliados en los siguientes pasos, en última instancia, estropeando la solución). Véase Kincaid y Cheney [2002] o a Forsythe y Wasow [1960] para un análisis de estabilidad. Con este algoritmo, podemos continuar indefinidamente la solución de la variable t porque los cálculos sólo implican los valores anteriores de t . Este es un ejemplo de un **problema de marcaje** o **método de marcaje**.

Seudocódigo para el método explícito

Para mayor simplicidad, se selecciona $h = 0.1$ y $k = 0.005$. El coeficiente σ es ahora 0.5. Esta opción hace que el coeficiente $1 - 2\sigma$ sea igual a cero. Nuestro seudocódigo imprime primero $u(ih, 0)$ para $0 \leq i \leq 10$ son valores en la frontera conocidos. Después calcula e imprime $u(ih, k)$ para $0 \leq i \leq 10$ utilizando la ecuación (3) y los valores en la frontera $u(0, t) = u(1, t) = 0$. Este procedimiento continúa hasta que t alcanza el valor 0.1. Los únicos arreglos subindizados (u_i) y (v_i) se utilizan para almacenar los valores de la solución aproximada en t y $t + k$, respectivamente. Puesto que la solución analítica del problema es $u(x, t) = e^{-\pi^2 t} \sin(\pi x)$ (véase el problema 15.1.3), el error se puede imprimir en cada paso.

El procedimiento descrito es un ejemplo de un **método explícito**. Los valores aproximados de $u(x, t + k)$ se calculan de forma explícita en términos de $u(x, t)$. No sólo esta situación es atípica, sino también en este problema el procedimiento es bastante lento debido a las consideraciones de estabilidad que nos obligan a seleccionar

$$k \leq \frac{1}{2}h^2$$

Puesto que h debe ser más bien pequeño para representar con precisión la derivada mediante la fórmula de diferencias finitas, la k correspondiente debe ser extremadamente pequeña. Valores tales como $h = 0.1$ y $k = 0.005$ son representativos, como lo son $h = 0.01$ y $k = 0.00005$. Con estos valores pequeños de k , se necesita una cantidad excesiva de cálculo para hacer que avance mucho la variable t .

```

program Parabólica 1
integer i, j; real array (ui)0:n, (vi)0:n
integer n ← 10, m ← 20
real h ← 0.1, k ← 0.005
real u0 ← 0, v0 ← 0, un ← 0, vn ← 0
for i = 1 to n – 1 do
    ui ← sin(π i h)
end for
output (ui)
for j = 1 to m do
    for i = 1 to n – 1 do
        vi ← (ui–1 + ui+1)/2
    end for
    output (vi)
    t ← jk
    for i = 1 to n – 1 do
        ui ← e–π2t sin(π i h) – vi
    end for
    output (ui)
    for i = 1 to n – 1 do
        ui ← vi
    end for
end for
end program Parabólica 1

```

Método de Crank-Nicolson

Un procedimiento alternativo del tipo implícito lleva el nombre de sus inventores, John Crank y Phyllis Nicolson, y se basa en una simple variante de la ecuación (2):

$$\frac{1}{h^2}[u(x+h,t) - 2u(x,t) + u(x-h,t)] = \frac{1}{k}[u(x,t) - u(x,t-k)] \quad (4)$$

Si una solución numérica en los puntos de la malla $x = ih$, $t = jk$ se ha obtenido hasta un cierto nivel en la variable t , la ecuación (4) gobierna los valores de u en el nivel t siguiente. Por lo tanto, la ecuación (4) se debe reescribir como

$$-u(x-h,t) + ru(x,t) - u(x+h,t) = su(x,t-k) \quad (5)$$

en la que

$$r = 2 + s \quad \text{y} \quad s = \frac{h^2}{k}$$

Las ubicaciones de los cuatro puntos en esta ecuación se muestran en la figura 15.4.

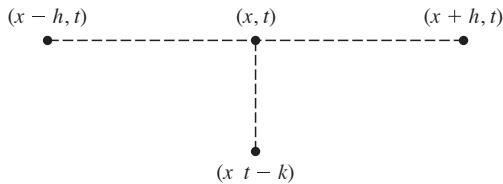


FIGURA 15.4
Método de
Crank-Nicolson:
plantilla
implícita

En el nivel t , u es desconocida, pero en el nivel $(t-k)$, u se conoce. Así podemos introducir incógnitas $u_i = u(ih, t)$ y las cantidades conocidas $b_i = su(ih, t-k)$ y escribir la ecuación (5) en forma de matriz:

$$\begin{bmatrix} r & -1 & & \\ -1 & r & -1 & \\ & -1 & r & -1 \\ & & \ddots & \ddots & \ddots \\ & & & -1 & r & -1 \\ & & & & -1 & r \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix} \quad (6)$$

La suposición de simplificación $u(0, t) = u(1, t) = 0$ se ha utilizado aquí. También, $h = 1/n$. El sistema de ecuaciones es tridiagonal y diagonalmente dominante porque $|r| = 2 + h^2/k > 2$. Por lo tanto, se puede resolver con el método eficaz de la sección 7.3.

Un argumento elemental muestra que este método es *estable*. Vamos a ver que si los valores iniciales $u(x, 0)$ se encuentran en un intervalo $[\alpha, \beta]$, entonces los valores posteriormente calculados utilizando la ecuación (5) también se encuentran en $[\alpha, \beta]$, lo que excluye cualquier crecimiento inestable. Como la solución se construye línea por línea de una manera uniforme, sólo tenemos que comprobar que los valores en la primera línea calculada, $u(x, k)$, se encuentran en $[\alpha, \beta]$. Sea j el índice de la más grande u_i que se produce en esta línea $t = k$. Entonces

$$-u_{j-1} + ru_j - u_{j+1} = b_j$$

Puesto que u_j es la más grande de las u , $u_{j-1} \leq u_j$ y $u_{j+1} \leq u_j$. Así,

$$ru_j = b_j + u_{j-1} + u_{j+1} \leq b_j + 2u_j$$

Como $r = 2 + s$ y $b_j = su(jh, 0)$, la desigualdad anterior conduce a la vez a

$$u_j \leq u(jh, 0) \leq \beta$$

Puesto que u_j es la más grande de las u_i , tenemos

$$u_i \leq \beta \quad \text{para toda } i$$

Del mismo modo,

$$u_i \geq \alpha \quad \text{para toda } i$$

así se establece nuestra afirmación.

Seudocódigo para el método de Crank-Nicolson

Un seudocódigo para realizar el método de Crank-Nicolson en el programa modelo se da a continuación. En este, $h = 0.1$, $k = h^2/2$ y la solución se continúa hasta que $t = 0.1$. El valor de r es 4 y $s = 2$. Es más fácil calcular e imprimir sólo los valores de u en puntos interiores de cada línea horizontal. En los puntos de frontera, tenemos $u(0, t) = u(1, t) = 0$. El programa llama al procedimiento *Tri* de la sección 7.3.

```

program Parabólica 2
integer i, j; real array (ci)1:n-1, (di)1:n-1, (ui)1:n-1, (vi)1:n-1
integer n ← 10, m ← 20
real h ← 0.1, k ← 0.005
real r, s, t
s ← h2/k
r ← 2 + s
for i = 1 to n - 1 do
    di ← r
    ci ← -1
    ui ← sin(π i h)
end for
output (ui)
for j = 1 to m do
    for i = 1 to n - 1 do
        di ← r
        vi ← s ui
    end for

```

```

call Tri(n − 1,(ci),(di),(ci),(vi),(vi))
output (vi)
t ← jk
for i = 1 to n − 1 do
    ui ←  $e^{-\pi^2 t} \sin(\pi i h) - v_i$ 
end for
output (ui)
for i = 1 to n − 1 do
    ui ← vi
end for
end for
end program Parabólica 2

```

Utilizamos los mismos valores para h y k en el seudocódigo de dos métodos (explícito y de Crank-Nicolson), por lo que se puede hacer una comparación justa de los productos. Debido a que el método de Crank-Nicolson es estable, se podría haber utilizado una k mucho mayor.

Versión alternativa del método de Crank-Nicolson

Otra versión del método de Crank-Nicolson se obtiene como siguen. Las diferencias centrales en $(x, t - \frac{1}{2}k)$ en la ecuación (4) producen

$$\begin{aligned} & \frac{1}{h^2} \left[u\left(x + h, t - \frac{1}{2}k\right) - 2u\left(x, t - \frac{1}{2}k\right) + u\left(x - h, t - \frac{1}{2}k\right) \right] \\ &= \frac{1}{k} [u(x, t) - u(x, t - k)] \end{aligned}$$

Puesto que los valores de u sólo se conocen en múltiplos enteros de k , términos tales como $u(x, t - \frac{1}{2}k)$ se sustituyen por el promedio de los valores de u en los puntos de la cuadrícula adyacente, es decir,

$$u\left(x, t - \frac{1}{2}k\right) \approx \frac{1}{2}[u(x, t) + u(x, t - k)]$$

Así, tenemos

$$\begin{aligned} & \frac{1}{2h^2} [u(x + h, t) - 2u(x, t) + u(x - h, t) + u(x + h, t - k) \\ & \quad - 2u(x, t - k) + u(x - h, t - k)] = \frac{1}{k} [u(x, t) - u(x, t - k)] \end{aligned}$$

La forma de cálculo de esta ecuación es

$$\begin{aligned} & -u(x - h, t) + 2(1 + s)u(x, t) - u(x + h, t) \\ &= u(x - h, t - k) + 2(s - 1)u(x, t - k) + u(x + h, t - k) \end{aligned} \tag{7}$$

donde

$$s = \frac{h^2}{k} \equiv \frac{1}{\sigma}$$

Los seis puntos en esta ecuación se muestran en la figura 15.5. Esto conduce a un sistema tridiagonal de la forma (6), con $r = 2(1 + s)$ y

$$b_i = u((i - 1)h, t - k) + 2(s - 1)u(ih, t - k) + u((i + 1)h, t - k)$$

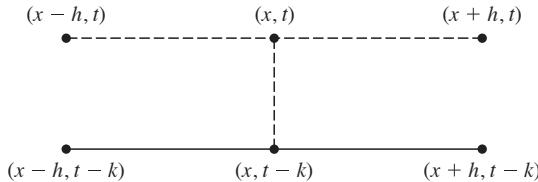


FIGURA 15.5
Método de
Crank-Nicolson:
plantilla
alternativa

Estabilidad

En el corazón del método explícito está la ecuación (3), que muestra cómo los valores de u para $t + k$ dependen de los valores de u en el paso de tiempo anterior, t . Si introducimos los valores de u en la malla escribiendo $u_{ij} = (ih, jk)$, entonces podemos reunir todos los valores de un nivel t en un vector $\mathbf{v}^{(j)}$ de la siguiente manera:

$$\mathbf{v}^{(j)} = [u_{0j}, u_{1j}, u_{2j}, \dots, u_{nj}]^T$$

La ecuación (3) ahora se puede escribir en la forma

$$u_{i,j+1} = \sigma u_{i+1,j} + (1 - 2\sigma)u_{ij} + \sigma u_{i-1,j}$$

Esta ecuación muestra cómo $\mathbf{v}^{(j+1)}$ se obtiene de $\mathbf{v}^{(j)}$. Es simplemente

$$\mathbf{v}^{(j+1)} = \mathbf{A}\mathbf{v}^{(j)}$$

donde \mathbf{A} es la matriz cuyos elementos son

$$\begin{bmatrix} 1 - 2\sigma & \sigma & & & \\ \sigma & 1 - 2\sigma & \sigma & & \\ & \sigma & 1 - 2\sigma & \sigma & \\ & & \ddots & \ddots & \ddots \\ & & & \sigma & 1 - 2\sigma & \sigma \\ & & & & \sigma & 1 - 2\sigma \end{bmatrix}$$

Nuestras ecuaciones nos dicen que

$$\mathbf{v}^{(j)} = \mathbf{A}\mathbf{v}^{(j-1)} = \mathbf{A}^2\mathbf{v}^{(j-2)} = \mathbf{A}^3\mathbf{v}^{(j-3)} = \cdots = \mathbf{A}^j\mathbf{v}^{(0)}$$

A partir de consideraciones físicas, la temperatura en la barra debe acercarse a cero. Después de todo, el calor se pierde a través de los extremos de la varilla, que se mantienen a temperatura 0. Por lo tanto, $\mathbf{A}^j\mathbf{v}^{(0)}$ debe converger a 0 conforme $j \rightarrow \infty$.

En este momento, necesitamos un teorema de álgebra lineal que afirma (para cualquier matriz) que $\mathbf{A}^j\mathbf{v} \rightarrow 0$ para todos los vectores \mathbf{v} si y sólo si todos los valores propios de \mathbf{A} satisfacen $|\lambda_i| < 1$. Los valores propios de la matriz \mathbf{A} en el presente análisis se sabe que son

$$\lambda_i = 1 - 2\sigma(1 - \cos \theta_i) \quad \theta_i = \frac{i\pi}{n+1}$$

En nuestro problema, por lo tanto, se debe tener

$$-1 < 1 - 2\sigma(1 - \cos \theta_i) < 1$$

Esto conduce a $0 < \sigma \leq \frac{1}{2}$, porque θ_i puede estar arbitrariamente cerca de π . Esto a su vez conduce a la condición de tamaño de paso $k \leq \frac{1}{2}h^2$.

Sistemas de software matemático como Matlab, Maple o Mathematica contienen rutinas que resuelven ecuaciones diferenciales parciales. Por ejemplo, en Maple y Mathematica podemos invocar comandos para verificar la solución analítica general (véase el problema 15.1.3). En Matlab, hay un programa de ejemplo para resolver numéricamente la ecuación de calor modelo. En la figura 15.6, se resuelve la ecuación de calor, se genera un gráfico tridimensional de la superficie de la solución y se produce una gráfica de curvas de nivel en dos dimensiones, que se muestra en color para indicar las diferentes curvas de nivel.

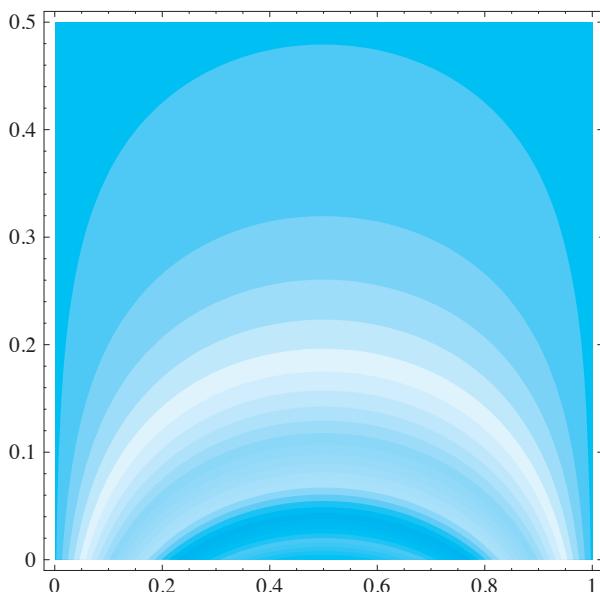
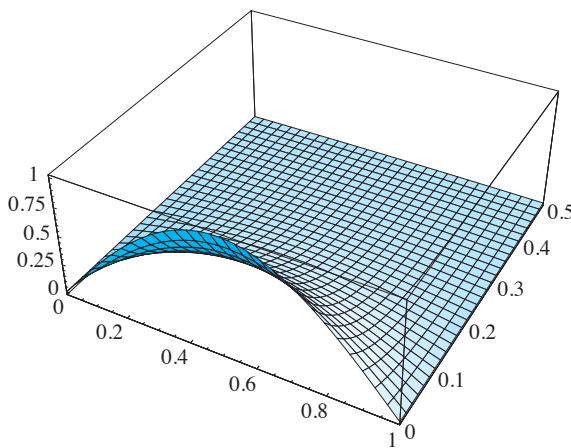


FIGURA 15.6
Ecuación de calor:
(a) Superficie solución;
(b) Gráfica de curvas de nivel

La caja de herramientas de EDP dentro de Matlab produce soluciones de las ecuaciones diferenciales parciales usando la formulación del elemento finito del problema EDP escalar. (Véase la sección 15.3 para el análisis adicional del método de elementos finitos.) Esta biblioteca de software contiene una interfaz gráfica de usuario con herramientas gráficas para la descripción de dominios, generación de mallas triangulares en ellos, discretizar las EDP sobre la malla, la construcción de sistemas de ecuaciones, la obtención numérica de aproximaciones para su solución y visualizar los resultados. En particular, Matlab tiene la función `parabolic` para resolver las EDP parabólicas. Como se encuentra en la documentación de Matlab, se puede resolver la ecuación de calor en dos dimensiones

$$\frac{\partial u}{\partial t} = \nabla^2 u$$

en el cuadrado $-1 \leq x, y \leq 1$. Hay condiciones de frontera de Dirichlet $u = 0$ y las condiciones iniciales discontinuas $u(0) = 1$ en el círculo $x^2 + y^2 < \frac{2}{5}$ y $u(0) = 0$ en caso contrario. Una demostración de Matlab continúa con una película de las curvas solución.

Resumen

(1) Consideremos un problema modelo que implica la siguiente ecuación diferencial parcial parabólica

$$\frac{\partial^2}{\partial x^2} u(x, t) = \frac{\partial}{\partial t} u(x, t)$$

Usando diferencias finitas con tamaño de paso h en la dirección x y k en la dirección t , se obtiene

$$\frac{1}{h^2} [u(x + h, t) - 2u(x, t) + u(x - h, t)] = \frac{1}{k} [u(x, t + k) - u(x, t)]$$

La forma de cálculo es

$$u(x, t + k) = \sigma u(x + h, t) + (1 - 2\sigma)u(x, t) + \sigma u(x - h, t)$$

donde $\sigma = k/h^2$. Un enfoque alternativo es el **método de Crank-Nicolson**, basado en otras diferencias finitas para el miembro derecho:

$$\frac{1}{h^2} [u(x + h, t) - 2u(x, t) + u(x - h, t)] = \frac{1}{k} [u(x, t) - u(x, t - k)]$$

Su forma de cálculo es

$$-u(x - h, t) + ru(x, t) - u(x + h, t) = su(x, t - k)$$

donde $r = 2 + s$ y $s = h^2/k$. Sin embargo, otra variante del método de Crank-Nicolson se basa en estas diferencias finitas:

$$\begin{aligned} \frac{1}{h^2} \left[u\left(x + h, t - \frac{1}{2}k\right) - 2u\left(x, t - \frac{1}{2}k\right) + u\left(x - h, t - \frac{1}{2}k\right) \right] \\ = \frac{1}{k} [u(x, t) - u(x, t - k)] \end{aligned}$$

Entonces, usando

$$u\left(x, t - \frac{1}{2}k\right) \approx \frac{1}{2} [u(x, t) + u(x, t - k)]$$

la forma de cálculo es

$$\begin{aligned} -u(x-h, t) + 2(1+s)u(x, t) - u(x+h, t) \\ = u(x-h, t-k) + 2(s-1)u(x, t-k) + u(x+h, t-k) \end{aligned}$$

donde $s = h^2/k$. Esto da como resultado un sistema tridiagonal ecuaciones por resolver.

Problemas 15.1

- 1.** Una ecuación diferencial lineal de segundo orden con dos variables tiene la forma

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + \dots = 0$$

Aquí, A , B y C son funciones de x y y , y los términos no escritos son de orden inferior. Se dice que la ecuación es **elíptica**, **parabólica** o **hiperbólica** en un punto (x, y) , dependiendo de si $B^2 - 4AC$ es negativo, cero o positivo, respectivamente. Clasifique cada una de estas ecuaciones de esta manera:

- a.** $u_{xx} + u_{yy} + u_x + \operatorname{sen} x u_y - u = x^2 + y^2$
- b.** $u_{xx} - u_{yy} + 2u_x + 2u_y + e^x u = x - y$
- c.** $u_{xx} = u_y + u - u_x + y$
- d.** $u_{xy} = u - u_x - u_y$
- e.** $3u_{xx} + u_{xy} + u_{yy} = e^{xy}$
- f.** $e^x u_{xx} + \cos y u_{xy} - u_{yy} = 0$
- g.** $u_{xx} + 2u_{xy} + u_{yy} = 0$
- h.** $xu_{xx} + yu_{xy} + u_{yy} = 0$

- 2.** Deduzca la forma bidimensional de la ecuación de Laplace en coordenadas polares.

- 3.** Demuestre que la función

$$u(x, t) = \sum_{n=1}^N c_n e^{-(n\pi)^2 t} \operatorname{sen} n\pi x$$

es una solución del problema de conducción de calor $u_{xx} = u_t$ y satisface la condición en la frontera

$$u(0, t) = u(1, t) = 0 \quad u(x, 0) = \sum_{n=1}^N c_n \operatorname{sen} n\pi x \quad \text{para toda } N \geq 1$$

- 4.** Consulte el problema modelo resuelto numéricamente en esta sección y demuestre que si no hay redondeo, los valores obtenidos con la solución aproximada utilizando la ecuación (3) se encuentran en el intervalo $[0, 1]$. (Suponga $1 \geq 2k/h^2$.)
- 5.** Encuentre una solución de la ecuación (3) que tenga la forma $u(x, t) = a' \operatorname{sen} \pi x$, donde a es una constante.
- 6.** Usando la ecuación (5), ¿cómo se debe modificar el sistema lineal (6) para $u(0, t) = c_0$ y $u(1, t) = c_n$ con $c_0 \neq 0$, $c_n \neq 0$? ¿Y cuando se utiliza la ecuación (7)?

- 7.** Describa en detalle cómo la ecuación (1) con condiciones en la frontera $u(0, t) = q(t)$, $u(1, t) = g(t)$ y $u(x, 0) = f(x)$ puede ser resuelta numéricamente usando el sistema (6). Aquí, q , g y f son funciones conocidas.
- 8.** ¿Qué ecuación de diferencias finitas debe ser un sustituto adecuado para la ecuación $\partial^2 u / \partial x^2 = \partial u / \partial t + \partial u / \partial x$ en el trabajo numérico?
- 9.** Considere la ecuación diferencial parcial $\partial u / \partial x + \partial u / \partial t = 0$ con $u = u(x, t)$ en la región $[0, 1] \times [0, \infty]$, sujeta a las condiciones en la frontera $u(0, t) = 0$ y $u(x, 0)$ dadas. Para t fija, discretizamos sólo el primer término utilizando $(u_{i+1} - u_{i-1})/(2h)$ para $i = 1, 2, \dots, n-1$ y $(u_n - u_{n-1})/h$, donde $h = 1/n$. Aquí, $u_i = u(x_i, t)$ y $x = ih$ con t fija. De esta manera, el problema original puede considerarse un problema de primer orden con valor inicial

$$\frac{dy}{dx} + \frac{1}{2h} A y = 0$$

donde

$$y = [u_1, u_2, \dots, u_n]^T \quad \frac{dy}{dx} = [u'_1, u'_2, \dots, u'_n]^T \quad u'_i = \frac{\partial u_i}{\partial t}$$

Determine la matriz A de $n \times n$.

- 10.** Consulte la explicación de la estabilidad del procedimiento de Crank-Nicolson y establezca la desigualdad $u_i \leq a$.
- 11.** ¿Qué sucede con el sistema (6) cuando $k = h^2$?

- 12.** (Opción múltiple) En la solución de la ecuación del calor $u_{xx} = u_t$ en el dominio $t \geq 1$ y $0 \leq x \leq 1$ se puede utilizar el **método explícito**. Supongamos que la solución aproximada en una recta horizontal es un vector V_j . Entonces, todo el proceso resulta estar descrito por

$$V_{j+1} = AV_j$$

en donde A es una matriz tridiagonal, que tiene con $1 - 2\sigma$ en su diagonal y σ en las posiciones superdiagonal y subdiagonal. Aquí $\sigma = k/h^2$, donde k es el paso del tiempo y h es el paso en x . Para la estabilidad en la solución numérica, ¿qué debemos exigir?

- a. $\sigma = \frac{1}{2}$ b. Todos los valores propios de A satisfacen $|\lambda| < 1$. c. $k \geq h^2/2$
d. $h = 0.01$ and $k = 5 \times 10^{-3}$ e. Ninguna de estas

- 13.** (Continuación) El **método totalmente implícito** para resolver el problema de la conducción del calor requiere en cada paso la solución de la ecuación de

$$AV_{j-1} = V_j$$

Aquí, A no es la misma que en el problema anterior, pero es similar: tiene $1 + 2\sigma$ en la diagonal y $-\sigma$ en la subdiagonal y superdiagonal. ¿Qué sabemos acerca de los valores propios de esta matriz? *Sugerencia:* esta pregunta se refiere a los valores propios de A , no de A^{-1} .

- a. Son todos negativos. b. Todos están en el intervalo abierto $(0, 1)$.
c. Son más grandes que 1. d. Están en el intervalo $(-1, 0)$.
e. Ninguno de estos.

Problemas de cómputo 15.1

- Resuelva el mismo problema de conducción de calor como en el libro, excepto que use $h = 2^{-4}$, $k = 2^{-10}$ y $u(x, 0) = x(1 - x)$. Realice la solución hasta $t = 0.0125$.
- Modifique el código de Crank-Nicolson en el libro de modo que utilice el esquema alternativo (7). Compare los dos programas en los mismos problemas con el mismo espaciado.
- Codifique de nuevo y pruebe el seudocódigo de esta sección usando un lenguaje de computadora que soporte operaciones vectoriales.
- Ejecute el código de Crank-Nicolson con diferentes opciones de h y k , en particular, haciendo k mucho mayor. Intente $k = h$, por ejemplo.
- Trate de aprovechar cualquier instrucción o procedimiento especial en software matemático como Matlab, Maple, o Mathematica para resolver el ejemplo numérico (1).
- (Continuación) Utilice la capacidad de manejo simbólico de Maple o Mathematica para verificar la solución analítica general de (1). *Sugerencia:* véase el problema 15.1.3.

15.2 Problemas hiperbólicos

Problema modelo de la ecuación de onda

La **ecuación de onda** con una variable de espacio

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \quad (1)$$

gobierna la vibración de una cuerda (vibración transversal en un plano) o la vibración de una varilla (vibración longitudinal). Este es un ejemplo de una ecuación diferencial lineal de segundo orden de tipo hiperbólico. Si la ecuación (1) se utiliza para modelar la cuerda que vibra, entonces $u(x, t)$ representa la deflexión en el tiempo t de un punto de la cuerda cuya coordenada es x cuando la cuerda está en reposo.

Para plantear un problema modelo definido, se supone que los puntos de la cuerda tienen coordenadas x en el intervalo $0 \leq x \leq 1$ (figura 15.7). Supongamos que en el tiempo $t = 0$, las deflexiones satisfacen las ecuaciones $u(x, 0) = f(x)$ y $u_t(x, 0) = 0$. Supongamos también que se mantienen fijos los extremos de la cuerda. Entonces $u(0, t) = u(1, t) = 0$. Un problema con valor en la frontera totalmente definido,

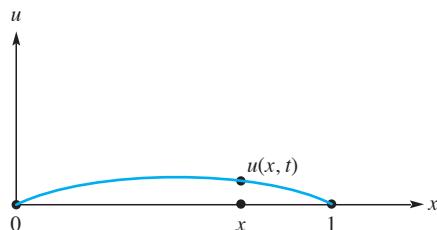


FIGURA 15.7
Cuerda vibrante

entonces, es

$$\begin{cases} u_{tt} - u_{xx} = 0 \\ u(x, 0) = f(x) \\ u_t(x, 0) = 0 \\ u(0, t) = u(1, t) = 0 \end{cases} \quad (2)$$

La región en el plano xt donde se solicita una solución es la tira semiinfinita definida por las desigualdades $0 \leq x \leq 1$ y $t \geq 0$. Como en el problema de conducción del calor de la sección 15.1, los valores de la función desconocida se dan en la frontera de la región mostrada (figura 15.8).

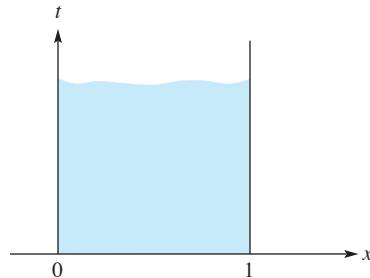


FIGURA 15.8
Ecuación de
onda: plano xt

Solución analítica

El problema modelo (2) es tan simple que puede ser resuelto inmediatamente. De hecho, la solución es

$$u(x, t) = \frac{1}{2}[f(x + t) + f(x - t)] \quad (3)$$

siempre que f tenga dos derivadas y se haya extendido a toda la recta real definiendo

$$f(-x) = -f(x) \quad \text{y} \quad f(x + 2) = f(x)$$

Para comprobar que la ecuación (3) es una solución, se calculan las derivadas utilizando la regla de la cadena:

$$\begin{aligned} u_x &= \frac{1}{2}[f'(x + t) + f'(x - t)] & u_t &= \frac{1}{2}[f'(x + t) - f'(x - t)] \\ u_{xx} &= \frac{1}{2}[f''(x + t) + f''(x - t)] & u_{tt} &= \frac{1}{2}[f''(x + t) + f''(x - t)] \end{aligned}$$

Obviamente,

$$u_{tt} = u_{xx}$$

También,

$$u(x, 0) = f(x)$$

Además, tenemos

$$u_t(x, 0) = \frac{1}{2}[f'(x) - f'(x)] = 0$$

En la comprobación de condiciones en los extremos usamos las fórmulas con las que f se amplió:

$$\begin{aligned} u(0, t) &= \frac{1}{2}[f(t) + f(-t)] = 0 \\ u(1, t) &= \frac{1}{2}[f(1+t) + f(1-t)] \\ &= \frac{1}{2}[f(1+t) - f(t-1)] \\ &= \frac{1}{2}[f(1+t) - f(t-1+2)] = 0 \end{aligned}$$

La extensión de f de su dominio original a toda la recta real la cubre en una función **periódica impar** de periodo 2. **Impar** significa que

$$f(x) = -f(-x)$$

y la **periodicidad** se expresa por

$$f(x+2) = f(x)$$

para toda x . Para calcular $u(x, t)$, necesitamos conocer f con sólo dos puntos en el eje x , $x+t$ y $x-t$, como se muestra en la figura 15.9.

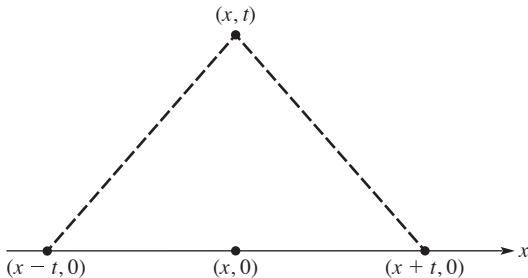


FIGURA 15.9
Ecuación de
onda: plantilla
de f

Solución numérica

El problema modelo se utiliza a continuación para ilustrar una vez más el principio de la solución numérica. La selección de tamaños de paso h y k para x y t , respectivamente, y utilizando las aproximaciones familiares de las derivadas, tenemos de la ecuación (1)

$$\begin{aligned} \frac{1}{h^2}[u(x+h, t) - 2u(x, t) + u(x-h, t)] \\ = \frac{1}{k^2}[u(x, t+k) - 2u(x, t) + u(x, t-k)] \end{aligned}$$

que se puede reordenar como

$$u(x, t+k) = \rho u(x+h, t) + 2(1-\rho)u(x, t) + \rho u(x-h, t) - u(x, t-k) \quad (4)$$

Aquí,

$$\rho = \frac{k^2}{h^2}$$

La figura 15.10 muestra el punto $(x, t + k)$ y los puntos cercanos que entran en la ecuación (4).

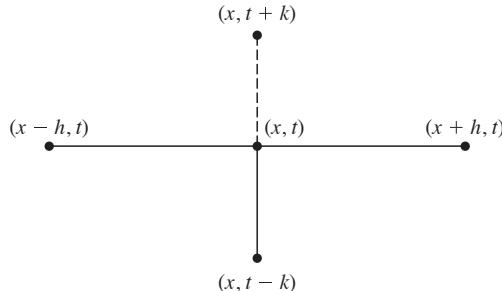


FIGURA 15.10
Ecuación de onda: plantilla explícita

Las condiciones en la frontera en el problema (2) se pueden escribir como

$$\begin{cases} u(x, 0) = f(x) \\ \frac{1}{k}[u(x, k) - u(x, 0)] = 0 \\ u(0, t) = u(1, t) = 0 \end{cases} \quad (5)$$

El problema definido por las ecuaciones (4) y (5) se puede resolver empezando en la recta $t = 0$, donde u se conoce, y después avanzando una recta a la vez con $t = k, t = 2k, t = 3k, \dots$. Observe que, debido a (5), nuestra solución aproximada satisface

$$u(x, k) = u(x, 0) = f(x) \quad (6)$$

El uso de la aproximación $\mathcal{O}(k)$ para u_t conduce a una baja exactitud en la solución calculada del problema (2). Supongamos que hay un renglón de puntos de la cuadrícula $(x, -k)$. Haciendo $t = 0$ en la ecuación (4), tenemos

$$u(x, k) = \rho u(x + h, 0) + 2(1 - \rho)u(x, 0) + \rho u(x - h, 0) - u(x, -k)$$

Ahora, la aproximación de diferencia central

$$\frac{1}{2k}[u(x, k) - u(x, -k)] = 0$$

para

$$u_t(x, 0) = 0$$

se puede utilizar para eliminar el punto de la cuadrícula ficticia $(x, -k)$. Así que en lugar de la ecuación (6), hacemos

$$u(x, k) = \frac{1}{2}\rho[f(x + h) + f(x - h)] + (1 - \rho)f(x) \quad (7)$$

ya que $u(x, 0) = f(x)$. Los valores de $u(x, nk)$, $n \geq 2$, ahora se puede calcular de la ecuación (4).

Seudocódigo

A continuación se presenta un seudocódigo que realiza este proceso numérico. Por simplicidad, se utilizan tres arreglos unidimensionales (u_i), (v_i) y (w_i): (u_i) representa la solución que se calcula en la nueva línea t ; (v_i) y (w_i) representan soluciones en las dos últimas líneas t .

```

program Hiperbólico
integer i, j; real t, x, ρ; real array (ui)0:n, (vi)0:n, (wi)0:n
integer n ← 10, m ← 20
real h ← 0.1, k ← 0.05
u0 ← 0; v0 ← 0; w0 ← 0; un ← 0; vn ← 0; wn ← 0
ρ ← (k/h)2
for i = 1 to n - 1 do
    x ← ih
    wi ← f(x)
    vi ← ½ρ[f(x - h) + f(x + h)] + (1 - ρ)f(x)
end for
for j = 2 to m do
    for i = 1 to n - 1 do
        ui ← ρ(vi+1 + vi-1) + 2(1 - ρ)vi - wi
    end for
    output j, (ui)
    for i = 1 to n - 1 do
        wi ← vi
        vi ← ui
        t ← jk
        x ← ih
        ui ← True_Solution(x, t) - vi
    end for
    output j, (ui)
end for
end program Hiperbólico

real function f(x) o
real x
f ← sin(πx)
end function f

real function True_Solution(x, t)
real t, x
True_Solution ← sin(πx) cos(πt)
end function True_Solution

```

Este seudocódigo requiere funciones de acompañamiento para calcular los valores de $f(x)$ y la verdadera solución. Elegimos $f(x) = \sin(\pi x)$ en nuestro ejemplo. Se supone que el intervalo x es $[0, 1]$, pero cuando se cambia h o n , el intervalo puede ser $[0, b]$; es decir, $nh = b$. La solución numérica se ha impreso en las líneas t que corresponden a $1k, 2k, \dots, mk$.

Los tratamientos más avanzados muestran que los cocientes

$$\rho = \frac{k^2}{h^2}$$

no deben exceder de 1 si la solución de las ecuaciones en diferencias finitas converge a una solución del problema diferencial cuando $k \rightarrow 0$ y $h \rightarrow 0$. Además, si $\rho > 1$, los errores de redondeo que se producen en una etapa del cálculo probablemente se verían agravados en las etapas posteriores y así se arruinaría la solución numérica.

En Matlab, la caja de herramientas de EDP tiene una función para producir la solución de problemas hiperbólicos mediante la formulación de elementos finitos del problema de la EDP escalar. Un ejemplo hallado en la documentación de Matlab encuentra la solución numérica del problema en dos dimensiones de la propagación de ondas

$$\frac{\partial^2 u}{\partial t^2} = \nabla^2 u$$

en el cuadrado $-1 \leq x, y \leq 1$ con condiciones en la frontera de Dirichlet en las fronteras izquierda y derecha, $u = 0$ para $x = \pm 1$ y valores cero de las derivadas normales en la parte superior e inferior. Además, existen condiciones en la frontera de Neumann $\partial u / \partial v = 0$ para $y = \pm 1$. Las condiciones iniciales $u(0) = \arctan(\cos(\frac{\pi}{2}x))$ y $du(0)/dt = 3 \sin(\pi x) \exp(\sin(\frac{\pi}{2}y))$ se eligen para evitar poner demasiada energía en los modos de vibración más alta.

Ecuación de advección

Nos centramos en la **ecuación de advección**

$$\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x}$$

En este caso, $u = u(x, t)$ y $c = c(x, t)$ en la que se puede considerar x como el espacio y t como el tiempo. La ecuación de advección es una ecuación diferencial parcial hiperbólica que rige el movimiento de un escalar conservado conforme se advecta en un campo de velocidad conocido. Por ejemplo, la ecuación de advección se aplica al transporte de sal disuelta en agua. Aun en una dimensión espacial y velocidad constante, el sistema sigue siendo difícil de resolver. Puesto que la ecuación de advección es difícil de resolver numéricamente, el interés general se centra en soluciones de choque discontinuas, que son sumamente difíciles para que los esquemas numéricos lo manejen.

Usando la aproximación por diferencias hacia adelante en el tiempo y la aproximación por diferencia central en el espacio, tenemos

$$\frac{1}{k}[u(x, t + k) - u(x, t)] = -c \frac{1}{2h} [u(x + h, t) - u(x - h, t)]$$

Esto da

$$u(x, t + k) = u(x, t) - \frac{1}{2}\sigma [u(x + h, t) - u(x - h, t)]$$

donde $\sigma = (k/h)c(x, t)$. Todas las soluciones numéricas crecen en magnitud para todos los pasos de tiempo k . Para toda $\sigma > 0$, este esquema es *inestable* mediante el análisis de la estabilidad de Fourier.

Método de Lax

En el esquema de diferencias centrales anterior, sustituimos el primer término del miembro derecho, $u(x, t)$, por $\frac{1}{2} [u(x, t - k) + u(x, t + k)]$. Entonces obtenemos

$$\begin{aligned} u(x, t + k) &= \frac{1}{2} [u(x, t - k) + u(x, t + k)] - \frac{1}{2}\sigma [u(x + h, t) - u(x - h, t)] \\ &= \frac{1}{2}(1 + \sigma)u(x - h, t) + \frac{1}{2}(1 - \sigma)u(x, t - k) \end{aligned}$$

Este es el **método de Lax** y este simple cambio hace el método condicionalmente estable.

Método contra el viento

Otra forma de obtener un método estable es usar la aproximación unilateral a u_x en la ecuación de advección, siempre que se tome este lado como la parte en dirección *contraria al viento*. Si $c > 0$, el transporte es a la derecha. Esto puede interpretarse como el viento de velocidad c que sopla de izquierda a derecha. Así, la dirección contra el viento es hacia la izquierda con $c > 0$ y hacia la derecha para $c < 0$. Así, la aproximación por diferencias contra el viento es

$$u_x(x, t) \approx \begin{cases} -c [u(x, t) - u(x - h, t)] / h & (c > 0) \\ -c [u(x + h, t) - u(x, t)] / h & (c < 0) \end{cases}$$

Entonces, el método contra el viento de la ecuación de advección es

$$u(x, t + k) = u(x, t) - \sigma \begin{cases} -c [u(x, t) - u(x - h, t)] / h & (c > 0) \\ -c [u(x + h, t) - u(x, t)] / h & (c < 0) \end{cases}$$

Método de Lax-Wendroff

El método de Lax-Wendroff es de segundo orden en espacio y en tiempo. La siguiente es una de las varias formas posibles de este método. Iniciamos con un desarrollo de la serie de Taylor sobre un paso de tiempo

$$u(x, t + k) = u(x, t) + ku_t(x, t) + \frac{1}{2}k^2u_{tt}(x, t) + \mathcal{O}(k^3)$$

Ahora use la ecuación de advección para sustituir las derivadas de tiempo en el miembro derecho por las derivadas espaciales:

$$\begin{aligned} u_t &= -cu_x \\ u_{tt} &= (-cu_x)_t \\ &= -c_tu_x - c(u_x)_t \\ &= -c_tu_x - c(u_t)_x \\ &= -c_tu_x + c(cu_x)_t \end{aligned}$$

Aquí, hacemos $c = c(x, t)$ y no hemos supuesto que c sea una constante. Sustituyendo u_t y u_{xx} obtenemos

$$u(x, t + k) = u(x, t) - cku_x + \frac{1}{2}k^2 [-c_tu_x + c(cu_x)_x] + \mathcal{O}(k^3)$$

donde todo en el lado derecho se evalúa en (x, t) . Si nos aproximamos a la derivada del espacio con diferencias de segundo orden, tendremos un esquema de segundo orden en espacio y en tiempo:

$$\begin{aligned} u(x, t + k) &\approx u(x, t) - ck \frac{1}{2h} [u(x + h, t) - u(x - h, t)] \\ &+ \frac{1}{2} k^2 \left[-c_t \frac{1}{2h} [u(x + h, t) - u(x - h, t)] + c(cu_x)_x \right] \end{aligned}$$

La dificultad con este esquema se plantea cuando c depende del espacio y se debe evaluar el último término en la expresión anterior. En el caso en que c sea una constante, se obtiene

$$\begin{aligned} c(cu_x)_x &= c^2 u_{xx} \\ &\approx \frac{1}{2h} [u(x + h, t) - 2u(x, t) + u(x - h, t)] \end{aligned}$$

El **esquema de Lax-Wendroff** se convierte en

$$\begin{aligned} u(x, t + k) &= u(x, t) - \frac{1}{2} \sigma [u(x + h, t) - u(x - h, t)] \\ &+ \frac{1}{2} c \sigma^2 [u(x + h, t) - 2u(x, t) + u(x - h, t)] \end{aligned}$$

donde $\sigma = c(k/h)$. Al igual que con el método de Lax, este método tiene disipación numérica (pérdida de amplitud); sin embargo, es relativamente débil.

Resumen

(1) Consideramos un problema modelo con la siguiente ecuación diferencial parcial hiperbólica:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2}$$

Usando diferencias finitas, lo aproximamos por

$$\begin{aligned} \frac{1}{h^2} [u(x + h, t) - 2u(x, t) + u(x - h, t)] \\ = \frac{1}{k^2} [u(x, t + k) - 2u(x, t) + u(x, t - k)] \end{aligned}$$

La forma de cálculo es

$$u(x, t + k) = \rho u(x + h, t) + 2(1 - \rho)u(x, t) + \rho u(x - h, t) - u(x, t - k)$$

donde $\rho = k^2/h^2 < 1$. En $t = 0$, usamos

$$u(x, k) = \frac{1}{2} \rho [f(x + h) + f(x - h)] + (1 - \rho) f(x)$$

Problemas 15.2

"1. ¿Cuál es la solución del problema con valores en la frontera

$$u_{tt} = u_{xx} \quad u(x, 0) = x(1 - x) \quad u_t(x, 0) = 0 \quad u(0, t) = u(1, t) = 0$$

en el punto donde $x = 0.3$ y $t = 4$?

"2. Demuestre que la función $u(x, t) = f(x + at) + g(x - at)$ satisface la ecuación de onda $u_{tt} = a^2 u_{xx}$.

"3. (Continuación) Utilizando la idea del problema anterior, resuelva este problema con valores en la frontera:

$$u_{tt} = u_{xx} \quad u(x, 0) = F(x) \quad u_t(x, 0) = G(x) \quad u(0, t) = u(1, t) = 0$$

4. Demuestre que el problema con valor de frontera

$$u_{tt} = u_{xx} \quad u(x, 0) = 2f(x) \quad u_t(x, 0) = 2g(x)$$

tiene la solución

$$u(x, t) = f(x + t) + f(x - t) + G(x + t) - G(x - t)$$

donde G es una primitiva (es decir, una integral indefinida) de g . Aquí, suponemos que $-\infty < x < \infty$ y $t \geq 0$.

5. (Continuación) Resuelva el problema anterior en un intervalo finito x , por ejemplo, $0 \leq x \leq -1$, agregando la condición en la frontera $u(0, t) = u(1, t) = 0$. En este caso, f y g se definen solamente para $0 \leq x \leq 1$.

Problemas de cómputo 15.2

"1. Dada $f(x)$ definida en $[0, 1]$, escriba y pruebe una función para calcular la f extendida que obedece las ecuaciones $f(-x) = -f(x)$ y $f(x+2) = f(x)$.

2. (Continuación) Escriba un programa para calcular la solución $u(x, t)$ en cualquier punto dado (x, t) para el problema con valor en la frontera de la ecuación (2).

3. Compare la exactitud de la solución calculada, utilizando la primera ecuación (6) y después la ecuación (7), en el programa de cómputo del libro.

4. Utilice el programa del libro para resolver el problema con valor en la frontera (2) con

$$f(x) = \frac{1}{4} \left(\frac{1}{2} - \left| x - \frac{1}{2} \right| \right) \quad h = \frac{1}{16} \quad k = \frac{1}{32}$$

5. Modifique el código del libro para resolver el problema con valor en la frontera (2) cuando $u(x, 0) = g(x)$. *Sugerencia:* las ecuaciones (5) y (7) serán ligeramente diferentes (un hecho que afecta solamente el ciclo inicial en el programa).

6. (Continuación) Utilice el programa que usted escribió para el problema de cómputo anterior para resolver el problema con valor en la frontera:

$$\begin{cases} u_{tt} = u_{xx} & (0 \leq x \leq 1, t \geq 0) \\ u(x, 0) = \sin \pi x \\ u_t(x, 0) = \frac{1}{4} \sin 2\pi x \\ u(0, t) = u(1, t) = 0 \end{cases}$$

7. Modifique el código en el libro para resolver el siguiente problema con valor en la frontera:

$$\begin{cases} u_{tt} = u_{xx} & (-1 \leq x \leq 1, t \geq 0) \\ u(x, 0) = |x| - 1 \\ u_t(x, 0) = 0 \\ u(-1, t) = u(1, t) = 0 \end{cases}$$

8. Modifique el código en el libro para evitar el almacenamiento de los arreglos (v_i) y (u_i).
9. Simplifique el código del libro para el caso especial en que $\rho = 1$. Compare la solución numérica en los puntos de la red para un mismo problema en el que $\rho = 1$ y $\rho \neq 1$.
10. Use software matemático como Matlab, Maple o Mathematica para resolver la ecuación de onda (2) y trace la gráfica tanto de la superficie solución como de las curvas de nivel.
11. Utilice las capacidades de manejo simbólico de Maple o Mathematica para comprobar que la ecuación (3) es la solución analítica general de la ecuación de onda.

15.3 Problemas elípticos

Una de las ecuaciones diferenciales parciales más importantes en física matemática y en ingeniería es la **ecuación de Laplace**, que en dos variables tiene la forma siguiente:

$$\nabla^2 u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Con esta se relaciona de cerca la **ecuación de Poisson**:

$$\nabla^2 u = g(x, y)$$

Estos son ejemplos de ecuaciones **elípticas**. (Consulte el problema 17.1.1 para la clasificación de ecuaciones.) Las condiciones en la frontera asociadas con las ecuaciones elípticas generalmente difieren de las ecuaciones parabólicas e hiperbólicas. Aquí se considera un problema modelo para ilustrar los procedimientos numéricos que con frecuencia se utilizan.

Problema modelo de la ecuación de Helmholtz

Suponga que una función $u = u(x, y)$ de dos variables es la solución a un problema físico determinado. Esta función no se conoce pero tiene algunas propiedades que, en teoría, la determinan en forma

única. Suponemos que en una región R en el plano xy ,

$$\begin{cases} \nabla^2 u + fu = g \\ u(x, y) \text{ conocida en la frontera de } R \end{cases} \quad (1)$$

Aquí, $f = f(x, y)$ y $g = g(x, y)$ son funciones continuas dadas definidas en R . Los valores en la frontera se podrían dar mediante una tercera función

$$u(x, y) = q(x, y)$$

en el perímetro de R . Cuando f es una constante, esta ecuación diferencial parcial se llama la **ecuación de Helmholtz**. Surge en la búsqueda de soluciones oscilatorias de las ecuaciones de onda.

Método de diferencias finitas

Como antes, encontramos una solución aproximada de este problema por el método de las diferencias finitas. El primer paso es seleccionar las fórmulas aproximadas para las derivadas en nuestro problema. En la situación actual, usamos la fórmula estándar

$$f''(x) \approx \frac{1}{h^2}[f(x+h) - 2f(x) + f(x-h)] \quad (2)$$

deducida en la sección 4.3. Si se utiliza en una función de dos variables, se obtiene la aproximación de la **fórmula de cinco puntos** de la ecuación de Laplace:

$$\nabla^2 u \approx \frac{1}{h^2}[u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h) - 4u(x, y)] \quad (3)$$

Esta fórmula implica los cinco puntos que se muestran en la figura 15.11.

El error local inherente a la fórmula de cinco puntos es

$$-\frac{h^2}{12} \left[\frac{\partial^4 u}{\partial x^4}(\xi, y) + \frac{\partial^4 u}{\partial y^4}(x, \eta) \right] \quad (4)$$

y por esta razón, se dice que la fórmula (3) proporciona una aproximación de orden $\mathcal{O}(h^2)$. En otras palabras, si las mallas se utilizan con un espaciado cada vez más pequeño, $h \rightarrow 0$, entonces el error que se cometió en la sustitución de $\nabla^2 u$ por su aproximación en diferencias finitas tiende a cero tan rápidamente como h^2 . La ecuación (3) se llama la fórmula de cinco puntos porque se trata de valores de u en (x, y) y en los cuatro puntos más cercanos de la cuadrícula.

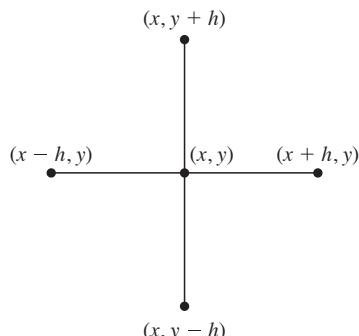


FIGURA 15.11

Ecuación
de Laplace:
plantilla de
cinco puntos

Cabe destacar que cuando la ecuación diferencial en (1) se sustituye por la análoga de diferencias finitas, hemos cambiado el problema. Aun si el problema análogo en diferencias finitas se ha resuelto con absoluta precisión, la solución es la de un problema que sólo *simula* el original. Esta simulación de un problema por otro se vuelve mejor y mejor conforme h se reduce a cero, pero el costo computacional aumentará inevitablemente.

Debemos indicar también que se pueden utilizar otras representaciones de las derivadas. Por ejemplo, la **fórmula de nueve puntos** es

$$\begin{aligned}\nabla^2 u \approx & \frac{1}{6h^2} [4u(x+h, y) + 4u(x-h, y) + 4u(x, y+h) + 4u(x, y-h) \\ & + u(x+h, y+h) + u(x-h, y+h) + u(x+h, y-h) \\ & + u(x-h, y-h) - 20u(x, y)]\end{aligned}\quad (5)$$

Esta fórmula es del orden $\mathcal{O}(h^2)$. En el caso especial en que u sea una **función armónica** (lo que significa que es una solución de la ecuación de Laplace), la fórmula de nueve punto es de orden $\mathcal{O}(h^0)$. Para más detalles, véase Forsythe y Wasow [1960, pp. 194-195]. Por lo tanto, es una aproximación muy precisa usar métodos de diferencias finitas y resolver la ecuación de Poisson $\nabla^2 u = g$, con g una función armónica. Para problemas más generales, la fórmula de nueve puntos (5) tiene el mismo orden del término de error que la fórmula de cinco puntos (3) [a saber, $\mathcal{O}(h^2)$] y no hay mejoría sobre esta.

Si el espaciado de la malla no es regular (por ejemplo, h_1, h_2, h_3 y h_4 son izquierda, abajo, derecha y el espacio superior, respectivamente), entonces no es difícil demostrar que en (x, y) la **fórmula irregular de cinco puntos** es

$$\begin{aligned}\nabla^2 u \approx & \frac{1}{\frac{1}{2}h_1h_3(h_1+h_3)}[h_1u(x+h_3, y) + h_3u(x-h_1, y)] \\ & + \frac{1}{\frac{1}{2}h_2h_4(h_2+h_4)}[h_2u(x, y+h_4) + h_4u(x, y-h_2)] \\ & - 2\left(\frac{1}{h_1h_3} + \frac{1}{h_2h_4}\right)u(x, y)\end{aligned}\quad (6)$$

que sólo es de orden h cuando $h_i = \alpha_i h$ para $0 < \alpha_i < 1$. Esta fórmula se utiliza generalmente cerca de los puntos en la frontera, como se muestra en la figura 15.12. Sin embargo, si la malla es pequeña, esos puntos se pueden mover un poco para evitar el uso de (6). Esta perturbación de la región R

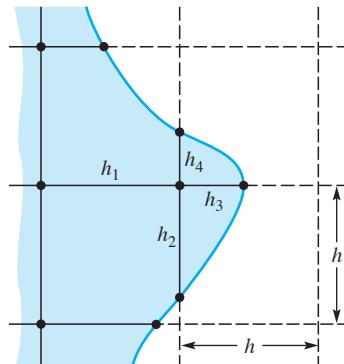


FIGURA 15.12
Puntos en
la frontera:
espaciado de
malla irregular

(en la mayoría de los casos para h pequeña) produce un error no mayor que el que se introduce al usar el esquema irregular (6).

Volviendo al problema modelo (1), cubrimos la región R con puntos de malla

$$x_i = ih \quad y_j = jh \quad (i, j \geq 0) \quad (7)$$

En este momento, es conveniente introducir una notación abreviada:

$$u_{ij} = u(x_i, y_i) \quad f_{ij} = f(x_i, y_i) \quad g_{ij} = g(x_i, y_i) \quad (8)$$

Con esto, la fórmula de cinco puntos adquiere una forma sencilla en el punto (x_i, y_i) :

$$(\nabla^2 u)_{ij} \approx \frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij}) \quad (9)$$

Si esta aproximación se hace en la ecuación diferencial (1), el resultado es (usted debe verificarlo)

$$-u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} + (4 - h^2 f_{ij}) u_{ij} = -h^2 g_{ij} \quad (10)$$

Los coeficientes de esta ecuación se puede ilustrar con una estrella de cinco puntas en el que a cada punta le corresponde el coeficiente de u en la malla (vea la figura 15.13).

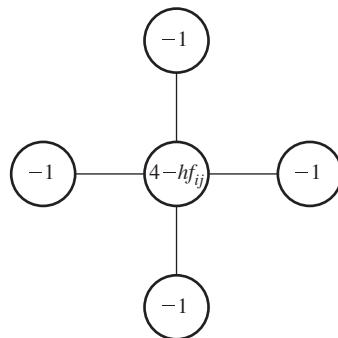


FIGURA 15.13
Ecuaciones
de Helmholtz:
estrella de cinco
puntas

Para ser más específicos, se supone que la región R es un cuadrado unitario y que la malla tiene un espaciado de $h = \frac{1}{4}$ (figura 15.14). Se obtiene una ecuación lineal simple de la forma (10) para cada uno de los nueve puntos de la malla interior.

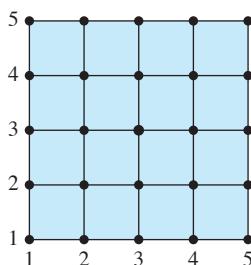


FIGURA 15.14
Espaciamiento
de la malla
uniforme

Estas nueve ecuaciones son las siguientes:

$$\left\{ \begin{array}{l} -u_{21} - u_{01} - u_{12} - u_{10} + (4 - h^2 f_{11})u_{11} = -h^2 g_{11} \\ -u_{31} - u_{11} - u_{22} - u_{20} + (4 - h^2 f_{21})u_{21} = -h^2 g_{21} \\ -u_{41} - u_{21} - u_{32} - u_{30} + (4 - h^2 f_{31})u_{31} = -h^2 g_{31} \\ -u_{22} - u_{02} - u_{13} - u_{11} + (4 - h^2 f_{12})u_{12} = -h^2 g_{12} \\ -u_{32} - u_{12} - u_{23} - u_{21} + (4 - h^2 f_{22})u_{22} = -h^2 g_{22} \\ -u_{42} - u_{22} - u_{33} - u_{31} + (4 - h^2 f_{32})u_{32} = -h^2 g_{32} \\ -u_{23} - u_{03} - u_{14} - u_{12} + (4 - h^2 f_{13})u_{13} = -h^2 g_{13} \\ -u_{33} - u_{13} - u_{24} - u_{22} + (4 - h^2 f_{23})u_{23} = -h^2 g_{23} \\ -u_{43} - u_{23} - u_{34} - u_{32} + (4 - h^2 f_{33})u_{33} = -h^2 g_{33} \end{array} \right.$$

Este sistema de ecuaciones se puede resolver con eliminación gaussiana, pero lo vamos a examinar más de cerca. Hay 45 coeficientes. Puesto que u se conoce en los puntos de frontera, movemos estos 12 términos a la derecha, dejando sólo 33 entradas diferentes de cero, de las 81, en nuestro sistema de 9×9 . La eliminación gaussiana estándar causa una gran cantidad de relleno en la fase de eliminación hacia adelante, es decir, las entradas cero se sustituirán por los valores distintos de cero. Así que buscamos un método que conserve la estructura dispersa de este sistema. Para ilustrar lo disperso de este sistema de ecuaciones, se escribe en notación matricial:

$$\mathbf{A}\mathbf{u} = \mathbf{b} \quad (11)$$

Supongamos que ordenamos las incógnitas de izquierda a derecha y de abajo arriba:

$$\mathbf{u} = [u_{11}, u_{21}, u_{31}, u_{12}, u_{22}, u_{32}, u_{13}, u_{23}, u_{33}]^T \quad (12)$$

Esto se llama el **orden natural**. Ahora, la matriz de coeficientes es

$$\mathbf{A} = \begin{bmatrix} 4 - h^2 f_{11} & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 - h^2 f_{21} & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 - h^2 f_{31} & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 - h^2 f_{12} & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 - h^2 f_{22} & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 - h^2 f_{32} & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 - h^2 f_{13} & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 - h^2 f_{23} & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 - h^2 f_{33} \end{bmatrix}$$

y el lado derecho es

$$\mathbf{b} = \begin{bmatrix} -h^2 g_{11} + u_{10} + u_{01} \\ -h^2 g_{21} + u_{20} \\ -h^2 g_{31} + u_{30} + u_{41} \\ -h^2 g_{12} + u_{02} \\ -h^2 g_{22} \\ -h^2 g_{32} + u_{42} \\ -h^2 g_{13} + u_{14} + u_{03} \\ -h^2 g_{23} + u_{24} \\ -h^2 g_{33} + u_{34} + u_{43} \end{bmatrix}$$

Observe que si $f(x, y) < 0$, entonces \mathbf{A} es una matriz diagonal dominante.

Método iterativo de Gauss-Seidel

Como las ecuaciones son de la misma forma, los métodos iterativos se utilizan con frecuencia para resolver estos sistemas dispersos. Resolviendo para la diagonal desconocida, tenemos que a partir de la ecuación (10) el **método de Gauss-Seidel** o **iteración** dada por

$$u_{ij}^{(k+1)} = \frac{1}{4 - h^2 f_{ij}} \left(u_{i+1,j}^{(k)} + u_{i-1,j}^{(k+1)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k+1)} - h^2 g_{ij} \right)$$

Si tenemos los valores aproximados de las incógnitas en cada punto de la malla, esta ecuación se puede utilizar para generar nuevos valores. Llamamos $u^{(k)}$ a los valores actuales de las incógnitas en la iteración k y $u^{(k+1)}$ el valor en la siguiente iteración. Además, los nuevos valores se utilizan en esta ecuación tan pronto como estén disponibles. El método de Gauss-Seidel y otros métodos iterativos se analizan en la sección 8.2.

El seudocódigo de este método en un rectángulo es el siguiente:

```

procedure Seidel(a_x, a_y, n_x, n_y, h, itmax, (u_ij))
integer i, j, k, n_x, n_y, itmax
real a_x, a_y, x, y;  real array (u_ij)_{0:n_x, 0:n_y}
for k = 1 to itmax do
    for j = 1 to n_y - 1 do
        y ← a_y + jh
        for i = 1 to n_x - 1 do
            x ← a_x + ih
            v ← u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}
            u_{ij} ← (v - h^2 g(x, y)) / (4 - h^2 f(x, y))
        end for
    end for
end for
end procedure Seidel

```

Al utilizar este procedimiento, se debe decidir sobre el número de pasos iterativos por calcular, $itmax$. Las coordenadas de la esquina inferior izquierda del rectángulo, (a_x, a_y) y el tamaño de paso h están dados. El número de puntos x de la red es n_x y el número de puntos y de la red es n_y .

Ejemplo numérico y seudocódigo

Vamos a ilustrar este procedimiento en el problema con valores en la frontera

$$\nabla^2 u - \frac{1}{25} u = 0 \quad \text{dentro de } R \text{ (cuadrado unitario)} \quad (13)$$

$$u = q \quad \text{en la frontera de } R$$

donde $q = \cosh\left(\frac{1}{5}x\right) + \cosh\left(\frac{1}{5}y\right)$. De este problema conocemos la solución $u = q$. Se presenta a continuación un seudocódigo de controlador para el procedimiento de Gauss-Seidel, comenzando con $u = 1$ y tomando 20 iteraciones. Considere que sólo se necesitan 81 palabras para el arreglo en la solución iterativa del sistema lineal de 49×49 . Aquí, $h = \frac{1}{8}$.

```

program Eliptico
integer i, j; real h, x, y; real array (uij)0:nx,0:ny
integer nx ← 8, ny ← 8, itmax ← 20
real ax ← 0, bx ← 1, ay ← 0, by ← 1
h ← (bx - ax)/nx
for j = 0 to ny do
    y ← ay + jh
    u0j ← Bndy(ax, y)
    unx,j ← Bndy(bx, y)
end for
for i = 0 to nx do
    x ← ax + ih
    ui0 ← Bndy(x, ay)
    ui,ny ← Bndy(x, by)
end for
for j = 1 to ny - 1 do
    y ← ay + jh
    for i = 1 to nx - 1
        x ← ax + ih
        uij ← Ustart(x, y)
    end for
end for
output 0, Norm((uij), nx, ny)
call Seidel(ax, ay, nx, ny, h, itmax, (uij))
output itmax, Norm((uij), nx, ny)
for j = 0 to ny do
    y ← ay + jh
    for i = 0 to nx do
        x ← ax + ih
        uij ← |True_Solution(x, y) - uij|
    end for
end for
output itmax, Norm((uij), nx, ny)
end program Eliptico

```

Para este problema modelo, las funciones acompañantes se presentan a continuación:

```

real function f(x, y)
real x, y
f ← -0.04
end function f

```

```

real function Bndy(x, y)
real x, y
Bndy ← True_Solution(x, y)
end function Bndy

```

```

real function g(x, y)
real x, y
g ← 0
end function g

```

```

real function Ustart(x, y)
real x, y
Ustart ← 1
end function Ustart

```

```

real function True_Solution(x,y)
real x,y
True_Solution  $\leftarrow \cosh(0.2x) + \cosh(0.2y)$ 
end function True_Solution

real function Norm((uij), nx, ny)
real array (uij)0:nx,0:ny
t  $\leftarrow 0$ 
for i = 1 to nx - 1 do
for j = 1 to ny - 1 do
    t  $\leftarrow t + u_{ij}^2$ 
    end for
end for
Norm  $\leftarrow \sqrt{t}$ 
end function Norm

```

Después de 75 iteraciones, los valores calculados en los 49 puntos de la malla interior son los siguientes:

2.0000	2.0003	2.0013	2.0028	2.0050	2.0078	2.0113	2.0154	2.0201
2.0003	2.0006	2.0016	2.0031	2.0053	2.0081	2.0116	2.0157	2.0204
2.0013	2.0016	2.0025	2.0041	2.0062	2.0091	2.0125	2.0166	2.0213
2.0028	2.0031	2.0041	2.0056	2.0078	2.0106	2.0141	2.0182	2.0229
2.0050	2.0053	2.0062	2.0078	2.0100	2.0128	2.0163	2.0204	2.0251
2.0078	2.0081	2.0091	2.0106	2.0128	2.0156	2.0191	2.0232	2.0279
2.0113	2.0116	2.0125	2.0141	2.0163	2.0191	2.0225	2.0266	2.0313
2.0154	2.0157	2.0166	2.0182	2.0204	2.0232	2.0266	2.0307	2.0354
2.0201	2.0204	2.0213	2.0229	2.0251	2.0279	2.0313	2.0354	2.0401

La norma euclíadiana $\|u\|_2^2 = \sum_{i=1}^{n_x-1} \sum_{j=1}^{n_y-1} u_{ij}^2$ de la diferencia entre los valores calculados y la solución conocida del problema con valores en la frontera (13) es de aproximadamente 0.47×10^{-4} .

Este ejemplo es una buena ilustración del hecho de que el problema numérico que se resolvió es el sistema de ecuaciones lineales (11), que es una aproximación discreta al problema con valor en la frontera continuo (13). Al comparar la verdadera solución de (13) con la solución calculada del sistema, recordemos el error de discretización implicado al hacer la aproximación. Este error es $\mathcal{O}(h^2)$. Con h tan grande como $h = \frac{1}{8}$, la mayoría de los errores en la solución calculada se debe al error de discretización! Para obtener un mejor acuerdo entre los problemas discreto y continuo, se selecciona un tamaño de malla mucho más pequeño. Por supuesto, el sistema lineal resultante tendrá una matriz de coeficientes grande y muy dispersa. Los métodos iterativos son ideales para la solución de estos sistemas que surgen de las ecuaciones diferenciales parciales. Para obtener información adicional, consulte las referencias listadas al final de esta sección.

Para una amplia gama de aplicaciones de ingeniería y la ciencia, Matlab tiene una caja de herramientas de EDP para la solución numérica de ecuaciones diferenciales parciales. Se pueden acomodar dos variables espaciales y una variable de tiempo. Después se discretiza la ecuación sobre una malla no estructurada, se aplican los elementos finitos para resolverla y se ofrece una facilidad para visualizar los resultados. El primer ejemplo

es la ecuación de Poisson

$$\nabla^2 u = -1$$

en el círculo unitario con $u = 0$ en la frontera. Se realiza una comparación de la solución de elemento finito con la solución exacta.

Métodos de elemento finito

El método de elemento finito se ha convertido en una de las principales estrategias para la resolución de ecuaciones diferenciales parciales. Ofrece una alternativa a los métodos de diferencias finitas analizados hasta ahora en este capítulo.

Como ejemplo, podemos desarrollar una versión del método de elemento finito para la ecuación de Poisson

$$\nabla^2 u \equiv u_{xx} + u_{yy} = r$$

donde r es una función constante. La ecuación diferencial parcial es válida en una región R dada en un plano bidimensional. Resolver la ecuación de Poisson equivale a minimizar la expresión

$$J(u) = \int \int_R \left[\frac{1}{2} (u_x^2 + u_y^2) + ru \right] dx dy$$

Esto significa que si la función u minimiza la expresión anterior, entonces u obedece a la ecuación de Poisson. Supongamos que la región se subdivide en triángulos usando tantas aproximaciones como se necesite. La función u se aproxima por una función φ que está compuesta de elementos triangulares planos, cada uno definido por una pieza triangular de R . Entonces, se considera el problema sustituto para minimizar

$$\sum_e J_e(\varphi^{(e)})$$

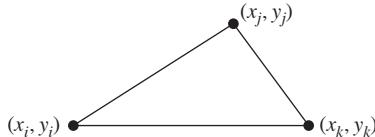
donde cada término de la suma se evaluará dentro de su propia base triangular T como se describe a continuación. (Aceptando esta teoría con confianza, se debe poder captar la idea general del método del elemento finito.)

Supongamos que una base triangular tiene vértices (x_i, y_i) , (x_j, y_j) y (x_k, y_k) . La superficie solución arriba del triángulo se approxima con un elemento triangular plano denotado por $\varphi^{(e)}(x, y)$, donde el superíndice indica este elemento. Sean z_i , z_j y z_k las distancias al plano de los vértices del triángulo llamados *nodos*. Sea $L_i^{(e)}$ uno en el nodo i y cero en los nodos j y k . Del mismo modo, sea $L_j^{(e)}$ uno en el nodo j y cero en los nodos i y k , y sea $L_k^{(e)}$ uno en el nodo k la cero en los nodos i y j .

Como se muestra en la figura 15.15, el área de la base triangular, que se denota por Δ_e , está dada por

$$\begin{aligned} \Delta_e &= \frac{1}{2} \text{Det} \begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix} \\ &= x_j y_k + x_i y_j + x_k y_i - x_j y_i - x_i y_k - x_k y_j \end{aligned}$$

FIGURA 15.15
Base triangular



En consecuencia, se obtiene

$$\begin{aligned} L_i^{(e)} &= \frac{1}{2} \Delta_e^{-1} \text{Det} \begin{bmatrix} 1 & x & y \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix} \\ &= \frac{1}{2} \Delta_e^{-1} [(x_j y_k - x_k y_j) + (y_j - y_k)x + (x_k - x_j)y] \\ &\equiv \frac{1}{2} \Delta_e^{-1} (a_i^{(e)} + b_i^{(e)}x + c_i^{(e)}y) \end{aligned}$$

Hemos definido los coeficientes $a_i^{(e)}$, $b_i^{(e)}$ y $c_i^{(e)}$. Del mismo modo, encontramos

$$\begin{aligned} L_j^{(e)} &= \frac{1}{2} \Delta_e^{-1} \text{Det} \begin{bmatrix} 1 & x & y \\ 1 & x_k & y_k \\ 1 & x_i & y_i \end{bmatrix} \\ &= \frac{1}{2} \Delta_e^{-1} [(x_k y_i - x_i y_k) + (y_k - y_i)x + (x_i - x_k)y] \\ &\equiv \frac{1}{2} \Delta_e^{-1} (a_j^{(e)} + b_j^{(e)}x + c_j^{(e)}y) \end{aligned}$$

y

$$\begin{aligned} L_k^{(e)} &= \frac{1}{2} \Delta_e^{-1} \text{Det} \begin{bmatrix} 1 & x & y \\ 1 & x_i & y_i \\ 1 & x_j & y_j \end{bmatrix} \\ &= \frac{1}{2} \Delta_e^{-1} [(x_i y_j - x_j y_i) + (y_i - y_j)x + (x_j - x_i)y] \\ &\equiv \frac{1}{2} \Delta_e^{-1} (a_k^{(e)} + b_k^{(e)}x + c_k^{(e)}y) \end{aligned}$$

Por último, se obtiene

$$\varphi^{(e)} = L_i^{(e)} z_i + L_j^{(e)} z_j + L_k^{(e)} z_k$$

Tenemos

$$J_e(\varphi^{(e)}) = \int \int_T \left[\frac{1}{2} \left((\varphi_x^{(e)})^2 + (\varphi_y^{(e)})^2 \right) + r \varphi^{(e)} \right] dx dy \equiv F(z_i, z_j, z_k)$$

Para resolver el problema de minimización, hacemos las derivadas adecuadas iguales a cero, lo que requiere derivar las componentes. Observe que

$$\varphi_x^{(e)} = \frac{1}{2} \Delta_e^{-1} (b_i^{(e)} z_i + b_j^{(e)} z_j + b_k^{(e)} z_k)$$

y

$$\varphi_y^{(e)} = \frac{1}{2} \Delta_e^{-1} (c_i^{(e)} z_i + c_j^{(e)} z_j + c_k^{(e)} z_k)$$

Realizamos las derivaciones

$$\begin{aligned}\partial F/\partial z_i &= \int_T \int_T (\varphi_x^{(e)} \varphi_{xz_i}^{(e)} + \varphi_y^{(e)} \varphi_{yz_i}^{(e)} + r \varphi_{z_i}^{(e)}) dx dy \\ &= \int_T \int_T \left(\varphi_x^{(e)} \frac{1}{2} \Delta_e^{-1} b_i^{(e)} + \varphi_y^{(e)} \frac{1}{2} \Delta_e^{-1} c_i^{(e)} + r L_i^{(e)} \right) dx dy \\ &= \frac{1}{4} \Delta_e^{-1} \left[\left(\left(b_i^{(e)} \right)^2 + \left(c_i^{(e)} \right)^2 \right) z_i + \left(b_i^{(e)} b_j^{(e)} + c_i^{(e)} c_j^{(e)} \right) z_j \right. \\ &\quad \left. + \left(b_i^{(e)} b_k^{(e)} + c_i^{(e)} c_k^{(e)} \right) z_k \right] + r \frac{1}{3} \Delta_e\end{aligned}$$

Aquí, las integrales son sencillas con cálculo elemental. Además, se puede demostrar que

$$\int_T \int_T L_i^{(e)} dx dy = \int_T \int_T L_j^{(e)} dx dy = \int_T \int_T L_k^{(e)} dx dy = \frac{1}{3} \Delta_e$$

donde Δ_e es el área de cada triángulo T . Resultados similares se obtienen para $\partial F/\partial z_j$ y $\partial F/\partial z_k$. En consecuencia, hacemos

$$\begin{bmatrix} \partial F/\partial z_i \\ \partial F/\partial z_j \\ \partial F/\partial z_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

y obtenemos

$$\begin{bmatrix} \left(b_i^{(e)} \right)^2 + \left(c_i^{(e)} \right)^2 & b_i^{(e)} b_j^{(e)} + c_i^{(e)} c_j^{(e)} & b_i^{(e)} b_k^{(e)} + c_i^{(e)} c_k^{(e)} \\ b_i^{(e)} b_j^{(e)} + c_i^{(e)} c_j^{(e)} & \left(b_j^{(e)} \right)^2 + \left(c_j^{(e)} \right)^2 & b_j^{(e)} b_k^{(e)} + c_j^{(e)} c_k^{(e)} \\ b_i^{(e)} b_k^{(e)} + c_i^{(e)} c_k^{(e)} & b_j^{(e)} b_k^{(e)} + c_j^{(e)} c_k^{(e)} & \left(b_k^{(e)} \right)^2 + \left(c_k^{(e)} \right)^2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = -\frac{4}{3} r \Delta_e^2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Esta ecuación matricial contiene todos los ingredientes que necesitamos para montar las derivadas parciales. En una aplicación particular, necesitamos hacer el montaje apropiado. Para cada elemento $\varphi^{(e)}$, los nodos activos i, j y k son los que contribuyen a los valores distintos de cero. Estas contribuciones se registran en las derivadas en relación con las variables correspondientes entre las z_i, z_j, z_k y así sucesivamente.

EJEMPLO 1 Aplique el método de elemento finito para resolver la ecuación de Poisson $u_{xx} + u_{yy} = 4$ en el cuadrado unitario con las triangulaciones que se muestran en la figura 15.16 y utilizando los valores en la frontera correspondientes a la solución exacta $u(x, y) = x^2 + y^2$.

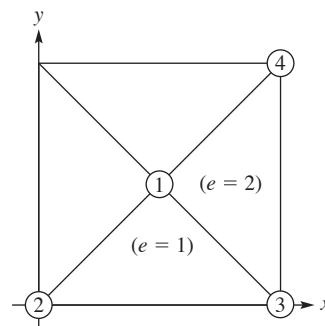


FIGURA 15.16
Triangulación

Solución Por simetría, necesitamos considerar sólo la parte inferior derecha del cuadrado, que se ha dividido en dos triángulos. Los ingredientes de entrada son los nodos 1 a 4, donde las coordenadas (x, y) son las siguientes: nodo 1: $(\frac{1}{2}, \frac{1}{2})$, nodo 2: $(0, 0)$, nodo 3: $(1, 0)$ y nodo 4: $(1, 1)$. Los elementos son dos triángulos con los números de nodo indicados: $e = 1: 1, 2, 3$ y $e = 2: 1, 3, 4$. El lector astuto se dará cuenta de que las coordenadas z se deben determinar sólo para el nodo 1, ¡ya que son los valores en la frontera para los nodos 2, 3, 4! Sin embargo, vamos a ignorar este hecho por el momento para ilustrar el proceso de montaje del método del elemento finito. Observe que las áreas de los elementos triangulares son $\Delta_1 = \Delta_2 = \frac{1}{4}$ y $r = 4$. Primero, calculamos los coeficientes $a^{(e)}, b^{(e)}, c^{(e)}$ a partir de esta información básica. En la siguiente tabla, cada columna corresponde a un nodo (i, j, k) :

	$e = 1$			$e = 2$		
$a^{(e)}$	0	$\frac{1}{2}$	0	1	0	$-\frac{1}{2}$
$b^{(e)}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	-1	$\frac{1}{2}$	$\frac{1}{2}$
$c^{(e)}$	1	$-\frac{1}{2}$	$-\frac{1}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$

Se puede verificar que las columnas producen las funciones deseadas $L_i^{(e)}, L_j^{(e)}$ y $L_k^{(e)}$. Por ejemplo, la primera columna indica $L_i^{(1)} = \frac{1}{2}\Delta_1^{-1}[0 + 0 \cdot x + 1 \cdot y] = 2y$. En el nodo 1, esto da el valor 1, mientras que en los nodos 2 y 3, da el valor 0. Del mismo modo, las otras columnas producen los resultados deseados.

A continuación, obtenemos la ecuación matricial para el elemento $e = 1$:

$$\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{3} \\ -\frac{1}{3} \\ -\frac{1}{3} \end{bmatrix}$$

y la ecuación matricial para el elemento $e = 2$:

$$\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} -\frac{1}{3} \\ -\frac{1}{3} \\ -\frac{1}{3} \end{bmatrix}$$

Entonces montamos las dos matrices, que quedan

$$\begin{bmatrix} 2 & -\frac{1}{2} & -1 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -\frac{1}{2} & 0 & 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} -\frac{2}{3} \\ -\frac{1}{3} \\ -\frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}$$

Ahora que hemos ilustrado el proceso de montaje de los elementos, podemos encontrar rápidamente la solución usando el hecho de que $z_2 = 0$, $z_3 = 1$ y $z_4 = 2$, ya que son los valores en la frontera. Usando estos valores en la última ecuación matricial anterior encontramos inmediatamente que $z_1 = \frac{2}{3}$. Esta es una burda aproximación, ya que el verdadero valor es $\frac{1}{2}$. Recuerde que $u(x, y) = x^2 + y^2$ es la solución exacta. █

Podemos obtener aproximaciones más precisas sumando más elementos y escribiendo un programa de computadora para manejar los cálculos (véase el problema de cómputo del 15.3.15). Para más detalles, consulte Scheid [1990] y Sauer [2006].

Más de elementos finitos

Primero, tomamos un enfoque muy general sobre este tema, suponiendo que tenemos una transformación lineal \mathbf{A} y que se quiere resolver la ecuación

$$\mathbf{A}\mathbf{u} = \mathbf{b}$$

para \mathbf{u} , cuando se conoce \mathbf{b} . Esto, obviamente, incluye el caso en que \mathbf{A} es una matriz de $m \times n$ y \mathbf{b} es un vector de m componentes. Pero hay muchos problemas complicados que se ajustan a este mismo molde.

Por ejemplo, \mathbf{A} puede ser un operador diferencial lineal y tal vez se quiere resolver un problema con dos valores en la frontera que implique a este, tal como

$$\begin{cases} u''(t) + 2u(t) = t^2 & (0 \leq t \leq 1) \\ u(0) = u(1) = 0 \end{cases}$$

Aquí, \mathbf{A} opera en funciones y se define con la ecuación $\mathbf{A}\mathbf{u} = u'' + 2\mathbf{u}$.

Otro ejemplo de gran importancia es el problema modelo de la ecuación (1). En este caso, \mathbf{A} sería el operador diferencial laplaciano. Este problema también se analiza en el capítulo 17.

La estrategia básica del método del elemento finito para resolver la ecuación $\mathbf{A}\mathbf{u} = \mathbf{b}$ es seleccionar las *funciones base* $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ y tratar de resolver la ecuación con una combinación lineal de estas funciones base. Puesto que se supone que \mathbf{A} es una transformación lineal, se obtiene

$$\mathbf{A}\mathbf{u} = \mathbf{A} \sum_{j=1}^n c_j \mathbf{v}_j = \sum_{j=1}^n c_j (\mathbf{A}\mathbf{v}_j) = \mathbf{b}$$

Ahora las incógnitas en el problema son los coeficientes c_j . Por lo general, la ecuación sólo muestra que es inconsistente porque \mathbf{b} no está en el espacio lineal de la serie de funciones $\{\mathbf{A}\mathbf{v}_1, \mathbf{A}\mathbf{v}_2, \dots, \mathbf{A}\mathbf{v}_n\}$. En este caso, hay que comprometerse y aceptar una solución aproximada para el conjunto de ecuaciones. Se pueden utilizar muchas tácticas para llegar a una solución aproximada al problema. Por ejemplo, se puede utilizar un método de mínimos cuadrados si el espacio lineal implicado tiene un producto interno, $\langle \cdot, \cdot \rangle$. Entonces los coeficientes c_j deben elegirse de manera que se cumpla con la condición de ortogonalidad, es decir,

$$\sum_{j=1}^n c_j \mathbf{A}\mathbf{v}_j - \mathbf{b} \perp \text{ Espacio } \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$$

Esto conduce a las **ecuaciones normales**

$$\sum_{j=1}^n \langle \mathbf{A}\mathbf{v}_j, \mathbf{v}_i \rangle c_j = \langle \mathbf{b}, \mathbf{v}_i \rangle \quad (1 \leq i \leq n)$$

Estas ecuaciones para los coeficientes c_j son también conocidas (en este contexto) como **ecuaciones de Galerkin**. Forman un sistema de n ecuaciones lineales con n incógnitas.

Vamos a ilustrar este proceso con un problema con dos valores en la frontera que implica una ecuación diferencial ordinaria de segundo orden:

$$\begin{cases} u''(t) + g(t)u(t) = f(t) \\ u(0) = a \quad u(1) = b \end{cases}$$

El método de elemento finito por lo general utiliza funciones *locales* como funciones base en el análisis anterior. Esto significa que cada función base debe ser cero, excepto en un intervalo corto. Los splines B tienen esta propiedad, por lo que con frecuencia se utilizan en el método de elemento

finito. En el problema actual, queremos utilizar splines B que tengan dos derivadas continuas, ya que el operador A se definirá por

$$Au = u'' + gu$$

Por lo tanto, se sugieren los splines cúbicos mismos. Se definen nudos $t_i = ih$, donde h es un tamaño de paso seleccionado. (Su recíproco debe ser un entero en este ejemplo.) Sean B_j^3 los splines cúbicos B correspondientes a los nudos dados. Esta es una lista infinita de splines B, como se analizó en el capítulo 9. Todos, salvo un número finito son iguales a cero en el intervalo $[0, 1]$. Los que no son idénticos a cero en el intervalo $[0, 1]$ puede ser reetiquetados como v_1, v_2, \dots, v_n . Estas son nuestras *funciones de prueba*. Procediendo como antes, llegamos a un conjunto de n ecuaciones lineales con n incógnitas. Los detalles requieren que se encuentren las funciones Av_j utilizando las fórmulas de spline B del capítulo 9. Es tedioso y no muy instructivo.

Se pueden aplicar consideraciones similares a la ecuación de Laplace en un dominio dado. Para ilustrar, tomamos el dominio como un cuadrado de lado 2, donde $0 \leq x, y \leq 2$. En la frontera del cuadrado, se requiere que $u(x, y) = \text{sen}(xy)$. Este problema se conoce como un **problema de Dirichlet**. Para las funciones de base, usamos las funciones v_j que ya satisfacen la parte homogénea del problema. Es decir, queremos que cada v_j satisfaga la ecuación de Laplace dentro del dominio del cuadrado. Las funciones que satisfacen la ecuación de Laplace son **armónicas**. Podemos aprovechar el hecho de que las partes real e imaginaria de una función analítica son armónicas. Así, si hacemos $z = x + iy$ y calculamos z^k , seremos capaces de extraer las funciones armónicas que son polinomios. Aquí se presentan algunos polinomios armónicos, v_j para $0 \leq j \leq 6$:

$$\begin{aligned} z &= 1 & v_0(x, y) &= 1 \\ z &= x + iy & v_1(x, y) &= x & v_2(x, y) &= y \\ z^2 &= (x + iy)^2 & v_3(x, y) &= x^2 - y^2 & v_4(x, y) &= 2xy \\ z^3 &= (x + iy)^3 & v_5(x, y) &= x^3 - 3xy^2 & v_6(x, y) &= 3x^2y - y^3 \end{aligned}$$

Usando estas siete funciones, formamos $u = \sum_{j=0}^6 c_j v_j$. Esta satisface la ecuación de Laplace y podemos concentrarnos en hacer que u se acerque al valor en la frontera $x^3 - y^2$ en el perímetro del cuadrado. Hay muchas maneras de proceder y elegimos como primera un método llamado **colocación**. En este proceso, seleccionamos un número de puntos en la frontera y escribimos una ecuación en cada punto que dice que el valor de $\sum_{j=0}^6 c_j v_j$ es igual al valor dado. Si el número de puntos es igual al número de funciones base, tenemos el método de colocación clásico. Aquí, tomamos ocho puntos, mientras que sólo hay siete funciones y siete coeficientes. Por lo tanto, pedimos una solución de mínimos cuadrados. Tomamos los así llamados **puntos de colocación** como $(0, 2), (1, 2), (2, 2), (2, 1), (2, 0), (1, 0), (0, 0)$ y $(0, 1)$. Esto nos conduce al siguiente sistema de ocho ecuaciones:

$$\left[\begin{array}{ccccccc} 1 & 0 & 2 & -4 & 0 & 0 & -8 \\ 1 & 1 & 2 & -3 & 4 & -11 & -2 \\ 1 & 2 & 2 & 0 & 8 & -16 & 16 \\ 1 & 2 & 1 & 3 & 4 & 2 & 11 \\ 1 & 2 & 0 & 4 & 0 & 8 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 & -1 \end{array} \right] \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix} = \begin{bmatrix} 0 \\ \text{sen}(2) \\ \text{sen}(4) \\ \text{sen}(2) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

La solución de mínimos cuadrados es un vector c con componentes

$$c = [0.3219, -0.8585, -0.8585, 0, 1.1931, 0.2146, -0.2146]^T$$

La **función residual** es $\sum_{j=0}^6 c_j \mathbf{v}_j - \mathbf{b}$, donde $\mathbf{b}(x, y) = \sin(xy)$. Su valor absoluto es 0.3219 en cada uno de los ocho puntos de colocación. Para mejorar la precisión, hay que emplear más funciones base y más puntos de colocación.

Otra técnica que se utiliza con frecuencia en el método de elemento finito es la sustitución de una ecuación diferencial por un problema de optimización. Esto se puede ilustrar con un problema de dos valores en la frontera, tal como

$$\begin{cases} (hu')' - gu = f \\ u(a) = \alpha \quad u(b) = \beta \end{cases}$$

En este caso, u es la función desconocida, mientras que h, g y f son funciones dadas, todas definidas en el intervalo $[a, b]$. Este problema se llama **problema de Sturm-Liouville**. Hay un acompañamiento funcional, definido por

$$\Phi(u) = \int_a^b [(u')^2 h + u^2 g + 2uf] dx$$

El problema funcional y con dos valores en la frontera están relacionados con varios teoremas. Uno de estos establece más o menos que si encontramos la función u que minimiza el funcional $\Phi(u)$ sujeto a las condiciones laterales $u(a) = \alpha$ y $u(b) = \beta$, entonces tendremos la solución del problema con valores en la frontera. Es posible aprovechar el hecho de que $\Phi(u)$ se define en tanto u tenga derivada, mientras que en la ecuación diferencial, se requiere una función que tenga dos derivadas. De hecho, para el problema funcional, sólo se requiere que u sea derivable por partes, una propiedad que las funciones spline de grado 0 y 1 tienen. Estas ideas se extienden a funciones de dos o más variables y permiten el uso de funciones spline de grado bajo en dos o más variables para aproximar la solución de una ecuación diferencial. Estas son las características principales del método de elemento finito. Para la teoría matemática de los métodos de elemento finito, consulte los libros de Brenner y Scott [2002], Strang [2006] y otros.

Resumen

(1) Estudiamos un problema modelo que implica las siguientes ecuaciones diferenciales parciales elípticas

$$\nabla^2 u + fu = g$$

en una región, con el valor de u dado en la frontera. El primer término implica el **operador de Laplace** ∇^2 , que es

$$\nabla^2 u \equiv \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

Colocando una rejilla sobre la región con espaciado uniforme h en ambas direcciones el término laplaciano se puede aproximar usando las **diferencias finitas de cinco puntos**

$$\nabla^2 u \approx \frac{1}{h^2} [u(x+h, y) + u(x-h, y) + u(x, y+h) + u(x, y-h) - 4u(x, y)]$$

En cada punto de la red interior, escribimos $u_{ij} = u(x_i, y_j) = u(ih, jh)$ y obtenemos la siguiente ecuación para nuestro problema modelo:

$$-u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} + (4 - h^2 f_{ij}) u_{ij} = -h^2 g_{ij}$$

Por lo general el sistema lineal de ecuaciones resultante es grande y disperso y se pueden utilizar los métodos iterativos para resolverlo. Por ejemplo, el **método iterativo de Gauss-Seidel** para nuestro

sistema lineal es

$$u_{ij}^{(k+1)} = \frac{1}{4 - h^2 f_{ij}} \left(u_{i+1,j}^{(k)} + u_{i-1,j}^{(k+1)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k+1)} - h^2 g_{ij} \right)$$

Los puntos de la malla se pueden ordenar en diferentes formas, tal como el orden natural o el ordenamiento rojo-negro, que afecta la rapidez de convergencia de los procedimientos iterativos.

(2) La característica distintiva del método de elemento finito es que resolvemos una ecuación $Ax = b$ aproximadamente haciendo $\mathbf{u} = \sum_{j=1}^n c_j \mathbf{v}_j$, donde $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ se eligen por el usuario. Los coeficientes desconocidos c_j se calculan de forma que $\sum_{j=1}^n c_j A \mathbf{v}_j$ esté lo más cerca posible a b . Normalmente, en las ecuaciones diferenciales parciales, las funciones \mathbf{v}_j serán funciones spline multidimensionales.

Referencias adicionales

Para más estudio y lectura, véase Ames [1992], Evans [2000], Forsythe y Wasow [1960], Gockenbach [2002], Mattheij, Rienstra y Boonkamp [2005], Ortega y Voigt [1985], Rice y Boisvert [1984], Smith [1965], Street [1973], Varga [1962, 2002], Wachspress [1966], Young [1971] y Young y Gregory [1972].

Problemas 15.3

1. Establezca la fórmula para el error en la
 - a. fórmula de cinco puntos, ecuación (3).
 - b. fórmula de nueve puntos, ecuación (5).
2. Establezca la fórmula irregular de los cinco puntos (6) y su término de error.
3. Escriba las matrices que se producen en la ecuación (11), cuando las incógnitas están ordenadas de acuerdo con el vector $\mathbf{u} = [u_{11}, u_{31}, u_{22}, u_{13}, u_{33}, u_{21}, u_{32}, u_{23}]^T$. Esto se conoce como **ordenamiento rojo-negro o de tablero de ajedrez**.
4. a. Verifique la ecuación (10).
b. Compruebe que la solución de la ecuación (13) es como se muestra en el libro.
5. Considere el problema de resolver la ecuación diferencial parcial

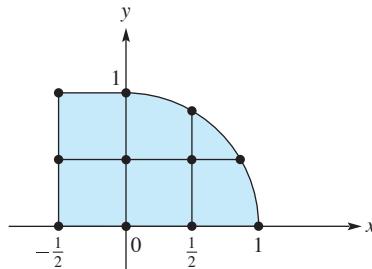
$$20u_{xx} - 30u_{yy} + \frac{5}{x+y}u_x + \frac{1}{y}u_y = 69$$

en una región R con u dada en la frontera. Deduzca una ecuación en diferencias finitas de cinco puntos de orden $\mathcal{O}(h^2)$ que corresponde a esta ecuación, en algún punto interior (x_i, y_j) .

6. Resuelva este problema con valor en la frontera para calcular $u\left(\frac{1}{2}, \frac{1}{2}\right)$ y $u(0, \frac{1}{2})$:

$$\begin{cases} \nabla^2 u = 0 & (x, y) \in R \\ u = x & (x, y) \in \partial R \end{cases}$$

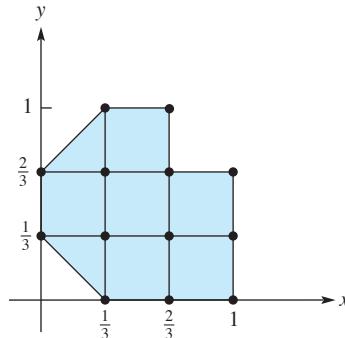
La región R con frontera ∂R se muestra en la figura (el arco es circular). Utilice $h = \frac{1}{2}$. Nota: este problema (y muchos otros en este libro) también se puede plantear en términos físicos. Por ejemplo, en este caso se busca la temperatura de estado estable en una viga de sección transversal R si la superficie de la viga se mantiene a una temperatura $u(x, y) = x$.



7. Considere el problema con valor en la frontera

$$\begin{cases} \nabla^2 u = 9(x^2 + y^2) & (x, y) \in R \\ u = x - y & (x, y) \in \partial R_1 \end{cases}$$

para la región en el cuadrado unitario con $h = \frac{1}{3}$ en la siguiente figura. Aquí, ∂R es la frontera de R , $\partial R_2 = \{(x, y) \in \partial R : \frac{2}{3} \leq x < 1, \frac{2}{3} \leq y < 1\}$ y $\partial R_1 = \partial R - \partial R_2$. En los puntos de la malla, determine el sistema de ecuaciones lineales que produce un valor aproximado de $u(x, y)$. Escriba el sistema en la forma $Au = b$.



8. Determine el sistema lineal por resolver si la fórmula de nueve puntos (5) se utiliza como aproximación en el problema de la ecuación (1). Observe el patrón de la matriz de coeficientes tanto en la fórmula de cinco puntos como en la de nueve puntos cuando se agrupan las incógnitas en cada renglón. (Dibuje líneas punteadas que pasen por A para formar submatrices de 3×3 .)

9. En la ecuación (11), muestre que A es diagonalmente dominante cuando $f(x, y) \leq 0$.

10. ¿Cuál es el sistema lineal si una **fórmula de nueve puntos alternativa**

$$\begin{aligned} \nabla^2 u \approx & \frac{1}{12h^2} [16u(x+h, y) + 16u(x-h, y) + 16u(x, y+h) \\ & + 16u(x, y-h) - u(x+2h, y) - u(x-2h, y) \\ & - u(x, y+2h) - u(x, y-2h) - 60u(x, y)] \end{aligned}$$

se utiliza? ¿Cuáles son las ventajas y desventajas de su uso? *Sugerencia:* tiene una precisión $\mathcal{O}(h^4)$.

11. (Opción múltiple) ¿Cuál es la ecuación de Laplace en tres variables?

- a. $u - x + u_y + u_z = 0$
- b. $u_{xx} + u_{yy} = 0$
- c. $u_{xx} + u_{yy} + u_{zz} = 0$
- d. $u_{xx} + u_{yy} = yu_t$
- e. Ninguna de estas.

12. (Opción múltiple) ¿Cuál de estas no es una función armónica de (x, y) ?

- a. $x^2 - y^2$
- b. $2xy$
- c. $x^3y - xy^3$
- d. $x^3 - xy^3$
- e. Ninguna de estas.

13. (Opción múltiple) En la solución del problema de Dirichlet en el cuadrado unitario, donde $0 < x < 1$ y $0 < y < 1$, supongamos que hemos elegido el tamaño de paso $h = 1/100$. ¿Cuántos valores desconocidos de la función $u(x, y)$ habrá en esta versión discreta del problema? Considere que $x_i = ih$ para $0 \leq i \leq n + 1$, y lo mismo para y_i . También, $x_0 = 0$ y $x_{n+1} = 1$, y lo mismo para y . *Sugerencia:* los valores en la frontera se dan en el perímetro del cuadrado y *no* son desconocidos.

- a. $9801 = 99^2$
- b. $10000 = 100^2$
- c. $10404 = 102^2$
- d. $10201 = 101^2$
- e. Ninguno de estos.

14. Sea $z^n = u_n + iv_n$. Verifique que u_n y v_n se pueden determinar mediante el algoritmo $u_0 = 1$, $v_0 = 1$, $u_{n+1} = xu_n - yv_n$ y $v_{n+1} = xv_n + yu_n$.

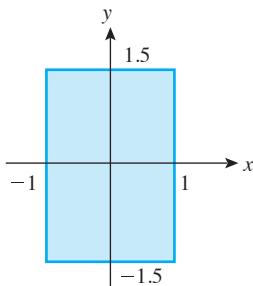
Problemas de cómputo 15.3

1. Imprima el sistema de ecuaciones lineales para resolver la ecuación (13) con $h = \frac{1}{4}$ y $\frac{1}{8}$. Resuelva estos sistemas utilizando los procedimientos *Gauss* y *Solve* del capítulo 7.

2. Trabaje con la rutina de Gauss-Seidel en el problema

$$\begin{cases} \nabla^2 u = 2e^{x+y} & (x, y) \in R \\ u = e^{x+y} & (x, y) \in \partial R \end{cases}$$

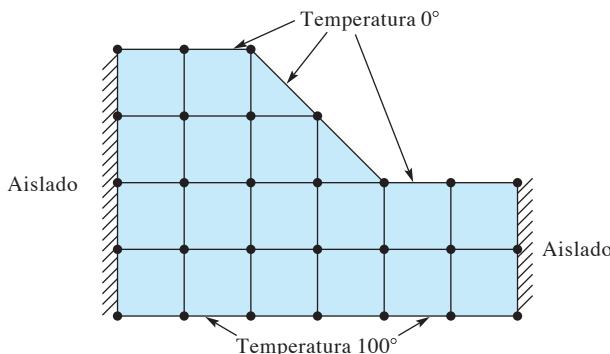
R es el rectángulo que se muestra en la figura. Los valores iniciales y las dimensiones de malla están en la tabla siguiente. Compare sus soluciones numéricas con las soluciones exactas después de *itmax* iteraciones.



Valores iniciales	h	itmax
$u = xy$	0.1	15
$u = 0$	0.2	20
$u = (1 + x)(1 + y)$	0.25	40
$u = \left(1 + x + \frac{1}{2}x^2\right) \left(1 + y + \frac{1}{2}y^2\right)$	0.05	100
$u = 1 + xy$	0.25	200

3. Modifique el procedimiento de Gauss-Seidel para manejar el ordenamiento rojo-negro. Repita el problema anterior de cómputo con este ordenamiento. ¿El ordenamiento hace alguna diferencia? (Véase el problema 15.3.3.)
4. Reescriba el seudocódigo de Gauss-Seidel de manera que pueda manejar cualquier ordenamiento, es decir, introduzca un arreglo de ordenamiento (ℓ_i). Pruebe algunos ordenamientos diferentes: natural, rojo-negro, espiral y en diagonal.
5. Considere el problema de transferencia de calor en la región irregular que se muestra en la siguiente figura. La expresión matemática de este problema es la siguiente:

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 & \text{interior} \\ \frac{\partial u}{\partial x} = 0 & \text{lados} \\ u = 0 & \text{arriba} \\ u = 100 & \text{abajo} \end{cases}$$



Aquí, la derivada parcial $\partial u / \partial x$ se puede aproximar con una fórmula de diferencia dividida. Establezca que las fronteras aisladas actúan como espejos de modo que podamos suponer que la temperatura es la misma que en un punto de la cuadrícula adyacente interior. Determine el sistema lineal asociado y resuelva para la temperatura u_i con $1 \leq i \leq 10$.

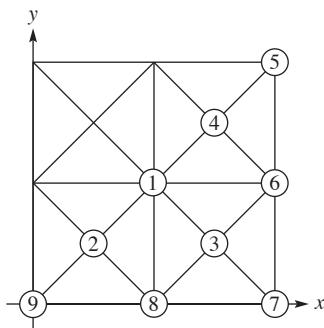
6. Modifique el procedimiento *Seidel*, de modo que utilice la fórmula de nueve puntos (5). Resuelva de nuevo el problema modelo (13) y compare los resultados.
7. Resuelva el ejemplo con que inicia este capítulo con $h = \frac{1}{9}$.

8. Resuelva el problema con valores en la frontera

$$\begin{cases} \nabla^2 u + 2u = g & \text{en la frontera de } R \\ u = 0 & \text{dentro de } R \end{cases}$$

donde $g(x, y) = (xy + 1)(xy - x - y) + x^2 + y^2$ y R es el cuadrado unitario. Esta problema tiene la solución conocida $u = \frac{1}{2}xy(x - 1)(y - 1)$. Utilice el procedimiento *Seidel* de Gauss-Seidel iniciando con $u = xy$ y tome 30 iteraciones.

9. (Continuación) Utilizando el procedimiento modificado *Seidel* del problema de cómputo 15.3.6, en el que se utiliza la fórmula de nueve puntos (5), vuelva a resolver este problema. Compare los resultados y explique la diferencia.
10. Para el problema de la EDP elíptica (13), use Maple, Mathematica o Matlab para encontrar la solución numérica del sistema lineal (11), donde $h = \frac{1}{4}$, $f_{ij} = \frac{1}{25}$ y $g_{ij} = 0$ en la matriz de coeficientes de 7×7 y el arreglo del lado derecho de 1×7 . Compárela con la solución exacta del problema con valores en la frontera, que es $u_{ij} = \cosh\left(\frac{1}{5}ih\right) + \cosh\left(\frac{1}{5}jh\right)$. También, compare estos resultados con los obtenidos en el ejemplo del libro cuando $h = \frac{1}{8}$ y se utilizó el método de Gauss-Seidel. ¿Qué conclusiones se pueden sacar?
11. Encuentre, aproximadamente, una función armónica en el dominio circular $x^2 + y^2 < 1$, que toma los valores $\sin 3\theta$ en el círculo frontera. En este caso, θ es la coordenada angular del punto en coordenadas polares. Utilice los siete polinomios base armónicos empleados en el ejemplo de esta sección. Elija 100 puntos igualmente espaciados en la circunferencia y use el método de colocación (ampliado), en el que se calcula una solución de mínimos cuadrados para el sistema de ecuaciones lineales.
12. En el ejemplo de colocación del texto, resuelva el problema de Dirichlet, pero sustituya los valores en la frontera $x^3 - x^2$.
13. Aproveche cualquier instrucción o procedimiento especial en los sistemas de software matemático como Matlab, Maple o Mathematica para resolver el ejemplo numérico (13).
14. (Continuación) Utilice la capacidad de manejo simbólico en el software matemático como Maple o Mathematica para verificar la solución general de (13).
15. Escriba un programa de cómputo utilizando el método de elemento finito para resolver la ecuación de Poisson $u_{xx} + u_{yy} = 4$, con condiciones en la frontera $u(x, y) = x^2 + y^2$ utilizando nueve nodos de la triangulación fina que se muestra. Véase Scheid (1988) para más detalles.



Minimización de funciones

Un problema de diseño de ingeniería conduce a una función

$$F(x, y) = \cos(x^2) + e^{(y-6)^2} + 3(x + y)^4$$

en el que x y y son parámetros que debe ser seleccionados y $F(x, y)$ es una función relacionada con el costo de fabricación y se debe minimizar. En este capítulo se desarrollan los métodos para localizar puntos óptimos (x, y) en este tipo de problemas.

16.1 Caso de una variable

Una importante aplicación del cálculo es el problema de encontrar el mínimo local de una función. Los problemas de maximización se cubren con la teoría de minimización, ya que los máximos de F ocurren en los puntos donde $-F$ tiene sus mínimos. En cálculo, la técnica principal de minimización es derivar la función cuyo mínimo se busca igualar la derivada a cero y localizar los puntos que satisfacen la ecuación resultante.

Esta técnica se puede utilizar en funciones de una o varias variables. Por ejemplo, si se desea un valor mínimo de $F(x_1, x_2, x_3)$, buscamos los puntos donde las tres derivadas parciales son al mismo tiempo iguales a cero:

$$\frac{\partial F}{\partial x_1} = \frac{\partial F}{\partial x_2} = \frac{\partial F}{\partial x_3} = 0$$

Este procedimiento no se puede aceptar como un método numérico de *propósito general* porque se requiere la derivación seguida de la solución de una o más de las ecuaciones en una o más de las variables utilizando los métodos del capítulo 3. Esta tarea puede ser tan difícil de realizar como un ataque directo y frontal del problema original.

Problemas de minimización con y sin restricciones

El problema de minimización tiene dos formas: *sin restricciones* y *con restricciones*. En un problema de minimización **sin restricciones**, se define una función F en el espacio n dimensional \mathbb{R}^n en la recta real \mathbb{R} y se busca un punto $z \in \mathbb{R}^n$ con la propiedad de que

$$F(z) \leq F(x) \quad \text{para toda } x \in \mathbb{R}^n$$

Es conveniente escribir los puntos en \mathbb{R}^n simplemente como x, y, z y así sucesivamente. Si se necesitan representar las componentes de un punto, podemos escribir $x = [x_1, x_2, \dots, x_n]^T$. En un pro-

blema de minimización con **restricciones**, está dado un subconjunto K en \mathbb{R}^n y se busca un punto $z \in K$ para el que

$$F(z) \leq F(x) \quad \text{para toda } x \in K$$

Estos problemas son más difíciles, ya que necesitan conservar los puntos dentro del conjunto K . A veces, el conjunto K se define de una manera complicada.

Considere el paraboloide elíptico $F(x_1, x_2) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 4$ que se muestra en la figura 16.1. El mínimo sin restricciones se presenta en $(1, 1)$, ya que $F(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 1)^2 + 2$. Si $K = \{(x_1, x_2) : x_1 \leq 0, x_2 \leq 0\}$, el mínimo restringido es 4 en $(0, 0)$.

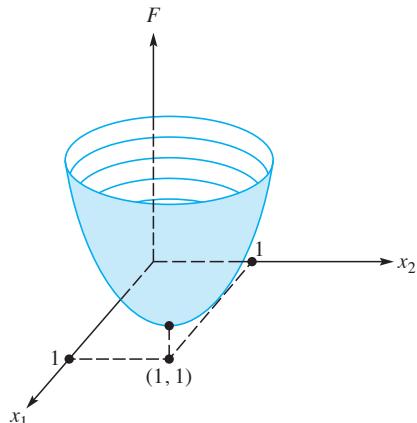


FIGURA 16.1
Paraboloide
elíptico

Sistemas de software matemático como Matlab, Maple, Mathematica tienen instrucciones para la optimización general de funciones lineales y no lineales. Por ejemplo, podemos resolver el problema de minimización correspondiente al paraboloide elíptico que se muestra en la figura 16.1. Primero, se define la función, se encuentra el valor mínimo cercano al punto $(\frac{1}{2}, \frac{1}{2})$ y se traza la gráfica de esta función. Se obtiene el punto mínimo como $(1, 1)$ y el valor de la función en este punto 2.

Caso de una variable

Se considera primero el caso especial en el que una función F se define en \mathbb{R} porque el problema más general con n variables con frecuencia se resuelve con una secuencia de problemas de una variable.

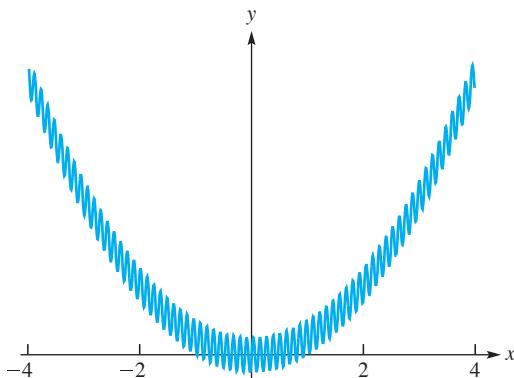
Suponga que $F: \mathbb{R} \rightarrow \mathbb{R}$ y que buscamos un punto $z \in \mathbb{R}$ con la propiedad de que $F(z) \leq F(x)$ para toda $x \in \mathbb{R}$. Observe que si no se hacen suposiciones acerca de F , este problema es difícil de resolver en su forma general. Por ejemplo, la función

$$f(x) = \frac{1}{1 + x^2}$$

no tiene un punto mínimo. Incluso en el caso de funciones relativamente bien comportadas, como

$$F(x) = x^2 + \operatorname{sen}(53x)$$

los métodos numéricos pueden encontrar algunas dificultades debido al gran número de mínimos puramente locales (figura 16.2). Recordemos que un punto z es un punto **mínimo local** de una función F si hay alguna vecindad de z en la que todos los puntos satisfacen $F(z) \leq F(x)$. Podemos

**FIGURA 16.2**

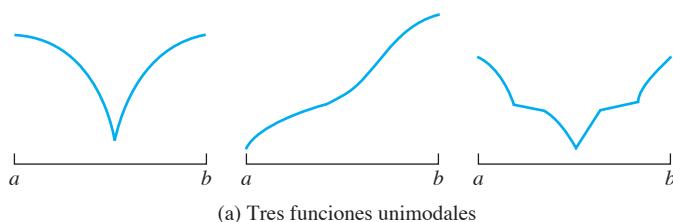
$$F(x) = x^2 + \operatorname{sen}(53x)$$

utilizar software matemático como Matlab y Mathematica para encontrar los valores del mínimo local de la función $F(x) = x^2 + \operatorname{sen}(53x)$. Primero, se define la función, luego se halla un valor mínimo local en el intervalo $[-\frac{1}{2}, \frac{1}{2}]$ y se traza la curva. El punto que se calcula ¡podría no ser un punto mínimo global! Para tratar de encontrar el punto mínimo global, podemos utilizar diversos valores iniciales para encontrar los valores mínimos locales y después hallar el mínimo de ellos (véase el problema de cómputo 16.1.6). De hecho, encontramos un mínimo local -0.99912 en $t = -0.0296166$, que es el mínimo global para esta función.

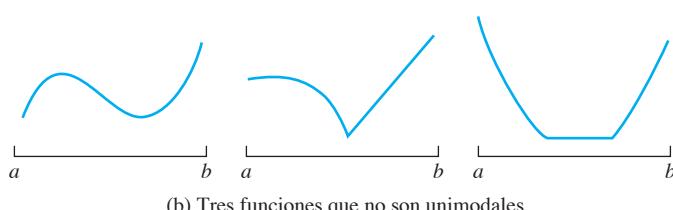
Funciones unimodales F

Al abordar un problema de minimización, una hipótesis razonable es que en algún intervalo $[a, b]$ dado por adelantado, F sólo tiene un mínimo local único. Esta propiedad con frecuencia se expresa diciendo que F es unimodal en $[a, b]$. (*Advertencia:* en estadística, *unimodal* se refiere a un máximo local único.) En la figura 16.3 se muestran algunas funciones unimodales.

Una propiedad importante de una función unimodal continua, que podría suponerse de la figura 16.3, es que es estrictamente decreciente hasta el punto mínimo y después estrictamente creciente.



(a) Tres funciones unimodales

**FIGURA 16.3**
Ejemplos de
funciones
unimodales y
no unimodales

Para convencerse de esto, sea x^* el punto mínimo de F en $[a, b]$ y suponga, por ejemplo, que F no es estrictamente decreciente en el intervalo $[a, x^*]$. Entonces deben existir los puntos x_1 y x_2 que satisfacen $a \leq x_1 < x_2 \leq x^*$ y $F(x_1) \leq F(x_2)$. Ahora sea x^{**} un punto mínimo de F en el intervalo $[a, x_2]$. (Recuerde que una función continua en un intervalo finito cerrado alcanza su valor mínimo.) Podemos suponer que $x^{**} \neq x_2$ porque si se eligiera x^{**} inicialmente igual a x_2 , se podría sustituir por x_1 , ya que $F(x_1) \leq F(x_2)$. Pero ahora vemos que x^{**} es un punto mínimo local de F en el intervalo $[a, b]$, porque este es un punto mínimo de F en $[a, x_2]$, pero no es x_2 misma. Por supuesto, la presencia de dos puntos mínimos locales contradice la unimodalidad de F .

Algoritmo de búsqueda de Fibonacci

Ahora nos planteamos un problema relativo a la búsqueda de un punto mínimo x^* de una función continua unimodal F en un intervalo $[a, b]$. *Con qué precisión se puede calcular el punto mínimo verdadero x^* con sólo n evaluaciones de F ?* Si evaluaciones de F , lo más que puede decirse es que $x^* \in [a, b]$; tomar al punto medio $\hat{x} = \frac{1}{2}(b + a)$ como el mejor cálculo da un error de $|x^* - \hat{x}| \leq \frac{1}{2}(b - a)$. Una sola evaluación no mejora esta situación, por lo que el mejor cálculo y el error son los mismos que en el caso anterior. Por consiguiente, se necesitan al menos dos evaluaciones de la función para obtener un mejor cálculo.

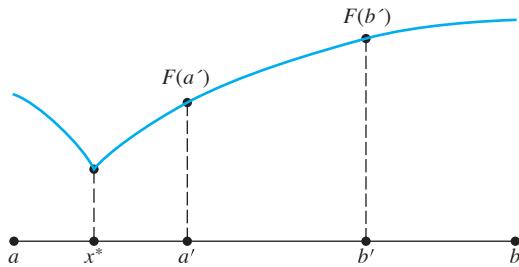


FIGURA 16.4
El algoritmo de búsqueda de Fibonacci: F evaluada en a' y en b'

Suponga que F se evalúa en a' y en b' con los resultados que se muestran en la figura 16.4. Si $F(a') < F(b')$, entonces, ya que F está cada vez más a la derecha de x^* , podemos estar seguros de que $x^* \in [a, b']$. Por otro lado, razonando similarmente para el caso $F(a') \geq F(b')$ muestra que $x^* \in [a', b]$. Para hacer ambos intervalos de incertidumbre lo más pequeños posible, movemos b' a la izquierda y a' a la derecha. Así, F se debe evaluar en dos puntos cercanos a cada lado del punto medio, como se muestra en la figura 16.5. Suponga que

$$a' = \frac{1}{2}(a + b) - 2\delta \quad \text{y} \quad b' = \frac{1}{2}(a + b) + 2\delta$$

Tomando el punto medio del subintervalo apropiado $[a, b']$ o $[a', b]$ como la mejor estimación de \hat{x} de x^* , encontramos que el error no excede a $\frac{1}{4}(b - a) + \delta$. El lector puede comprobar esto fácilmente.

Para $n = 3$, primero se hacen dos evaluaciones en los puntos $\frac{1}{3}$ y $\frac{2}{3}$ del intervalo inicial $[a, b]$, es decir,

$$a' = a + \frac{1}{3}(b - a) \quad \text{y} \quad b' = a + \frac{2}{3}(b - a)$$

De los dos valores $F(a')$ y $F(b')$, se puede determinar si $x^* \in [a, b']$ o $x^* \in [a', b]$. Los dos casos son, por supuesto, similares. Supongamos que $F(a') \geq F(b')$, por lo que nuestro punto

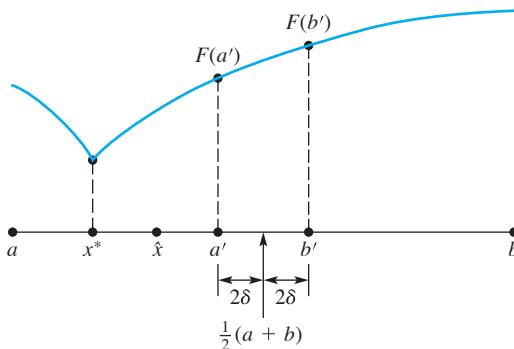


FIGURA 16.5
Algoritmo de búsqueda de Fibonacci: F evaluada a ambos lados del punto medio

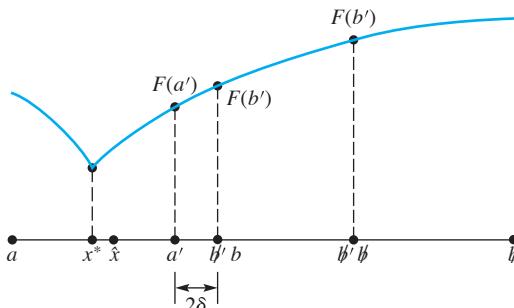


FIGURA 16.6
Algoritmo de búsqueda de Fibonacci:
Hace $b = b'$

mínimo x^* debe estar en $[a', b]$, como se muestra en la figura 16.6. La tercera (y última) evaluación se realiza cerca de b' , por ejemplo, en $b' + \delta$ (donde $\delta > 0$). Si $F(b') \geq F(b' + \delta)$, entonces $x^* \in [b', b]$. Tomando el punto medio de este intervalo, se obtiene $\hat{x} = \frac{1}{2}(b' + b)$ como nuestro cálculo de x^* y se encuentra que $|\hat{x} - x^*| \leq \frac{1}{6}(b - a)$. Por otra parte, si $F(b') < F(b' + \delta)$, entonces $x^* \in [a', b' + \delta]$. Una vez más tomamos el punto medio, $\hat{x} = \frac{1}{2}(a' + b' + \delta)$ y se encuentra que $|\hat{x} - x^*| \leq \frac{1}{6}(b - a) + \frac{1}{2}\delta$. Así, si despreciamos la pequeña cantidad $\delta/2$, nuestra precisión es $\frac{1}{6}(b - a)$ usando tres evaluaciones de F .

Continuando con el patrón de búsqueda indicado, encontramos un cálculo \hat{x} de x^* con sólo n evaluaciones de F y con un error que no es superior a

$$\frac{1}{2} \left(\frac{b - a}{\lambda_n} \right) \quad (1)$$

donde λ_n es el $(n + 1)$ -ésimo miembro de la **serie de Fibonacci**:

$$\begin{cases} \lambda_1 = 1, & \lambda_2 = 1 \\ \lambda_k = \lambda_{k-1} + \lambda_{k-2} & (k \geq 3) \end{cases} \quad (2)$$

Por ejemplo, los elementos λ_1 a λ_8 son 1, 1, 2, 3, 5, 8, 13 y 21.

En el **algoritmo de búsqueda de Fibonacci**, inicialmente determinamos el número de pasos N para una precisión deseada $\epsilon > \delta$ al seleccionar N como el subíndice del número más pequeño de Fibonacci mayor que $\frac{1}{2}(b - a)/\epsilon$. Se define una sucesión de intervalos, comenzando con el intervalo dado $[a, b]$ de longitud $\ell = b - a$, y para $k = N, N - 1, \dots, 3$, se usan estas fórmulas

de actualización:

$$\Delta = \left(\frac{\lambda_{k-2}}{\lambda_k} \right) (b - a) \quad (3)$$

$$a' = a + \Delta \quad b' = b - \Delta$$

$$\begin{cases} a = a' & \text{si } F(a') \geq F(b') \\ b = b' & \text{si } F(a') < F(b') \end{cases}$$

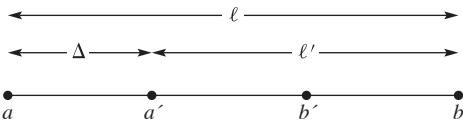
En el paso $k = 2$, hacemos

$$a' = \frac{1}{2}(a + b) - 2\delta \quad b' = \frac{1}{2}(a + b) + 2\delta$$

$$\begin{cases} a = a' & \text{si } F(a') \geq F(b') \\ b = b' & \text{si } F(a') < F(b') \end{cases}$$

y tenemos el intervalo final $[a, b]$, para el que se calcula $\hat{x} = \frac{1}{2}(a + b)$. Este algoritmo sólo requiere una evaluación de la función por paso después del primer paso.

FIGURA 16.7
Algoritmo de búsqueda de Fibonacci: comprobación con una situación típica



Para comprobar el algoritmo, considere la situación que se muestra en la figura 16.7. Puesto que $\lambda_k = \lambda_{k-1} + \lambda_{k-2}$, tenemos

$$\ell' = \ell - \Delta = \ell - \left(\frac{\lambda_{k-2}}{\lambda_k} \right) \ell = \left(\frac{\lambda_{k-1}}{\lambda_k} \right) \ell \quad (4)$$

y la longitud del intervalo de incertidumbre se ha reducido por el factor $(\lambda_{k-1}/\lambda_k)$. El siguiente paso produce

$$\Delta' = \left(\frac{\lambda_{k-3}}{\lambda_{k-1}} \right) \ell' \quad (5)$$

y Δ' es en realidad la distancia entre a' y b' . Por lo tanto, uno de los puntos anteriores en los que se evaluó la función está en un extremo o en el otro de $[a, b]$; es decir,

$$\begin{aligned} b' - a' &= \ell = 2\Delta = \left(\frac{\lambda_k - 2\lambda_{k-2}}{\lambda_k} \right) \ell \\ &= \left(\frac{\lambda_{k-1} - \lambda_{k-2}}{\lambda_k} \right) \ell = \left(\frac{\lambda_{k-3}}{\lambda_k} \right) \ell \\ &= \left(\frac{\lambda_{k-3}}{\lambda_{k-1}} \right) \ell' = \Delta' \end{aligned}$$

por las ecuaciones (2), (4) y (5).

Es evidente por la ecuación (4) que, después de $N - 1$ evaluaciones de la función, el penúltimo intervalo tiene una longitud $(1/\lambda_N)$ veces la longitud del intervalo inicial $[a, b]$. De modo que el intervalo final tiene $(b - a)(1/\lambda_N)$ de ancho y se ha establecido error máximo (1). El paso final es similar al descrito y F se evalúa en un punto que está a una distancia 2δ del punto medio del penúltimo intervalo. Finalmente, se hace $\hat{x} = \frac{1}{2}(b + a)$ a partir del último intervalo $[a, b]$.

Una desventaja de la búsqueda de Fibonacci es que el algoritmo es bastante complicado. Además, la precisión deseada se debe dar de antemano y el número de pasos que se calcula para esta precisión debe determinarse antes de empezar el cálculo. Así, los puntos de evaluación inicial de la función F dependen de N , el número de pasos.

Algoritmo de búsqueda de la sección áurea

A continuación se describe un algoritmo similar que está libre de estos inconvenientes. Se llama **búsqueda de la sección áurea** ya que depende de un cociente ρ conocido por los antiguos griegos como la **razón de la sección áurea**:

$$\rho = \frac{1}{2}(1 + \sqrt{5}) \approx 1.61803\ 39887$$

La historia matemática de este número puede encontrarse en Roger [1998] y ρ satisface la ecuación $\rho^2 = \rho + 1$, que tiene raíces $\frac{1}{2}(1 + \sqrt{5}) \approx 1.61803\dots$ y $\frac{1}{2}(1 - \sqrt{5}) \approx -0.61803\dots$. En cada paso de este algoritmo iterativo, hay disponible un intervalo $[a, b]$ del trabajo anterior. Es un intervalo que se sabe que tiene el punto mínimo x^* y nuestro objetivo es remplazarlo por un intervalo más pequeño que también se sabe que tiene a x^* . En cada paso se necesitan dos valores de F :

$$\begin{cases} x = a + r(b - a) & u = F(x) \\ y = a + r^2(b - a) & v = F(y) \end{cases} \quad (6)$$

donde $r = 1/\rho$, $r^2 + r = 1$, que tiene raíces $\frac{1}{2}(-1 + \sqrt{5}) \approx 0.61803$ y $\frac{1}{2}(-1 - \sqrt{5}) \approx -1.61803\dots$. Hay que considerar dos casos: ya sea que $u > v$ o que $u \leq v$. Tomamos el primero. La figura 16.8 muestra esta situación. Puesto que se supone que F es continua y unimodal, el mínimo de F debe estar en el intervalo $[a, x]$. Este intervalo es el intervalo de entrada al comienzo del siguiente paso. Observe ahora que en el intervalo $[a, x]$, hay una evaluación de F disponible, a saber, en y . También observe que

$$a + r(x - a) = y$$

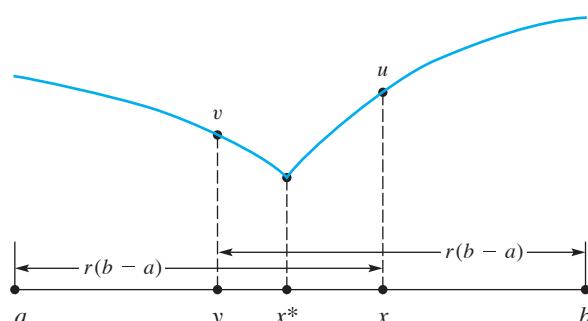


FIGURA 16.8
Algoritmo de
búsqueda de la
sección áurea:
 $u > v$

ya que $x - a = r(b - a)$. Por lo tanto, en el siguiente paso y desempeñará el papel de x y tendremos el valor de F en el punto $a + r^2(b - a)$.

Lo que se debe hacer en este paso es realizar sustituciones en el siguiente *orden*:

$$\begin{aligned} b &\leftarrow x \\ x &\leftarrow y \\ u &\leftarrow v \\ y &\leftarrow a + r^2(b - a) \\ v &\leftarrow F(y) \end{aligned}$$

El otro caso es similar. Si $u \leq v$, la imagen podría ser como en la figura 16.9. En este caso, el punto mínimo debe estar en $[y, b]$. Dentro de este intervalo, hay un valor de F disponible, a saber, en x . Observe que

$$y + r^2(b - y) = x$$

(véase el problema 16.1.9). Así, x ahora debe tener el papel de y y el valor de F se calcula en $y + r(b - y)$. Las sustituciones siguientes hacen esto en tal orden:

$$\begin{aligned} a &\leftarrow y \\ y &\leftarrow x \\ v &\leftarrow u \\ x &\leftarrow a + r(b - a) \\ u &\leftarrow F(x) \end{aligned}$$

Los problemas de 16.1.10 y 16.1.11 indican una deficiencia de este procedimiento: es muy lento. La lentitud en este contexto se refiere al gran número de evaluaciones de la función que se necesitan para lograr una precisión razonable. Se puede suponer que esta lentitud se debe a la extrema generalidad del algoritmo. No se ha aprovechado toda la suavidad que la función F puede tener.

Si $[a, b]$ es el intervalo inicial en la búsqueda de un mínimo de F , entonces, al principio, con una evaluación de F , podemos estar seguros de que sólo el punto mínimo, x^* , se encuentra en un intervalo de ancho $b - a$. En la búsqueda de la sección áurea, las longitudes correspondientes en los pasos sucesivos son $r(b - a)$ para dos evaluaciones de F , $r^2(b - a)$ para tres evaluaciones de F y así sucesivamente. Después de n pasos, el punto mínimo se ha inmovilizado en un intervalo de longitud $r^{n-1}(b - a)$. ¿Cómo se compara esto con el algoritmo de búsqueda de Fibonacci utilizando n evaluaciones? El ancho correspondiente del intervalo, en el último paso de este algoritmo, es $\lambda_n^{-1}(b - a)$. Ahora, el algoritmo de Fibonacci debe ser mejor, ya que está diseñado para hacer lo mejor posible con el número de pasos dado.

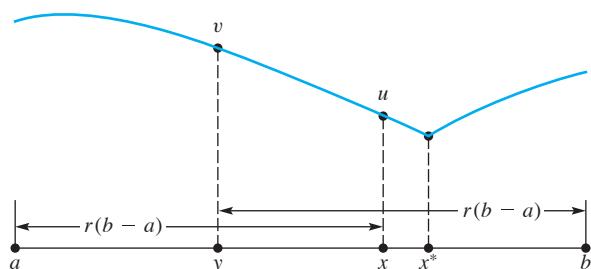
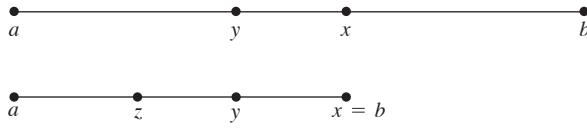


FIGURA 16.9

Algoritmo de búsqueda de la sección áurea:
 $u \leq v$

Así, esperamos que el cociente r^{n-1}/l_n^{-1} sea mayor que 1. Pero este tiende a 1.17 cuando $n \rightarrow \infty$ (véase el problema 16.1.8). Así, se puede concluir que la complejidad adicional del algoritmo de Fibonacci, junto con la desventaja de tener que el propio algoritmo dependa del número de evaluaciones permitidas, debilita su uso en general.

En el algoritmo de búsqueda de la sección áurea, ¿cómo se determina el cociente correcto r ? Recuerde que en el algoritmo cuando se pasa de un intervalo al siguiente, uno de los puntos x o y se conserva en el paso siguiente. Aquí presentamos un primer boceto del primer intervalo en el que hacemos $x = a + r(b - a)$ e $y = b + r(a - b)$. Esto seguido de un esquema del siguiente intervalo.



En este nuevo intervalo se deben conservar los mismos cocientes, por lo que tenemos $y = a + r(x - a)$. Como $x - a = r(b - a)$, podemos escribir $y = a + r[r(b - a)]$. Igualando una con otra las dos fórmulas para y se obtiene

$$a + r^2(b - a) = b + r(a - b)$$

donde

$$a - b + r^2(b - a) = r(a - b)$$

Dividiendo entre $(a - b)$ se obtiene

$$r^2 + r - 1 = 0$$

Las raíces de esta ecuación cuadrática son como se dieron antes.

Algoritmo de interpolación cuadrática

Supongamos que F se representa con una serie de Taylor en la vecindad del punto x^* . Entonces

$$F(x) = F(x^*) + (x - x^*)F'(x^*) + \frac{1}{2}(x - x^*)^2F''(x^*) + \dots$$

Puesto que x^* es un punto mínimo de F , tenemos $F'(x^*) = 0$. Por lo tanto,

$$F(x) \approx F(x^*) + \frac{1}{2}(x - x^*)^2F''(x^*)$$

Esta nos dice que, en la vecindad de x^* , $F(x)$ se aproxima con una función cuadrática cuyo mínimo está también en x^* . Puesto que no conocemos a x^* y no queremos implicar derivadas en nuestros algoritmos, un artificio natural es interpolar F mediante con un polinomio de segundo grado. Cualquier combinación de tres valores $(x_i, F(x_i))$, $i = 1, 2, 3$ se puede utilizar para este propósito. El punto mínimo de la función cuadrática resultante puede ser una mejor aproximación a x^* que x_1, x_2 o x_3 . Escribir un algoritmo que realice esta idea de forma iterativa no es trivial y se deben manejar muchos casos desagradables. ¿Qué debe hacerse si, por ejemplo, el polinomio cuadrático de interpolación tiene un máximo en lugar de un mínimo? También existe la posibilidad de que $F''(x^*) = 0$, en cuyo caso los términos de orden superior de la serie de Taylor determinan la naturaleza de F cerca de x^* .

Aquí se presenta el esquema de un algoritmo para este procedimiento. Al inicio, tenemos una función F cuyo mínimo se busca. Se dan dos puntos iniciales, x y y , así como dos

números de control δ y ε . El cálculo comienza al evaluar los dos números

$$\begin{cases} u = F(x) \\ v = F(y) \end{cases}$$

Ahora sea

$$z = \begin{cases} 2x - y & \text{si } u < v \\ 2y - x & \text{si } u \geq v \end{cases}$$

En cualquier caso, el número

$$w = F(z)$$

se va a calcular.

En este momento, tenemos tres puntos x , y y z , junto con los correspondientes valores de la función u , v y w . En el paso de la iteración principal del algoritmo, uno de estos puntos y el valor de su función acompañante se sustituyen por un nuevo punto y el nuevo valor de la nueva función. El proceso se repite hasta que se ha alcanzado el éxito o el fracaso.

En el cálculo principal, se determina un polinomio cuadrático q para interpolar a F en los tres puntos actuales x , y y z . Las fórmulas se analizan a continuación. Después, se determina el punto t , donde $q'(t) = 0$. Bajo circunstancias ideales, t es un punto *mínimo* de q y un punto *mínimo aproximado* de F . Así, uno de x , y o z se debe sustituir por t . Estamos interesados en analizar $q''(t)$ para determinar la forma de la curva q cerca de t .

Para la descripción completa de este algoritmo, se deben determinar las fórmulas para t y $q''(t)$. Se obtienen de la siguiente manera:

$$\begin{cases} a = \frac{v - u}{y - x} \\ b = \frac{w - v}{z - y} \\ c = \frac{b - a}{z - x} \\ t = \frac{1}{2} \left[x + y - \frac{a}{c} \right] \\ q''(t) = 2c \end{cases}$$

Su deducción se describe en el problema 16.1.12.

El **caso solución** se produce si

$$q''(t) > 0 \quad \text{y} \quad \max \{ |t - x|, |t - y|, |t - z| \} < \varepsilon$$

La condición $q''(t) > 0$ indica, por supuesto, que q' está *creciendo* en la vecindad de t , así que t es realmente un punto *mínimo* de q . La segunda condición indica que este cálculo, t , del punto *mínimo* de F está a una distancia ε de cada uno de los tres puntos x , y y z . En este caso, t se acepta como una solución.

El **caso habitual** se produce si

$$q''(t) > 0 \quad \text{y} \quad \delta \geq \max \{ |t - x|, |t - y|, |t - z| \} \geq \varepsilon$$

Estas desigualdades indican que t es un punto *mínimo* de q , pero no lo suficientemente cerca de los tres puntos iniciales para ser aceptado como una solución. También, t no está más allá de δ unidades de cada uno de x , y y z y puede aceptarse como un nuevo punto razonable. El punto viejo que tiene el mayor valor de la función ahora se sustituye por t , y su valor de la función por $F(t)$.

El **primer caso malo** se produce si

$$q''(t) > 0 \quad \text{y} \quad \max\{|t - x|, |t - y|, |t - z|\} > \delta$$

En este caso, t es un punto mínimo de q pero está tan lejos que hay cierto peligro en usarlo como un nuevo punto. Identificamos uno de los tres puntos originales que está más alejado de t , por ejemplo, x , y también identificamos el punto más cercano a t , por ejemplo, z . Entonces reemplazamos x por $z + \delta \operatorname{signo}(t - z)$ y u por $F(x)$. La figura 16.10 muestra este caso. La curva es la gráfica de q .

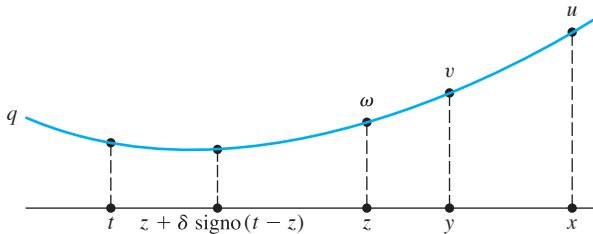


FIGURA 16.10
Algoritmo de la serie de Taylor:
primer caso malo

El **segundo caso malo** se produce si

$$q''(t) < 0$$

lo que indica que t es un punto máximo de q . En este caso, se identifica el mayor y el menor de u , v y w . Supongamos, por ejemplo, que $u \geq v \geq w$. Entonces se sustituye x por $z + \operatorname{signo}(z - x)$. En la figura 16.11 se muestra un ejemplo.

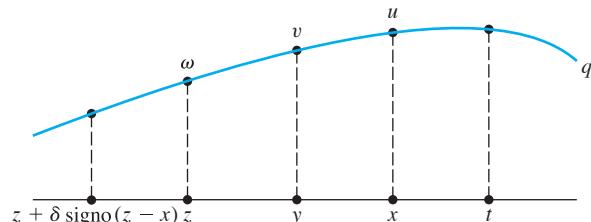


FIGURA 16.11
Algoritmo de la serie de Taylor:
segundo caso malo

Resumen

Consideraremos el problema de encontrar el **mínimo local** de una función unimodal de una sola variable. Los algoritmos que se analizan son **búsqueda de Fibonacci**, **búsqueda de la sección áurea** e **interpolación cuadrática**.

Problemas 16.1

1. Para la función $F(x_1, x_2, x_3) = x_1^2 + 3x_2^2 + 2x_3^2 - 4x_1 - 6x_2 + 8x_3$, encuentre el punto mínimo sin restricciones. Despues, encuentre el mínimo con restricciones en el conjunto K definido por las desigualdades $x_1 \leq 0, x_2 \leq 0$ y $x_3 \leq 0$. Despues, resuelva el mismo problema cuando se define K por $x_1 \leq 0, x_2 \leq 0$ y $x_3 \leq -2$.

“2. Para la función $F(x, y) = 13x^2 + 13y^2 - 10xy - 18x - 18y$, encuentre el mínimo sin restricciones.

Sugerencia: intente sustituyendo $x = u + v$ y $y = u - v$.

3. Si F es unimodal y continua en el intervalo $[a, b]$, ¿cuántos máximos locales pueden tener F en $[a, b]$?

“4. Para el algoritmo de búsqueda de Fibonacci, escriba expresiones para \hat{x} en los dos casos $n = 2, 3$.

5. Realice cuatro pasos del algoritmo de búsqueda de Fibonacci utilizando $\epsilon = \frac{1}{4}$ para determinar lo siguiente:

a. Mínimo de $F(x) = x^2 - 6x + 1$ en $[0, 10]$

b. Mínimo de $F(x) = 2x^3 - 9x^2 + 12x + 2$ en $[0, 3]$

c. Máximo de $F(x) = 2x^3 - 9x^2 + 12x$ en $[0, 2]$

6. Sea F una función unimodal continua definida en el intervalo $[a, b]$. Supongamos que los valores de F son conocidos en los n puntos, a saber, $a = t_1 < t_2 < \dots < t_n = b$. ¿Cómo se puede calcular con precisión el punto mínimo x^* a partir sólo de los valores de t_i y $F(t_i)$?

“7. La ecuación que satisface los números de Fibonacci, es decir, $\lambda_n - \lambda_{n-1} - \lambda_{n-2} = 0$, es un ejemplo de una ecuación de diferencias lineal con coeficientes constantes. Resuélvala haciendo $\lambda_n = \lambda^n$ y hallando que $\alpha = \frac{1}{2}(1 + \sqrt{5})$ o $\beta = \frac{1}{2}(1 - \sqrt{5})$ servirá para λ . Se pueden encontrar las condiciones iniciales $\lambda_1 = \lambda_2 = 1$ con una solución de la forma $\lambda_n = A\alpha^n + B\beta^n$. Encuentre A y B . Establezca

$$\lim_{n \rightarrow \infty} \left(\frac{\lambda_n}{\lambda_{n-1}} \right) = \alpha = \frac{1}{2}(1 + \sqrt{5})$$

Muestre que esto concuerda con las ecuaciones (10) y (11) de la sección 3.3.

8. (Continuación) Consulte el algoritmo de búsqueda de la sección áurea y el problema anterior. Demuestre que $\alpha\beta = -1$ y $\alpha + \beta = 1$, de modo que $\alpha = 1/r$ y $\beta = -r$. Después haga que $r^n\lambda^n$ converja a $1/\sqrt{5}$, cuando $n \rightarrow \infty$.

“9. Verifique que $y + r^2(b - y) = x$ en el algoritmo de la sección áurea. *Sugerencia:* utilice $r^2 + r = 1$.

“10. Si F es unimodal en un intervalo de longitud ℓ , ¿cuántas evaluaciones son necesarias en el algoritmo de la sección áurea para calcular el punto mínimo con un error a lo más de 10^{-k} ?

“11. (Continuación) En el problema anterior, ¿cuán grande debe ser n si $\ell = 1$ y $k = 10$?

12. Utilizando el algoritmo de la diferencia dividida en la tabla

x	y	z
u	v	w

muestre que la interpolación cuadrática en la forma de Newton es

$$q(t) = u + a(t - x) + c(t - x)(t - y)$$

con a, b, c dadas por la ecuación (7). Después compruebe las fórmulas para t y $q''(t)$ dadas en (7).

“13. Si las rutinas se pueden escribir fácilmente para F, F' y F'' , ¿cómo se puede utilizar el método de Newton para localizar el punto mínimo de F ? Escriba la fórmula que define el proceso iterativo. ¿Implica a F ?

“14. Si hay rutinas para F y F' , ¿cómo se puede usar el método de la secante para minimizar F ?

15. El cociente de la sección áurea, $\rho = \frac{1}{2}(1 + \sqrt{5})$, tiene muchas propiedades místicas, por ejemplo,

$$\text{a. } \rho = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}}}$$

$$\text{c. } \rho^n = \rho^{n-1} + \rho^{n-2}$$

$$\text{a. } \rho = \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \cdots}}}}$$

$$\text{d. } \rho = \rho^{-1} + \rho^{-2} + \rho^{-3} + \cdots$$

Establezca estas propiedades.

16. (Opción múltiple) En el algoritmo de búsqueda de la sección áurea, se usa un número $r = 0.618 \dots$, que es la mayor de las dos raíces de la ecuación cuadrática $r^2 + r - 1 = 0$. Sea f una función unimodal en el intervalo $[a, b]$. Por lo tanto, f tiene un mínimo local en $[a, b]$, en este caso, suponemos que $a < b$. Sea $x = a + r(b - a)$ y $y = a + r^2(b - a)$. También, sea $u = f(x)$ y $v = f(y)$, donde se supone que $u < v$. ¿Qué intervalo debe contener el punto mínimo de f ?

- a.** $[y, b]$ **b.** $[a, x]$ **c.** $[a, y]$ **d.** $[y, x]$ **e.** Ninguno de estos.

Problemas de cómputo 16.1

1. Escriba una rutina para realizar el algoritmo de la sección áurea para una función y el intervalo dados. La búsqueda debe continuar hasta que se alcance un error acotado preasignado, pero no realice más de 100 pasos en cualquier caso.

2. (Continuación) Pruebe la rutina del problema de cómputo anterior con estos ejemplos o use una rutina de un paquete como Matlab, Maple o Mathematica:

a. $F(x) = \sin x$	en $[0, \pi/2]$	b. $F(x) = (\arctan x)^2$	en $[-1, 1]$
c. $F(x) = \ln x $	en $[\frac{1}{2}, 4]$	d. $F(x) = x $	en $[-1, 1]$

3. Codifique y pruebe el siguiente algoritmo para la aproximación de los mínimos de una función F de una variable en un intervalo $[a, b]$. El algoritmo define una sucesión de cuádruples $a < a' < b' < b$ haciendo inicialmente $a' = \frac{2}{3}a + \frac{1}{3}b$ y $b' = \frac{1}{3}a + \frac{2}{3}b$ y actualizando repetidas veces mediante $a = a'$, $a' = b'$ y $b' = \frac{1}{2}(b + b')$ si $F(a') > F(b')$; $b = b'$, $a' = \frac{1}{2}(a + a')$ y $b' = a$ si $F(a') < F(b')$; $a = a'$, $b = b'$, $a' = \frac{2}{3}a + \frac{1}{3}b$ y $b' = \frac{1}{3}a + \frac{2}{3}b$ si $F(a') = F(b')$. Nota: la construcción asegura que $a < a' < b' < b$, y el mínimo de F siempre se produce entre a y b . Además, sólo se necesita calcular un nuevo valor de la función en cada etapa del cálculo después del primer caso, a menos de que se obtenga el caso $F(a') = F(b')$. Los valores de a , a' , b' y b tienden al mismo límite, que es un punto mínimo de F . Observe la similitud con el método de biseción de la sección 3.1.

4. Escriba y pruebe rutinas para el algoritmo de búsqueda de Fibonacci. Compruebe que un algoritmo parcial para la búsqueda de Fibonacci es como sigue: inicialmente, se hace

$$\begin{aligned}\Delta &= \left(\frac{\lambda_{N-2}}{\lambda_N} \right) (b - a) \\ a' &= a + \Delta \\ b' &= b - \Delta \\ u &= F(a') \\ v &= F(b')\end{aligned}$$

Después se hace un ciclo en k desde $N - 1$ bajando hasta 3, y se actualiza de la siguiente manera:

$$\begin{array}{ll} \text{Si } u \geq v: & \text{Si } v > u: \\ \left\{ \begin{array}{l} a \leftarrow a' \\ a' \leftarrow b' \\ u \leftarrow v \\ \Delta \leftarrow \left(\frac{\lambda_{k-2}}{\lambda_k} \right) (b - a) \\ b' \leftarrow b - \Delta \\ v \leftarrow F(b') \end{array} \right. & \left\{ \begin{array}{l} b \leftarrow b' \\ b' \leftarrow a' \\ v \leftarrow u \\ \Delta \leftarrow \left(\frac{\lambda_{k-2}}{\lambda_k} \right) (b - a) \\ a' \leftarrow a + \Delta \\ u \leftarrow F(a') \end{array} \right. \end{array}$$

Añada pasos para $k = 2$.

5. (**Algoritmo de Berman**) Supongamos que F es unimodal en $[a, b]$. Entonces, si x_1 y x_2 son dos puntos tales que $a \leq x_1 < x_2 \leq b$, tenemos que

$$F(x_1) > F(x_2) \quad \text{implica} \quad x^* \in (x_1, b]$$

$$F(x_1) = F(x_2) \quad \text{implica} \quad x^* \in [x_1, x_2]$$

$$F(x_1) < F(x_2) \quad \text{implica} \quad x^* \in [a, x_2)$$

Así, evaluando F en x_1 y en x_2 y comparando con los valores de la función, podemos reducir el tamaño del intervalo que sabemos que contienen a x^* . El método más simple es comenzar en el punto medio $x_0 = \frac{1}{2}(a + b)$ y si F está, por ejemplo, decreciendo para $x > x_0$, probamos F en $x_0 + ih$, $i = 1, 2, \dots, q$, con $h = (b - a)/2q$, hasta que encontramos un punto x_1 desde el cual F empieza a aumentar de nuevo (o hasta llegar a, b). Después repita este procedimiento iniciando en x_1 y use una menor longitud de paso h/q . Aquí, q es el número máximo de evaluaciones en cada paso, digamos, 4. Escriba una subrutina para ejecutar el algoritmo de Berman y pruébelo para evaluar la minimización aproximada de funciones de una sola dimensión. *Nota:* el número total de evaluaciones de F necesarias para ejecutar este algoritmo hasta algún paso iterativo k depende de la ubicación de x^* . Si, por ejemplo, $x^* = b$, entonces es claro que necesitamos q evaluaciones en cada iteración y, por tanto, kq evaluaciones. Este número decrecerá cuanto más cerca esté x^* de x_0 , y se puede demostrar que con $q = 4$, el número *esperado* de evaluaciones es tres por paso. Es interesante comparar la eficacia del algoritmo de Berman ($q = 4$) con la del algoritmo de búsqueda de Fibonacci. El número esperado de evaluaciones por paso es tres y el intervalo de incertidumbre disminuye en un factor de $4^{-1/3} \approx 0.63$ por evaluación. En comparación, el algoritmo de búsqueda de Fibonacci tiene un factor de reducción

de $\frac{1}{2}(1 + \sqrt{5}) \approx 0.62$. Por supuesto, el factor 0.63 en el algoritmo de Berman representa sólo un promedio y puede ser considerablemente más bajo pero también tan grande como $4^{-1/4} \approx 0.87$.

6. Seleccione una rutina de su biblioteca de programas o de un paquete como Matlab, Maple o Mathematica para encontrar el punto mínimo de una función de una variable. Experimente con la función $F(x) = x^4 + \operatorname{sen}(23x)$ para determinar si esta rutina encuentra alguna dificultad en hallar un punto mínimo global. Utilice los valores iniciales, tan cerca y tan lejos del punto mínimo global (véase la figura 16.2).
7. (**Proyecto estudiantil**) El matemático griego Euclides de Alejandría (325–265 a.C.) escribió una colección de 13 libros acerca de matemáticas y geometría. En el libro seis, la proposición 30 muestra cómo dividir una recta en su media y su media extrema, que es encontrar el punto de la sección áurea en una recta. Esto indica que la razón de la parte más pequeña de un segmento de recta a la mayor parte es la misma que la razón de la mayor parte de la serie de secciones a toda la recta. Para un segmento de recta de longitud 1, r representa la parte más grande y $1 - r$ la parte más pequeña, como aquí se muestra:



Por lo tanto, tenemos las razones

$$\frac{1-r}{r} = \frac{r}{1}$$

y obtenemos la ecuación cuadrática

$$r^2 = 1 - r$$

Esta ecuación tiene dos raíces, una positiva y la otra negativa. El recíproco de la raíz positiva es la **razón áurea** $\frac{1}{2}(1 + \sqrt{5})$, que fue de interés para Pitágoras (580–500 a. C.). También se usó en la construcción de la Gran Pirámide de Giza. Sistemas de software matemático como Matlab, Maple o Mathematica tienen la constante de la razón áurea. De hecho, la razón entre ancho y altura que se toma para trazar la gráfica de la función es la razón áurea. Investigue la razón de la sección áurea y su uso en computación científica.

8. Usando un sistema de software matemático como Matlab, Maple o Mathematica, escriba un programa de cómputo para reproducir la
 - a. Figura 16.1.
 - b. Figura 16.2. También, encuentre el mínimo global de la función, así como varios puntos mínimos locales cerca del origen.

16.2 Caso de variables múltiples

Ahora consideraremos una función de n variables reales $F: \mathbb{R}^n \rightarrow \mathbb{R}$. Como antes, se busca un punto x^* tal que

$$F(x^*) \leq F(x) \quad \text{para toda } x \in \mathbb{R}^n$$

Se debe desarrollar algo de teoría de funciones de múltiples variables para entender los algoritmos de minimización más refinados de uso actual.

Series de Taylor para F : vector gradiente y matriz hessiana

Si la función F tiene derivadas parciales de ciertos órdenes bajos (que generalmente se supone en el desarrollo de estos algoritmos), entonces en cualquier punto dado \mathbf{x} , un vector gradiente $\mathbf{G}(\mathbf{x}) = (G_i)_n$ se define con las componentes

$$G_i = G_i(\mathbf{x}) = \frac{\partial F(\mathbf{x})}{\partial x_i} \quad (1 \leq i \leq n) \quad (1)$$

y una **matriz hessiana** $\mathbf{H}(\mathbf{x}) = (H_{ij})_{n \times n}$ se define con las componentes

$$H_{ij} = H_{ij}(\mathbf{x}) = \frac{\partial^2 F(\mathbf{x})}{\partial x_i \partial x_j} \quad (1 \leq i, j \leq n) \quad (2)$$

Interpretamos $\mathbf{G}(\mathbf{x})$ como un vector de n componentes y $\mathbf{H}(\mathbf{x})$ como una matriz de $n \times n$, ambas en función de \mathbf{x} .

Utilizando el gradiente y la hessiana, se pueden escribir los primeros términos de la serie de Taylor para F como

$$F(\mathbf{x} + \mathbf{h}) = F(\mathbf{x}) + \sum_{i=1}^n G_i(\mathbf{x})h_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n h_i H_{ij}(\mathbf{x})h_j + \dots \quad (3)$$

La ecuación (3) también se puede escribir en forma de una elegante matriz vectorial:

$$F(\mathbf{x} + \mathbf{h}) = F(\mathbf{x}) + \mathbf{G}(\mathbf{x})^T \mathbf{h} + \frac{1}{2} \mathbf{h}^T \mathbf{H}(\mathbf{x}) \mathbf{h} + \dots \quad (4)$$

Aquí, \mathbf{x} es el punto fijo de expansión en \mathbb{R}^n y \mathbf{h} es la variable en \mathbb{R}^n con componentes, h_1, h_2, \dots, h_n . Los tres puntos indican los términos de orden superior en \mathbf{h} que no son necesarios en este análisis.

Un resultado en cálculo establece que el *orden* en que se toman las derivadas parciales no es importante si todas las derivadas parciales que se producen son continuas. En el caso particular de la matriz hessiana, si las segundas derivadas parciales de F son todas continuas, entonces \mathbf{H} es una **matriz simétrica**, es decir, $\mathbf{H} = \mathbf{H}^T$, ya que

$$H_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j} = \frac{\partial^2 F}{\partial x_j \partial x_i} = H_{ji}$$

EJEMPLO 1 Para ilustrar la fórmula (4), calculemos los tres primeros términos de la serie de Taylor para la función

$$F(x_1, x_2) = \cos(\pi x_1) + \sin(\pi x_2) + e^{x_1 x_2}$$

tomando $(1, 1)$ como el punto de expansión.

Solución Las derivadas parciales son

$$\begin{aligned}\frac{\partial F}{\partial x_1} &= -\pi \operatorname{sen}(\pi x_1) + x_2 e^{x_1 x_2} & \frac{\partial F}{\partial x_2} &= \pi \cos(\pi x_2) + x_1 e^{x_1 x_2} \\ \frac{\partial^2 F}{\partial x_1^2} &= -\pi^2 \cos(\pi x_1) + x_2^2 e^{x_1 x_2} & \frac{\partial^2 F}{\partial x_2 \partial x_1} &= (x_1 x_2 + 1) e^{x_1 x_2} \\ \frac{\partial^2 F}{\partial x_1 \partial x_2} &= (x_1 x_2 + 1) e^{x_1 x_2} & \frac{\partial^2 F}{\partial x_2^2} &= -\pi^2 \operatorname{sen}(\pi x_2) + x_1^2 e^{x_1 x_2}\end{aligned}$$

Observe la igualdad de las derivadas cruzadas, es decir, $\partial^2 F / \partial x_1 \partial x_2 = \partial^2 F / \partial x_2 \partial x_1$. En el punto específico $\mathbf{x} = [1, 1]^T$, tenemos

$$F(\mathbf{x}) = -1 + e, \quad \mathbf{G}(\mathbf{x}) = \begin{bmatrix} e \\ -\pi + e \end{bmatrix}, \quad \mathbf{H}(\mathbf{x}) = \begin{bmatrix} \pi^2 + e & 2e \\ 2e & e \end{bmatrix}$$

Así, por la ecuación (4),

$$\begin{aligned}F(1 + h_1, 1 + h_2) &= -1 + e + [e, -\pi + e] \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \\ &\quad + \frac{1}{2}[h_1, h_2] \begin{bmatrix} \pi^2 + e & 2e \\ 2e & e \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \dots\end{aligned}$$

o de manera equivalente, por la ecuación (3)

$$\begin{aligned}F(1 + h_1, 1 + h_2) &= -1 + e + eh_1 + (-\pi + e)h_2 \\ &\quad + \frac{1}{2}[(\pi^2 + e)h_1^2 + (2e)h_1h_2 + (2e)h_2h_1 + eh_2^2] + \dots\end{aligned}$$

■

En los sistemas de software matemático como Mathematica o Maple, se pueden comprobar estos cálculos utilizando rutinas integradas para el gradiente y la hessiana. Además, podemos obtener dos términos en la serie de Taylor en dos variables desarrollada con respecto al punto $(1, 1)$ y luego hacer un cambio de variable para obtener resultados similares a los anteriores.

Forma alternativa de la serie de Taylor

Otra forma de la serie de Taylor es útil. Primero sea \mathbf{z} el punto de expansión y después sea $\mathbf{h} = \mathbf{x} - \mathbf{z}$. Ahora, de la ecuación (4)

$$F(\mathbf{x}) = F(\mathbf{z}) + \mathbf{G}(\mathbf{z})^T(\mathbf{x} - \mathbf{z}) + \frac{1}{2}(\mathbf{x} - \mathbf{z})^T \mathbf{H}(\mathbf{z})(\mathbf{x} - \mathbf{z}) + \dots \quad (5)$$

Se ilustra con dos tipos especiales de funciones.

Primero, la **función lineal** tiene la forma

$$F(\mathbf{x}) = c + \sum_{i=1}^n b_i x_i = c + \mathbf{b}^T \mathbf{x}$$

con coeficientes adecuados c, b_1, b_2, \dots, b_n . Evidentemente, el gradiente y la hessiana son $G_i(z) = b_i$ y $H_{ij}(z) = 0$, por lo que la ecuación (5) produce

$$F(\mathbf{x}) = F(\mathbf{z}) + \sum_{i=1}^n b_i(x_i - z_i) = F(\mathbf{z}) + \mathbf{b}^T(\mathbf{x} - \mathbf{z})$$

Segundo, considere una **función cuadrática** general. Por simplicidad, tenemos sólo dos variables. La forma de la función es

$$F(x_1, x_2) = c + (b_1 x_1 + b_2 x_2) + \frac{1}{2}(a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2) \quad (6)$$

que puede interpretarse como la serie de Taylor de F cuando el punto de expansión es $(0, 0)$. Para comprobar esta afirmación se deben calcular las derivadas parciales y evaluar en $(0, 0)$:

$$\begin{aligned}\frac{\partial F}{\partial x_1} &= b_1 + a_{11}x_1 + a_{12}x_2 & \frac{\partial F}{\partial x_2} &= b_2 + a_{22}x_2 + a_{12}x_1 \\ \frac{\partial^2 F}{\partial x_1^2} &= a_{11} & \frac{\partial^2 F}{\partial x_1 \partial x_2} &= a_{12} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} &= a_{12} & \frac{\partial^2 F}{\partial x_2^2} &= a_{22}\end{aligned}$$

Haciendo $\mathbf{z} = [0, 0]^T$, se obtiene de la ecuación (5)

$$F(\mathbf{x}) = c + [b_1, b_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \frac{1}{2}[x_1, x_2] \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Esta es la forma de matriz de la función cuadrática original de dos variables. También se puede escribir como

$$F(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \quad (7)$$

donde c es un escalar, \mathbf{b} es un vector y \mathbf{A} una matriz. La ecuación (7) es válida para una función cuadrática general de n variables, con un vector \mathbf{b} de n componentes y una matriz \mathbf{A} de $n \times n$.

Volviendo a la ecuación (3), ahora escribimos la complicada doble sumatoria con todo detalle para ayudar a comprenderla:

$$\begin{aligned}\mathbf{x}^T \mathbf{H} \mathbf{x} &= \sum_{i=1}^n \sum_{j=1}^n x_i H_{ij} x_j = \left\{ \begin{array}{l} \sum_{j=1}^n x_1 H_{1j} x_j \\ + \sum_{j=1}^n x_2 H_{2j} x_j \\ + \cdots \\ + \sum_{j=1}^n x_n H_{nj} x_j \end{array} \right\} \\ &= \left\{ \begin{array}{l} x_1 H_{11} x_1 + x_1 H_{12} x_2 + \cdots + x_1 H_{1n} x_n \\ + x_2 H_{21} x_1 + x_2 H_{22} x_2 + \cdots + x_2 H_{2n} x_n \\ + \cdots \qquad \qquad \qquad + \cdots \\ + x_n H_{n1} x_1 + x_n H_{n2} x_2 + \cdots + x_n H_{nn} x_n \end{array} \right\}\end{aligned}$$

Por lo tanto, $\mathbf{x}^T \mathbf{H} \mathbf{x}$ se puede interpretar como la suma de todos los n^2 términos de una matriz cuadrada de la cual el elemento (i, j) es $x_i H_{ij} x_j$.

Procedimiento de máxima pendiente

Una característica esencial del vector gradiente $\mathbf{G}(\mathbf{x})$ es que apunta en la dirección de más rápido crecimiento de la función F , que es la dirección de la **pendiente más pronunciada**. Por lo contrario, $-\mathbf{G}(\mathbf{x})$ apunta en la dirección de **máxima pendiente**. Este hecho es tan importante que valen la pena algunas palabras de justificación. Supongamos que \mathbf{h} es un vector unitario, $\sum_{i=1}^n h_i^2 = 1$. La razón de cambio de F (en \mathbf{x}) en la dirección \mathbf{h} se define naturalmente por

$$\frac{d}{dt} F(\mathbf{x} + t\mathbf{h}) \Big|_{t=0}$$

Esta razón de cambio se puede evaluar usando la ecuación (4). De esa ecuación se deduce que

$$F(\mathbf{x} + t\mathbf{h}) = F(\mathbf{x}) + t\mathbf{G}(\mathbf{x})^T \mathbf{h} + \frac{1}{2}t^2 \mathbf{h}^T \mathbf{H}(\mathbf{x}) \mathbf{h} + \dots \quad (8)$$

Derivando con respecto a t se obtiene

$$\frac{d}{dt} F(\mathbf{x} + t\mathbf{h}) = \mathbf{G}(\mathbf{x})^T \mathbf{h} + t\mathbf{h}^T \mathbf{H}(\mathbf{x}) \mathbf{h} + \dots \quad (9)$$

Haciendo aquí $t = 0$, vemos que la razón de cambio de F en la dirección \mathbf{h} no es más que

$$\mathbf{G}(\mathbf{x})^T \mathbf{h}$$

Ahora nos preguntamos: *¿para qué vector unitario \mathbf{h} la razón de cambio es un máximo?* El camino más simple para la respuesta es invocar la poderosa **desigualdad de Cauchy-Schwarz**:

$$\sum_{i=1}^n u_i v_i \leq \left(\sum_{i=1}^n u_i^2 \right)^{1/2} \left(\sum_{i=1}^n v_i^2 \right)^{1/2} \quad (10)$$

donde la igualdad sólo es válida si uno de los vectores \mathbf{u} o \mathbf{v} es un múltiplo negativo del otro. Aplicando esto a

$$\mathbf{G}(\mathbf{x})^T \mathbf{h} = \sum_{i=1}^n G_i(\mathbf{x}) h_i$$

y recordando que $\sum_{i=1}^n h_i^2 = 1$, se concluye que el máximo se produce cuando \mathbf{h} es un múltiplo positivo de $\mathbf{G}(\mathbf{x})$, es decir, cuando \mathbf{h} apunta en la dirección de \mathbf{G} .

Con base en el análisis anterior, se puede describir un procedimiento de minimización llamado el **mejor paso para el método de máxima pendiente**. En cualquier punto dado \mathbf{x} , se calcula el vector gradiente $\mathbf{G}(\mathbf{x})$. Entonces se resuelve un problema de minimización unidimensional al determinar el valor de t^* para el que la función

$$\phi(t) = F(\mathbf{x} + t\mathbf{G}(\mathbf{x}))$$

sea un mínimo. Despues se remplaza \mathbf{x} por $\mathbf{x} + t^* \mathbf{G}(\mathbf{x})$ y se empieza de nuevo.

El método general de máxima pendiente da un paso de cualquier tamaño en la dirección del gradiente negativo. Normalmente, no es competitivo con otros métodos, pero tiene la ventaja de la simplicidad. En el problema de cómputo 16.2.2 se describe una forma de acelerarlo.

Diagramas de contorno

Para comprender cómo trabajan estos métodos con funciones de dos variables, con frecuencia es útil dibujar diagramas de contorno o curvas de nivel. Un **contorno** de una función F es un conjunto de la forma

$$\{\mathbf{x} : F(\mathbf{x}) = c\}$$

donde c es una constante. Por ejemplo, los contornos de la función

$$F(\mathbf{x}) = 25x_1^2 + x_2^2$$

son elipses, como se muestra en la figura 16.12. Los contornos también se llaman **curvas de nivel** por algunos autores. En cualquier punto del contorno, el gradiente de F es perpendicular a la curva. Así, en general, la trayectoria de máxima pendiente puede parecerse a la figura 16.13.

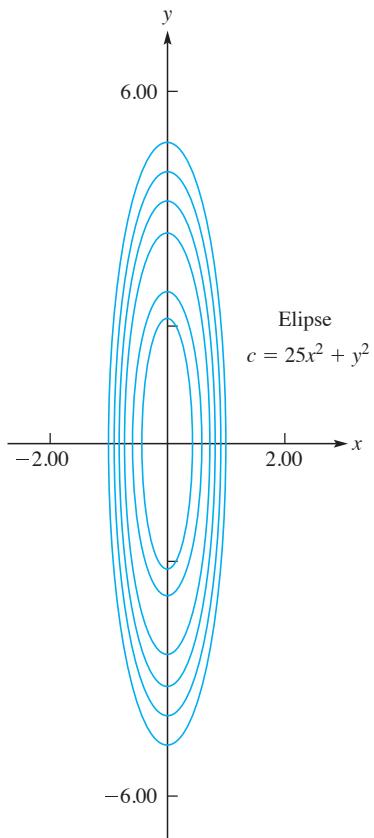


FIGURA 16.12
Contornos de
 $F(\mathbf{x}) = 25x_1^2 + x_2^2$

Algoritmos más avanzados

Para explicar los algoritmos más avanzados, consideramos una función F en general de valor real de n variables. Supongamos que tenemos los tres primeros términos de la serie de Taylor de F en la vecindad de un punto \mathbf{z} . ¿Cómo se pueden utilizar para intuir el punto mínimo de F ? Obviamente,

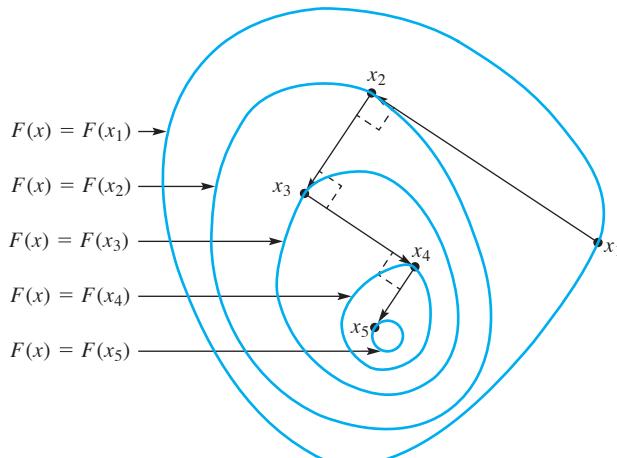


FIGURA 16.13
Trayectoria
de máxima
pendiente

podemos ignorar todos los términos más grandes que los cuadráticos y encontrar el mínimo de la función cuadrática resultante:

$$F(\mathbf{x} + \mathbf{z}) = F(\mathbf{z}) + \mathbf{G}(\mathbf{z})^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{H}(\mathbf{z}) \mathbf{x} + \dots \quad (11)$$

Aquí, \mathbf{z} está fija y \mathbf{x} es la variable. Para encontrar el mínimo de esta función cuadrática de \mathbf{x} , debemos calcular las primeras derivadas parciales y hacerlas igual a cero. Denotando esta función cuadrática por Q y simplificando un poco la notación, tenemos

$$Q(\mathbf{x}) = F(\mathbf{z}) + \sum_{i=1}^n G_i x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n x_i H_{ij} x_j \quad (12)$$

de lo que se deduce que

$$\frac{\partial Q}{\partial x_k} = G_k + \sum_{j=1}^n H_{kj} x_j \quad (1 \leq k \leq n) \quad (13)$$

(véase el problema 16.2.13). El punto \mathbf{x} que se busca es, por lo tanto, una solución del sistema de n ecuaciones

$$\sum_{j=1}^n H_{kj} x_j = -G_k \quad (1 \leq k \leq n)$$

o, equivalentemente,

$$\mathbf{H}(\mathbf{z}) \mathbf{x} = -\mathbf{G}(\mathbf{z}) \quad (14)$$

El análisis anterior sugiere el procedimiento iterativo siguiente para localizar un punto mínimo de una función F . Se inicia con un punto \mathbf{z} que es una estimación actual del punto mínimo. Se calcula el gradiente y la hessiana de F en el punto \mathbf{z} . Se pueden denotar por \mathbf{G} y \mathbf{H} , respectivamente. Por supuesto, \mathbf{G} es un vector de n componentes de números y \mathbf{H} es una matriz de números de $n \times n$. Despues se resuelve la ecuación matricial

$$\mathbf{H} \mathbf{x} = -\mathbf{G}$$

con lo que se obtiene un vector de n componentes \mathbf{x} . Se remplaza z por $z + \mathbf{x}$ y se regresa al inicio del procedimiento.

Mínimo, máximo y puntos silla

Hay muchas razones para esperar problemas en el procedimiento iterativo que acabamos de esbozar. Un aspecto especialmente nocivo es que podemos esperar encontrar un punto en el que sólo las primeras derivadas parciales de F se hacen cero; no es necesario que sean un punto mínimo. Es lo que llamamos un **punto estacionario**. Estos puntos se pueden clasificar en tres tipos: **punto mínimo**, **punto máximo** y **punto silla**. Se puede ilustrar con simples superficies cuadráticas familiares de geometría analítica:

- Mínimo de $F(x, y) = x^2 + y^2$ en $(0, 0)$ (figura 16.14(a)).
- Máximo de $F(x, y) = 1 - x^2 - y^2$ en $(0, 0)$ (figura 16.14(b)).
- Punto silla de $F(x, y) = x^2 - y^2$ en $(0, 0)$ (figura 16.14(c)).

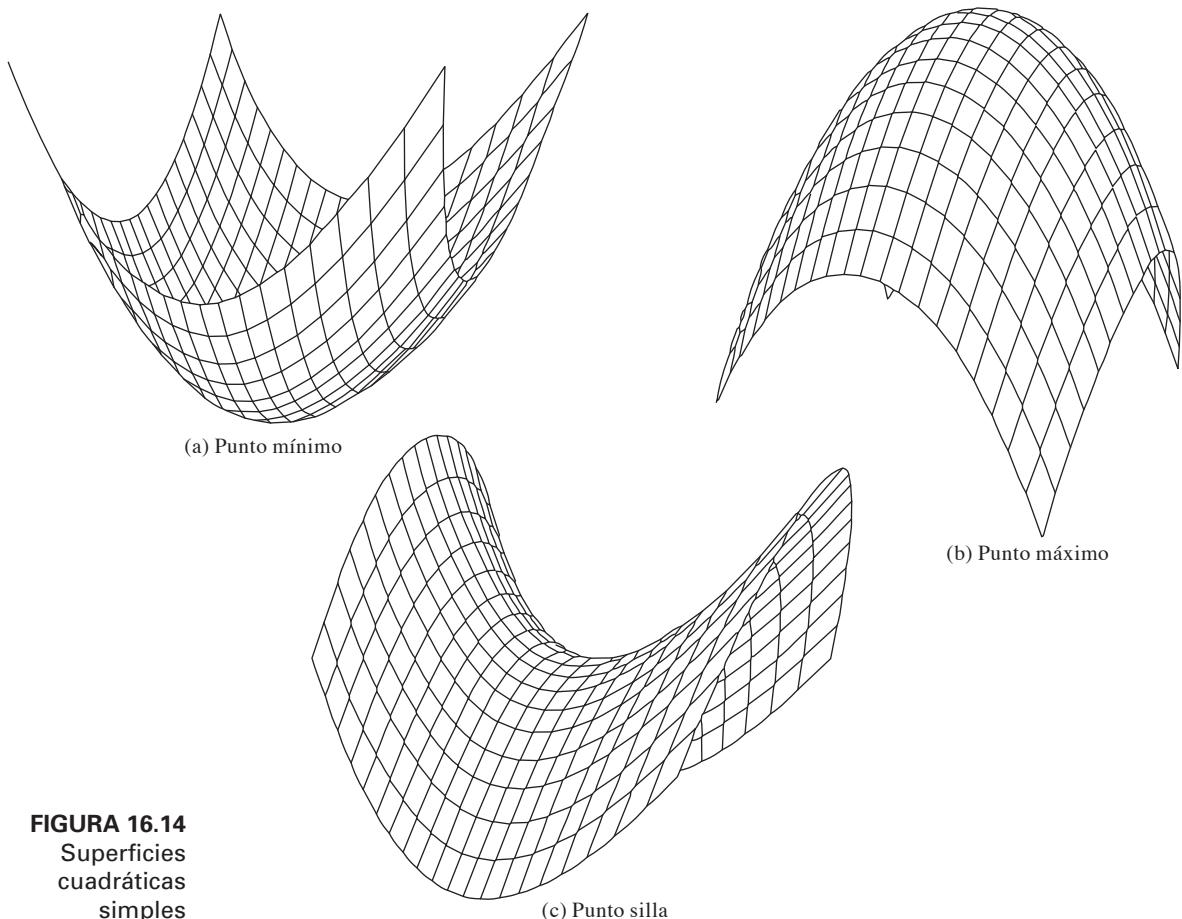


FIGURA 16.14
Superficies cuadráticas simples

Matriz positiva definida

Si z es un punto estacionario de F , entonces

$$\mathbf{G}(z) = 0$$

Por otra parte, un criterio que asegura que Q , como se define en la ecuación (12), tiene un punto mínimo es el siguiente:

TEOREMA 1

Teorema de la función cuadrática

Si la matriz \mathbf{H} tiene la propiedad de $\mathbf{x}^T \mathbf{H} \mathbf{x} > 0$ para cada vector \mathbf{x} distinto de cero, entonces la función cuadrática Q tiene un punto mínimo.

(Véase el problema 16.2.15). Una matriz que tiene esta propiedad se dice que es **definida positiva**. Observe que este teorema sólo implica términos de segundo grado en la función cuadrática Q .

Como ejemplos de funciones cuadráticas que no tienen mínimos, considere las siguientes:

$$\begin{array}{ll} -x_1^2 - x_2^2 + 13x_1 + 6x_2 + 12 & x_1^2 - x_2^2 + 3x_1 + 5x_2 + 7 \\ x_1^2 - 2x_1x_2 + x_1 + 2x_2 + 3 & 2x_1 + 4x_2 + 6 \end{array}$$

En los dos primeros ejemplos, sea $x_1 = 0$ y $x_2 \rightarrow \infty$. En el tercero, sea $x_1 = x_2 \rightarrow \infty$. En el último, sea $x_1 = 0$ y $x_2 \rightarrow -\infty$. En cada caso, los valores de la función tienden a $-\infty$, y no puede existir un mínimo global.

Métodos de quasiNewton

Los algoritmos que en general convergen más rápido que el de máxima pendiente y se recomiendan actualmente para la minimización son del tipo llamado **cuasiNewton**. El principal ejemplo es un algoritmo introducido en 1959 por Davidon, llamado el **algoritmo de métrica variable**. Posteriormente, se hicieron cambios y mejoras significativas por otros, como R. Fletcher, M. J. D. Powell, C. G. Broyden, P. E. Gill y W. Murray. Estos algoritmos trabajan en forma iterativa, suponiendo que en cada paso se conoce una aproximación cuadrática local para la función F cuyo mínimo se busca. El mínimo de esta función cuadrática proporciona ya sea el nuevo punto directamente o se utiliza para determinar una línea a lo largo de la cual puede realizarse una búsqueda unidimensional. Para implementar el algoritmo, el gradiente se puede proporcionar en forma de un procedimiento o calculado numéricamente por diferencias finitas. La hessiana \mathbf{H} no se calcula, pero un cálculo con factorización LU se actualiza conforme el proceso continúa.

Algoritmo de Nelder-Mead

Para minimizar una función $F: \mathbb{R}^n \rightarrow \mathbb{R}$, existe otro método llamado el **algoritmo de Nelder-Mead**. Se trata de un método de *búsqueda directa* y funciona sin implicar las derivadas de la función F y sin ningún tipo de *búsquedas de línea*.

Antes de comenzar los cálculos, el usuario asigna valores a tres parámetros: α, β y γ . Los valores predeterminados son $1, \frac{1}{2}$ y 1 , respectivamente. En cada paso del algoritmo, se da un conjunto

de $n + 1$ puntos en $\mathbb{R}^n \{x_0, x_1, \dots, x_n\}$. Este conjunto está en la *posición general* en \mathbb{R}^n . Esto significa que el conjunto de n puntos $x_i - x_0$, con $1 \leq i \leq n$, es linealmente independiente. Una consecuencia de esta hipótesis es que el extremo convexo del conjunto original $\{x_0, x_1, \dots, x_n\}$ es un **n -simplex**. Por ejemplo, un **2-simplex** es un triángulo en \mathbb{R}^2 y un **3-simplex** es un tetraedro en \mathbb{R}^3 . Para hacer la descripción del algoritmo lo más simple posible, se supone que los puntos se han etiquetado nuevamente (si es necesario) para que $F(x_0) \geq F(x_1) \geq \dots \geq F(x_n)$. Puesto que estamos tratando de minimizar la función F , el punto x_0 es el peor del conjunto actual, ya que produce el mayor valor de F .

Calculamos el punto

$$u = \frac{1}{n} \sum_{i=1}^n x_i$$

Este es el **centroide** de la cara del simplex actual opuesto al peor vértice, x_0 . A continuación, calculamos un **punto reflejado** $v = (1 + \alpha)u - \alpha x_0$.

Si $F(v)$ es menor que $F(x_n)$, entonces esta es una situación favorable y uno se siente tentado a sustituir x_0 por v y a empezar de nuevo. Sin embargo, primero se calcula un **punto reflejado expandido** $w = (1 + \gamma)v - \gamma u$ y se prueba para ver si $F(w)$ es menor que $F(x_n)$. Si es así, sustituimos x_0 por w y se empieza de nuevo. De lo contrario, sustituimos x_0 por v , como se propuso inicialmente e iniciamos con el nuevo simplex.

Supongamos ahora que $F(v)$ no es menor que $F(x_n)$. Si $F(v) \leq F(x_1)$, entonces se sustituye x_0 por v y se empieza de nuevo. Habiendo dispuesto de todos los casos, cuando $F(v) \leq F(x_1)$, ahora consideramos dos casos más. Primero, si $F(v) \leq F(x_0)$, entonces se define $w = u + \beta(v - u)$. Si $F(v) > F(x_0)$, se calcula $w = u + \beta(x_0 - u)$. Con w definida ahora, probamos si $F(w) < F(x_0)$. Si esto es cierto, sustituimos x_0 por w y se empieza de nuevo. Sin embargo, si $F(w) \leq F(x_0)$, el simplex se reduce usando $x_i \leftarrow \frac{1}{2}(x_i + x_n)$ para $0 \leq i \leq n - 1$. Entonces, empezamos de nuevo.

El algoritmo necesita una prueba de paro en cada paso importante. Una de esas pruebas es la de si el **aplanado** relativo es pequeño. Este es la cantidad

$$\frac{|F(x_0) - F(x_n)|}{|F(x_0)| + |F(x_n)|}$$

Se pueden agregar otras pruebas para asegurar el avance que se está haciendo. En la programación del algoritmo se conserva el número de evaluaciones de f en un mínimo. De hecho, sólo se necesitan tres índices: los índices de los más grandes $F(x_i)$, el siguiente más grande y el menor.

Además del artículo original de Nelder y Mead [1965], se puede consultar Dennis y Woods [1987] Dixon [1974] y Torczon [1997]. Diferentes autores dan versiones ligeramente distintas del algoritmo. Hemos seguido la descripción original de Nelder y Mead.

Método de recocido simulado

Este método se ha propuesto y se ha comprobado que es eficaz para la *minimización* de funciones **difíciles**, especialmente si tienen muchos puntos mínimos de carácter puramente local. No implica derivadas o *búsqueda de líneas*; tiene gran éxito de hecho al minimizar funciones discretas, como las que surgen en el *problema del agente de ventas*.

Supongamos que se tiene una función real de n variables reales, es decir, $F: \mathbb{R}^n \rightarrow \mathbb{R}$. Debemos poder calcular los valores de $F(x)$ para cualquier x de \mathbb{R}^n . Queremos buscar un *punto mínimo global* de F , que es un punto x^* tal que $F(x^*) \leq F(x)$ para toda x en \mathbb{R}^n . En otras palabras,

$F(x^*)$ es igual a $\inf_{x \in \mathbb{R}^n} F(x)$. El algoritmo genera una sucesión de puntos x_1, x_2, x_3, \dots , y se espera que $\min_{j \leq k} F(x_j)$ converja a $\inf F(x)$ cuando $k \rightarrow \infty$.

Es suficiente describir el cálculo que conduce a x_{k+1} , suponiendo que se ha calculado x_k . Empezamos generando un reducido número de puntos aleatorios u_1, u_2, \dots, u_m , en una gran vecindad de x_k . En cada uno de estos puntos se debe calcular el valor de F . El siguiente punto, x_{k+1} , en nuestra sucesión se elige como uno de los puntos u_1, u_2, \dots, u_m . Esta elección se realiza de la siguiente manera. Se selecciona un índice j tal que

$$F(u_j) = \min \{F(u_1), F(u_2), \dots, F(u_m)\}$$

Si $F(u_j) < F(x_k)$, entonces se hace $x_{k+1} = u_j$. En otro caso, a cada i le asignamos una probabilidad p_i a u_i con la fórmula

$$p_i = e^{\alpha[F(x_k) - F(u_i)]} \quad (1 \leq i \leq m)$$

En este caso, α es un parámetro positivo elegido por el usuario del código. Normalizamos las probabilidades dividiendo cada una entre su suma. Es decir, calculamos

$$S = \sum_{i=1}^m p_i$$

y después hacemos la sustitución

$$p_i \leftarrow p_i / S$$

Por último, se hace una selección aleatoria entre los puntos u_1, u_2, \dots, u_m , teniendo en cuenta las probabilidades p_i que se les han asignado. Los u_i elegidos al azar serán los x_{k+1} .

La forma más sencilla de hacer esta elección aleatoria es emplear un generador de números aleatorios para obtener un punto ξ aleatoriamente en el intervalo $(0, 1)$. Seleccione a i como el primer entero tal que

$$\xi \leq p_1 + p_2 + \dots + p_i$$

Así, si $\xi \leq 1$, sea $i = 1$ (y $x_{n+1} = u_1$). Si $p_1 < \xi \leq p_1 + p_2$, entonces sea $i = 2$ (y $x_{n+1} = u_2$) y así sucesivamente.

La fórmula para las p_i probabilidades se toma de la teoría de la termodinámica. Si usted está interesado puede consultar el artículo original de Metropolis y colaboradores [1953] o el de Otten y Van Ginneken [1989]. Presumiblemente, otras funciones pueden hacer este papel.

¿Cuál es el propósito de la complicada elección de x_{k+1} ? Debido a la posibilidad de encontrar mínimos locales, el algoritmo debe de vez en cuando elegir un punto que *está arriba* del punto actual. Entonces, existe la posibilidad de que los puntos subsecuentes pudieran comenzar a moverse hacia un mínimo local diferente. Para hacer esto posible se introduce un elemento de aleatoriedad.

Con pequeñas modificaciones, el algoritmo se puede utilizar para funciones $f: X \rightarrow \mathbb{R}$, donde X es cualquier conjunto. Por ejemplo, en el *problema del agente de ventas*, X será el conjunto de todas las permutaciones de un conjunto de números enteros $\{1, 2, 3, \dots, n\}$. Todo lo que se requiere es un procedimiento para generar permutaciones aleatorias y, por supuesto, un código para evaluar la función f .

Los programas de computadora para este algoritmo se pueden encontrar en Internet como en el sitio web <http://www.netlib.gov> y <http://www.ingber.com>. Un conjunto de artículos sobre este tema, enfatizando el cálculo paralelo, está en Azencott [1992].

Resumen

(1) En un problema de minimización típico buscamos un punto \mathbf{x}^* tal que

$$F(\mathbf{x}^*) \leq F(\mathbf{x}) \quad \text{para toda } \mathbf{x} \in \mathbb{R}^n$$

donde F es una función de múltiples variables con valores reales.

(2) Un **vector gradiente** $\mathbf{G}(\mathbf{x})$ tiene componentes

$$G_i = G_i(\mathbf{x}) = \frac{\partial F(\mathbf{x})}{\partial x_i} \quad (1 \leq i \leq n)$$

y una **matriz hessiana** $\mathbf{H}(\mathbf{x})$ tiene componentes

$$H_{ij} = H_{ij}(\mathbf{x}) = \frac{\partial^2 F(\mathbf{x})}{\partial x_i \partial x_j} \quad (1 \leq i, j \leq n)$$

Se trata de una matriz simétrica si las derivadas de segunda orden son continuas.

(3) La **serie de Taylor** de F es

$$F(\mathbf{x} + \mathbf{h}) = F(\mathbf{x}) + \mathbf{G}(\mathbf{x})^T \mathbf{h} + \frac{1}{2} \mathbf{h}^T \mathbf{H}(\mathbf{x}) \mathbf{h} + \dots$$

Aquí, \mathbf{x} es el punto fijo de expansión en \mathbb{R}^n y \mathbf{h} es la variable en \mathbb{R}^n , con componentes h_1, h_2, \dots, h_n . Los tres puntos indican los términos de orden superior en \mathbf{h} que no se necesitan en este análisis.

(4) Una forma alternativa de la serie de Taylor es

$$F(\mathbf{x}) = F(\mathbf{z}) + \mathbf{G}(\mathbf{z})^T (\mathbf{x} - \mathbf{z}) + \frac{1}{2} (\mathbf{x} - \mathbf{z})^T \mathbf{H}(\mathbf{z}) (\mathbf{x} - \mathbf{z}) + \dots$$

Por ejemplo, una **función lineal** $F(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x}$ tiene la serie de Taylor

$$F(\mathbf{x}) = F(\mathbf{z}) + \mathbf{b}^T (\mathbf{x} - \mathbf{z})$$

Una **función cuadrática** es

$$F(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$$

(5) Un procedimiento iterativo para la localización de un punto mínimo de una función F es empezar con un punto \mathbf{z} que es un cálculo actual del punto mínimo, luego se calcula el gradiente \mathbf{G} y la hessiana \mathbf{H} de F en el punto \mathbf{z} y se resuelve la ecuación matricial

$$\mathbf{H} \mathbf{x} = -\mathbf{G}$$

para \mathbf{x} . Después se remplaza \mathbf{z} por $\mathbf{z} + \mathbf{x}$ y se repite.

(6) Si la matriz \mathbf{H} tiene la propiedad de $\mathbf{x}^T \mathbf{H} \mathbf{x} > 0$ para cada vector \mathbf{x} distinto de cero, entonces la función cuadrática \mathbf{Q} tiene un punto mínimo único.

(7) Los algoritmos que se analizan son el de **máxima pendiente**, de **Nelder-Mead** y de **recocido simulado**.

Referencias adicionales

Para leer más sobre el tema de la optimización, consulte los libros y artículos de Azencott [1992], Baldick [2006], Beale [1988], Cvijovic y Kilnowski [1995], Dennis y Schnabel [1983, 1996], Dennis y Woods [1987], Dixon [1974], Fletcher [1976], Floudas y Pardalos [1992], Gill Murray y Wright [1981], Herz-Fischer [1998], Horst, Pardalos y Thoai [2000], Kelley [2003], Kirkpatrick et al. [1983], Lootsma [1972], Moré y Wright [1993], Nelder y Mead [1965], Nocedal y Wright [2006], Otten y Van Ginneken [1989], Rheinboldt [1998], Roos, Terlaky y Vial [1997], Toreczon [1997] y Törn y Zilinskas [1989].

Problemas 16.2

- 1.** Determine si estas funciones tienen valores mínimos en \mathbb{R}^2 :

- a.** $x_1^2 - x_1x_2 + x_2^2 + 3x_1 + 6x_2 - 4$
- b.** $x_1^2 - 3x_1x_2 + x_2^2 + 7x_1 + 3x_2 + 5$
- c.** $2x_1^2 - 3x_1x_2 + x_2^2 + 4x_1 - x_2 + 6$
- d.** $ax_1^2 - 2bx_1x_2 + cx_2^2 + dx_1 + ex_2 + f$

Sugerencia: utilice el método de completar el cuadrado.

- 2.** Determine el punto mínimo de $3x^2 - 2xy + y^2 + 3x - 46 + 7$ encontrando el gradiente y la hessiana y resolviendo las ecuaciones lineales apropiadas.
- 3.** Usando $(0, 0)$ como el punto de expansión, escriba los tres primeros términos de la serie de Taylor para $F(x, y) = e^x \cos y - y \ln(x + 1)$.
- 4.** Usando $(1, 1)$ como punto de expansión, escriba los tres primeros términos de la serie de Taylor para $F(x, y) = 2x^2 - 4xy + 7y^2 - 3x + 5y$.
- 5.** La expansión de la serie de Taylor alrededor de cero se puede escribir como

$$F(\mathbf{x}) = F(0) + \mathbf{G}(0)^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{H}(0) \mathbf{x} + \dots$$

Muestre que la serie de Taylor alrededor de \mathbf{z} se puede escribir en una forma similar utilizando la notación de matriz y vector, es decir,

$$F(\mathbf{x}) = F(\mathbf{z}) + \mathcal{G}(\mathbf{z})^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathcal{H}(\mathbf{z}) \mathbf{x} + \dots$$

donde

$$\mathbf{x} = \begin{bmatrix} x \\ z \end{bmatrix}, \quad \mathcal{G}(\mathbf{z}) = \begin{bmatrix} \mathbf{G}(\mathbf{z}) \\ -\mathbf{G}(\mathbf{z}) \end{bmatrix}, \quad \mathcal{H}(\mathbf{z}) = \begin{bmatrix} \mathbf{H}(\mathbf{z}) & -\mathbf{H}(\mathbf{z}) \\ -\mathbf{H}(\mathbf{z}) & \mathbf{H}(\mathbf{z}) \end{bmatrix}$$

- 6.** Demuestre que el gradiente de $F(x, y)$ es perpendicular al contorno. *Sugerencia:* interprete la ecuación $F(x, y) = c$, definiendo y como una función de x . Entonces, por la regla de la cadena,

$$\frac{\partial F}{\partial x} + \frac{\partial F}{\partial y} \frac{dy}{dx} = 0$$

Obtenga la pendiente de la tangente al contorno.

7. Considere la función

$$F(x_1, x_2, x_3) = 3e^{x_1 x_2} - x_3 \cos x_1 + x_2 \ln x_3$$

- a. Determine el vector gradiente y la matriz hessiana.
 - ^ab. Deduzca los tres primeros términos de la expansión de la serie de Taylor alrededor de $(0, 1, 1)$.
 - c. ¿Qué sistema lineal debe ser resuelto para una suposición razonable como el punto mínimo de F ? ¿Cuál es el valor de F en este punto?
8. Se afirma que la hessiana de una función F desconocida en un cierto punto es

$$\begin{bmatrix} 3 & 2 \\ 1 & 4 \end{bmatrix}$$

¿Qué conclusión puede sacar acerca de F ?

9. ¿Cuáles son los gradientes de las siguientes funciones en los puntos indicados?
- ^aa. $F(x, y) = x^2y - 2x + y$ en $(1, 0)$
 - ^ab. $F(x, y, z) = xy + yz^2 + x^2z$ en $(1, 2, 1)$
10. Considere $F(x, y, z) = y^2z^2(1 + \sin^2 x) + (y + 1)^2(z + 3)^2$. Queremos encontrar el mínimo de la función. El programa que se utilizará requiere el gradiente de la función. ¿Qué fórmulas se deben programar para el gradiente?
11. Sea F una función de dos variables cuyo gradiente en $(0, 0)$ es $[-5, 1]^T$ y cuya matriz hessiana es

$$\begin{bmatrix} 6 & -1 \\ -1 & 2 \end{bmatrix}$$

Haga una suposición razonable del punto mínimo de F . Explique.

12. Escriba la función $F(x_1, x_2) = 3x_1^2 + 6x_1x_2 - 2x_2^2 + 5x_1 + 3x_2 + 7$ en la forma de la ecuación (7) con las \mathbf{A} , \mathbf{b} y c adecuadas. Muestre las ecuaciones lineales en forma matricial que se deben resolver para encontrar un punto en el que las primeras derivadas parciales de F sean iguales a cero. Por último, resuelva estas ecuaciones numéricamente para localizar este punto.
13. Verifique la ecuación (13). Al derivar la suma doble de la ecuación (12), primero escriba todos los términos que contienen a x_k . Despues derive y utilice la simetría de la matriz \mathbf{H} .
14. Considere la función cuadrática Q en la ecuación (12). Demuestre que si \mathbf{H} es definida positiva, entonces el punto estacionario es un punto mínimo.
15. **(Función cuadrática general)** Generalice la ecuación (6) a n variables. Demuestre que una función cuadrática general $Q(\mathbf{x})$ de n variables se puede escribir en la forma de matriz-vector de la ecuación (7), donde \mathbf{A} es una matriz simétrica de $n \times n$, \mathbf{b} es un vector de longitud n y c es un escalar. Establezca que el gradiente y la hessiana son

$$\mathbf{G}(\mathbf{x}) = \mathbf{Ax} + \mathbf{b} \quad \text{y} \quad \mathbf{H}(\mathbf{x}) = \mathbf{A}$$

respectivamente.

16. Sea A una matriz simétrica de $n \times n$ y defina una matriz triangular superior $U = (u_{ij})$ haciendo

$$u_{ij} = \begin{cases} a_{ij} & i = j \\ 2a_{ij} & i < j \\ 0 & i > j \end{cases}$$

Demuestre que $\mathbf{x}^T U \mathbf{x} = \mathbf{x}^T A \mathbf{x}$ para todos los vectores \mathbf{x} .

17. Demuestre que la función general de segundo grado $Q(\mathbf{x})$ de n variables se puede escribir

$$Q(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T U \mathbf{x}$$

donde U es una matriz triangular superior. ¿Puede esto simplificar el trabajo de encontrar el punto estacionario de Q ?

18. Demuestre que el gradiente y la hessiana satisfacen la ecuación

$$\mathbf{H}(\mathbf{z})(\mathbf{x} - \mathbf{z}) = \mathbf{G}(\mathbf{x}) - \mathbf{G}(\mathbf{z})$$

para una función cuadrática general de n variables.

19. Usando la serie de Taylor, demuestre que una función cuadrática general de n variables se puede escribir en forma de bloque

$$Q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathcal{A} \mathbf{x} + \mathcal{B}^T \mathbf{x} + c$$

donde

$$\mathbf{x} = \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} \mathbf{A} & -\mathbf{A} \\ -\mathbf{A} & \mathbf{A} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix}$$

Aquí \mathbf{z} es el punto de expansión.

20. (Problema de mínimos cuadrados) Considere la función

$$F(\mathbf{x}) = (\mathbf{b} - \mathbf{Ax})^T (\mathbf{b} - \mathbf{Ax}) + \alpha \mathbf{x}^T \mathbf{x}$$

donde A es una matriz real de $m \times n$, \mathbf{b} es un vector columna real de orden m y α es un número real positivo. Queremos el punto mínimo de F para A , \mathbf{b} y α dados. Demuestre que

$$F(\mathbf{x} + \mathbf{h}) - F(\mathbf{x}) = (\mathbf{Ah})^T (\mathbf{Ah}) + \alpha \mathbf{h}^T \mathbf{h} \geq 0$$

para \mathbf{h} un vector de orden n , suponiendo que

$$(\mathbf{A}^T \mathbf{A} + \alpha \mathbf{I}) \mathbf{x} = \mathbf{A}^T \mathbf{b}$$

Esto significa que cualquier solución de este sistema lineal minimiza a $F(\mathbf{x})$, por lo que esta es la ecuación normal.

21. (Opción múltiple) ¿Cuál es el gradiente de la función $f(\mathbf{x}) = 3x_1^2 - \operatorname{sen}(x_1 x_2)$ en el punto $(3, 0)$?

- a. $(6, -3)$
- b. $(3, -1)$
- c. $(18, 0)$
- d. $(18, -3)$
- e. Ninguno de estos.

22. (Opción múltiple, continuación) La derivada direccional de la función f en el punto x en la dirección u está dada por la expresión

$$\frac{d}{dt} f(x + tu)|_{t=0}$$

En esta descripción, \mathbf{u} debe ser un vector unitario. ¿Cuál es el valor numérico de la derivada direccional donde $f(\mathbf{x})$ es la función definida en el problema anterior, $\mathbf{x} = (1, \pi/2)$ y $\mathbf{u} = (1, 1)/\sqrt{2}$.

- a. $6/\sqrt{2}$ b. 6 c. 18 d. 3 e. Ninguno de estos.

23. (Opción múltiple, continuación) Si f es una función real de n variables, la matriz hessiana $\mathbf{H} = (H_{ij})$ está dada por $H_{ij} = \partial^2 f / \partial x_i \partial x_j$, todos los términos se están evaluando en un punto específico \mathbf{x} . ¿Cuál es la entrada H_{22} de esta matriz en el caso de que f sea como se indica en el problema anterior y $\mathbf{x} = (1, \pi/2)$?

- a. 6 b. $6/\sqrt{2}$ c. 1 d. $\pi^2/2$ e. Ninguno de estos.

24. (Opción múltiple) Sea f una función real de n variables reales. Sean \mathbf{x} y \mathbf{u} dadas como vectores numéricos y $\mathbf{u} \neq 0$. Entonces, la expresión $f(\mathbf{x} + t\mathbf{u})$ define una función de t . Supongamos que el mínimo de $f(\mathbf{x} + t\mathbf{u})$ se presenta en $t = 0$. ¿Qué conclusión se puede sacar?

- a. El gradiente de f en x , denotado por $G(x)$, es 0.
 - b. u es perpendicular al gradiente de f en x .
 - c. $u = G(x)$, donde $G(x)$ denota el gradiente de f en x .
 - d. $G(x)$ es perpendicular a x .
e. Ninguna de estas.

- 25.** (Opción múltiple) Si f es una función cuadrática real de n variables reales, se puede escribir en la forma $f(\mathbf{x}) = c - \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$. El gradiente de f es entonces:

- a. Ax b. $b - Ax$ c. $Ax - b$ d. $\frac{1}{2}Ax - b$ e. Ninguno de estos.

Problemas de cómputo 16.2

1. Seleccione una rutina de su biblioteca de programas o de un paquete como Matlab, Maple o Mathematica para minimizar una función de muchas variables, sin necesidad de programar derivadas. Pruebelas con una o más de las siguientes funciones bien conocidas. El orden de nuestras variables es (x, y, z, w) .

- a.** Rosenbrock: $100(y - x^2)^2 + (1 - x)^2$. Inicie en $(-1.2, 1.0)$.

b. Powell: $(x + 10y)^2 + 5(z - w)^2 + (y - 2z)^4 + 10(x - w)^4$. Inicie en $(3, -1, 0, 1)$.

c. Powell: $x^2 + 2y^2 + 3z^2 + 4w^2 + (x + y + z + w)^4$. Inicie en $(1, -1, 1, 1)$.

d. Fletcher y Powell: $(z - 10\phi)^2 + (\sqrt{x^2 + y^2} - 1)^2 + z^2$ en la que ϕ es un ángulo determinado a partir de (x, y) mediante

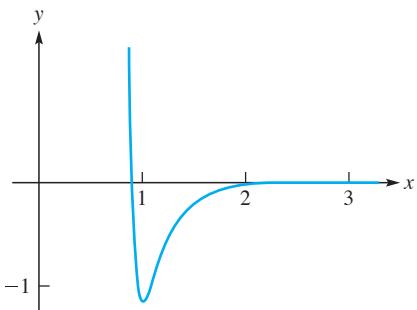
$$\frac{\cos 2\pi \phi = x}{\sqrt{x^2 + y^2}} \quad y \quad \frac{\sin 2\pi \phi = y}{\sqrt{x^2 + y^2}}$$

donde $-\pi/2 < 2\pi\phi \leq 3\pi/2$. Inicie en $(1, 1, 1)$.

- e. Wood: $100(x^2 - y)^2 + (1 - x)^2 + 90(z^2 - w)^2 + (1 - z)^2 + 10(y - 1)^2 + (w - 1)^2 + 19.8(y - 1)(w - 1)$. Inicie en $(-3, -1, -3, -1)$.
2. **(Máxima pendiente acelerado)** Esta versión del método de máxima pendiente es superior a la básica. Una sucesión de puntos $\mathbf{x}_1, \mathbf{x}_2, \dots$ se genera de la siguiente manera. El punto \mathbf{x}_1 se especifica como el punto de partida. Luego, \mathbf{x}_2 se obtiene por un paso del de máxima pendiente de \mathbf{x}_1 . En el paso general, si se han obtenido $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m$, encontramos un punto \mathbf{z} de máxima pendiente a partir de \mathbf{x}_m . Entonces \mathbf{x}_{m+1} se toma como el punto mínimo de la recta $\mathbf{x}_{m-1} + t(\mathbf{z} - \mathbf{x}_{m-1})$. Programe y pruebe este algoritmo en uno de los ejemplos en el problema de cálculo 16.2.1.
3. Utilizando una rutina de su biblioteca de programas o de Matlab, Maple o Mathematica y
- resuelva el problema de minimización con que inicia este capítulo.
 - trace y resuelva para el punto mínimo, el punto máximo y el punto de silla de estas funciones, respectivamente: $x^2 + y^2, 1 - x^2 - y^2, x^2 - y^2$.
 - trace la gráfica y realice el experimento numérico con estas funciones que no tienen mínimos: $-x^2 - y^2 + 13x + 6y + 12, x^2 - y^2 + 3x + 5y + 7, x^2 - 2xy + x + 2y + 3, 2x + 4y + 6$.
4. Queremos encontrar el mínimo de $F(x, y, z) = z^2 \cos x + x^2 y^2 + x^2 e^x$ utilizando un programa informático que requiera procedimientos para el gradiente de F junto con F . Escriba los procedimientos necesarios. Encuentre el mínimo usando un código preprogramado que utilice el gradiente.
5. Suponga que
- ```
procedure $Xmin(f, (grad_i), n, (xi), (g_{ij}))$
```
- está disponible para calcular el valor mínimo de una función de dos variables. Suponga que esta rutina requiere no sólo la función sino también su gradiente. Si vamos a utilizar esta rutina con la función  $F(x, y) = e^x \cos^2(xy)$ , ¿qué procedimiento será necesario? Escriba el código apropiado. Encuentre el mínimo usando un código preprogramado que utilice el gradiente.
6. Programe y pruebe el algoritmo de Nelder-Mead.
7. Programe y pruebe el algoritmo de recocido simulado.
8. **(Proyecto de investigación estudiantil)** Investigue uno de los nuevos métodos para minimizar como el de los algoritmos genéticos, los métodos de recocido simulado o el algoritmo de Nelder-Mead. Utilice algo del software disponible para ellos.
9. Utilice las rutinas incorporadas en los sistemas de software matemático como Maple o Mathematica para verificar los cálculos del ejemplo 1. *Sugerencia:* en Maple, use `grad` y `Hessian`, y en Matemática, use `Series`. Por ejemplo, obtenga los dos términos de la serie de Taylor en dos variables desarrollada alrededor del punto  $(1, 1)$  y después haga un cambio de variables.
10. **(Conformación molecular: proyecto de plegamiento de proteínas)** Las fuerzas que rigen el plegamiento de las proteínas en aminoácidos se deben a enlaces entre los átomos individuales y al debilitamiento de las interacciones entre los átomos no enlazados, como las fuerzas electrostáticas y de Van der Waals. Las fuerzas de Van der Waals son modeladas por el potencial de Lennard-Jones

$$U(r) = \frac{1}{r^{12}} - \frac{2}{r^6}$$

donde  $r$  es la distancia entre los átomos.



En la figura, la energía mínima es  $-1$  y se obtiene en  $r = 1$ . Investigue este tema y los métodos numéricos utilizados. Un método es predecir la conformación de las proteínas al encontrar la energía potencial mínima de la configuración total de los aminoácidos. Para un grupo de átomos con posiciones  $(x_1, y_1, z_1)$  a  $(x_n, y_n, z_n)$ , la función objetivo que se debe minimizar es

$$U = \sum_{i < j} \frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^6}$$

sobre todos los pares de átomos. En este caso,  $r_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]^{1/2}$  es la distancia entre los átomos  $i$  y  $j$ . Este problema de optimización determina las coordenadas rectangulares de los átomos. Consulte Sauer [2006] para más detalles.

# Programación lineal

En el estudio de cómo la economía de Estados Unidos se ve afectada por cambios en la oferta y el costo de la energía, se ha encontrado conveniente utilizar un modelo de programación lineal. Este es un gran sistema de desigualdades lineales que gobiernan las variables del modelo, junto con una función lineal de estas variables que debe ser maximizada. Normalmente, las variables son los niveles de actividad de los distintos procesos en la economía, tales como el número de barriles de petróleo extraídos por día o el número de camisas para hombres producidas por día. Un modelo que contiene un detalle razonable podría fácilmente implicar miles de variables y miles de desigualdades lineales. Estos problemas se tratan en este capítulo y en algunos se ofrece orientación sobre cómo utilizar el software existente.

---

## 17.1 Formas estándar y dualidad

### Primera forma primal

La **programación lineal** es una rama de las matemáticas que trata de encontrar los valores extremos de las funciones lineales cuando las variables se ven limitadas por desigualdades lineales. Cualquier problema de este tipo se puede poner en un formulario estándar conocido como la *primera forma primal* mediante manipulaciones simples (que se analizarán más adelante).

En notación matricial, el problema de programación lineal en la primera forma primal es como sigue:

$$\begin{cases} \text{maximizar: } c^T x \\ \text{restricciones: } \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \end{cases} \quad (1)$$

**TEOREMA 1****Primera forma primal**

Dados los datos  $c_j, a_{ij}, b_i$  (para  $1 \leq j \leq n, 1 \leq i \leq m$ ), queremos determinar las  $x_j$  ( $1 \leq j \leq n$ ) que maximizan la función lineal

$$\sum_{j=1}^n c_j x_j$$

sujeta a las restricciones

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i & (1 \leq i \leq m) \\ x_j \geq 0 & (1 \leq j \leq n) \end{cases}$$

Aquí,  $c$  y  $\mathbf{x}$  son vectores de  $n$  componentes,  $\mathbf{b}$  es un vector de  $m$  componentes y  $\mathbf{A}$  es una matriz de  $m \times n$ . Una **desigualdad vectorial**  $\mathbf{u} \leq \mathbf{v}$  significa que  $\mathbf{u}$  y  $\mathbf{v}$  son vectores con el mismo número de componentes y que *todos* los componentes satisfacen la desigualdad  $u_i \leq v_i$ . La función lineal  $\mathbf{c}^T \mathbf{x}$  se llama la **función objetivo**.

En un problema de programación lineal, el conjunto de todos los vectores que satisfacen las restricciones se llama el **conjunto factible** y sus elementos son los **puntos factibles**. Así, en la notación anterior, el conjunto factible es

$$\mathbf{K} = \{ \mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq 0 \text{ y } \mathbf{Ax} \leq \mathbf{b} \}$$

Un enunciado más preciso (y conciso) del problema de programación lineal, entonces, es el siguiente: determine  $\mathbf{x}^* \in \mathbf{K}$  tal que  $\mathbf{c}^T \mathbf{x}^* \geq \mathbf{c}^T \mathbf{x}$  para todo  $\mathbf{x} \in \mathbf{K}$ .

**Ejemplo numérico**

Para tener una idea del tipo de problema práctico que se puede resolver mediante programación lineal, consideremos un ejemplo simple de optimización. Supongamos que una fábrica utiliza dos materias primas para producir dos productos. Supongamos también que se cumplen los siguientes enunciados:

- Cada unidad del primer producto requiere de 5 unidades de la primera materia prima y 3 de la segunda.
- Cada unidad del segundo producto requiere de 3 unidades de la primera materia prima y 6 de la segunda.
- Por un lado son 15 unidades de la primera materia prima y 18 unidades de la segunda.
- Las ganancias de las ventas de los productos son 2 por unidad del primer producto y 3 por unidad del segundo producto.

*¿Cómo deberían utilizarse las materias primas para lograr una ganancia máxima?* Para responder a esta pregunta se introducen las variables  $x_1$  y  $x_2$  para representar el número de unidades de los dos productos que se fabrican. En términos de estas variables, la ganancia es

$$2x_1 + 3x_2$$

El proceso utiliza hasta  $5x_1 + 3x_2$  unidades de la primera materia prima y  $3x_1 + 6x_2$  unidades de la segunda. Las limitaciones del tercer enunciado anterior se expresan con estas desigualdades:

$$\begin{cases} 5x_1 + 3x_2 \leq 15 \\ 3x_1 + 6x_2 \leq 18 \end{cases}$$

Por supuesto,  $x_1 \geq 0$  y  $x_2 \geq 0$ . Así, la solución al problema es un vector  $\mathbf{x} \geq 0$  que maximiza la función objetivo  $2x_1 + 3x_2$  mientras satisface las restricciones anteriores. Así, el problema de programación lineal es

$$\begin{cases} \text{maximizar: } & 2x_1 + 3x_2 \\ \text{restricciones: } & \begin{cases} 5x_1 + 3x_2 \leq 15 \\ 3x_1 + 6x_2 \leq 18 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{cases} \end{cases} \quad (2)$$

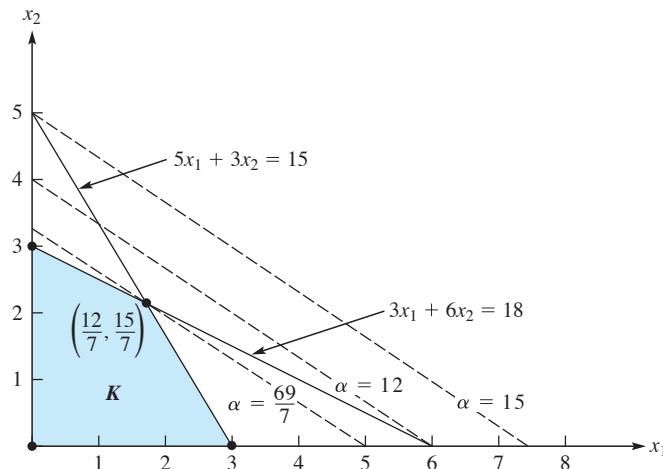
Más precisamente, entre todos los vectores  $\mathbf{x}$  en el conjunto

$$K = \{ \mathbf{x}: \mathbf{x} \geq 0, 5x_1 + 3x_2 \leq 15, 3x_1 + 6x_2 \leq 18 \}$$

queremos el que hace a  $2x_1 + 3x_2$  tan grande como sea posible.

Debido a que el número de variables en este ejemplo es de sólo dos, el problema se puede resolver de forma gráfica. Para encontrar la solución, se empieza trazando la gráfica del conjunto  $K$ . Esta es la región sombreada de la figura 17.1. Después se dibujan algunas de las rectas  $2x_1 + 3x_2 = \alpha$ , donde  $\alpha$  se le dan diferentes valores. Estas son las rectas punteadas de la figura y se etiquetan con los valores de  $\alpha$ . Por último, se selecciona una de estas rectas con una  $\alpha$  máxima que interseque a  $K$ . Esta intersección es el punto solución y un vértice de  $K$ . Se obtiene numéricamente al resolver simultáneamente las ecuaciones  $5x_1 + 3x_2 = 15$  y  $3x_1 + 6x_2 = 18$ . Así,  $\mathbf{x} = [\frac{12}{7}, \frac{15}{7}]^T$  y la ganancia correspondiente de la ecuación (2) es  $2(\frac{12}{7}) + 3(\frac{15}{7}) = \frac{69}{7}$ .

Podemos utilizar los sistemas de software matemático como Matlab, Maple o Mathematica para resolver este problema de programación lineal. Por ejemplo, se obtiene la solución  $\mathbf{x} = \frac{12}{7}$



**FIGURA 17.1**  
Método de  
solución gráfica

y  $y = \frac{15}{7}$  con el valor de la función objetivo  $\frac{69}{7}$  usando un solo sistema, y se obtiene la solución  $x = 1.7143$  y  $y = 2.1429$  con el valor de la función objetivo utilizada como  $-9.8571$  en otro. (¿Por qué?)

Algunos de estos sistemas matemáticos contienen grandes colecciones de comandos para la optimización general de las funciones lineales y no lineales. Para la optimización no lineal, estas funciones se pueden manejar sin y con restricciones, así como la minimización de un gran número de otras tareas. Si el programa realiza la minimización de la función objetivo y queremos maximizarla, debemos minimizar lo negativo de la función objetivo. También, se pueden permitir restricciones de igualdad adicionales y puesto que no hay ninguna, se hacen entradas iguales a cero.

Observe en este ejemplo que las unidades que se utilizan (ya sea dólares, pesos, libras o kilogramos), no importan en el método matemático siempre que se utilicen en forma coherente. Observe también que  $x_1$  y  $x_2$  pueden ser números reales arbitrarios. El problema sería muy diferente si sólo valores enteros fueran aceptables como solución. Esta situación se presentaría si los productos que se producen consisten en unidades indivisibles, como un artículo manufacturado. Si se impone la restricción de número entero, sólo son aceptables los puntos de coordenadas enteras dentro de  $K$ . Por eso,  $(0, 3)$  es el mejor de ellos. Observe en particular que *no podemos* redondear la solución  $(1.71, 2.14)$ , a los enteros más cercanos para resolver el problema con restricciones de números enteros. El punto  $(2, 2)$  se encuentra fuera de  $K$ . Sin embargo, si la empresa puede alterar las restricciones ligeramente al aumentar la cantidad de la primera materia prima a 16, la solución entera  $(2, 2)$  sería admisible. Programas especiales para la programación lineal entera están disponibles, pero están fuera del alcance de este libro.

Observe cómo se podría modificar la solución si nuestra ganancia o función objetivo fuera  $2x_1 + x_2$ . En este caso, las líneas punteadas de la figura tendrían una pendiente diferente (a saber,  $-2$ ) y un vértice distinto de la región sombreada se produciría como solución, a saber,  $(3, 0)$ . Un rasgo característico de los problemas de programación lineal es que las soluciones (si existen) siempre se pueden encontrar entre los vértices.

## Transformación de problemas en la primera forma primal

Un problema de programación lineal que no está en la primera forma primal se pueden poner en esa forma por algunas de las técnicas estándar:

- Si el problema original pide la minimización de la función lineal  $\mathbf{c}^T \mathbf{x}$ , es lo mismo que pedir la maximización  $(-\mathbf{c})^T \mathbf{x}$ .
- Si el problema original contiene una restricción como  $\mathbf{a}^T \mathbf{x} \leq \beta$ , se puede sustituir por la restricción  $(-\mathbf{a})^T \mathbf{x} \leq -\beta$ .
- Si la función objetivo contiene una constante, este hecho no tiene ningún efecto sobre la solución. Por ejemplo, el máximo de  $\mathbf{c}^T \mathbf{x} + \lambda$  se obtiene para la misma  $\mathbf{x}$  que el máximo de  $\mathbf{c}^T \mathbf{x}$ .
- Si el problema original contiene restricciones de igualdad, cada una se puede sustituir con dos restricciones de desigualdad. Así, la ecuación  $\mathbf{a}^T \mathbf{x} = \beta$  es equivalente a  $\mathbf{a}^T \mathbf{x} \leq \beta$  y  $\mathbf{a}^T \mathbf{x} \geq \beta$ .
- Si el problema original no requiere que una variable (por ejemplo,  $x_i$ ) sea no negativa, se puede sustituir  $x_i$  por la diferencia de dos variables no negativas, por ejemplo,  $x_i = u_i - v_i$ , donde  $u_i \leq 0$  y  $v_i \leq 0$ .

Aquí se presenta un ejemplo que ilustra las cinco técnicas. Considere el problema de programación lineal

$$\begin{cases} \text{minimizar: } & 2x_1 + 3x_2 - x_3 + 4 \\ \text{restricciones: } & \begin{cases} x_1 - x_2 + 4x_3 \geq 2 \\ x_1 + x_2 + x_3 = 15 \\ x_2 \geq 0 \geq x_3 \end{cases} \end{cases} \quad (3)$$

Es equivalente al siguiente problema en la primera forma primal:

$$\begin{cases} \text{maximizar: } & -2u + 2v - 3z - w \\ \text{restricciones: } & \begin{cases} -u + v + z + 4w \leq -2 \\ u - v + z - w \leq 15 \\ -u + v - z + w \leq -15 \\ u \geq 0 \quad v \geq 0 \quad z \geq 0 \quad w \geq 0 \end{cases} \end{cases}$$

## Problema dual

Correspondiente a un problema dado de programación lineal en la primera forma primal hay otro problema, conocido como su **dual**. Se obtiene a partir del problema primal original

$$(P) \begin{cases} \text{maximizar: } & c^T x \\ \text{restricciones: } & \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \end{cases}$$

definiendo el dual el problema será

$$(D) \begin{cases} \text{minimizar: } & b^T y \\ \text{restricciones: } & \begin{cases} A^T y \geq c \\ y \geq 0 \end{cases} \end{cases}$$

Por ejemplo, el dual del problema

$$\begin{cases} \text{maximizar: } & 2x_1 + 3x_2 \\ \text{restricciones: } & \begin{cases} 4x_1 + 5x_2 \leq 6 \\ 7x_1 + 8x_2 \leq 9 \\ 10x_1 + 11x_2 \leq 12 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{cases} \end{cases} \quad (4)$$

es este problema:

$$\begin{cases} \text{minimizar: } & 6y_1 + 9y_2 + 12y_3 \\ \text{restricciones: } & \begin{cases} 4y_1 + 7y_2 + 10y_3 \geq 2 \\ 5y_1 + 8y_2 + 11y_3 \geq 3 \\ y_1 \geq 0 \quad y_2 \geq 0 \quad y_3 \geq 0 \end{cases} \end{cases}$$

Observe que, en general, el problema del dual tiene dimensiones diferentes que las del problema original. Así, el número de *desigualdades* en el problema original se convierte en el número de *variables* en el problema dual.

Una relación elemental entre el problema primal original y su dual es la siguiente:

## ■ TEOREMA 2

### Teorema de problemas primales y duales

Si  $\mathbf{x}$  satisface las restricciones del problema primal y  $\mathbf{y}$  satisface las restricciones de su dual, entonces  $\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}$ . En consecuencia, si  $\mathbf{c}^T \mathbf{x} = \mathbf{b}^T \mathbf{y}$ , entonces  $\mathbf{x}$  y  $\mathbf{y}$  son soluciones del problema primal y del problema dual, respectivamente.

Demostración Por las suposiciones hechas,  $\mathbf{x} \geq 0$ ,  $\mathbf{Ax} \leq \mathbf{b}$ ,  $\mathbf{y} \geq 0$  y  $\mathbf{A}^T \mathbf{y} \geq \mathbf{c}$ . En consecuencia,

$$\mathbf{c}^T \mathbf{x} \leq (\mathbf{A}^T \mathbf{y})^T \mathbf{x} = \mathbf{y}^T \mathbf{A} \mathbf{x} \leq \mathbf{y}^T \mathbf{b} = \mathbf{b}^T \mathbf{y}$$



Esta relación se puede utilizar para estimar el número  $\lambda = \max\{\mathbf{c}^T \mathbf{x} : \mathbf{x} \geq 0 \text{ y } \mathbf{Ax} \leq \mathbf{b}\}$ . (Este número se llama a menudo el **valor** del problema de programación lineal.) Para calcular  $\lambda$ , tome cualquier  $\mathbf{x}$  y  $\mathbf{y}$  que satisfaga  $\mathbf{x} \geq 0$ ,  $\mathbf{y} \geq 0$ ,  $\mathbf{Ax} \leq \mathbf{b}$  y  $\mathbf{A}^T \mathbf{y} \geq \mathbf{c}$ . Entonces  $\mathbf{c}^T \mathbf{x} \leq \lambda \leq \mathbf{b}^T \mathbf{y}$ . La importancia del problema dual proviene del hecho de que los valores extremos en los problemas primal y dual son los mismos. El enunciado formal es el siguiente:

## ■ TEOREMA 3

### Teorema de dualidad

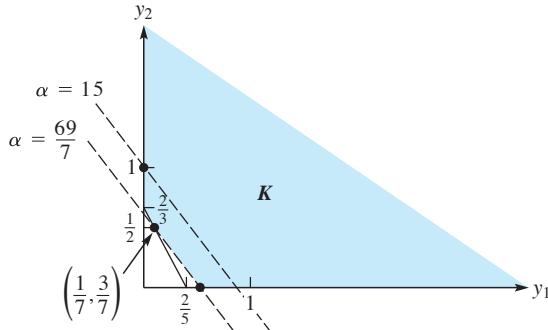
Si el problema original tiene una solución  $\mathbf{x}^*$ , entonces el problema dual tiene una solución  $\mathbf{y}^*$ ; además,  $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$ .

Este resultado está muy bien ilustrado en el ejemplo numérico del comienzo de esta sección. El dual de este problema es

$$\left\{ \begin{array}{l} \text{minimizar: } 15y_1 + 18y_2 \\ \text{restricciones: } \begin{cases} 5y_1 + 3y_2 \geq 2 \\ 3y_1 + 6y_2 \geq 3 \\ y_1 \geq 0 \quad y_2 \geq 0 \end{cases} \end{array} \right. \quad (5)$$

La gráfica de este problema se presenta en la figura 17.2. Trasladando la recta  $15y_1 + 18y_2 = \alpha$ , vemos que el vértice  $(\frac{1}{7}, \frac{3}{7})$  es el punto mínimo. Los valores de las funciones objetivo son realmente idénticos porque  $15(\frac{1}{7}) + 18(\frac{3}{7}) = \frac{69}{7}$ . Además, las soluciones  $\mathbf{x} = [\frac{12}{7}, \frac{15}{7}]^T$  y  $\mathbf{y} = [\frac{1}{7}, \frac{3}{7}]^T$  pueden estar relacionadas, pero no vamos a analizar esto.

Podemos utilizar sistemas de software matemático como Matlab, Maple o Mathematica para resolver este problema de programación lineal. Por ejemplo, obtenemos  $x = 0.1429$  y  $y = 0.4286$  con  $f(x, y) = 9.8571$ .



**FIGURA 17.2**  
Método gráfico  
del problema  
dual

## Segunda forma primal

Volviendo al problema general de la primera forma primal, se introducen variables no negativas adicionales  $x_{n+1}, x_{n+2}, \dots, x_{n+m}$ , conocidas como **variables de holgura**, para que algunas de las desigualdades se puedan escribir como igualdades. Usando este dispositivo, podemos poner el problema original en la forma estándar siguiente.

### ■ TEOREMA 4

#### Segunda forma primal

Maximice la función lineal

$$\sum_{j=1}^n c_j x_j$$

sujeto a las restricciones

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i & (1 \leq i \leq m) \\ x_j \geq 0 & (1 \leq j \leq m+n) \end{cases}$$

Usando notación matricial, tenemos

$$\begin{aligned} &\text{maximizar: } \mathbf{c}^T \mathbf{x} \\ &\text{restricciones: } \begin{cases} \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \end{aligned}$$

Aquí, se supone que la matriz  $\mathbf{A}$  de  $m \times n$  contiene una matriz identidad de  $m \times m$  en sus últimas  $m$  columnas y que las últimas  $m$  entradas de  $\mathbf{c}$  son 0. También, observe que cuando un problema en la primera forma primal se cambia a la segunda forma primal, se incrementa el número de variables, lo que altera las cantidades  $n, \mathbf{x}, \mathbf{c}$  y  $\mathbf{A}$ . Esto es, un problema en la primera forma primal con  $n$  variables contendría  $n+m$  variables en la segunda forma.

Para ilustrar la transformación de un problema de la primera forma primal a la segunda, considere el ejemplo que se presentó al inicio de esta sección:

$$\begin{cases} \text{maximizar: } 2x_1 + 3x_2 \\ \text{restricciones: } \begin{cases} 5x_1 + 3x_2 \leq 15 \\ 3x_1 + 6x_2 \leq 18 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{cases} \end{cases} \quad (6)$$

Se introdujeron dos variables de holgura  $x_3$  y  $x_4$  para dar espacio a las dos desigualdades. El nuevo problema en la segunda forma primal es entonces

$$\begin{cases} \text{maximizar: } 2x_1 + 3x_2 + 0x_3 + 0x_4 \\ \text{restricciones: } \begin{cases} 5x_1 + 3x_2 + x_3 = 15 \\ 3x_1 + 6x_2 + x_4 = 18 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0 \end{cases} \end{cases}$$

Los problemas que afectan los valores absolutos de las variables o los valores absolutos de las expresiones lineales con frecuencia se pueden convertir en problemas de programación lineal. Para ilustrarlo, considere el problema de minimizar  $|x - y|$  sujeto a restricciones lineales en  $x$  y en  $y$ . Podemos introducir una nueva variable  $z \geq 0$  y después imponer las restricciones  $x - y \leq z$ ,  $-x + y \leq z$ . Entonces tratamos de minimizar la forma lineal  $0x + 0y + 1z$ .

## Resumen

---

(1) El problema de programación lineal en la **primera forma primal** es

$$\begin{cases} \text{maximizar: } \mathbf{c}^T \mathbf{x} \\ \text{restricciones: } \begin{cases} A\mathbf{x} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \end{cases}$$

(2) Su **problema dual** es

$$\begin{cases} \text{minimizar: } \mathbf{b}^T \mathbf{y} \\ \text{restricciones: } \begin{cases} \mathbf{A}^T \mathbf{y} \geq \mathbf{c} \\ \mathbf{y} \geq 0 \end{cases} \end{cases}$$

(3) La **segunda forma primal** es

$$\begin{cases} \text{maximizar: } \mathbf{c}^T \mathbf{x} \\ \text{restricciones: } \begin{cases} A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \end{cases}$$

donde la matriz  $A$  de  $m \times n$  contiene una matriz de identidad de  $m \times m$  en sus últimas  $m$  columnas y donde las  $m$  últimas entradas de  $\mathbf{c}$  son 0.

(4) Si  $x$  satisface las restricciones del problema primal y  $y$  satisface y las restricciones de su dual, entonces  $c^T x \leq b^T y$ . En consecuencia, si  $c^T x = b^T y$ , entonces  $x$  y  $y$  son soluciones del problema primal y del problema dual, respectivamente.

(5) Los valores extremos en los problemas primal y dual son los mismos.

## Problemas 17.1

1. Escriba el siguiente problema en la primera forma primal:

$$\left\{ \begin{array}{l} \text{minimizar: } |x_1 + 2x_2 - x_3| \\ \text{restricciones: } \begin{cases} x_1 + 3x_2 - x_3 \leq 8 \\ 2x_1 - 4x_2 - x_3 \geq 1 \\ |4x_1 + 5x_2 + 6x_3| \leq 12 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \end{cases} \end{array} \right.$$

Sugerencia:  $|a| \leq \beta$  se puede escribir como  $-\beta \leq a \leq \beta$ .

2. Un programa está disponible para resolver problemas de programación lineal en la primera forma primal. Escriba el siguiente problema en esa forma:

$$\left\{ \begin{array}{l} \text{minimizar: } 5x_1 + 6x_2 - 2x_3 + 8 \\ \text{restricciones: } \begin{cases} 2x_1 - 3x_2 \geq 5 \\ x_1 + x_2 \leq 15 \\ 2x_1 - x_2 + x_3 \leq 25 \\ x_1 + x_2 - x_3 \geq 1 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \end{cases} \end{array} \right.$$

3. Considere los siguientes problemas de programación lineal:

a. maximizar:  $2x_1 + 3x_2$

$$\text{restricciones: } \left\{ \begin{array}{l} x_1 + 2x_2 \geq -6 \\ -x_1 + 3x_2 \leq 3 \\ |2x_1 - 5x_2| \leq 5 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{array} \right.$$

b. minimizar:  $7x_1 + x_2 - x_3 + 4$

$$\text{restricciones: } \left\{ \begin{array}{l} x_1 - x_2 + x_3 \geq 2 \\ x_1 + x_2 + x_3 \leq 10 \\ -2x_1 - x_2 \leq -4 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{array} \right.$$

Vuelva a escribir cada problema en la primera forma primal y presente el problema dual.

4. Dibuje la región factible para las siguientes restricciones:

$$\begin{cases} x - y \leq 2 \\ x + y \leq 3 \\ 2x + y \geq 3 \\ x \geq 0 \quad y \geq 0 \end{cases}$$

<sup>a</sup>a. Sustituyendo los vértices en la función objetivo

$$z(x, y) = x + 2y$$

determine el valor mínimo de esta función en la región factible.

b. Sea

$$z(x, y) = \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2$$

Demuestre que el valor mínimo de  $z$  en la región factible no se produce en un vértice.

5. Escriba los siguientes problemas de programación lineal en la primera forma primal. ¿Cuál es el dual de cada uno?

$$\begin{array}{ll} \text{a.} & \begin{cases} \text{minimizar: } 2x + y - 3z + 1 \\ \text{restricciones: } \begin{cases} x - y \geq 3 \\ |x - z| \leq 2 \\ x \geq 0 \quad y \geq 0 \end{cases} \end{cases} \\ \text{b.} & \begin{cases} \text{minimizar: } 3x - 2y + 5z + 3 \\ \text{restricciones: } \begin{cases} x + y + z \geq 4 \\ x - y - z = 2 \\ x \geq 0 \quad y \geq 0 \quad z \geq 0 \end{cases} \end{cases} \\ \text{c.} & \begin{cases} \text{maximizar: } 3x + 2y \\ \text{restricciones: } \begin{cases} 6x + 5y \leq 17 \\ 2x + 11y \leq 23 \\ x \leq 0 \end{cases} \end{cases} \end{array}$$

6. Considere el siguiente problema de programación lineal:

$$\begin{cases} \text{maximizar: } 2x_1 + 2x_2 - 6x_3 - x_4 \\ \text{restricciones: } \begin{cases} 3x_1 + x_4 = 25 \\ x_1 + x_2 + x_3 + x_4 = 20 \\ 4x_1 + 6x_3 \geq 5 \\ 2x_1 + 3x_3 + 2x_4 \geq 0 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0 \end{cases} \end{cases}$$

- a.* Reformule este problema en la segunda forma primal.  
*b.* Formule el problema dual.

**7.** Resuelva gráficamente el siguiente problema de programación lineal:

$$\left\{ \begin{array}{l} \text{maximizar: } 3x_1 + 5x_2 \\ \text{restricciones: } \begin{cases} x_1 \leq 4 \\ x_2 \leq 6 \\ 3x_1 + 2x_2 \leq 18 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{cases} \end{array} \right.$$

**8.** (Continuación) Resuelva el problema dual del problema anterior.

**9.** Demuestre que el problema dual puede escribirse como

$$\left\{ \begin{array}{l} \text{maximizar: } b^T y \\ \text{restricciones: } \begin{cases} y^T A \geq c^T \\ y \geq 0 \end{cases} \end{array} \right.$$

**10.** Describa cómo  $\max \{ |x - y - 3|, |2x + y + 4|, |x + 2y - 7| \}$  se puede minimizar usando un código de programación lineal.

**11.** Muestre cómo se puede resolver este problema mediante programación lineal:

$$\left\{ \begin{array}{l} \text{minimizar: } |x - y| \\ \text{restricciones: } \begin{cases} x \leq 3y \\ x \geq y \\ y \leq x - 2 \end{cases} \end{array} \right.$$

**12.** Considere el problema de programación lineal

$$\left\{ \begin{array}{l} \text{minimizar: } x_1 + x_4 + 25 \\ \text{restricciones: } \begin{cases} 2x_1 + 2x_2 + x_3 < 7 \\ 2x_1 - 3x_2 + x_4 = 4 \\ x_2 - x_4 > 1 \\ 3x_2 - 8x_3 + x_4 = 5 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases} \end{array} \right.$$

Escriba en forma matriz-vector el problema dual y el segundo problema primal.

**13.** Resuelva cada uno de los problemas de programación lineal por el método gráfico. Determine  $x$  para

$$\left\{ \begin{array}{l} \text{maximizar: } c^T x \\ \text{restricciones: } \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \end{array} \right.$$

En este caso, se pueden obtener “soluciones” no únicas y no acotadas.

$$\text{a. } \mathbf{c} = [2, -4]^T \quad \mathbf{A} = \begin{bmatrix} -3 & -5 \\ 4 & 9 \end{bmatrix} \quad \mathbf{b} = [-15, 36]$$

$$\text{b. } \mathbf{c} = \begin{bmatrix} 2, & 1 \\ & 2 \end{bmatrix}^T \quad \mathbf{A} = \begin{bmatrix} 6 & 5 \\ 4 & 1 \end{bmatrix} \quad \mathbf{b} = [30, 12]^T$$

$$\text{c. } \mathbf{c} = [3, 2]^T \quad \mathbf{A} = \begin{bmatrix} -3 & 2 \\ -4 & 9 \end{bmatrix} \quad \mathbf{b} = [6, 36]^T$$

$$\text{d. } \mathbf{c} = [2, -3]^T \quad \mathbf{A} = \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix} \quad \mathbf{b} = [0, 5]^T$$

$$\text{e. } \mathbf{c} = [-4, 11]^T \quad \mathbf{A} = \begin{bmatrix} -3 & 4 \\ -4 & 11 \end{bmatrix} \quad \mathbf{b} = [12, 44]^T$$

$$\text{f. } \mathbf{c} = [-3, 4]^T \quad \mathbf{A} = \begin{bmatrix} 2 & 3 \\ -4 & -5 \end{bmatrix} \quad \mathbf{b} = [6, -20]^T$$

$$\text{g. } \mathbf{c} = [2, 1]^T \quad \mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \quad \mathbf{b} = [0, -2]^T$$

$$\text{h. } \mathbf{c} = [3, 1]^T \quad \mathbf{A} = \begin{bmatrix} 2 & 4 \\ 5 & 3 \end{bmatrix} \quad \mathbf{b} = [21, 18]^T$$

**14.** Resuelva a mano el siguiente problema de programación lineal, usando una gráfica de ayuda:

$$\left\{ \begin{array}{l} \text{maximizar: } 4x + 4y + z \\ \text{restricciones: } \begin{cases} 3x + 2y + z = 12 \\ 7x + 7y + 2z \leq 144 \\ 7x + 5y + 2z \leq 80 \\ 11x + 7y + 3z \leq 132 \\ x \geq 0 \quad y \geq 0 \end{cases} \end{array} \right.$$

*Sugerencia:* use la ecuación para eliminar  $z$  de todas las otras expresiones. Resuelva el problema bidimensional resultante.

**15.** Escriba este problema de programación lineal en la segunda forma primal. Es posible que quiera realizar cambios de variables. Si es así, incluya un *diccionario* que relacione las variables nuevas con las viejas.

$$\left\{ \begin{array}{l} \text{minimizar: } \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \\ \text{restricciones: } \begin{cases} |3x + 4y + 6| \leq \varepsilon_1 \\ |2x - 8y - 4| \leq \varepsilon_2 \\ |-x - 3y + 5| \leq \varepsilon_3 \\ \varepsilon_1 > 0 \quad \varepsilon_2 > 0 \quad \varepsilon_3 > 0 \quad x > 0 \quad y > 0 \end{cases} \end{array} \right.$$

Resuelva el problema resultante.

- 16.** Considere el siguiente problema de programación lineal:

$$\begin{cases} \text{maximizar: } c_1x_1 + c_2x_2 \\ \text{restricciones: } \begin{cases} a_1x_1 + a_2x_2 \leq b \\ x_1 \geq 0 \quad x_2 \geq 0 \end{cases} \end{cases}$$

En el caso especial en que todos los datos sean positivos, muestre que el problema dual tiene el mismo valor extremo que el del problema original.

- 17.** Supongamos que un problema de programación lineal en primera forma primal tiene la propiedad de que  $c^T x$  no está acotada por el conjunto factible. ¿Qué conclusión se puede sacar del problema dual?
- 18.** (Opción múltiple) ¿Cuál de los siguientes problemas se formula en la primera forma primal para un problema de programación lineal?
- a. maximizar  $c^T x$  sujeto a  $Ax \leq b$
  - b. minimizar  $c^T x$  sujeto a  $Ax \leq b, x \geq 0$
  - c. maximizar  $c^T x$  sujeto a  $Ax = b, x \geq 0$
  - d. maximizar  $c^T x$  sujeto a  $Ax \leq b, x \geq 0$
  - e. Ninguno de estos.

## Problemas de cómputo 17.1

1. Una tienda del oeste desea comprar 300 sombreros de vaquero de fieltro y 200 sombreros de paja. Se han recibido ofertas de tres mayoristas. Sombreros Texas ha aceptado hacer no más de 200 sombreros, Sombreros Estrella Solitaria no más de 250 y Ropa de Rancho Lazo no más de 150. El propietario de la tienda ha estimado que su ganancia por sombrero vendido de Sombreros Texas sería de \$3/fieltro y \$4/paja, de Sombreros Estrella Solitaria \$3.80/fieltro y \$3.50/paja y de Ropa de Rancho Lazo \$4/fieltro y \$3.60/paja. Establezca un problema de programación lineal para maximizar las ganancias de los propietarios. Resuelva usando un programa de su biblioteca de software.
2. La compañía farmacéutica ABC hace dos tipos de analgésico líquido que llevan por nombres Relieve (R) y Facilidad (F) y contienen diferentes mezclas de los tres medicamentos básicos, A, B y C, producidos por la compañía. Cada botella de R requiere  $\frac{7}{9}$  unidades del medicamento A,  $\frac{1}{2}$  unidades del medicamento B y  $\frac{3}{4}$  unidades del medicamento C. Cada botella de E requiere  $\frac{4}{9}$  unidades del medicamento A,  $\frac{5}{2}$  unidades del medicamento B y  $\frac{1}{4}$  unidades del medicamento C. La compañía es capaz de producir cada día sólo 5 unidades del medicamento A, 7 unidades del medicamento B y 9 unidades del C. Además, los reglamentos de Administración de alimentos y medicina establecen que el número de botellas de R fabricadas no puede superar el doble del número de botellas de E. El margen de ganancia por cada botella de E y R es de \$7 y \$3, respectivamente. Configure el problema de programación lineal en la primera forma primal para determinar el número de botellas de los dos analgésicos que la empresa debe producir cada día a fin de maximizar sus ganancias. Resuelva usando el software disponible.
3. Supongamos que la sociedad de estudiantes de una universidad desea fletar aviones para el transporte de al menos 750 estudiantes al juego del tazón. Dos líneas aéreas,  $\alpha$  y  $\beta$ , acuerdan

suministrar las aeronaves para el viaje. La compañía aérea  $\alpha$  cuenta con cinco aeronaves disponibles para llevar 75 pasajeros cada una y la línea aérea  $\beta$  tiene tres aviones disponibles para llevar 250 pasajeros cada uno. El costo por avión es de \$900 y \$3250 para el viaje de las compañías aéreas  $\alpha$  y  $\beta$ , respectivamente. La sociedad de estudiantes quiere alquilar un máximo de seis aviones. ¿Cuántos de cada tipo debe alquilar para minimizar el costo del transporte aéreo? ¿Cuánto debe cobrar la sociedad de estudiantes para tener 50 centavos de ganancia por estudiante? Resuelva por el método gráfico y verifique usando una rutina de su biblioteca de programas.

4. (Continuación) Vuelva a resolver el problema de cómputo anterior con las dos diferentes formas siguientes:
  - a. El número de estudiantes que van en el transporte aéreo se maximiza.
  - b. El costo por estudiante se reduce al mínimo.
5. (**Problema de dieta**) El comedor universitario desea ofrecer al menos 5 unidades de vitamina C y 3 unidades de vitamina E por porción. Tres alimentos están disponibles con estas vitaminas. El alimento  $f_1$  contiene 2.5 y 1.25 unidades por onza de vitaminas C y E, respectivamente, mientras que el alimento  $f_2$  contiene exactamente cantidades opuestas. El tercer alimento  $f_3$  contiene la misma cantidad de cada vitamina en 1 unidad por onza. El alimento  $f_1$  cuesta 25¢ por onza, el alimento  $f_2$  cuesta 56¢ por onza, el alimento  $f_3$  cuesta 10¢ por onza. El dietista desea proporcionar la comida a un costo mínimo por porción que satisfaga los requisitos mínimos de las vitaminas. Establezca este problema de programación lineal en la segunda forma primal. Resuelva con la ayuda de un código de su biblioteca de programas de cómputo.
6. Utilice rutinas integradas en los sistemas de software matemático como Matlab, Maple o Mathematica para resolver el problema de programación lineal con el número de ecuación a continuación en la primera forma primal y en la segunda forma primal, y en forma dual:

a. (2)

b. (3)

c. (4)

d. (5)

e. (6)

## 17.2 Método Simplex

El algoritmo principal que se utiliza en la resolución de problemas de programación lineal es el *método simplex*. Se presentan suficientes bases de este método para que usted pueda utilizar los programas informáticos disponibles que lo incorporan.

Considere un problema de programación lineal en la segunda forma primal

$$\left\{ \begin{array}{l} \text{maximizar: } \mathbf{c}^T \mathbf{x} \\ \text{restricciones: } \begin{cases} \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geq 0 \end{cases} \end{array} \right.$$

Se supone que  $\mathbf{c}$  y  $\mathbf{x}$  son vectores de  $n$  componentes,  $\mathbf{b}$  es un vector de  $m$  componentes y  $\mathbf{A}$  es una matriz de  $m \times n$ . También se supone que  $\mathbf{b} \geq 0$  y que  $\mathbf{A}$  contiene una matriz identidad de  $m \times m$

en sus últimas  $m$  columnas. Como antes, definimos el conjunto de puntos factibles como

$$K = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$$

Los puntos de  $K$  son exactamente los puntos que están compitiendo para maximizar  $c^T x$ .

## Vértices en $K$ y columnas de $A$ linealmente independientes

El conjunto  $K$  es un conjunto **poliédrico** en  $\mathbb{R}^n$ , y el algoritmo que se describirá va de vértice a vértice en  $K$ , siempre aumentando el valor de  $c^T x$  conforme va de uno a otro. Daremos una definición más precisa de *vértice*. Un punto  $x$  en  $K$  se llama un **vértice** si es imposible expresarlo como  $x = \frac{1}{2}(u + v)$ , con  $u$  y  $v$  en  $K$  y  $u \neq v$ . En otras palabras,  $x$  no está en el punto medio de ningún segmento cuyos extremos estén en  $K$ .

Denotamos por  $a^{(1)}, a^{(2)}, \dots, a^{(n)}$  los vectores columna que forman la matriz  $A$ . El teorema siguiente relaciona las columnas de  $A$  con los vértices de  $K$ :

### ■ TEOREMA 1

#### Teorema de vértices y vectores columna

Sea  $x \in K$  y definimos  $\mathcal{I}(x) = \{i: x_i > 0\}$ . Entonces los siguientes enunciados son equivalentes:

1.  $x$  es un vértice de  $K$ .
2. El conjunto  $\{a^{(i)}: i \in \mathcal{I}(x)\}$  es linealmente independiente.

#### Demostración

Si el enunciado 1 es falso, entonces podemos escribir  $x = \frac{1}{2}(u + v)$ , con  $u \in K$ ,  $v \in K$  y  $u \neq v$ . Para cada índice  $i$  que no esté en el conjunto  $\mathcal{I}(x)$ , tenemos  $x_i = 0$ ,  $u_i \geq 0$ ,  $v_i \geq 0$  y  $x_i = \frac{1}{2}(u_i + v_i)$ . Esto obliga a que  $u_i$  y  $v_i$  sean cero. Por lo tanto, todas las componentes diferentes de cero de  $u$  y  $v$  corresponden a índices  $i$  en  $\mathcal{I}(x)$ . Puesto que  $u$  y  $v$  pertenecen a  $K$ ,

$$b = Au = \sum_{i=1}^n u_i a^{(i)} = \sum_{i \in \mathcal{I}(x)} u_i a^{(i)}$$

y

$$b = Av = \sum_{i=1}^n v_i a^{(i)} = \sum_{i \in \mathcal{I}(x)} v_i a^{(i)}$$

Por ende, obtenemos

$$\sum_{i \in \mathcal{I}(x)} (u_i - v_i) a^{(i)} = 0$$

que muestra la dependencia lineal del conjunto  $\{a^{(i)}: i \in \mathcal{I}(x)\}$ . Por ello, el enunciado 2 es falso. Por consiguiente, el enunciado 2 implica el enunciado 1.

Para lo inverso, suponga que el enunciado 2 es falso. De la dependencia lineal de los vectores columna  $a^{(i)}$  para  $i \in \mathcal{I}(x)$  tenemos

$$\sum_{i \in \mathcal{I}(x)} y_i a^{(i)} = 0 \quad \text{con} \quad \sum_{i \in \mathcal{I}(x)} |y_i| \neq 0$$

para los coeficientes adecuados  $y_i$ . Para cada  $i \notin \mathcal{I}(\mathbf{x})$ , sea  $y_i = 0$ . Se forma el vector  $\mathbf{y}$  con las componentes  $y_i = 1, 2, \dots, n$ . Entonces, para cualquier  $\lambda$ , vemos que en virtud de que  $\mathbf{x} \in \mathbf{K}$ ,

$$\mathbf{A}(\mathbf{x} \pm \lambda \mathbf{y}) = \sum_{i=1}^n (x_i \pm \lambda y_i) \mathbf{a}^{(i)} = \sum_{i=1}^n x_i \mathbf{a}^{(i)} \pm \lambda \sum_{i \in \mathcal{I}(\mathbf{x})} y_i \mathbf{a}^{(i)} = \mathbf{Ax} = \mathbf{b}$$

Ahora se selecciona el número real  $\lambda$  positivo, pero tan pequeño que  $\mathbf{x} + \lambda \mathbf{y} \geq 0$  y que  $\mathbf{x} - \lambda \mathbf{y} \geq 0$ . [Para ver que esto es posible, considere por separado las componentes para  $i \in \mathcal{I}(\mathbf{x})$  y para  $i \notin \mathcal{I}(\mathbf{x})$ .] Los vectores resultantes,  $\mathbf{u} = \mathbf{x} + \lambda \mathbf{y}$  y  $\mathbf{v} = \mathbf{x} - \lambda \mathbf{y}$ , pertenecen a  $\mathbf{K}$ . Son diferentes y obviamente,  $\mathbf{x} = \frac{1}{2}(\mathbf{u} + \mathbf{v})$ . Así,  $\mathbf{x}$  no es un vértice de  $\mathbf{K}$ , es decir, el enunciado 1 es falso. Por lo tanto, el enunciado 1 implica el enunciado 2. ■

Dado un problema de programación lineal, hay tres posibilidades:

1. No hay puntos factibles, es decir, el conjunto  $\mathbf{K}$  está vacío.
2.  $\mathbf{K}$  no está vacío y  $c^T \mathbf{x}$  no está dentro de  $\mathbf{K}$ .
3.  $\mathbf{K}$  no está vacío y  $c^T \mathbf{x}$  está dentro de  $\mathbf{K}$ .

Es cierto (pero no obvio) que en el tercer caso, hay un punto  $\mathbf{x}$  en  $\mathbf{K}$  tal que  $c^T \mathbf{x} \geq c^T \mathbf{y}$  para toda  $\mathbf{y}$  en  $\mathbf{K}$ . Hemos supuesto que nuestro problema está en la segunda forma primal, por lo que la posibilidad 1 no puede ocurrir. En efecto,  $\mathbf{A}$  contiene una matriz identidad de  $m \times m$ , así que tiene la forma

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & 1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2k} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} & 0 & 0 & \cdots & 1 \end{bmatrix}$$

donde  $k = n - m$ . En consecuencia, podemos construir fácilmente un punto factible  $\mathbf{x}$  haciendo  $x_1 = x_2 = \cdots = x_k = 0$  y  $x_{k+1} = b_1, x_{k+2} = b_2$  y así sucesivamente. Queda claro entonces que  $\mathbf{Ax} = \mathbf{b}$ . La desigualdad  $\mathbf{x} \geq 0$  se sigue de la suposición inicial de que  $\mathbf{b} \leq 0$ .

## Método simplex

A continuación presentamos un breve resumen del método simplex para resolver problemas de programación lineal. Implica una secuencia de intercambios, por lo que la solución de prueba sistemática de un vértice a otro en  $\mathbf{K}$ . Este procedimiento se detiene cuando el valor de  $c^T \mathbf{x}$  no aumenta como resultado del intercambio.

Lo siguiente es un esquema del **algoritmo simplex**.

## ■ ALGORITMO 1 Simplex

Seleccione un pequeño valor positivo para  $\varepsilon$ . En cada paso, tenemos un conjunto de  $m$  índices  $\{k_1, k_2, \dots, k_m\}$ .

1. Coloque las columnas  $a^{(k_1)}, a^{(k_2)}, \dots, a^{(k_m)}$  en  $B$  y resuelva  $Bx = b$ .
2. Si  $x_i > 0$  para  $1 \leq i \leq m$ , continúe. De lo contrario, salga porque el algoritmo ha fallado.
3. Haga  $e = [c_{k_1}, c_{k_2}, \dots, c_{k_m}]^T$ , y resuelva  $B^T y = e$ .
4. Elija cualquier  $s$  en  $\{1, 2, \dots, n\}$ , (pero no en  $\{k_1, k_2, \dots, k_m\}$ ) para el que  $c_s - y^T a^{(s)}$  es el mayor.
5. Si  $c_s - y^T a^{(s)} < \varepsilon$ , salga porque  $x$  es la solución.
6. Resuelva  $Bz = a^{(s)}$ .
7. Si  $z_i \leq \varepsilon$  para  $1 \leq i \leq m$ , entonces salga porque la función objetivo está dentro de  $K$ .
8. Entre los cocientes  $x_i/z_i$  que tienen  $z_i > 0$  para  $1 \leq i \leq m$ , sea  $x_r/z_r$  el más pequeño. En caso de empate, sea  $r$  el primero en presentarse.
9. Reemplace  $k_r$  con  $s$  y vaya al paso 1.

Algunas observaciones acerca de este algoritmo están en orden. Al principio, seleccione los índices  $k_1, k_2, \dots, k_m$  tal que  $a^{(k_1)}, a^{(k_2)}, \dots, a^{(k_m)}$  formen una matriz identidad de  $m \times m$ . En el paso 5, donde decimos que  $x$  es una solución, queremos decir que el vector  $v = (v_i)$  dado por  $v_{k_i} = x_i$  para  $1 \leq i \leq n$  y  $v_i = 0$  para  $i \notin \{k_1, k_2, \dots, k_m\}$  es la solución. Una opción conveniente para la tolerancia  $\varepsilon$  que se presenta en los pasos 5 y 7 puede ser  $10^{-6}$ .

En cualquier aplicación razonable del método simplex, debe aprovecharse el hecho de que los incidentes de éxito de la etapa 1 son muy similares. De hecho, sólo una columna de  $B$  cambia a la vez. Observaciones similares son también válidas para los pasos 3 y 6.

No recomendamos que usted intente programar el algoritmo simplex. Códigos eficaces, refinados a lo largo de muchos años de experiencia, están generalmente disponibles en las bibliotecas de software. Muchos de ellos pueden aportar soluciones a un problema dado y a su dual con un poco más de programación. A veces esta característica se puede aprovechar para disminuir el tiempo de ejecución de un problema. Para ver por qué, considere un problema de programación lineal en la **primera forma primal**:

$$(P) \quad \begin{aligned} &\text{maximizar: } c^T x \\ &\text{restricciones: } \begin{cases} Ax \leq b \\ x \geq 0 \end{cases} \end{aligned}$$

Como de costumbre, suponemos que  $x$  es un  $n$  vector y que  $A$  es una matriz de  $m \times n$ . Cuando el algoritmo simplex se aplica a este problema, realiza un proceso iterativo en una matriz de  $m \times m$  denominada por  $B$  en la descripción anterior. Si el número de  $m$  desigualdades de restricciones es muy grande con respecto a  $n$ , entonces el problema dual puede ser más fácil de resolver, ya que las matrices  $B$  para este serán de dimensión  $n \times n$ . De hecho, el **problema dual** es

$$(D) \quad \begin{aligned} &\text{minimizar: } b^T y \\ &\text{restricciones: } \begin{cases} A^T y \geq c \\ y \geq 0 \end{cases} \end{aligned}$$

y aquí el número de desigualdades de restricciones es  $n$ . Un ejemplo de esta técnica se presenta en la sección siguiente.

## Resumen

- (1) Para la segunda forma primal, el conjunto de **puntos factibles** es

$$K = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$$

que son los puntos de  $K$  que compiten para maximizar  $c^T x$ .

- (2) Para un problema de programación lineal, se tienen estas posibilidades: no existen puntos factibles, es decir, el conjunto  $K$  está vacío,  $K$  no está vacío y  $c^T x$  no está dentro de  $K$ ,  $K$  no está vacío,  $c^T x$  está dentro de  $K$ .

- (3) Se denotan por  $a^{(1)}, a^{(2)}, \dots, a^{(n)}$  los vectores columna que constituyen la matriz  $A$ . Sea  $x \in K$  y se define  $\mathcal{I}(x) = \{i : x_i > 0\}$ . Entonces  $x$  es un vértice de  $K$  si y sólo si el conjunto  $\{a^{(i)} : i \in \mathcal{I}(x)\}$  es linealmente independiente.

- (4) El **método simplex** consiste en una secuencia de intercambios para que la solución de prueba vaya sistemáticamente de un vértice a otro en el conjunto de puntos factibles  $K$ . Este procedimiento se detiene cuando el valor de  $c^T x$  ya no aumenta como resultado de los intercambios.

## Problemas 17.2

- <sup>a</sup>1. Demuestre que el problema de programación lineal

$$\begin{cases} \text{maximizar: } c^T x \\ \text{restricciones: } Ax \leq b \end{cases}$$

se puede escribir en primera forma primal aumentando el número de variables en exactamente una. *Sugerencia:* sustituya  $x_j$  por  $y_j - y_0$ .

- <sup>a</sup>2. Demuestre que el conjunto  $K$  puede tener sólo un número finito de vértices.
3. Suponga que  $u$  y  $v$  son los puntos solución para un problema de programación lineal y que  $x = \frac{1}{2}(u + v)$ . Demuestre que  $x$  es también una solución.
4. Utilizando el método simplex como se describe, resuelva el ejemplo numérico del libro.
- <sup>a</sup>5. Usando técnicas estándar, escriba el problema dual (D) en la primera y segunda forma primal.
- <sup>a</sup>6. Muestre cómo un código para resolver un problema de programación lineal en primera forma primal se puede utilizar para resolver un sistema de  $n$  ecuaciones lineales con  $n$  variables.
7. Usando técnicas estándar, escriba el problema dual (D) en la primera forma primal (P); luego tome el dual de la misma. ¿Cuál es el resultado?

## Problemas de cómputo 17.2

1. Seleccione un código de programación lineal de su biblioteca central de informática y utilícelo para resolver estos problemas:

$$\begin{array}{ll}
 \text{a.} & \left\{ \begin{array}{l} \text{minimizar: } 8x_1 + 6x_2 + 6x_3 + 9x_4 \\ \text{restricciones: } \begin{cases} x_1 + 2x_2 + x_4 \geq 2 \\ 3x_1 + x_2 + x_4 \geq 4 \\ x_3 + x_4 \geq 1 \\ x_1 + x_3 \geq 1 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0 \end{cases} \end{array} \right. \\
 \\ 
 \text{b.} & \left\{ \begin{array}{l} \text{minimizar: } 10x_1 - 5x_2 - 4x_3 + 7x_4 + x_5 \\ \text{restricciones: } \begin{cases} 4x_1 - 3x_2 - x_3 + 4x_4 + x_5 = 1 \\ -x_1 + 2x_2 + 2x_3 + x_4 + 3x_5 = 4 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0 \quad x_5 \geq 0 \end{cases} \end{array} \right. \\
 \\ 
 \text{c.} & \left\{ \begin{array}{l} \text{maximizar: } 2x_1 + 4x_2 + 3x_3 \\ \text{restricciones: } \begin{cases} 4x_1 + 2x_2 + 3x_3 \leq 15 \\ 3x_1 + 2x_2 + x_3 \leq 7 \\ x_1 + x_2 + 2x_3 \leq 6 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \end{cases} \end{array} \right. 
 \end{array}$$

2. (Proyecto de investigación estudiantil) Investigue los recientes desarrollos de algoritmos computacionales de programación lineal, en especial con métodos de puntos interiores.

## 17.3 Solución aproximada de sistemas lineales inconsistentes

La programación lineal se puede utilizar para la solución aproximada de sistemas de ecuaciones lineales que son inconsistentes. Un sistema de ecuaciones de  $m \times n$

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (1 \leq i \leq m)$$

se dice que es **inconsistente** si no hay vector  $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$  que satisfaga simultáneamente todas las  $m$  ecuaciones del sistema. Por ejemplo, el sistema

$$\left\{ \begin{array}{l} 2x_1 + 3x_2 = 4 \\ x_1 - x_2 = 2 \\ x_1 + 2x_2 = 7 \end{array} \right. \tag{1}$$

es inconsistente, como puede verse al intentar llevar a cabo el proceso de eliminación gaussiana.

## Problema $\ell_1$

Puesto que ningún vector  $x$  puede resolver un sistema inconsistente de ecuaciones, los **residuos**

$$r_i = \sum_{j=1}^n a_{ij}x_j - b_i \quad (1 \leq i \leq m)$$

no se pueden hacer cero al mismo tiempo. Por lo tanto,  $\sum_{i=1}^m |r_i| > 0$ . Ahora es natural preguntarse por un vector  $x$  que hace que la expresión  $\sum_{i=1}^m |r_i|$  tan pequeña como sea posible. Este problema se llama **problema  $\ell_1$**  para este sistema de ecuaciones. Otros criterios, que conducen a diferentes soluciones aproximadas, podría ser minimizar  $\sum_{i=1}^m r_i^2$  o  $\max_{1 \leq i \leq m} |r_i|$ . El capítulo 12 analiza en detalle el problema de minimizar  $\sum_{i=1}^m r_i^2$ .

La minimización de  $\sum_{i=1}^m |r_i|$  usando la elección adecuada del vector  $x$  es un problema para el que se han diseñado los algoritmos especiales (véase Barrodale y Roberts [1974]). Sin embargo, si uno de estos programas especiales no está disponible o si el problema es de pequeño alcance, se puede utilizar programación lineal.

Un simple, nuevo enunciado directo del problema es

$$\left\{ \begin{array}{l} \text{minimizar: } \sum_{i=1}^m \varepsilon_i \\ \text{restricciones: } \begin{cases} \sum_{j=1}^n a_{ij}x_j - b_i \leq \varepsilon_i & (1 \leq i \leq m) \\ -\sum_{j=1}^n a_{ij}x_j + b_i \leq \varepsilon_i & (1 \leq i \leq m) \end{cases} \end{array} \right. \quad (2)$$

Si se tiene a la mano un código de programación lineal en el que no se requiere que las variables sean no negativas, entonces se puede usar en el problema (2). Si las variables deben ser no negativas, se puede aplicar la siguiente técnica. Se introduce una variable  $y_{n+1}$  y se escribe  $x_j = y_j - y_{n+1}$ . Después se define  $a_{i,n+1} = -\sum_{j=1}^n a_{ij}$ . Este paso crea una columna adicional en la matriz  $A$ . Ahora considere el problema de programación lineal

$$\left\{ \begin{array}{l} \text{maximizar: } -\sum_{i=1}^m \varepsilon_i \\ \text{restricciones: } \begin{cases} \sum_{j=1}^{n+1} a_{ij}y_j - \varepsilon_i \leq b_i & (1 \leq i \leq m) \\ -\sum_{j=1}^{n+1} a_{ij}y_j - \varepsilon_i \leq -b_i & (1 \leq i \leq m) \\ y \geq 0 \quad \varepsilon \geq 0 \end{cases} \end{array} \right. \quad (3)$$

que está en la primera forma primal con  $m + n + 1$  variables y  $2m$  desigualdades de restricciones.

No es difícil comprobar que el problema (3) es equivalente al problema (2). El punto principal es que

$$\begin{aligned}\sum_{j=1}^{n+1} a_{ij} y_j &= \sum_{j=1}^n a_{ij} (x_j + y_{n+1}) + a_{i,n+1} y_{n+1} \\&= \sum_{j=1}^n a_{ij} x_j + y_{n+1} \sum_{j=1}^n a_{ij} + y_{n+1} \left( -\sum_{j=1}^n a_{ij} \right) \\&= \sum_{j=1}^n a_{ij} x_j\end{aligned}$$

Otra técnica que puede utilizarse es sustituir las  $2m$  desigualdades de restricciones en el problema (3) por un conjunto de  $m$  igualdad de restricciones. Escribimos

$$\varepsilon_i = |r_i| = u_i + v_i$$

donde  $u_i = r_i$  y  $v_i = 0$  si  $r_i \geq 0$  pero  $v_i = -r_i$  y  $u_i = 0$  si  $r_i < 0$ . El problema de programación lineal resultante es

$$\begin{cases} \text{maximizar:} & -\sum_{i=1}^m u_i - \sum_{i=1}^m v_i \\ \text{restricciones:} & \begin{cases} \sum_{j=1}^{n+1} a_{ij} y_j - u_i + v_i = b_i & (1 \leq i \leq m) \\ u \geq 0 \quad v \geq 0 \quad y \geq 0 \end{cases} \end{cases}$$

Utilizando las fórmulas anteriores, tenemos

$$\begin{aligned}r_i &= \sum_{j=1}^n a_{ij} x_j - b_i = \sum_{j=1}^n a_{ij} (y_j - y_{n+1}) - b_i \\&= \sum_{j=1}^n a_{ij} y_j - y_{n+1} \sum_{j=1}^n a_{ij} - b_i \\&= \sum_{j=1}^{n+1} a_{ij} y_j - b_i = u_i - v_i\end{aligned}$$

A partir de aquí podemos concluir que  $r_i + v_i = u_i \geq 0$ . Ahora  $v_i$  y  $u_i$  deben ser lo más pequeño posible, de acuerdo con esta restricción, porque estamos tratando de minimizar  $\sum_{i=1}^m (u_i + v_i)$ . Por ello, si  $r_i \geq 0$ , tomamos  $v_i \geq 0$  y  $u_i = r_i$ , mientras que si  $r_i < 0$ , hacemos  $v_i = -r_i$  y  $u_i = 0$ . En cualquier caso,  $|r_i| = u_i + v_i$ . Así, minimizar  $\sum_{i=1}^m (u_i + v_i)$  es igual a minimizar  $\sum_{i=1}^m |r_i|$ .

El ejemplo del sistema lineal inconsistente dado por (1) podría resolverse en el sentido de  $\ell_1$  resolviendo el problema de programación lineal

$$\begin{cases} \text{minimizar:} & u_1 + v_1 + u_2 + v_2 + u_3 + v_3 \\ \text{restricciones:} & \begin{cases} 2y_1 + 3y_2 - 5y_3 - u_1 + v_1 = 4 \\ y_1 - y_2 - u_2 + v_2 = 2 \\ y_1 + 2y_2 - 3y_3 - u_3 + v_3 = 7 \\ y_1, y_2, y_3 \geq 0 \quad u_1, u_2, u_3 \geq 0 \quad v_1, v_2, v_3 \geq 0 \end{cases} \end{cases} \quad (4)$$

La solución es

$$\begin{array}{lll} u_1 = 0 & u_2 = 0 & u_3 = 0 \\ v_1 = 0 & v_2 = 0 & v_3 = 5 \\ y_1 = 2 & y_2 = 0 & y_3 = 0 \end{array}$$

De esta ecuación, recuperamos la solución  $\ell_1$  del sistema (1) en la forma

$$\begin{array}{lll} x_1 = y_1 - y_3 = 2 & r_1 = u_1 - v_1 = & 0 \\ x_2 = y_2 - y_3 = 0 & r_2 = u_2 - v_2 = & 0 \\ & r_3 = u_3 - v_3 = -5 & \end{array}$$

Podemos utilizar sistemas de software matemático como Matlab, Maple o Mathematica para resolver este problema de programación lineal. Por ejemplo, obtenemos  $u_1 = v_1 = u_2 = v_2 = u_3 = y_2 = y_3 = 0$ ,  $v_3 = 5$  y  $y_1 = 2$ , con 5 el valor de la función objetivo. Para otro sistema, hay que establecer las igualdades de restricciones. Se obtiene la solución correspondiente a  $y_1 = y_2 = y_3 = 684.2887$ ,  $u_1 = u_2 = u_3 = v_1 = v_2 = 0$  y  $v_3 = 5$ , con 5 como el valor de la función objetivo. El vector  $x$  es  $x_1 = 2$  y  $x_2 = 3.1494 \times 10^{-11}$ . Esta solución es ligeramente diferente de la obtenida antes, debido a errores de redondeo, pero el valor mínimo de la función objetivo es el mismo y todas las restricciones se satisfacen.

## Problema $\ell_\infty$

Consideremos de nuevo un sistema de  $m$  ecuaciones lineales con  $n$  incógnitas:

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (1 \leq i \leq m)$$

Si el sistema es inconsistente, sabemos que los residuos  $r_i = \sum_{j=1}^n a_{ij}x_j - b_i$  no todos pueden ser iguales cero para cualquier vector  $x$ . Así, la cantidad  $\varepsilon = \max_{1 \leq i \leq m} |r_i|$  es positiva. El problema de hacer a  $\varepsilon$  un mínimo se llama el **problema  $\ell_\infty$**  para el sistema de ecuaciones. Un problema de programación lineal equivalentes es

$$\left\{ \begin{array}{l} \text{minimizar: } \varepsilon \\ \text{restricciones: } \left\{ \begin{array}{ll} \sum_{j=1}^n a_{ij}x_j - \varepsilon \leq b_i & (1 \leq i \leq m) \\ -\sum_{j=1}^n a_{ij}x_j - \varepsilon \leq -b_i & (1 \leq i \leq m) \end{array} \right. \end{array} \right.$$

Si está disponible un código de programación lineal en el que las variables no tienen que ser mayores o iguales a cero, entonces se puede utilizar para resolver el problema  $\ell_\infty$  formulado anteriormente. Si las variables deben ser no negativas, primero introducimos una variable  $y_{n+1}$  tan grande que las cantidades  $y_j = x_j + y_{n+1}$  sean positivas. Despues, resolvemos

el problema de programación lineal

$$\left\{ \begin{array}{l} \text{minimizar: } \varepsilon \\ \text{restricciones: } \begin{cases} \sum_{j=1}^{n+1} a_{ij} y_j - \varepsilon \leq b_i & (1 \leq i \leq m) \\ -\sum_{j=1}^{n+1} a_{ij} y_j - \varepsilon \leq -b_i & (1 \leq i \leq m) \\ \varepsilon \geq 0 \quad y_j \geq 0 & (1 \leq j \leq n+1) \end{cases} \end{array} \right. \quad (5)$$

Aquí, de nuevo hemos definido  $a_{i,n+1} = -\sum_{j=1}^n a_{ij}$ .

Para nuestro sistema (1), la solución que minimiza la cantidad

$$\max\{|2x_1 + 3x_2 - 4|, |x_1 - x_2 - 2|, |x_1 + 2x_2 - 7|\}$$

se obtiene a partir del problema de programación lineal

$$\left\{ \begin{array}{l} \text{minimizar: } \varepsilon \\ \text{restricciones: } \begin{cases} 2y_1 + 3y_2 - 5y_3 - \varepsilon \leq 4 \\ y_1 - y_2 - \varepsilon \leq 2 \\ y_1 + 2y_2 - 3y_3 - \varepsilon \leq 7 \\ -2y_1 - 3y_2 + 5y_3 - \varepsilon \leq -4 \\ -y_1 + y_2 - \varepsilon \leq -2 \\ -y_1 - 2y_2 + 3y_3 - \varepsilon \leq -7 \\ y_1, y_2, y_3 \geq 0 \quad \varepsilon \geq 0 \end{cases} \end{array} \right. \quad (6)$$

La solución es

$$y_1 = \frac{8}{9} \quad y_2 = \frac{5}{3} \quad y_3 = 0 \quad \varepsilon = \frac{25}{9}$$

A partir de aquí, la solución  $\ell_\infty$  de (1) se recupera de la siguiente manera:

$$x_1 = y_1 - y_3 = \frac{8}{9} \quad x_2 = y_2 - y_3 - \frac{5}{3}$$

Podemos utilizar los sistemas de software matemático como Matlab, Maple o Mathematica para resolver el problema de programación lineal (6). Por ejemplo, se obtiene la solución  $y_1 = \frac{8}{9}$ ,  $y_2 = \frac{5}{3}$ ,  $y_3 = 0$  y  $\varepsilon = \frac{25}{9}$  de dos de estos sistemas. Pero para uno de los sistemas matemáticos, se obtiene la solución correspondiente a  $y_1 = 1.0423 \times 10^3$ ,  $y_2 = 1.0431 \times 10^3$ ,  $y_3 = 1.0414 \times 10^3$  y  $\varepsilon = 2.778$ . Hemos obtenido los mismos resultados que antes  $(0.8889, 1.6667) \approx (\frac{8}{9}, \frac{5}{3})$ .

En problemas como (6),  $m$  es a menudo mucho mayor que  $n$ . Así, de acuerdo con las observaciones formuladas en la sección 17.2, puede ser preferible para resolver el problema dual porque tendría  $2m$  variables, pero sólo  $n + 2$  desigualdades de restricción. Para ilustrar, el dual del problema (6) es

$$\left\{ \begin{array}{l} \text{maximizar: } 4u_1 + 2u_2 + 7u_3 - 4u_4 - 2u_5 - 7u_6 \\ \text{restricciones: } \begin{cases} 2u_1 + u_2 + u_3 - 2u_4 - u_5 - u_6 \geq 0 \\ 3u_1 - u_2 + 2u_3 - 3u_4 + u_5 - 2u_6 \geq 0 \\ -5u_1 - 3u_3 + 5u_4 + 3u_6 \geq 0 \\ -u_1 - u_2 - u_3 - u_4 - u_5 - u_6 \geq -1 \\ u_i \geq 0 \quad (1 \leq i \leq 6) \end{cases} \end{array} \right.$$

Los tres tipos de solución aproximada que se han analizado (para un sistema sobre determinado de ecuaciones lineales) son útiles en diferentes situaciones. En términos generales, una solución  $\ell_\infty$  es preferible cuando se sabe que los datos conocidos son precisos. Una solución  $\ell_2$  es preferible cuando los datos están contaminados con errores que se cree que se ajustan a la distribución de probabilidad normal. La solución  $\ell_1$  se utiliza a menudo cuando se sospecha que los datos contienen puntos *salvajes* que resultan de errores graves, tales como la colocación incorrecta de un punto decimal. Información adicional se puede encontrar en Rice y White [1964]. El problema  $\ell_2$  se analizó también en el capítulo 12.

## Resumen

---

(1) Consideramos un **sistema inconsistente** de  $m$  ecuaciones lineales con  $n$  incógnitas

$$\sum_{j=1}^n a_{ij}x_j = b_i \quad (1 \leq i \leq m)$$

Para los residuos  $r_i = \sum_{j=1}^n a_{ij}x_j - b_i$ , el **problema  $\ell_1$**  para este sistema es minimizar la expresión  $\sum_{i=1}^m |r_i|$ . Un nuevo enunciado directo del problema es

$$\left\{ \begin{array}{l} \text{minimizar: } \sum_{i=1}^m \varepsilon_i \\ \text{restricciones: } \begin{cases} \sum_{j=1}^n a_{ij}x_j - b_i \leq \varepsilon_i & (1 \leq i \leq m) \\ -\sum_{j=1}^n a_{ij}x_j + b_i \leq \varepsilon_i & (1 \leq i \leq m) \end{cases} \end{array} \right.$$

donde  $\varepsilon_i = |r_i|$ . Si las variables deben ser no negativas, se introduce una variable  $y_{n+1}$  y se escribe  $x_j = y_j - y_{n+1}$ . Se define  $a_{i,n+1} = -\sum_{j=1}^n a_{ij}$ ; una problema equivalente de programación

lineal es

$$\left\{ \begin{array}{l} \text{maximizar: } -\sum_{i=1}^m \varepsilon_i \\ \text{restricciones: } \begin{cases} \sum_{j=1}^{n+1} a_{ij}y_j - \varepsilon_i \leq b_i & (1 \leq i \leq m) \\ -\sum_{j=1}^{n+1} a_{ij}y_j - \varepsilon_i \leq -b_i & (1 \leq i \leq m) \\ y \geq 0 \quad \varepsilon \geq 0 \end{cases} \end{array} \right.$$

que está en la primera forma primal con  $m + n + 1$  variables y  $2m$  desigualdades de restricción.

(2) Otra técnica consiste en sustituir las  $2m$  desigualdades de restricción mediante conjunto de  $m$  igualdades de restricción. Escribimos  $\varepsilon_i = |r_i| = u_i + v_i$ , donde  $u_i = r_i$  y  $v_i = 0$  si  $r_i \geq 0$  pero  $v_i = -r_i$  y  $u_i = 0$  si  $r_i < 0$ . El problema de programación lineal resultante es

$$\left\{ \begin{array}{l} \text{maximizar: } -\sum_{i=1}^m u_i - \sum_{i=1}^m v_i \\ \text{restricciones: } \begin{cases} \sum_{j=1}^{n+1} a_{ij}y_j - u_i + v_i = b_i & (1 \leq i \leq m) \\ u \geq 0 \quad v \geq 0 \quad y \geq 0 \end{cases} \end{array} \right.$$

(3) Para un sistema inconsistente, el problema de hacer  $\varepsilon = \max_{1 \leq i \leq m} |r_i|$  un mínimo es el **problema  $\ell_\infty$**  para el sistema. Un problema de programación lineal equivalente es

$$\left\{ \begin{array}{l} \text{minimizar: } \varepsilon \\ \text{restricciones: } \begin{cases} \sum_{j=1}^n a_{ij}x_j - \varepsilon \leq b_i & (1 \leq i \leq m) \\ -\sum_{j=1}^n a_{ij}x_j - \varepsilon \leq -b_i & (1 \leq i \leq m) \end{cases} \end{array} \right.$$

Si las variables deben ser no negativas, se introduce una variable grande  $y_{n+1}$  para que las cantidades  $y_j = x_j + y_{n+1}$  sean positivos y tenemos un problema equivalente de programación lineal:

$$\left\{ \begin{array}{l} \text{minimizar: } \varepsilon \\ \text{restricciones: } \begin{cases} \sum_{j=1}^{n+1} a_{ij}y_j - \varepsilon \leq b_i & (1 \leq i \leq m) \\ -\sum_{j=1}^{n+1} a_{ij}y_j - \varepsilon \leq -b_i & (1 \leq i \leq m) \\ \varepsilon \geq 0 \quad y_j \geq 0 & (1 \leq j \leq n+1) \end{cases} \end{array} \right.$$

donde hemos definido  $a_{i,n+1} = -\sum_{j=1}^n a_{ij}$ .

## Referencias adicionales

Véase Armstrong y Godfrey [1979], Barrodale y Phillips [1975], Barrodale y Roberts [1974], Bartels [1971], Bloomfield y Steiger [1983], Branham [1990], Cärtner [2006], Cooper y Steinberg [1974] , Dantzi, Orden y Wolfe [1963], Huard [1979], Nering y Tucker [1992], Orchard-Hays [1968], Rabinowitz [1968], Roos et al. [1997], Schrijver [1986], Wright [1997], Ye [1997] y Zhang [1995].

### Problemas 17.3

1. Considere el sistema lineal inconsistente

$$\begin{cases} 5x_1 + 2x_2 = 6 \\ x_1 + x_2 + x_3 = 2 \\ 7x_2 - 5x_3 = 11 \\ 6x_1 + 9x_3 = 9 \end{cases}$$

Escriba las siguientes variables no negativas:

- <sup>a</sup>**a.** El problema equivalente de programación lineal para resolver el sistema en el sentido  $\ell_1$ .  
<sup>a</sup>**b.** El problema equivalente de programación lineal para resolver el sistema en el sentido  $\ell_\infty$ .

2. (Continuación) Repita el problema anterior para el sistema

$$\begin{cases} 3x + y = 7 \\ x - y = 11 \\ x + 6y = 13 \\ -x + 3y = -12 \end{cases}$$

- <sup>a</sup>**3.** Queremos encontrar un polinomio de grado  $n$  que se aproxime a una función  $f$  lo mejor posible *por abajo*, es decir, queremos  $0 \leq f - p \leq \varepsilon$  para  $\varepsilon$  mínima. Muestre cómo puede obtenerse  $p$  con una precisión razonable resolviendo un problema de programación lineal.

- <sup>a</sup>**4.** Para resolver el problema  $\ell_1$  para el sistema de ecuaciones

$$\begin{cases} x - y = 4 \\ 2x - 3y = 7 \\ x + y = 2 \end{cases}$$

podemos resolver un problema de programación lineal. ¿Cuál es este?

### Problemas de cómputo 17.3

- <sup>a</sup>**1.** Obtenga respuestas numéricicas para los incisos **a** y **b** del problema 17.3.1.  
**2.** (Continuación) Repita el problema 17.3.2.



# A

## Asesoramiento de buenas prácticas en programación

Ya que para programar métodos numéricos es esencial comprenderlos, aquí le ofrecemos algunos consejos sobre buenas prácticas de programación.

### A.1 Sugerencias de programación

Las sugerencias y las técnicas que aquí se presentan se deben considerar en contexto. No intentan abarcarlo todo y se han omitido algunas buenas sugerencias de programación para hacer un análisis breve. Nuestro propósito es motivarlo a estar atento a las consideraciones de eficiencia, economía, facilidad de lectura y errores de redondeo. Por supuesto, algunas de estas sugerencias y advertencias pueden variar de acuerdo con el lenguaje de programación que se utilice y sus características.

**Sea cuidadoso y correcto** Trate de escribir programas con cuidado y correctamente. Esto es de suma importancia.

**Utilice seudocódigo** Antes de comenzar la codificación, escriba el algoritmo con todo el detalle matemático que se utilizará en el *seudocódigo* como el utilizado en este libro. El seudocódigo sirve como un puente entre las matemáticas y el programa de cómputo. No necesita estar definido de manera formal, como se hace en un lenguaje de cómputo, pero debe contener suficiente detalle para que la implementación sea sencilla. Cuando escriba el seudocódigo, utilice un estilo que sea fácil de leer y entender. En cuanto al mantenimiento, debe ser fácil para una persona que no está familiarizada con el código leerlo y entender lo que hace.

**Revise una y dos veces** Revise todos los errores y omisiones del código antes de comenzar a editar en una terminal de computadora. Dedique tiempo para revisar el código antes de ejecutarlo para evitar que al correr el programa y cuando se muestre la salida, se descubra un error, corrija el error y repita el proceso hasta *el cansancio*.\*

\* En 1962, el cohete que transportaba la sonda espacial Mariner I a Venus se salió de ruta y después de sólo cinco minutos de vuelo fue destruido. Una investigación reveló que un solo renglón defectuoso del código Fortran ocasionó el desastre. Se escribió un punto en lugar de una coma en el código DO 5 T=1, 3, lo que ocasionó que el ciclo se ejecutara sólo una vez en lugar de tres veces. Se ha estimado que este único error tipográfico costó a la Administración de la Aeronáutica Nacional y del Espacio de los Estados Unidos \$18.5 millones de dólares! Para más detalles, consulte el material disponible en línea como en [www-aix.gsi.de/~giese/swr/mariner1.html](http://www-aix.gsi.de/~giese/swr/mariner1.html) y [www-aix.gsi.de/~giese/swr/literatur1.html](http://www-aix.gsi.de/~giese/swr/literatur1.html).

Entornos informáticos modernos pueden permitir al usuario realizar este proceso en sólo unos segundos, pero este consejo sigue siendo válido, sin más razón que es peligrosamente fácil escribir programas que pueden funcionar en una prueba sencilla, pero no en una forma más complicada. ¡Y ninguna tecla de función o ratón puede decir lo que está mal!

**Use casos de prueba** Despues de escribir el seudocódigo, revíselo y siga los cálculos usando lápiz y papel con un ejemplo típico y sencillo. Revise los casos límite, como los valores de las primera y segunda iteraciones en un ciclo y el procesamiento de los primeros y últimos elementos en una estructura de datos, pues con frecuencia se presentan errores embarazosos. Estos mismos casos muestra se pueden usar como la primera serie de casos de prueba en el equipo.

**Codifique en módulos** Construya un programa en pasos escribiendo y probando una serie de segmentos (subprogramas, procedimientos o funciones), es decir, escriba subtareas autónomas como rutinas separadas. Trate de mantener estos segmentos del programa razonablemente pequeños, siempre que sea posible de menos de una página, para que la lectura y la depuración sean más fáciles.

**Generalice un poco** Si el código se puede escribir para manejar una situación un poco más general, entonces en muchos casos, vale la pena el esfuerzo adicional de hacerlo. Un programa que fue escrito para sólo un conjunto particular de números debe ser completamente reescrito para otro conjunto. Por ejemplo, sólo se necesitan algunos enunciados adicionales para escribir un programa con un tamaño de paso arbitrario comparado con un programa en el que el tamaño de paso esté fijo numéricamente. Sin embargo, se debe tener cuidado de no introducir demasiada generalidad al código, ya que puede hacer que una tarea de programación sencilla sea demasiado complicada.

**Muestre resultados intermedios** Imprima o muestre resultados intermedios y mensajes de diagnóstico para ayudar en la depuración y en la comprensión de la operación del programa. Siempre reimprima los datos de entrada, a menos que no sea práctico hacerlo, como con una gran cantidad de datos. Utilizar el método de leer e imprimir las instrucciones libera al programador de errores asociados con la desalineación de los datos. No se necesitan complicados formatos de salida, pero se recomiendan algunas simples etiquetas de salida.

**Incluya mensajes de precaución** Un programa sólido siempre advierte al usuario de una situación que no está diseñada para manejarse. En general, escriba programas de manera que sean fáciles de depurar cuando el error parece inevitable.

**Use nombres de variables significativas** A menudo es útil asignar nombres significativos a las variables, ya que pueden tener un mayor valor mnemotécnico que las variables de una sola letra. Hay una confusión perenne entre el carácter  $\text{\O}$  (letra “o”) y el 0 (número cero) y entre la 1 (letra “ele”) y el 1 (número uno).

**Declare todas las variables** Todas las variables deben listarse en declaraciones de tipo en cada programa o segmento de programa. Asignaciones de tipo implícito se pueden ignorar cuando se escriben enunciados de declaración que incluyan todas las variables utilizadas. Históricamente, en Fortran, las variables que comienzan con  $I/i$ ,  $J/j$ ,  $K/k$ ,  $L/l$ ,  $M/m$  y  $N/n$  son variables de tipo entero y las que comienzan con las demás letras son variables reales de punto flotante. Puede ser una buena idea adoptar este sistema para que se pueda reconocer inmediatamente el tipo de una variable sin buscar su tipo en las declaraciones.

En este libro, presentamos algoritmos utilizando seudocódigo y por tanto no siempre se sigue este consejo.

**Incluya comentarios** Los comentarios dentro de una rutina son útiles para revelar en algún momento más adelante lo que hace el programa. No se necesitan grandes comentarios, pero le recomendamos que incluya un prólogo a cada programa o segmento del programa en donde explique la finalidad, las variables de entrada y salida y el algoritmo utilizado y que proporcione algunos comentarios entre los principales segmentos del código. Coloque en cada bloque de código un número considerable de espacios para mejorar la legibilidad. Insertar renglones de comentarios en blanco y espacios en blanco puede mejorar la legibilidad del código. Para ahorrar espacio, no hemos incluido ningún comentario en el seudocódigo en este libro.

**Utilice ciclos limpios** Nunca ponga declaraciones innecesarias dentro de los ciclos. Mueva expresiones y variables fuera de un ciclo si no dependen de él o no cambian. También, marcar los ciclos puede aumentar la legibilidad del código, en particular con los anidados. Utilice una instrucción no ejecutable como el final de un ciclo para que el código se pueda modificar fácilmente.

**Declare constantes que no cambian** Utilice una instrucción de parámetro para asignar valores a las constantes fundamentales. Los valores de los parámetros corresponden a las constantes que no cambian a lo largo de la rutina. Estas declaraciones de parámetros son fáciles de cambiar cuando se quiere volver a ejecutar el programa con diferentes valores. Además, aclaran el papel clave que desempeñan las constantes en el código y hacen que las rutinas sean más legibles y más fáciles de entender.

**Use estructuras de datos adecuadas** Use estructuras de datos que sean naturales para el problema en cuestión. Si el problema se adapta más fácilmente a una matriz tridimensional que a varios arreglos unidimensionales, entonces se debe utilizar una matriz tridimensional.

**Use arreglos de todo tipo** Los elementos de los arreglos, ya sea en una, dos o más dimensiones, se suelen almacenar en palabras consecutivas de la memoria. Puesto que el compilador puede asignar el valor de un índice para dos o más arreglos subindizados en un único valor de subíndice que se utiliza como indicador para determinar la ubicación de los elementos de almacenamiento, el uso de arreglos de dos o más dimensiones puede considerarse como notación conveniente para el usuario. Sin embargo, cualquier ventaja al utilizar sólo un arreglo bidimensional y la realización de cálculos de subíndices complicados es leve. Estos asuntos los maneja mejor el compilador.

**Utilice funciones integradas** En los lenguajes de programación científica, muchas funciones integradas de matemáticas están disponibles para funciones comunes, tales como *sen*, *log*, *exp*, *arc sen*, etcétera. Además, las funciones numéricas como *integer*, *real*, *complex* e *imaginary* suelen estar disponibles para la conversión de tipos. Se debe utilizar éstas y otras tanto como sea posible. Algunas de estas funciones intrínsecas aceptan los argumentos de más de un tipo y devuelven un resultado cuyo patrón puede variar dependiendo del tipo de argumentos utilizados. Tales funciones se llaman **funciones genéricas**, porque representan a toda una familia de funciones relacionadas. Por supuesto, se debe procurar no utilizar el tipo de argumento equivocado.

**Use bibliotecas de programas** De preferencia, al escribir un proyecto de programación, cuando sea aplicable se debe utilizar una rutina *preprogramada* de una biblioteca de programas.

Se puede esperar que dichas rutinas sean lo más moderno en software, bien probadas y, por supuesto, totalmente depuradas.

**No sobreoptimice** Los estudiantes deben ser los principales interesados en escribir códigos legibles que calculen correctamente los resultados deseados. Hay cualquier cantidad de trucos para hacer que el código sea más rápido o más eficiente. Guárdelos para utilizarlos posteriormente en su carrera de programador. Nos ocupamos principalmente de la comprensión y análisis de diversos métodos numéricos. No sacrifique la claridad de un programa en un esfuerzo por hacer funcionar el código más rápido. Puede ser preferible la claridad del código a la optimización de código cuando hay conflicto entre los dos criterios.

## Casos prácticos

Se presentan algunos casos prácticos que pueden ser útiles.

**Cálculo de sumas** Cuando se suma una gran lista de números de punto flotante en el equipo, generalmente habrá menos error de redondeo si los números se suman en orden de magnitud creciente. (Los errores de redondeo se analizan en detalle en el capítulo 2.)

**Constantes matemáticas** Algunos estudiantes se sorprenden al saber que en muchos lenguajes de programación, el equipo no conoce automáticamente los valores de las constantes matemáticas comunes como  $\pi$  y  $e$ , y que se deben decir explícitamente sus valores. Puesto que es fácil equivocarse al escribir una gran secuencia de dígitos en una constante matemática, tales como el número real  $\pi$ ,

$$pi \leftarrow 3.14159\ 26535\ 89793$$

se recomienda el uso de cálculos simples que implican funciones matemáticas. Por ejemplo, los números reales  $\pi$  y  $e$  pueden introducirse fácilmente y seguramente con casi total precisión de máquina utilizando las funciones normales intrínsecas tales como

$$\begin{aligned} pi &\leftarrow 4.0 \arctan(1.0) \\ e &\leftarrow \exp(1.0) \end{aligned}$$

Otra razón de este consejo es evitar el problema que se plantea si se utiliza una aproximación corta como  $pi \leftarrow 3.14159$  en una computadora con precisión limitada, pero posteriormente se mueve el código a otra computadora que tiene más precisión. Si usted omite cambiar este enunciado de asignación, entonces todos los resultados que dependan de este valor serán menos precisos de lo que deberían ser.

**Exponentes** En la codificación para la computadora, escriba con cuidado los enunciados que implican exponentes. La función general  $x^y$  se calcula en muchos equipos como  $\exp(y \ln x)$  siempre que  $y$  no sea un entero. A veces esto es innecesariamente complicado y puede contribuir a los errores de redondeo. Por ejemplo, es preferible escribir el código entero con exponentes como  $5$  en lugar de  $5.0$ . De manera similar, usar exponentes tales como  $\frac{1}{2}$  o  $0.5$ , no se recomienda porque se puede utilizar la función integrada *sqrt*.

Rara vez hay necesidad de un cálculo como  $j \leftarrow (-1)^k$  porque hay mejores maneras de obtener el mismo resultado. Por ejemplo, en un ciclo, podemos escribir  $j \leftarrow 1$  antes del ciclo y  $j \leftarrow -j$  dentro de él.

**Evite el modo mixto** En general, se debe evitar la mezcla de expresiones reales y enteras en el código informático. *Expresiones mixtas* son fórmulas en las que las variables y constantes de

diferentes tipos aparecen juntas. Si se necesita la forma de punto flotante de una variable entera, use una función *real*. Del mismo modo, una función como *integer* está generalmente disponible para obtener la parte entera de una variable real. En otras palabras, utilice funciones de conversión de tipo intrínseco cuando convierta de complejos a reales, de reales a enteros o viceversa. Por ejemplo, en cálculos de punto flotante,  $m/n$  se debe codificar como *real(m)/real(n)* cuando *m* y *n* son variables enteras, de modo que se calcule el valor real correcto de  $m/n$ . Del mismo modo,  $1/m$  se debe codificar como *1.0/real(m)* y  $1/2$  como  $0.5$  y así sucesivamente.

**Precisión** En el modo usual de representar números en una computadora, se utiliza una palabra de almacenamiento para cada número. Este modo de representación se llama **precisión simple**. En los cálculos que requieren mayor precisión (llamada **doble precisión** o **precisión extendida**), es posible asignar dos o más palabras de almacenamiento para cada número. En una computadora de 32 bits, aproximadamente se pueden obtener siete decimales de precisión con precisión simple y aproximadamente 17 decimales con doble precisión. Con doble precisión normalmente se consume más tiempo que con precisión simple, ya que puede utilizar software en lugar de hardware para realizar la aritmética. Sin embargo, si se necesita más precisión que la que puede proporcionar la precisión simple, entonces se debe utilizar precisión doble o extendida. Esto es particularmente cierto en computadoras con precisión limitada, tal como una computadora de 32 bits, en la que los errores de redondeo pueden acumularse con rapidez en cálculos largos y se reduce la precisión ¡a sólo tres o cuatro cifras decimales! (Este tema se analizó en el capítulo 2.)

Por lo general, se utilizan dos palabras de memoria para almacenar las partes real e imaginaria de un número complejo. Las variables complejas y los arreglos se deben declarar explícitamente como de tipo complejo. Las expresiones que implican variables y constantes de tipo complejo se evalúan de acuerdo con las reglas normales de la aritmética compleja. Las funciones intrínsecas, tales como *complex*, *real* e *imaginary* se deben utilizar para convertir entre los tipos reales y complejos.

**Búsquedas de memoria** Cuando use ciclos, escriba el código de manera que las búsquedas se hagan en palabras *adyacentes* en la memoria. Como ejemplo, supongamos que queremos almacenar los valores en un arreglo bidimensional ( $a_{ij}$ ) en el que los elementos de cada columna se almacenan en las ubicaciones de memoria consecutivas. Usando los ciclos *i* y *j* con el *i*-ésimo ciclo como el más interno se establecen los elementos de las columnas. Para algunos programas y lenguajes de programación, este detalle puede ser sólo de preocupación secundaria. Sin embargo, algunas computadoras tienen acceso inmediato a sólo una parte o a unas cuantas *páginas* de memoria a la vez. En este caso, es ventajoso procesar los elementos de una matriz para que se consideren o almacenen en las localidades de memoria adyacentes.

**Cuando evitar arreglos** Aunque la descripción matemática de un algoritmo puede indicar que se calcula una sucesión de valores, y por tanto parece implicar la necesidad de un arreglo, con frecuencia es posible evitar arreglos (esto es especialmente cierto si sólo se requiere el valor final de una sucesión). Por ejemplo, la descripción teórica del método de Newton (capítulo 3) se escribe como

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

pero el seudocódigo se puede escribir dentro de un ciclo simplemente como

```
for n = 1 to 10 do
 x ← x - f(x)/f'(x)
end for
```

donde  $x$  es una variable real y se han escrito los procedimientos de la función para  $f$  y  $f'$ . Esta declaración de asignación automáticamente efectúa la sustitución del valor *viejo* de  $x$  por el *nuevo* valor numérico de  $x - f(x)/f'(x)$ .

**Límite las iteraciones** En un algoritmo repetitivo, siempre se debe limitar el número de pasos permitidos por el uso de un ciclo con una variable de control. Esto impedirá repeticiones sin fin debido a problemas imprevistos (por ejemplo, errores de programación y errores de redondeo). Por ejemplo, en el método de Newton anterior, se podría escribir

```

 $d \leftarrow f(x)/f'(x)$
while $|d| > \frac{1}{2} \times 10^{-6}$ do
 $x \leftarrow x - d$
 output x
 $d \leftarrow f(x)/f'(x)$
end while
```

Si la función implica algún comportamiento irregular, existe un peligro al no limitar el número de repeticiones. Es mejor utilizar un ciclo con una variable de control:

```

for $n = 1$ to n_max do
 $d \leftarrow f(x)/f'(x)$
 $x \leftarrow x - d$
 output n, x
 if $|d| \leq \frac{1}{2} \times 10^{-6}$ then exit loop
end for
```

donde  $n$  y  $n\_max$  son variables enteras y el valor de  $n\_max$  es un límite superior del número de repeticiones deseado. Todas las demás son variables reales.

**Igualdad de punto flotante** La secuencia de pasos en una rutina no debe depender de si dos números de punto flotante son iguales. En cambio, se deben permitir razonables tolerancias aritméticas de punto flotante en errores de redondeo. Por ejemplo, una declaración de ramificación adecuada para  $n$  dígitos decimales de precisión puede ser

```
if $|x - y| < \varepsilon$ then ... end if
```

suponiendo que se conoce que  $x$  y  $y$  tienen una magnitud comparable a 1. Aquí  $x$ ,  $y$  y  $\varepsilon$  son variables reales con  $\varepsilon = \frac{1}{2} \times 10^{-n}$ . Esto corresponde a la exigencia de que el *error absoluto* entre  $x$  y  $y$  es menor que  $\varepsilon$ . Sin embargo, si  $x$  y  $y$  tienen órdenes de magnitud muy grandes o muy pequeños, entonces se necesitaría el *error relativo* entre  $x$  y  $y$ , como en la declaración de ramificación

```
if $|x - y| < \varepsilon \max\{|x|, |y|\}$ then ... end if
```

**Pasos iguales de punto flotante** En algunas situaciones, especialmente en la solución de ecuaciones diferenciales (capítulo 8), una variable  $t$  supone una sucesión de valores igualmente espaciados una distancia  $h$  a lo largo de la recta real. Una manera de codificar esto es

```

 $t \leftarrow t_0$
output $0, t$
for $i = 1$ to n do
 :
 $t \leftarrow t + h$
 output i, t
end for

```

Aquí,  $i$  y  $n$  son variables de tipo entero, y  $t_0$ ,  $t$  y  $h$  son las variables reales. Una forma alternativa es

```

for $i = 0$ to n do
 :
 $t \leftarrow t_0 + \text{real}(i)h$
 output i, t
end for

```

En el primer seudocódigo se presentan  $n$  sumas, cada una con un error de redondeo posible. En el segundo, se evita esta situación pero a cambio de agregar  $n$  multiplicaciones. La mejor depende de la situación que se encare.

**Evaluaciones de la función** Cuando se necesitan los valores de una función en puntos arbitrarios en un programa, hay varias formas de codificar. Por ejemplo, supongamos que se necesitan los valores de la función

$$f(x) = 2x + \ln x - \sin x$$

Un método sencillo es utilizar una instrucción de asignación como

$$y \leftarrow 2x + \ln(x) - \sin(x)$$

en lugares apropiados dentro del programa. Aquí,  $x$  y  $y$  son variables reales. De manera equivalente, un procedimiento de función *interno* que corresponde al seudocódigo

$$f(x) \leftarrow 2x + \ln(x) - \sin(x)$$

podría ser evaluado en 2.5 por medio de

$$y \leftarrow f(2.5)$$

o en cualquier valor de  $x$  que se desee. Por último, un subprograma de función se puede utilizar como en el siguiente seudocódigo:

```

real function $f(x)$
real x
 $f \leftarrow 2x + \ln(x) - \sin(x)$
end function f

```

¿Qué implementación es la mejor? Depende de la situación. El enunciado de asignación es simple y seguro. Se puede utilizar un procedimiento de función interno o externo para evitar la

la duplicación del código. Un subprograma de función externo independiente es la mejor manera de evitar dificultades que inadvertidamente se producen cuando alguien tiene que insertar código en otro programa. En el uso de rutinas de biblioteca del programa, el usuario puede ser obligado a proporcionar un procedimiento de función externo para comunicar los valores de la función a la rutina de la biblioteca. Si el procedimiento externo de la función  $f$  se pasa como un argumento a otro procedimiento, entonces se debe utilizar una *interfaz* especial para designarla como una función externa.

## Desarrollo de software matemático

Fred Krogh [2003] ha escrito un artículo que enumera algunas de las cosas que ha aprendido de su carrera en el Jet Propulsion Laboratory que implica el desarrollo y la escritura de software matemático utilizando paquetes de aplicaciones. Algunas de sus sugerencias y pensamientos elegidos aleatoriamente que conviene recordar cuando se desarrolla un código son los siguientes: incluya salidas internas, para ver lo que su algoritmo está haciendo; apoye la depuración incluyendo salidas en las interfaces y proporcione mensajes de error detallados; afine su código; proporcione casos de prueba comprensibles; verifique los resultados con cuidado, aproveche sus errores; mantenga unidades consistentes; pruebe los valores extremos; el objeto del algoritmo; trabaje en lo que hace; tire lo que no funciona; no se rinda antes de tiempo respecto a las ideas o depure su código; su subconsciente es una herramienta poderosa, aprenda a usarlo; pruebe su hipótesis; en los comentarios, mantenga un diccionario de variables en orden alfabético, ya que es bastante útil cuando se busca en un código años después de que se ha escrito; escriba primero la documentación del usuario; conozca los resultados que espera obtener, no preste demasiada atención a los demás, sólo la necesaria; vea los contratiempos como oportunidades de aprendizaje y para mantener el espíritu en alto; cuando compare los códigos, no cambie sus características o capacidades para hacer la comparación justa, ya que puede no comprender plenamente el código de otra persona; mantenga listas de acciones; categorice las características del código; organice las cosas en grupos; la organización del código puede ser una de las más importantes decisiones del programador; separe las partes de álgebra lineal del código en un paquete de aplicación para que el usuario pueda hacerles modificaciones; la *comunicación inversa* es una característica útil que permite a los usuarios salir del código y realizar operaciones matriz-vector utilizando su propia estructura de datos; guarde y restaure las variables cuando el usuario está autorizado para salir del código y regresar; la portabilidad es más importante que la eficiencia. Esto es sólo una muestra aleatoria de algunos de los elementos en dicho artículo.

# Representación de números en diferentes bases

En este apéndice se revisan algunos conceptos básicos acerca de la representación de números en diferentes bases.

## B.1 Representación de números en diferentes bases

Empezamos con un análisis de la representación de un número general, pero nos movemos rápidamente a las bases de 2, 8 y 16, ya que son las que se utilizan principalmente en la aritmética de la computadora.

La notación decimal para números utiliza los dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. Cuando se escribe un número entero como 37294, los dígitos individuales representan coeficientes de las potencias de 10 como se muestra a continuación:

$$\begin{aligned} 37294 &= 4 + 90 + 200 + 7000 + 30000 \\ &= 4 \times 10^0 + 9 \times 10^1 + 2 \times 10^2 + 7 \times 10^3 + 3 \times 10^4 \end{aligned}$$

Así, en general, una cadena de dígitos representa un número de acuerdo con la fórmula

$$a_n a_{n-1} \dots a_2 a_1 a_0 = a_0 \times 10^0 + a_1 \times 10^1 + \dots + a_{n-1} \times 10^{n-1} + a_n \times 10^n$$

Ésta sólo se ocupa para los números enteros positivos. Un número entre 0 y 1 se representa mediante una cadena de dígitos a la derecha de un punto decimal. Por ejemplo, vemos que

$$\begin{aligned} 0.7215 &= \frac{7}{10} + \frac{2}{100} + \frac{1}{1000} + \frac{5}{10000} \\ &= 7 \times 10^{-1} + 2 \times 10^{-2} + 1 \times 10^{-3} + 5 \times 10^{-4} \end{aligned}$$

En general, tenemos la fórmula

$$0.b_1 b_2 b_3 \dots = b_1 \times 10^{-1} + b_2 \times 10^{-2} + b_3 \times 10^{-3} + \dots$$

Observe que puede haber una cadena infinita de dígitos a la derecha del punto decimal, de hecho, *debe* haber una cadena infinita para representar algunos números. Por ejemplo, notamos que

$$\sqrt{2} = 1.41421\ 35623\ 73095\ 04880\ 16887\ 24209\ 69\dots$$

$$e = 2.71828\ 18284\ 59045\ 23536\ 02874\ 71352\ 66\dots$$

$$\pi = 3.14159\ 26535\ 89793\ 23846\ 26433\ 83279\ 50\dots$$

$$\ln 2 = 0.69314\ 71805\ 59945\ 30941\ 72321\ 21458\ 17\dots$$

$$\frac{1}{3} = 0.33333\ 33333\ 33333\ 33333\ 33333\ 33\dots$$

Para un número real de la forma

$$(a_n a_{n-1} \dots a_1 a_0.b_1 b_2 b_3 \dots)_{10} = \sum_{k=0}^n a_k 10^k + \sum_{k=1}^{\infty} b_k 10^{-k}$$

la **parte entera** es la primera suma en el desarrollo y la **parte fraccionaria** es la segunda suma. Si puede parecer ambiguo, un número representado en base  $\beta$  se representa entre paréntesis y se le agrega un subíndice  $\beta$ .

## Base $\beta$ de números

El análisis anterior se refiere a la representación habitual de los números con base 10. También se utilizan otras bases, sobre todo en las computadoras. Por ejemplo, el sistema **binario** utiliza 2 como la base, el sistema **octal** utiliza 8 y el sistema **hexadecimal** utiliza 16.

En la representación octal de un número, los dígitos que se usan son 0, 1, 2, 3, 4, 5, 6 y 7. Así, vemos que

$$\begin{aligned}(21467)_8 &= 7 + 6 \times 8 + 4 \times 8^2 + 1 \times 8^3 + 2 \times 8^4 \\ &= 7 + 8(6 + 8(4 + 8(1 + 8(2)))) \\ &= 9015\end{aligned}$$

Un número entre 0 y 1, expresado en octal, se representa con las combinaciones de  $8^{-1}$ ,  $8^{-2}$  y así sucesivamente. Por ejemplo, tenemos

$$\begin{aligned}(0.36207)_8 &= 3 \times 8^{-1} + 6 \times 8^{-2} + 2 \times 8^{-3} + 0 \times 8^{-4} + 7 \times 8^{-5} \\ &= 8^{-5}(3 \times 8^4 + 6 \times 8^3 + 2 \times 8^2 + 7) \\ &= 8^{-5}(7 + 8^2(2 + 8(6 + 8(3)))) \\ &= \frac{15495}{32768} \\ &= 0.47286987\dots\end{aligned}$$

Vamos a ver cómo convertir fácilmente a la forma decimal, sin tener que encontrar un denominador común.

Si usamos otra base, por ejemplo,  $\beta$ , entonces los números representados en el sistema  $\beta$  se representan como:

$$(a_n a_{n-1} \dots a_1 a_0.b_1 b_2 b_3 \dots)_{\beta} = \sum_{k=0}^n a_k \beta^k + \sum_{k=1}^{\infty} b_k \beta^{-k}$$

Los dígitos son 0, 1, ...,  $\beta - 2$  y  $\beta - 1$  en esta representación. Si  $\beta > 10$ , es necesario introducir símbolos para 10, 11, ...,  $\beta - 1$ . La separación entre la parte entera y parte fraccionaria se llama el **punto de base**, ya que el **punto decimal** está reservado para los números de base 10.

## Conversión de partes enteras

Ahora formalizaremos el proceso de conversión de un número de una base a otra. Es conveniente examinar por separado las partes entera y fraccionaria de un número. Consideremos, entonces, un número entero positivo  $N$  en el sistema numérico de base  $\gamma$ :

$$N = (a_n a_{n-1} \dots a_1 a_0)_{\gamma} = \sum_{k=0}^n a_k \gamma^k$$

Supongamos que queremos convertirlo al sistema numérico de base  $\beta$  y que los cálculos se realizarán en aritmética de base  $\beta$ . Escribimos  $N$  en su forma anidada:

$$N = a_0 + \gamma(a_1 + \gamma(a_2 + \cdots + \gamma(a_{n-1} + \gamma(a_n)) \cdots))$$

y después sustituimos cada uno de los números de la derecha por su representación en base  $\beta$ . A continuación, realizamos los cálculos en aritmética  $\beta$ . La sustitución de las  $a_k$  y  $\gamma$  por números equivalentes de base  $\beta$  requiere una tabla que muestre cómo se representan cada uno de los números  $0, 1, \dots, \gamma - 1$  en el sistema  $\beta$ . Además, se puede requerir una tabla de multiplicar de base  $\beta$ .

Para ilustrar este procedimiento, considere la conversión del número decimal 3781 a la forma binaria. Usando las equivalencias decimales binarias y la multiplicación a mano en base 2, tenemos

$$\begin{aligned} (3781)_{10} &= 1 + 10(8 + 10(7 + 10(3))) \\ &= (1)_2 + (1010)_2((1000)_2 + (1010)_2((111)_2 + (1010)_2(11)_2)) \\ &= (111\ 011\ 000\ 101)_2 \end{aligned}$$

Este cálculo aritmético en binario es fácil para una computadora que opera en binario, pero tedioso para los seres humanos.

Se debe utilizar otro procedimiento para los cálculos a mano. Escriba una ecuación que contenga los dígitos  $c_0, c_1, \dots, c_m$  que buscamos:

$$N = (c_m c_{m-1} \dots c_1 c_0)_\beta = c_0 + \beta(c_1 + \beta(c_2 + \cdots + \beta(c_m) \cdots))$$

Después, observe que si  $N$  se divide entre  $\beta$ , entonces el *residuo* de esta división es  $c_0$  y el *cociente* es

$$c_1 + \beta(c_2 + \cdots + \beta(c_m) \cdots)$$

Si este número se divide entre  $\beta$ , el residuo es  $c_1$  y así sucesivamente. Por lo tanto, se divide repetidamente entre  $\beta$ , guardando los residuos  $c_0, c_1, \dots, c_m$  y los cocientes.

**EJEMPLO 1** Convierta el número decimal 3781 a la forma binaria utilizando el algoritmo de la división.

**Solución** Como se indicó antes, se divide repetidamente entre 2, guardando los residuos en el camino. Aquí se presenta el trabajo:

**Cocientes   Residuos**

$$\begin{array}{r} 2 ) 3781 \\ 2 ) 1890 \quad 1 = c_0 \qquad \downarrow \\ 2 ) 945 \quad 0 = c_1 \\ 2 ) 472 \quad 1 = c_2 \\ 2 ) 236 \quad 0 = c_3 \\ 2 ) 118 \quad 0 = c_4 \\ 2 ) 59 \quad 0 = c_5 \\ 2 ) 29 \quad 1 = c_6 \\ 2 ) 14 \quad 1 = c_7 \\ 2 ) 7 \quad 0 = c_8 \\ 2 ) 3 \quad 1 = c_9 \\ 2 ) 1 \quad 1 = c_{10} \\ 0 \quad 1 = c_{11} \end{array}$$

Aquí, el símbolo  $\downarrow$  se utiliza para recordarnos que los dígitos  $c_i$  se obtienen a partir del dígito siguiente al punto binario. Por lo tanto, tenemos

$$(3781.)_{10} = (111\ 011\ 000\ 101.)_2$$

y no al revés:  $(101\ 000\ 110\ 111.)_2 = (2615)_{10}$ . ■

**EJEMPLO 2** Convierta el número  $N = (111\ 011\ 000\ 101)_2$  a la forma decimal con multiplicación anidada.

Solución

$$\begin{aligned} N &= 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 \\ &\quad + 1 \times 2^6 + 1 \times 2^7 + 0 \times 2^8 + 1 \times 2^9 + 1 \times 2^{10} + 1 \times 2^{11} \\ &= 1 + 2(0 + 2(1 + 2(0 + 2(0 + 2(0 + 2(1 + 2(1 + 2(0 \\ &\quad + 2(1 + 2(1 + 2(1))))))))))) \\ &= 3781 \end{aligned}$$

La *multiplicación anidada* con multiplicación repetida y la suma se pueden realizar con una calculadora manual más fácilmente que la forma anterior con exponentiación. ■

Existe otro problema en la conversión al pasar un entero de base  $\gamma$  a un número entero de base  $\beta$  utilizando cálculos de base  $\gamma$ . Como antes, se determinan los coeficientes desconocidos en la ecuación

$$N = c_0 + c_1\beta + c_2\beta^2 + \cdots + c_m\beta^m$$

mediante un proceso de divisiones sucesivas y este cálculo se realiza en el sistema  $\gamma$ . Al final, los números  $c_k$  están en la base  $\gamma$  y se utiliza una tabla de equivalencias  $\gamma$ - $\beta$ .

Por ejemplo, podemos convertir un entero binario a decimal dividiendo repetidamente entre  $(1010)_2$  [que es igual a  $(10)_{10}$ ], realizando las operaciones en binario. En el paso final se utiliza una tabla de equivalencias binario-decimal. Sin embargo, puesto que la división binaria es fácil sólo para las computadoras, vamos a desarrollar procedimientos alternativos actuales.

## Conversión de partes fraccionarias

Podemos convertir un número fraccionario, como  $(0.372)_{10}$  a binario utilizando el siguiente método simple y directo:

$$\begin{aligned} (0.372)_{10} &= 3 \times 10^{-1} + 7 \times 10^{-2} + 2 \times 10^{-3} \\ &= \frac{1}{10} \left( 3 + \frac{1}{10} \left( 7 + \frac{1}{10} (2) \right) \right) \\ &= \frac{1}{(1010)_2} \left( (011)_2 + \frac{1}{(1010)_2} \left( (111)_2 + \frac{1}{(1010)_2} (010)_2 \right) \right) \end{aligned}$$

La división en la aritmética binaria no es sencilla, por lo que buscamos maneras más fáciles de hacer esta conversión.

Supongamos que  $x$  está en el rango  $0 < x < 1$  y que los dígitos  $c_k$  en la representación

$$x = \sum_{k=1}^{\infty} c_k \beta^{-k} = (0.c_1c_2c_3\dots)_{\beta}$$

están por determinarse. Observe que

$$\beta x = (c_1.c_2c_3c_4\dots)_{\beta}$$

ya que es necesario cambiar el punto de base sólo cuando se multiplica por la base  $\beta$ . Así, el dígito desconocido  $c_1$  se puede describir como la **parte entera** de  $\beta x$ . Se denota por  $\mathcal{I}(\beta x)$ . La **parte fraccionaria**  $(0.c_2c_3c_4\dots)_\beta$ , se denota por  $\mathcal{F}(\beta x)$ . El proceso se repite siguiendo el patrón:

$$\begin{aligned}d_0 &= x \\d_1 &= \mathcal{F}(\beta d_0) & c_1 = \mathcal{I}(\beta d_0) & \downarrow \\d_2 &= \mathcal{F}(\beta d_1) & c_2 = \mathcal{I}(\beta d_1) \\&&\text{etc.}\end{aligned}$$

En este algoritmo la aritmética se realiza en el sistema decimal.

**EJEMPLO 3** Use el algoritmo anterior para convertir el número decimal  $x = (0.372)_{10}$  a la forma binaria.

**Solución** El algoritmo consiste en multiplicar repetidamente por 2 eliminando las partes enteras. Aquí se presenta el trabajo:

$$\begin{array}{r} 0.372 \\ \downarrow \qquad c_1 = \overline{0.744}^2 \\ c_2 = \overline{1.488}^2 \\ c_3 = \overline{0.976}^2 \\ c_4 = \overline{1.952}^2 \\ c_5 = \overline{1.904}^2 \\ c_6 = \overline{1.808}^2 \\ \text{etc.} \end{array}$$

Por lo tanto, tenemos  $(0.372)_{10} = (0.010111\dots)_2$ .



## Conversión de base 10 ↔ 8 ↔ 2

La mayoría de las computadoras usan el sistema binario (base 2) para su representación numérica interna. El sistema octal (base 8) es particularmente útil en la conversión del sistema decimal (de base 10) al sistema binario, y viceversa. Con base 8, el valor posicional de los números son  $8^0 = 1$ ,  $8^1 = 8$ ,  $8^2 = 64$ ,  $8^3 = 512$ ,  $8^4 = 4096$  y así sucesivamente. Así, por ejemplo, tenemos

$$\begin{aligned}(26031)_8 &= 2 \times 8^4 + 6 \times 8^3 + 0 \times 8^2 + 3 \times 8 + 1 \\&= (((2)8 + 6)8 + 0)8 + 3)8 + 1 \\&= 11289\end{aligned}$$

y

$$\begin{aligned}(7152.46)_8 &= 7 \times 8^3 + 1 \times 8^2 + 5 \times 8 + 2 + 4 \times 8^{-1} + 6 \times 8^{-2} \\&= (((7)8 + 1)8 + 5)8 + 2 + 8^{-2}[(4)8 + 6] \\&= 3690 + \frac{38}{64} \\&= 3690.59375\end{aligned}$$

Cuando los números se convierten a mano entre la forma decimal y la binaria, es conveniente utilizar la representación octal como un paso intermedio. En el sistema octal, la base es 8 y, por supuesto, los dígitos 8 y 9 no se utilizan. La conversión entre octal y decimal se realiza de acuerdo con los principios ya establecidos. La conversión entre octal y binario es especialmente sencilla. Los grupos de tres dígitos binarios se pueden traducir directamente a octal de acuerdo con la tabla siguiente:

|         |     |     |     |     |     |     |     |     |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|
| Binario | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Octal   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   |

Este agrupamiento inicia en el punto binario y se continúa en ambas direcciones. Por lo tanto, tenemos

$$(101\ 101\ 001.110\ 010\ 100)_2 = (551.624)_8$$

Para justificar esta prestidigitación manual conveniente, consideremos, por ejemplo, una fracción expresada en forma binaria:

$$\begin{aligned}x &= (0.b_1b_2b_3b_4b_5b_6\dots)_2 \\&= b_12^{-1} + b_22^{-2} + b_32^{-3} + b_42^{-4} + b_52^{-5} + b_62^{-6} + \dots \\&= (4b_1 + 2b_2 + b_3)8^{-1} + (4b_4 + 2b_5 + b_6)8^{-2} + \dots\end{aligned}$$

En el último renglón de esta ecuación, los paréntesis encierran números del conjunto  $(0, 1, 2, 3, 4, 5, 6, 7)$ , ya que los  $b_i$  son 0 o 1. Por lo tanto, ésta debe ser la representación octal de  $x$ .

La conversión de un número octal a binario se puede hacer de una manera similar, pero en orden inverso. ¡Es fácil! Basta con sustituir cada dígito octal con los correspondientes tres dígitos binarios. Así, por ejemplo,

$$(5362.74)_8 = (101\ 011\ 110\ 010.111\ 100)_2$$

**EJEMPLO 4** ¿Cómo es  $(2576.35546\ 875)_{10}$  en forma octal y binaria?

**Solución** Convertimos primero el número decimal original a octal y después a binario. Para la parte entera, dividimos repetidamente entre 8:

$$\begin{array}{r} 8 ) 2576 \\ 8 ) 322 \quad 0 \quad \downarrow \\ 8 ) 40 \quad 2 \\ 8 ) 5 \quad 0 \\ 0 \quad 5 \end{array}$$

Por lo tanto, tenemos

$$2576. = (5020.)_8 = (101\ 000\ 010\ 000.)_2$$

utilizando las reglas de agrupamiento de dígitos binarios. Para la parte fraccionaria, multiplicamos repetidamente por 8

$$\begin{array}{r} 0.35546875 \\ \times 8 \\ \hline 2.84375000 \\ \times 8 \\ \hline 6.75000000 \\ \times 8 \\ \hline 6.00000000 \end{array}$$

de modo que

$$0.35546\,875 = (0.266)_8 = (0.010\,110\,110)_2$$

Por último, se obtiene el resultado

$$2576.35546\,875 = (101\,000\,010\,000.010\,110\,110)_2$$

Aunque este método es más largo para este ejemplo, creemos que es más fácil en general y tiene menos probabilidades de inducir un error, porque se está trabajando la mayor parte del tiempo con números de un solo dígito. ■

## Base 16

Algunas computadoras cuyas longitudes de palabra son múltiplos de 4 utilizan el sistema hexadecimal (base 16) en el que A, B, C, D, E y F representan 10, 11, 12, 13, 14 y 15, respectivamente, como se presenta en la siguiente tabla de equivalencias:

|             |      |      |      |      |      |      |      |      |
|-------------|------|------|------|------|------|------|------|------|
| Hexadecimal | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    |
| Binario     | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| Hexadecimal | 8    | 9    | A    | B    | C    | D    | E    | F    |
| Binario     | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

La conversión entre números binarios y números hexadecimales es particularmente fácil. Sólo tenemos que reagrupar los dígitos binarios de grupos de tres en grupos de cuatro. Por ejemplo, tenemos

$$(010\,101\,110\,101\,101)_2 = (0010\,1011\,1010\,1101)_2 = (2BAD)_{16}$$

y

$$\begin{aligned} (111\,101\,011\,110\,010.\,110\,010\,011\,110)_2 &= (1010\,1111\,0010.\,1100\,1001\,1110)_2 \\ &= (7AF2.C9E)_{16} \end{aligned}$$

## Más ejemplos

Siguiendo con más ejemplos, convertimos  $(0.276)_8$ ,  $(0.C8)_{16}$  y  $(492)_{10}$  en diferentes sistemas de números. Se muestra una forma para cada número y se invita al lector a trabajar en los detalles de las otras formas y a comprobar las respuestas convirtiéndolas de nuevo a la base original.

$$\begin{aligned} (0.276)_8 &= 2 \times 8^{-1} + 7 \times 8^{-2} + 6 \times 8^{-3} \\ &= 8^{-3}[(2)8 + 7]8 + 6 \\ &= (0.37109\,375)_{10} \end{aligned}$$

$$\begin{aligned} (0.C8)_{16} &= (0.110\,010)_2 \\ &= (0.62)_8 \\ &= 6 \times 8^{-1} + 2 \times 8^{-2} \\ &= 8^{-2}[(6)8 + 2] \\ &= (0.78125)_{10} \end{aligned}$$

$$\begin{aligned}
 (492)_{10} &= (754)_8 \\
 &= (111\ 101\ 100)_2 \\
 &= (1EC)_{16}
 \end{aligned}$$

porque

$$\begin{array}{r}
 8) \overline{492} \\
 8) \overline{61} \quad 4 \\
 8) \overline{7} \quad 5 \\
 \hline
 0 \quad 7
 \end{array}
 \quad \downarrow$$

## Resumen

**(1)** Podría parecer que hay diferentes procedimientos para la conversión entre sistemas numéricos. En realidad, sólo hay *dos* técnicas básicas. El primer procedimiento para convertir el número  $(N)_\gamma$  a la base  $\beta$  se puede describir en la forma siguiente:

- Exprese  $(N)_\gamma$  en forma anidada usando las potencias de  $\gamma$ .
- Reemplace cada dígito por los números correspondientes de base  $\beta$ .
- Realice las operaciones aritméticas indicadas en la base  $\beta$ .

Este esquema vale si  $N$  es un entero o una fracción. El segundo procedimiento es ya sea dividir entre  $\beta$  y realizar el *proceso de separar el cociente y el residuo* para un número entero  $N$  o multiplicar por  $\beta$  y realizar el proceso de separar en entero y fracción para una fracción  $N$ . Se prefiere el primer procedimiento cuando  $\gamma < \beta$  y el segundo cuando  $\gamma > \beta$ . Por supuesto, el procedimiento de conversión de base  $10 \leftrightarrow 8 \leftrightarrow 2 \leftrightarrow 16$  se debe usar siempre que sea posible, porque es la forma más fácil de convertir números entre los sistemas decimal, octal, binario o hexadecimal.

## Problemas B.1

**1.** Encuentre la representación binaria y compruébela al reconvertir a la representación decimal.

<sup>a</sup>**a.**  $e \approx (2.718)_{10}$     **b.**  $\frac{7}{8}$     **c.**  $(592)_{10}$

**2.** Convierta los siguientes números decimales a números octales.

**a.** 27.1    **b.** 12.34    **c.** 3.14    **d.** 23.58    **e.** 75.232    **f.** 57.321

**3.** Convierta a hexadecimal, a octal y después a decimal.

<sup>a</sup>**a.**  $(110\ 111\ 001.101\ 011\ 101)_2$     **b.**  $(1\ 001\ 100\ 101.011\ 01)_2$

**4.** Convierta los siguientes números:

**a.**  $(100\ 101\ 101)_2 = (\quad)_8 = (\quad)_{10}$

**b.**  $(0.782)_{10} = (\quad)_8 = (\quad)_2$

<sup>a</sup>**c.**  $(47)_{10} = (\quad)_8 = (\quad)_2$

**d.**  $(0.47)_{10} = (\quad)_8 = (\quad)_2$

- a.** e.  $(51)_{10} = (\quad)_8 = (\quad)_2$
- f.**  $(0.694)_{10} = (\quad)_8 = (\quad)_2$
- g.**  $(110011.1110101101101)_2 = (\quad)_8 = (\quad)_{10}$
- h.**  $(361.4)_8 = (\quad)_2 = (\quad)_{10}$
- 5.** Convierta  $(45653.127664)_8$  a binario y a decimal.
- 6.** Convierta  $(0.4)_{10}$  primero a octal y después a binario. Compruebe convirtiendo directamente a binario.
- 7.** Demuestre que el número decimal  $\frac{1}{5}$  no se puede representar con un desarrollo finito en el sistema binario.
- 8.** ¿Espera que su computadora calcule  $3 \times \frac{1}{3}$  con precisión infinita? ¿Qué hay con  $2 \times \frac{1}{2}$  o  $10 \times \frac{1}{10}$ ?
- 9.** Explique el algoritmo para la conversión de un entero en base 10 a uno en base 2, suponiendo que los cálculos se realizan en aritmética binaria. Ilustre convirtiendo  $(479)_{10}$  a binario.
- 10.** Justifique matemáticamente la conversión entre números binarios y hexadecimales reagrupando.
- 11.** Justifique para los enteros la regla dada para la conversión entre los números octales y los binarios.
- 12.** Demuestre que un número real tiene una representación finita en el sistema numérico binario si y sólo si es de la forma  $\pm m/2^n$ , donde  $n$  y  $m$  son números enteros positivos.
- 13.** Demuestre que cualquier número que tiene una representación finita en el sistema binario debe tener una representación finita en el sistema decimal.
- 14.** Algunos países miden la temperatura en grados Fahrenheit (F), mientras que otros utilizan grados Celsius (C). Del mismo modo, para la distancia, algunos usan millas y otros utilizan kilómetros. Como viajero frecuente, puede tener necesidad de un método rápido de conversión aproximada que pueda realizar mentalmente.
- Fahrenheit y Celsius están relacionados por la ecuación  $F = 32 + (9/5)C$ . Compruebe el siguiente método de conversión simple para pasar de Celsius a Fahrenheit: una aproximación es el doble de la temperatura en grados Celsius y se suma 32. Para refinar su aproximación, corra el punto decimal a la izquierda en el número que duplicó ( $2C$ ) y reste éste de la aproximación obtenida anteriormente:  $F = [(2C) + 32] - (2C)/10$ .
  - Determine un método simple para convertir de grados Fahrenheit a Celsius.
  - Determine un método simple para convertir de millas a kilómetros.
  - Determine un método simple de convertir de kilómetros a millas.
- 15.** Convierta fracciones, tal como  $\frac{1}{3}$  y  $\frac{1}{11}$  en su representación binaria.
- 16.** **(Aritmética maya)** La civilización maya de América Central (2600 AC a 1200 DC) entiende el concepto de cero cientos de años antes que muchas otras civilizaciones. En sus cálculos, se utilizaba el sistema vigesimal (base 20), no el sistema decimal (base 10). Así, en lugar de 1, 10, 100, 1000, 10000, utilizaban 1, 20, 400, 8000, 16000. Utilizaban un punto para 1 y una barra

para 5, y el cero se representaba con el símbolo de concha. Por ejemplo, los cálculos de  $11131 + 7520 = 18651$  y  $11131 - 7520 = 3611$  eran como se muestra a continuación:

|      | 11131 | 7520 | 18651 | 3611 |
|------|-------|------|-------|------|
| 8000 | •     |      | ••    |      |
| 400  | ••    | •••  | •     | •••• |
| 20   | •     | •    | ••    | •••  |
| 1    | •     | •••  | •     | •    |

Aquí, como una ayuda, se han incluido algunos de nuestros números; a la izquierda, se indican las potencias utilizadas y arriba se presentan los números representados por las columnas.

Haga estos cálculos utilizando los símbolos y la aritmética maya:

- a.  $92819 + 56313 = 149132$ ,     $92819 - 56313 = 36506$
  - b.  $3296 + 853 = 4149$ ,     $3296 - 853 = 2443$
  - c.  $2273 + 729 = 1544$ ,     $2273 - 729 = 1544$
  - d. Investigue cómo los mayas podrían haber hecho la multiplicación y la división en su sistema de numeración. Trabaje con algunos ejemplos sencillos.
17. (Aritmética babilónica) Los babilonios de la antigua Mesopotamia (ahora Irak) utilizaron un sistema de numeración posicional sexagesimal (base 60) con sistema base decimal (10) dentro. Los babilonios basaban su sistema de numeración ¡en sólo dos símbolos! La influencia de la aritmética babilónica está aún presente. Una hora consta de 60 minutos y se divide en 60 segundos y un círculo se mide en divisiones de 360 grados. Los números son frecuentemente llamados *dígitos*, de la palabra latina para “dedo”. Los sistemas de base 10 y de base 20 muy probablemente surgieron del hecho de que se podrían utilizar para contar los diez dedos y los diez dedos de los pies. Investigue los orígenes históricos de los números y realice cálculos aritméticos con diferentes sistemas numéricos.

## Problemas de cómputo B.1

- Escriba en su computadora  $x = 1.1$  (base 10) e imprima utilizando varios formatos. Explique los resultados.
- Demuestre que  $e^{\pi\sqrt{163}}$  está muy cerca de ser el décimo octavo dígito entero 262 53741 26407 68744. *Sugerencia:* se necesitarán más de 30 dígitos decimales para ver alguna diferencia.
- Escriba y pruebe una rutina para convertir números enteros en la forma octal y binaria.

4. (Continuación) Escriba y pruebe una rutina para la conversión de fracciones decimales en la forma octal y binaria.
5. (Continuación) Usando las dos rutinas de los problemas anteriores, escriba y pruebe un programa que lea los números decimales e imprima las representaciones decimal, octal, binaria de estos números.
6. Vea cuántos dígitos binarios tiene su computadora para  $(0.1)_{10}$ . Vea las observaciones preliminares al comienzo de este capítulo.
7. Algunos sistemas de software matemático tienen instrucciones para la conversión de números entre binario, decimal, hexadecimal, octal y viceversa. Explore estas instrucciones utilizando diversos valores numéricos. También, vea si hay instrucciones para determinar la *precisión* (el número de dígitos significativos decimales en un número) y la *exactitud* (el número de dígitos significativos decimales a la derecha del punto decimal en un número).
8. Escriba un programa de computadora para comprobar las conclusiones en la evaluación de  $f(x) = x - \sin x$  para diferentes valores de  $x$  cerca de 1.9, por ejemplo, en el intervalo  $[0.1, 2.5]$ , con incrementos de 0.1. Para estos valores, calcule el valor aproximado de  $f$ , el valor calculado verdadero y el error absoluto entre ellos. Pueden ser necesarios cálculos de precisión simple y de doble precisión.

# Detalles adicionales de la aritmética de punto flotante del IEEE

En este apéndice se resumen algunas características adicionales de la aritmética de punto flotante estándar del IEEE (véase Overton [2001] para más detalles).

## C.1 Más de la aritmética estándar de punto flotante del IEEE

En la década de 1980, un comité de trabajo del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) estableció un sistema de aritmética de punto flotante estándar para las computadoras que ahora se conoce como el **estándar de punto flotante del IEEE**. Anteriormente, cada uno de los fabricantes de computadoras desarrollaba sus propios sistemas internos de números de punto flotante. Esto condujo a incoherencias en los resultados numéricos cuando se cambiaba el código de máquina a máquina, por ejemplo, al cambiar el código fuente de una computadora IBM a una máquina Cray. Algunos requisitos importantes para que todas las máquinas adopten el estándar del IEEE de punto flotante son los siguientes:

- Redondeado aritmético correcto
- Representación consistente de los números de punto flotante a través de las máquinas
- Tratamiento consistente y sensible de situaciones excepcionales

Supongamos que estamos utilizando una computadora de 32 bits con la aritmética estándar de punto flotante del IEEE. Hay exactamente 23 bits de precisión en el campo de fracción en un número normalizado de precisión simple. Al contar el bit escondido, esto significa que hay 24 bits en la mantisa y el **error de redondeo unitario** es  $u = 2^{-24}$ . Con precisión simple, el **épsilon de la máquina** es  $\epsilon_{\text{simple}} = 2^{-23}$ , ya que  $1 + 2^{-23}$  es el *primer* número de precisión simple mayor que 1. Puesto que  $2^{-23} \approx 1.19 \times 10^{-7}$ , podemos esperar sólo alrededor de seis dígitos de precisión decimal en la salida. Esta precisión se puede reducir aún más por errores de diferentes tipos, tales como los errores de redondeo en la aritmética, errores de truncamiento en las fórmulas utilizadas, etcétera.

Por ejemplo, cuando se calcula la aproximación de precisión simple para  $\pi$ , obtenemos seis dígitos exactos: 3.14159. Convirtiendo e imprimiendo el número binario de 24 bits da como resultado un número decimal real con más de seis dígitos distintos de cero, pero sólo los seis primeros dígitos se consideran aproximaciones precisas a  $\pi$ .

El *primer* número de doble precisión mayor que 1 es  $1 + 2^{-52}$ . Así el **épsilon de la máquina** de doble precisión es  $\epsilon_{\text{doble}} = 2^{-52}$ . Puesto que  $2^{-52} \approx 2.22 \times 10^{-16}$ , sólo hay alrededor de 15 dígitos de precisión decimal en el resultado en ausencia de errores. El campo de fracción tiene exactamente 52 bits de precisión y esto se traduce en 53 bits en la mantisa, cuando se cuenta el bit escondido.

Por ejemplo, cuando se aproxima  $\pi$  con doble precisión, se obtienen 15 dígitos exactos: 3.14159 26535 8979. Como en el caso de precisión simple, al convertir e imprimir la mantisa binaria de 54 bits se obtienen más de 15 dígitos, pero sólo los 15 primeros son aproximaciones precisas para  $\pi$ .

Hay algunos números de interés especial en el estándar del IEEE. En lugar de terminar con un desbordamiento al dividir un número distinto de cero entre 0, la representación de la máquina para  $\infty$  se almacena, que es lo matemáticamente sensato. Debido a la representación del bit escondido, se necesita una técnica especial para almacenar el cero. Observe que todos los ceros en el campo de la fracción (mantisa) representan la mantisa 1.0 más que 0.0. Además, hay dos representaciones para el mismo número cero, es decir, 0 y  $-0$ . Por otra parte, hay dos representaciones de infinito, que corresponden a dos números muy diferentes,  $+\infty$  y  $-\infty$ . **NaN** significa **No un Número** y es un patrón de error más que un número.

¿Es posible representar números *más pequeños que* el número de punto flotante más pequeño normalizado  $2^{-126}$  en el formato del estándar de punto flotante del IEEE? ¡Sí! Si el campo de exponentes contiene una cadena de bits de todos los ceros y el campo de fracción contiene una cadena de bits de no ceros, entonces esta representación se llama un **número subnormal**. Los números subnormales no pueden normalizarse, porque se obtendría un exponente que no se ajustaría en el campo del exponente. Estas cifras subnormales son menos precisas que los números normales, porque tienen menos espacio en el campo de la fracción para bits diferentes de cero.

Al utilizar las funciones de investigación de diversos sistemas (como las de la Tabla C.1 de Fortran 90), podemos determinar algunas de las características del sistema de punto flotante de un número en una PC típica con 32 bits con aritmética estándar de punto flotante del IEEE. La tabla C.2 contiene los resultados. En la mayoría de los casos, los programas simples también se pueden escribir para determinar estos valores.

En la tabla C.3, presentamos la relación entre el campo de exponente y los posibles números de precisión simple de 32 bits de punto flotante que le corresponden. En esta tabla, todos los renglones excepto el primero y el último son números de punto flotante normalizados. El primer renglón muestra que el cero está representado por  $+0$  cuando todos los bits  $b_i = 0$ , y por  $-0$  cuando todos los bits son iguales a cero, excepto  $b_1 = 1$ . El último renglón muestra que  $+\infty$  y  $-\infty$  tienen cadenas de bits de todos los unos que están en el campo del exponente excepto posiblemente por el bit del signo, junto con todos los ceros en el campo de mantisa.

**TABLA C.1** Algunas funciones de investigación numérica en Fortran 90

|                     |                                                                                |
|---------------------|--------------------------------------------------------------------------------|
| <b>EPSILON(X)</b>   | Épsilon de la máquina (número casi insignificante en comparación con 1)        |
| <b>TINY(X)</b>      | El número positivo más pequeño                                                 |
| <b>HUGE(X)</b>      | El número más grande                                                           |
| <b>PRECISION(X)</b> | Precisión decimal (número de dígitos decimales significativos en el resultado) |

**TABLA C.2** Resultados con el estándar de punto flotante del IEEE en una máquina de 32 bits

|                     | X Precisión simple                       | X Doble precisión                                             |
|---------------------|------------------------------------------|---------------------------------------------------------------|
| <b>EPSILON(X)</b>   | $1.192 \times 10^{-7} \approx 2^{-23}$   | $2.220 \times 10^{-16} \approx 2^{-52}$                       |
| <b>TINY(X)</b>      | $1.175 \times 10^{-38} \approx 2^{-126}$ | $2.225 \times 10^{-308} \approx (2 - 2^{-23}) \times 2^{127}$ |
| <b>HUGE(X)</b>      | $3.403 \times 10^{38} \approx 2^{128}$   | $1.798 \times 10^{308} \approx 2^{1024}$                      |
| <b>PRECISION(X)</b> | 6                                        | 15                                                            |

**TABLA C.3** Palabra de 32 bits de precisión simple  $b_1 b_2 b_3 b_4 \dots b_9 b_{10} b_{11} \dots b_{32}$  con bit de signo  $b_1 = 0$  para + y  $b_1 = 1$  para -.

| $(b_2 b_3 \dots b_9)_2$     | Campo del exponente | Representación numérica                                                                                                 |
|-----------------------------|---------------------|-------------------------------------------------------------------------------------------------------------------------|
| $(00000000)_2 = (0)_{10}$   |                     | $\begin{cases} \pm 0, & \text{si } b_{10} = b_{11} = \dots = b_{32} = 0 \\ \text{subnormal, de otra forma} \end{cases}$ |
| $(00000001)_2 = (1)_{10}$   |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{-126}$                                                              |
| $(00000010)_2 = (2)_{10}$   |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{-125}$                                                              |
| $(00000011)_2 = (3)_{10}$   |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{-124}$                                                              |
| $(00000100)_2 = (4)_{10}$   |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{-123}$                                                              |
| ⋮                           |                     | ⋮                                                                                                                       |
| $(01111101)_2 = (125)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{-2}$                                                                |
| $(01111110)_2 = (126)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{-1}$                                                                |
| $(01111111)_2 = (127)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^0$                                                                   |
| $(10000000)_2 = (128)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^1$                                                                   |
| $(10000001)_2 = (129)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^2$                                                                   |
| ⋮                           |                     | ⋮                                                                                                                       |
| $(11111101)_2 = (251)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{124}$                                                               |
| $(11111100)_2 = (252)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{125}$                                                               |
| $(11111101)_2 = (253)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{126}$                                                               |
| $(11111110)_2 = (254)_{10}$ |                     | $\pm(1.b_{10}b_{11}b_{12} \dots b_{32})_2 \times 2^{127}$                                                               |
| $(11111111)_2 = (255)_{10}$ |                     | $\begin{cases} \pm\infty, & \text{si } b_{10} = b_{11} = \dots = b_{32} = 0 \\ \text{NaN, de otra forma} \end{cases}$   |

En la estándar de punto flotante del IEEE, el **redondeo al más cercano** o el valor **correctamente redondeado** del número real  $x$ , se denota por  $\text{round}(x)$  y se define como sigue. Primero, sea  $x_+$  el número de punto flotante más cercano mayor que  $x$  y sea  $x_-$  el más cercano menor que  $x$ . Si  $x$  es un número de punto flotante, entonces  $\text{round}(x) = x$ . De lo contrario, el valor de  $\text{round}(x)$  depende del **modo de redondeo** seleccionado:

- **Redondeado al más cercano:**  $\text{round}(x)$  es ya sea  $x_-$  o  $x_+$ , que es el más cercano a  $x$ . (Si hay un empate, se elige el que tenga el bit menos significativo igual a 0).
- **Redondeado hacia 0:**  $\text{round}(x)$  es  $x_-$  o  $x_+$ , que esté entre 0 y  $x$ .
- **Redondeado hacia  $-\infty$ /redondeado hacia abajo:**  $\text{round}(x) = x_-$ .
- **Redondeado hacia  $+\infty$ /redondeado hacia arriba:**  $\text{round}(x) = x_+$ .

Redondear al más cercano casi siempre se utiliza, ya que es el más útil y da el número de punto flotante más cercano a  $x$ .

# Conceptos y notación de álgebra lineal

En este apéndice se revisan algunos conceptos básicos y la notación común que se utiliza en álgebra lineal.

## D.1 Conceptos elementales

Los dos conceptos de álgebra lineal que más nos preocupan son los *vectores* y las *matrices*, debido a su utilidad para comprimir expresiones complicadas en una notación compacta. Los vectores y las matrices en este libro son con frecuencia *reales*, ya que están compuestas de números reales. Estos conceptos se generalizan fácilmente a vectores y matrices *complejos*.

### Vectores

Un **vector**  $\mathbf{x} \in \mathbb{R}^n$  se puede pensar como un arreglo unidimensional de números y se escribe como

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

donde  $x_i$  se llama el *iésimo elemento, entrada o componente*. Una notación alternativa que es útil en seudocódigos es  $\mathbf{x} = (x_i)_n$ . A veces, se dice que el vector  $\mathbf{x}$  que se acaba de presentar es un **vector columna** para distinguirlo de un **vector renglón** y que se escribe como

$$\mathbf{y} = [y_1, y_2, \dots, y_n]$$

Por ejemplo, aquí se presentan algunos vectores:

$$\begin{bmatrix} \frac{1}{5} \\ 3 \\ -\frac{5}{6} \\ \frac{2}{7} \end{bmatrix} \quad [\pi, e, 5, -4] \quad \begin{bmatrix} \frac{1}{2} \\ \frac{1}{3} \end{bmatrix}$$

Para ahorrar espacio, un vector columna  $\mathbf{x}$  se puede escribir como un vector renglón, como

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T \quad \text{o} \quad \mathbf{x}^T = [x_1, x_2, \dots, x_n]$$

agregando una  $T$  (por **transpuesta**) para indicar que estamos intercambiando o transponiendo un vector renglón o columna. Como ejemplo, tenemos

$$[1 \ 2 \ 3 \ 4]^T = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Muchas operaciones vectoriales son operaciones componente por componente. Para los vectores  $\mathbf{x}$  y  $\mathbf{y}$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

se aplican las siguientes definiciones.

**Igualdad**  $\mathbf{x} = \mathbf{y}$  si y sólo si  $x_i = y_i$  para toda  $i$  ( $1 \leq i \leq n$ )

**Desigualdad**  $\mathbf{x} < \mathbf{y}$  si y sólo si  $x_i < y_i$  para toda  $i$  ( $1 \leq i \leq n$ )

### Suma/Resta

$$\mathbf{x} \pm \mathbf{y} = \begin{bmatrix} x_1 \pm y_1 \\ x_2 \pm y_2 \\ \vdots \\ x_n \pm y_n \end{bmatrix}$$

### Producto escalar

$$\alpha \mathbf{x} = \begin{bmatrix} \alpha x_1 \\ \alpha x_2 \\ \vdots \\ \alpha x_n \end{bmatrix} \quad \text{para } \alpha \text{ una constante o escalar}$$

Aquí se presenta un ejemplo

$$\begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix} = 2 \begin{bmatrix} 0 \\ 2 \\ 0 \\ 4 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \\ 6 \\ 0 \end{bmatrix}$$

Para  $m$  vectores,  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}$  y  $m$  escalares  $\alpha_1, \alpha_2, \dots, \alpha_m$ , se define una **combinación lineal** como

$$\sum_{i=1}^m \alpha_i \mathbf{x}^{(i)} = \alpha_1 \mathbf{x}^{(1)} + \alpha_2 \mathbf{x}^{(2)} + \cdots + \alpha_m \mathbf{x}^{(m)} = \begin{bmatrix} \sum_{i=1}^m \alpha_i x_1^{(i)} \\ \sum_{i=1}^m \alpha_i x_2^{(i)} \\ \vdots \\ \sum_{i=1}^m \alpha_i x_n^{(i)} \end{bmatrix}$$

Los vectores especiales son los **vectores unitarios** comunes:

$$\mathbf{e}^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \mathbf{e}^{(2)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad \mathbf{e}^{(n)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Obviamente,

$$\sum_{i=1}^n \alpha_i \mathbf{e}^{(i)} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$$

Por lo tanto, cualquier vector  $\mathbf{x}$  se puede escribir como una combinación lineal de los vectores unitarios comunes

$$\mathbf{x} = x_1 \mathbf{e}^{(1)} + x_2 \mathbf{e}^{(2)} + \dots + x_n \mathbf{e}^{(n)} = \sum_{i=1}^n x_i \mathbf{e}^{(i)}$$

Como un ejemplo, observe que

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} + 4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

El **producto punto, escalar o interno** de vectores  $\mathbf{x}$  y  $\mathbf{y}$  es el número

$$\mathbf{x}^T \mathbf{y} = [x_1, x_2, \dots, x_n] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^n x_i y_i$$

Como ejemplo, vemos que

$$[1, 1, 1, 1] \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 4$$

## Matrices

Una **matriz** es un arreglo bidimensional de números que se puede escribir como

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

donde  $a_{ij}$  se llama **elemento** o **entrada** en el  $i$ ésimo renglón y en la  $j$ ésima columna. Una notación alternativa es  $\mathbf{A} = (a_{ij})_{n \times m}$ . Un vector de columna es también una matriz de  $n \times 1$  y un vector

renglón es también una matriz de  $1 \times m$ . Por ejemplo, aquí están tres matrices:

$$\begin{bmatrix} \frac{1}{5} & \frac{2}{7} & -1 \\ 3 & 2 & \frac{1}{8} \\ -\frac{5}{6} & \frac{2}{5} & 3 \end{bmatrix} \quad \begin{bmatrix} 1 & 6 & \frac{9}{8} & -5 \end{bmatrix} \quad \begin{bmatrix} \frac{11}{2} & \frac{4}{9} \\ \frac{2}{3} & -\frac{7}{8} \\ \pi & e \\ \frac{1}{\pi} & \frac{1}{e} \end{bmatrix}$$

Las entradas en  $A$  se pueden agrupar en vectores columna:

$$A = \begin{bmatrix} [a_{11}] & [a_{12}] & [a_{1m}] \\ [a_{21}] & [a_{22}] & [a_{2m}] \\ \vdots & \vdots & \vdots \\ [a_{n1}] & [a_{n2}] & [a_{nm}] \end{bmatrix} = [\mathbf{a}^{(1)} \quad \mathbf{a}^{(2)} \quad \cdots \quad \mathbf{a}^{(m)}]$$

donde  $\mathbf{a}^{(j)}$  es el  $j$ -ésimo vector columna. También,  $A$  se puede agrupar en vectores renglón:

$$A = \begin{bmatrix} [a_{11} & a_{12} & \cdots & a_{1m}] \\ [a_{21} & a_{22} & \cdots & a_{2m}] \\ \vdots \\ [a_{n1} & a_{n2} & \cdots & a_{nm}] \end{bmatrix} = \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(n)} \end{bmatrix}$$

donde  $A^{(j)}$  es el  $i$ -ésimo vector renglón. Observe que

$$\begin{bmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{bmatrix} = \left[ \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \\ 7 \\ 8 \end{bmatrix} \begin{bmatrix} 9 \\ 10 \\ 11 \\ 12 \end{bmatrix} \begin{bmatrix} 13 \\ 14 \\ 15 \\ 16 \end{bmatrix} \right] = \begin{bmatrix} [1 & 5 & 9 & 13] \\ [2 & 6 & 10 & 14] \\ [3 & 7 & 11 & 15] \\ [4 & 8 & 12 & 16] \end{bmatrix}$$

Una matriz de  $n \times n$  de especial importancia es la matriz **identidad**, que se denota por  $I$ , y está compuesta toda por ceros, con excepción en la diagonal que consta de unos:

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = [\mathbf{e}^{(1)} \quad \mathbf{e}^{(2)} \quad \cdots \quad \mathbf{e}^{(n)}]$$

Una matriz de esta misma forma general con entradas  $d_i$  en la diagonal principal se llama una matriz **diagonal** y se escribe como

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} = \text{diag}(d_1, d_2, \dots, d_n)$$

donde el espacio en blanco indica entradas 0. Una matriz **tridiagonal** es una matriz cuadrada de la forma

$$T = \begin{bmatrix} d_1 & c_1 & & & \\ a_1 & d_2 & c_2 & & \\ & a_2 & d_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ & & & a_{n-2} & d_{n-1} & c_{n-1} \\ & & & & a_{n-1} & d_n \end{bmatrix}$$

donde los elementos de la diagonal  $\{a_i\}$ ,  $\{d_i\}$  y  $\{c_i\}$  se llaman la **subdiagonal**, **diagonal principal** y **superdiagonal**, respectivamente.

Para la matriz general  $A = (a_{ij})$  de  $n \times n$ ,  $A$  es una matriz diagonal si  $a_{ij} = 0$  cuando  $i \neq j$  y  $A$  es una matriz tridiagonal si  $a_{ij} = 0$  cuando  $|i - j| \geq 2$ . La matriz  $A$  es una **matriz triangular inferior** siempre que  $a_{ij} = 0$  para toda  $i < j$ , y es una **matriz triangular superior** cada vez  $a_{ij} = 0$  para toda  $i > j$ . Ejemplos de matrices identidad, diagonal, tridiagonal, triangular inferior y superior respectivamente, son los siguientes:

$$\begin{array}{ccc} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 9 \end{bmatrix} & \begin{bmatrix} 5 & 3 & 0 & 0 & 0 \\ 2 & 5 & 3 & 0 & 0 \\ 0 & 2 & 9 & 2 & 0 \\ 0 & 0 & 3 & 7 & 2 \\ 0 & 0 & 0 & 3 & 7 \end{bmatrix} \\ \begin{bmatrix} 6 & 0 & 0 & 0 \\ 3 & 6 & 0 & 0 \\ 4 & -2 & 7 & 0 \\ 5 & -3 & 9 & 21 \end{bmatrix} & \begin{bmatrix} 1 & -1 & 2 & 1 \\ 0 & 5 & -5 & 1 \\ 0 & 0 & 9 & -3 \\ 0 & 0 & 0 & 2 \end{bmatrix} & \end{array}$$

Como con los vectores, muchas operaciones con matrices corresponden a operaciones con las componentes. Para las matrices  $A$  y  $B$ ,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix}$$

se aplican las siguientes definiciones:

**Igualdad**  $A = B$  si y sólo si  $a_{ij} = b_{ij}$  para toda  $i$  ( $1 \leq i \leq n$ ) y toda  $j$  ( $1 \leq j \leq m$ )

**Desigualdad**  $A < B$  si y sólo si  $a_{ij} < b_{ij}$  para toda  $i$  ( $1 \leq i \leq n$ ) y toda  $j$  ( $1 \leq j \leq m$ )

### Suma/Resta

$$A \pm B = \begin{bmatrix} a_{11} \pm b_{11} & a_{12} \pm b_{12} & \cdots & a_{1m} \pm b_{1m} \\ a_{21} \pm b_{21} & a_{22} \pm b_{22} & \cdots & a_{2m} \pm b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} \pm b_{n1} & a_{n2} \pm b_{n2} & \cdots & a_{nm} \pm b_{nm} \end{bmatrix}$$

### Producto escalar

$$\alpha A = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \cdots & \alpha a_{1m} \\ \alpha a_{21} & \alpha a_{22} & \cdots & \alpha a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha a_{n1} & \alpha a_{n2} & \cdots & \alpha a_{nm} \end{bmatrix} \quad \text{para } \alpha \text{ una constante}$$

Como ejemplo, tenemos

$$\begin{bmatrix} \frac{1}{5} & \frac{7}{5} & -1 \\ -3 & 2 & -8 \\ \frac{6}{5} & \frac{2}{5} & -3 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 1 & 7 & 0 \\ 0 & 10 & 0 \\ 6 & 2 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 1 \\ 3 & 0 & 8 \\ 0 & 0 & 3 \end{bmatrix}$$

## Producto matriz-vector

El producto de una matriz  $A$  de  $n \times m$  y un vector  $b$  de  $m \times 1$  es de especial interés. Considerando la matriz  $A$  en términos de sus columnas, tenemos

$$\begin{aligned} Ab &= [a^{(1)} \quad a^{(2)} \quad \cdots \quad a^{(m)}] \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \\ &= b_1 a^{(1)} + b_2 a^{(2)} + \cdots + b_m a^{(m)} \\ &= \sum_{i=1}^m b_i a^{(i)} \end{aligned}$$

Así,  $Ab$  es un vector y se puede pensar como una combinación lineal de las columnas de  $A$  con los coeficientes de las entradas de  $b$ . Considerando la matriz  $A$  en términos de sus renglones, tenemos

$$Ab = \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(n)} \end{bmatrix} b = \begin{bmatrix} A^{(1)}b \\ A^{(2)}b \\ \vdots \\ A^{(n)}b \end{bmatrix}$$

Así pues, el  $j$ -ésimo elemento de  $Ab$  se puede ver como el producto escalar del  $j$ -ésimo renglón de  $A$  y el vector  $b$ .

## Producto matricial

El producto de la matriz  $A = (a_{ij})_{n \times m}$  y la matriz  $B = (b_{ij})_{m \times r}$  es la matriz  $C = (c_{ij})_{n \times r}$  tal que

$$AB = C$$

donde

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{im}b_{mj} = \sum_{k=1}^m a_{ik}b_{kj} \quad (1 \leq i \leq n, 1 \leq j \leq r)$$

El elemento  $c_{ij}$  es el producto del  $i$ -ésimo vector renglón de  $A$

$$A^{(i)} = [a_{i1}, a_{i2}, \dots, a_{im}]$$

y el  $j$ -ésimo vector columna de  $B$

$$b^{(j)} = \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{mj} \end{bmatrix}$$

esto es,

$$c_{ij} = A^{(i)}b^{(j)}$$

Del mismo modo, el producto matricial  $\mathbf{AB}$  se puede considerar de dos maneras. Podemos escribir

$$\begin{aligned}\mathbf{AB} &= \mathbf{A} \begin{bmatrix} \mathbf{b}^{(1)} & \mathbf{b}^{(2)} & \cdots & \mathbf{b}^{(r)} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{Ab}^{(1)} & \mathbf{Ab}^{(2)} & \cdots & \mathbf{Ab}^{(r)} \end{bmatrix} \\ &= \mathbf{C}\end{aligned}\quad (1)$$

o

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \vdots \\ \mathbf{A}^{(n)} \end{bmatrix} \mathbf{B} = \begin{bmatrix} \mathbf{A}^{(1)}\mathbf{B} \\ \mathbf{A}^{(2)}\mathbf{B} \\ \vdots \\ \mathbf{A}^{(n)}\mathbf{B} \end{bmatrix} = \mathbf{C} \quad (2)$$

La ecuación (1) implica que la  $j$ -ésima columna de  $\mathbf{C} = \mathbf{AB}$  es

$$\mathbf{c}^{(j)} = \mathbf{Ab}^{(j)}$$

Es decir, cada columna de  $\mathbf{C}$  es el resultado de multiplicar por la derecha  $\mathbf{A}$  por la  $j$ -ésima columna de  $\mathbf{B}$ . En otras palabras, cada columna de  $\mathbf{C}$  se puede obtener tomando productos internos de una columna de  $\mathbf{B}$  con todas las filas de  $\mathbf{A}$ :

$$\mathbf{c}^{(j)} = \mathbf{Ab}^{(j)} = \begin{bmatrix} \leftarrow \\ \leftarrow \\ \vdots \\ \leftarrow \end{bmatrix} \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{mj} \end{bmatrix} = \begin{bmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{nj} \end{bmatrix}$$

La flecha larga hacia la izquierda significa un producto interno que se forma con los elementos del renglón, es decir  $c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$ . La ecuación (2) implica que el  $i$ -ésimo renglón de la  $\mathbf{C}$  que resulta se obtiene de multiplicar  $\mathbf{A}$  por  $\mathbf{B}$  es

$$\mathbf{C}^{(i)} = \mathbf{A}^{(i)}\mathbf{B}$$

Es decir, cada renglón de  $\mathbf{C}$  es el resultado de *multiplicar por la izquierda*  $\mathbf{B}$  por el  $i$ -ésimo renglón de  $\mathbf{A}$ . En otras palabras, cada renglón de  $\mathbf{C}$  se puede obtener tomando productos internos de un renglón de  $\mathbf{A}$  con todas las columnas de  $\mathbf{B}$ :

$$\begin{aligned}\mathbf{C}^{(i)} = \mathbf{A}^{(i)}\mathbf{B} &= [a_{i1} \quad a_{i2} \quad \cdots \quad a_{im}] \begin{bmatrix} \uparrow & \uparrow & \cdots & \uparrow \end{bmatrix} \\ &= [c_{i1} \quad c_{i2} \quad \cdots \quad c_{ir}]\end{aligned}$$

La flecha larga hacia arriba indica un producto interno que se forma por los elementos en la columna, es decir,  $c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$ .

Como ejemplo, podemos determinar el producto matricial como columna como

$$\begin{bmatrix} 3 & 1 & 7 \\ 2 & 4 & -5 \\ 1 & -3 & 2 \end{bmatrix} \begin{bmatrix} -1 & -3 & 2 \\ 1 & 1 & 1 \\ -3 & -2 & 1 \end{bmatrix} = [\mathbf{c}^{(1)} \quad \mathbf{c}^{(2)} \quad \mathbf{c}^{(3)}]$$

donde

$$\mathbf{c}^{(1)} = \begin{bmatrix} 3 & 1 & 7 \\ 2 & 4 & -5 \\ 1 & -3 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -3 \end{bmatrix} = \begin{bmatrix} -23 \\ 17 \\ -10 \end{bmatrix}$$

$$\mathbf{c}^{(2)} = \begin{bmatrix} 3 & 1 & 7 \\ 2 & 4 & -5 \\ 1 & -3 & 2 \end{bmatrix} \begin{bmatrix} -3 \\ 1 \\ -2 \end{bmatrix} = \begin{bmatrix} -22 \\ 8 \\ -10 \end{bmatrix}$$

$$\mathbf{c}^{(3)} = \begin{bmatrix} 3 & 1 & 7 \\ 2 & 4 & -5 \\ 1 & -3 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 14 \\ 3 \\ 1 \end{bmatrix}$$

o podemos determinarlo como vector como

$$\begin{bmatrix} 3 & 1 & 7 \\ 2 & 4 & -5 \\ 1 & -3 & 2 \end{bmatrix} \begin{bmatrix} -1 & -3 & 2 \\ 1 & 1 & 1 \\ -3 & -2 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{C}^{(1)} \\ \mathbf{C}^{(2)} \\ \mathbf{C}^{(3)} \end{bmatrix}$$

donde

$$\mathbf{C}^{(1)} = [3 \ 1 \ 7] \begin{bmatrix} -1 & -3 & 2 \\ 1 & 1 & 1 \\ -3 & -2 & 1 \end{bmatrix} = [-23 \ -22 \ 14]$$

$$\mathbf{C}^{(2)} = [2 \ 4 \ -5] \begin{bmatrix} -1 & -3 & 2 \\ 1 & 1 & 1 \\ -3 & -2 & 1 \end{bmatrix} = [17 \ 8 \ 3]$$

$$\mathbf{C}^{(3)} = [1 \ -3 \ 2] \begin{bmatrix} -1 & -3 & 2 \\ 1 & 1 & 1 \\ -3 & -2 & 1 \end{bmatrix} = [-10 \ -10 \ 1]$$

## Otros conceptos

La **transpuesta** de la matriz  $\mathbf{A}$  de  $n \times m$ , que se denota por  $\mathbf{A}^T$ , se obtiene intercambiando los renglones y las columnas de  $\mathbf{A} = (a_{ij})_{n \times m}$ :

$$\mathbf{A}^T = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \vdots \\ \mathbf{A}^{(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(1)^T} & \mathbf{A}^{(2)^T} & \cdots & \mathbf{A}^{(n)^T} \end{bmatrix}$$

o

$$\mathbf{A}^T = [\mathbf{a}^{(1)} \ \mathbf{a}^{(2)} \ \cdots \ \mathbf{a}^{(m)}]^T = \begin{bmatrix} \mathbf{a}^{(1)^T} \\ \mathbf{a}^{(2)^T} \\ \vdots \\ \mathbf{a}^{(m)^T} \end{bmatrix}$$

Por lo tanto,  $\mathbf{A}^T$  es la matriz de  $m \times n$ :

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{bmatrix} = (a_{ji})_{m \times n}$$

Como ejemplo, tenemos

$$\begin{bmatrix} 2 & 4 & 9 \\ 5 & 7 & 3 \\ 10 & 6 & 2 \end{bmatrix}^T = \begin{bmatrix} 2 & 5 & 10 \\ 4 & 7 & 6 \\ 9 & 3 & 2 \end{bmatrix}$$

Una matriz  $\mathbf{A}$  de  $n \times n$  es **simétrica** si  $a_{ij} = a_{ji}$  para toda  $i$  ( $1 \leq i \leq n$ ) y toda  $j$  ( $1 \leq j \leq n$ ). En otras palabras,  $\mathbf{A}$  es simétrica si  $\mathbf{A} = \mathbf{A}^T$ .

Algunas propiedades útiles para matrices de tamaños compatibles son los siguientes:

## ■ PROPIEDADES Consecuencias elementales de las definiciones

1.  $\mathbf{AB} \neq \mathbf{BA}$  (en general)
2.  $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$
3.  $\mathbf{A}\mathbf{0} = \mathbf{0}\mathbf{A} = \mathbf{0}$
4.  $(\mathbf{A}^T)^T = \mathbf{A}$
5.  $(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$
6.  $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$

Si  $\mathbf{A}$  y  $\mathbf{B}$  son matrices cuadradas que cumplen  $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$ , entonces se dice que  $\mathbf{B}$  es la inversa de  $\mathbf{A}$ , lo que se denota por  $\mathbf{A}^{-1}$ .

Para ilustrar la propiedad 1, se forman los siguientes productos y se observa que la multiplicación de matrices no es commutativa:

$$\begin{bmatrix} 3 & 1 & 7 \\ 2 & 4 & -5 \\ 1 & -3 & 2 \end{bmatrix} \begin{bmatrix} -1 & -3 & 2 \\ 1 & 1 & 1 \\ -3 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -3 & 2 \\ 1 & 1 & 1 \\ -3 & -2 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 7 \\ 2 & 4 & -5 \\ 1 & -3 & 2 \end{bmatrix} \neq \mathbf{I}$$

También se comprueba que  $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$  para

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 3 & 2 \\ 2 & 1 & 1 \end{bmatrix}$$

y

$$\mathbf{A}^{-1} = \begin{bmatrix} -1 & 0 & 1 \\ -5 & 1 & 3 \\ 7 & -1 & -4 \end{bmatrix}$$

Como con nuestra última serie de ejemplos, tenemos el producto de una matriz por un vector y de dos matrices:

$$\begin{bmatrix} 3 & 2 & -1 \\ 5 & 3 & 2 \\ -1 & 1 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3x_1 + 2x_2 - x_3 \\ 5x_1 + 3x_2 + 2x_3 \\ -x_1 + x_2 - 3x_3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -\frac{5}{3} & 1 & 0 \\ -8 & 5 & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 & -1 \\ 5 & 3 & 2 \\ -1 & 1 & -3 \end{bmatrix} = \begin{bmatrix} 3 & 2 & -1 \\ 0 & -\frac{1}{3} & \frac{11}{3} \\ 0 & 0 & 15 \end{bmatrix}$$

El lector debe comprobarlos y observar cómo se relacionan con la solución del problema usando la eliminación gaussiana simple (véase la sección 7.1):

$$\begin{cases} 3x_1 + 2x_2 - x_3 = 7 \\ 5x_1 + 3x_2 + 2x_3 = 4 \\ -x_1 + x_2 - 3x_3 = -1 \end{cases}$$

Asimismo, calcular los productos que se muestran y relacionarlos con este problema:

$$\begin{bmatrix} 1 & 0 & 0 \\ -\frac{5}{8} & 1 & 0 \\ -8 & 5 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 4 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{3} & 2 & -\frac{7}{15} \\ 0 & -3 & \frac{11}{15} \\ 0 & 0 & \frac{1}{15} \end{bmatrix} \begin{bmatrix} 3 & 2 & -1 \\ 0 & -\frac{1}{3} & \frac{11}{3} \\ 0 & 0 & 15 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{3} & 2 & -\frac{7}{15} \\ 0 & -3 & \frac{11}{15} \\ 0 & 0 & \frac{1}{15} \end{bmatrix} \begin{bmatrix} 7 \\ -\frac{23}{3} \\ -37 \end{bmatrix}$$

## Regla de Cramer

La solución de un sistema lineal de  $2 \times 2$  de la forma

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

está dada por

$$x = \frac{1}{D} \text{Det} \begin{bmatrix} f & c \\ g & d \end{bmatrix} = \frac{1}{D}(fd - gc)$$

$$y = \frac{1}{D} \text{Det} \begin{bmatrix} a & f \\ b & g \end{bmatrix} = \frac{1}{D}(ag - bf)$$

donde

$$D = \text{Det} \begin{bmatrix} a & c \\ b & d \end{bmatrix} = ad - bc \neq 0$$

## D.2 Espacios vectoriales abstractos

Los vectores que se han considerado hasta el momento en este apéndice son miembros de un particular espacio vectorial  $\mathbb{R}^n$ . Existe un concepto general de un espacio vectorial abstracto que incluye a  $\mathbb{R}^n$  como un caso particular. Un espacio vectorial abstracto (un espacio lineal) es un cuádrupla  $(X, F, +, *)$ , donde  $X$  es un conjunto de elementos llamados **vectores**,  $F$  es un **campo**,  $+$  es una operación y  $*$  es una operación. Hay diez axiomas que deben cumplir y todos ellos son familiares para cualquier lector que haya trabajado con el caso especial de  $\mathbb{R}^n$ . Primero, vamos a fijar el campo como  $\mathbb{R}$ . (El otro campo que se necesita con frecuencia es  $\mathbb{C}$ , pero otros campos diferentes de estos dos se usan muy poco en esta situación.)

### ■ TEOREMA 1

#### Axiomas para un espacio vectorial

1. Si  $x$  y  $y$  pertenecen a  $X$ , entonces  $x + y$  también pertenece a  $X$ .
2. Para  $x$  y  $y$  en  $X$ ,  $x + y = y + x$ .
3. Para  $x$ ,  $y$  y  $z$  en  $X$ ,  $(x + y) + z = x + (y + z)$ .
4. El conjunto  $X$  contiene un elemento especial  $\mathbf{0}$  tal que  $x + \mathbf{0} = x$  para todo  $x$  en  $X$ .
5. Para cada  $x$ , hay un elemento  $-x$  con la propiedad de que  $x + (-x) = \mathbf{0}$ .
6. Si  $a \in \mathbb{R}$ , entonces, para toda  $x$  en  $X$   $ax \in X$  ( $ax$  significa  $a * x$ ).
7. Si  $a \in \mathbb{R}$  y  $x$ ,  $y \in X$ , entonces  $a(x + y) = ax + ay$ .
8. Si  $a$ ,  $b \in \mathbb{R}$  y  $x \in X$ , entonces  $(a + b)x = ax + bx$ .
9. Si  $a$ ,  $b \in \mathbb{R}$  y  $x \in X$ , entonces  $a(bx) = (ab)x$ .
10. Para  $x \in X$ ,  $1x = x$ .

A partir de estos axiomas, se pueden demostrar muchas propiedades adicionales, como las siguientes

### ■ PROPIEDADES Consecuencias inmediatas de los axiomas

1. El elemento cero,  $\mathbf{0}$ , de  $X$  es único.
2.  $0x = \mathbf{0}$  y  $a\mathbf{0} = \mathbf{0}$  para  $a \in \mathbb{R}$ . (Aquí observe los diferentes ceros!)
3. Para toda  $x$  en  $X$ , el elemento  $-x$  en el axioma 5 es único.
4. Para toda  $x$  en  $X$ ,  $(-1)x = -x$ .
5. Si  $ax = \mathbf{0}$  y  $a \neq 0$ , entonces  $x = \mathbf{0}$ .

Un buen ejemplo de un espacio vectorial (que no sea  $\mathbb{R}^n$ ) es el conjunto de todos los polinomios. Sabemos que la suma de dos polinomios es otro polinomio y que un múltiplo escalar de un polinomio es un polinomio. Todos los otros axiomas de un espacio vectorial se comprueban rápidamente. El elemento cero es el polinomio que se define por la ecuación  $\mathbf{0}(t) = 0$  para todos los valores reales de  $t$ .

## Subespacios

Si  $U$  es un subconjunto del espacio vectorial  $X$  y si  $U$  es un espacio vectorial también (con las mismas definiciones de  $+$  y  $*$  que se utiliza en  $X$ ), entonces  $U$  se llama un subespacio de  $X$ . En la comprobación para determinar si un subconjunto dado  $U$  es un subespacio, basta con comprobar los axiomas 1 y 6. De hecho, una vez que se ha hecho, del axioma 6 y de la propiedad 4 se obtiene que  $-\mathbf{u} \in U$  cuando  $\mathbf{u} \in U$ . Entonces del axioma 1 se obtiene  $\mathbf{0} = \mathbf{u} + (-\mathbf{u}) \in U$ . El resto de los axiomas son ciertos para  $U$ , simplemente porque  $U \subset X$ .

## Independencia lineal

Un conjunto de puntos ordenado finito  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  en un espacio vectorial es **linealmente dependiente** si existe una ecuación no trivial de la forma

$$\sum_{i=1}^n a_i \mathbf{x}_i = \mathbf{0}$$

El término *no trivial* significa que no todos los coeficientes  $a_i$  son iguales a cero. Por ejemplo, si  $n = 3$  y  $\mathbf{x}_1 = 3\mathbf{x}_2 - \mathbf{x}_3$ , entonces el conjunto ordenado  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$  es linealmente dependiente. Si  $n = 3$  y  $\mathbf{x}_3 = \mathbf{x}_1$ , lo que se permite en un conjunto ordenado, entonces  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$  es linealmente dependiente. Observe que si éstos fueron interpretados como conjuntos del plano, tendríamos  $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1\} = \{\mathbf{x}_1, \mathbf{x}_2\}$ , porque en la descripción de un conjunto plano se puede quitar la entrada repetida ¡sin modificar el conjunto! Esto explica la necesidad de tratar con conjuntos ordenados o conjuntos indizados en la definición de dependencia lineal. (La dificultad surge sólo para conjuntos indizados en los que dos elementos son iguales pero tienen diferentes índices.) Un conjunto finito que consta de  $n$  (distintos) elementos  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  es **linealmente independiente** si la ecuación

$$\sum_{i=1}^n a_i \mathbf{x}_i = \mathbf{0}$$

es verdadera sólo cuando todos los coeficientes  $a_i$  son cero. Un conjunto arbitrario, posiblemente infinito, es linealmente independiente si todo subconjunto finito de ese conjunto es linealmente independiente.

Para ilustrar la independencia lineal, considere los tres polinomios  $\mathbf{p}_1(t) = t^3 - 2t$ ,  $\mathbf{p}_2(t) = t^2 + 4$  y  $\mathbf{p}_3(t) = 2t^2 + t$ . ¿Es el conjunto  $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$  linealmente independiente? Para averiguarlo, supongamos que  $a_1\mathbf{p}_1 + a_2\mathbf{p}_2 + a_3\mathbf{p}_3 = 0$ . Entonces, para toda  $t$ ,

$$a_1(t^3 - 2t) + a_2(t^2 + 4) + a_3(2t^2 + t) = 0$$

Agrupando términos, tenemos

$$a_3t^3 + (a_2 + 2a_3)t^2 + (-2a_1 + a_3)t + 4a_2 = 0 \quad (t \in \mathbb{R})$$

Puesto que un polinomio cúbico puede tener como máximo tres raíces (si no es cero), los coeficientes de cada potencia de  $t$  en la ecuación anterior deben ser cero:

$$a_3 = a_2 + 2a_3 = -2a_1 + a_3 = 4a_2 = 0$$

Por lo tanto, todas las  $a_i$  deben ser cero. El conjunto es linealmente independiente.

**TEOREMA 2****Teorema de dependencia lineal**

Un conjunto finito ordenado  $\{x_1, x_2, \dots, x_n\}$ , con  $n \geq 2$ , es linealmente dependiente si y sólo si algún elemento del conjunto, por ejemplo,  $x_k$ , es una combinación lineal de sus predecesores en el conjunto:

$$x_k = \sum_{i=1}^{k-1} a_i x_i$$

**Bases**

Una **base** de un espacio vectorial es un conjunto **linealmente independiente máximo** en el espacio vectorial. **Máximo** significa que no se puede agregar ningún vector al conjunto sin perder la independencia lineal. Por ejemplo, una base para el espacio de todos los polinomios está dada por las funciones  $u_i(t) = t^i$  para  $i = 0, 1, 2, \dots$ . Para ver que se trata de un máximo conjunto linealmente independiente, supongamos que agregamos al conjunto un polinomio de  $p$ . Sea el grado de  $p$  igual a  $n$ . Entonces, el conjunto  $\{u_0, u_1, \dots, u_n, p\}$  es linealmente dependiente. De hecho, un elemento (es decir,  $p$ ) es una combinación lineal de sus predecesores en el conjunto, y se aplica el teorema anterior.

Si un espacio vectorial  $X$  tiene una base finita,  $\{u_0, u_1, \dots, u_n\}$ , entonces toda base para  $X$  tiene  $n$  elementos. Este número se llama la **dimensión** de  $X$ , y decimos que  $X$  es de **dimensión finita**. Cada  $x$  en  $X$  tiene una representación única  $x = \sum_{i=1}^n a_i u_i$ . La existencia de esta representación es una consecuencia de la maximalidad y la unicidad es una consecuencia de la independencia lineal de la base.

**Transformaciones lineales**

Si  $X$  y  $Y$  son espacios vectoriales y si  $L$  es un mapeo de  $X$  en  $Y$  tal que

$$L(au + bv) = aL(u) + bL(v)$$

para todos los escalares  $a$  y  $b$  y para todos los vectores  $u$  y  $v$  en  $X$ , entonces decimos que  $L$  es **lineal**. Muchas de las operaciones familiares que se estudian en matemáticas son lineales. Por ejemplo, la **derivación** es un operador lineal:

$$(f + g)' = f' + g' \quad (af)' = af'$$

La transformada de Laplace es lineal, y también lo es el mapeo  $f \mapsto \int_1^b f(t) dt$ .

Si el espacio  $X$  es de dimensión finita y si se selecciona una base  $\{u_1, u_2, \dots, u_n\}$  para  $X$ , entonces un mapeo lineal  $L: X \rightarrow Y$  es completamente conocido si se conocen los  $n$  vectores  $Lu_1, Lu_2, \dots, Lu_n$ . En efecto, cualquier vector  $x$  en  $X$  es representable en términos de la base,  $x = \sum_{j=1}^n c_j u_j$  y a partir de esto obtenemos  $Lx = \sum_{j=1}^n c_j Lu_j$ . Yendo más allá, supongamos que  $Y$  también es de dimensión finita. Seleccione una base para  $Y$ , digamos,  $\{v_1, v_2, \dots, v_m\}$ . Entonces, cada imagen  $Lu_j$  se expresa en términos de la base seleccionada para  $Y$ , y tenemos, para coeficientes adecuados  $a_{ij}$ ,

$$Lu_j = \sum_{i=1}^m a_{ij} v_i$$

De esto se deduce que

$$Lx = L \left( \sum_{j=1}^n c_j u_j \right) = \sum_{j=1}^n c_j Lu_j = \sum_{j=1}^n c_j \sum_{i=1}^m a_{ij} v_i$$

De este modo, una matriz  $A = (a_{ij})$  se asocia con  $L$ , pero sólo después de que se han elegido las bases en  $X$  y  $Y$ .

El caso especial en el que  $Y = X$  y se utiliza la misma base en ambas funciones da lugar a estas ecuaciones:

$$\begin{aligned}\mathbf{x} &= \sum_{j=1}^n c_j \mathbf{u}_j \\ L\mathbf{u}_j &= \sum_{i=1}^n a_{ij} \mathbf{u}_i \\ L\mathbf{x} &= \sum_{j=1}^n c_j \sum_{i=1}^n a_{ij} \mathbf{u}_i\end{aligned}$$

## Valores y vectores propios

Sea  $A$  una matriz de  $n \times n$ . Si  $\mathbf{x}$  es un vector distinto de cero con la propiedad de que  $A\mathbf{x}$  es un múltiplo escalar de  $\mathbf{x}$ , entonces llamamos a  $\mathbf{x}$  un **vector propio** de  $A$ . Cuando esto ocurre, la ecuación

$$A\mathbf{x} = \lambda\mathbf{x}$$

se cumple para algún escalar  $\lambda$  (que puede ser cero). Entonces al escalar  $\lambda$  se le llama un **valor propio** de  $A$ . Puesto que tenemos una solución distinta de cero de la ecuación  $A\mathbf{x} - \lambda\mathbf{x} = \mathbf{0}$ , la matriz  $A - \lambda I$  debe ser singular. Por lo tanto, su determinante es cero. La ecuación

$$\text{Det}(A - \lambda I) = 0$$

se llama la **ecuación característica** de  $A$ . En función de  $\lambda$ , el lado izquierdo de esta ecuación es un polinomio de grado  $n$ , que tiene exactamente  $n$  raíces si contamos cada raíz con su multiplicidad.

## Cambio de base y similitud

Si  $L$  es una transformación lineal obtenida en un espacio vectorial  $n$ -dimensional en sí mismo, entonces, una vez seleccionada una base  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ , podemos asignar una matriz  $A$  a  $L$ . Por lo tanto, tenemos que

$$L\mathbf{u}_j = \sum_{i=1}^n A_{ij} \mathbf{u}_i$$

Si se elige otra base para  $X$ , por ejemplo,  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}$ , entonces otra matriz,  $B$ , surge de la misma manera, y tenemos

$$L\mathbf{v}_j = \sum_{i=1}^n B_{ij} \mathbf{v}_i$$

¿Cuál es la relación entre  $A$  y  $B$ ? Se define la matriz  $P$  mediante la ecuación

$$\mathbf{u}_k = \sum_{i=1}^n P_{ik} \mathbf{v}_i \quad 1 \leq k \leq n$$

Entonces

$$B = PAP^{-1}$$

Para probar esto debemos hacer que

$$\mathbf{L}\mathbf{v}_j = \sum_{i=1}^n (\mathbf{PAP}^{-1})_{ij} \mathbf{v}_i$$

Las ecuaciones antes dadas justifican los pasos en el siguiente cálculo:

$$\begin{aligned} \sum_{i=1}^n (\mathbf{PAP}^{-1})_{ij} \mathbf{v}_i &= \sum_{i=1}^n \sum_{k=1}^n \sum_{r=1}^n \mathbf{P}_{ik} \mathbf{A}_{kr} \mathbf{P}_{rj}^{-1} \mathbf{v}_i \\ &= \sum_{k=1}^n \sum_{r=1}^n \mathbf{A}_{kr} \mathbf{P}_{rj}^{-1} \mathbf{u}_k \\ &= \sum_{r=1}^n \mathbf{P}_{rj}^{-1} L \mathbf{u}_r \\ &= L \left( \sum_{r=1}^n \mathbf{P}_{rj}^{-1} \mathbf{u}_r \right) \\ &= L \left( \sum_{r=1}^n \sum_{i=1}^n \mathbf{P}_{rj}^{-1} \mathbf{P}_{ir} \mathbf{v}_i \right) \\ &= L \left( \sum_{i=1}^n \mathbf{I}_{ij} \mathbf{v}_i \right) = \mathbf{L}\mathbf{v}_j \end{aligned}$$

## Matrices ortogonales y el teorema espectral

Una matriz  $\mathbf{Q}$  es **ortogonal** si

$$\mathbf{QQ}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$$

Esto obliga a que  $\mathbf{Q}$  sea cuadrada y no singular. Además,

$$\mathbf{Q}^{-1} = \mathbf{Q}^T$$

Con este nuevo concepto podemos establecer uno de los principales teoremas del álgebra lineal: el teorema espectral para matrices simétricas.

### TEOREMA 3

#### Teorema espectral para matrices simétricas

Si  $A$  es una matriz real simétrica, entonces existe una matriz ortogonal  $\mathbf{Q}$  tal que  $\mathbf{Q}^T\mathbf{AQ}$  es una matriz diagonal.

La ecuación

$$\mathbf{Q}^T\mathbf{AQ} = \mathbf{D}$$

es equivalente a

$$\mathbf{AQ} = \mathbf{QD}$$

Si  $\mathbf{D}$  es diagonal, las columnas  $\mathbf{v}_i$  de  $\mathbf{Q}$  obedecen la ecuación

$$\mathbf{Av}_i = d_{ii} \mathbf{v}_i$$

En otras palabras, las columnas de  $\mathbf{Q}$  forman un sistema ortonormal de vectores propios de  $\mathbf{A}$  y los elementos diagonales de  $\mathbf{D}$  son los valores propios de  $\mathbf{A}$ .

## Normas

Una **norma vectorial** en un espacio vectorial  $X$  es una función real en  $X$ , que se escribe como  $x \mapsto \|x\|$  y que tienen estas tres propiedades:

### ■ PROPIEDADES Propiedades de las normas vectoriales

1.  $\|x\| > 0$  para todos los vectores  $x$  diferentes de cero.
2.  $\|ax\| = |a|\|x\|$  para todos los vectores  $x$  y todos los escalares  $a$ .
3.  $\|x + y\| \leq \|x\| + \|y\|$  para todos los vectores  $x$  y  $y$ .

En  $\mathbb{R}^n$ , las normas vectoriales más simples son

$$\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n| \quad (\text{Norma vectorial } \ell_1)$$

$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \quad (\text{Euclídea/norma vectorial } \ell_2)$$

$$\|x\|_\infty = \max\{|x_1|, |x_2|, \dots, |x_n|\} \quad (\text{Norma vectorial } \ell_\infty)$$

Aquí,  $x_i$  denota la  $i$ -ésima componente del vector. Cualquier norma se puede pensar como la asignación de una *longitud* a cada vector. Es la norma euclídea la que corresponde directamente a nuestro concepto habitual de longitud, pero las otras normas son a veces mucho más convenientes para nuestros propósitos. Por ejemplo, si sabemos que  $\|x - y\|_\infty < 10^{-8}$ , entonces sabemos que cada componente de  $x$  difiere de la componente correspondiente de  $y$  a lo más en  $10^{-8}$  y que lo contrario también es cierto. Cuando resolvemos numéricamente un sistema de ecuaciones lineales  $\mathbf{Ax} = \mathbf{b}$ , se quiere saber (entre otras cosas) de qué tamaño es el vector residual. Este es convenientemente medido por  $\|\mathbf{Ax} - \mathbf{b}\|$ , donde se ha especificado alguna norma.

Las matrices de  $n \times n$  también pueden tener normas matriciales, sujetas a los requisitos siguientes:

### ■ PROPIEDADES Propiedades de normas matriciales

1.  $\|A\| > 0$  si  $A \neq 0$
2.  $\|\alpha A\| = |\alpha| \|A\|$
3.  $\|A + B\| \leq \|A\| + \|B\|$  (**desigualdad triangular**)

para matrices  $A, B$  y escalares  $\alpha$

Por lo general se prefieren normas matriciales que están relacionadas con una norma vectorial. Cuando una norma vectorial se ha especificado en  $\mathbb{R}^n$ , hay una forma estándar para introducir una **norma matricial** relacionada para matrices de  $n \times n$ , a saber,

$$\|A\| = \sup\{\|Ax\| : x \in \mathbb{R}^n, \|x\| \leq 1\}$$

Decimos que esta norma matricial es la **norma subordinada** a una norma vectorial dada o la **norma inducida** por la norma vectorial dada. La estrecha relación entre las dos es útil,

porque conduce a la siguiente desigualdad, que es válida para todos los vectores  $\mathbf{x}$ :

$$\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\| \|\mathbf{x}\|$$

La matriz de normas subordinadas a las normas vectoriales analizadas antes son, respectivamente,

$$\begin{aligned} \|\mathbf{A}\|_1 &= \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| && \text{(Norma matricial } \ell_1\text{)} \\ \|\mathbf{A}\|_2 &= \max_{1 \leq k \leq n} \sigma_k && \text{(Espectral/norma matricial } \ell_2\text{)} \\ \|\mathbf{A}\|_\infty &= \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| && \text{(Norma matricial } \ell_\infty\text{)} \end{aligned}$$

Aquí,  $\sigma_k$  son los valores singulares de  $\mathbf{A}$ . (Consulte la sección 8.2 para las definiciones.) De lo anterior, observe que la norma matricial subordinada a la norma vectorial euclídea no es lo que la mayoría de los estudiantes piensan que debería ser, a saber,

$$\|\mathbf{A}\|_{\text{F}} = \left\{ \sum_{i=1}^n \sum_{j=1}^n a_{ij}^2 \right\}^{1/2} \quad (\text{norma de Frobenius})$$

Ésta es, de hecho, una norma matricial, sin embargo, no es la inducida por la norma vectorial euclídea.

## Proceso de Gram-Schmidt

El **operador de proyección** se define como

$$\text{proy}_{\mathbf{y}} \mathbf{x} = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\langle \mathbf{y}, \mathbf{y} \rangle} \mathbf{y}$$

que proyecta el vector  $\mathbf{x}$  ortogonalmente en el vector  $\mathbf{y}$ . El proceso de Gram-Schmidt se puede escribir como

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{v}_1, & \mathbf{q}_1 &= \frac{\mathbf{z}_1}{\|\mathbf{z}_1\|} \\ \mathbf{z}_2 &= \mathbf{v}_2 - \text{proy}_{\mathbf{z}_1} \mathbf{v}_2, & \mathbf{q}_2 &= \frac{\mathbf{z}_2}{\|\mathbf{z}_2\|} \\ \mathbf{z}_3 &= \mathbf{v}_3 - \text{proy}_{\mathbf{z}_1} \mathbf{v}_3 - \text{proy}_{\mathbf{z}_2} \mathbf{v}_3, & \mathbf{q}_3 &= \frac{\mathbf{z}_3}{\|\mathbf{z}_3\|} \end{aligned}$$

En general, el paso  $k$  es

$$\mathbf{z}_k = \mathbf{v}_k - \sum_{j=1}^{k-1} \text{proy}_{\mathbf{v}_j} \mathbf{v}_k, \quad \mathbf{q}_k = \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|}$$

Aquí  $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots, \mathbf{z}_k\}$  es un conjunto ortogonal y  $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots, \mathbf{q}_k\}$  es un conjunto ortonormal. Cuando se implementa en un equipo, el proceso de Gram-Schmidt es numéricamente inestable debido a que los vectores  $\mathbf{z}_k$  no pueden ser exactamente ortogonales debido a errores de redondeo. Con una modificación menor, se puede estabilizar el proceso de Gram-Schmidt. En lugar de calcular los vectores  $\mathbf{u}_k$ , como antes, se puede calcular un término a la vez. Un algoritmo informático para el proceso de Gram-Schmidt modificado es

```

for $j = 1$ to k
 for $i = 1$ to $j - 1$
 $s \leftarrow \langle \mathbf{v}_j, \mathbf{v}_i \rangle$
 $\mathbf{v}_j \leftarrow \mathbf{v}_j - s \mathbf{v}_i$
 end for
 $\mathbf{v}_i \leftarrow \mathbf{v}_j / \|\mathbf{v}_j\|$
end for

```

Aquí, los vectores  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$  se remplazan con vectores ortonormales que extienden el mismo subespacio. El ciclo  $i$  elimina las componentes en la dirección  $\mathbf{v}_i$  seguido por la normalización del vector. En aritmética exacta, este cálculo arroja los mismos resultados que la forma original anterior. Sin embargo, produce pequeños errores en computadoras de aritmética de precisión finita.

**EJEMPLO 1** Considere los vectores  $\mathbf{v}_1 = (1, \varepsilon, 0, 0)$ ,  $\mathbf{v}_2 = (1, 0, \varepsilon, 0)$  y  $\mathbf{v}_3 = (1, 0, 0, \varepsilon)$ . Suponga que  $\varepsilon$  es un número pequeño. Realice el procedimiento de Gram-Schmidt estándar y el procedimiento modificado de Gram-Schmidt. Compruebe las condiciones de ortogonalidad de los vectores resultantes.

**Solución** Usando el proceso clásico de Gram-Schmidt, se obtiene  $\mathbf{u}_1 = (1, \varepsilon, 0, 0)$  y  $\mathbf{u}_2 = (0, -1, 1, 0)/\sqrt{2}$  y  $\mathbf{u}_3 = (0, -1, 0, 1)/\sqrt{2}$ . Utilizando el proceso de Gram-Schmidt modificado, encontramos que  $\mathbf{z}_1 = (1, \varepsilon, 0, 0)$ ,  $\mathbf{z}_2 = (0, -1, 1, 0)/\sqrt{2}$  y  $\mathbf{z}_3 = (0, -1, -1, 2)/\sqrt{6}$ . Como comprobación la ortogonalidad, encontramos  $\langle \mathbf{u}_2, \mathbf{u}_3 \rangle = \frac{1}{2}$  y  $\langle \mathbf{z}_2, \mathbf{z}_3 \rangle = 0$ . ■

# Respuestas a los problemas seleccionados\*

## Problemas 1.1

2.  $x = \frac{6032}{9990}$ ;  $x = \frac{6032}{10010}$       3.  $6 \times 10^{-5}$       4. Otras dos maneras:  $pi \leftarrow 2.0 \arcsen(1.0)$  o  $pi \leftarrow 2.0 \arccos(0.0)$

5a.  $sum \leftarrow 0$

```
for i = 1 to n do
 for j = 1 to n do
 sum ← sum + aij
 end for
end for
```

5d.  $sum \leftarrow 0.0$

```
for i = 1 to n do
 sum ← sum + aii
end for
for j = 2 to n do
 for i = j to n do
 sum ← sum + ai,i-j+1 + ai-j+1,i
 end for
end for
```

6.  $n$  multiplicaciones y  $n$  sumas/restas

8a. **for**  $i = 1$  to 5 **do**

```
 x ← x · x
```

```
end for
```

```
p ← x
```

8c.  $z \leftarrow x + 2$

```
p ← z3 (6 + z4 (9 + z8 (3 - z10)))
```

10.  $z \leftarrow a_n/b_n$

```
for i = 1 to n - 1 do
```

```
 z ← an-i(z + 1/bn-i)
```

```
end for
```

\*Las respuestas a los problemas marcados en el libro con el símbolo <sup>a</sup> se presentan aquí y en el manual de soluciones para el estudiante con más detalle.

**11b.**  $z \leftarrow 1$   
 $v \leftarrow 1$   
**for**  $i = 1$  **to**  $n - 1$  **do**  
 $v \leftarrow vx$   
 $z \leftarrow vz + 1$   
**end for**  
 $z \leftarrow vxz$

**12b.**  $v = \sum_{i=0}^n a_i x^i$

**12e.**  $v = a_n x^n + x \sum_{i=1}^n a_{n-i} x^{n-i}$

**13.**  $z = 1 + \sum_{i=2}^n \prod_{j=2}^i b_j$

**14.**  $n(n + 1)/2$

**15b.** **for**  $j = 1$  **to**  $n$  **do**  
**for**  $i = 1$  **to**  $n$  **do**  
 $a_{ij} \leftarrow 1.0/\text{real}(i + j - 1)$   
**end for**  
**end for**

## Problemas de cómputo 1.1

**4.**  $\exp(10) \approx 2.71828\ 18284\ 6$

**9.** El cálculo se desvía de la teoría por ejemplo cuando  $a_1 = 10^{-12}, 10^{-8}, 10^{-4}, 10^{20}$ .

**10.**  $x$  puede tener un subflujo y hacerse cero.    **12.** 40 diferentes deletreos

**20a.** El cálculo  $m/n$  que puede dar como resultado un truncamiento tal que  $x \neq y$ .

## Problemas 1.2

**4a.** Primera derivada  $+\infty$  en 0.

**4b.** Primera derivada no continua.

**4e.** Función  $-\infty$  en 0.

**5.**  $\cosh x = \sum_{k=0}^{\infty} \frac{x^{2k}}{(2k)!}; \quad \cosh 0.7 \approx 1.25517$

**6a.**  $e^{\cos x} = e \left( 1 - \frac{x^2}{2} + \dots \right)$

**6b.**  $\sin(\cos x) = (\sin 1) - (\cos 1) \left( \frac{x^2}{2} \right) + \dots$

**7.**  $m = 2$

**8.** Al menos 18 términos

**9.** Sí. Usando esta fórmula, evitamos la serie para  $e^{-x}$  y úsela para  $e^x$

**11.**  $\ln(1 - x) = -\sum_{k=1}^{\infty} \frac{x^k}{k}; \quad \ln\left(\frac{1+x}{1-x}\right) = 2 \sum_{k=1}^{\infty} \frac{x^{2k-1}}{(2k-1)}$

**12.**  $x = \frac{1}{3}, \quad \ln 2 = 0.69313$  (cuatro términos);    Al menos 10 términos.

**15a.**  $\sin x + \cos x = 1 + x - \frac{x^2}{2} - \frac{x^3}{6} + \dots; \quad \sin(0.001) + \cos(0.001) \approx 1.00099\ 94998\ 3$

**15b.**  $(\sin x)(\cos x) = x - \frac{2}{3}x^3 + \frac{2}{15}x^5 - \frac{4}{315}x^7 + \dots; \quad \sin(0.0006)\cos(0.0006) \approx 0.00059\ 99998\ 57$

**16.**  $\ln(e + x) = 1 + \sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{n} \left( \frac{x}{e} \right)^n$

**17.** Al menos siete términos.    **18.** Al menos 100 términos.    **20.**  $-\frac{5}{8}h^4$

**23.**  $\frac{1}{8} \left( x - \frac{17}{4} \right)$

**24.**  $s \leftarrow 0$

**for**  $i = 2$  **to**  $n$  **do**  
 $s \leftarrow s + \log(i)$   
**output**  $i, s$

**end for**

**28.**  $\left| \cos x - \left( 1 - \frac{x^2}{2} \right) \right| < \frac{1}{16 \times 24} = \frac{1}{384}$

**32.** Serie de Maclaurin:  $f(x) = 3 + 7x - 1.33x^2 + 19.2x^4$ ;

$$f(x) = 318.88 + (x-2)616.08 + \frac{(x-2)^2}{2!}918.94 + \frac{(x-2)^3}{3!}921.6 + \frac{(x-2)^4}{4!}460.8$$

**35.** 400 términos.

$$\mathbf{38.} \cos\left(\frac{\pi}{3} + h\right) = \frac{1}{2} \sum_{k=0}^{\infty} (-1)^k \frac{h^{2k}}{(2k)!} + \frac{\sqrt{3}}{2} \sum_{k=1}^{\infty} (-1)^k \frac{h^{2k-1}}{(2k-1)!}; \quad \cos(60.001^\circ) \approx 0.49998488$$

$$\mathbf{39.} \sin(45.0005^\circ) \approx 0.70711295 \quad \mathbf{42.} f(x-h) = (x-h)^m = x^m - mh x^{m-1} + m(m-1) \frac{h^2}{2!} x^{m-2} + \dots$$

$$\mathbf{47.} n = 16 \text{ o } n = 17 \quad \mathbf{50b.} \lim_{x \rightarrow 0} \frac{\arctan x}{x} = 1 \quad \mathbf{50c.} \lim_{x \rightarrow \pi} \frac{\cos x + 1}{\sin x} = 0 \quad \mathbf{51.} \text{Al menos 38 términos.}$$

$$\mathbf{52.} \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \left[ x - \frac{x^3}{3} + \frac{x^5}{5(2!)} - \frac{x^7}{7(3!)} + \dots \right]; \quad \operatorname{erf}(1) \approx 0.8382 \quad \mathbf{53.} 10^{10} \quad \mathbf{54.} 10^5$$

## Problemas de cómputo 1.2

|       | $c = 1$ | $c = 10^8$ |
|-------|---------|------------|
| $x_1$ | 0       | -1         |
| $x_2$ | $-10^8$ | $-10^8$    |

**14.**  $g$  converge más rápido (en cinco iteraciones) **16.**  $\lambda_{50} = 12586269025$       **17.**  $\alpha_{50} = 28143753123$

## Problemas 2.1

**1c.** [B5 000000]16

**2d.** [3FA 000000000000]16; [BFA 000000000000]16

**4d.** [3E7 00000]16, [3FCE 000000000000]16

**5d.**  $-\infty$     **8a.**  $-3.131968 \times 10^6$     **8d.**  $9.992892 \times 10^6$     **8g.**  $-3.39 \times 10^3$

**11c.**  $m = -1, 0, 1$ . Números de máquina no negativos: 0,  $\frac{1}{8}$ ,  $\frac{1}{4}$ ,  $\frac{3}{8}$ ,  $\frac{1}{2}$ ,  $\frac{3}{4}$ , 1,  $\frac{3}{2}$

**15.** 1    **17.** 1.00005;    **18.**  $|x| < 5 \times 10^{-5}$     **19.**  $\beta^{1-n}$

**21.**  $\approx 3 \times 2^{-25}$     **25.**  $\approx 3 \times 2^{-24}$     **26.**  $\approx 2^{-22}$     **30.**  $\approx n \times 2^{-24}$ ;  $n = 1000, \approx 2^{-14}$

**37.**  $\frac{1}{2} \times 10^{-12}$  redondeo;     $10^{-12}$  truncamiento    **38.** 9%    **39.** El error relativo no puede ser mayor que:  $05 \times 2^{-24}$ .

**42.**  $((q - 2^{-25}) 2^m, (q + 2^{-25}) 2^m)$

## Problemas 2.2

$$\mathbf{4.} y = \frac{\cos^2 x}{1 + \sin x} \quad \mathbf{6.} f(x) = -\frac{1}{2}x^3 - \frac{1}{2}x^4; \quad f(0.0125) \approx -9.888 \times 10^{-7}$$

$$\mathbf{8.} f(x) = \frac{1}{\sqrt{1+x^2+1}} + 3 - 1.7x^2; \quad f(0) = 3.5 \quad \mathbf{10.} f(x) = \frac{1}{\sqrt{x^2+1}+x}$$

$$\mathbf{11.} f(x) = \begin{cases} \ln(x + \sqrt{x^2 + 1}) & x > 0 \\ 0 & x = 0 \\ -\ln(-x + \sqrt{x^2 + 1}) & x < 0 \end{cases} \quad \mathbf{13.} z = \frac{x^4}{\sqrt{x^4 + 4} + 2}$$

$$\mathbf{16.} f(x) \approx 1 - x + \frac{x^2}{3} - \frac{x^3}{6}; \quad f(0.008) \approx 0.992020915 \quad \mathbf{20.} \arctan x - x \approx x^3 \left( -\frac{1}{3} + x^2 \left( \frac{1}{5} + x^2 \left( -\frac{1}{7} \right) \right) \right)$$

- 22.**  $(e^{2x} - 1)/2x \approx 1 + x(1 + x/3)(2 + x)$       **24a.** Cerca de  $\pi/2$ , la curva seno es relativamente plana.
- 26b.**  $\ln x - 1 = \ln(x/e)$       **26d.**  $x^{-2}(\sin x - e^x + 1) \approx -\frac{1}{2} - \frac{x}{3}$  cuando  $x \rightarrow 0$
- 28.**  $|x| < \sqrt{6\varepsilon}$ , donde  $\varepsilon$  es la precisión de la máquina    **29.**  $x_1 \approx 10^5$ ,  $x_2 \approx 10^{-5}$
- 30.** No mucho. Excepto para calcular  $b^2 - 4ac$  con doble precisión

## Problemas de cómputo 2.2

- 1.** No hay solución;  $(0, 0)$ ;  $(0, 0)$ ; Cualquier solución;  $(-1, 0)$ ;  $(-0.1020842383, -4.8979157617)$ ;  $(4.0000000001, 4.00099999)$ ;  $(-0.1020842383, -4.8979157617)$ ;  $(1.000000000, 1.000000000E34)$ ;  $(1.9968377223, 2.0031622777)$

| 10. $x$ | Series                         | $n$ |
|---------|--------------------------------|-----|
| 0       | 1.0                            | 1   |
| 1       | 2.7182818285                   | 10  |
| -1      | 0.3678794412                   | 10  |
| 0.5     | 1.6487212707                   | 8   |
| -0.123  | 0.8842636626                   | 5   |
| -25.5   | $8.4234637545 \times 10^{-12}$ | 25  |
| -1776   | 0                              | 25  |
| 3.14159 | 23.1406312270                  | 17  |

- 14.**  $|x| < 10^{-15}$       **15.**  $\rho_{50} = 2.85987$

## Problemas 3.1

- 1.** 0.61906; 1.51213  
**4.**  $\left\{-\frac{\pi}{4} - \delta, 0, \frac{\pi}{4} + \varepsilon, \frac{3\pi}{4} + \varepsilon, \frac{5\pi}{4} + \varepsilon, \dots\right\}$ , donde  $\delta \approx 0.2$  y  $\varepsilon$  inicia aproximadamente en 0.4 y decrece.  
**9.**  $\left\{0, \pm\frac{\pi}{2}, \pm\pi, \pm\frac{3\pi}{2}, \pm2\pi, \dots\right\}$       **10.**  $x = 0$

- 12.** Si el intervalo original tiene un ancho  $h$ , entonces después de, digamos,  $k$  pasos, hemos reducido el intervalo que contiene la raíz al ancho  $h^{2-k}$ . A partir de esto, agregamos un bit en cada paso. Se necesitan cerca de tres pasos para cada dígito decimal.  
**17.** 20 pasos      **18b.** Esto podría ser falso, ya que si  $r$  está más cerca de  $b_n$ , entonces  $r - a_n \approx b_n - a_n = 2^{-n}(b_0 - a_0)$ .  
**18d.** Esto es cierto, ya que  $0 \leq r - a_n$  (obvio) y  $r - a_n \leq b_n - a_n = 2^{-n}(b_0 - a_0)$ .      **19a.** Falso en algunos casos.  
**19e.** Verdadero.      **21.**  $n \geq 23$ .      **23.** No; No.

## Problemas de cómputo 3.1

- 10.**  $1, 2, 3, 3 - 2i, 3 + 2i, 5 + 5i, 5 - 5i, 16$       **11.** 2.365

## Problemas 3.2

- 3.**  $x_{n+1} = \frac{1}{2}[x_n + 1/(Rx_n)]$       **4.** 0.79; 1.6      **7.**  $y = \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}\left(1 - \frac{\pi}{4}\right)$       **9.**  $\pi$   
**11.**  $x_{n+1} = 2x_n / (x_n^2 R + 1)$ ; -0.49985      **12a.** Sí,  $-\sqrt[3]{R}$ .      **13a.**  $x_{n+1} = \frac{1}{3}(2x_n + R/x_n^2)$

- 13c.**  $x_{n+1} = x_n(x_n^3 + 2R)/(2x_n^3 + R)$     **13e.**  $x_{n+1} = \frac{x_n}{3R}(4R - x_n^3)$     **13g.**  $x_{n+1} = \frac{R}{x_n^2}(2x_n^6 + 1)/(2Rx_n^3 + 1)$
- 15.**  $x_1 = \frac{1}{2}$     **17.**  $x_{n+1} = -\frac{1}{2}$     **19.**  $|x_0| < \sqrt{3}$     **21.** El método de Newton se repite si  $x_0 \neq 0$ .
- 22.**  $x \leftarrow R$   
**for**  $n = 1$  **to**  $n\_max$  **do**  
 $x \leftarrow (2x + Rx^2)/3$   
**end for**
- 27.**  $x_{n+1} = [(m-1)x_n^m + R]/(mx_n^{m-1})$ ;     $x_{n+1} = x_n[(m+1)R - x_n^m]/(mR)$
- 29.** Diverge.    **31.**  $x_{n+1} = x_n - \frac{f(x_n)f'(x_n)}{[f'(x_n)]^2 - f(x_n)f''(x_n)}$
- 32.**  $x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} + \frac{\sqrt{[f'(x_n)]^2 - 2f(x_n)f''(x_n)}}{f''(x_n)}$
- 35.**  $e_{n+1} = e_n^2 \left[ \frac{\frac{f^{(m+1)}(\eta_n)}{m!} - \frac{f^{(m+1)}(\xi_n)}{(m+1)(m-1)!}}{\frac{f^{(m)}(r)}{(m-1)!} + \frac{e_n f^{(m+1)}(\eta_n)}{m!}} \right]$
- 36.**  $e_{n+1} = \frac{1}{2} e_n^2 \frac{f''}{g}$     **37.**  $|g'(r)| < 1$  si  $0 < \omega < 2$     **41.** 4º orden

## Problemas de cómputo 3.2

- 4.** 0.32796 77853 31818 36223 77546    **5.** 2.09455 14815 42326 59148 23865 40579  
**8.** 1.83928 67552    **9.** 0.47033 169    **10a.** 1.89549 42670 340    **10b.** 1.99266 68631 307  
**10c.** 0.51097 34293 8857    **10d.** 2.58280 14730 552    **14.** 3.13108; 3.15145 (dos raíces cercanas)

## Problemas 3.3

- 1.** 2.7385    **3.**  $-\frac{3}{2}$     **4.**  $\ln 2$     **9.**  $x_{n+1} = x_n - \frac{x_n^2 - R}{x_n + x_{n-1}}$
- 12.**  $e_{n+1} = \left[ 1 - \left( \frac{x_n - x_0}{f(x_n) - f(x_0)} \right) f'(\xi_n) \right] e_n$     **13a.** Convergencia lineal
- 13c.** Convergencia cuadrática    **15.** Muestre  $|\xi - x_{n+1}| \leq c|\xi - x_n|$ .    **16.**  $\sqrt{2}$     **17.**  $x = 4.510187$

## Problemas de cómputo 3.3

- 1.** -0.45896; 3.73308    **6a.** 1.53209    **6b.** 1.23618    **7.** 1.36880 81078 21373    **9.** 20.80485 4

## Problemas 4.1

- 1.**  $p_3(x) = 7 - 2x + x^3$     **3.**  $\ell_2(x) = -(x-4)(x^2-1)/8$   
**7a.**  $p_3(x) = 2 + (x+1)(-3 + (x-1)(2 + (x-3)(-11/24)))$   
**8.**  $p_4(x) = -1 + (x-1) \left( \frac{2}{3} + (x-2) \left( \frac{1}{8} + (x-2.5) \left( \frac{3}{4} + (x-3) \frac{11}{6} \right) \right) \right)$

$$\begin{array}{r} \text{9a. } \left| \begin{array}{cc|c} 0 & 1 & \\ 1 & \boxed{9} & 8 \\ 2 & 23 & \boxed{14} \\ 4 & 93 & 35 \\ 6 & 259 & 83 \end{array} \right| \quad \left| \begin{array}{cc|c} 3 & 1 & \\ 7 & \boxed{1} & 0 \\ 12 & & \end{array} \right| \end{array}$$

**9b.**  $f(4.2) = 104.49$     **12.**  $q(x) = x^4 - x^3 + x^2 - x + 1 - \frac{31}{120}(x+2)(x+1)(x)(x-1)(x-2)$

**13a.**  $x^3 - 3x^2 + 2x - 1$     **14.**  $p(x) = x - 2.5$     **16.**  $a_0 = \frac{1}{2}$

**18.**  $2 + x(-1 + (x-1)(1-(x-3)x))$

**19.**  $p_4(x) = -1 + 2(x+2) - (x+2)(x+1) + (x+2)(x+1)x$ ;     $p_2(x) = 1 + 2(x+1)x$

**22.**  $p(x) = 0.76(x-1.73)(x-1.82)(x-5.22)(x-8.26)$     **25.** 1.5727; No tiene ventaja

**27.**  $p(x) = -\frac{3}{5}x^3 - \frac{2}{5}x^2 + 1$     **28.** 0.38099; 0.077848

**39.** 0.85527; 0.87006    **40.** Divisiones:  $\frac{1}{2^n}(n-1)$ ; sumas/restas:  $n(n-1)$

**42.** cero    **45.** Falso, sólo para un polinomio  $p$  de grado  $\leq n-1$ .

## Problemas de cómputo 4.1

**1.**  $p(x) = 2 + 46(x-1) + 89(x-1)(x-2) + 6(x-1)(x-2)(x-3) + 4(x-1)(x-2)(x-3)(x+4)$

## Problemas 4.2

**1.**  $f[x_0, x_1, x_2, x_3, x_4] = 0$     **6.**  $1.25 \times 10^{-5}$     **7.** Errores:  $8.1 \times 10^{-6}$ ,  $6.1 \times 10^{-6}$     **8.** 497 entradas de la tabla

**9.**  $4.105 \times 10^{-14}$  (Thm 1),  $1.1905 \times 10^{-23}$  (Thm 2)    **10.**  $2.6 \times 10^{-6}$     **13.**  $n \geq 7$

**14.**  $\prod_{i=0}^{n-1} |x - x_i| \leq \frac{h^n (2n)!}{2^{2n} n!}$     **16.** Sí.

## Problemas 4.3

**1.**  $-hf''(\xi)$     **2.** Término de error  $= -hf''(\xi)$  para  $\xi \in (0, 2h)$     **4.** No existe dicha fórmula.

**6.** El punto  $\xi$  para la primera serie de Taylor es tal que  $\xi \in (x, x+h)$ , mientras el segundo es  $\xi \in (x-h, x)$ . Claramente no son los mismos

**8a.**  $-\frac{2}{3}h^2 f'''(\xi)$     **9a.**  $-\frac{h^2}{4} f^{(5)}(\xi)$     **9b.**  $-\frac{h^2}{6} f^{(6)}(\xi)$

**11.**  $\alpha = 1$ , término de error  $= -\frac{h^2}{6} f'''(\xi)$ ;  $\alpha \neq 1$ , término de error  $= -(\alpha-1)\frac{h}{2}f''(\xi)$

**12.** Término de error  $= -\frac{h^2}{6} \left[ f'''(\xi_1) + \frac{1}{2}f^{(4)}(\xi_2) \right]$  para alguna  $\xi_i \in (x-h, x+h)$ .    **13.**  $p' \left( \frac{x_0 + x_1}{2} \right) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$

**16.**  $L \approx 2\varphi \left( \frac{h}{2} \right) - \varphi(h)$     **20.**  $L \approx \left\{ \left[ \varphi \left( \frac{h}{2} \right) \right]^2 - \varphi(h)\varphi \left( \frac{h}{3} \right) \right\} \Bigg/ \left\{ 2\varphi \left( \frac{h}{2} \right) - \varphi(h) - \varphi \left( \frac{h}{3} \right) \right\}$

## Problemas de cómputo 4.3

3. 0.20211 58503

## Problemas 5.1

1.  $\frac{7}{18}$     3. 0.00010 00025 0006    6.  $n \geq 56738$     7.  $U - L = \frac{1}{n}[f(1) - f(0)]$   
 11.  $L(f; P) \leq M(f; P) \leq U(f; P)$

## Problemas de cómputo 5.1

2. 0.94598 385; 0.94723 395    4. 4.84422

## Problemas 5.2

1.  $\approx 0.70833$

2.  $T(f; P) = 0.775$ ;  $\int_0^1 \frac{dx}{x^2 + 1} \approx 0.7854$ ; Error = 0.0104

| $h$ | 2 | 1 | 1/2 | 1/4 |
|-----|---|---|-----|-----|
| $L$ | 0 | 0 | 1/2 | 3/4 |
| $U$ | 2 | 2 | 3/2 | 5/4 |
| $T$ | 2 | 1 | 1   | 1   |

7.  $n \geq 16\,07439$ ; demasiado pequeño    8.  $T = \frac{1}{n^3} \left[ \frac{1}{6}(n-1)(2n-1)n \right] + \frac{1}{2n}$

12. 0.000025    13.  $T(f; P) \approx 4.37132$     14.  $T(f; P) \approx 0.43056$     15.  $|$  término de error  $| \leq 0.3104$

16.  $T(f; P) = 7.125$ ; No, no se pueden calcular a partir de los datos dados.

17a.  $= -\frac{(b-a)h}{2} f'(\xi)$  para alguna  $\xi \in (a, b)$ .    17b.  $= -\frac{(b-a)h^2}{6} f''(\xi)$  para alguna  $\xi \in (a, b)$ .

18a.  $\frac{1}{24}h^3 f'''(\xi)$     18b.  $\frac{1}{24} \sum_{i=1}^n h_i^3 f''(\xi_i)$     18c.  $\frac{b-a}{24}h^2 f''(\xi)$

24.  $f(x) = x^n$  ( $n > 3$ ) o  $[0, 1]$ , con partición  $\{0, 1\}$

25.  $L \leq \int_a^b f(x) dx \leq T \leq U$     26.  $n \geq 1155$     29.  $-(b-a)hf'(\xi)/2$

30.  $\int_a^b f(x) dx = h \sum_{i=0}^{2^n} f(a + ih) + E$ , donde  $E = \frac{1}{2}(b-a)hf''(\xi)$  para  $\xi \in (a, b)$

## Problemas de cómputo 5.2

- 2a. 2    2b. 1.71828    2c. 0.43882

## Problemas 5.3

1. 13    3.  $-\frac{136}{15}$     5. 4.267    7. No bien.

8.  $R(1, 1) = \frac{h}{3}\{f(-h) + 4f(0) + f(h)\}$  Regla de Simpson    10.  $1 + 2^{m-1}$

13.  $R(2, 2) = \frac{2h}{45} [7f(a) + 32f(a+h) + 12f(a+2h) + 32f(a+3h) + 7f(b)]$

**14.**  $X = (27v - u)/26$     **15.**  $Z = \frac{4096}{2835}f\left(\frac{h}{8}\right) - \frac{1344}{2835}f\left(\frac{h}{4}\right) + \frac{84}{2835}f\left(\frac{h}{2}\right) - \frac{1}{2835}f(h)$

**17.**  $x_{n+1} + n^3(x_{n+1} - x_n)/(3n^2 + 3n + 1)$     **18.**  $|I - R(n, m)| = \mathcal{O}(h^{2m})$  cuando  $h \rightarrow 0$

**22.**  $R(n+1, m+1) = R(n+1, m) + [R(n+1, m) - R(n, m)]/(8^m - 1)$

**23.** Muestre  $\int_a^b f(x) dx - R(n, 0) \approx c4^{-(n+1)}$ .    **24.** Sea  $m = 1$  y sea  $n \rightarrow \infty$  en la fórmula (2).

**27.**  $E = A_{2m}(2\pi) \left(\frac{2\pi}{4}\right)^{2m} [\pm 4^{2m} \cos(4\xi)] \pm (2\pi)^{2m+1} 4^{2m+1} A_{2m} \cos(4\xi)$

## Problemas de cómputo 5.3

**1.**  $R(7, 7) = 0.499969819$     **5.**  $R(5, 0) = 1.813799364$     **6.**  $\frac{2}{9} = 0.22222 \dots$

**7.**  $0.62135732$     **11.**  $R(7, 7) = 0.765197687$

## Problemas 6.1

**1.**  $\frac{\pi}{4}$     **2a.**  $h < 0.03$  o  $n > 33.97$ .    **2b.**  $h < 0.15$  o  $n > 7.5$ .

**3a.**  $7.1667$     **3b.**  $7.0833$     **3c.**  $7.0777$     **4.**  $\int_1^2 \frac{dx}{x} = 0.6933$ ; El límite es  $5.2 \times 10^{-4}$ .

**7.**  $\int_a^b f(x) dx = \frac{16}{15}S_{2(n-1)} - \frac{1}{15}S_{n-1}$     **8.**  $-\frac{3}{80}h^5 f^{(4)}(\xi)$

## Problemas 6.2

**1.**  $\approx 0.91949$     **4a.**  $x = \pm \sqrt{\frac{1}{3}}$     **4b.**  $x = \pm 0.861136$ ,  $\pm 0.339981$

**5.**  $\alpha = \gamma = \frac{4}{3}$ ,  $\beta = -\frac{2}{3}$     **6.**  $A = (b - a)$ ,  $B = \frac{1}{2}(b - a)^2$

**7.**  $\frac{5h}{12}f(a) + \frac{2h}{3}f(a+h) - \frac{h}{12}f(a+2h)$     **9.**  $\alpha = \sqrt{\frac{5}{7}}$ ,  $a = c = \frac{7}{25}$ ,  $b = \frac{8}{75}$

**10.**  $w_1 = w_2 = \frac{h}{2}$ ,  $w_3 = w_4 = -\frac{h^3}{24}$     **11.**  $A = 2h$ ,  $B = 0$ ;  $C = \frac{h^3}{3}$

**12.**  $A = \frac{8}{3}$ ,  $B = -\frac{4}{3}$ ,  $C = \frac{8}{3}$  Sí. Exacto para polinomios de grado  $\leq 3$ .

**13.**  $A = \frac{h}{3}$ ,  $D = 0$ ,  $C = \frac{h}{3}$ ,  $B = \frac{4}{3}h$     **14.** Verdadero para  $n \leq 3$

## Problemas de cómputo 6.2

**2a.**  $1.4183$     **8a.**  $2.034805318577$     **8b.**  $0.892979511569$     **8c.**  $0.43398771$

## Problemas 7.1

**1.** Homogénea:  $\alpha = 0$ , solución cero;  $\alpha = \pm 1$ , número infinito de soluciones

**2.** En  $\alpha \approx 1$ , se produce la respuesta errónea    **3a.** No hay solución    **3b.** Número infinito de soluciones

**4.**  $\begin{cases} x_1 = -697.3 \\ x_2 = 343.9 \end{cases}$      $\begin{cases} x_1 = -720.79976 \\ x_2 = 356.28760 \end{cases}$

**5.**  $r = \begin{bmatrix} -0.001343 \\ -0.001572 \end{bmatrix}$ ,  $\hat{r} = \begin{bmatrix} -0.0000001 \\ 0.0000000 \end{bmatrix}$ ,  $e = \begin{bmatrix} -0.001 \\ -0.001 \end{bmatrix}$ ,  $\hat{e} = \begin{bmatrix} -0.659 \\ 0.913 \end{bmatrix}$

**6a.**  $x_2 = 1$ ,  $x_1 = 0$     **6b.**  $x_2 = 1$ ,  $x_1 = 1$     **6c.** Se  $a b_1 = b_2 = 1$ . Entonces  $x_2 = 1$ ,  $x_1 = 0$ , que es exacta

- 7a.**  $x_1 = 2, \quad x_2 = 1, \quad x_3 = 0 \quad \quad \text{7b. } x_1 = x_2 = x_3 = 1$   
**7c.**  $x_1 \approx -7.233, \quad x_2 \approx 1.133, \quad x_3 \approx 2.433, \quad x_4 = 4.5$

## Problemas de cómputo 7.1

- 6.**  $z = [2i, i, i, i]^T, \lambda = 1 + 5i; \quad z = [1, 2, 1, 1]^T, \lambda = 2 + 6i; \quad z = [-i, -i, 0, -i]^T, \lambda = -3 - 7i;$   
 $z = [1, 1, 1, 0]^T, \lambda = -4 - 8i$   
**7a.**  $(3.75, 90^\circ); \quad (3.27, -65.7^\circ); \quad (0.775, 172.9^\circ) \quad \quad \text{7b. } (2.5, -90^\circ); \quad (2.08, 56.3^\circ); \quad (1.55, -60.2^\circ)$

## Problemas 7.2

**1.**  $\begin{bmatrix} 1/2 & 5/2 & -4 & -1 \\ 1/4 & -1/2 & -5/19 & -62/19 \\ 3/4 & 9/10 & 38/5 & 9/10 \\ 4 & 1 & 0 & 4 \end{bmatrix} \quad \text{2. } x = [1/3, 3, 1/3]^T$

**3.**  $\begin{bmatrix} 1 & 0 & 3 & 0 \\ 0 & 1 & 3 & -1 \\ 3 & -3 & 0 & 6 \\ 0 & 2 & 4 & -6 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 & 3 & -2 \\ 0 & 1 & 3 & -1 \\ 3 & -3 & 0 & 6 \\ 0 & 2 & 4 & -6 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 1 & 3 & -2 \\ 0 & 0 & 0 & 1 \\ 3 & -3 & 0 & 6 \\ 0 & 0 & -2 & -2 \end{bmatrix}$

**5.**  $\begin{bmatrix} 1/4 & 5/2 & 7/4 & 1/2 \\ 4 & 2 & 1 & 2 \\ 1/2 & 0 & 5/9 & 17/9 \\ 1/4 & 3/5 & 27/10 & 1/5 \end{bmatrix}$

**6.**  $\ell = (1, 3, 2)$ , el segundo renglón pivote es el tercer renglón. **8.**  $x_3 = -1, \quad x_2 = 1, \quad x_1 = 0$

**10.**  $x_4 = -1, \quad x_3 = 0, \quad x_2 = 2, \quad x_1 = 1 \quad \quad \text{13b. } x_3 = 1, \quad x_2 = 1, \quad x_1 = 1$

**13d.**  $x_1 \approx 4.267, \quad x_2 \approx -4.133, \quad x_3 \approx -2.467 \quad \quad \text{17. } n(n+1)$

**18.**  $\left[ \frac{29}{10}(n^2 - 1) + \frac{7}{30}n(n-1)(2n-1) \right] 10^{-6}$  segundos

| <b>19.</b> $n$ | 10                             | $10^2$          | $10^3$   | $10^4$      |
|----------------|--------------------------------|-----------------|----------|-------------|
| Tiempo         | $\frac{1}{3} \times 10^{-3}$ s | $\frac{1}{3}$ s | 5.56 min | 3.86 días   |
| Costo          | 0.005¢                         | 5¢              | \$46.30  | \$46 296.30 |

**21.** Resuelva esto:  $U^T y = b, \quad L^T x = y \quad \quad \text{23a. } x_1 = \frac{5}{9}, \quad x_2 = \frac{2}{9}, \quad x_3 = \frac{1}{9} \times 10^{-9}$

## Problemas de cómputo 7.2

- 2.**  $[3.4606, 1.5610, -2.9342, -0.4301]^T \quad \quad \text{3. } [6.7831, 3.5914, -6.4451, -1.5179]^T$   
**4.**  $2 \leq n \leq 10, x_i \approx 1$  para toda  $i$ ; para  $n$  grande, hay muchas  $x_i \neq 1 \quad \quad \text{5. } b_i = n^2 + 2(i-1)$   
**6.**  $x_2 = 1, \quad x_i = 0, \quad \text{para } i \neq 2$

## Problemas 7.3

- 2a.**  $5n - 4 \quad \quad \text{3. } n + 2nk - k(k + 1) \quad \quad \text{6. } \text{Sí, así es.}$   
**7.**  $D^{-1}AD = \text{tridiagonal} \left[ \pm \sqrt{a_{i-1}c_{i-1}}, \quad d_i, \quad \pm \sqrt{a_i c_i} \right]$

## Problemas de cómputo 7.3

3. 
$$\begin{cases} d_i \leftarrow d_i - 1/d_{i-1} \\ b_i \leftarrow b_i - b_{i-1}/d_{i-1} \quad (2 \leq i \leq n) \end{cases}$$

4. 
$$\begin{cases} x_1 = 1 \\ x_i = 1 - (4x_{i-1})^{-1} \quad (2 \leq i \leq 100) \end{cases}$$

12. 
$$\begin{cases} c_i \leftarrow c_i/d_i \\ b_i \leftarrow b_i/d_i \\ d_{i+1} \leftarrow d_{i+1} - a_{i+1}c_i \\ b_{i+1} \leftarrow b_{i+1} - a_{i+1}b_i \quad (1 \leq i \leq n-1) \end{cases}$$

$$\begin{cases} x_n \leftarrow b_n \\ x_i \leftarrow (b_i - x_{i+1})/x_i \quad (n-1 \geq j \geq 1) \end{cases}$$

11a. 
$$\begin{cases} x_1 \leftarrow b_1/a_{11} \\ x_i \leftarrow \left( b_i - \sum_{j=1}^{n-1} a_{ij}x_j \right) / a_{ii} \quad (2 \leq i \leq n) \end{cases}$$

$$\begin{cases} b_n \leftarrow b_n/d_n \\ b_i \leftarrow b_i - c_i b_{i+1} \quad (1 = n-1, \dots, 1) \end{cases}$$

## Problemas 8.1

1a.  $L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/3 & -3 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 3 & 0 & 3 \\ 0 & -1 & 3 \\ 0 & 0 & 8 \end{bmatrix}$

2a.  $M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ -5 & 0 & -2 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

3a.  $M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 \\ 0 & 0 & -2 & 1 & 0 \\ -4 & 0 & 0 & 0 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 25 & 0 & 0 & 0 & 1 \\ 0 & 27 & 4 & 3 & 2 \\ 0 & 0 & 50 & -6 & -4 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 \end{bmatrix}$

5a.  $M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -x/b & 1 & 0 \\ -w/a & (xy)/(bc) & -y/c & 1 \end{bmatrix} \quad U = \begin{bmatrix} a & 0 & 0 & z \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d - (wz)/a \end{bmatrix}$

5b.  $L = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & x & c & 0 \\ 0 & 0 & y & d - (wz)/a \end{bmatrix} \quad U' = \begin{bmatrix} 1 & 0 & 0 & z/a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

6a.  $L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/4 & 1 & 0 & 0 \\ -1/4 & -1/15 & 1 & 0 \\ 0 & -4/15 & -2/7 & 1 \end{bmatrix} \quad U = \begin{bmatrix} 4 & -1 & -1 & 0 \\ 0 & 15/4 & -1/4 & -1 \\ 0 & 0 & 56/15 & -16/15 \\ 0 & 0 & 0 & 24/7 \end{bmatrix}$

6b.  $D = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 15/4 & 0 & 0 \\ 0 & 0 & 56/15 & 0 \\ 0 & 0 & 0 & 24/7 \end{bmatrix} \quad U' = \begin{bmatrix} 1 & -1/4 & -1/4 & 0 \\ 0 & 1 & -1/15 & -4/15 \\ 0 & 0 & 1 & -2/7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

6c.  $L' = \begin{bmatrix} 4 & 0 & 0 & 0 \\ -1 & 15/4 & 0 & 0 \\ -1 & -1/4 & 56/15 & 0 \\ 0 & -1 & -16/15 & 24/7 \end{bmatrix}$

6d.  $L' = \begin{bmatrix} 2 & 0 & 0 & 0 \\ -1/2 & (1/2)\sqrt{15} & 0 & 0 \\ -1/2 & -1/(2\sqrt{15}) & 2\sqrt{14/15} & 0 \\ 0 & -2/(\sqrt{15}) & -(4/7)\sqrt{14/15} & 2\sqrt{6/7} \end{bmatrix}$

6e. 192

8.  $\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & -8 \end{bmatrix}$      $\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 1 & -1 & 1 & 1 \end{bmatrix}$

9a.  $\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 3 & -1 & 1 \end{bmatrix}$      $\mathbf{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$      $\mathbf{U}' = \begin{bmatrix} 1 & -1/2 & 1 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{bmatrix}$     9b.  $\mathbf{x} = [-1, 2, 1]^T$

10a.  $\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 3 & 1 \end{bmatrix}$ ,     $\mathbf{D} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$ ,     $\mathbf{U}' = \begin{bmatrix} 1 & -1/2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$     10b.  $\mathbf{x} = [-1, 1, 1]^T$

12.  $A^{-1} = \frac{1}{15} \begin{bmatrix} 11 & -5 & -7 \\ -13 & 10 & 11 \\ -8 & 5 & 1 \end{bmatrix}$     14a.  $\begin{bmatrix} \ell_{11} & \ell_{11}u_{12} & 0 & 0 \\ \ell_{21} & \ell_{21}u_{12} + \ell_{22} & \ell_{22}u_{23} & 0 \\ 0 & \ell_{32} & \ell_{32}u_{23} + \ell_{33} & \ell_{33}u_{34} \\ 0 & 0 & \ell_{43} & \ell_{43}u_{34} + \ell_{44} \end{bmatrix}$

16a.  $\mathcal{X}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & 1 & -1 & 1 \\ -1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$     16b.  $\mathcal{X}^{-1} = \begin{bmatrix} 0 & -1 & -1 & 1 \\ -1 & 0 & -1 & 1 \\ -1 & -1 & 0 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$

## Problemas de cómputo 8.1

3. Caso 4:  $p_5(A) = \begin{bmatrix} 536 & -668 & 458 & -186 \\ -668 & 994 & -854 & 458 \\ 458 & -854 & 994 & -668 \\ -186 & 458 & -668 & 536 \end{bmatrix}$

## Problemas 8.2

3. d.    5. e.    9. b.

## Problemas 8.3

9. c.    11. d.

## Problemas de cómputo 8.3

11. Valores propios/vectores propios: 1,  $(-1, 1, 0, 0)$ ; 2,  $(0, 0, -1, 1)$ ; 5,  $(-1, 1, 2, 2)$

## Problemas 8.4

1. a.

## Problemas 9.1

1. Sí

6. En el problema 9.1.5, la expresión entre paréntesis es  $f'(x_1) - f'(x_2)$  y la magnitud no es mayor de  $2C$ .

9. Nudos  $\geq 50\pi 10^8 \approx 1.57 \times 10^{10}$ .

10.  $\sum_{i=1}^n f(t_i)S_i$  es una combinación lineal de las funciones spline de primer grado que tiene los nudos  $t_0, \dots, t_n$ . Por tanto, es también una función. Su valor en  $t_j$  es  $\sum_{i=1}^n f(t_i)S_i f(t_j) = f(t_j)$ .  $S_i(x) = 0$  si  $x < t_{i-1}$  o  $x > t_{i+1}$ . En  $(t_{i-1}, t_i)$ ,  $S_i(x)$  está dada por  $(x - t_{i-1})/(t_i - t_{i-1})$ . En  $(t_i, t_{i+1})$ ,  $S_i(x)$  está dado por  $(x - t_{i+1})/(t_i - t_{i+1})$ .  $S_0$  y  $S_n$  son ligeramente diferentes.

- 12.** Si  $S$  es cuadrática por partes, entonces obviamente  $S'$  es lineal por partes. Si  $S$  es un spline cuadrático entonces  $S \in C^1$ . Por lo tanto  $S' \in C$ . Por lo tanto  $S'$  es lineal y continua por partes.
- 17.**  $\begin{cases} Q_0(x) = -(x+1)^2 + 2, & Q_1(x) = -2x + 1, & Q_2(x) = 8\left(x-\frac{1}{2}\right)^2 - 2\left(x-\frac{1}{2}\right) \\ Q_3(x) = -5(x-1)^2 + 6(x-1) + 1, & Q_4(x) = 12(x-2)^2 - 4(x-2) + 2 \end{cases}$
- 19.** La respuesta está dada por la ecuación (8). **20a.** Sí **20b.** No **20c.** No **21.** Sí

## Problemas 9.2

**1.** No **2.** No **4.**  $a = -4$ ,  $b = -6$ ,  $c = -3$ ,  $d = -1$ ,  $e = -3$

**5.**  $a = -5$ ,  $b = -26$ ,  $c = -27$ ,  $d = \frac{27}{2}$  **6.** No

**7a.**  $S(x)$  no es continua en  $x = -1$ .  $S'''(x)$  no es continua en  $x = -1$ ,

**8a.**  $(m+1)n$  **8b.**  $2n$  **8c.**  $(m-1)(n-1)$  **8d.**  $m-1$

$$\mathbf{10.} \quad S = \begin{cases} x^2 & [0, 1] \\ 1 + 2(x-1) + (x-1)^2 + (x-1)^3 & [1, 2] \\ 5 + 7(x-2) + 4(x-2)^2 & [2, 3] \end{cases}$$

**12.**  $a = 3$ ,  $b = 3$ ,  $c = 1$  **13.** No **15.**  $a = -1$ ,  $b = 3$ ,  $c = -2$ ,  $d = 2$  **17.**  $n+3$

**19.**  $f$  no es un spline cúbico **22.**  $p_3(x) = x - 0.0175x^2 + 0.1927x^3$ ; No **26.**  $S$  es lineal

$$\mathbf{32.} \quad S_0(x) = \left(-\frac{5}{7}\right)(x-1)^3 + \left(\frac{12}{7}\right)(x-1)$$

$$S_1(x) = \left(\frac{6}{7}\right)(x-2)^3 - \left(\frac{5}{7}\right)(3-x)^3 - \left(\frac{6}{7}\right)(x-2) + \left(\frac{12}{7}\right)(3-x)$$

$$S_2(x) = \left(-\frac{5}{7}\right)(x-3)^3 + \left(\frac{6}{7}\right)(4-x)^3 + \left(\frac{12}{7}\right)(x-3) - \left(\frac{6}{7}\right)(4-x)$$

$$S_3(x) = \left(-\frac{5}{7}\right)(5-x)^3 + \left(\frac{12}{7}\right)(5-x)$$

**33.** La condición de  $S$  la hace una función par. Si  $S(x) = S_0(x)$  en  $[-1, 0]$  y  $S(x) = S_1(x)$  en  $[0, 1]$ , entonces  $S_1(0) = 1$ ,  $S'_1(0) = 0$ ,  $S''_1(1) = 0$  y  $S_1(1) = 0$ . Un cálculo fácil produce

$$S_1(x) = 1 - \frac{3}{2}x^2 + \frac{1}{2}x^3.$$

**38.**  $5n, n+4$  **39.** Sí

## Problemas 9.3

**2.** Relación de recurrencia de los polinomios de Chebyshev. Véase la sección 12.2

$$\mathbf{3.} \quad B_i^2(x) = \begin{cases} \frac{(x-t_i)^2}{(t_{i+2}-t_i)(t_{i+1}-t_i)}, & \text{en } [t_i, t_{i+1}] \\ \frac{(x-t_i)(t_{i+2}-x)}{(t_{i+2}-t_i)(t_{i+2}-t_{i+1})} + \frac{(t_{i+3}-x)(x-t_{i+1})}{(t_{i+3}-t_{i+1})(t_{i+2}-t_{i+1})}, & \text{en } [t_{i+1}, t_{i+2}] \\ \frac{(t_{i+3}-x)^2}{(t_{i+3}-t_{i+1})(t_{i+3}-t_{i+2})}, & \text{en } [t_{i+2}, t_{i+3}] \\ 0, & \text{en otra parte} \end{cases}$$

$$\mathbf{5.} \quad \sum_{i=-\infty}^{\infty} f(t_i)B_i^0(x) \quad \mathbf{14.} \quad n-k \leq i \leq m-1$$

$$\mathbf{15.} \quad \text{Use inducción en } k \text{ y } B_{i+i}^{k+i}(x) = 0 \text{ en } [t_i, t_{i+1}]. \quad \mathbf{16.} \quad \text{No} \quad \mathbf{17.} \quad \text{No} \quad \mathbf{19.} \quad \sum_{i=-\infty}^{\infty} t_{i+1}B_i^1(x)$$

$$\mathbf{20.} \quad \text{En la ecuación (9) toma todos los } c_i = 1. \text{ Entonces } d_i = 0. \text{ Por tanto, } \frac{d}{dx} \sum_{i=1}^n B_i^k(x) = 0 \text{ y } \sum_{i=1}^n B_i^k(x) \text{ es constante}$$

**24.** Use la ecuación (14) con todas las  $A$  excepto  $A_j = 1$ . Despues tome todas las  $A$  excepto  $A_{j+1} = 1$

**28.** No **30.** Sea  $C_i^2 = t_{i+1}t_{i+2}$ , entonces  $C_i^1 = xt_{i+1}$  y  $C_i^0 = x^2$ .

**32.**  $B_i^k(t_j) = 0$  si  $t_j \geq t_{i+k+1}$  o  $t_j \leq t_i$  **33.**  $x = (t_{i+3}t_{i+2} - t_i t_{i+1}) / (t_{i+3} + t_{i+2} - t_{i+1} - t_i)$

## Problemas de cómputo 9.3

7. 47040

## Problemas 10.1

**1a.**  $x = \frac{1}{4}t^4 + \frac{7}{3}t^3 - \frac{2}{3}t^{3/2} + c$

**1b.**  $x = ce^t$

**1e.**  $x = c_1 e^t + c_2 e^{-t}$

o

$x = c_1 \cosh t + c_2 \sinh t$

**2a.**  $x = \frac{1}{3}t^3 + \frac{3}{4}t^{4/3} + 7$

**3c.**  $x = \sum_{n=0}^{\infty} (-1)^n \frac{t^{2n+1}}{(2n+1)(2n+1)!} + c$

**3d.**  $x = e^{-t^2/2} \left[ \int t^2 e^{t^2/2} dt + c \right]$

**4.**  $x = a_0 + a_0 \sum_{n=1}^{\infty} (-1)^n \left( \frac{(2n-1)!}{2^{n-1}(2n)!} \right) t^{2n} + \sum_{n=1}^{\infty} (-1)^{n-1} \left( \frac{n! 2^n}{(2n+1)!} \right) t^{2n+1}$

**6.** Sea  $p(t) = a_0 + a_1 t + a_2 t^2 + \dots$  y determine  $a_i$ .

**9.**  $t = 10$ , Error  $= 2.2 \times 10^4 \varepsilon$ ;  $t = 20$ , Error  $= 4.8 \times 10^8 \varepsilon$

**10.**  $x^{(4)} = 18xx'x'' + 6(x')^3 + 3x^2x'''$

**11a.**  $x' = x + e^x$ ;  $x'' = (1 + e^x)x'$ ;  $x''' = (1 + e^x)x'' + e^x(x')^2$ ;  $x^{(4)} = (1 + e^x)x''' + 3e^x x' x'' + e^x (x')^3$ .

**12.**  $x(0.1) = 1.21633$

**14.**  $n \leftarrow 20$

$s \leftarrow x^{(n)}$

**for**  $i = 1$  **to**  $n - 1$  **do**

$s \leftarrow x^{(n-i)} + [h/\text{real}(n+1-i)]s$

**end for**

$s \leftarrow x + h[s]$

## Problemas de cómputo 10.1

**1.**  $x(2.77) = 385.79118$

**2b.**  $x(1.75) = 0.63299\ 9983$

**2c.**  $x(5) = -0.20873\ 51554$

**3.**  $x(10) = 22026.47$

**4a.** Error en  $t = 1$  es  $1.8 \times 10^{-10}$ .

**5.**  $x(0) = 0.03245\ 34427$

**7.**  $x(1) = 1.64872\ 12691$

**9.**  $x(0) = 1.6798409205 \times 10^{-3}$

**10.**  $x(0) = -3.75940\ 73450$

## Problemas 10.2

**2c.**  $f(t, x) = + \sqrt{x / (1 - t^2)}$

**3.**  $x(-0.2) = 1.92$

**5a.** **real function**  $f(t, x)$

**real**  $t, x$

$f \leftarrow t^2 / (1 - t + 2x)$

**end function**  $f$

**8.** Resuelva  $\frac{df}{dx} = e^{-x^2}$ ,  $f(0) = 0$ .

**10.**  $h^3 \left( \frac{1}{6} - \frac{\alpha}{4} \right) D^2 f + \frac{h^3}{6} f_x Df$  donde  $D = \frac{\partial}{\partial t} + f \frac{\partial}{\partial x}$  y  $D^2 = \frac{\partial^2}{\partial t^2} + 2f \frac{\partial^2}{\partial x \partial t} + f^2 \frac{\partial^2}{\partial x^2}$

**11.**  $h = \frac{1}{1024}$

**12.** Vamos a hacer el error de truncamiento local  $\leq 10^{-13}$ . Así,  $100h^5 \leq 10^{-13}$  o  $h \leq 10^{-3}$ . Así, tome  $h = 10^{-3}$  y esperamos que los tres dígitos adicionales sean suficientes para conservar la precisión de 10 dígitos.

- 14b.**  $x^{(4)} = D^3 f + f_x D^2 f + 3Df_x Df + f_x^2 Df$  donde  $D^3 = \frac{\partial^3}{\partial t^3} + 3f \frac{\partial^3}{\partial x \partial t^2} + 3f^2 \frac{\partial^3}{\partial t \partial x^2} + f^3 \frac{\partial^3}{\partial x^3}$
- 15.**  $f(x+th, y+tk) = f(x, y) + t[f_1(x, y)h + f_2(x, y)k] + (1/2)t^2 [f_{11}(x, y)h^2 + 2f_{12}(x, y)hk + f_{22}(x, y)k^2] + \dots$   
Ahora sea  $t = 1$  para obtener la forma usual de la serie de Taylor en dos variables.
- 17.** La serie de Taylor de  $f(x, y) = g(x) + h(y)$  alrededor de  $(a, b)$  es igual a la serie de Taylor de  $g(x)$  alrededor de  $a$  más la de  $h(y)$  alrededor de  $b$ .
- 18.**  $f(1+h, k) \approx -3h + \frac{3}{2}h^2 + k^2$     **19.**  $e^{1-xy} \approx 3 - x - y$     **20.**  $A = 1 + k + \frac{1}{2}k^2, B = h(1+k)$
- 21.**  $A = 1, B = h-k, C = (h-k)^2$
- 22.**  $f(x+h, y+k) \approx (1 + 2xh + k + (1 + 2x^2)h^2 + 2hkh + \frac{1}{2}k^2)f; f(0.001, 0.998) \approx 2.7128534$

## Problemas de cómputo 10.2

- 2.**  $x(1) = 1.5708$     **3b.**  $n = 7; x(2) = 0.8235678972$  (RK),  $0.8235678970$  (TS)
- 3c.**  $n = 7; x(2) = -0.4999999998$  (RK),  $-0.5000000012$  (TS)    **4.**  $x(1) = 0.60653 = x(3)$
- 5.**  $x(3) = 1.5$     **6.**  $x(0) = 1.0 = x(1.6)$     **8.**  $x(1) = 3.95249$     **9.**  $x(10) = 1.344 \times 10^{43}$

## Problemas 10.3

- 1.** Sea  $h = 1/n$ . Entonces  $x(1) = e^{-1}$  (solución verdadera) y  $x_n = \{[1 - 1/(2n)]/[1 + 1/(2n)]\}^n$  solución aproximada.
- 2.**  $x(t+h) = x(t-h) + \frac{h}{3}[f(t-h, x(t-h)) + 4f(t, x(t)) + f(t+h, x(t+h))]$
- 4.**  $a = \frac{24}{13}, b = -\frac{11}{13}, c = \frac{2}{13}, d = \frac{10}{13}, e = -\frac{2}{39}h^2$     **5.**  $a = 1, b = c = \frac{h}{2};$  El término de error es  $\mathcal{O}(h^3)$ .
- 8.**  $\frac{\partial}{\partial s}x(9, s) = e^{252} \approx 10^{109}$     **9a.** Toda  $t$ .    **9c.**  $t$  positiva.    **9e.** Ninguna  $t$ .    **11.** Divergente para toda  $t$ .

## Problemas de cómputo 10.3

- 5.**  $x\left(\frac{1}{2}\right) = 2.25$     **6.**  $x\left(-\frac{1}{2}\right) = -4.5$     **9.**  $y(e) = -6.3890560989$  donde  $y(x) = [1 - \ln v(x)]v(x)$
- 12.**  $0.2193839244$     **13.**  $0.9953087432$     **15.** Si(1) = 0.9460830703

## Problemas 11.1

- 1.**  $x(t+h) = x\left(1 + \frac{1}{2}h^2 + \frac{1}{24}h^4\right) + y\left(h + \frac{1}{6}h^3 + \frac{1}{120}h^5\right), y(t+h) = y\left(1 + \frac{1}{2}h^2 + \frac{1}{24}h^4\right) + x\left(h + \frac{1}{6}h^3 + \frac{1}{120}h^5\right)$
- 2.** Ya que el sistema no está acoplado se resuelven dos problemas separados.
- 3.** El sistema no está acoplado por lo que con el programa se puede resolver cada ecuación diferencial por separado.
- 4.**  $X' = \begin{bmatrix} 1 \\ x_1^2 + \log x_2 + x_0^2 \\ e^{x_2} - \cos x_1 + \operatorname{sen}(x_0 x_1) - (x_1 x_2)^7 \end{bmatrix}, X(0) = [0, 1, 3]^T$

## Problemas de cómputo 11.1

- 1.**  $x(1) = 2.4686939399, y(1) = 1.2873552872$     **2.**  $x(0.38) = 1.90723 \times 10^{12}, y(0.38) = -8.28807 \times 10^4$
- 4.**  $x(-1) = 3.36788, y(-1) = 2.36788$     **5.**  $x_1\left(\frac{\pi}{2}\right) = x_4\left(\frac{\pi}{2}\right) = 0, x_2\left(\frac{\pi}{2}\right) = 1, x_3\left(\frac{\pi}{2}\right) = -1$
- 7.**  $x(6) = 4.39411, y(6) = 3.10378$

## Problemas 11.2

1.  $X' = \begin{bmatrix} x_2 \\ x_3 \\ 2x_2 + \log x_3 + \cos x_1 \end{bmatrix}$        $X(0) = [1, -3, 5]^T$

3. Resuelva cada ecuación aparte, ya que no están acopladas

4.  $X' = \begin{bmatrix} x_2 \\ -x_1 (x_1^2 + x_3^2)^{-3/2} \\ x_4 \\ -x_3 (x_1^2 + x_3^2)^{-3/2} \\ 1 \end{bmatrix}$        $X(0) = \begin{bmatrix} 0.5 \\ 0.75 \\ 0.25 \\ 1.0 \end{bmatrix}$

5.  $X' = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ x_4^2 + \cos(x_2 x_3) - \sin(x_0 x_1) + \log(x_1/x_0) \\ x_4 \\ x_5 \\ x_6 \\ 2x_1 x_3 x_4 + 3x_1^2 x_2 t^2 \\ e^{x_2} x_5 + 4x_1 t^2 x_3 \\ 2t x_6 + 2t e^{x_1 x_3} \end{bmatrix}$        $X(0) = [0, 1, 3, 4, 5]^T$

6.  $X' = \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ 2x_1 x_3 x_4 + 3x_1^2 x_2 t^2 \\ e^{x_2} x_5 + 4x_1 t^2 x_3 \\ 2t x_6 + 2t e^{x_1 x_3} \end{bmatrix}$        $X(1) = \begin{bmatrix} 3 \\ 3 \\ 2 \\ -79/12 \\ 2 \\ 3 \end{bmatrix}$

7a. Sea  $x_1 = x$ ,  $x_2 = x'$ ,  $x_3 = x''$ . Entonces  $X' = \begin{bmatrix} x_2 \\ x_3 \\ -x_3 \sin x_1 - tx_2 - x_3 \end{bmatrix}$

8.  $X' = \begin{bmatrix} x_2 \\ x_2 - x_1 \end{bmatrix}$        $X(0) = [0, 1]^T$

9. Sea  $x_0 = t$ ,  $x_1 = x$ ,  $x_2 = y$ ,  $x_3 = x'$ ,  $x_4 = y'$ . Entonces  $X' = \begin{bmatrix} 1 \\ x_3 \\ x_4 \\ x_1 + x_2 - 2x_3 + 3x_4 + \log x_0 \\ 2x_1 - 3x_2 + 5x_3 + x_0 x_2 - \sin x_0 \end{bmatrix}$   
 $X(0) = [0, 1, 3, 2, 4]^T$

## Problemas 11.3

1.  $x_j(t) = e^{\lambda_j t} x_j(0)$

## Problemas 12.1

1.  $y(x) = 1$

2.  $f(x) = \frac{1}{m+1} \sum_{k=0}^m y_k = (y_0 + \dots + y_m)/(m+1)$ , el promedio de los valores y no implica ninguna  $x_i$ .

3.  $a = (1+2e)/(1+2e^2)$ ,     $b = 1$       5.  $a = 2.1$ ,     $b = 0.9$       7.  $c = [\sum_{k=0}^m y_k \log x_k] / [\sum_{k=0}^m (\log x_k)^2]$

11.  $\varphi$  implica la suma de  $m+1$  polinomios de grado dos en  $c$ , lo cual es cóncava hacia arriba o una constante. Por lo tanto no hay máximos, sólo un mínimo.

12.  $c = 10^{**} [(m+1)^{-1} \sum_{k=0}^m (y_k - \log x_k)]$ .      13.  $y = (6x - 5)/10$

16.  $a \approx 2.5929$ ,     $b \approx -0.32583$ ,     $c \approx 0.022738$

**18.**  $a = 1, b = \frac{1}{3}$     **19.**  $y(x) = \frac{2}{7}x^2 + \frac{29}{35}$     **20.**  $y = x + 1$     **21.**  $c = \left[ \sum_{k=0}^m e^{x_k} f(x_k) \right] \Bigg/ \left[ \sum_{k=0}^m e^{2x_k} \right]$

## Problemas 12.2

- 2.**  $\begin{cases} w_{n+2} = w_{n+1} = 0 \\ w_k = c_k + 3xw_{k+1} + 2w_{k+2} \quad (k = n, n-1, \dots, 0) \\ f(x) = w_0 - (1+2x)w_1 \end{cases}$
- 3.** Puesto que  $\cos(n-2)\theta = \cos[(n-1)\theta - \theta] = \cos(n-1)\theta \cos \theta + \sin(n-1)\theta \sin \theta$ , tenemos  $2 \cos \theta \cos(n-1)\theta - \cos(n-2)\theta = \cos(n-1)\theta \cos \theta - \sin(n-1)\theta \sin \theta = \cos(n\theta)$ . Observe que si  $g_n(\theta) = \cos n\theta$ , entonces  $g_n(\theta) = 2 \cos \theta g_{n-1}(\theta) - g_{n-2}(\theta)$ .
- 5.** Por el problema anterior, la relación recursiva es igual a (2) por lo que  $T_n(x) = f_n(x) = \cos(n \arccos x)$ .
- 6.**  $T_n(T_m(x)) = \cos(n \arccos(\cos(m \arccos x))) = \cos(n m \arccos x) = T_{nm}(x)$ .
- 7.**  $|T_n(x)| = |\cos(n \arccos x)| \leq 1$  para toda  $x \in [-1, 1]$ , puesto que  $|\cos y| \leq 1$  y para que  $\arccos x$  exista  $x$  debe ser  $|x| \leq 1$ .
- 8.**  $\begin{cases} g_0(x) = 1 \\ g_1(x) = (x+1)/2 \\ g_j(x) = (x+1)g_{j-1}(x) - g_{j-2}(x) \quad (j \geq 2) \end{cases}$
- 10.**  $n+2$  multiplicaciones,  $2n+1$  sumas/restas si  $2x$  se calcula como  $x+x$
- 12.**  $n$  multiplicaciones,  $2n$  sumas/restas
- 13.**  $T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$
- 17.**  $\alpha = \frac{y_1 x_2^{13} - y_2 x_1^{13}}{x_1^{12} x_2^{12} (x_2 - x_1)}$   $\alpha$  es muy sensible a las perturbaciones en  $y_1$ .

## Problemas de cómputo 12.2

**7.**  $a_{ij} = \begin{cases} 0 & (i \neq j) \\ (m+1) & (i = j = 1) \\ (m+1)/2 & (i = j > 1) \end{cases}$

## Problemas 12.3

- 2.** La matriz de coeficientes para las ecuaciones normales tiene elementos  $a_{ij} = \frac{1}{i+j-1}$  por (5).
- 3.**  $c = 0$     **4.**  $y = b^x$     **6.**  $c = \ln 2$     **8.**  $x = -1, y = \frac{20}{13}$     **9a.**  $c = \frac{24}{\pi^3}$     **9b.**  $c = 3$     **14.** No.
- 15.**  $y \approx \frac{1}{a+bx}$ . Cambiar a  $\frac{1}{y} \approx a+bx$ .
- 16.**  $\begin{bmatrix} \pi & 0 & 2 \\ 0 & \pi/2 & 0 \\ 2 & 0 & \pi/2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} (1/2)(e^{2\pi} - 1) \\ -(2/5)(e^{2\pi} + 1) \\ (1/5)(e^{2\pi} + 1) \end{bmatrix}$
- 17.**  $c = 3$     **20.**  $c = [\sum_{i=1}^n y_i \sin x_i] / [\sum_{i=1}^n (\sin x_i)^2]$

## Problemas de cómputo 12.3

**1.**  $a = 2, b = 3$

## Problemas 13.1

**1.**  $\ell_0 = 123456; x_1 = .96621\ 2243; x_2 = .12917\ 3003; x_3 = .01065\ 6910$

## Problemas de cómputo 13.1

8. 32.5%      11. La sucesión no es periódica.

|    |    |    |     |    |     |    |     |     |    |
|----|----|----|-----|----|-----|----|-----|-----|----|
| 0  | 1  | 2  | 3   | 4  | 5   | 6  | 7   | 8   | 9  |
| 97 | 93 | 97 | 107 | 90 | 115 | 88 | 101 | 113 | 99 |

13. 15. 5.6%      16. 200

## Problemas 13.2

1.  $m > 4$  millones

## Problemas de cómputo 13.2

2. 1.71828      4. 8      5. 49.9      7. 0.518      9. 1.11      10. 2.00034 6869

14. 0.635      17b. 8.3

## Problemas de cómputo 13.3

1.  $\frac{2}{3}$       2. 0.898      4.  $\frac{7}{16}$       6. 1.05      7. 5.24      9. 0.996      12. 0.6394

14. 11.6 kilómetros      15. 0.14758      17. 0.009      21. 24.2 revoluciones      23. 0.6617

## Problemas 14.1

2.  $c_1 = (1 - 2e)/(1 - e^2)$ ,  $c_2 = (2e - e^2)/(1 - e^2)$       3a.  $x(t) = (e^{\pi+t} - e^{\pi-t})/(e^{2\pi} - 1)$

3b.  $x(t) = (t^4 - 25t + 12)/12$       4a.  $x(t) = \beta \operatorname{sen} t + \alpha \cos t$  para todo  $(\alpha, \beta)$

4b.  $x(t) = c_1 \operatorname{sent} t + \alpha \cos t$  para todo  $\alpha + \beta = 0$  con  $c_1$  arbitrario      6.  $\varphi(t) = z$       7.  $\varphi(z) = z$       8.  $\varphi(z) = \sqrt{9 + 6z}$

9.  $\varphi(z) = (e^5 + e + ze^4 - z)/(2e^2)$       10. Dos caminos: use  $x''(a) = z$  o  $x'(b) = z$ ,  $x''(b) = w$ .

11.  $x(t) = -e^t + 2 \ln(t+1) + 3t$

14a. Éste es un problema lineal. Así, los problemas con dos valores iniciales se pueden resolver como en el libro

para obtener la solución. Los dos conjuntos de valores iniciales serían  $\begin{cases} x(0) = 0 \\ x'(0) = 1 \end{cases}$  y  $\begin{cases} x(0) = 1 \\ x'(0) = 0 \end{cases}$ .

15. La solución de  $x'' = -x$ ,  $x(0) = 1$ ,  $x'(0) = z$  es  $x(t) = \cos t + z \operatorname{sent} t$ . Por tanto,  $\varphi(z) = x(\pi) = -1$ . Puesto que  $\varphi$  es constante, no podemos obtener  $\varphi(z) = 3$  ¡para ninguna elección de  $z$ !

## Problemas 14.2

1.  $-\left(1 - \frac{h}{2}\right)x_{i-1} + 2(1 + h^2)x_i - \left(1 - \frac{h}{2}\right)x_{i+1} = -h^2t$       2.  $x_1 \approx 0.29427$ ,  $x_2 \approx 0.57016$ ,  $x_3 \approx 0.81040$

4.  $x'(0) = \frac{5}{3}$       8.  $-x_{i-1} + [2 + (1 + t_i)h^2]x_i - x_{i+1} = 0$

9.  $x(t) = [7/u(2)]u(t)$

11.  $x_1'' = -x_1$ ,  $x_1(0) = 3$ ,  $x_1'(0) = z_1$  implica  $x = A \cos t + B \operatorname{sent} t$ ,  $3 = x(0) = A$ ,  $x' = -A \operatorname{sent} t + B \cos t$ . Sea  $z_1 = x'(0) = B$ . Así,  $x_1 = 3 \cos t + z_1 \operatorname{sent} t$ ,  $x_2 = 3 \cos t + z_2 \operatorname{sent} t$ . Por la ecuación (10),  $x = \lambda x_1 + (1 - \lambda)x_2$  y  $\lambda = [\beta - x_2(b)]/[x_1(b) - x_2(b)] = [7 - (-3)]/[-3 - (-3)] = 10/6$ .

## Problemas de cómputo 14.2

2a.  $x = 1/(1 + t)$       2b.  $x = -\log(1 + t)$

## Problemas 15.1

1a. Elíptica.    1c. Parabólica.    1f. Hiperbólica.    2.  $\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = 0$

4. La ecuación (3) muestra que  $u(x, t+k)$  es una combinación convexa de valores de  $u(x, t)$  en el intervalo  $[0, 1]$ . Por lo tanto, permanece en el intervalo.

5.  $a = [1 + 2kh^{-2}(\cos \pi h - 1)]^{1/k}$

6. El miembro derecho se cambió por  $b_1 + c_0$  en lugar de  $b_1$  y  $b_{n-1} + c_n$  sustituye a  $b_{n-1}$  tanto en (5) como en (7).

7. En (6),  $b_1$  se sustituye por  $b_1 + g(t)$ ,  $b_{n-1}$  por  $b_{n-1} + g(t)$ . A nivel cero,  $b_i = f(ih)$  para  $1 \leq i \leq n-1$ .

8.  $u(x, t+k) = \frac{k}{h^2}(1-h)u(x+h, t) + \frac{k}{h^2} \left( \frac{h^2}{k} + h - 2 \right) u(x, t) + \frac{k}{h^2}u(x-h, t)$

$$9. A = \begin{bmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -2 & 2 \end{bmatrix}$$

## Problemas 15.2

1.  $-0.21$     2.  $u_{xx} = f''(x+at) + g''(x-at)$ ,  $u_{tt} = a^2 f''(x+at) + a^2 g''(x-at) = a^2 u_{xx}$

3.  $u(x, t) = \frac{1}{2}[F(x+t) - F(-x+t)] + \frac{1}{2}[\overline{G}(x+t) - \overline{G}(-x+t)]$ , donde  $\overline{G}$  es la antiderivada de  $G$

## Problemas de cómputo 15.2

1. real function  $fbar(x)$

real  $x, xbar$

$xbar \leftarrow x + 2 \text{real}(\text{integer}(-(1+x)/2))$

if  $xbar < 0$  then

$fbar \leftarrow -f(-xbar)$

else

$fbar \leftarrow f(xbar)$

end if

end function  $fbar$

## Problemas 15.3

5.  $\left(20 + \frac{2.5h}{x_i + y_j}\right)u_{i+1,j} + \left(20 - \frac{2.5h}{x_i + y_j}\right)u_{i-1,j} + \left(-30 + \frac{0.5h}{y_j}\right)u_{i,j+1} +$

$$\left(-30 + \frac{0.5h}{y_j}\right)u_{i,j-1} + 20u_{ij} = 69h^2$$

6.  $u(0, \frac{1}{2}) \approx -8.932 \times 10^{-3}$ ;     $u(\frac{1}{2}, \frac{1}{2}) \approx 4.643 \times 10^{-1}$     7.  $A = \begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix}$

## Problemas de cómputo 15.3

5.  $18.41^\circ \quad 13.75^\circ$   
 $41.47^\circ \quad 36.60^\circ \quad 24.41^\circ$   
 $69.41^\circ \quad 66.77^\circ \quad 61.05^\circ \quad 53.01^\circ \quad 51.00^\circ$

## Problemas 16.1

1.  $\mathbf{F}(2, 1, -2) = -15; \quad \mathbf{F}(0, 0, -2) = -8; \quad \mathbf{F}(2, 0, -2) = -12 \quad 2. \quad \mathbf{F}\left(\frac{9}{8}, \frac{9}{8}\right) = -20.25$
4. Caso  $n = 2$ :  $\begin{cases} \hat{x} = (3a + b)/4 + \delta & \text{si } a \leq x^* \leq b' \\ \hat{x} = (a + 3b)/4 - \delta & \text{si } a' \leq x^* \leq b \end{cases}$  5a. Solución exacta  $\mathbf{F}(3) = -7$ .
7.  $A = \alpha/\sqrt{5}, \quad A = -\beta/\sqrt{5}$
9. Por (6),  $y + rb = a + r^2(b - a) + rb = ar + b$ , ya que  $r^2 + r = 1$ . Además,  $r(y + rb) = a + r(b - a) = x$ . Por lo tanto,  $yr + r^2b = x$  o  $y + r^2(b - y) = x$ .
10.  $n \geq 1 + (k + \log \ell - \log 2) / |\log r| \quad 11. \quad n \geq 48$
13. El punto mínimo de  $F$  es una raíz de  $F'$ . El método de Newton para encontrar la raíz de  $F'$ :  $x_{n+1} = x_n - \frac{F(x_n)}{F'(x_n)}$ . La fórmula *no* implica a  $F$  misma
14. Para encontrar el mínimo de  $F$ , busque la raíz de  $F'$ . El método de la secante para encontrar la raíz de  $F'$  es  $x_{n+1} = x_n - F'(x_n) \left[ \frac{x_n - x_{n-1}}{F'(x_n) - F'(x_{n-1})} \right]$ . La fórmula *no* implica a  $F$ .
- 15b. Elevando al cuadrado ambos miembros se obtiene  $r^2 = 1 + \sqrt{1 + \sqrt{1 + \dots}} = 1 + r$ .
- 15d.  $1 + r^{-1} + r^{-2} + \dots = (1 - r^{-1})^{-1}$  por la expansión de la serie. Por lo tanto,  $r = (1 - r^{-1})^{-1} - 1 = \frac{1}{r - 1}$  o  $r^2 = r + 1$ .

## Problemas 16.2

- 1a. Sí      1b. No      2.  $(\frac{1}{4}, \frac{9}{4})$       3.  $\mathbf{F}(x, y) = 1 + x - xy + \frac{1}{2}x^2 - \frac{1}{2}y^2 + \dots$
6. La pendiente de la tangente  $\frac{dy}{dx} = -\frac{F_x}{F_y} \equiv m_1$ . El gradiente tiene números de dirección  $F_x$  y  $F_y$  y su pendiente es  $\frac{F_y}{F_x} \equiv m_2$ . La condición de perpendicularidad  $m_1 m_2 = -1$  se encuentra.
- 7b.  $\mathbf{F}(x) = \frac{3}{2} - \frac{1}{2}x_2 + 3x_1x_2 + x_2x_3 + 2x_1^2 - \frac{1}{2}x_3^2 + \dots \quad 9a. \quad G(1, 0) = \begin{bmatrix} -2 \\ 2 \end{bmatrix} \quad 9b. \quad G(1, 2, 1) = \begin{bmatrix} 5 \\ 2 \\ 5 \end{bmatrix}$
10.  $\mathbf{G} = \begin{bmatrix} 2y^2z^2 \operatorname{sen} x \cos x \\ 2yz^2(1 + \operatorname{sen}^2 x) + 2(y + 1)(z + 3)^2 \\ 2y^2z(1 + \operatorname{sen}^2 x) + 2(y + 1)^2(z + 3) \end{bmatrix} \quad 12. \quad \left(-\frac{19}{30}, -\frac{1}{5}\right)$

## Problemas 17.1

2. maximice:  $-5x_1 - 6x_2 + 2x_3$   
 restricciones:  $\begin{cases} -2x_1 + 3x_2 \leq -5 \\ x_1 + x_2 \leq 15 \\ 2x_1 - x_2 + x_3 \leq 25 \\ -x_1 - x_2 + x_3 \leq -1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{cases}$
- 4a. Valor mínimo 1.5 en  $(1.5, 0)$

**5b.** maximice:  $-3x + 2y - 5z$

restricciones:  $\begin{cases} -x - y - z \leq -4 \\ x - y - z \leq 2 \\ -x + y + z \leq -2 \\ x \geq 0, y \geq 0, z \geq 0 \end{cases}$

**6a.** maximice:  $2x_1 + 2x_2 - 6x_3 - x_4$

restricciones:  $\begin{cases} 3x_1 + x_4 = 25 \\ x_1 + x_2 + x_3 + x_4 = 20 \\ -4x_1 - 6x_3 + x_5 = -5 \\ -2x_1 - 3x_3 - 2x_4 + x_6 = 0 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{cases}$

**7.** Máximo de 36 en  $(2, 6)$

**13a.** Máximo de 18 en  $(9, 0)$

**13h.** Máximo de  $\frac{54}{5}$  en  $(\frac{18}{5}, 0)$

**8.** Mínimo de 36 en  $(0, 3, 1)$

**13c.** Solución no acotada

**14.** Máximo de 100 en  $(24, 32, -124)$

**6b.** minimice:  $25y_1 + 20y_2 - 5y_3$

restricciones:  $\begin{cases} 3y_1 + y_2 - 4y_3 - 2y_4 \leq 2 \\ y_2 \leq 2 \\ y_2 - 6y_3 - 3y_4 \leq -6 \\ y_1 + y_2 - 2y_4 \leq -1 \\ y_1, y_2, y_3, y_4 \geq 0 \end{cases}$

**11.** Mínimo 2 para  $(x, x - 2)$  donde  $x \geq 3$

**13f.** No hay solución

**17.** Su conjunto factible está vacío.

## Problemas de cómputo 17.1

|                              | Fieltro | Pajas |
|------------------------------|---------|-------|
| Sombreros Texas              | 0       | 200   |
| Sombreros Estrella solitaria | 150     | 0     |
| Ropa de Rancho Lazo          | 150     | 0     |

**3.** \$13.50

**5.** Cuesta  $50\text{¢}$  por 1.6 onzas de comida  $f_1$ , 1 onza de comida  $f_3$  y nada de comida  $f_2$ .

## Problemas 17.2

**1.** maximice:  $\sum_{j=0}^n c_j y_j$  Aquí  $c_0 = -\sum_{j=1}^n c_j$  y  $a_{i0} = -\sum_{j=1}^n a_{ij}$ .

restricciones:  $\begin{cases} \sum_{j=0}^n a_{ij} y_j \leq b_i \\ y_i \geq 0 \quad (0 \leq i \leq n) \end{cases}$

**2.** A lo más  $2^n$ .

**5.** Primera forma primal: maximice:  $-b^T y$

restricciones:  $\begin{cases} -A^T y \leq -c \\ y \geq 0 \end{cases}$

**6.** Dada  $Ax = b$ . Sea  $y_j = x_j + y_{n+1}$ . Ahora  $\sum_{j=1}^n a_{ij} x_j - b_i = \sum_{j=1}^n a_{ij} y_j - y_{n+1} \sum_{j=1}^n a_{ij} - b_i$ .

minimice:  $y_{n+1}$

restricciones:  $\begin{cases} \sum_{j=1}^n a_{ij} y_j + \left( -\sum_{j=1}^n a_{ij} \right) y_{n+1} = b_i \quad (1 \leq i \leq n+1) \\ y \geq 0 \end{cases}$

## Problemas de cómputo 10.2

**1b.**  $x = [0, 0, \frac{5}{3}, \frac{2}{3}, 0]^T$

**1c.**  $x = [0, \frac{8}{3}, \frac{5}{3}]^T$

## Problemas 17.3

**1a.** maximice:  $-\sum_{i=1}^4(u_i + v_i)$

restricciones: 
$$\begin{cases} 5y_1 + 2y_2 - 7y_4 - u_1 + v_1 = 6 \\ y_1 + y_2 + y_3 - 3y_4 - u_2 + v_2 = 2 \\ 7y_2 - 5y_3 - 2y_4 - u_3 + v_3 = 11 \\ 6y_1 + 9y_3 - 15y_4 - u_4 + v_4 = 9 \\ u \geq 0 \quad v \geq 0 \quad y \geq 0 \end{cases}$$

**1b.** minimice:  $\varepsilon$

restricciones: 
$$\begin{cases} 5y_1 + 2y_2 - 7y_4 - \varepsilon \leq 6 \\ y_1 + y_2 + y_3 - 3y_4 - \varepsilon \leq 2 \\ 7y_2 - 5y_3 - 2y_4 - \varepsilon \leq 11 \\ 6y_1 + 9y_3 - 15y_4 - \varepsilon \leq 9 \\ -5y_1 - 2y_2 + 7y_4 - \varepsilon \leq -6 \\ -y_1 - y_2 - y_3 + 3y_4 - \varepsilon \leq -2 \\ -7y_2 + 5y_3 + 2y_4 - \varepsilon \leq -11 \\ -6y_1 - 9y_3 + 15y_4 - \varepsilon \leq -9 \\ \varepsilon \geq 0 \quad y_j \geq 0 \quad (1 \leq j \leq 4) \end{cases}$$

**3.** Tome  $m$  puntos  $x_i$  ( $i = 1, 2, \dots, m$ ). Sea  $p(x) = \sum_{j=0}^n a_j x^j$ .

minimice:  $\varepsilon$

restricciones: 
$$\begin{cases} \sum_{j=0}^n a_j x_i^j \leq f(x_i) \quad (1 \leq i \leq m) \\ \sum_{j=0}^n a_j x_i^j + \varepsilon \geq f(x_i) \quad (1 \leq i \leq m) \\ \varepsilon \geq 0 \end{cases}$$

**4.** minimice:  $u_1 + v_1 + u_2 + v_2 + u_3 + v_3$

restricciones: 
$$\begin{cases} y_1 - y_2 - u_1 + v_1 = 4 \\ 2y_1 - 3y_2 + y_3 - u_2 + v_2 = 7 \\ y_1 + y_2 - 2y_3 - u_3 + v_3 = 2 \\ y_1, y_2, y_3 \geq 0, u_1, u_2, u_3 \geq 0, v_1, v_2, v_3 \geq 0 \end{cases}$$

## Problemas de cómputo 17.3

**1a.**  $x_1 = 0.353$ ,  $x_2 = 2.118$ ,  $x_3 = 0.765$     **1b.**  $x_1 = 0.671$ ,  $x_2 = 1.768$ ,  $x_3 = 0.453$

**3.**  $p(x) = 1.0001 + 0.9978x + 0.51307x^2 + 0.13592x^3 + 0.071344x^4$

## Problemas B

**1a.**  $e \approx (2.718)_{10} = (010.101\ 101\ 111\ 100\ 111\dots)_2$     **2d.**  $(27.45075\ 341\dots)_8$

**2e.**  $(113.16662\ 13\dots)_8$     **2f.**  $(71.24426\ 416\dots)_8$     **3a.**  $(441.68164\ 0625)_{10}$     **3b.**  $(613.40625)_{10}$

**4c.**  $(101\ 111)_2$     **4e.**  $(110\ 011)_2$     **4g.**  $(33.72664)_8$     **6.**  $(0.3146\ 3146\dots)_8$     **9.**  $(479)_{10} = (111\ 011\ 111)_2$

**12.** Un número real  $R$  tiene una representación finita en el sistema binario.  $\Leftrightarrow R = (a_m a_{m-1} \dots a_1 a_0.b_1 b_2 \dots b_n)_2$ .  $\Leftrightarrow R = (a_m \dots a_1 a_0 b_1 b_2 \dots b_n)_2 \times 2^{-n} = m \times 2^{-n}$  donde  $m = (a_m a_{m-1} \dots a_1 a_0 b_1 b_2 \dots b_n)_2$ .

# Bibliografía

- Abell, M. L., y J. P. Braselton. 1993. *The Mathematical Handbook*. Nueva York: Academic Press.
- Abramowitz, M., e I. A. Stegun (eds.). 1964. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards. Nueva York: Dover, 1965 (reimpreso).
- Acton, F. S. 1959. *Analysis of Straight-Line Data*. Nueva York: Wiley. Nueva York: Dover, 1966 (reimpreso).
- Acton, F. S. 1990. *Numerical Methods That (Usually) Work*. Washington, D.C.: Mathematical Association of America.
- Acton, F. S. 1996. *Real Computing Made Real: Preventing Errors in Scientific and Engineering Calculations*. Princeton, New Jersey: Princeton University Press.
- Ahlberg, J. H., E. N. Nilson, y J. L. Walsh. 1967. *The Theory of Splines and Their Applications*. Nueva York: Academic Press.
- Aiken, R. C. (ed.). 1985. *Stiff Computation*. Nueva York: Oxford University Press.
- Ames, W. F. 1992. *Numerical Methods for Partial Differential Equations*, 3<sup>a</sup> ed. Nueva York: Academic Press.
- Ammar, G. S., D. Calvetti y L. Reichel. 1999. "Computation of Gauss-Kronrod quadrature rules with non-positive weights", *Electronic Transactions on Numerical Analysis* **9**, 26–38. <http://etna.mcs.kent.edu>
- Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney y D. Sorensen. 1999. LAPACK User's Guide, 3<sup>a</sup> ed. Filadelfia: SIAM.
- Armstrong, R. D. y J. Godfrey. 1979. "Two linear programming algorithms for the linear discrete  $\ell_1$  norm problem." *Mathematics of Computation* **33**, 289–300.
- Ascher, U. M. R. M. M. Mattheij y R. D. Russell. 1995. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Filadelfia: SIAM.
- Ascher, U. M. y L. R. Petzold. 1998. *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*. Filadelfia: SIAM.
- Atkinson, K. 1993. *Elementary Numerical Analysis*. Nueva York: Wiley.
- Atkinson, K. A. 1988. *An Introduction to Numerical Analysis*, 2<sup>a</sup> ed. Nueva York: Wiley.
- Axelsson, O. 1994. *Iterative Solution Methods*. Nueva York: Cambridge University Press.
- Axelsson, O. y V.A. Barker. 2001. *Finite Element Solution of Boundary Value Problems: Theory and Computations*. Filadelfia: SIAM.
- Azencott, R. (ed.). 1992. *Simulated Annealing: Parallelization Techniques*. Nueva York: Wiley.
- Bai, Z., J. Demmel, J. Dongarra, A. Ruhe y H. van der Vorst. 2000. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Filadelfia: SIAM.
- Baldick, R. 2006. *Applied Optimization*. Nueva York, Cambridge University Press.
- Barnsley, M. F. 2006. *SuperFractals*. Nueva York, Cambridge University Press.
- Barrett, R., M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine y H. van der Vorst. 1994. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods* Filadelfia: SIAM.
- Barrodale, I. y C. Phillips. 1975. "Solution of an over-determined system of linear equations in the Chebyshev norm." *Association for Computing Machinery Transactions on Mathematical Software* **1**, 264–270.
- Barrodale, I. y F. D. K. Roberts. 1974. "Solution of an over-determined system of equations in the  $\ell_1$  norm." *Communications of the Association for Computing Machinery* **17**, 319–320.
- Barrodale, I., F. D. K. Roberts y B. L. Ehle. 1971. *Elementary Computer Applications*. Nueva York: Wiley.
- Bartels, R. H. 1971. "A stabilization of the simplex method." *Numerische Mathematik* **16**, 414–434.
- Bartels, R., J. Beatty y B. Barsky. 1987. *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. San Francisco: Morgan Kaufmann.
- Bassien, S. 1998. "The dynamics of a family of one-dimensional maps." *American Mathematical Monthly* **105**, 118–130.

- Bayer, D. y P. Diaconis. 1992. "Trailing the dovetail shuffle to its lair." *Annals of Applied Probability*, **2**, 294–313.
- Beale, E. M. L. 1988. *Introduction to Optimization*. Nueva York: Wiley.
- Björck, Å. 1996. *Numerical Methods for Least Squares-Problems*. Filadelfia: SIAM.
- Bloomfield, P. y W. Steiger. 1983. *Least Absolute Deviations, Theory, Applications, and Algorithms*. Boston: Birkhäuser.
- Bornemann, F., D. Laurie, S. Wagon y J. Waldvogel. 2004. *The SIAM 100-Digit Challenge: A Study in High-Accuracy Numerical Computing*. Filadelfia: SIAM.
- Borwein, J. M. y P. B. Borwein. 1984. "The arithmetic-geometric mean and fast computation of elementary functions." *Society for Industrial and Applied Mathematics Review* **26**, 351–366.
- Borwein, J. M. y P. B. Borwein. 1987. *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*. Nueva York: Wiley.
- Boyce, W. E. y R. C. DiPrima. 2003. *Elementary Differential Equations and Boundary Value Problems*, 7<sup>a</sup> ed. Nueva York: Wiley.
- Branham, R. 1990. *Scientific Data Analysis: An Introduction to Overdetermined Systems*. Nueva York: Springer-Verlag.
- Brenner, S. y R. Scott. 2002. *The Mathematical Theory of Finite Element Methods*. Nueva York: Springer-Verlag.
- Brent, R. P. 1976. "Fast multiple precision evaluation of elementary functions." *Journal of the Association for Computing Machinery* **23**, 242–251.
- Briggs, W. 2004. *Ants, Bikes, and Clocks: Problems Solving for Undergraduates*. Filadelfia: SIAM.
- Buchanan, J. L. y P. R. Turner. 1992. *Numerical Methods and Analysis*. Nueva York: McGraw-Hill.
- Burden, R. L. y J. D. Faires. 2001. *Numerical Analysis*, 7<sup>a</sup> ed. Pacific Grove, California: Brooks/Cole.
- Bus, J. C. P. y T. J. Dekker. 1975. "Two efficient algorithms with guaranteed convergence for finding a zero of a function." *Association for Computing Machinery Transactions on Mathematical Software* **1**, 330–345.
- Butcher, J. C. 1987. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*. Nueva York: Wiley.
- Calvetti, D., G. H. Golub, W. B. Gragg y L. Reichel. 2000. "Computation of Gauss-Kronrod quadrature rules." *Mathematics of Computation* **69**, 1035–1052.
- Carrier, G. y C. Pearson. 1991. *Ordinary Differential Equations*. Filadelfia: SIAM.
- Cärtner, B. 2006. *Understanding and Using Linear Programming*. Nueva York: Springer.
- Cash, J. "Mesh selection for nonlinear two-point boundary-value problems." *Journal of Computational Methods in Science and Engineering*, 2003.
- Chaitlin, G. J. 1975. "Randomness and mathematical proof." *Scientific American*, mayo, 47–52.
- Chapman, S. J. 2000. *MATLAB Programming for Engineering*, Pacific Grove, California: Brooks/Cole.
- Cheney, E. W. 1982. *Introduction to Approximation Theory*, 2<sup>a</sup> ed. Washington, D.C.: AMS.
- Cheney, E. W. 2001. *Analysis for Applied Mathematics*, Nueva York: Springer.
- Chicone, C. 2006. *Ordinary Differential Equations with Applications*, 2<sup>a</sup> ed. Nueva York: Springer.
- Clenshaw, C. W. y A. R. Curtis. 1960. "A method for numerical integration on an automatic computer". *Numerische Mathematik* **2**, 197–205.
- Colerman, T. F. y C. Van Loan. 1988. *Handbook for Matrix Computations*. Filadelfia: SIAM.
- Collatz, L. 1966. *The Numerical Treatment of Differential Equations*, 3<sup>a</sup> ed. Berlín: Springer-Verlag.
- Conte, S. D. y C. de Boor. 1980. *Elementary Numerical Analysis*, 3<sup>a</sup> ed. Nueva York: McGraw-Hill.
- Cooper, L. y D. Steinberg. 1974. *Methods and Applications of Linear Programming*. Filadelfia: Saunders.
- Crilly, A. J., R. A. Earnshaw, H. Jones (eds.), 1991. *Fractals and Chaos*. Nueva York: Springer-Verlag.
- Cvijovic, D. y J. Klinowski. 1995. "Taboo search: An approach to the multiple minima problem." *Science* **267**, 664–666.
- Dahlquist, G. y A. Björck. 1974. *Numerical Methods*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Dantzi, G. B., A. Orden y P. Wolfe. 1963. "Generalized simplex method for minimizing a linear from under linear inequality constraints." *Pacific Journal of Mathematics* **5**, 183–195.
- Davis, P. J. y P. Rabinowitz. 1984. *Methods of Numerical Integration*, 2<sup>a</sup> ed. Nueva York: Academic Press.
- Davis, T. 2006. *Direct Methods for Sparse Linear Systems*. Filadelfia: SIAM.
- de Boor, C. 1971. "CADRE: An algorithm for numerical quadrature." En *Mathematical Software*, editado por J.R. Rice, 417–49. Nueva York: Academic Press.
- de Boor, C. 1984. *A Practical Guide to Splines*, 2<sup>a</sup> ed. Nueva York: Springer-Verlag.
- Dekker, T. J. 1969. "Finding a zero by means of successive linear interpolation." En *Constructive Aspects of the Fundamental Theorem of Algebra*, editado por B. Dejon y R Henrici. Nueva York: Wiley-Interscience.
- Dekker, T. J. y W. Hoffmann. 1989. "Rehabilitation of the Gauss-Jordan algorithm." *Numerische Mathematik* **54**, 591–599.

- Dekker, T. J., W. Hoffmann y K. Potma. 1997. "Stability of the Gauss-Huard algorithm with partial pivoting." *Computing* **58**, 225–244.
- Dekker, K. y J. G. Verwer. 1984. "Stability of Runge-Kutta methods for stiff nonlinear differential equations." *CWI Monographs* **2**. Amsterdam: Elsevier Science.
- Demmel, J. W., 1997. *Applied Numerical Linear Algebra*. Filadelfia: SIAM.
- Dennis, J. E. y R. Schnabel. 1983. *Quasi-Newton Methods for Nonlinear Problems*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Dennis, J. E. y R. B. Schnabel. 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Filadelfia: SIAM.
- Dennis, J. E. y D. J. Woods. 1987. "Optimization on microcomputers: The Nelder-Mead simplex algorithm." En *New Computing Environments*, editado por A. Wouk. Filadelfia: SIAM.
- de Temple, D. W. 1993. "A quicker convergence to Euler's Constant." *American Mathematical Monthly* **100**, 468–470.
- Devitt, J. S. 1993. *Calculus with Maple V*. Pacific Grove, California: Brooks/Cole.
- Dixon, V. A. 1974. "Numerical quadrature: a survey of the available algorithms." En *Software for Numerical Mathematics*, editado por D. J. Evans. Nueva York: Academic Press.
- Dongarra, J. J., I. S. Duff, D. C. Sorenson y H. van der Vorst. 1990. *Solving Linear Systems on Vector and Shared Memory Computers*. Filadelfia: SIAM.
- Dorn, W. S. y D. D. McCracken. 1972. *Numerical Methods with FORTRAN IV Case Studies*. Nueva York: Wiley.
- Edwards, C. y D. Penny. 2004. *Differential Equations and Boundary Value Problems*, 5<sup>a</sup> ed. Upper Saddle River: New Jersey: Prentice-Hall.
- Ellis, W., Jr., E. W. Johnson, E. Lodi y D. Schwalbe. 1997. *Maple V Flight Manual: Tutorials for Calculus, Linear Algebra, and Differential Equations*. Pacific Grove, California: Brooks/Cole.
- Ellis, W., Jr. y E. Lodi. 1991. *A Tutorial Introduction to Mathematica*. Pacific Grove, California: Brooks/Cole.
- Elman, H., D. J. Silvester y A. Wathen. 2004. *Finite Element and Fast Iterative Solvers*. Nueva York: Oxford University Press.
- England, R. 1969. "Error estimates for Runge-Kutta type solutions of ordinary differential equations." *Computer Journal* **12**, 166–170.
- Enright, W. H. 2006. "Verifying approximate solutions to differential equations." *Journal of Computational and Applied Mathematics* **185**, 203–311.
- Epureanu, B. I. y H. S. Greenside. 1998. "Fractal basins of attraction associated with a damped Newton's method." *SIAM Review* **40**, 102–109.
- Evans, G., J. Blackledge y P. Yardley. 2000. *Numerical Methods for Partial Differential Equations*. Nueva York: Springer-Verlag.
- Evans, G. W., G. F. Wallace y G. L. Sutherland. 1967. *Simulation Using Digital Computers*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Farin, G. 1990. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 2<sup>a</sup> ed. Nueva York: Academic Press.
- Fauvel, J., R. Flood, M. Shortland y R. Wilson (eds.). 1988. *Let Newton Be!* Londres: Oxford University Press.
- Feder, J. 1988. *Fractals*. Nueva York: Plenum Press.
- Fehlberg, E. 1969. "Klassische Runge-Kutta formeln fünfter und siebenter ordnung mit schrittweitenkontrolle." *Computing* **4**, 93–106.
- Flehinger, B. J. 1966. "On the probability that a random integer has initial digit A." *American Mathematical Monthly* **73**, 1056–1061.
- Fletcher, R. 1976. *Practical Methods of Optimization*. Nueva York: Wiley.
- Floudas, C. A. y P. M. Pardalos (eds.). 1992. *Recent Advances in Global Optimization*. Princeton, New Jersey: Princeton University Press.
- Flowers, B. H. 1995. *An Introduction to Numerical Methods in C++*. Nueva York: Oxford University Press.
- Ford, J. A. 1995. "Improved Algorithms of Illinois-Type for the Numerical Solution of Nonlinear Equations." Reporte técnico, Departamento de Ciencias Computacionales, University of Essex, Colchester, Essex, UK.
- Forsythe, G. E. 1957. "Generation and use of orthogonal polynomials for data-fitting with a digital computer." *Society for Industrial and Applied Mathematics Journal* **5**, 74–88.
- Forsythe, G. E. 1970. "Pitfalls in computation, or why a math book isn't enough," *American Mathematical Monthly* **77**, 931–956.
- Forsythe, G. E., M. A. Malcolm y C. B. Moler. 1977. *Computer Methods for Mathematical Computations*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Forsythe, G. E. y C. B. Moler. 1967. *Computer Solution of Linear Algebraic Systems*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Forsythe, G. E. y W. R. Wasow. 1960. *Finite Difference Methods for Partial Differential Equations*. Nueva York: Wiley.
- Fox, L. 1957. *The Numerical Solution of Two-Point Boundary Problems in Ordinary Differential Equations*. Oxford: Clarendon Press.

- Fox, L. 1964. *An Introduction to Numerical Linear Algebra, Monograph on Numerical Analysis*. Oxford: Clarendon Press. Reimpreso 1974. Nueva York: Oxford University Press.
- Fox, L., D. Juskey y J. H. Wilkinson, 1948. "Notes on the solution of algebraic linear simultaneous equations," *Quarterly Journal of Mechanics and Applied Mathematics*, 1, 149–173.
- Frank, W. 1958. "Computing eigenvalues of complex matrices by determinant evaluation and by methods of Danilewski and Wielandt." *Journal of SIAM* 6, 37–49.
- Fraser, W. y M. W. Wilson. 1966. "Remarks on the Clenshaw-Curtis quadrature scheme." *SIAM Review* 8, 322–327.
- Friedman, A. y N. Littman. 1994. *Industrial Mathematics: A Course in Solving Real-World Problems*. Filadelfia: SIAM.
- Fröberg, C.-E. 1969. *Introduction to Numerical Analysis*. Reading, Massachusetts: Addison-Wesley.
- Gallivan, K. A., M. Heath, E. Ng, B. Peyton, R. Plemmons, J. Ortega, C. Romine, A. Sameh y R. Voigt. 1990. *Parallel Algorithms for Matrix Computations*. Filadelfia: SIAM.
- Gander, W. y W. Gautschi. 2000. "Adaptive quadrature—revisited." *BIT* 40, 84–101.
- Garvan, F. 2002. *The Maple Book*. Boca Raton, Florida: Chapman & Hall/CRC.
- Gautschi, W. 1990. "How (un)stable are Vandermonde systems?", en *Asymptotic and Computational Analysis*, 193–210, Lecture Notes in Pure and Applied Mathematics, 124. Nueva York: Dekker.
- Gautschi, W. 1997. *Numerical Analysis: An Introduction*. Boston, Massachusetts: Birkhäuser.
- Gear, C. W. 1971. *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Gentle, J. E. 2003. *Random Number Generation and Monte Carlo Methods*, 2<sup>a</sup> ed. Nueva York: Springer-Verlag.
- Gentleman, W. M. 1972. "Implementing Clenshaw-Curtis quadrature." *Communications of the ACM* 15, 337–346, 353.
- Gerald, C. F. y P. O. Wheatley 1999. *Applied Numerical Analysis*, 6<sup>a</sup> ed. Reading, Massachusetts: Addison-Wesley.
- Ghizetti, A. y A. Ossicini. 1970. *Quadrature Formulae*. Nueva York: Academic Press.
- Gill, P. E., W. Murray y M. H. Wright. 1981. *Practical Optimization*. Nueva York: Academic Press.
- Gleick, J. 1992. *Genius: The Life and Science of Richard Feynman*. Nueva York: Pantheon.
- Gockenbach, M. S., 2002. *Partial Differential Equations: Analytical and Numerical Methods*. Filadelfia: SIAM.
- Goldberg, D. 1991. "What every computer scientist should know about floating-point arithmetic." *ACM Computing Surveys* 23, 5–48.
- Goldstine, H. H. 1977. *A History of Numerical Analysis from the 16th to the 19th Century*. Nueva York: Springer-Verlag.
- Golub, G. H. y J. M. Ortega. 1992. *Scientific Computing and Differential Equations*. Nueva York: Harcourt Brace Jovanovich.
- Golub, G. H. y J. M. Ortega. 1993. *An Introduction with Parallel Scientific Computing*. Nueva York: Academic Press.
- Golub, G. H. y C. F. Van Loan. 1996. *Matrix Computations*, 3<sup>a</sup> ed. Baltimore: Johns Hopkins University Press.
- Good, I. J. 1972. "What is the most amazing approximate integer in the universe?" *Pi Mu Epsilon Journal* 5, 314–315.
- Greenbaum, A. 1997. *Iterative Methods for Solving Linear Systems*. Filadelfia: SIAM.
- Greenbaum, A. 2002. "Card Shuffling and the Polynomial Numerical Hull of Degree  $k$ ," Departamento de Matemáticas, Washington University Seattle, Washington.
- Gregory, R. T. y D. Karney, 1969. *A Collection of Matrices for Testing Computational Algorithms*. Nueva York: Wiley.
- Griewank, A. 2000. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Filadelfia: SIAM.
- Groetsch, C. W. 1998. "Lanczos' generalized derivative." *American Mathematical Monthly* 105, 320–326.
- Haberman, R. 2004. *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*. Upper Saddle River: New Jersey: Prentice-Hall.
- Hageman, L. A. y D. M. Young. 1981. *Applied Iterative Methods*. Nueva York: Academic Press; Dover 2004 (reimpreso).
- Hämmerlin, G. y K.-H. Hoffmann. 1991. *Numerical Mathematics*. Nueva York: Springer-Verlag.
- Hammersley, J. M. y D. C. Handscomb. 1964. *Monte Carlo Methods*. Londres: Methuen.
- Hansen, T., G. L. Mullen y H. Niederreiter. 1993. "Good parameters for a class of node sets in quasi-Monte Carlo integration." *Mathematics of Computation* 61, 225–234.
- Haruki, H. y S. Haruki. 1983. "Euler's Integrals." *American Mathematical Monthly* 7, 465.
- Hastings, H. M. y G. Sugihara. 1993. *Fractals: A User's Guide for the Natural Sciences*. Nueva York: Oxford University Press.

- Havie, T. 1969. "On a modification of the Clenshaw-Curtis quadrature formula." *BIT* **9**, 338–350.
- Heath, J. M. 2002. *Scientific Computing: An Introductory Survey*, 2<sup>a</sup> ed. Nueva York: McGraw-Hill.
- Henrici, P. 1962. *Discrete Variable Methods in Ordinary Differential Equations*. Nueva York: Wiley.
- Heroux, M., P. Raghavan y H. Simon. 2006. *Parallel Processing for Scientific Computing*. Filadelfia: SIAM.
- Herz-Fischler, 1998. R. *A Mathematical History of the Golden Number*. Nueva York: Dover
- Hestenes, M. R. y E. Stiefel. 1952. "Methods of conjugate gradient for solving linear systems." *Journal Research National Bureau of Standards* **49**, 409–436.
- Higham, D. y N. J. Higham. 2006. *MATLAB Guide*, 2<sup>a</sup> ed. Filadelfia: SIAM.
- Higham, N. J. 2002. *Accuracy and Stability of Numerical Algorithms*, 2<sup>a</sup> ed. Filadelfia: SIAM.
- Hildebrand, F. B. 1974. *Introduction to Numerical Analysis*. Nueva York: McGraw-Hill.
- Hodges, A. 1983. *Alan Turing: The Enigma*. Nueva York: Simon & Schuster.
- Hoffmann, W. 1989. "A fast variant of the Gauss-Jordan algorithm with partial pivoting. Basic transformations in linear algebra for vector computing." Disertación doctoral, Universidad de Amsterdam, Países Bajos.
- Hofmann-Wellenhof, B., H. Lichtenegger y J. Collins. 2001. *Global Positioning System: Theory and Practice*, 5<sup>a</sup> ed. Nueva York: Springer-Verlag.
- Horst, R., P. M. Pardalos y N. V. Thoai. 2000. *Introduction to Global Optimization*, 2<sup>a</sup> ed. Boston: Kluwer.
- Householder, A. S. 1970. *The Numerical Treatment of a Single Nonlinear Equation*. Nueva York: McGraw-Hill.
- Huard, P. 1979. "La méthode du simplexe sans inverse explicite." *Bull. E.D.F. Série C* **2**.
- Huddleston, J. V. 2000. *Extensibility and Compressibility in One-Dimensional Structures*. 2<sup>a</sup> ed. Buffalo, NY: ECS Publ.
- Hull, T. E. y A. R. Dobell. 1962. "Random number generators." *Society for Industrial and Applied Mathematics Review* **4**, 230–254.
- Hull, T. E., W. H. Enright, B. M. Fellen y A. E. Sedgwick. 1972. "Comparing numerical methods for ordinary differential equations." *Society for Industrial and Applied Mathematics Journal on Numerical Analysis* **9**, 603–637.
- Hundsdorfer, W. H. 1985. "The numerical solution of nonlinear stiff initial value problems: an analysis of one step methods." CWI Tract, 12. Amsterdam: Stichting Mathematisch Centrum, Centrum voor Wiskunde en Informatica.
- Isaacson, E. y H. B. Keller. 1966. *Analysis of Numerical Methods*. Nueva York: Wiley.
- Jeffrey, A. 2000. *Handbook of Mathematical Formulas and Integrals*. Boston: Academic Press,
- Jennings, A. 1977. *Matrix Computation for Engineers and Scientists*. Nueva York: Wiley.
- Johnson, L. W., R. D. Riess y J. T. Arnold. 1997. *Introduction to Linear Algebra*. Nueva York: Addison-Wesley.
- Kahaner, D. K. 1971. "Comparison of numerical quadrature formulas." En *Mathematical Software*, editado por J. R. Rice. Nueva York: Academic Press.
- Kahaner, D., C. Moler y S. Nash. 1989. *Numerical Methods and Software*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Keller, H. B. 1968. *Numerical Methods for Two-Point Boundary-Value Problems*. Toronto: Blaisdell.
- Keller, H. B. 1976. *Numerical Solution of Two-Point Boundary Value Problems*. Filadelfia: SIAM.
- Kelley, C. T. 1995. *Iterative Methods for Linear and Non-linear Equations*. Filadelfia: SIAM.
- Kelley, C. T. 2003. *Solving Nonlinear Equations with Newton's Method*. Filadelfia: SIAM.
- Kincaid, D. y W. Cheney. 2002. *Numerical Analysis: Mathematics of Scientific Computing*, 3<sup>a</sup> ed. Belmont, California: Thomson Brooks/Cole.
- Kincaid, D. R. y D. M. Young. 1979. "Survey of iterative methods." En *Encyclopedia of Computer Science and Technology*, editado por J. Belzer, A. G. Holzman y A. Kent. Nueva York: Dekker.
- Kincaid, D. R. y D. M. Young. 2000. "Partial differential equations." En *Encyclopedia of Computer Science*, 4<sup>a</sup> ed., editado por A. Ralston, E. D. Reilly, D. Hemmendinger. Nueva York: Grove's Dictionaries.
- Kinderman, A. J. y J. F. Monahan. 1977. "Computer generation of random variables using the ratio of uniform deviates." *Association of Computing Machinery Transactions on Mathematical Software* **3**, 257–260.
- Kirkpatrick, S., C. D. Gelatt, Jr. y M. P. Vecchi. 1983. "Optimization by simulated annealing." *Science* **220**, 671–680.
- Knight, A. 2000. *Basics of MATLAB and Beyond*. Boca Raton, Florida: CRC Press.
- Knuth, D. E. 1997. *The Art of Computer Programming*, 3<sup>a</sup> ed. Vol. 2, *Seminumerical Algorithms*. Nueva York: Addison-Wesley.
- Krogh, F. T. 2003. "On developing mathematical software." *Journal of Computational and Applied Mathematics* **185**, 196–202.

- Kronrod, A. S. 1964. "Nodes and Weights of Quadrature Rules." *Doklady Akad. Nauk SSSR*, **154**, 283–286. [Ruso] (1965). Nueva York: Consultants Bureau.
- Krylov, V. I. 1962. *Approximate Calculation of Integrals*, traducido por A. Stroud. Nueva York: Macmillan.
- Lambert, J. D. 1973. *Computational Methods in Ordinary Differential Equations*. Nueva York: Wiley.
- Lambert, J. D. 1991. *Numerical Methods for Ordinary Differential Equations*. Nueva York: Wiley.
- Lapidus, L. y J. H. Seinfeld. 1971. *Numerical Solution of Ordinary Differential Equations*. Nueva York: Academic Press.
- Laurie, D. P. 1997. "Calculation of Gauss-Kronrod quadrature formulae." *Mathematics of Computation*, 1133–1145.
- Lawson, C. L. y R. J. Hanson. 1995. *Solving Least-Squares Problems*. Filadelfia: SIAM.
- Leva, J. L. 1992. "A fast normal random number generator." *Association of Computing Machinery Transactions on Mathematical Software* **18**, 449–455.
- Lindfield, G. y J. Penny. 2000. *Numerical Methods Using MATLAB*, 2<sup>a</sup> ed. Upper Saddle River: New Jersey: Prentice-Hall.
- Lootsam, F. A. (ed.). 1972. *Numerical Methods for Non-linear Optimization*. Nueva York: Academic Press.
- Lozier, D. W. y F. W. J. Olver. 1994. "Numerical evaluation of special functions." En *Mathematics of Computation 1943–1993: A Half-Century of Computational Mathematics* **48**, 79–125. Providence, Rhode Island: AMS.
- Lynch, S. 2004. *Dynamical Systems with Applications*. Boston: Birkhäuser.
- MacLeod, M. A. 1973. "Improved computation of cubic natural splines with equi-spaced knots." *Mathematics of Computation* **27**, 107–109.
- Maron, M. J. 1991. *Numerical Analysis: A Practical Approach*. Boston: PWS Publishers.
- Marsaglia, G. 1968. "Random numbers fall mainly in the planes." *Proceedings of the National Academy of Sciences* **61**, 25–28.
- Marsaglia, G. y W. W. Tsang. 2000. "The Ziggurat Method for generating random variables." *Journal of Statistical Software* **5**, 1–7.
- Mattheij, R. M. M., S. W. Rienstra y J. H. M. ten Thije Boonkamp. 2005. *Partial Differential Equations: Modeling, Analysis, Computation*. Filadelfia: SIAM.
- McCartin, B. J. 1998. "Seven deadly sins of numerical computations," *American Mathematical Monthly* **105**, No. 10, 929–941.
- McKenna, P. J. y C. Tuama. 2001. "Large torsional oscillations in suspension bridges visited again: Vertical forcing creates torsional response." *American Mathematical Monthly* **108**, 738–745.
- Mehrotra, S. 1992. "On the implementation of a primal-dual interior point method." *SIAM Journal on Optimization* **2**, 575–601.
- Metropolis, N. y cols. 1953. "Equation of state calculations by fast computing machines." *Journal of Physical Chemistry* **21**, 1087–1092.
- Meurant, G. 2006. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*. Filadelfia: SIAM.
- Meyer, C. D., 2000. *Matrix Analysis and Applied Linear Algebra*. Filadelfia: SIAM.
- Miranker, W. L. 1981. "Numerical methods for stiff equations and singular perturbation problems." En *Mathematics and its Applications*, Vol. 5. Dordrecht-Boston, Massachusetts: D. Reidel.
- Moler, C. B., 2004. *Numerical Computing with MATLAB*. Filadelfia: SIAM.
- Moré, J. J. y S. J. Wright. 1993. *Optimization Software Guide*. Filadelfia: SIAM.
- Moulton, F. R. 1930. *Differential Equations*. Nueva York: Macmillan.
- Nelder, J. A. y R. Mead. 1965. "A simplex method for function minimization." *Computer Journal* **7**, 308–313.
- Nerinckx, D. y A. Haegemans. 1976. "A comparison of nonlinear equation solvers." *Journal of Computational and Applied Mathematics* **2**, 145–148.
- Nering, E. D. y A. W. Tucker. 1992. *Linear Programs and Related Problems*. Nueva York: Academic Press.
- Niederreiter, H. 1978. "Quasi-Monte Carlo methods." *Bulletin of the American Mathematical Society* **84**, 957–1041.
- Niederreiter, H. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*. Filadelfia: SIAM.
- Nievergelt, J., J. G. Farrar y E. M. Reingold. 1974. *Computer Approaches to Mathematical Problems*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Noble, B. y J. W. Daniel. 1988. *Applied Linear Algebra*, 3<sup>a</sup> ed. Englewood Cliffs, New Jersey: Prentice-Hall.
- Nocedal, J. y S. Wright. 2006. *Numerical Optimization*, 2<sup>a</sup> ed. Nueva York: Springer.
- Novak, E., K. Ritter y H. Wozniakowski. 1995. "Average-case optimality of a hybrid secant-bisection method." *Mathematics of Computation* **64**, 1517–1540.
- Novak, M. (ed.). 1998. *Fractals and Beyond: Complexities in the Sciences*. River Edge, NJ: World Scientific.

- O'Hara, H. y E J. Smith. 1968. "Error estimation in Clenshaw-Curtis quadrature formula." *Computer Journal* **11**, 213–219.
- Oliveira, S. y D. E. Stewart. 2006. *Writing Scientific Software: A Guide to Good Style*. Nueva York: Cambridge University Press.
- Orchard-Hays, W. 1968. *Advanced Linear Programming Computing Techniques*. Nueva York: McGraw-Hill.
- Ortega, J. y R. G. Voigt. 1985. *Solution of Partial Differential Equations on Vector and Parallel Computers*. Filadelfia: SIAM.
- Ortega, J. M. 1990a. *Numerical Analysis: A Second Course*. Filadelfia: SIAM.
- Ortega, J. M. 1990b. *Introduction to Parallel and Vector Solution of Linear Systems*. Nueva York: Plenum.
- Ortega, J. M. y W. C. Rheinboldt. 1970. *Iterative Solution of Nonlinear Equations in Several Variables*. Nueva York: Academic Press. (2000. Reimpreso. Filadelfia: SIAM.)
- Ostrowski, A. M. 1966. *Solution of Equations and Systems of Equations*, 2<sup>a</sup> ed. Nueva York: Academic Press.
- Overton, M. L. 2001. *Numerical Computing with IEEE Floating Point Arithmetic*. Filadelfia: SIAM.
- Otten, R. H. J. M. y L. P. P. van Ginneken. 1989. *The Annealing Algorithm*. Dordrecht, Alemania: Kluwer.
- Pacheco, P. 1997. *Parallel Programming with MPI*. San Francisco: Morgan Kaufmann.
- Patterson, T. N. L. 1968. "The optimum addition of points to quadrature formulae." *Mathematics of Computations* **22**, 847–856, y en 1969 *Mathematics of Computations* **23**, 892.
- Parlett, B. N. 1997. *The Symmetric Eigenvalue Problem*. Filadelfia: SIAM.
- Parlett, B. 2000. "The QR Algorithm," *Computing in Science and Engineering* **2**, 38–42.
- Pessens, R., E. de Doncker, C. W. Uberhuber y D. K. Kahaner. 1983. *QUADPACK: A Subroutine Package for Automatic Integration*. Nueva York: Springer-Verlag.
- Peterson, I. 1997. *The Jungles of Randomness: A Mathematical Safari*. Nueva York: Wiley.
- Phillips, G. M. y P. J. Taylor. 1973. *Theory and Applications of Numerical Analysis*. Nueva York: Academic Press.
- Press, W. H., S. A. Teukolsky, W. T. Vetterling y B. P. Flannery. 2002. *Numerical Recipes in C++, 2<sup>a</sup> ed*. Nueva York: Cambridge University Press.
- Quinn, M. J. 1994. *Parallel Computing: Theory and Practice*. Nueva York: McGraw-Hill.
- Rabinowitz, P. 1968. "Applications of linear programming to numerical analysis." *Society for Industrial and Applied Mathematics Review* **10**, 121–159.
- Rabinowitz, P. 1970. *Numerical Methods for Nonlinear Algebraic Equations*. Londres: Gordon & Breach.
- Raimi, R. A. 1969. "On the distribution of first significant figures." *American Mathematical Monthly* **76**, 342–347.
- Ralston, A. 1965. *A First Course in Numerical Analysis*. Nueva York: McGraw-Hill.
- Ralston, A. y C. L. Meek (eds.) 1976. *Encyclopedia of Computer Science*. Nueva York: Petrocelli/Charter.
- Ralston, A. y P. Rabinowitz. 2001. *A First Course in Numerical Analysis*, 2<sup>a</sup> ed. Nueva York: Dover.
- Recktenwald, G. 2000. *Numerical Methods with MATLAB: Implementation and Applications*. Nueva York: Prentice-Hall.
- Reid, J. 1971. "On the method of conjugate gradient for the solution of large sparse systems of linear equations." En *Large Sparse Sets of Linear Equations*, J. Reid (ed.), Londres: Academic Press.
- Rheinboldt, 1998. *Methods for Solving Systems of Nonlinear Equations*, 2<sup>a</sup> ed. Filadelfia: SIAM.
- Rice, J. R. 1971. "SQUARES: An algorithm for least squares approximation." En *Mathematical Software*, editado por J. R. Rice. Nueva York: Academic Press.
- Rice, J. R. 1983. *Numerical Methods, Software, and Analysis*. Nueva York: McGraw-Hill.
- Rice, J. R. y R. F. Boisvert. 1984. *Solving Elliptic Problems Using ELLPACK*. Nueva York: Springer-Verlag.
- Rice, J. R., y J. S. White. 1964. "Norms for smoothing and estimation." *Society for Industrial and Applied Mathematics Review* **6**, 243–256.
- Rivlin, T. J. 1990. *The Chebyshev Polynomials*, 2<sup>a</sup> ed. Nueva York: Wiley.
- Roger, H.-F. 1998. *A Mathematical History of the Golden Number*. Nueva York: Dover.
- Roos, C, T. Terlaky y J.-Ph. Vial. 1997. *Theory and Algorithms for Linear Optimization: An Interior Point Approach*. Nueva York: Wiley.
- Saad, Y., 2003. *Iterative Methods for Sparse Linear Systems*. Filadelfia: SIAM.
- Salamin, E. 1976. "Computation of  $\pi$  using arithmetic-geometric mean." *Mathematics of Computation* **30**, 565–570.
- Sauer, T. 2006. *Numerical Analysis*. Nueva York: Pearson, Addison-Wesley.
- Scheid, F. 1968. *Theory and Problems of Numerical Analysis*. Nueva York: McGraw-Hill.

- Scheid, F. 1990. 2000 *Solved Problems in Numerical Analysis*. Schaum's Solved Problem Series. Nueva York: McGraw-Hill.
- Schilling, R. J. y S. L. Harris. 2000. *Applied Numerical Methods for Engineering Using MATLAB* y C. Pacific Grove, California: Brooks/Cole.
- Schmidt 1908. Título desconocido. *Rendiconti del Circolo Matemático di Palermo* **25**, 53–77.
- Schoenberg, I. J. 1946. “Contributions to the problem of approximation of equidistant data by analytic functions.” *Quarterly of Applied Mathematics* **4**, 45–99, 112–141.
- Schoenberg, I. J. 1967. “On spline functions.” En *Inequalities*, editado por O. Shisha, 255–291. Nueva York: Academic Press.
- Schrage, L. 1979. “A more portable Fortran random number generator.” *Association for Computing Machinery Transactions on Mathematical Software* **5**, 132–138.
- Schrijver, A. 1986. *Theory of Linear and Integer Programming*. Somerset, New Jersey: Wiley.
- Schultz, M. H. 1973. *Spline Analysis*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Schumaker, L. L. 1981. *Spine Function: Basic Theory*. Nueva York: Wiley.
- Shampine, J. D. 1994. *Numerical Solutions of Ordinary Differential Equations*. Londres: Chapman and Hall.
- Shampine, L. F., R. C. Allen y S. Pruess. 1997. *Fundamentals of Numerical Computing*. Nueva York: Wiley.
- Shampine, L. F. y M. K. Gordon. 1975. *Computer Solution of Ordinary Differential Equations*. San Francisco: W. H. Freeman.
- Shewchuk, J. R. 1994. “An introduction to the conjugate gradient method without the agonizing pain”, Wikipedia.
- Skeel, R. D. y J. B. Keiper. 1992. *Elementary Numerical Computing with Mathematica*. Nueva York: McGraw-Hill.
- Smith, G. D. 1965. *Solution of Partial Differential Equations*. Nueva York: Oxford University Press.
- Sobol, I. M. 1994. *A Primer for the Monte Carlo Method*. Boca Raton, Florida: CRC Press.
- Southwell, R. V. 1946. *Relaxation Methods in Theoretical Physics*. Oxford: Clarendon Press.
- Späth, H. 1992. *Mathematical Algorithms for Linear Regression*. Nueva York: Academic Press.
- Stakgold, I., 2000. *Boundary Value Problems of Mathematical Physics*. Filadelfia: SIAM.
- Steele, J. M., 1997. *Random Number Generation and Quasi-Monte Carlo Methods*. Filadelfia: SIAM.
- Stetter, H. J. 1973. *Analysis of Discretization Methods for Ordinary Differential Equations*. Berlín: Springer-Verlag.
- Stewart, G. W. 1973. *Introduction to Matrix Computations*. Nueva York: Academic Press.
- Stewart, G. W. 1996. *Afternotes on Numerical Analysis*. Filadelfia: SIAM.
- Stewart, G. W. 1998a. *Afternotes on Numerical Analysis: Afternotes Goes to Graduate School*. Filadelfia: SIAM.
- Stewart, G. W. 1998b. *Matrix Algorithms: Basic Decompositions*, vol. 1. Filadelfia: SIAM.
- Stewart, G. W. 2001. *Matrix Algorithms: Eigensystems*, vol. 2. Filadelfia: SIAM.
- Stoer, J. y R. Bulirsch. 1993. *Introduction to Numerical Analysis*, 2<sup>a</sup> ed. Nueva York: Springer-Verlag.
- Strang, G. 2006. *Linear Algebra and Its Applications*. Belmont, California: Thomson Brooks/Cole.
- Strang, G. y K. Borre. 1997. *Linear Algebra, Geodesy, and GPS*. Cambridge, MA: Wellesley Cambridge Press.
- Street, R. L. 1973. *The Analysis and Solution of Partial Differential Equations*. Pacific Grove, California: Brooks/Cole.
- Stroud, A. H. 1974. *Numerical Quadrature and Solution of Ordinary Differential Equations*. Nueva York: Springer-Verlag.
- Stroud, A. H. y D. Secrest. 1966. *Gaussian Quadrature Formulas*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Subbotin, Y. N. 1967. “On piecewise-polynomial approximation.” *Matematicheskie Zametki* **1**, 63–70. (Traducción: 1967. *Math. Notes* 1, 41–46).
- Szabo, F. 2002. *Linear Algebra: An Introduction Using MAPLE*. San Diego, California: Harcourt/Academic Press.
- Torczon, V. 1997. “On the convergence of pattern search methods.” *Society for Industrial and Applied Mathematics Journal on Optimization* **7**, 1–25.
- Törn, A. y A. Zilinskas. 1989. *Global Optimization*. Notas de conferencia sobre ciencia de la computación 350. Berlín: Springer-Verlag.
- Traub, J. F. 1964. *Iterative Methods for the Solution of Equations*. Englewood Cliffs, New Jersey: Prentice-Hall.
- Trefethen, L. N. y D. Bau. 1997. *Numerical Linear Algebra*. Filadelfia: SIAM.
- Turner, P. R. 1982. “The distribution of leading significant digits.” *Journal of the Institute of Mathematics and Its Applications* **2**, 407–412.
- van Huffel, S. y J. Vandewalle. 1991. *The Total Least Squares Problem: Computational Aspects and Analysis*. Filadelfia: SIAM.

- Van Loan, C. F. 1997. *Introduction to Computational Science and Mathematics*. Sudbury, Massachusetts: Jones and Bartlett.
- Van Loan, C. F. 2000. *Introduction to Scientific Computing*, 2<sup>a</sup> ed. Upper Saddle River: New Jersey: Prentice-Hall.
- Van der Vorst, H. A. 2003. *Iterative Krylov Methods for Large Linear Systems*. Nueva York: Cambridge University Press.
- Varga, R. S. 1962. *Matrix Iterative Analysis*. Englewood Cliffs: New Jersey: Prentice-Hall. (2000. *Matrix Iterative Analysis: Second Revised and Expanded Edition*. Nueva York: Springer-Verlag).
- Wachspress, E. L. 1966. *Iterative Solutions to Elliptic Systems*. Englewood Cliffs: New Jersey: Prentice-Hall.
- Watkins, D. S. 1991. *Fundamentals of Matrix Computation*. Nueva York: Wiley.
- Westfall, R. 1995. *Never at Rest: A Biography of Isaac Newton*, 2<sup>a</sup> ed. Londres: Cambridge University Press.
- Whittaker, E. y G. Robinson. 1944. *The Calculus of Observation*, 4<sup>a</sup> ed. Londres: Blackie. Nueva York: Dover, 1967 (reimpreso).
- Wilkinson, J. H. 1965. *The Algebraic Eigenvalue Problem*. Oxford: Clarendon Press. Reimpreso 1988. Nueva York: Oxford University Press.
- Wilkinson, J. H. 1963. *Rounding Errors in Algebraic Processes*. Englewood Cliffs, New Jersey: Prentice-Hall. Nueva York: Dover 1994 (reimpreso).
- Wood, A. 1999. *Introduction to Numerical Analysis*. Nueva York: Addison-Wesley.
- Wright, S. J. 1997. *Primal-Dual Interior-Point Methods*. Filadelfia: SIAM.
- Yamaguchi, F. 1988. *Curves and Surfaces in Computer Aided Geometric Design*. Nueva York: Springer-Verlag.
- Ye, Yinyu. 1997. *Interior Point Algorithms*. Nueva York: Wiley.
- Young, D. M. 1950. Iterative methods for solving partial difference equations of elliptic type. Tesis de doctorado. Cambridge, MA: Harvard University. Véase [www.sccm.stanford.edu/pub/sccm/david\\_young\\_thesis.ps.gz](http://sccm.stanford.edu/pub/sccm/david_young_thesis.ps.gz).
- Young, D. M., 1971. *Iterative Solution of Large Linear Systems*. Nueva York: Academic Press: Dover 2003 (reimpreso).
- Young, D. M. y R. T. Gregory. 1972. *A Survey of Numerical Mathematics*, vols. 1-2. Reading, Massachusetts: Addison-Wesley. Nueva York: Dover 1988 (reimpreso).
- Ypma, T. J. 1995, "Historical development of the Newton-Raphson method." *Society for Industrial and Applied Mathematics Review* 37, 531–551.
- Zhang, Y. 1995. "Solving large-scale linear programs by interior-point methods under the MATLAB environment." Reporte técnico TR96-01, Departamento de Matemáticas y Estadística, Universidad de Maryland, Baltimore County, Baltimore, MD.

# Índice

$A^{-1}$ , cálculo de, 307  
Agencia Espacial Europea, 54  
Ajuste de bondad, 374  
Álgebra. Véase Álgebra lineal  
Álgebra lineal, 706–723  
    bases para, 718–720  
    cambio en similaridad de, 719, 720  
    espacios vectoriales abstractos en, 716–723  
    independencia lineal en, 717, 718  
    matrices en, 708–710  
    matrices ortogonales y teorema especial en, 720, 721  
    matrices simétricas en, 714, 715  
    matrices transpuestas en, 713, 714  
    normas para, 721, 722  
    proceso de Gram-Schmidt para, 722, 723  
    producto matricial en, 711–713  
        producto matriz–vector en, 711  
Regla de Cramer y, 715  
subespacios en, 717  
transformaciones lineales para, 718, 719  
valores propios y vectores propios en, 719  
vectores en, 706–708  
Algoritmo completo de Horner, 23, 24  
Algoritmo de Berman, 638 (problema de cómputo 16.1.5)  
Algoritmo de búsqueda de la sección áurea, 631–633  
Algoritmo de Horner, 7, 23, 24  
Algoritmo de interpolación cuadrática, 633–635  
Algoritmo de Moler-Morrison, 122 (problema de cómputo 3.3.14)  
Algoritmo de Neider-Mead, 647, 648  
Algoritmo de Neville, 142–144  
Algoritmo de Romberg  
    convergencia en, 165  
    descripción de, 204, 205  
    extrapolación de Richardson y, 168, 209–211  
    fórmula de Euler-Maclaurin y, 206–209  
    notación para, 196  
    seudocódigo para, 205, 206  
Algoritmo de variable métrica, 647  
Algoritmo tridiagonal normalizado, 289 (problema de cómputo 7.2.12)

Algoritmos  
    Berman, 638 (16.1.5)  
    Búsqueda de Fibonacci, 628–631  
    búsqueda de la sección áurea, 631–633  
    caso de minimización de funciones de múltiples variables, 644–646  
    completo de Horner, 7, 23, 24  
    de conversión de bases de números, 696  
    de Neville, 142–144  
    funciones splines cúbicas naturales, 388–392  
    Gauss-Huard, 279, 280 (problema de cómputo 7.2.24)  
    Gaussiano, 248, 250, 251  
    gradiente conjugado, 334  
    interpolación cuadrática, 633–635  
    interpolación polinomial, 136–138  
    método de disparo para ecuaciones diferenciales ordinarias, 565–567  
    método de la secante para raíces de ecuaciones, 112, 113  
    método de potencias, 361, 362  
    Moler-Morrison, 122 (problema de cómputo 3.3.14)  
    Neider-Mead, 647, 648  
    Newton, 129  
    números aleatorios, 533–535, 535  
    proceso de Gram-Schmidt, 519  
    recta de mínimos cuadrados, 497  
    simplex, 672, 673  
    sistemas ortogonales, 508–510  
        tridiagonal normalizada, 289 (problema de cómputo 7.2.12)  
    variable métrica, 647  
Análisis de convergencia  
    en el método de bisección, 81–83  
    en el método de la secante, 114–116  
    en el método de Newton, 93–96  
Análisis directo de error, 52  
Análisis hacia atrás del error, 52  
Antiderivada, 181. Véase también Integración numérica  
A producto interno, de vectores, 332  
Aproximación racional de Padé, 41 (problema de cómputo 1.2.22), 73 (problema de cómputo 2.2.17)  
Aproximación. Véase Método de mínimos cuadrados;  
    Funciones spline

Aritmética  
    Babilónica, 701  
    IEEE norma de punto flotante del, 703–705  
    Maya, 700, 701  
    parcial de doble precisión, 492 (problema de cómputo 11.3.2)  
Arreglos, 686, 688, 689  
Atraso de ecuaciones diferenciales ordinarias, 450 (problema de cómputo 10.2.17)  
Aumento de sucesiones, 562 (problema de cómputo 13.3.27)  
Barajear, 562 (problema de cómputo 13.3.27)  
Base independiente de linealidad máxima, 718  
Bases numéricas, 692–702  
     $\beta$ , 693  
    conversión entre, 693–696  
    16, 698  
    10, 692, 693  
    de 10 a 8 a 2, 696–698  
Bibliotecas de programa, 686, 687  
Bibliotecas, programa, 10, 686, 687  
Bits escondidos, 47  
Bloque pentadiagonal de sistemas de ecuaciones lineales, 285, 286  
Búsqueda binaria, para intervalos (problema de cómputo 9.1.2)  
Cálculo, Teorema fundamental del, 181, 195  
Cálculos simbólicos, 435  
Caso de minimización de funciones de una variable, 625–639  
    algoritmo búsqueda de Fibonacci y, 628–631  
    algoritmo de búsqueda de la sección áurea y 631–633  
    algoritmo de interpolación cuadrática y 633–635  
    caso especial de, 626, 627  
    funciones unimodales  $F$  como, 627, 628  
    problemas con y sin restricciones en, 625, 626  
Caso de varias variables de minimización de funciones  
    algoritmo de Neider-Mead para, 647, 648

- algoritmos avanzados para, 644–646  
 diagramas de contorno para, 644  
 matriz definida positiva  $y$ , 647  
 método de recocido simulado para, 648, 649  
 métodos de quasi-Newton para, 647  
 mínimo, máximo y puntos silla en, 646  
 procedimiento de descenso abrupto para, 643  
 Series de Taylor para  $F$  en, 640–642  
 Caso práctico en programación, 687–691  
 Caso solución, algoritmo de interpolación cuadrática, 634  
 Caso usual, de algoritmo de interpolación cuadrática, 634  
 Casos de prueba, 685  
 Casos frontera, 685  
 Centroides, 648  
 Ciclos limpios, 686  
 Cociente de Rayleigh, 368 (problema 8.3.7)  
 Código, en módulos, 685, 687, 688  
 Coeficientes  $a_j$ , 131–136  
 Coeficientes indeterminados, método de, 233  
 Combinaciones lineales, 707  
 Componentes, en vectores, 706  
 Comprobación simbólica, 20 (problema de cómputo 1.1.26)  
 Computación, ruido en, 174  
 Conjunto factible, de vectores, 658  
 Conjunto poliedro, 671  
 Conjuntos 2 simplex, 648  
 Conjuntos 3 simplex, 648  
 Conjuntos linealmente independientes, 501  
 Conjuntos  $n$  simplex, 648  
 Constante de Euler, 59, 60 (problema de cómputo 2.1.7)  
 Constantes de Lebesgue, 73 (problema de cómputo 2.2.15)  
 Continuidad de funciones, 373–375  
 Convergencia cuadrática, 93, 100  
 Convergencia lineal, 82  
 Convergencia superlineal, 84, 115  
 Convergencia, teoremas de, 328–331  
 Corolarios en diferencias divididas, 160  
 Cuenca de atracción fractales, 99, 100, 108 (problema de cómputo 3.2.27)  
 Curvas. Véase Ecuaciones diferenciales ordinarias;  
   Funciones spline  
 Curvas de Bézier, 416–418  
 Curvas divergentes, 458  
 Curvas envolventes, 371  
 Curvas francesas, 371  
 Curvas serpentinas, 395  
 Deficiencia cercana en rango, matriz con, 526  
 Definición inductiva, en el método de Newton, 91  
 Deflación de polinomios, 8, 11  
 Derivada generalizada de Lanczos, 178 (problema 4.3.21)  
 Derivadas, 164–179  
   cálculo de series de Taylor de, 164–166  
   de funciones, 9, 10  
   de Lanczos generalizada, 178 (problema 4.3.21)  
   de splines B, 408  
   diferencias divididas  $y$ , 159  
     estimación de la interpolación de polinomios de, 170–174  
   extrapolación de Richardson para, 166–170  
   ruido en computación  $y$ , 174  
 Desbordamiento, de rango, 45  
 Descomposición de Pierce, 356 (problema 8.2.6 p, valor calculado de, 12 (problema 1.1.1, problema 1.1.4))  
 Descomposición de valor singular (SVD)  
   ejemplos numéricos de, 351–353  
   método de mínimos cuadrados  $y$ , 519, 522–527  
   teoría espectral de matrices  $y$ , 350  
   valores propios y vectores propios  $y$ , 348, 349  
   versión económica de, 356 (problema 8.3.5)  
 Descomposición, en factorizaciones matriciales, 296  
 Desigualdad de Cauchy-Schwartz, 503 (problema 12.1.9), 643  
 Desigualdad triangular, 320, 721  
 Desviación estándar, 15 (problema de cómputo 1.1.7)  
 Determinantes, 278 (problema de cómputo 7.2.14)  
 Diagramas de contorno, 644  
 Diferenciación, 718  
 Diferencias divididas para el cálculo de coeficientes  $a_j$ , 131–136  
   corolario en, 160  
   derivadas  $y$ , 159  
 Dimensión, 718  
 Diseño geométrico asistido por computadora, 425 (problema de cómputo 9.3.19)  
 División sintética, 7  
 Doble precisión parcial aritmética, 492 (problema de cómputo 11.3.2)  
 Ecuación biarmónica, 583  
 Ecuación de aceleración de Aiken, 363  
 Ecuación de advección, 601, 602  
 Ecuación de Cauchy-Riemann, 105 (problema 3.2.40)  
 Ecuación de difusión, 584  
 Ecuación de Kepler, 106 (problema de cómputo 3.2.6)  
 Ecuación delta de Kronecker, 145  
 $k$ -ésimo residual, 519  
 Ecuación de Navier-Stokes, 583, 584  
 Ecuación de Neumann, 584  
 Ecuación de onda, modelo 582, 584, 596, 597  
 Ecuación de Poisson, 584, 605, 613, 615  
 Ecuación diferencial de Airy, 483 (problema de cómputo 11.2.2)  
 Ecuación generalizada de Neumann, 584  
 Ecuación mixta de Dirichlet/Neumann, 584  
 Ecuaciones características, 719  
 Ecuaciones de Laplace, 286, 583, 584, 605, 606, 618  
 Ecuaciones diferenciales, 353–355. Véase también Ecuaciones diferenciales ordinarias; ecuaciones diferenciales parciales  
 Ecuaciones diferenciales ordinarias (EDO), 426–464  
   algoritmo para, 565–567  
   análisis de estabilidad para, 456–459  
   campos vectoriales en, 429–431  
   en el caso lineal, 574, 575  
   fórmulas de Adams-Basforth-Moulton para 455, 456  
   integración y 428, 429  
   método de discretización para, 570–572  
   métodos de Runge-Kutta para, 439–450  
     adaptado, 450–454  
     de orden 2, 441, 442  
     de orden 4, 442, 443  
     ejemplo de, 454, 455  
     seudocódigo para, 443, 444  
     series de Taylor en dos variables  $y$ , 440, 441  
   métodos de series de Taylor para, 431–435  
   problema con valor en la frontera en, 563–581  
   problema con valor inicial en, 426–428  
   seudocódigo del método de Euler para, 432, 433  
   seudocódigo para, 575–577  
   tipos de error en, 435  
   vista general de, 563–565  
 Ecuaciones diferenciales ordinarias, sistemas de, 465–494  
   métodos de Adams-Basforth-Moulton para, 483–494  
   ecuaciones rígidas  $y$ , 489–491  
     ejemplo de, 488, 489  
   esquema adaptado para, 488  
   esquema predictor-corregidor en, 483, 484  
   seudocódigo para, 484–488  
   métodos de primer orden para, 465–477  
   notación vectorial para, 467–469  
     para EDO autónoma, 471  
   Runge-Kutta, 469–471  
   Series de Taylor, 466–469  
   sistemas desacoplados y acoplados en, 465, 466  
   orden más alto, 477–483  
 Ecuaciones diferenciales ordinarias autónomas, 471, 472, 479, 480  
 Ecuaciones diferenciales parciales, 582–624  
   problemas elípticos en, 605–624  
   método de diferencias finitas para, 606–609

- método de elemento finito para, 613–619  
 método iterativo de Gauss-Seidel para, 610  
 modelo de la ecuación de Helmholtz, 605, 606  
 seudocódigo para, 610–613  
 problemas hiperbólicos en, 596–605  
 ecuación de advección como, 601  
 método contra el viento para, 602  
 método de Lax para, 602  
 método de Lax-Wendroff para, 602, 603  
 modelo de la ecuación de onda, 596, 597  
 seudocódigo para, 600, 601  
 solución analítica para, 597, 598  
 solución numérica para, 598, 599  
 problemas parabólicos en, 582–596  
 aplicados, 582–585  
 estabilidad y, 591–593  
     método alternativo de Crank–Nicolson para, 590, 591  
     método de Crank-Nicolson para, 588, 589  
     modelo de la ecuación de calor como, 585, 586  
     seudocódigo para Crank–Nicolson,  
         método para, 589, 590  
         seudocódigo para modelo explícito de, 587
- Ecuaciones lineales, sistemas de, 245–370  
 bloque pentadiagonal, 285, 286  
 dominio estrictamente diagonal en, 282, 283  
     en banda, 280–292  
     pentadiagonal, 283–285  
     tridiagonal, 280–282  
 valores propios y vectores propios en, 342–360  
     cálculo de, 343, 344  
     descomposición de valor singular de, 348, 349, 351–353  
     en ecuaciones diferenciales lineales, 353–355  
     en software matemático, 344  
     formulación matricial para, 331, 332  
     método de gradiente conjugado de, 332–335  
     métodos básicos de, 322–327  
     normas de vector y matriz en, 319, 320  
     número de condición y mal condicionado en, 321, 322  
     propiedades de, 345–347  
     seudocódigo para, 327, 328  
     sobrerrrelajación en, 332  
     Teorema de Gershgorin y, 347, 348  
     teoremas de convergencia para, 328–331  
     teoría espectral de matrices de, 349–351  
         desplazamiento inverso, 365, 366  
         eliminación gaussiana simple de, 245–258  
             algoritmo para, 248–250  
             ejemplo de, 247, 248  
             falla de, 259, 260  
             seudocódigo para, 250–254  
             vectores residual y de error en, 254, 255, 279 (problema de cómputo 7.2.19)  
         factorizaciones matriciales en, 293–319  
              $A^{-1}$  en, 307  
             deducción de, 296–300  
             ejemplo de, 294–296  
             ejemplo de paquete de software, 307–309  
             factorización de Cholesky como, 305, 306  
             factorización  $LDL^T$  como, 302–304  
             factorización  $LU$  como, 300–302  
             múltiples lados derechos en, 306, 307  
             seudocódigo para, 300  
         método de potencias para, 360–370  
             algoritmos para, 361, 362  
             en software matemático, 363  
             fórmula de la aceleración de Aiken para, 363  
             inversa, 364, 365
- Ecuaciones normales, 497, 499, 501, 617
- Ecuaciones, raíces de. Véase Raíces de ecuaciones, localización  
 Error. Véase también Interpolación polinomial  
 absoluta y relativa, 5  
 análisis de la regla del trapezo de, 192–196  
 en ecuaciones diferenciales ordinarias (EDO), 435  
 redondeo, 50, 52, 54, 63, 253, 687  
 redondeo unitario, 703  
 truncamiento, 165, 166, 174  
 un solo paso, 453  
 vectores de, 254, 255, 279 (problema de cómputo 7.2.19)
- Ecuaciones rígidas, 489–491
- Elementos, en vectores, 706, 708
- Eliminación gaussiana con pivoteo escalado parcial de, 259–280  
     ejemplo de, 265, 266  
     estabilidad numérica de, 271  
     inconsistente, 675–683  
     operación de conteo largo para, 269, 270  
     pivoteo parcial completo contra, 261–264  
     seudocódigo para, 266–269  
     soluciones iterativas de, 319–341
- Eliminación gaussiana con pivoteo escalado parcial, 259, 280  
     pivoteo completo *contra*, 261–264  
     ejemplo de, 265, 266  
     estabilidad numérica de, 271  
     operación de conteo largo para, 269, 270  
     seudocódigo para, 266–269  
     simple, 245–258  
         algoritmo para, 248–250  
         ejemplo de, 247, 248  
         en factorizaciones matriciales, 295, 296, 311 (problema 8.1.1)  
         falla de, 259, 260  
         seudocódigo para, 250–254  
         vectores residuales y de error en, 254, 255
- Eliminación hacia adelante, en el algoritmo gaussiano, 248, 250
- Entrada de datos, en vectores, 706, 708
- Envoltura convexa, de vectores, 417
- Épsilon de la máquina, 47, 48, 703
- Error de redondeo, 50, 52, 54, 63, 253, 435, 687, 703
- Error de truncamiento, 165, 166, 174, 435 local, 435
- Error de un solo paso, 453
- Error unitario de redondeo, 50, 703
- Errores absolutos, 5
- Errores relativos, 5
- Escalamiento, 271
- Espacios vectoriales abstractos en álgebra lineal, 716–723  
 bases para, 718  
 cambio en similaridad de, 719, 720  
 independencia lineal para, 717, 718  
 matrices ortogonales y el teorema espectral en, 720, 721  
     normas para, 721, 722  
     proceso de Gram-Schmidt para, 722, 723  
     subespacios en, 717  
     transformaciones lineales para, 718, 719  
     valores y vectores propios en, 719
- Espectral/ $l_2$ -norma matricial, 320. Véase también Teoría espectral de matrices
- Espectral/ $l_2$ -norma vectorial, 722
- Esquema adaptado para los métodos de Adams-Bashforth-Moulton, 488  
 ecuaciones rígidas y, 489–491  
 ejemplo de, 488, 489  
 esquema predictor-corrector en, 483, 484  
     para ecuaciones diferenciales de primer orden, 455, 456
- problemas en, 241 (problema 6.2.15), 461 (Problema de cómputo 10.3.2–4)  
 seudocódigo para, 484–488
- Esquema predictor-corrector, 461 (problema de cómputo 10.3.4), 483, 484
- Estabilidad  
 en ecuaciones diferenciales ordinarias (EDO), 456–459

- en ecuaciones diferenciales parciales, 591–593  
numérica, 271
- Estado estable de sistemas, 489
- Estándar aritmética de punto flotante del IEEE, 703–705
- Estimación de área y volumen, 544–552  
cálculo, 547, 548  
“barquillo de helado”, ejemplo de, 548  
integración numérica para, 544, 545  
seudocódigo para, 545–547
- Estructura molecular, 655 (problema de cómputo 16.2.2), 655 (problema de cómputo 16.2.10)
- Euclidiana/ $l_2$ -vector norma, 721
- Exactitud  
en soluciones de una ecuación diferencial ordinaria (EDO), 435  
polinomio de primer grado, 375  
precisión y 5, 6  
spline de primer grado, 375
- Exactitud del teorema del polinomio de primer grado, 375
- Exactitud del teorema del spline de primer grado, 375
- Expansión finita, 44
- Exponentes, 44, 544 (problema de cómputo 13.1.20), 687
- Extrapolación de Richardson  
algoritmo de Romberg, 209–211  
estimación de derivadas y, 166–170, 177  
(Problema 4.3.19) fórmula de Euler-Maclaurin y, 207
- Factor de relajación, 326. *Véase también* Sobrrelajación
- Factores, 296. *Véase también* Factorizaciones matriciales
- Factorización de Cholesky, 305, 306, 315 (problema 8.1.24)
- Factorización de Crout, 317 (problema de cómputo 8.1.2)
- Factorización de Doolittle, 300, 317 (problema de cómputo 8.1.2)
- Factorización dispersa, 315 (problema 8.1.24)
- Factorización  $LU$   
deducción de, 296–300  
descripción de, 294  
problemas en, 314, 315 (problema 8.1.18), 319 (problema de cómputo 8.1.14)  
solución de sistemas lineales con, 300–302
- Factorizaciones  $LDL^T$ , 302–304, 315 (problema 8.1.24)
- Factorizaciones matriciales, 293–319  
 $A^{-1}$  en, 307  
Cholesky, 305, 306  
deducción de, 296–300  
ejemplo de paquete de software de, 307–309  
ejemplo de, 294–296  
 $LDL^T$ , 302–304  
 $LU$ , 300–302
- múltiples lados derechos en, 306, 307  
seudocódigo para, 300
- Filtro spline no periódico, 291 (problema de cómputo 7.2.22)
- Filtro spline periódico, 292 (problema de cómputo 7.2.23)
- Forma anidada de interpolación polinomial, 130, 131
- Forma cuadrática, 333
- Forma de Newton de la interpolación de polinomios, 128–130, 133, 150, 151 (problema 4.1.38), 164 (problema de cómputo 4.2.14)
- Fórmula de cinco puntos para la ecuación de Laplace, 606, 607
- Fórmula de Euler-Maclaurin, 206–209, 214 (problema 5.3.26)
- Fórmula de la diferencia central, 15 (problema de cómputo 1.1.3), 166, 171
- Fórmula de nueve puntos para la ecuación de Laplace, 607, 621 (problema 15.3.10)
- Fórmula de Stirling, 34 (problema 1.2.47)
- Fórmula irregular de cinco puntos para la ecuación de Laplace, 607
- Fórmula recursiva trapezoidal para subintervalos iguales, 196, 197
- Formulaciones matriciales, 331, 332
- Fórmulas de cuadratura gaussiana, 230–244  
cambio de intervalos en, 231  
comuesta de tres puntos, 243 (problema de cómputo 6.2.11)  
descripción de, 230, 231  
integrales con singularidades en, 237–239  
nodos y pesos en, 232–234
- polinomios de Legendre en, 234–237
- Fórmulas de cuadratura de Gauss-Legendre, 232
- Fórmulas de primera derivada, 164–166, 170–174
- Fórmulas de segunda derivada, 173, 174
- Función de Dirichlet, 154, 184, 584, 593, 618
- Función de error, 34 (problema 1.2.52), 185, 186
- Función de Runge, 125, 154–156
- Función peso, 519, 520
- Función tienda de campaña, 122 (problema de cómputo 3.3.15)
- Funciones racionales telescopiadas, 73 (problema de cómputo 2.2.18)
- Funciones armónicas, 607, 618
- Funciones base, 500, 501, 505–508
- Funciones cuadráticas, 642, 652 (problema 16.2.15)
- Funciones cuadráticas generales, 652 (problema 16.2.15)
- Funciones de Bessel, 42 (problema de cómputo 1.2.23), 186, 215 (problema de cómputo 5.3.11)
- Funciones de dos variables, 144, 145
- Funciones de splines cúbicos naturales  
algoritmo para, 388–392
- curvas espaciales de, 394–396  
introducción a, 385–387  
seudocódigo para, 392–394  
propiedad de suavidad de, 396–398
- Funciones gaussianas continuadas, 73 (problema de cómputo 2.2.18)
- Funciones impares periódicas, 598
- Funciones inestables, raíces como, 88 (problema de cómputo 3.1.12)
- Funciones integrables de Riemann, 183, 184
- Funciones lineales, 361, 641
- Funciones lineales en partes, 372
- Funciones, minimización de, 625–658  
caso de varias variables de, 639–656  
algoritmo de Neider-Mead para, 647, 648  
algoritmos avanzados para, 644–646  
diagramas de contorno para, 644  
matriz definida positiva y 647  
método de recocido simulado para, 648, 649  
métodos de quasi-Newton para, 647  
mínimo, máximo y puntos silla en, 646  
procedimiento del descenso pronunciado para, 643
- Series de Taylor para  $F$  en, 640–642  
caso de una variable de, 625–639  
algoritmo de búsqueda de Fibonacci y, 628–631  
algoritmo de búsqueda de la sección áurea y, 631–633  
algoritmo interpolación cuadrática y, 633–635  
caso especial de, 626, 627  
funciones unimodales  $F$  como, 627, 628  
problemas con y sin restricciones en, 625, 626
- Funciones objetivo, 658
- Funciones poligonales, 372
- Funciones sombrero de splines B, 406
- Funciones spline, 371–425
- B, splines, 404–425  
interpolación y aproximación por, 410–412  
para curvas de Bézier, 416–418  
proceso de Schoenberg para, 414, 415  
seudocódigo y ejemplo de, 412, 413  
teoría de, 404–410  
primer-grado, 371–374  
algoritmo para, 388–392  
cúbico natural, 385–404  
curvas espaciales de, 394–396  
interpolación cuadrática,  $Q(x)$ , 376–378  
introducción a, 385–387  
módulos de continuidad en, 374, 375  
propiedad de suavidad de, 396–398  
seudocódigo para, 392–394  
segundo-grado, 376  
cuadrático de Subbotin, 378–380

- Funciones spline B, 404–425  
interpolación y aproximación por, 410–412  
para Curvas de Bézier, 416–418  
seudocódigo y ejemplo de, 412, 413  
proceso de Schoenberg para, 414, 415  
teoría de, 404–410  
Funciones unimodales F, 627, 628
- Galerkin ecuación de, 617  
Gauss-Huard algoritmo de, 279, 280 (problema de cómputo 7.2.24)  
Gradiente de formas cuadráticas, 333  
Gradiente vector matriz, 640, 641  
Gran búsqueda de números primos de Mersenne por Internet (GIMPS), 541
- Histogramas, 560 (problema de cómputo 13.3.13)
- IMSL, biblioteca matemática, 10  
Integración bidimensional sobre el cuadrado unitario, 198  
Integración gaussiana adaptada de dos puntos, 242 (problema de cómputo 6.2.7)  
Integración multidimensional, 198, 199  
Integración numérica, 180–244, cálculo para área y volumen, 544, 545  
algoritmo de Romberg en, 204–215  
descripción de, 204, 205  
extrapolación de Richardson de, 209–211  
fórmula de Euler-Maclaurin y, 206–209  
seudocódigo para, 205, 206  
cambio de intervalos en, 231  
de ecuaciones diferenciales ordinarias (EDO), 428, 429  
definida e indefinida, 180, 181  
descripción de, 230, 231  
fórmulas de cuadratura gaussiana en, 230–244  
funciones integrables de Riemann en, 183, 184  
integrales con singularidades en, 237–239  
nodos y pesos en, 232–234  
polinomios de Legendre en, 234–237  
seudocódigo y ejemplos de, 184–187  
sumas inferior y superior en, 181–183  
regla del trapecio en, 190–204  
análisis de error en, 192–197  
espaciado uniforme en, 191, 192  
integración multidimensional en, 198, 199  
regla de Simpson en, 216–229  
adaptada, 221–225  
básica, 216–220  
compuesta, 220, 221  
reglas de Newton-Cotes y, 225, 226
- Integral de Dawson, 439 (problema de cómputo 10.1.12)  
Integral de Fresnel, 186, 204 (problema de cómputo 5.2.5)  
Integral de probabilidad, 204 (problema de cómputo 5.2.5)  
Integral logarítmica, 186, 189 (problema de cómputo 5.1.3)  
Integrales  
Dawson, 439 (problema de cómputo 10.1.12)  
elípticas, 39 (problema de cómputo 1.2.14), 180, 186 seno, 189 (problema de cómputo 5.1.2), 204  
(problema de cómputo 5.2.5), 463 (problema de cómputo 10.3.15)  
Integrales elípticas, 39 (problema de cómputo 1.2.14), 180, 186  
Interpolación de Padé, 153 (CPb 4.1.17)  
Interpolación de spline cúbico, 371. Véase también Funciones spline  
Interpolación del producto tensorial, 144  
Interpolación lineal, 162 (problema 4.2.8)  
Interpolación polinomial, 124–164  
algoritmo de Neville para, 142–144  
algoritmos y seudocódigo para, 136–138  
de funciones de dos variables, 144, 145  
diferencias divididas para calcular coeficientes  $a_j$  en, 131–136  
errores en, 153–164  
función de Dirichlet como, 154  
función de Runge como, 154–156  
teoremas en, 156–160  
estimación de derivadas por, 170–174  
forma anidada de, 130, 131  
forma de Lagrange de, 126–128  
forma de Newton de, 128–130  
inversa, 141, 142  
lineal, 125, 126, 162 (problema 4.2.8)  
matriz de Vandermonde para, 139–141  
Interpolación polinomial inversa, 141, 142, 567  
Interpolación polinomial lineal, 125, 126  
Interpolación. Véase Funciones spline B; Interpolación polinomial; Algoritmo de interpolación cuadrática  
Iteración de punto fijo, 117, 118  
Iteración de Richardson, 322, 323  
Iteraciones. Véase también Ecuaciones lineales, sistemas de  
aproximación  $l_1$ , 496  
limitante, 689  
norma matricial  $l_\infty$ , 320, 722  
Newton-Raphson, 89  
 $l_\infty$  norma vectorial, 320, 721  
 $l_\infty$  problema, 678–680  
 $l_1$  norma matricial, 320, 722  
punto fijo, 117, 118  
Richardson, 322, 323
- $l_1$  norma vectorial, 320, 721  
 $l_1$  problema, 676, 678  
LAPACK, software matemático, 344, 351  
Lagrange, forma del polinomio de interpolación, 25, 126–128, 144  
Lema, límite superior, 157  
Leyes de movimiento, de Newton, 428, 465  
Límite inferior más grande, en integración, 182  
Límite inferior mínimo, en integración, 182  
Límite superior mínimo, de un conjunto de números, 374  
Linealización y método de solución para la resolución de ecuaciones no lineales, 96, 117  
Llamadas de memoria, 688  
Logaritmo natural (ln), 1  
Longitud de vectores, 320  
Lugares decimales, exactitud, 5
- Macsyma, software matemático, 10  
Magnitud de vectores, 320  
Mal condicionamiento, 321, 322, 448 (problema de cómputo 10.2.5)  
Mantisa, 47  
Mantisa normalizada, 44, 47  
Manual de funciones matemáticas con fórmulas, gráficas y tablas matemáticas (Abramowitz y Stegun), 186  
Maple, software matemático, 10  
comprobación simbólica en, 20 (problema de cómputo 1.1.26)  
descomposición de valor singular, 351  
ecuaciones diferenciales parciales, 592  
ecuaciones diferenciales, 427  
ecuaciones no lineales, 99, 111  
(problema de cómputo 3.2.42), 123 (problema de cómputo 3.3.19)  
factorización LU en, 308  
función error en, 186  
interpolación polinomial en, 153  
(problema de cómputo 4.1.11), 164  
(problema de cómputo 4.2.12)  
números aleatorios, 533, 535  
problema con valor en la frontera, 577  
problemas de minimización, 626  
programación lineal, 678, 679  
raíces de ecuaciones en, 81, 88  
(problema de cómputo 3.1.12), 93  
solución mínima, 526  
splines, 409, 410, 418  
valores propios, 343, 344
- Mathematica, software matemático, 10  
comprobación simbólica en, 20 (problema de cómputo 1.1.26)  
ecuaciones diferenciales parciales, 592  
ecuaciones diferenciales, 427  
ecuaciones no lineales, 99, 111 (problema de cómputo 3.2.42), 123  
(problema de cómputo 3.3.19)  
factorización LU en, 308  
función de error en, 186  
interpolación polinomial en, 153

- (problema de cómputo 4.1.11), 164  
 (problema de cómputo 4.2.12)
- números aleatorios, 533, 535  
 problema con valor en la frontera, 577  
 problemas de minimización, 626  
 programación lineal, 678, 679  
 raíces de ecuaciones en, 81, 88 (problema de cómputo 3.1.12), 93  
 solución mínima, 526  
 splines, 418  
 valores propios, 343, 344
- Matlab, software matemático, 10  
 Caja de herramientas EDP, 584, 592, 593, 612  
 campos vectoriales, 430  
 condición final de un no nudo, de splines, 394  
 descomposición de valor singular, 351  
 ecuaciones no lineales en, 99, 111  
 (Problema de cómputo 3, 2.42), 123  
 (Problema de cómputo 3.3.19)  
 factorización LU en, 308  
 función de error en, 186  
 interpolación polinomial en, 153  
 (problema de cómputo 4.1.11), 164  
 (problema de cómputo 4.2.11)  
 números aleatorios, 533, 535  
 problema con valor en la frontera, 577  
 problemas de minimización, 626  
 programación lineal, 678, 679  
 raíces de ecuaciones en, 81, 88 (problema de cómputo 3.1.12), 93  
 solución mínima, 526  
 splines, 409  
 valores propios, 343, 344
- Matrices. Véase también Álgebra lineal;  
 Descomposición en valor singular (SVD)  
 acompañante, 358 (problema de cómputo 8.2.3)  
 deficiencia cercana en rango, 526  
 de Hilbert, 276 (problema de cómputo 7.2.4), 527 (problema 12.3.2)  
 de Vandermonde, 139–141, 152 (problema 4.1.47), 254  
 diagonal, 346, 347  
 hermitiana, 345, 346  
 hessiana, 640, 641  
 jacobiana, 97–98  
 permutación, 307  
 positiva definida, 305, 332, 333, 345, 647  
 renglón-equilibrado, 275 (problema 7.2.23)  
 seudoinversa de, 525, 526  
 simétrica definida positiva (SPD) 305, 330  
 simétrica, 332, 345, 640  
 similar, 345  
 Teorema de Gershgorin y, 348  
 transpuesta de, 345  
 triangular, 346  
 unitariamente similar, 345, 346  
 valores singulares de, 349  
 vector gradiente, 640, 641
- Marca de splines B, 424 (problema de cómputo 9.3.6)
- Marcaje problema/método, 586
- Matrices definidas positivas, 305, 332, 333, 345, 647
- Matrices diagonales, 346, 347, 709
- Matrices hermitianas, 345
- Matrices ortogonales, 720, 721
- Matrices simétricas definidas positivas (SPD), 305, 330
- Matrices simétricas, 332, 345, 640, 714, 715
- Matrices similares, 345
- Matrices unitariamente similares, 345, 346
- Matriz compañera, 358 (problema de cómputo 8.2.3)
- Matriz de Hilbert, 276 (problema de cómputo 7.2.4), 527 (problema 12.3.2)
- Matriz de renglón-equilibrado, 275 (problema 7.2.23)
- Matriz diagonal principal, 710
- Matriz hessiana, 640, 641
- Matriz identidad, 709
- Matriz jacobiana, 97, 98, 100
- Matriz subdiagonal, 280, 710
- Matriz superdiagonal, 280, 710
- Matriz triangular, 346, 710
- Matriz triangular inferior, 710
- Matriz triangular superior, 710
- Matriz tridiagonal, 709
- Media aritmética, 15 (problema de cómputo 1.1.7)
- Mejor paso para el procedimiento de descenso pronunciado, 643
- Mensajes de advertencia, 685
- Método completamente implícito para ecuaciones diferenciales parciales, 595 (problema 15.1.13)
- Método contra el viento, 602
- Método de bisección para localizar raíces de ecuaciones, 76–85  
 análisis de convergencia, 81–83  
 ejemplo de, 79–81  
 método de falsa posición en, 83, 84  
 método de la secante y método de Newton *contra*, 117  
 seudocódigo en, 78, 79
- Método de colocación, 618
- Método de Crank-Nicolson, 588–591
- Método de desplazamiento de la potencia inversa, 365, 366
- Método de diferencias finitas, 570, 571, 574, 606–609
- Método de discretización, 570–572
- Método de disparo para ecuaciones diferenciales ordinarias (EDO), 563–570  
 algoritmo para, 565, 567  
 en el caso lineal, 574, 575  
 seudocódigo para, 575–577  
 refinamientos a, 567  
 vista general de, 563–565
- Método de Euler, 432, 433, 437 (problema 10.1.15)
- Método de falsa posición, 83, 84
- Método de Fehlberg de orden 4, 451
- Método de Gauss-Seidel, 323–325, 330, 331, 610
- Método de Halley, 122 (problema de cómputo 3.3.13)
- Método de Heun, 437 (problema 10.1.15)
- Método de Jacobi, 323–325, 330, 331
- Método de la potencia inversa, 364, 365
- Método de la secante para localización de raíces de ecuaciones, 111–119  
 algoritmo para, 112, 113  
 análisis de convergencia en, 114–116  
 bisección y métodos de Newton *contra*, 117  
 iteración de punto fijo y, 117, 118
- Método de Lax, 602
- Método de Lax-Wendroff, 602, 603
- Método de mínimos cuadrados, 495–505, 518–531, 652 (problema 16.1.20)  
 descomposición de valor singular (SVD) y, 522–527  
 ejemplo lineal de, 521, 522  
 ejemplo no lineal de, 520–522  
 ejemplo no polinomial de, 499, 500  
 función base en, 500, 501  
 función peso en, 519, 520
- Método de Muller, 123 (problema de cómputo 3.3.17)
- Método de Newton para localización de raíces de ecuaciones, 89–100  
 análisis de convergencia en, 93–96  
 cuencas de atracción fractales en, 99, 100  
 generalizado, 104 (problema 3.2.37)  
 interpretación de, 90, 91  
 método de bisección y método de la secante *contra*, 117  
 modificado, 104 (problema 3.2.35)  
 seudocódigo en, 92, 93  
 sistemas de ecuaciones no lineales en, 96–99
- Método de Newton para sistemas no lineales, 98
- Método de Oliver, 122 (problema de cómputo 3.3.12)
- Método de potencias para ecuaciones lineales. Véase también Valores propios  
 algoritmos para, 361, 362  
 desplazamiento inverso, 365, 366  
 en software matemático, 363  
 fórmula de la aceleración de Aiken para, 363  
 inversa, 364, 365
- Método de Prony, 530 (problema de cómputo 12.3.2)
- Método de recocido simulado, 648, 649
- Método de Runge-Kutta-England, 463, 464 (problema de cómputo 10.3.19)
- Método de sobrerrelajación sucesiva (SRS), 324, 326, 331, 332
- Método del elemento finito, 613–619
- Método del gradiente conjugado, 332–335
- Método del punto medio, 188 (problema

- 5.1.10), 188 (problema 5.1.12), 201 (problema 5.2.18), 462 (problema de cómputo 10.3.8)
- Método directo, para valores propios, 343
- Método explícito para ecuaciones diferenciales parciales, 587, 591, 595 (problema 15.1.12)
- Método gaussiano para integrales elípticas, 39 (problema de cómputo 1.2.14)
- Método generalizado de Newton, 104 (problema 3.2.36)
- Método mejorado de Euler, 437 (problema 10.1.15)
- Método modificado de falsa posición, 84
- Método modificado de Newton, 104 (problema 3.2.35)
- Método redondeo a un par, 6
- Método Regula Falsi, 83, 84
- Método *Simplex*, 670–675
- Método de sobrerelajación de Jacobi (MSJ), 332
- Métodos adaptados de Runge-Kutta, 450–454
- Métodos de cuasi-Newton para minimización de funciones, 647
- Métodos de Monte Carlo. *Véase también Simulación*
- algoritmos y generadores para, 533–535
  - cálculo de área y volumen por, 544–552
  - cómputo, 547, 548
  - ejemplo del “cono de helado”, 548
  - ejemplos de, 535–537
  - integración numérica para, 544, 545
  - números aleatorios y, 532–544
  - seudocódigo para, 537–541, 545–547
- Métodos de Runge-Kutta, 439–450
- adaptado, 450–454
  - ejemplo de, 454, 455
  - de orden 4, 451
  - de orden 4, 442, 443
  - de orden 3, 445, 446 (problema 10.2.7)
  - de orden 2, 441, 442
  - seudocódigo para, 443, 444
  - para sistemas de ecuaciones diferenciales ordinarias, 469–472
  - series de Taylor en dos variables y, 440, 441
- Métodos de un solo paso, 483
- Métodos multipasos, 483
- Minimización de funciones. *Véase Funciones, minimización de puntos mínimos de funciones*, 626, 646
- Modelo de ecuación de calor, 583–586
- Modelo de ecuación de Helmholtz, 584, 605, 606
- Modelos de depredador-presa, 465
- Modo almacenado en banda, 291 (problema de cómputo 7.2.19)
- Modo de almacenamiento simétrico en banda, 291 (problema de cómputo 7.2.20)
- Modo de almacenamiento simétrico, 278 (problema de cómputo 7.2.13)
- Modo mixto codificado, 687, 688
- Modos de redondeo, 705
- Módulos de continuidad en funciones spline, 374, 375
- Multiplicación anidada, 7–9, 12 (problema 1.1.6), 131
- Multiplicadores, en el algoritmo gaussiano, 249
- NAG, biblioteca matemática, 10
- NaN (No un número), 704
- Newton, leyes de movimiento de, 428, 465
- Newton-Raphson, iteración, 89
- Nodos
- Chebyshev, 155, 156, 158, 163 (problema de cómputo 4.2.10), 174
  - en interpolación polinomial, 125
  - en teoría de splines, 378
  - gaussiano, 230, 232–234
- Norma de Frobenius, 338 (problema 8.1.10)
- Norma de representación de punto flotante, 46
- Norma inducida, 721
- Normas, 319, 320, 721, 722
- Normas de matrices, 319, 320, 721, 722
- Normas subordinadas, 721
- Notación científica normalizada, 43
- Notación factorial, 21
- Notación O grande, 27
- Notación vectorial, 467–469
- Nudos, en teoría de splines, 372, 378
- Número de condición, en ecuaciones lineales, 321, 322
- Número de dimensión finita, 718
- Número primo de Mersenne, 534
- Números aleatorios, 532–544
- algoritmos y generadores para, 533–535
  - ejemplos de, 535–537
  - seudocódigo para, 537–541
- Números de Bernoulli, 208
- Números de corte, 6, 51
- Números de Fibonacci, 40 (problema de cómputo 1.2.16), 115, 628–631
- Números de máquina, 44, 51. *Véase también Números de punto flotante*
- Números de punto flotante, 43–55, 102 (problema 3.2.24)
- doble precisión, 48, 49
  - estándar aritmética del IEEE para, 703–705
  - errores de computación en, 50, 51, 54, 687
  - igualdad de, 689, 690
  - norma, 46
  - normalizada, 44–46
  - número de punto flotante de la máquina  $[f(x)]_y$ , 51–55
  - precisión simple, 46, 47
- Números fraccionarios, bases de conversión de, 695, 696
- Números primos, 534, 540
- Números seudoaleatorios, 533
- Números subnormales, 704
- Números uniformemente distribuidos, 533
- Octave, software matemático, 10
- Operador de proyección, 722
- Optimización, ejemplo, de programación lineal, 658–660
- Ordenamiento de tablero de ajedrez, 620 (problema 15.3.3)
- Ordenamiento natural, 262–264, 609
- Ordenamiento, rojo-negro (tablero de ajedrez), 620 (problema 15.3.3)
- Partes enteras, 696
- Partes fraccionarias, 696
- Partición de la unidad en el intervalo, 417
- Periodicidad, 67, 598
- Permutación, matrices de, 307
- Pesos, gaussiana, 230, 232, 234
- Pivoteo, 246
- ecuación pivot para, 247, 249
  - elemento pivot para, 249, 271
  - parcial escalado, 259–280
  - conteo operacional largo y, 269, 270
  - ejemplo de, 265, 266
  - eliminación gaussiana con, 262–264
  - estabilidad numérica y, 271
  - pivoteo parcial completo y, 261
  - seudocódigo para, 266–269
- Pivoteo completo y parcial, 261–264
- Polinomio bilineal por partes, 384 (problema de cómputo 9.1.3)
- Polinomio de Wilkinson, 88 (problema de cómputo 3.1.12), 121 (problema de cómputo 3.3.9)
- Polinomio(s), 8, 11, 343
- Polinomios característicos, 343
- Polinomios cardinales, 126, 127
- Polinomios de Bernstein, 416
- Polinomios de Chebyshev
- propiedades de, 140, 141
  - sistemas ortogonales y, 505–518
  - algoritmo para, 508–510
  - funciones base ortonormales en, 505–508
  - regresión polinomial en, 510–515
- Polinomios de Legendre, 234–237
- Precondicionado, 335
- Precisión, 3–6, 63, 64, 688. *Véase también Estándar aritmética de punto flotante del IEEE*
- Predominio diagonal, 282, 283, 330
- Primer caso malo, del algoritmo de interpolación cuadrática, 635
- Primera forma primal, en programación lineal, 657, 658, 660, 661, 673
- Problema con valor inicial, 426–428, 431, 463 (problema de cómputo 10.3.17)
- Problema de Bratu, 581 (problema de cómputo 14.2.7)
- Problema de dieta, 670 (problema de cómputo 17.1.5)
- Problema de dos dados, 556, 557
- Problema de la aguja de Buffon, 555, 556
- Problema de Troesch, 581 (problema de cómputo 14.2.7)

- Problema del caminante aleatorio, 561 (problema de cómputo 13.3.17–18)
- Problema del cumpleaños, 553–555
- Problema del dado cargado, 552, 553
- Problema del sistema de ferrocarril francés, 559 (problema de cómputo 13.3.3)
- Problema dual en programación lineal, 661–663, 673
- Problemas con valores en la frontera. Véase Ecuaciones diferenciales ordinarias, problemas con valores en la frontera
- Problemas de minimización restringida, 625, 626
- Problemas de minimización sin restricciones, 625, 626
- Problemas elípticos, en ecuaciones diferenciales, 584, 594 (problema 15.1.1), 605–624
- método de diferencias finitas para, 606–609
  - método iterativo de Gauss-Seidel para, 610
  - métodos de elemento finito para, 613–619
  - modelo de la ecuación de Helmholtz, 605, 606
  - seudocódigo para, 610–613
- Problemas hiperbólicos, en ecuaciones diferenciales, 584, 594 (problema 15.1.1), 596–605
- ecuación de advección como, 601
  - método contra el viento para, 602
  - método de Lax para, 602
  - método de Lax-Wendroff para, 602, 603
  - modelo de la ecuación de onda como, 596, 597
  - seudocódigo para, 600, 601
- Romberg, 165, 168, 204–215
- descripción de, 204, 205
  - extrapolación de Richardson de, 209–211
  - fórmula de Euler-Maclaurin y, 206–209
  - seudocódigo para, 205, 206
  - solución analítica para, 597, 598
  - solución numérica para, 598, 599
- Problemas no lineales de mínimos cuadrados, 520–522
- Problemas parabólicos, en ecuaciones diferenciales, 582–596, 594 (PbLS.1.1)
- aplicados, 582–585
  - estabilidad y, 591–593
  - método alternativo de Crank-Nicolson para, 590, 591
  - método de Crank-Nicolson para, 588, 589
  - modelo de la ecuación de calor como, 585, 586
  - seudocódigo para el modelo explícito de, 587
  - seudocódigo para el método de Crank-Nicolson para, 589, 590
- Procedimiento de descenso abrupto, 643, 655 (problema de cómputo 16.2.2)
- Procedimiento de descenso acelerado y pronunciado, 655 (problema de cómputo 16.2.2)
- Proceso de Gram-Schmidt, 506, 519, 722, 723
- Proceso de Schoenberg, 414, 415
- Producto interno, 332, 512, 708
- Producto punto de vectores, 708
- Producto, matriz, 711–713
- Programación de derivadas, 9, 10
- Programación lineal, 657–683
- solución aproximada de sistemas lineales inconsistentes de, 675–683
  - ejemplo de optimización de, 658–660
  - método simplex para, 670–675
  - primera forma primal en, 657, 658, 660, 661
  - problema dual en, 661–663
  - problema  $l_{\infty}$  para, 678–680
  - problema  $l_1$  para, 676–678
  - segunda forma primal en, 663, 664
- Propiedad de oscilación igual, 141
- Propiedad recursiva del teorema de diferencias divididas, 134
- Propiedades de Penrose, 526, 527
- Proteína plegable, 655 (problema de cómputo 16.2.10)
- Proyección, 356 (problema 8.2.6)
- Proyecto del pandeo de un anillo circular, 581 (problema de cómputo 14.2.8)
- Proyecto Puente de Tacoma Narrows, 493 (problema de cómputo 11.3.9)
- Prueba de aplanoado, 648
- Prueba de Lucas-Lehmer, 540
- Seudocódigo
- algoritmo de gradiente conjugado, 334
  - algoritmo de Romberg, 205, 206
  - cálculo de área y volumen, 545–547
  - como puente, 684
  - ecuaciones lineales, 327, 328
  - eliminación gaussiana con pivoteo escalado parcial, 266–269
  - eliminación gaussiana simple, 250–254
  - factorizaciones matriciales, 300
  - funciones spline B, 412, 413
  - funciones spline cúbicas naturales, 392–394
  - integración numérica, 184–187
  - interpolación polinomial, 136–138
- método de bisección, 78, 79
- Método de Crank-Nicolson, 589, 590
- método de disparo para ecuaciones diferenciales ordinarias (EDO), 575–577
- método de Euler, 432, 433
- método de Gauss-Seidel, 327, 610
- método de Jacobi, 327
  - método de la secante, 112
- método de Newton, 92, 93
- método de potencias, 361, 362
- método sucesivo de sobrerelajación (SRS), 327
- métodos de Adams-Basforth-Moulton, 484–488
- métodos de Runge-Kutta, 443, 444, 453, 454
- métodos de Runge-Kutta-Fehlberg, 452
- modelo explícito de ecuaciones diferenciales parciales, 587
- números aleatorios, 535, 537–541
- problemas del dado cargado, 552, 553
- problemas elípticos, 610–613
- problemas hiperbólicos, 600, 601
- proceso de Schoenberg, 415
- serie de Taylor de orden 4, 468, 469
- Seudoíversa, de matrices, 525, 526
- Punto base, 693
- Punto decimal, 693
- Puntos de control, en el dibujo de curvas, 371, 416
- Puntos de funciones estacionarios, 646
- Puntos máximos de funciones, 646
- Puntos mínimos locales de funciones, 626
- Puntos reflejados, 648
- Puntos reflejados expandidos, 648
- Puntos silla de funciones, 646
- Racionalización de numeradores, 64
- Radio espectral, 320, 329
- Raíces
- de  $f$ , 76–77, 81
  - de multiplicidad, 96, 104 (problema 3.2.35)
  - espurias, 62
  - simple, 93
- Raíces de ecuaciones, localización, 76–123
- método de bisección para, 76–85
- análisis de convergencia en, 81–83
  - ejemplo de, 79–81
  - método de falsa posición en, 83, 84
  - seudocódigo para, 78, 79
  - método de Newton para, 89–100
  - análisis de convergencia en, 93–96
  - cuencas de atracción fractales en, 99, 100
  - interpretación de, 90, 91
  - seudocódigo en, 92, 93
  - sistemas de ecuaciones no lineales en, 96–99
- método de la secante para, 111–119
- algoritmo para, 112, 113
  - análisis de convergencia en, 114–116
  - bisección y métodos de Newton *contra*, 117
  - iteración de punto fijo y, 117, 118
- Raíz múltiple, 96, 104 (problema 3.2.35)
- Raíz simple, 93
- Rango, de computadora, 45
- Razón áurea, 115, 638 (problema de cómputo 16.1.5)
- Recíprocos de números, 102 (problema 3.2.23)

- Redondeo a un valor cercano, 705  
 Redondeo correcto, 50  
 Redondeo de banquero, 6  
 Redondeo de números, 6, 50  
 Redondeo estadístico, 6  
 Reducción de rango, 67, 68  
 Regla adaptada de Simpson, 221–225  
 Regla básica de Simpson, 216–220, 228  
     (problema de cómputo 6.1.8)  
 Regla compuesta de Simpson, 220, 221, 228  
     (problema 6.1.6), 243 (problema de cómputo 6.2.11)  
 Regla compuesta del punto medio para subintervalos iguales, 188 (problema 5.1.12)  
 Regla de Cramer, 715  
 Regla de L'Hópital, 34 (problema 1.2.49)  
 Regla del trapecio, 190–204. *Véase también*  
     Regla de Simpson  
     análisis de error en, 192–196  
     compuesta, 191, 194, 243 (problema de cómputo 6.2.11)  
     compuesta con espaciamiento desigual, 203 (problema 5.2.32)  
     espaciado uniforme en, 191, 192  
     fórmula recursiva para subintervalos iguales en, 196, 197  
     integración multidimensional en, 198, 199  
 Regla de Simpson, 216–229  
     adaptada, 221–225  
     básica, 216–220, 228 (problema 6.1.8)  
     compuesta, 220, 221, 228 (problema 6.1.6), 243 (problema de cómputo 6.2.11)  
 Regla de tres puntos compuesta gaussiana, 243 (problema de cómputo 6.2.11)  
 Regla del rectángulo compuesta (izquierda), 202 (problema 5.2.28)  
 Regla del rectángulo compuesta con espaciado uniforme, 202, 203 (problema 5.2.29)  
 Regla del trapecio compuesta con espaciado desigual, 203 (problema 5.2.32)  
 Regla del trapecio compuesta, 191, 194, 243  
     (problema de cómputo 6.2.11)  
 Regla trapezoidal básica, 190  
 Reglas de cuadratura, 187  
 Reglas de Newton-Cotes, 225, 226, 229  
     (problema de cómputo 6.1.7)  
 Regresión polinomial, 510–515  
 Relación de la integral elíptica de Legendre, 39 (problema de cómputo 1.2.14)  
 Representación de doble precisión y de punto flotante, 48, 49  
 Representación de punto flotante de precisión simple, 46, 47  
 Representación normalizada de punto flotante, 44–46  
 Representación numérica. *Véase* Números de punto flotante  
 Representación paramétrica, de curvas, 394  
 Residual, 254, 255, 279 (problema de cómputo 7.2.19), 519, 619  
 Residuo, 25  
 Resta, significancia y, 64–67  
 Rojo-negro ordenamiento, 620 (problema 15.3.3)  
 Ruido en computación, 174  
 Secuencias de números quasi-aleatorios, 540  
 Secuencias periódicas de números aleatorios, 535  
 Segunda forma primal, en programación lineal, 663, 664  
 Segundo caso malo, de algoritmo de interpolación cuadrática, 635  
 Semilla, para secuencia de números aleatorios, 534  
 sen  $x$ , periodicidad de, 67  
 Serie binomial, 31 (problema 1.2.1)  
 Serie truncada, 25, 28  
 Series armónicas, 59, 60 (problema de cómputo 2.1.7)  
 Series de Fourier, 73 (problema de cómputo 2.2.15)  
 Series de Maclaurin, 31 (problema 1.2.1), 41  
     (problema de cómputo 1.2.21)  
 Series de Taylor, 20–31, 177 (problema 4.3.19)  
     algoritmo completo de Horner en, 23, 24  
     cálculo de derivadas por, 164–166  
     de  $f$  al punto  $c$ , 22, 23  
     ejemplos de, 20–22  
     en el Teorema del valor medio, 26  
     métodos de Runge-Kutta y, 440, 441  
     para ecuaciones diferenciales ordinarias, 431–435, 466–469  
     para  $F$  en minimización de funciones, 640–642  
     para logaritmo natural ( $\ln$ ), 1  
     precisión de la máquina y, 70 (problema 2.2.28)  
     series alternantes y, 28–30  
     teorema de Taylor en términos de  $(x - c)$  y, 24–26  
     teorema de Taylor en términos de  $h$  y, 27, 28  
 Significancia  
     causada por computadora, 62, 63  
     dígitos significativos en, 3–5, 61  
     evitada en resta, 64–67  
     pérdida de, 61–68  
     reducción de rango y, 67, 68  
     teorema para, 63, 64  
 Simulación, 552–562. *Véase también* Métodos de Monte Carlo  
 Ecuaciones no lineales simultáneas, 104  
     (problema 3.2.39)  
 escudo de neutrones, 557, 558  
 Integral del seno, 189 (problema de cómputo 5.1.2), 204 (problema de cómputo 5.2.5), 463 (problema de cómputo 10.3.15)  
 problema de cumpleaños como, 553–555  
 problema de dos dados como, 556, 557  
 problema de la aguja de Bufón como, 555, 556  
 problema del dado cargado como, 552, 553  
 Simulación de escudo de neutrones, 557, 558  
 Sistema binario, 693, 696, 697. *Véase también* Bases numéricas  
 Sistema hexadecimal, 693, 698. *Véase también* Bases numéricas  
 Sistema octal, 693, 696, 697. *Véase también* Bases numéricas  
 Sistema triangular superior, 248  
 Sistemas de ecuaciones no lineales, 83, 96–99, 104 (problema 3.2.39)  
 Sistemas de posicionamiento global, 111  
     (problema de cómputo 3.2.41)  
 Sistemas en banda de ecuaciones lineales, 280–292  
     bloque pentadiagonal, 285, 286  
     dominio estrictamente diagonal en, 282, 283  
     pentadiagonal, 283–285  
     tridiagonal, 280–282  
 Sistemas incompatibles, 519  
 Sistemas inconsistentes, 519  
 Sistemas ortogonales. *Véase también* Polinomios de Chebyshev  
     algoritmo para, 508–510  
     funciones base ortonormales en, 505–508  
     regresión polinomial en, 510–515  
 Sistemas pentadiagonales de ecuaciones lineales, 280, 283–285  
 Sistemas tridiagonales de ecuaciones lineales, 280–282, 289 (problema de cómputo 7.2.12)  
 Sobrerrelajación, 324, 326, 327, 331, 332  
 Software matemático, 10, 11  
     campos vectoriales, 430  
     comprobación simbólica, 20 (problema de cómputo 1.1.26)  
     desarrollo de, 691  
     descomposición de valor singular, 351  
     ecuaciones diferenciales parciales, 584, 592  
     ecuaciones diferenciales, 427  
     ecuaciones no lineales, 99, 111 (problema de cómputo 3.2.42), 123  
     (problema de cómputo 3.3.19)  
     factorización LU, 308  
     factorizaciones matriciales, 307–309  
     función de error en, 186  
     interpolación polinomial, 153 (problema de cómputo 4.1.11), 164  
     (problema de cómputo 4.2.12)  
     método de potencias para ecuaciones lineales, 363  
     números aleatorios, 533, 535  
     problema con valor en la frontera, 577  
     problemas de minimización, 626  
     programación lineal, 678, 679  
     raíces de ecuaciones, 81, 88 (Problema de cómputo 3.1.12), 93  
     robusto, 269  
     solución mínima, 526  
     splines, 394, 409, 410, 418

- valores propios y vectores propios, 343, 344
- Software robusto, 269
- Solución mínima, a ecuaciones lineales, 524–526
- Soluciones para ecuaciones diferenciales, 426
- Spline B cúbico, 423 (problema 9.3.38)
- Spline cuadrático B, 423 (problema 9.3.37)
- Spline lineal B, 422 (problema 9.3.36)
- Splines cuadráticos, 376–378
- Splines cúbicos periódicos, 387, 401 (problema 9.2.23)
- Splines cúbicos sujetos, 387
- Steffensen, método de, 104 (problema 3.2.36)
- Suavización de datos, 396–398. *Véase también* Polinomios de Chebyshev; Método de mínimos cuadrados
- Subbotin, funciones cuadráticas de splines de, 378–380
- Subdesbordamiento, de rango, 45
- Sugerencias de programación, 684–691
- Sumas inferior y superior, en integración, 181–183
- Supremo (mínimo límite superior), 374
- Sustitución hacia atrás en el algoritmo gaussiano, 248, 250, 251
- Teorema de Cayley-Hamilton, 358 (problema de cómputo 8.2.5)
- Teorema de Gershgorin, 347, 348
- Teorema de invariancia, 135
- Teorema de la aproximación de Weierstrass, 416
- Teorema de Rolle, 156, 157
- Teorema de series alternativas, 28–30, 32 (problema 1.2.13)
- Teorema de Shure, 346
- Teorema del valor medio, 26, 78, 193, 194, 397
- Teorema espectral, 720, 721
- Teorema fundamental del cálculo, 181, 195
- Teoremas
- aproximación de Weierstrass, 416
  - axiomas para un espacio vectorial, 716
  - base ortogonal, 350
  - Cayley-Hamilton, 358 (problema de cómputo 8.2.5)
  - convergencia de Jacobi y Gauss-Seidel, 330
  - cuadratura gaussiana pesada, 232
  - cuadratura gaussiana, 232
  - de error de interpolación polinomial, 156–160
  - de Gershgorin, 347
  - de la existencia de interpolación polinomial, 128
  - de la unicidad del problema con valor inicial, 431
  - de las propiedades de interpolación, 143
  - de los errores de interpolación, 156–160
  - de pérdida de precisión, 63, 64
- de Rolle, 156, 157
- de Shure, 346
- de Taylor, 166
- diferencias divididas y derivadas, 159
- dualidad, 662
- ecuaciones diferenciales lineales, 354
- espectral, 720, 721
- extrapolación de Richardson, 168
- factorización LU, 298
- factorizaciones de Cholesky, 305
- fórmula de Euler-Maclaurin, 208
- fórmula recursiva trapezoidal, 197
- independencia lineal, 718
- integral de Riemann, 183
- invariancia, 135
- localización, 347
- matriz espectral, 349
- método de biseción, 82
- Método de Newton de localización de raíces de ecuaciones, 94
- operaciones largas, 270
- precisión de la regla trapezoidal, 192
- precisión del polinomio de primer grado, 375
- precisión del spline de primer grado, 375
- primera forma primal, 658
- problemas primal y dual, 662
- propiedad recursiva de diferencias divididas, 134
- propiedades de Penrose de seudo inversa, 526
- radio espectral, 329
- segunda forma primal, 6 63
- series alternantes, 28–30, 32 (problema 1.2.13)
- sobrerelajación sucesiva (SRS), 331
- solución mínima, 525
- suavidad de spline cúbico, 397
- SVD mínimos cuadrados, 523
- teorema de Taylor en términos de  $(x - c)$ , 24–26
- teorema de Taylor en términos de  $h$ , 27, 28
- teorema del valor medio para integrales, 193
- teorema espectral para matrices simétricas, 720
- teorema fundamental del cálculo, 181, 195
- valor intermedio, 78, 194
- valor medio, 26, 397
- valores propios de matrices similares, 345
- vértices y vectores columna, 671
- Teoría espectral de matrices, 349–351
- Término de error, 25, 27, 174
- Transpuesta de matrices, 345, 707, 713, 714
- Triángulo de Pascal, 37 (problema de cómputo 1.2.10c)
- Valor redondeado correctamente, 705
- Valores propios y vectores propios, 258 (problema de cómputo 7.1.6), 342–360. *Véase también* Método de potencias para el cálculo de ecuaciones lineales, 343, 344
- descomposición en valor singular de, 348, 349, 351–353
- en álgebra lineal, 719
- en ecuaciones diferenciales lineales, 353–355
- en software matemático, 344
- propiedades de, 345–347
- Teorema de Gershgorin y, 347, 348
- teoría espectral de matrices, 349–351
- Valores singulares, 320
- Vandermonde, matriz de, 139–141, 152 (problema 4.1.47), 254
- Variables, declaración, 685, 686
- Varianza, 15 (problema de cómputo 1.1.7, problema de cómputo 1.1.8)
- Vector escala, 262
- Vector índice, 262, 266
- Vector, normas de, 319, 320, 721
- Vectores. *Véase también* Espacios vectoriales abstractos en álgebra lineal; Valores propios y vectores propios
- A conjugada, 332
  - columna, 671
  - desigualdad vectorial de, 658
  - dirección, 333
  - en álgebra lineal, 706–708
  - en ecuaciones diferenciales ordinarias (EDO), 429–431
  - envoltura convexa de, 417
  - escala, 262
  - gradiente, 640, 641
  - índice, 262, 266
  - producto interno de, 332
  - producto matriz-vector y, 711
  - residual, 254–255, 279 (problema de cómputo 7.2.19)
- Vectores renglón, 706
- Vectores unitarios, 708
- Versión económica de descomposición de valores singulares, 356 (problema 87.2.5)
- Vértices en  $K$ , 671, 672 estimación del volumen. *Véase* Cálculo de área y volumen
- Viga cantiléver, 341 (problema de cómputo 8.1.10)
- Viga de Euler-Bernoulli, 340 (problema de cómputo 8.1.10)

## Fórmulas de cálculo integral

$$\int x^a dx = \frac{x^{a+1}}{(a+1)} + C \quad (a \neq -1)$$

$$\int e^x dx = e^x + C$$

$$\int e^{ax} dx = \frac{1}{a} e^{ax} + C$$

$$\int xe^{ax} dx = \frac{1}{a^2} e^{ax} (ax - 1) + C$$

$$\int x^{-1} dx = \ln|x| + C$$

$$\int \ln x dx = x \ln|x| - x + C$$

$$\int x \ln x dx = \frac{x^2}{2} \ln|x| - \frac{x^2}{4} + C$$

$$\int \frac{dx}{a+bx} = \frac{1}{b} \ln|a+bx| + C$$

$$\int \frac{dx}{(a+bx)^2} = \frac{-1}{b(a+bx)} + C$$

$$\int \frac{dx}{x(ax+b)} = \frac{1}{b} \ln \left| \frac{x}{ax+b} \right| + C$$

$$\int \frac{dx}{a+bx^2} = \frac{1}{\sqrt{ab}} \arctan \left( \frac{1}{a} x \sqrt{ab} \right) + C$$

$$\int \frac{dx}{a^2+x^2} = \frac{1}{a} \arctan \left( \frac{x}{a} \right) + C \quad (a \neq 1)$$

$$\int \frac{dx}{\sqrt{a^2-x^2}} = \arcsen \left( \frac{x}{a} \right) + C \quad (a \neq 1)$$

$$\int \frac{1}{\sqrt{x^2+a^2}} dx = \ln \left| \sqrt{x^2+a^2} + x \right| + C$$

$$\int \sqrt{x^2 \pm a^2} dx = \frac{x}{2} \sqrt{x^2 \pm a^2} \pm \frac{a^2}{2} \ln \left| x + \sqrt{x^2 \pm a^2} \right| + C$$

$$\int \operatorname{sen} x dx = -\cos x + C$$

## Teorema fundamental de cálculo

$$\frac{d}{dx} \int_a^x f(t) dt = f(x)$$

## Valor medio para integrales

$$\int_a^b f(x)g(x) dx = f(\xi) \int_a^b g(x) dx \quad (g(x) \geq 0)$$

$$\int \cos x dx = \operatorname{sen} x + C$$

$$\int \tan x dx = \ln|\sec x| + C$$

$$\int \sec x dx = \ln|\sec x + \tan x| + C$$

$$\int x \operatorname{sen} x dx = \operatorname{sen} x - x \cos x + C$$

$$\int \sec^2 x dx = \tan x + C$$

$$\int \sec x \tan x dx = \sec x + C$$

$$\int \operatorname{senh} x dx = \cosh x + C$$

$$\int \cosh x dx = \operatorname{senh} x + C$$

$$\int \tanh x dx = \ln|\cosh x| + C$$

$$\int \coth x dx = \ln|\operatorname{senh} x| + C$$

$$\int \operatorname{sen}^2 x dx = \frac{x}{2} - \frac{1}{4} \operatorname{sen} 2x + C$$

$$\int \cos^2 x dx = \frac{x}{2} + \frac{1}{4} \operatorname{sen} 2x + C$$

$$\int \operatorname{arc sen} x dx = x \operatorname{arc sen} x + \sqrt{1-x^2} + C$$

$$\int \operatorname{arccos} x dx = x \operatorname{arccos} x - \sqrt{1-x^2} + C$$

$$\int \operatorname{arctan} x dx = x \operatorname{arctan} x - \frac{1}{2} \ln(1+x^2) + C$$

$$\int F'(g(x))g'(x) dx = F(g(x)) + C$$

## Integración por partes

$$\int u dv = uv - \int v du$$

## Series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \frac{x^6}{6!} + \cdots = \sum_{k=0}^{\infty} \frac{x^k}{k!} \quad (|x| < \infty)$$

$$a^x = 1 + x \ln a + \frac{(x \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + \cdots = \sum_{k=0}^{\infty} \frac{(x \ln a)^k}{k!} \quad (|x| < \infty)$$

$$\operatorname{sen} x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} \quad (|x| < \infty)$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} \quad (|x| < \infty)$$

$$\tan x = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \frac{62x^9}{2835} + \cdots \quad \left( x^2 < \frac{\pi^2}{4} \right)$$

$$\arcsen x = x + \frac{x^3}{3!} + \frac{1}{2} \frac{3}{5} \frac{x^5}{5!} + \frac{1}{2} \frac{3}{4} \frac{5}{6} \frac{x^7}{7!} + \cdots \quad (x^2 < 1)$$

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)} \quad (x^2 < 1)$$

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{x^k}{k} \quad (-1 < x \leq 1)$$

$$\ln\left(\frac{1+x}{1-x}\right) = 2 \left[ x + \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots \right] = 2 \sum_{k=1}^{\infty} \frac{x^{2k-1}}{2k-1} \quad (|x| < 1)$$

$$(x+y)^n = x^n + nx^{n-1}y + \frac{n(n-1)}{2!}x^{n-2}y^2 + \frac{n(n-1)(n-2)}{3!}x^{n-3}y^3 + \cdots = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k$$

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + x^5 + \cdots = \sum_{k=0}^{\infty} x^k \quad (|x| < 1)$$

## Serie de Taylor formal para $f$ con respecto a $c$

$$f(x) \sim f(c) + f'(c)(x-c) + \frac{f''(c)}{2!}(x-c)^2 + \frac{f'''(c)}{3!}(x-c)^3 + \cdots = \sum_{k=0}^{\infty} \frac{f^{(k)}(c)}{k!}(x-c)^k$$

## Serie de Taylor para $f(x)$

$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(c)}{k!}(x-c)^k + E_{n+1} \quad \text{donde } E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-c)^{n+1}$$

## Serie de Taylor para $f(x+h)$

$$f(x+h) = \sum_{k=0}^n \frac{f^{(k)}(x)}{k!} h^k + E_{n+1} \quad \text{donde } E_{n+1} = \frac{f^{(n+1)}(\xi)}{(n+1)!} h^{n+1}$$

## Series alternantes

Si  $a_1 \geq a_2 \geq \cdots \geq a_n \geq \cdots \geq 0$  para toda  $n$  y  $\lim_{n \rightarrow \infty} a_n = 0$  entonces

$$\sum_{k=1}^{\infty} (-1)^{k-1} a_k = \lim_{n \rightarrow \infty} \sum_{k=1}^n (-1)^{k-1} a_k = \lim_{n \rightarrow \infty} S_n = S. \quad \text{Además, } |S - S_n| \leq a_{n+1} \text{ para toda } n.$$

## Teorema del valor medio

$$f(b) = f(a) + (b-a)f'(\xi) \quad \text{para alguna } \xi \text{ en } (a, b)$$





# Métodos numéricos y computación

Sexta Edición

Ward Cheney • David Kincaid

Los autores Ward Cheney y David Kincaid muestran a los estudiantes de ciencias e ingenierías el potencial que las computadoras tienen para solucionar problemas numéricos y les dan oportunidades amplias de afinar sus habilidades en la programación y solución de problema. El texto también ayuda a los estudiantes a aprender sobre los errores que inevitablemente acompañan los cálculos científicos y los dota de los métodos para detectar, predecir y controlar estos errores.

*Características importantes:*

- ▶ Más accesible: Los códigos de computadora y otros materiales ahora se incluyen en la web del texto dándole al profesor y a sus estudiantes fácil acceso sin el aburrido mecanografiar. Los códigos de computadora de Matlab, de Mathematica, y de Maple y la “descripción del software matemático” en el apéndice ahora son todos accesibles en línea.
- ▶ El aprender de la representación visual: Porque los códigos concretos y las ayudas visuales son provechosos a cada lector, los autores han agregado aún más figuras y ejemplos numéricos a través del texto, asegurando a estudiantes la comprensión sólida antes de avanzar a los nuevos temas.
- ▶ Accesible y actualizado: Totalmente actualizada, la nueva edición incluye nuevas secciones y material en temas tales como el método de la posición falsa, el método conjugado del gradiente, el método de Simpson y más.
- ▶ Aplicaciones a la mano: Da a los estudiantes oportunidades innumerables de poner los conceptos del capítulo en práctica verdadera, los ejercicios de aplicación adicionales se presentan a través del libro.
- ▶ Referencias actualizadas: Las citas a referencias recientes reflejan los últimos progresos en el área.
- ▶ Nuevos apéndices: Los apéndices reorganizados y mejorados ofrecen una abundancia del material suplementario, incluyendo consejos sobre buenas prácticas de programación, la cobertura de números en diversas bases, los detalles en la aritmética de punto flotante de IEEE y las discusiones de los conceptos lineales y de la notación del álgebra.

