

STEVEN C.
CHAPRA



MÉTODOS NUMÉRICOS APLICADOS COM MATLAB® PARA ENGENHEIROS E CIENTISTAS

3^a EDIÇÃO





C467m Chapra, Steven C.

Métodos numéricos aplicados com MATLAB para engenheiros e cientistas [recurso eletrônico] / Steven C. Chapra ; tradução: Rafael Silva Alípio ; revisão técnica: Antonio Pertence Júnior. – 3. ed. – Dados eletrônicos. – Porto Alegre : AMGH, 2013.

Editedo também como livro impresso em 2013.
ISBN 978-85-8055-177-8

1. Engenharia – MATLAB. 2. Computação – Programa – MATLAB. I. Título.

CDU 62:004.4MATLAB

Catalogação na publicação: Ana Paula M. Magnus – CRB 10/2052

Steven C. Chapra

Tufts University

MÉTODOS NUMÉRICOS APLICADOS COM MATLAB® PARA ENGENHEIROS E CIENTISTAS

3^a EDIÇÃO

Tradução

Rafael Silva Alípio

Engenheiro Eletricista pelo Centro Federal de Educação

Tecnológica de Minas Gerais (CEFET/MG)

Licenciado em Física pela Universidade Federal de Minas Gerais (UFMG)

Mestre em Modelagem Matemática e Computacional

pelo Centro Federal de Educação Tecnológica de Minas Gerais (CEFET/MG)

Revisão técnica

Antonio Pertence Júnior

Engenheiro Eletrônico e de Telecomunicações pela IPUC/MG

Mestre em Engenharia pelo DEMEC/UFMG

Membro do MATLAB ACCESS GROUP desde 1993

Professor da Faculdade de Engenharia da FUMEC/MG

Versão impressa
desta obra: 2013



AMGH Editora Ltda.

2013

Obra originalmente publicada sob o título
Applied Numerical Methods with MATLAB for Engineers and Scientists, 3rd Edition.
ISBN 0073401102/9780073401102

Original edition copyright © 2012, The McGraw-Hill Companies Inc., New York, New York 10020.
All rights reserved.

Gerente editorial: *Arysinha Jacques Affonso*

Colaboraram nesta edição:

Editora: *Viviane R. Nepomuceno*

Assistente editorial: *Caroline L. Silva*

Capa: *MSDE/Manu Santos Design*

Leitura final: *Cristiane Silva Trindade e Mariana Zanini*

Editoração: *Roberta Pereira de Paula*

Reservados todos os direitos de publicação, em língua portuguesa, à
AMGH Editora Ltda., uma parceria entre GRUPO A EDUCAÇÃO S.A e McGRAW-HILL EDUCATION.
Av. Jerônimo de Ornelas, 670 – Santana
90040-340 – Porto Alegre – RS
Fone: (51) 3027-7000 Fax: (51) 3027-7070

É proibida a duplicação ou reprodução deste volume, no todo ou em parte, sob quaisquer
formas ou por quaisquer meios (eletrônico, mecânico, gravação, fotocópia, distribuição na Web
e outros), sem permissão expressa da Editora.

Unidade São Paulo
Av. Embaixador Macedo Soares, 10.735 – Pavilhão 5 – Cond. Espace Center
Vila Anastácio – 05095-035 – São Paulo – SP
Fone: (11) 3665-1100 Fax: (11) 3667-1333

SAC 0800 703-3444 – www.grupoa.com.br

Para
meus irmãos,
John e Bob Chapra

O autor

Steven Chapra leciona no Departamento de Engenharia Civil e Ambiental na Tufts University, onde ocupa a posição de Louis Berger Chair em Computação e Engenharia. Entre seus outros livros, estão *Numerical Methods for Engineers* e *Surface Water-Quality Modeling*.

Chapra recebeu diplomas de engenharia do Manhattan College e da University of Michigan. Antes de se juntar ao corpo docente da Tufts, trabalhou para a Agência de Proteção Ambiental e para a Administração Nacional Atmosférica e Oceânica e lecionou na Texas A&M University e na University of Colorado. Seus interesses gerais de pesquisa se concentram na modelagem da qualidade da água superficial e em aplicações computacionais avançadas na engenharia ambiental.

Ele recebeu diversos prêmios por suas contribuições acadêmicas, incluindo a medalha Rudolph Hering, o Prêmio Merriam-Wiley, como autor notável, e o Prêmio Chandler-Misener. Também foi reconhecido como professor em destaque entre o corpo docente tanto da Texas A&M University (Prêmio Tenneco de 1986) quanto da University of Colorado (Prêmio Hutchinson de 1992).

Steven foi originalmente atraído para engenharia ambiental e ciência por causa de seu amor pela vida ao ar livre. Ele é um pescador ávido e alpinista. Um “nerd” sem remorsos, seu caso de amor com a computação começou quando ele foi introduzido pela primeira vez à programação em Fortran na graduação, em 1966. Hoje, ele se sente abençoado por poder fundir seu amor pela matemática, ciência e computação com sua paixão pelo meio ambiente natural. Além disso, ele recebe o bônus de compartilhá-lo com os outros através de seu ensino e da escrita!

Além de seus interesses profissionais, ele gosta de arte, música (especialmente música clássica, jazz e bluegrass), e ler histórias. Apesar dos rumores infundados, ele nunca praticou nem praticará voluntariamente *bungee jumping* ou parapente.

Se você quiser entrar em contato com Steven, ou saber mais sobre ele, visite sua página pessoal em <http://engineering.tufts.edu/cee/people/chapra/> ou envie um e-mail para ele em steven.chapra@tufts.edu.

Prefácio

Este livro foi projetado para auxiliar um curso de métodos numéricos com duração de um semestre, e foi escrito para estudantes que querem aprender e aplicar métodos numéricos para resolver problemas em engenharia e ciência. Como tal, os métodos são motivados por problemas e não por matemática. Exposto isso, teoria suficiente é fornecida de modo que os estudantes obtenham uma visão das técnicas numéricas e de suas principais deficiências.

O MATLAB® fornece um ótimo ambiente para um curso desse tipo. Embora outros ambientes (por exemplo, Excel/VBA, Mathcad) ou linguagens (por exemplo, Fortran 90, C++) pudesse ter sido escolhidos, o MATLAB oferece atualmente uma boa combinação de recursos de programação úteis e práticos com recursos numéricos nativos poderosos. Por um lado, seu ambiente de programação, os arquivos-M, permite que os estudantes implementem algoritmos moderadamente complicados de maneira estruturada e coerente; por outro, seus recursos numéricos nativos capacitam os estudantes a resolverem problemas mais difíceis, sem que seja preciso “reinventar a roda”.

O conteúdo básico, a organização e a pedagogia da segunda edição estão essencialmente preservados nesta terceira edição. O estilo de escrita simples foi intencionalmente mantido com o objetivo de facilitar a leitura do livro, que tenta falar diretamente com o leitor e é concebido, em parte, para ser uma ferramenta de autoaprendizado.

Desse modo, esta edição difere da anterior em três aspectos principais: (1) dois novos capítulos, (2) várias seções novas, e (3) problemas revisados.

1. Novos capítulos. Desenvolvi novos capítulos para esta edição, cuja inclusão foi principalmente motivada por minha experiência em sala de aula. Isto é, eles foram incluídos porque funcionam bem no curso de métodos numéricos para graduação que eu leciono na Tufts University. Esse curso geralmente abrange todas as áreas da engenharia e alcança desde estudantes que estão cursando o segundo ano da faculdade até aqueles que estão no último ano, sendo a maioria no nível inicial. Além disso, também, atraímos alguns poucos alunos dos cursos de matemática e ciências. Os dois novos capítulos são:

- ✓ **Autovalores.** Quando escrevi este livro pela primeira vez, considerei que este tema poderia ser considerado um tópico “avançado”. Portanto, apresentei o material desse tópico no fim do semestre e o abordei no livro como um apêndice. Essa sequência tinha a vantagem acessória de que o assunto poderia ser parcialmente motivado pelo papel dos autovalores na solução de sistemas lineares de EDOs. Nos últimos anos, eu comecei a deslocar esse material para onde considero ser sua posição matemática mais natural: no final da seção de equações algébricas lineares. Enfatizando as aplicações (especificamente, o uso de autovalores para o estudo

de vibrações), descobri que os estudantes respondem muito positivamente ao tema com essa mudança. Além disso, esse deslocamento permite que eu retorne ao tópico em capítulos subsequentes, o que serve para aumentar a apreciação do tópico pelos estudantes.

- ✓ **Análise de Fourier.** Nos últimos anos, se o tempo permitisse, eu normalmente também apresentava uma palestra no final do semestre sobre análise de Fourier. Ao longo dos últimos dois anos, comecei a apresentar esse material em sua posição mais natural, logo após o tópico de mínimos quadrados linear. Eu motivo o assunto pela utilização da abordagem de mínimos quadrados linear para ajustar senoides aos dados. Então, pela ênfase em aplicações (novamente vibrações), descobri que os estudantes imediatamente absorvem o tópico e apreciam seu valor dentro da engenharia e da ciência. Deve-se enfatizar que ambos os capítulos são escritos de maneira modular e poderiam ser omitidos sem prejuízo para o ciclo pedagógico do curso. Portanto, se o professor escolher, pode omiti-los ou deslocá-los para o final do semestre. Em todo caso, eu não teria incluído esses capítulos na presente edição se eles não representassem uma melhoria dentro de minha experiência atual em sala de aula. Em particular, com base em minhas avaliações de ensino, percebi que os estudantes mais motivados realmente veem esses tópicos como destaques. Isso é especialmente verdadeiro, porque o MATLAB facilita muito sua aplicação e interpretação.
- 2. **Novo conteúdo.** Além dos capítulos novos, foram incluídas seções novas e melhoradas em diversos tópicos. Os acréscimos principais incluem seções sobre animação (Capítulo 3), método de Brent para localização de raízes (Capítulo 6), decomposição LU com pivotamento (Capítulo 10), *números aleatórios* e *simulação de Monte Carlo* (Capítulo 14), *quadratura adaptativa* (Capítulo 20) e término de eventos (opção *events* do MATLAB) de EDOs (Capítulo 23).
- 3. **Novos problemas.** A maioria dos problemas no final dos capítulos foi modificada e uma série de novos problemas foi adicionada. Um esforço foi feito para incluir vários problemas novos em cada capítulo que são mais desafiadores e difíceis do que os problemas da edição anterior.

Com exceção do material e problemas novos, a terceira edição é muito similar à segunda. Em particular, eu me esforcei para manter a maioria das características que contribuem para a sua eficácia pedagógica, incluindo o uso extensivo de exemplos resolvidos e aplicações de engenharia e ciência. Como na edição anterior, eu fiz um esforço concentrado para tornar o livro “amigável ao estudante” tanto quanto possível. Assim, tentei manter minhas explicações simples e práticas.

Embora meu principal interesse seja capacitar estudantes proporcionando-lhes uma boa introdução à resolução de problemas numéricos, eu tenho o objetivo complementar de tornar essa introdução estimulante e prazerosa. Acredito que estudantes motivados, que gostam de engenharia e ciência, resolução de problemas, matemática e também programação, acabarão por se tornar melhores profissionais. Se o meu livro promover entusiasmo e apreço por esses assuntos, considerarei que meu esforço obteve sucesso.

Agradecimentos. Vários membros da McGraw-Hill contribuíram para esse projeto. Agradecimentos especiais são direcionados a Lorraine Buczek, Bill Stenquist e Melissa Leick pelo incentivo, apoio e orientação. Ruma Khurana da MPS Limitada, uma empresa da Macmillan, também fez um excelente trabalho na fase de produção final do livro. Por último, mas não menos importante, Beatrice Sussman mais uma vez demonstrou porque é a melhor editora de texto do ramo.

No decorrer deste projeto, o pessoal da MathWorks Inc., demonstrou sua excelência geral, bem como o seu forte compromisso com a educação em engenharia e ciência. Em particular, Courtney Esposito e Naomi Fernandes do Programa do Livro da MathWorks Inc., foram especialmente prestativas.

A generosidade da família Berger, e em particular Fred Berger, proporcionou-me a oportunidade de trabalhar em projetos criativos como este livro tratando de computação e engenharia. Além disso, meus colegas na Escola de Engenharia em Tufts, especialmente Masoud Sanaye, Lew Edgers, Vince Manno, Luis Dorfmann, Rob White, Linda Abriola e Laurie Baise, foram muito solidários e prestativos.

Sugestões significativas também foram dadas por uma série de colegas. Em particular, Dave Clough (University of Colorado – Boulder) e Mike Gustafson (Duke University) contribuíram com ideias e sugestões valiosas. Além disso, vários revisores forneceram *feedback* e conselhos úteis, incluindo Karen Dow Ambtman (University of Alberta), Jalal Behzadi (Shahid Chamran University), Eric Cochran (Iowa State University), Frederic Gibou (University of California at Santa Barbara), Jane Grande-Allen (Rice University), Raphael Haftka (University of Florida), Scott Hendricks (Virginia Tech University), Ming Huang (University of San Diego), Oleg Igoshin (Rice University), David Jack (Baylor University), Clare McCabe (Vanderbilt University), Eckart Meiburg (University of California at Santa Barbara), Luis Ricardez (University of Waterloo), James Rottman (University of California, San Diego), Bingjing Su (University of Cincinnati), Chin-An Tan (Wayne State University), Joseph Tipton (The University of Evansville), Marion W. Vance (Arizona State University), Jonathan Vande Geest (University of Arizona) e Leah J. Walker (Arkansas State University).

Ressalto que, embora eu tenha recebido conselhos úteis das pessoas aqui mencionadas, sou responsável por eventuais imprecisões ou erros que você possa encontrar neste livro. Por favor, entre em contato comigo pelo e-mail steven.chapra@tufts.edu, se você detectar quaisquer erros.

Por fim, quero agradecer a minha família, e em particular a minha esposa, Cynthia, pelo amor, paciência e apoio que tem prestado todo o tempo que passei neste projeto.

Steven C. Chapra

Medford, Massachusetts

FERRAMENTAS PEDAGÓGICAS

Teoria apresentada como forma de conceitos-chave. O texto é destinado a usuários de métodos numéricos, não desenvolvedores. Portanto, teoria não é incluída “por amor à teoria”, por exemplo, não são incluídas provas demasiadamente complicadas. A teoria é incluída à medida em que ela informa conceitos-chave, como a série de Taylor, a convergência, o condicionamento, etc. Assim, mostra-se ao estudante como a teoria se conecta com questões práticas na resolução de problemas.

Material introdutório de MATLAB. O texto inclui dois capítulos introdutórios sobre como usar o MATLAB. O Capítulo 2 mostra aos estudantes como realizar cálculos e criar gráficos no modo de comando padrão do MATLAB. O Capítulo 3 fornece uma cartilha sobre o desenvolvimento de programas numéricos via funções/arquivos-M do MATLAB. Dessa forma, o texto fornece aos estudantes os meios para desenvolverem seus próprios algoritmos numéricos, bem como para explorar as poderosas rotinas nativas do MATLAB.

Algoritmos apresentados usando arquivos-M do MATLAB. Em vez de usar pseudocódigos, este livro apresenta algoritmos assim como arquivos-M bem estruturados do MATLAB. Além de serem programas de computador úteis, estes proporcionam aos estudantes modelos para os seus próprios arquivos-M, que eles desenvolverão como exercícios para casa.

Exemplos resolvidos e estudos de caso. Extensos exemplos resolvidos são desenvolvidos em detalhes, para que os alunos possam seguir claramente os passos em cada cálculo numérico. Os estudos de caso consistem em aplicações de engenharia e ciência que são mais complexas e ricas do que os exemplos resolvidos. Eles são colocados no final de capítulos selecionados, com a intenção de (1) ilustrar as nuances dos métodos, e (2) mostrar de forma mais realista como os métodos combinados com o MATLAB são aplicados para a resolução de problemas.

Conjunto de problemas. O texto inclui uma ampla variedade de problemas. Muitos são provenientes de disciplinas científicas e de engenharia. Outros são usados para ilustrar técnicas numéricas e conceitos teóricos. Os problemas incluem aqueles que podem ser resolvidos com uma calculadora de bolso, bem como outros que necessitam de solução computacional com o MATLAB.

Apêndices e índices úteis. O Apêndice A contém comandos do MATLAB, e o Apêndice B contém as funções do MATLAB desenvolvidas ao longo do livro.

Material de apoio. No site do Grupo A, www.grupoa.com.br, há vários materiais de apoio ao estudo, incluindo biblioteca de imagens, em português, apresentações em Power-Point, arquivos-M e recursos adicionais do MATLAB, em inglês.

Sumário

Parte 1 Modelagem, computadores e análise de erros

1.1 Motivação	1
1.2 Organização da Parte 1	2

Capítulo 1 Modelagem matemática, métodos numéricos e solução

de problemas	4
--------------------	---

1.1 Um modelo matemático simples	5
1.2 Leis de conservação e engenharia	11
1.3 Métodos numéricos abordados neste livro	14
1.4 Estudo de caso: Força de arraste real	16
Problemas	19

Capítulo 2 Princípios básicos do MATLAB

2.1 O ambiente do MATLAB	25
2.2 Atribuição	26
2.3 Operações matemáticas	31
2.4 Utilização de funções nativas	35
2.5 Gráficos	37
2.6 Outros recursos	41
2.7 Estudo de caso: Análise exploratória de dados	41
Problemas	44

Capítulo 3 Programação com MATLAB

3.1 Arquivos-M	49
3.2 Entrada e saída	53
3.3 Programação estruturada	57
3.4 Aninhamento e indentação	71
3.5 Passando funções para arquivos-M	74
3.6 Estudo de caso: Velocidade do saltador de <i>bungee jumping</i>	79
Problemas	82

Capítulo 4	Erros de arredondamento e de truncamento.....	88
4.1	Erros	89
4.2	Erros de arredondamento	95
4.3	Erros de truncamento	103
4.4	Erro numérico total	115
4.5	Enganos, erros de formulação e incerteza nos dados.....	119
	Problemas	121

Parte 2 Raízes e otimização

2.1	Visão geral.....	123
2.2	Organização da Parte 2	124

Capítulo 5	Raízes: métodos intervalares.....	126
5.1	Raízes em engenharia e ciência	127
5.2	Métodos gráficos	128
5.3	Métodos intervalares e estimativas iniciais.....	129
5.4	O método da bissecção.....	134
5.5	O método da falsa posição	140
5.6	Estudo de caso: Gases de efeito estufa e água da chuva	144
	Problemas	147

Capítulo 6	Raízes: métodos abertos.....	151
6.1	Iteração de ponto fixo simples.....	152
6.2	O método de Newton-Raphson.....	156
6.3	O método da secante	161
6.4	Método de Brent	163
6.5	Função do MATLAB: fzero	168
6.6	Polinômios	170
6.7	Estudo de caso: Atrito da tubulação	173
	Problemas	177

Capítulo 7	Otimização.....	182
7.1	Introdução e contextualização	183
7.2	Otimização unidimensional	186
7.3	Otimização multidimensional	195
7.4	Estudo de caso: Equilíbrio e energia potencial mínima.....	197
	Problemas	199

Parte 3 Sistemas lineares

3.1	Visão geral.....	205
3.2	Organização da Parte 3	207

Capítulo 8	Equações algébricas lineares e matrizes	209
8.1	Álgebra matricial: uma visão geral.....	210
8.2	Resolvendo equações algébricas lineares com o MATLAB	220
8.3	Estudo de caso: Correntes e tensões em circuitos	222
Problemas		226
Capítulo 9	Eliminação de Gauss	229
9.1	Resolução de um número pequeno de equações	230
9.2	Eliminação de Gauss ingênua.....	235
9.3	Pivotamento.....	242
9.4	Sistemas tridiagonais	245
9.5	Estudo de caso: Modelo de uma haste aquecida.....	248
Problemas		251
Capítulo 10	Decomposição LU	254
10.1	Uma visão geral da decomposição LU.....	255
10.2	A versão da eliminação de Gauss por decomposição LU.....	256
10.3	Decomposição de Cholesky	263
10.4	Divisão à esquerda do MATLAB	266
Problemas		267
Capítulo 11	Matriz inversa e condicionamento	268
11.1	A matriz inversa.....	268
11.2	Análise de erro e condicionamento do sistema	272
11.3	Estudo de caso: Poluição do ar de interiores	277
Problemas		280
Capítulo 12	Métodos iterativos	284
12.1	Sistemas lineares: Gauss-Seidel.....	284
12.2	Sistemas não lineares	291
12.3	Estudo de caso: Reações químicas.....	298
Problemas		301
Capítulo 13	Autovalores	303
13.1	Fundamentos matemáticos.....	304
13.2	Fundamentos físicos.....	308
13.3	O método da potência.....	310
13.4	Função do MATLAB: <code>eig</code>	313
13.5	Estudo de caso: Autovalores e terremotos	314
Problemas		317

Parte 4 Ajuste de curvas

4.1 Visão geral.....	321
4.2 Organização da Parte 4	323
 Capítulo 14 Regressão linear	324
14.1 Revisão de estatística	325
14.2 Números aleatórios e simulação	332
14.3 Regressão linear por mínimos quadrados	336
14.4 Linearização de relações não lineares	344
14.5 Aplicações computacionais.....	348
14.6 Estudo de caso: Cinética enzimática.....	351
Problemas	355
 Capítulo 15 Mínimos quadrados: modelo linear geral e regressão não linear.....	361
15.1 Regressão polinomial.....	361
15.2 Regressão linear múltipla.....	365
15.3 Mínimos quadrados linear geral	367
15.4 Decomposição QR e operador barra invertida	370
15.5 Regressão não linear	371
15.6 Estudo de caso: Ajustando dados experimentais.....	373
Problemas	376
 Capítulo 16 Análise de Fourier.....	380
16.1 Ajuste de curvas com funções senoidais	381
16.2 Série de Fourier contínua	387
16.3 Domínios do tempo e da frequência	390
16.4 Integral e transformada de Fourier	393
16.5 Transformada de Fourier discreta (TFD).....	394
16.6 O espectro de potência	399
16.7 Estudo de caso: Manchas solares.....	400
Problemas	402
 Capítulo 17 Interpolação polinomial	404
17.1 Introdução à interpolação	405
17.2 Polinômios interpoladores de Newton.....	408
17.3 Polinômios interpoladores de Lagrange	416
17.4 Interpolação inversa	419
17.5 Extrapolação e oscilações	420
Problemas	424

Capítulo 18	Interpolação por splines e por partes	428
18.1	Introdução aos splines.....	428
18.2	Splines lineares	430
18.3	Splines quadráticos	434
18.4	Splines cúbicos	437
18.5	Interpolação por partes no MATLAB	444
18.6	Interpolação multidimensional	448
18.7	Estudo de caso: Transferência de calor.....	451
	Problemas	455

Parte 5 Integração e derivação

5.1	Visão geral.....	459
5.2	Organização da Parte 5	460

Capítulo 19	Fórmulas de integração numérica.....	462
19.1	Introdução e contextualização	463
19.2	Fórmulas de Newton-Cotes.....	466
19.3	A regra do trapézio.....	468
19.4	Regras de Simpson	474
19.5	Fórmulas de Newton-Cotes de ordem superior	480
19.6	Integração com segmentos desiguais.....	481
19.7	Fórmulas de integração abertas	485
19.8	Integrais múltiplas.....	486
19.9	Estudo de caso: Integração numérica para calcular o trabalho	488
	Problemas	492

Capítulo 20	Integração numérica de funções	496
20.1	Introdução	496
20.2	Integração de Romberg	497
20.3	Quadratura de Gauss.....	502
20.4	Quadratura adaptativa.....	509
20.5	Estudo de caso: Valor médio quadrático da corrente	513
	Problemas	517

Capítulo 21	Derivação numérica	521
21.1	Introdução e contextualização	522
21.2	Fórmulas de derivação de alta acurácia	525
21.3	Extrapolação de Richardson	528
21.4	Derivadas de dados desigualmente espaçados	530
21.5	Derivadas e integrais para dados com erros.....	532

21.6 Derivadas parciais.....	533
21.7 Derivação numérica com o MATLAB	533
21.8 Estudo de caso: Visualização de campos.....	538
Problemas	540

Parte 6 Equações diferenciais ordinárias

6.1 Visão geral.....	547
6.2 Organização da Parte 6	551

Capítulo 22 Problemas de valor inicial.....

22.1 Visão geral.....	555
22.2 Método de Euler	555
22.3 Melhorias no método de Euler	561
22.4 Métodos de Runge-Kutta	567
22.5 Sistemas de equações.....	572
22.6 Estudo de caso: Modelos predador-presa e caos.....	578
Problemas	583

Capítulo 23 Métodos adaptativos e sistemas rígidos.....

23.1 Métodos adaptativos de Runge-Kutta.....	588
23.2 Métodos de passo múltiplo.....	597
23.3 Rigidez.....	602
23.4 Aplicação do MATLAB: saltador de <i>bungee jumping</i> com corda	607
23.5 Estudo de caso: Fonte intermitente de Plínio	609
Problemas	613

Capítulo 24 Problemas de valor de contorno

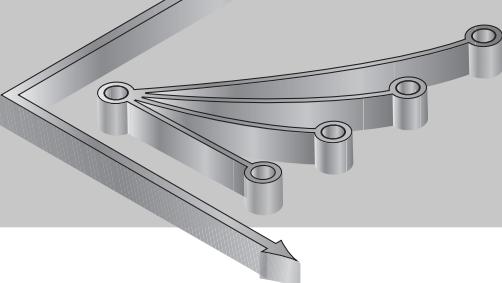
24.1 Introdução e contextualização	617
24.2 O método <i>shooting</i>	621
24.3 Métodos de diferenças finitas	628
Problemas	635

Referências.....

Apêndice A: Funções nativas do MATLAB.....	643
--	-----

Apêndice B: Funções desenvolvidas no MATLAB.....	645
--	-----

Índice.....	647
-------------	-----



Parte 1

Modelagem, computadores e análise de erros

1.1 MOTIVAÇÃO

O que são métodos numéricos e por que devemos estudá-los?

Métodos numéricos são técnicas pelas quais os problemas matemáticos são formulados de modo que possam ser resolvidos com operações lógicas e aritméticas. Como os computadores digitais se sobressaem na execução de tais operações, os métodos numéricos são, às vezes, referidos de forma mais ampla como *matemática computacional*.

Na era pré-computacional, a implementação das operações citadas, além de ser entediante, demandava muito tempo (isso logo ficará claro para o leitor), o que limitava seriamente a utilização prática dos métodos numéricos. No entanto, com o advento dos computadores e a um custo financeiro acessível, o papel dos métodos numéricos na solução de problemas de engenharia e de ciência de forma geral aumentou significativamente. Como eles figuram de forma tão proeminente em grande parte do nosso trabalho, que os métodos numéricos devem ser parte da educação básica de todo engenheiro e cientista. Assim como todos nós devemos ter bases sólidas em outras áreas da matemática e da ciência, também devemos ter uma compreensão básica dos métodos numéricos. Em particular, devemos ter uma noção sólida de suas capacidades e limitações.

Além das contribuições para a educação geral, existem diversas razões adicionais pelas quais os métodos numéricos devem ser estudados:

1. Os métodos numéricos são ferramentas extremamente poderosas na resolução de problemas. Eles são capazes de lidar com um grande número de equações, não linearidades e geometrias complicadas recorrentes na prática de engenharia e que, em geral, são impossíveis de resolver de maneira analítica. Dessa forma, eles aumentam muito a capacidade de resolver problemas.
2. Durante a carreira, o profissional de engenharia frequentemente terá oportunidade de usar softwares disponíveis no mercado, ou pacotes “prontos” de software, que envolvem métodos numéricos. O uso inteligente desses programas depende do conhecimento da teoria básica fundamental dos métodos; sem esse conhecimento, o profissional utiliza um software comercial como uma “caixa preta”, com pouca visão crítica de seu funcionamento interno ou da validade dos resultados produzidos.

3. Muitos problemas não podem ser solucionados utilizando-se pacotes “prontos” de softwares. Se o engenheiro estiver familiarizado com métodos numéricos e souber programar no computador, poderá projetar seu próprio programa para resolver problemas sem precisar comprar ou contratar softwares caros.
4. Os métodos numéricos são um meio eficiente para o aprendizado de programação, já que uma forma eficiente de aprendê-la é justamente escrever um programa de computador. Como os métodos numéricos são, na maior parte, projetados para implementação em computadores, eles se mostram ideais para esse propósito; além disso, são especialmente adequados para ilustrar o poder e as limitações dos computadores. Quando se implementam com sucesso métodos numéricos em um computador aplicando-os na solução de problemas insolúveis de outra forma, tem-se acesso a uma demonstração impressionante de como os computadores podem auxiliar o desenvolvimento profissional. Ao mesmo tempo, aprende-se a identificar e a controlar os erros das aproximações, que são parte essencial de cálculos numéricos em grande escala.
5. Os métodos numéricos auxiliam o profissional a reforçar seu entendimento da matemática. Como uma função dos métodos numéricos é reduzir a matemática mais avançada a operações aritméticas básicas, eles alcançam detalhes práticos de alguns tópicos que, de outra forma, seriam incompreensíveis. Como resultado dessa perspectiva alternativa, podemos apontar o aprimoramento da compreensão e da percepção.

Com essas razões como motivação, podemos agora começar a entender como métodos numéricos e computadores digitais trabalham em conjunto para gerar soluções confiáveis para os problemas matemáticos. O restante deste livro é dedicado a essa tarefa.

1.2 ORGANIZAÇÃO DA PARTE 1

Este livro está dividido em seis partes. As últimas cinco partes focam nas principais áreas de métodos numéricos. Embora seja tentador pular diretamente para esse material, a Parte 1 é composta por quatro capítulos que abordam um material de apoio essencial para as demais partes do livro.

O Capítulo 1 fornece um exemplo concreto de como um método numérico pode ser empregado para resolver um problema real. Para fazer isso, é desenvolvido um *modelo matemático* de um saltador de *bungee jumping* em queda livre. O modelo, que é baseado na segunda lei de Newton, resulta em uma equação diferencial ordinária. Depois de obter uma solução na forma fechada a partir da aplicação do cálculo tradicional, é mostrado como uma solução comparável pode ser gerada com um método numérico simples. O capítulo é finalizado com uma visão geral das principais áreas de métodos numéricos que são abordadas nas Partes 2 a 6 deste livro.

Os Capítulos 2 e 3 fornecem uma introdução ao ambiente do software MATLAB®. O Capítulo 2 aborda o modo padrão de operar o MATLAB entrando com um comando de cada vez no chamado *modo de comando* ou *modo calculadora*. Esse modo interativo pode orientá-lo no ambiente de forma simples e ilustra como ele é usado para operações comuns, como fazer cálculos e criar gráficos.

O Capítulo 3 mostra como o *modo de programação* do MATLAB fornece um meio de agrupar comandos individuais em algoritmos. Assim, nossa intenção é ilustrar como o MATLAB oferece um ambiente de programação conveniente para desenvolver seus próprios programas computacionais.

O Capítulo 4 trata do importante tópico de análise de erros, que deve ser compreendido para o uso efetivo dos métodos numéricos. A primeira parte do capítulo concentra-se nos *erros de arredondamento* que resultam da impossibilidade de representar algumas quantidades de forma exata nos computadores digitais. A última parte aborda os *erros de truncamento* que surgem ao utilizar uma aproximação em lugar de um procedimento matemático exato.

Modelagem matemática, métodos numéricos e solução de problemas

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é fornecer uma ideia concreta do que são métodos numéricos e como eles estão relacionados à solução de problemas científicos e de engenharia. Após esta leitura completa, você será capaz de:

- ✓ Aprender como modelos matemáticos podem ser formulados com base em princípios científicos para simular o comportamento de um sistema físico simples.
- ✓ Entender como os métodos numéricos fornecem meios de gerar soluções de maneira que possa ser implementada em um computador digital.
- ✓ Compreender os diferentes tipos de leis de conservação presentes nos modelos empregados nas várias disciplinas de engenharia e avaliar a diferença entre as soluções desses modelos para condições dinâmicas e de estado estacionário.
- ✓ Conhecer os aspectos básicos dos diferentes tipos de métodos numéricos que serão abordados neste livro.

VOCÊ TEM UM PROBLEMA



FIGURA 1.1
Forças agindo em um saltador de bungee jumping em queda livre.

Considere que você seja contratado por uma empresa especializada em saltos de *bungee jumping*. Sua tarefa é prever a velocidade do saltador (Figura 1.1) em função do tempo durante o período de queda livre do salto. Essa informação será utilizada como parte de uma análise mais extensa para determinar o comprimento e a resistência da corda elástica para saltadores de diferentes pesos.

Você deve saber, de acordo com seus estudos de física, que a aceleração de um corpo deve ser igual à razão entre a força agindo sobre esse corpo e sua massa (segunda lei de Newton). Considerando isso e seu conhecimento de física e mecânica dos fluidos, podemos desenvolver o seguinte modelo matemático para a taxa de variação da velocidade em relação ao tempo

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2$$

onde v é a velocidade vertical para baixo (m/s), t é o tempo (s), g é a aceleração da gravidade ($\approx 9,81$ m/s), c_d é um coeficiente de arraste “concentrado” (kg/m) e m é a massa do saltador (kg). O coeficiente de arraste, como aparece na equação anterior, é chamado “concentrado”,

pois seu valor depende de fatores como a área do saltador e a densidade do fluido (ver Seção 1.4).

Como essa é uma equação diferencial, o cálculo pode ser empregado para obter uma solução analítica ou exata para v como uma função de t . No entanto, nas páginas seguintes vamos ilustrar uma solução alternativa que envolverá o desenvolvimento de uma solução numérica computacional ou aproximada.

Além de mostrar como o computador pode ser usado para resolver esse problema específico, nosso objetivo geral será ilustrar (a) o que são métodos numéricos e (b) como eles são empregados na solução de problemas científicos e de engenharia. Ao fazer isso, também mostraremos como os modelos matemáticos desempenham um papel importante no modo como engenheiros e cientistas utilizam os métodos numéricos em seu trabalho.

1.1 UM MODELO MATEMÁTICO SIMPLES

Um *modelo matemático* pode ser definido, de forma geral, como uma formulação ou equação que expressa as características essenciais de um sistema ou processo físico em termos matemáticos. Em um sentido muito geral, ele pode ser representado como uma relação funcional da forma

$$\text{Variável dependente} = f\left(\begin{array}{l} \text{variáveis independentes}, \\ \text{parâmetros, funções forçantes} \end{array}\right) \quad (1.1)$$

onde a *variável dependente* é uma característica que normalmente reflete o comportamento ou estado do sistema; as *variáveis independentes* normalmente são dimensões, como tempo e espaço, ao longo das quais o comportamento do sistema é determinado; os *parâmetros* refletem propriedades ou composição do sistema; e as *funções forçantes* são as influências externas agindo sobre o sistema.

A expressão matemática real da Equação (1.1) pode variar de uma simples relação algébrica a um conjunto grande e complicado de equações diferenciais. Por exemplo, com base em suas observações, Newton formulou sua segunda lei do movimento, que afirma que a taxa de variação no tempo do momento de um corpo é igual à força resultante agindo sobre ele. A expressão matemática, ou modelo, da segunda lei de Newton é a equação bastante conhecida

$$F = ma \quad (1.2)$$

onde F é a força resultante agindo sobre o corpo (N, ou Kg m/s²), m é a massa do objeto (kg) e a é a sua aceleração (m/s²).

A segunda lei de Newton pode ser reescrita na forma da Equação (1.1) simplesmente dividindo-se ambos os lados por m para obter

$$a = \frac{F}{m} \quad (1.3)$$

onde a é a variável dependente refletindo o comportamento do sistema, F é a função forçante e m é um parâmetro representando uma propriedade do sistema. Observe que, para esse caso simples, não há nenhuma variável independente, porque não estamos prevendo como a aceleração varia no tempo ou no espaço.

A Equação (1.3) possui diversas características típicas de modelos matemáticos do mundo físico:

- ✓ Descreve um processo ou sistema natural em termos matemáticos.
- ✓ Representa uma idealização e simplificação da realidade. Isto é, o modelo ignora detalhes desnecessários do processo natural e se concentra em suas características essenciais. Portanto, a segunda lei de Newton não inclui os efeitos da relatividade, que são de importância mínima quando aplicados a objetos e forças que interagem perto da superfície da Terra ou sobre ela, com velocidades e em escalas visíveis aos humanos.
- ✓ Produz resultados que podem ser reproduzidos e, consequentemente, usados para propósitos de previsão. Por exemplo, se a força sobre um objeto e sua massa é conhecida, a Equação (1.3) pode ser usada para calcular a aceleração.

Devido à sua forma algébrica simples, a solução da Equação (1.2) é facilmente obtida. Entretanto, outros modelos matemáticos de fenômenos físicos podem ser muito mais complexos e não podem ser resolvidos exatamente (ou exigem técnicas matemáticas mais sofisticadas que a álgebra simples para sua solução). Para ilustrar um modelo mais complexo desse tipo, a segunda lei de Newton pode ser usada para determinar a velocidade terminal de um corpo em queda livre, perto da superfície da Terra. Nossa corpora em queda livre será um saltador de *bungee jumping* (Figura 1.1). Um modelo para esse caso pode ser deduzido expressando a aceleração como taxa de variação no tempo da velocidade (dv/dt) e substituindo-a na Equação (1.3) para obter

$$\frac{dv}{dt} = \frac{F}{m} \quad (1.4)$$

onde v é a velocidade (m/s) e t é o tempo (s). Portanto, a massa multiplicada pela taxa de variação da velocidade é igual à força resultante agindo no corpo. Se a força resultante for positiva, o objeto irá acelerar. Se for negativa, o objeto vai desacelerar. Se a força resultante for nula, a velocidade do objeto permanecerá em um nível constante.

A seguir, vamos expressar a força resultante em termos de variáveis e parâmetros mensuráveis. Para um corpo em queda livre próximo ao solo, a força resultante será composta de duas forças opostas: a força gravitacional, para baixo, F_D e a força de resistência do ar, para cima, F_U :

$$F = F_D + F_U \quad (1.5)$$

Se associarmos um sinal positivo à força para baixo, a segunda lei pode ser usada para escrever a força devida à gravidade como

$$F_D = mg \quad (1.6)$$

onde g é a aceleração da gravidade, que é aproximadamente igual a 9,81 m/s².

A resistência do ar pode ser formulada de diversas maneiras. A partir dos conhecimentos de mecânica dos fluidos, uma primeira boa aproximação seria assumir que ela é proporcional ao quadrado da velocidade,

$$F_U = -c_d v^2 \quad (1.7)$$

onde c_d é uma constante de proporcionalidade chamada de *coeficiente de arraste concentrado* (kg/m). Portanto, quanto maior a velocidade de queda, maior a força

para cima por conta da resistência do ar. O parâmetro c_d representa as propriedades de objetos em queda livre, como a forma ou a rugosidade (aspereza) da superfície, que afetam a resistência do ar. No caso presente, c_d poderia ser uma função do tipo de vestuário ou da orientação usada pelo saltador durante a queda livre.

A força resultante é a diferença entre a força para baixo e a força para cima. Portanto, as Equações (1.4) até (1.7) podem ser combinadas para fornecer

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2 \quad (1.8)$$

A Equação (1.8) é um modelo que relaciona a aceleração do objeto em queda às forças agindo sobre ele. Ela é uma *equação diferencial* porque é escrita em termos da taxa de variação diferencial (dv/dt) da variável que estamos interessados em prever. Entretanto, em contraste com a solução da segunda lei de Newton na Equação (1.3), a solução exata da Equação (1.8) para a velocidade do saltador não pode ser obtida utilizando manipulação algébrica simples. Em vez disso, técnicas mais avançadas de cálculo devem ser aplicadas para se obter uma solução exata ou analítica. Por exemplo, se o saltador está inicialmente em repouso ($v = 0$ em $t = 0$), o cálculo pode ser usado para resolver a Equação (1.8), fornecendo

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) \quad (1.9)$$

onde \tanh é a tangente hiperbólica que pode ser calculada diretamente¹ ou por meio da função exponencial elementar de acordo com

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1.10)$$

Observe que a Equação (1.9) possui a forma geral da Equação (1.1), onde $v(t)$ é a variável dependente, t é a variável independente, c_d e m são parâmetros e g é a função forçante.

EXEMPLO 1.1 Solução analítica para o problema do saltador de *bungee jumping*

Um saltador de *bungee jumping* com uma massa de 68,1 kg pula de um balão de ar quente parado. Use a Equação (1.9) para determinar a velocidade para os 12 segundos iniciais da queda livre. Determine também a velocidade terminal que será atingida, considerando uma corda infinitamente longa. Use um coeficiente de arraste de 0,25 kg/m.

Solução. Inserindo os parâmetros na Equação (1.9), obtemos

$$v(t) = \sqrt{\frac{9,81(68,1)}{0,25}} \tanh\left(\sqrt{\frac{9,81(0,25)}{68,1}} t\right) = 51,6938 \tanh(0,18977t)$$

¹ O MATLAB permite o cálculo direto da tangente hiperbólica pela utilização da função nativa `tanh(x)`.

que pode ser usada para calcular

t, s	$v, \text{m/s}$
0	0
2	18,7292
4	33,1118
6	42,0762
8	46,9575
10	49,4214
12	50,6175
∞	51,6938

De acordo com o modelo, o saltador acelera rapidamente (Figura 1.2). Uma velocidade de 49,4214 m/s (cerca de 178 km/h) é atingida após 10 s. Observe também que, após um tempo suficientemente longo, é atingida uma velocidade constante, chamada de *velocidade terminal*, de 51,6983 m/s (cerca de 186 km/h). Essa velocidade é constante porque, após um longo tempo, a força da gravidade estará em equilíbrio com a resistência do ar. Portanto, a força resultante é nula e a aceleração deixa de existir.

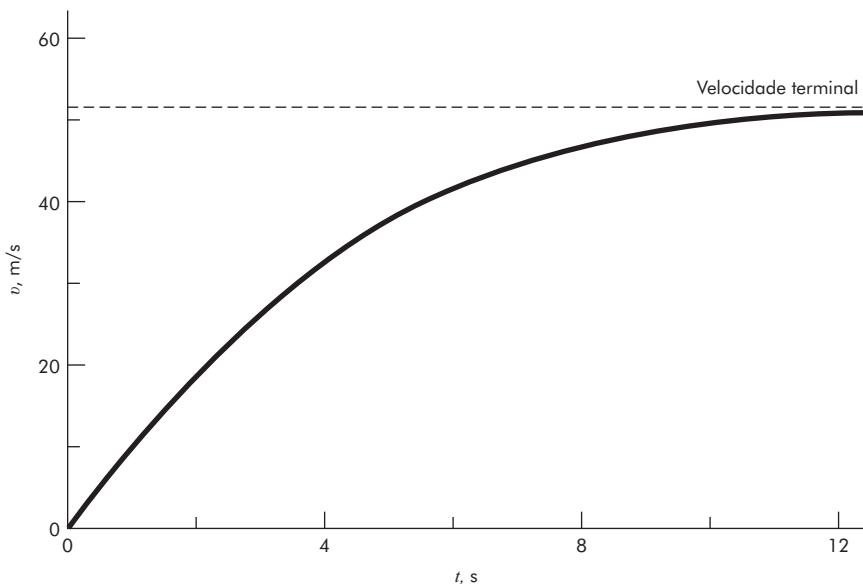


FIGURA 1.2 A solução analítica para o problema do saltador de *bungee jumping* como calculada no Exemplo 1.1. A velocidade aumenta com o tempo e se aproxima assintoticamente da velocidade terminal.

A Equação (1.9) é chamada de *solução analítica* ou *exata* porque satisfaz exatamente a equação diferencial original. Infelizmente, existem muitos modelos matemáticos que não possuem solução exata, tornando necessário desenvolver uma solução numérica que aproxime a solução exata.

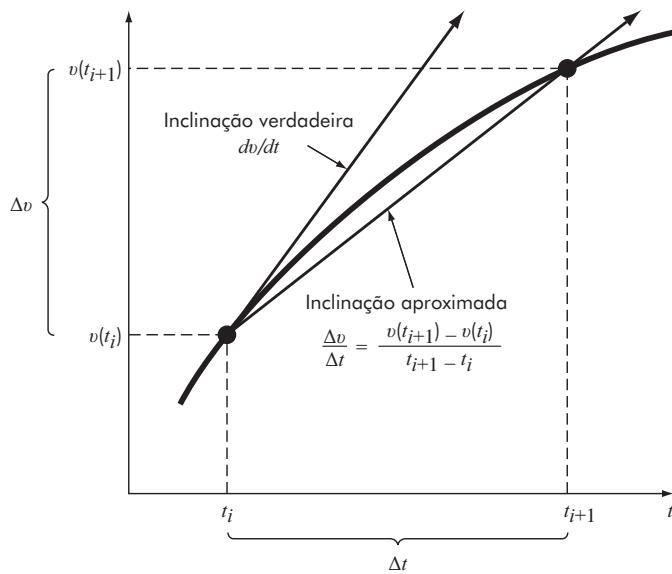


FIGURA 1.3 O uso de uma diferença finita para aproximar a primeira derivada de v com relação a t .

Como já foi mencionado, os *métodos numéricos* são aqueles nos quais os problemas matemáticos são reformulados de forma que possam ser resolvidos por operações aritméticas. Isso pode ser ilustrado para a segunda lei de Newton, observando que a taxa de variação no tempo da velocidade pode ser aproximada por (Figura 1.3):

$$\frac{dv}{dt} \approx \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} \quad (1.11)$$

onde Δv e Δt são respectivamente as diferenças na velocidade e no tempo calculadas sobre intervalos finitos, $v(t_i)$ é a velocidade em um instante inicial t_i e $v(t_{i+1})$ é a velocidade em um instante posterior t_{i+1} . Observe que $dv/dt \approx \Delta v/\Delta t$ é aproximado porque Δt é finito. Lembre-se do cálculo que

$$\frac{dv}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\Delta v}{\Delta t}$$

A Equação (1.11) representa o processo reverso, e é chamada de *aproximação por diferença dividida finita* da derivada no instante t_i . Ela pode ser substituída na Equação (1.8) para fornecer

$$\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} = g - \frac{c_d}{m} v(t_i)^2$$

Esta equação pode ser rearranjada para fornecer

$$v(t_{i+1}) = v(t_i) + \left[g - \frac{c_d}{m} v(t_i)^2 \right] (t_{i+1} - t_i) \quad (1.12)$$

Observe que o termo entre colchetes é o lado direito da equação diferencial propriamente dita [Equação (1.8)]. Isto é, ela fornece um meio de calcular a taxa de variação ou a inclinação de v . Assim, a equação pode ser reescrita de forma mais concisa como

$$v_{i+1} = v_i + \frac{dv_i}{dt} \Delta t \quad (1.13)$$

onde a nomenclatura v_i designa velocidade no instante t_i e $\Delta t = t_{i+1} - t_i$.

Podemos ver agora que a equação diferencial foi transformada em uma equação que pode ser usada para determinar algebricamente a velocidade em t_{i+1} usando a inclinação e os valores anteriores de v e t . Se for dado um valor inicial para a velocidade em algum instante t_i , pode-se facilmente calcular a velocidade em um instante posterior t_{i+1} . Esse novo valor de velocidade em t_{i+1} pode, por sua vez, ser usado para estender o cálculo da velocidade a t_{i+2} , e assim por diante, ou seja, em qualquer instante ao longo do caminho.

$$\text{Valor novo} = \text{valor inicial} + \text{inclinação} \times \text{tamanho do passo}$$

Essa abordagem é formalmente chamada de *método de Euler*. Ela será discutida em detalhes quando falarmos de equações diferenciais mais à frente neste livro.

EXEMPLO 1.2 Solução numérica para o problema do saltador de *bungee jumping*

Faça os mesmos cálculos que foram feitos no Exemplo 1.1, mas use a Equação (1.12) para calcular a velocidade com o método de Euler. Use um tamanho de passo de 2 s para os cálculos.

Solução. No início dos cálculos ($t_0 = 0$), a velocidade do saltador é zero. Usando essa informação e os valores dos parâmetros do Exemplo 1.1, a Equação (1.12) pode ser usada para calcular a velocidade em $t_1 = 2$ s.

$$v = 0 + \left[9,81 - \frac{0,25}{68,1}(0)^2 \right] \times 2 = 19,62 \text{ m/s}$$

Para o intervalo seguinte (de $t = 2$ a 4 s), os cálculos são repetidos, com o resultado

$$v = 19,62 + \left[9,81 - \frac{0,25}{68,1}(19,62)^2 \right] \times 2 = 36,4137 \text{ m/s}$$

Continuam-se os cálculos, de forma análoga, para se obter os valores adicionais:

t, s	$v, \text{m/s}$
0	0
2	19,6200
4	36,4137
6	46,2983
8	50,1802
10	51,3123
12	51,6008
∞	51,6938

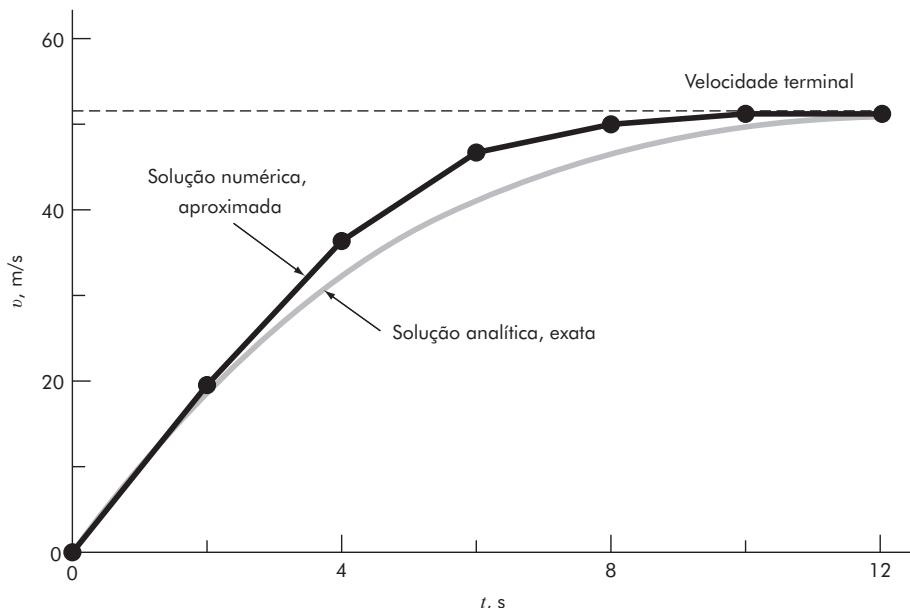


FIGURA 1.4 Comparação das soluções numérica e analítica do problema do saltador de *bungee jumping*.

O resultado é apresentado na Figura 1.4 com a solução exata. É possível verificar que o método numérico retrata as características essenciais da solução exata; entretanto, como foram usados segmentos de reta para aproximar uma função que se curva continuamente, existe alguma discrepância entre os dois resultados. Uma forma de minimizar tais discrepâncias seria usar um passo de tamanho menor. Por exemplo, a aplicação da Equação (1.12) em intervalos de 1 s resulta em um erro menor, já que os segmentos de reta acompanham a solução verdadeira mais de perto. Fazendo-se os cálculos à mão, o esforço associado ao uso de passos cada vez menores tornaria tais soluções numéricas impraticáveis. Entretanto, com o auxílio do computador, uma grande variedade de cálculos pode ser feita facilmente. Portanto, pode-se modelar com exatidão a velocidade do saltador em queda livre sem ter de resolver a equação diferencial exatamente.

Como no Exemplo 1.2, para resultados numéricos mais exatos, deve-se contar com o auxílio do computador. Para cada vez que dividirmos o tamanho do passo pela metade para obter mais acurácia, deverá ser feito o dobro do número de cálculos. Assim, vemos que há uma relação entre a acurácia e o reforço computacional. Os prós e contras dessa relação figuram de forma proeminente nos métodos numéricos e constituem um tema importante neste livro.

1.2 LEIS DE CONSERVAÇÃO E ENGENHARIA

Além da segunda lei de Newton, existem outros princípios de organização importantes na engenharia. Entre os mais importantes estão as *leis de conservação*. Embora elas formem a base para uma variedade de modelos matemáticos complicados e

poderosos, as grandes leis de conservação da ciência e da engenharia são conceitualmente fáceis de entender. Todas elas se reduzem a

$$\text{Variação} = \text{aumento} - \text{diminuição} \quad (1.14)$$

Essa é precisamente a forma empregada quando se usa a lei de Newton para deduzir o balanço de forças para o saltador de *bungee jumping* [Equação (1.8)].

Embora simples, a Equação (1.14) representa uma das formas mais fundamentais de uso das leis de conservação em engenharia – isto é, para prever variações com relação ao tempo. Dá-se à Equação (1.14) o nome especial de cálculo *dependente do tempo* (ou *transiente*).

Além de prever variações, uma outra forma pela qual as leis de conservação são aplicadas é no caso em que não existe a variação. Se a variação for nula, a Equação (1.14) se torna

$$\text{Variação} = 0 = \text{aumento} - \text{diminuição}$$

ou

$$(1.15)$$

$$\text{Aumento} = \text{diminuição}$$

Portanto, se não ocorrer nenhuma variação, o aumento e a diminuição devem estar balanceados. Esse caso, que também possui um nome especial – o cálculo de *estado estacionário* ou *regime permanente* – tem muitas aplicações em engenharia. Por exemplo, para escoamento estacionário de fluidos incompressíveis em tubos, o escoamento entrando em uma junção deve ser balanceado pelo escoamento saindo, como em

$$\text{Escoamento entrando} = \text{escoamento saindo}$$

Para a junção na Figura 1.5, o balanço pode ser usado para calcular que o escoamento saindo do quarto duto deve ser 60.

Para o saltador de *bungee jumping* em queda livre, as condições estacionárias corresponderiam ao caso cuja força resultante fosse nula ou [Equação (1.8) com $dv/dt = 0$]

$$mg = c_d v^2 \quad (1.16)$$

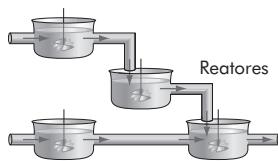
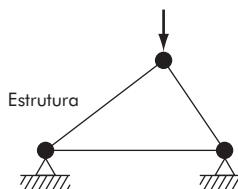
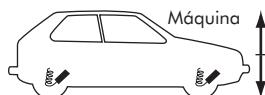
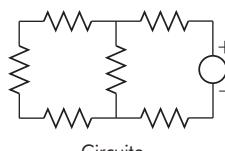
Desse modo, no estado estacionário, as forças para baixo e para cima estão平衡adas e a Equação (1.16) pode ser resolvida para se determinar a velocidade terminal

$$v = \sqrt{\frac{gm}{c_d}}$$

Embora as Equações (1.14) e (1.15) possam parecer muito simples, elas englobam as duas formas fundamentais pelas quais as leis de conservação são empregadas na engenharia e na ciência. Assim, elas constituirão uma parte importante de nossos esforços nos capítulos subsequentes para ilustrar a conexão entre os métodos numéricos e a engenharia e ciência.

A Tabela 1.1 resume alguns modelos e as leis de conservação associadas que formam a base para muitas aplicações de engenharia. A maioria dos problemas de engenharia química envolve balanços de massa para reatores. O balanço de massa é deduzido da conservação de massa. Ele especifica que a variação de massa de um produto químico no reator depende da diferença da quantia de massa escoando para dentro e da quantia de massa escoando para fora.

TABELA 1.1 Dispositivos e tipos de balanços que são comumente usados nas quatro áreas principais da engenharia. Para cada caso, a lei de conservação na qual o balanço é baseado está especificada.

Área	Dispositivo	Princípio organizacional	Expressão matemática
Engenharia química		Conservação da massa	Balanço de massa: Entrada → Saída Ao longo de uma unidade de período de tempo $\Delta\text{massa} = \text{entradas} - \text{saídas}$
Engenharia civil		Conservação do momento	Balanço de força: $+F_V$ $-F_H$ ← → $+F_H$ $-F_V$ Em cada nó $\Sigma \text{Forças horizontais} (F_H) = 0$ $\Sigma \text{Forças verticais} (F_V) = 0$
Engenharia mecânica		Conservação do momento	Balanço de força: Força para cima $x = 0$ Força para baixo $m \frac{d^2x}{dt^2} = \text{força para baixo} - \text{força para cima}$
Engenharia elétrica		Conservação da carga	Balanço de corrente: $+i_1$ → $-i_3$ $+i_2$ Em cada nó $\Sigma \text{corrente} (i) = 0$
		Conservação da energia	Balanço de tensão: $i_1 R_1$ $i_2 R_2$ $i_3 R_3$ ξ Em torno de cada laço $\Sigma \text{fem's} - \Sigma \text{queda de tensão nos resistores} = 0$ $\Sigma \xi - \Sigma iR = 0$

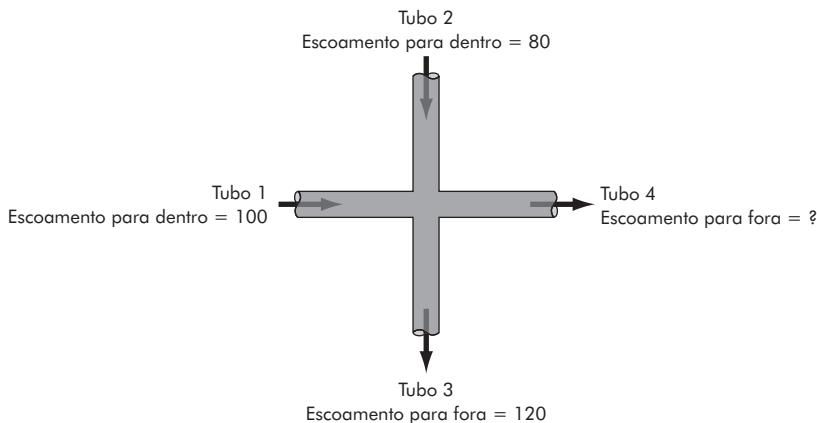


FIGURA 1.5 Balanço de escoamento para o escoamento estacionário de um fluido incompressível na junção de dois tubos.

As aplicações tanto em engenharia civil quanto em engenharia mecânica frequentemente se concentram em modelos desenvolvidos a partir da conservação do momento. Para a engenharia civil, os balanços de força são utilizados para analisar estruturas como a treliça simples na Tabela 1.1. Os mesmos princípios são usados nas aplicações em engenharia mecânica para analisar o movimento transiente para cima e para baixo ou as vibrações de um automóvel.

Finalmente, as aplicações em engenharia elétrica usam tanto balanços de corrente quanto de energia para modelar os circuitos elétricos. O balanço de corrente que resulta da conservação da carga é parecido, em essência, com o balanço de escoamento mostrado na Figura 1.5. Da mesma forma que o escoamento deve ser balanceado em uma junção de tubos, também a corrente elétrica deve ser balanceada em uma junção de fios elétricos. O balanço de energia especifica que as variações de tensão (mais precisamente as quedas de tensão) ao redor de qualquer laço de um circuito devem totalizar zero.

É importante salientar que existem diversos outros ramos da engenharia além da química, civil, mecânica e elétrica. A maior parte deles está relacionada às Quatro Grandes Engenharias. Por exemplo, os conhecimentos da engenharia química são extensivamente utilizados em áreas como engenharia ambiental, de petróleo e biomédica. De modo similar, a engenharia aeroespacial tem muito em comum com a engenharia mecânica. Abordaremos essas áreas por meio de exemplos ao longo deste livro.

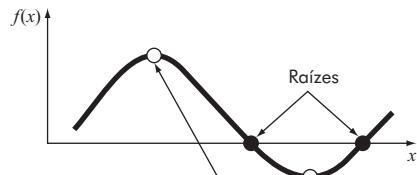
1.3 MÉTODOS NUMÉRICOS ABORDADOS NESTE LIVRO

O método de Euler foi escolhido para este capítulo introdutório porque é típico de várias outras classes de métodos numéricos. Em essência, a maior parte dos métodos consiste na reformulação de operações matemáticas mais complexas, transformando-as em operações algébricas e lógicas simples compatíveis com computadores digitais. A Figura 1.6 resume as principais áreas abordadas neste livro.

(a) Parte 2: Raízes e otimização

Raízes: Resolva para x de modo que $f(x) = 0$

Otimização: Resolva para x de modo que $f'(x) = 0$

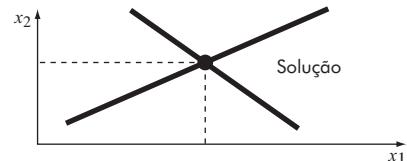


(b) Parte 3: Equações algébricas lineares

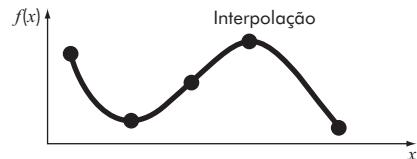
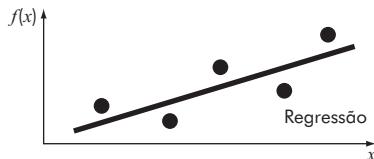
Dados a e b , resolva para x

$$a_{11}x_1 + a_{12}x_2 = b_1$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$



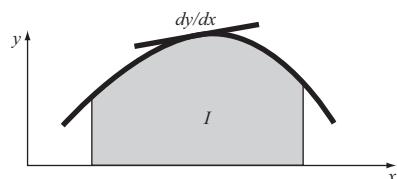
(c) Parte 4: Ajuste de curvas



(d) Parte 5: Integração e diferenciação

Integração: Encontre a área sob a curva

Diferenciação: Encontre a inclinação da curva



(e) Parte 6: Equações diferenciais

Dado

$$\frac{dy}{dt} \approx \frac{\Delta y}{\Delta t} = f(t, y),$$

resolva para y como uma função de t

$$y_{i+1} = y_i + f(t_i, y_i)\Delta t$$

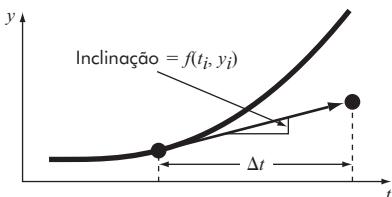


FIGURA 1.6 Resumo dos métodos numéricos abordados neste livro.

A *Parte 2* trata de dois tópicos relacionados: determinação de raízes e otimização. Como mostra a Figura 1.6a, a *localização de raízes* está relacionada à busca dos zeros de uma função. A *otimização*, por sua vez, envolve a determinação de um valor (ou valores) de uma variável independente que corresponde ao “melhor” valor ou valor ótimo de uma função. Portanto, como indicado na Figura 1.6a, a otimização envolve a identificação de máximos e mínimos. Embora sejam empregadas abordagens um pouco diferentes, tanto a localização de raízes como a otimização surgem, normalmente, em contextos de projetos de engenharia.

A *Parte 3* é dedicada à solução de sistemas de equações algébricas lineares simultâneas (Figura 1.6b). Esses problemas são similares, em essência, aos das raízes de equações, uma vez que dizem respeito aos valores que satisfazem tais equações. Entretanto, em vez de satisfazer uma única equação, é procurado um conjunto de

valores que satisfaça simultaneamente um conjunto de equações algébricas lineares. Essas equações aparecem em uma grande variedade de problemas e em todas as disciplinas da engenharia. Em particular, elas se originam na modelagem matemática de grandes sistemas de elementos interconectados, como estruturas, circuitos elétricos e redes de fluidos. Entretanto, também são encontradas em outras áreas dos métodos numéricos, como ajuste de curvas e equações diferenciais.

Como um engenheiro ou cientista, você frequentemente terá a oportunidade (e a necessidade) de ajustar curvas a um conjunto de dados. As técnicas desenvolvidas para esse propósito podem ser divididas em duas categorias gerais: regressão e interpolação. Como descrito na *Parte 4* (Figura 1.6c), a *regressão* é empregada quando existe um grau significativo de erro associado aos dados. Os resultados experimentais são geralmente desse tipo. Para tais situações, a estratégia é desenvolver uma única curva que represente a tendência geral dos dados, sem necessariamente coincidir com nenhum ponto individual.

A *interpolação*, por sua vez, é usada quando o objetivo é determinar valores intermediários entre dados relativamente livres de erros – normalmente, é o caso para informações tabuladas. Para tais situações, a estratégia é ajustar a curva diretamente pelos pontos dados e usar essa curva para prever os valores intermediários.

Como mostra a Figura 1.6d, a *Parte 5* é dedicada à integração e à diferenciação. Uma interpretação física da *integração numérica* é a determinação da área sob a curva, e possui muitas aplicações na engenharia e na ciência, variando da determinação de centroides de objetos de forma estranha a cálculos de quantidades totais baseados em conjuntos de medidas discretas. Além disso, as fórmulas de integração numérica desempenham um papel importante na solução de equações diferenciais. A *Parte 5* também aborda métodos para *diferenciação numérica*. Como você deve ter aprendido nos estudos de cálculo, a diferenciação envolve a determinação da inclinação de uma função ou sua taxa de variação.

Finalmente, a *Parte 6* se concentra na solução de *equações diferenciais ordinárias* (Figura 1.6e). Essas equações possuem grande importância em todas as áreas da engenharia e da ciência, e isso ocorre porque muitas leis físicas são descritas em termos da taxa de variação de uma quantidade em vez do valor da quantidade propriamente dita. Os exemplos variam de modelos de previsão populacional (taxa de variação da população) à aceleração de um corpo em queda livre (taxa de variação da velocidade). Dois tipos de problemas são tratados: problemas de valor inicial e valor de contorno.

► 1.4 ESTUDO DE CASO FORÇA DE ARRASTE REAL

Contextualização. Em nosso modelo do saltador de *bungee jumping* em queda livre, foi assumido que a força de arraste (força de resistência do ar) depende do quadrado da velocidade (Equação 1.7). Uma representação mais detalhada, originalmente formulada por Lord Rayleigh, pode ser escrita como

$$F_d = -\frac{1}{2}\rho v^2 AC_d \vec{v} \quad (1.17)$$



onde F_d é a força de arraste (N), ρ é a densidade do fluido (kg/m^3), A é a área frontal do objeto em um plano perpendicular à direção de movimento (m^2), C_d é um coeficiente de arraste adimensional e \vec{v} é um vetor unitário indicando o sentido da velocidade.

Essa relação, que assume condições turbulentas (isto é, um *número de Reynolds* elevado), permite expressar o coeficiente de arraste concentrado da Equação (1.7) em termos mais fundamentais como

$$C_d = \frac{1}{2} \rho A C_d \quad (1.18)$$

Assim, o coeficiente de arraste concentrado depende da área do objeto, da densidade do fluido e de um coeficiente de arraste adimensional. Este último leva em conta todos os outros fatores que contribuem para a resistência do ar, como a “rugosidade” (ou aspereza) do objeto. Por exemplo, um saltador vestindo um traje largo terá um C_d maior que outro vestindo um macacão liso próprio para saltos.

Observe que para os casos em que a velocidade é muito baixa, o regime de fluxo em torno do objeto será laminar e a relação entre a força de arraste e a velocidade torna-se linear. Essa relação linear é chamada de *lei de Stokes* para o coeficiente de arraste.

No desenvolvimento de nosso modelo para o saltador de *bungee jumping* foi assumido que o sentido para baixo era positivo. Assim, a Equação (1.7) é uma representação acurada da Equação (1.17), pois $\vec{v} = +1$ e a força de arraste é negativa. Portanto, o arraste diminui a velocidade.

Mas o que acontece se o saltador possuisse uma velocidade para cima (isto é, negativa)? Nesse caso, $\vec{v} = -1$ e a Equação (1.17) forneceria uma força de arraste positiva. Novamente, teríamos uma representação fisicamente correta, uma vez que a força de arraste positiva age para baixo contra a velocidade negativa para cima.

Infelizmente, neste estudo de caso, a Equação (1.7) fornece uma força de arraste negativa, porque ela não inclui o vetor unitário direcional. Em outras palavras, ao elevar a velocidade ao quadrado, seu sinal – e, portanto, seu sentido – é perdido. Como consequência, o modelo fornece um resultado fisicamente inconsistente, que corresponde à resistência do ar agindo de modo a acelerar o objeto para cima (ou seja, a força de arraste age a favor e não contra o movimento do saltador).

Modificaremos então nosso modelo de modo que ele funcione adequadamente para velocidades orientadas tanto para baixo como para cima. Após isso, vamos testar o modelo modificado para o mesmo caso do Exemplo 1.2, porém com um valor inicial de $v(0) = -40 \text{ m/s}$. Além disso, ilustraremos como estender a análise numérica para determinar a posição do saltador.

Solução. A modificação simples a seguir permite incorporar o sinal na força de arraste:

$$F_d = -\frac{1}{2} \rho v |v| A C_d \quad (1.19)$$

ou, em termos do coeficiente de arraste concentrado:

$$F_d = -c_d v |v| \quad (1.20)$$

Assim, a equação diferencial a ser resolvida é

$$\frac{dv}{dt} = g - \frac{c_d}{m} v |v| \quad (1.21)$$



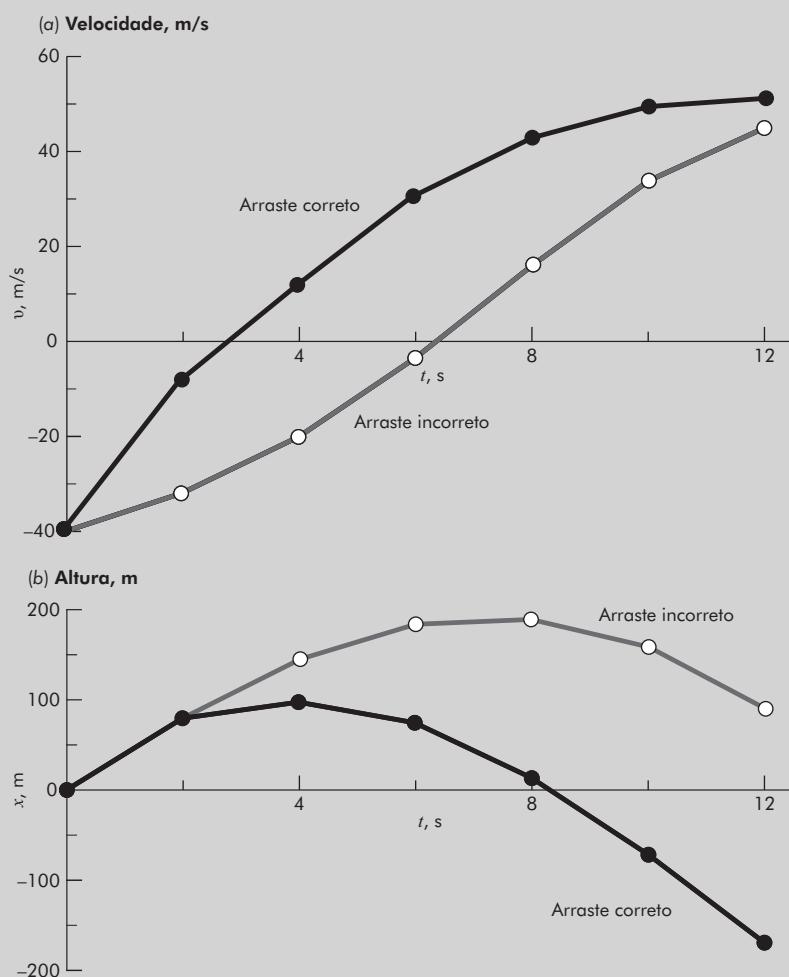


FIGURA 1.7 Gráficos de (a) velocidade e (b) altura para o saltador de *bungee jumping* em queda livre com uma velocidade inicial para cima (negativa) obtidos com o método de Euler. São exibidos os resultados para a formulação de arraste correto (Equação 1.20) e incorreta (Equação 1.7).

Com o objetivo de determinar a posição do saltador, consideramos que a distância percorrida, $x(m)$, é relacionada à velocidade por

$$\frac{dx}{dt} = -v \quad (1.22)$$

Em contraste com a velocidade, essa formulação assume que a distância para cima é positiva. Da mesma forma que a Equação (1.12), essa equação pode ser integrada numericamente com o método de Euler:

$$x_{i+1} = x_i - v(t_i)\Delta t \quad (1.23)$$

Assumindo que a posição inicial do saltador é definida como $x(0) = 0$, e usando os valores dos parâmetros dos Exemplos 1.1 e 1.2, a velocidade e a distância em $t = 2$ s podem ser calculadas como

$$v(2) = -40 + \left[9,81 - \frac{0,25}{68,1}(-40)(40) \right] 2 = -8,6326 \text{ m/s}$$

$$x(2) = 0 - (-40)2 = 80 \text{ m}$$

Observe que, se tivéssemos utilizado a formulação incorreta para o arraste, os resultados seriam $-32,1274$ m/s e 80 m.

O cálculo pode ser repetido para o próximo intervalo ($t = 2$ a 4 s):

$$v(4) = -8,6326 + \left[9,81 - \frac{0,25}{68,1}(-8,6326)(8,6326) \right] 2 = 11,5346 \text{ m/s}$$

$$x(4) = 80 - (-8,6326)2 = 97,2651 \text{ m}$$

A formulação incorreta para o arraste fornece $-20,0858$ m/s e 144,2549 m.

Dando continuidade aos cálculos, são obtidos os resultados mostrados na Figura 1.7 com aqueles obtidos com o modelo de arraste incorreto. Observe que a formulação correta indica uma desaceleração mais rápida porque a força de arraste sempre age diminuindo a velocidade.

Neste caso, com o tempo, ambas as soluções da velocidade convergirão para a mesma velocidade terminal, pois por acaso estão orientadas para baixo; isto é, a Equação (1.7) está correta. No entanto, o impacto sobre a predição da altura do objeto é bastante significativo para o caso de arraste incorreto, resultando em uma trajetória muito mais alta.

Este estudo de caso mostra como é importante empregar um modelo fisicamente correto. Em alguns casos, a solução fornecerá resultados que são claramente inconsistentes. Por outro lado, o exemplo apresentado é mais “traiçoeiro”, uma vez que não há evidência visual de que a solução com arraste incorreto está errada. Isto é, a solução incorreta parece razoável.

►► PROBLEMAS

1.1 Use o cálculo para verificar que a Equação (1.9) é uma solução da Equação (1.8) para a condição inicial $v(0) = 0$.

1.2 Use o cálculo para resolver a Equação (1.21) para o caso em que a velocidade inicial é (a) positiva e (b) negativa. (c) Com base em seus resultados para (a) e (b), efetue os mesmos cálculos como no Exemplo 1.1, mas com uma velocidade inicial de -40 m/s. Calcule os valores de velocidade de $t = 0$ até 12 s em intervalos de 2 s. Observe que para esse caso, a velocidade nula ocorre em $t = 3,470239$ s.

1.3 As seguintes informações estão disponíveis para uma conta bancária:

Data	Depósitos	Saque	Balanço
1/5		220,13	1.512,33
1/6	216,80		378,61
1/7	450,25		106,80
1/8	127,31		350,61
1/9			

Observe que o dinheiro rende juros, calculados da seguinte forma:

$$\text{Juros} = i B_i$$

onde i = a taxa de juros expressa como uma fração por mês, e B_i = o balanço inicial no começo do mês.

- (a) Use a conservação do dinheiro para calcular o balanço em $1/6$, $1/7$, $1/8$ e $1/9$ se a taxa de juros for 1% por mês ($i = 0,01/\text{mês}$). Mostre cada etapa dos cálculos.
- (b) Escreva uma equação diferencial para o balanço de dinheiro na forma

$$\frac{dB}{dt} = f[D(t), W(t), i]$$

onde t = tempo (meses), $D(t)$ = depósito como uma função do tempo (R\$/mês), $W(t)$ = saques como uma função do tempo (R\$/mês). Para esse caso, assuma que os juros são compostos continuamente; isto é, juros = iB .

- (c) Use o método de Euler com um passo de tempo de $0,5$ mês para simular o balanço. Assuma que os depósitos e saques são efetuados uniformemente ao longo do mês.
- (d) Faça um gráfico de balanço *versus* tempo para (a) e (c).

1.4 Repita o Exemplo 1.2. Calcule a velocidade para $t = 12$ s com um tamanho de passo de (a) 1 e (b) $0,5$ s. Com base nos resultados, você pode fazer alguma afirmação com relação aos erros nos cálculos?

1.5 Em vez da relação não linear da Equação (1.7), você poderia escolher modelar a força para cima no saltador de *bungee jumping* como uma relação linear:

$$F_U = c'v$$

onde c' é um coeficiente de arraste de primeira ordem (kg/s).

- (a) Usando o cálculo, obtenha a solução na forma fechada, no caso em que o saltador está inicialmente em repouso ($v = 0$ em $t = 0$).
- (b) Repita o cálculo numérico do Exemplo 1.2 com a mesma condição inicial e os mesmos valores de parâmetros. Use um valor de $11,5$ kg/s para c' .

1.6 Para o saltador de *bungee jumping* em queda livre com arraste linear (Problema 1.5), considere inicialmente um indivíduo de 70 kg com um coeficiente de arraste de 12 kg/s. Se um segundo saltador tem um coeficiente de arraste de 15 kg/s e uma massa de 80 kg, quanto tempo levará para ele atingir a mesma velocidade que o primeiro atingiu em 9 s?

1.7 Para o modelo de arraste de segunda ordem [Equação (1.8)], calcule a velocidade de um paraquedista em queda livre utilizando o método de Euler para o caso em que $m = 80$ kg e $c_d = 0,25$ kg/m. Realize os cálculos de $t = 0$ até 20 s com um tamanho de passo de 1 s. Use uma condição inicial em que o paraquedista tenha uma velocidade para cima de 20 m/s em $t = 0$. Em $t = 10$ s, considere que o paraquedas é instantaneamente aberto, de forma que o coeficiente de arraste pule para $1,5$ kg/m.

1.8 A quantidade de contaminante radioativo uniformemente distribuído em um reator fechado é medida pela sua concentração c (becquerel/litro ou Bq/L). O contaminante diminui a uma taxa de decaimento proporcional à sua concentração; isto é,

$$\text{Taxa de decaimento} = -kc$$

onde k é uma constante, com unidade de dia⁻¹. Portanto, de acordo com a Equação (1.14), um balanço de massa para o reator pode ser escrito como

$$\frac{dc}{dt} = -kc$$

$$\left(\begin{array}{l} \text{variação} \\ \text{na massa} \end{array} \right) = \left(\begin{array}{l} \text{diminuição} \\ \text{por decaimento} \end{array} \right)$$

- (a) Use o método de Euler para resolver essa equação de $t = 0$ até 1 d com $k = 0,175$ d⁻¹. Use um tamanho de passo $\Delta t = 0,1$ d. A concentração em $t = 0$ é 100 Bq/L.
- (b) Trace a solução em um gráfico *semilog* (isto é, $\ln c$ *versus* t) e determine a inclinação. Interprete seus resultados.

1.9 Um tanque de armazenamento (Figura P1.9) contém um líquido à profundidade y , onde $y = 0$ quando o tanque está cheio até a metade. O líquido é retirado do tanque a uma vazão constante Q para atender à demanda. O conteúdo é reposto a uma taxa senoidal de $3Q \sin^2(t)$. A Equação (1.14) pode ser reescrita para esse sistema como

$$\frac{d(Ay)}{dt} = 3Q \sin^2(t) - Q$$

$$\left(\begin{array}{l} \text{variação} \\ \text{no volume} \end{array} \right) = \left(\begin{array}{l} \text{fluxo} \\ \text{para dentro} \end{array} \right) - \left(\begin{array}{l} \text{fluxo} \\ \text{para fora} \end{array} \right)$$

ou, já que a área da superfície, A , é constante

$$\frac{dy}{dt} = 3 \frac{Q}{A} \sin^2(t) - \frac{Q}{A}$$

Use o método de Euler para encontrar a profundidade y de $t = 0$ a 10 d com um tamanho de passo de

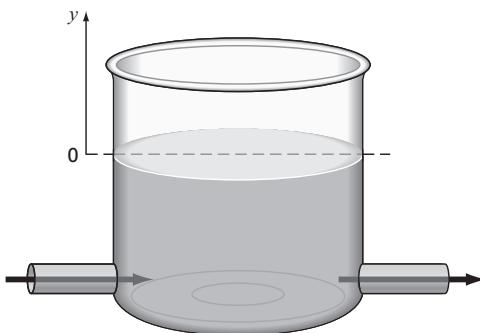


FIGURA P1.9

0,5 d. Os valores dos parâmetros são $A = 1.250 \text{ m}^2$ e $Q = 450 \text{ m}^3/\text{d}$. Suponha que a condição inicial seja $y = 0$.

1.10 Para o mesmo tanque de armazenamento descrito no Problema 1.9, considere que o fluxo de líquido para fora não seja constante, mas dependa da profundidade. Nesse caso, a equação diferencial para a profundidade pode ser escrita como

$$\frac{dy}{dt} = 3 \frac{Q}{A} \operatorname{sen}^2(t) - \frac{\alpha(1+y)^{1,5}}{A}$$

Use o método de Euler para encontrar a profundidade y de $t = 0$ a 10 d com um tamanho de passo de 0,5 d. Os valores dos parâmetros são $A = 1.250 \text{ m}^2$, $Q = 450 \text{ m}^3/\text{d}$ e $\alpha = 150$. Considere que a condição inicial seja $y = 0$.

1.11 Aplique a conservação de volume (ver Problema 1.9) para simular o nível de líquido em um tanque de armazenamento cônico (Figura P1.11). O líquido flui para dentro do tanque a uma taxa senoidal de $Q_{\text{entrada}} = 3 \operatorname{sen}^2(t)$ e flui para fora de acordo com

$$\begin{aligned} Q_{\text{saída}} &= 3(y - y_{\text{saída}})^{1,5} & y > y_{\text{saída}} \\ Q_{\text{saída}} &= 0 & y \leq y_{\text{saída}} \end{aligned}$$

onde o fluxo tem unidade de m^3/d e y é a elevação da superfície da água acima do fundo do tanque (m). Use o método de Euler para resolver para a profundidade y para $t = 0$ a 10 d com um tamanho de passo de 0,5 d. Os valores dos parâmetros são $r_{\text{topo}} = 2,5 \text{ m}$, $y_{\text{topo}} = 4 \text{ m}$ e $y_{\text{saída}} = 1 \text{ m}$. Considere que o nível esteja inicialmente abaixo do tubo de saída com $y(0) = 0,8 \text{ m}$.

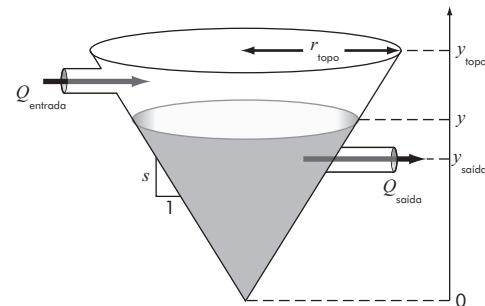


FIGURA P1.11

1.12 Um grupo de 35 alunos assiste aula em uma sala que mede $11 \text{ m} \times 8 \text{ m} \times 3 \text{ m}$. Cada estudante ocupa cerca de $0,075 \text{ m}^3$ e libera cerca de 80 W de calor ($1 \text{ W} = 1 \text{ J/s}$). Calcule o aumento da temperatura do ar durante os primeiros 20 minutos de aula, se a sala estiver completamente selada e isolada. Considere que a capacidade térmica C_v do ar seja $0,718 \text{ kJ/(kg K)}$, e que o ar seja um gás ideal a 20°C e $101,325 \text{ kPa}$. Observe que o calor absorvido pelo ar, Q , está relacionado à massa de ar m , à capacidade térmica e à variação de temperatura pela seguinte relação:

$$Q = m \int_{T_1}^{T_2} C_v dT = m C_v (T_2 - T_1)$$

A massa de ar pode ser obtida da lei dos gases ideais:

$$PV = \frac{m}{M_w t} RT$$

onde P é a pressão do gás, V é o volume do gás, T é a temperatura do gás, $M_w t$ é o peso molecular do gás (para o ar, $28,97 \text{ kg/kmol}$) e R é a constante dos gases ideais [$8,314 \text{ kPa m}^3/\text{kmol K}$].

1.13 A Figura P1.13 mostra as diversas formas pelas quais um homem comum ganha e perde água durante um dia. Um litro é ingerido com a comida e o corpo metabolicamente produz 0,3 L. Respirando o ar, a troca é 0,05 L enquanto inala, e 0,4 L enquanto exala, em um período de um dia. O corpo também vai perder 0,3, 1,4, 0,2 e 0,35 L através de suor, urina, fezes e pela pele, respectivamente. Para manter uma condição estacionária, quanta água deve ser bebida por dia?

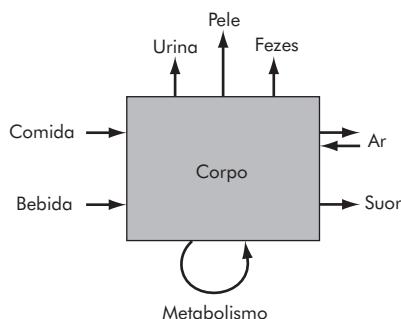


FIGURA P1.13

1.14 No exemplo do saltador de *bungee jumping* em queda livre, assumimos que a aceleração da gravidade era uma constante de valor $9,81 \text{ m/s}^2$. Embora esta seja uma boa aproximação ao examinar objetos caindo perto da superfície da Terra, a força gravitacional decresce quando nos movemos para cima do nível do mar. Uma representação mais geral, baseada na lei da gravitação universal de Newton, pode ser escrita como

$$g(x) = g(0) \frac{R^2}{(R + x)^2}$$

onde $g(x)$ é a aceleração da gravidade na altura x (em m) medida para cima a partir da superfície da Terra (m/s^2), $g(0)$ é a aceleração da gravidade na superfície da Terra ($\approx 9,81 \text{ m/s}^2$) e R é o raio da Terra ($\approx 6,37 \times 10^6 \text{ m}$).

- (a) Em uma forma similar à dedução da Equação (1.8), utilize o balanço de forças para deduzir uma equação diferencial para a velocidade como uma função do tempo que use essa representação mais completa da gravitação. Entretanto, para tal dedução, considere que a velocidade para cima seja positiva.
- (b) Para o caso de o arraste ser desprezível, use a regra da cadeia para expressar a equação diferencial como uma função da altura em vez do tempo. Lembre-se de que a regra da cadeia é

$$\frac{dv}{dt} = \frac{dv}{dx} \frac{dx}{dt}$$

- (c) Use o cálculo para obter uma solução na forma fechada em que $v = v_0$ em $x = 0$.
- (d) Use o método de Euler para obter uma solução numérica de $x = 0$ até 100.000 m utilizando

zando um passo de 10.000 m , com velocidade inicial de 1.500 m/s para cima. Compare seu resultado com a solução analítica.

1.15 Considere uma gotícula esférica de líquido que evapora a uma taxa que é proporcional à área de sua superfície.

$$\frac{dV}{dt} = -kA$$

onde V é o volume (mm^3), t é o tempo (min), k é a taxa de evaporação (mm/min) e A é a área da superfície (mm^2). Use o método de Euler para calcular o volume da gotícula de $t = 0$ a 10 min usando um tamanho de passo de $0,25 \text{ min}$. Considere que $k = 0,08 \text{ mm/min}$ e que a gotícula tenha inicialmente um raio de $2,5 \text{ mm}$. Avalie a validade de seus resultados determinando o raio do seu volume final calculado e verificando que ele é consistente com a taxa de evaporação.

1.16 Um fluido é bombeado para dentro da rede mostrada na Figura P1.16. Se $Q_2 = 0,7$, $Q_3 = 0,5$, $Q_7 = 0,1$ e $Q_8 = 0,3 \text{ m}^3/\text{s}$, determine os outros fluxos.

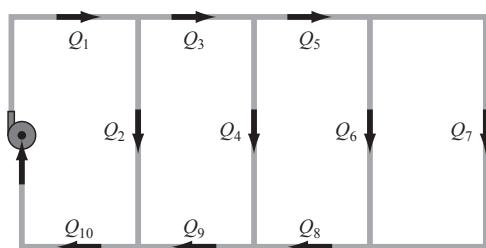


FIGURA P1.16

1.17 A *lei de resfriamento de Newton* diz que a temperatura de um corpo varia a uma taxa proporcional à diferença entre a sua temperatura e a temperatura do meio que o cerca (a temperatura ambiente),

$$\frac{dT}{dt} = -k(T - T_a)$$

onde T é a temperatura do corpo ($^\circ\text{C}$), t é o tempo (min), k é a constante de proporcionalidade (por minuto) e T_a é a temperatura ambiente ($^\circ\text{C}$). Considere uma xícara de café que originalmente tenha a temperatura de $70 \text{ }^\circ\text{C}$. Use o método de Euler para calcular a temperatura de $t = 0$ a 20 min utilizando um tamanho de passo de 2 min se $T_a = 20 \text{ }^\circ\text{C}$ e $k = 0,019/\text{min}$.

1.18 Você está trabalhando como investigador de uma cena de crime e deve prever a temperatura de uma vítima de homicídio ao longo de um período de 5 h. Você sabe que a temperatura da sala onde a vítima foi encontrada estava em 10 °C quando o corpo foi descoberto.

- (a) Use a lei de resfriamento de Newton (Problema 1.17) e o método de Euler para calcular a temperatura do corpo da vítima para o período de 5 h usando valores de $k = 0,12/\text{h}$ e $\Delta t = 0,5 \text{ h}$. Assuma que a temperatura do corpo da vítima no instante da morte era 37 °C e que a temperatura da sala estava em um valor constante de 10 °C ao longo do período de 5 h.
- (b) Investigações adicionais revelam que, na realidade, a temperatura da sala decaiu linearmente de 20 para 10 °C ao longo do período de 5 h. Repita os mesmos cálculos do item (a), porém incorporando essa nova informação.
- (c) Compare os resultados de (a) e (b) traçando-os em um mesmo gráfico.

1.19 A velocidade é igual à taxa de variação da posição x (m):

$$\frac{dx}{dt} = v(t) \quad (\text{P1.19})$$

Use o método de Euler para integrar numericamente as Equações (P1.19) e (1.8) para determinar a velocidade e a posição do objeto durante a queda como uma função do tempo para os primeiros 10 segundos de queda livre, utilizando os mesmos

parâmetros e condições do Exemplo 1.2. Faça um gráfico de seus resultados.

1.20 Além da força de gravidade para baixo (peso) e do arraste, um objeto caindo através de um fluido também está sujeito à força de empuxo que é proporcional ao volume deslocado. Por exemplo, para uma esfera com diâmetro d (m), o volume da esfera é $V = \pi d^3/6$ e sua área projetada é $A = \pi d^2/4$. A força de empuxo pode ser calculada como $F_b = -\rho V g$. O empuxo foi desconsiderado na dedução da Equação (1.8) porque ele é relativamente pequeno para um objeto como um saltador de *bungee jumping* se movendo através do ar. No entanto, para fluidos mais densos como a água, ele torna-se mais significativo.

- (a) Deduza uma equação diferencial da mesma forma como a Equação (1.8), mas inclua a força de empuxo e represente a força de arraste como descrito na Seção 1.4.
- (b) Reescreva a equação diferencial obtida em (a) para o caso especial de uma esfera.
- (c) Use a equação determinada em (b) para calcular a velocidade terminal (isto é, a velocidade para o estado estacionário). Use os seguintes valores de parâmetros para a esfera em queda através da água: diâmetro da esfera = 1 cm, densidade da esfera = 2.700 kg/m³, densidade da água = 1.000 kg/m³, e $C_d = 0,47$.
- (d) Use o método de Euler com um tamanho de passo de $\Delta t = 0,03125 \text{ s}$ para resolver numericamente a equação para velocidade de $t = 0$ a 0,25 s, considerando a velocidade inicial igual a zero.

Princípios básicos do MATLAB

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é fornecer uma introdução e uma visão geral de como o modo calculadora do MATLAB é usado para implementar cálculos interativos. Após esta leitura completa, você será capaz de:

- ✓ Aprender como números reais e complexos são atribuídos a variáveis.
- ✓ Aprender como atribuir valores a vetores e matrizes utilizando a atribuição simples, o operador dois-pontos e as funções `linspace` e `logspace`.
- ✓ Entender as regras de prioridade para construir expressões matemáticas.
- ✓ Compreender o que são funções nativas e aprender mais sobre elas com os recursos de Help (Ajuda) do MATLAB.
- ✓ Aprender como usar vetores para gerar um gráfico simples com base em uma equação.

VOCÊ TEM UM PROBLEMA

No Capítulo 1, usamos um balanço de forças para determinar a velocidade terminal de um corpo em queda livre, por exemplo, um saltador de *bungee jumping*,

$$v_t = \sqrt{\frac{gm}{c_d}}$$

onde v_t é a velocidade terminal (m/s), g é a aceleração da gravidade (m/s^2), m é a massa (kg) e c_d é o coeficiente de arraste (kg/m). Com essa equação é possível prever a velocidade terminal; além disso, ela pode ser reorganizada para calcular o coeficiente de arraste

$$c_d = \frac{mg}{v_t^2} \quad (2.1)$$

Assim, se medirmos a velocidade terminal de vários saltadores de massa conhecida, essa equação fornecerá um meio para estimar o coeficiente de arraste. Os dados na Tabela 2.1 foram coletados para esse propósito.

TABELA 2.1 Dados para a massa e velocidades terminais associadas a uma série de saltadores.

m, kg	83,6	60,2	72,1	91,1	92,9	65,3	80,9
v_t, m/s	53,4	48,5	50,9	55,7	54	47,7	51,1

Neste capítulo, aprenderemos como o MATLAB pode ser usado para analisar tais dados e calcular quantidades como coeficientes de arraste. Ilustraremos também como as capacidades gráficas do MATLAB fornecem informações que complementam essas análises.

2.1 O AMBIENTE DO MATLAB

O MATLAB é um software que fornece ao usuário um ambiente adequado à realização de diversos tipos de cálculos; além disso, contém ferramentas bastante úteis para implementação de métodos numéricos.

Pode-se operar o MATLAB inserindo um comando de cada vez no prompt de comando (command window). Neste capítulo, usaremos esse modo interativo ou *modo calculadora* para introduzir operações comuns, como realização de cálculos simples e criação de gráficos. No Capítulo 3, mostraremos como tais comandos podem ser usados para desenvolver programas no MATLAB.

Salientamos que este capítulo foi escrito como um exercício; isto é, você deve lê-lo enquanto está sentado na frente do computador. A maneira mais fácil para se tornar proficiente é implementar na prática os comandos no MATLAB conforme for avançando no material a seguir.

O MATLAB utiliza três janelas (windows) principais:

- ✓ Command window: Utilizada para inserir comandos e dados.
- ✓ Graphics window: Utilizada para exibir gráficos.
- ✓ Edit window: Utilizada para criar e editar arquivos-M (M-files ou programa/ funções do MATLAB).

Neste capítulo usaremos as janelas command window e graphics window; no Capítulo 3, usaremos a janela edit window para desenvolver programas no MATLAB.

Após inicializar o MATLAB, a janela command window abrirá com o prompt de comando a seguir

```
>>
```

O modo calculadora do MATLAB opera em um modo sequencial à medida que os comandos são digitados linha por linha. Para cada comando aparece um resultado, portanto, pode-se pensar nesse modo de operação do MATLAB como uma calculadora sofisticada. Por exemplo, digitando-se

```
>> 55 - 16
```

O MATLAB exibe o resultado¹

```
ans =  
39
```

Observe que o MATLAB atribuiu automaticamente a resposta a uma variável, ans. Portanto, agora pode-se utilizar a variável ans no cálculo seguinte:

```
>> ans + 11
```

¹ O MATLAB pula uma linha entre o rótulo (ans =) e o número (39). Aqui, omitimos essa linha vazia por simples economia de espaço, mas você pode controlar a inserção ou não de linhas vazias com os comandos format compact e format loose.

com o resultado

```
ans =
50
```

O MATLAB sempre atribuirá os resultados à variável-padrão `ans`, se os cálculos não forem explicitamente atribuídos a uma variável definida pelo usuário.

2.2 ATRIBUIÇÃO

A ação de atribuir valores a uma variável chama-se atribuição, o que resulta no armazenamento dos valores na posição de memória correspondente à variável.

2.2.1 Escalares

A atribuição de valores a variáveis escalares é similar a outras linguagens de programação. Digite

```
>> a = 4
```

e observe que, após o comando de atribuição, o nome e o valor da variável são impressos na tela para confirmar o que foi feito:

```
a =
4
```

Essa é uma característica do MATLAB, que pode ser suprimida finalizando a linha de comando com um ponto e vírgula (`;`). Digite

```
>> A = 6;
```

É possível digitar vários comandos na mesma linha separando-os com vírgulas ou pontos e vírgulas. Os comandos terminados por vírgulas serão exibidos, enquanto aqueles terminados por ponto e vírgula não serão exibidos. Por exemplo,

```
>> a = 4, A = 6; x = 1;
a =
4
```

O MATLAB é sensível a maiúsculas e minúsculas (*case sensitive*) – isto é, a variável `a` não é o mesmo que `A`. Para ilustrar isso, digite

```
>> a
```

e, em seguida, digite

```
>> A
```

Veja como os valores são distintos. Essas variáveis, portanto, têm nomes diferentes.

É possível atribuir valores complexos às variáveis, uma vez que o MATLAB efetua manipulações aritméticas complexas de forma automática. Por exemplo, número imaginário unitário $\sqrt{-1}$ é pré-atribuído à variável `i`; como consequência, um valor complexo pode ser atribuído como em

```
>> x = 2+i*4
x =
2.0000 + 4.0000i
```

Observe que o MATLAB permite que o símbolo j seja utilizado para representar o número unitário de entrada, ainda que ele sempre use um i para exibir o número. Por exemplo,

```
>> x = 2 + j * 4
x =
2.0000 + 4.0000i
```

Existem diversas variáveis predefinidas, por exemplo, π :

```
>> pi
ans =
3.1416
```

Observe que o MATLAB exibe apenas quatro casas decimais. Caso deseje ser mais preciso, digite o seguinte comando:

```
>> format long
```

Agora, quando π é digitado, o resultado é exibido com 15 algarismos significativos:

```
>> pi
ans =
3.14159265358979
```

Para retornar ao formato com quatro casas decimais, digite

```
>> format short
```

A tabela a seguir apresenta um resumo dos comandos de formato de exibição (de números) rotineiramente empregados em cálculos científicos e de engenharia. Todos eles têm a sintaxe `format tipo`.

Tipo	Resultado	Exemplo
short	Formato de ponto fixo com 5 dígitos	3.1416
long	Formato de ponto fixo com 15 dígitos para precisão dupla e 7 dígitos para precisão simples	3.14159265358979
short e	Formato de ponto flutuante com 5 dígitos	3.1416e+000
long e	Formato de ponto flutuante com 15 dígitos para precisão dupla e 7 dígitos para precisão simples	3.141592653589793e+000
short g	Melhor entre os formatos de ponto fixo e ponto flutuante com 5 dígitos	3.1416
long g	Melhor entre os formatos de ponto fixo e ponto flutuante com 15 dígitos para precisão dupla e 7 dígitos para precisão simples	3.14159265358979
short eng	Formato de engenharia com pelo menos 5 dígitos e uma potência que é um múltiplo de 3	3.1416e+000
long eng	Formato de engenharia com exatamente 16 dígitos significativos e uma potência que é um múltiplo de 3	3.14159265358979e+000
bank	2 dígitos decimais	3.14

2.2.2 Arranjos, vetores e matrizes

Um arranjo é uma coleção de valores representados por um único nome de variável. Podem ser de dois tipos: arranjos unidimensionais são chamados *vetores* e arranjos

bidimensionais são chamados *matrizes*. Em uma abordagem mais geral, os escalares usados na Seção 2.2.1 são, na realidade, matrizes com uma linha e uma coluna.

Para inserir arranjos no prompt de comando, usam-se colchetes. Por exemplo, um vetor linha pode ser atribuído como a seguir:

```
>> a = [1 2 3 4 5]
a =
1 2 3 4 5
```

Observe que essa atribuição substitui a atribuição anterior de $a = 4$.

Quando falamos de vetores, normalmente estamos nos referindo a vetores coluna, mais usados na resolução de problemas matemáticos que os vetores linha. Pode-se inserir um vetor de várias maneiras. Experimente-as.

```
>> b = [2; 4; 6; 8; 10]
```

ou

```
>> b = [2
4
6
8
10]
```

ou, transpondo um vetor linha com o operador ',

```
>> b = [2 4 6 8 10]'
```

O resultado em todos os três casos será

```
b =
2
4
6
8
10
```

Uma matriz de números pode ser atribuída como a seguir:

```
>> A = [1 2 3; 4 5 6; 7 8 9]
A =
1 2 3
4 5 6
7 8 9
```

Também é possível utilizar a tecla Enter para separar as linhas. Por exemplo, no caso a seguir, a tecla Enter seria pressionada depois do 3, do 6 e do] para atribuir a matriz:

```
>> A = [1 2 3
        4 5 6
        7 8 9]
```

Finalmente, a mesma matriz poderia ser construída *concatenando* (isto é, juntando) os vetores que representam cada coluna:

```
>> A = [[1 4 7]' [2 5 8]' [3 6 9]']
```

Em qualquer ponto em uma sessão, uma lista de todas as variáveis correntes pode ser obtida digitando-se o comando `who`:

```
>> who
Your variables are:
A    a    ans    b    x
```

ou, para mais detalhes, insira o comando `whos`:

```
>> whos
  Name      Size            Bytes  Class
  A            3x3             72  double array
  a            1x5              40  double array
  ans          1x1               8  double array
  b            5x1              40  double array
  x            1x1              16  double array (complex)
Grand total is 21 elements using 176 bytes
```

Perceba que a notação de subscrito pode ser usada para acessar um elemento individual de um arranjo. Por exemplo, o quarto elemento do vetor coluna `b` pode ser exibido como

```
>> b(4)
ans =
8
```

Para um arranjo, `A(m, n)` selecione o elemento na m -ésima linha e na n -ésima coluna. Por exemplo,

```
>> A(2, 3)
ans =
6
```

Diversas funções nativas do MATLAB podem ser usadas para criar matrizes, como as funções `ones` e `zeros`, que criam vetores ou matrizes preenchidas 1's e 0's, respectivamente. Ambas têm dois argumentos: o primeiro para o número de linhas e o segundo para o número de colunas. Por exemplo, para criar uma matriz 2×3 de 0's:

```
>> E = zeros(2, 3)
E =
  0     0     0
  0     0     0
```

De forma similar, a função `ones` pode ser usada para criar um vetor linha de 1's:

```
>> u = ones(1, 3)
u =
  1     1     1
```

2.2.3 O operador dois-pontos

O operador dois-pontos (`:`) é uma ferramenta poderosa para criação e manipulação de arranjos. Se dois-pontos são usados para separar dois números, o MATLAB gera um conjunto de números entre eles utilizando um incremento unitário:

```
>> t = 1:5
t =
1     2     3     4     5
```

Se o operador dois-pontos for usado para separar três números, o MATLAB gerará um conjunto de números entre o primeiro e o terceiro números utilizando um incremento igual ao segundo número:

```
>> t = 1:0.5:3
t =
1.0000    1.5000    2.0000    2.5000    3.0000
```

Observe que incrementos negativos (às vezes chamados decrementos) também podem ser usados:

```
>> t = 10:-1:5
t =
10     9     8     7     6     5
```

Além de criar uma série de números, o operador dois-pontos também pode ser usado como um “curinga” para selecionar linhas e colunas individuais de uma matriz. Quando um operador dois-pontos é usado em lugar de um subscrito específico, o dois-pontos representa a linha ou coluna inteira. A segunda linha da matriz A a seguir, por exemplo, pode ser selecionada como em

```
>> A(2,:)
ans =
4     5     6
```

Também possível usar a notação de dois-pontos para extrair seletivamente uma série de elementos de dentro de um arranjo. Por exemplo, com base na definição anterior do vetor t :

```
>> t(2:4)
ans =
9     8     7
```

Assim, o segundo até o quarto elementos são retornados.

2.2.4 As funções linspace e logspace

As funções `linspace` e `logspace` fornecem outras ferramentas úteis para gerar vetores de pontos espaçados. A função `linspace` gera um vetor linha de pontos igualmente – e linearmente – espaçados. Ele tem a forma

```
linspace(x1, x2, n)
```

que gera n pontos entre $x1$ e $x2$. Por exemplo

```
>> linspace(0,1,6)
ans =
0     0.2000    0.4000    0.6000    0.8000    1.0000
```

Se n for omitido, a função gerará automaticamente 100 pontos.

A função `logspace` gera um vetor linha de pontos logaritmicamente espaçados. Ela tem a forma

```
logspace(x1, x2, n)
```

que gera n pontos logaritmicamente espaçados entre as décadas 10^{x1} e 10^{x2} . Por exemplo,

```
>> logspace(-1, 2, 4)
ans =
    0.1000    1.0000   10.0000  100.0000
```

Se n é omitido, a função gera automaticamente 50 pontos.

2.2.5 Sequência de caracteres (strings)

Além dos números, informações *alfanuméricas* ou *sequências de caracteres* ou *strings* podem ser representadas colocando-se a sequência de caracteres dentro de aspas simples. Por exemplo,

```
>> f = 'Miles ';
>> s = 'Davis';
```

Cada caractere em uma string é um elemento em um arranjo; assim, podemos *concatenar* ("colar") strings como em

```
>> x = [f s]
x =
Miles Davis
```

Observe que linhas muito longas podem ser continuadas colocando-se *reticências* no fim da linha. Poderíamos, por exemplo, inserir um vetor linha da seguinte forma

```
>> a = [1 2 3 4 5 ...
6 7 8]
a =
    1     2     3     4     5     6     7     8
```

Não é permitido, no entanto, utilizar reticências para continuar uma string. Para entrar com uma string que se estende além de uma única linha, junte strings mais curtas, como em

```
>> citacao = ['Qualquer idiota pode fazer uma regra,' ...
'e qualquer idiota a seguirá']
citacao =
Qualquer idiota pode fazer uma regra, e qualquer idiota
a seguirá
```

2.3 OPERAÇÕES MATEMÁTICAS

Operações com quantidades escalares são tratadas de modo similar a outras linguagens de computador. Os operadores comuns, em ordem de prioridade, são

\wedge	Exponenciação
$-$	Negação
$* /$	Multiplicação e divisão
\backslash	Divisão à esquerda ²
$+ -$	Adição e subtração

Esses operadores funcionam como em uma calculadora. Digite

```
>> 2*pi
ans =
6.2832
```

Variáveis escalares reais também podem ser incluídas:

```
>> y = pi/4;
>> y^2.45
ans =
0.5533
```

Os resultados dos cálculos podem ser atribuídos a uma variável ou simplesmente exibidos na tela.

Assim como em outros cálculos de computador, a ordem de prioridade das operações pode ser modificada com parênteses; dessa forma, como a exponenciação tem maior prioridade que a negação, por exemplo, o seguinte resultado seria obtido:

```
>> y = -4^2
y =
-16
```

Assim, 4 é primeiro elevado ao quadrado e depois negado. Os parênteses também podem ser usados para alterar as prioridades, como em

```
>> y = (-4)^2
y =
16
```

Os cálculos também podem envolver quantidades complexas, como os exemplos a seguir, que utilizam os valores de $x (2 + 4i)$ e $y (16)$ definidos anteriormente:

```
>> 3 * x
ans =
6.0000 + 12.0000i
>> 1 / x
ans =
0.1000 - 0.2000i
>> x ^ 2
ans =
-12.0000 + 16.0000i
>> x + y
ans =
18.0000 + 4.0000i
```

² A divisão à esquerda se aplica à álgebra matricial, que será discutida em detalhes mais adiante neste livro.

Um dos poderes do MATLAB é sua capacidade de executar cálculos com vetores e matrizes. Embora tais cálculos sejam descritos em detalhes no Capítulo 8, consideramos importante mostrar alguns exemplos aqui.

O *produto interno* de dois vetores (produto escalar) pode ser calculado com o operador `*`. Considerando, por exemplo, o vetor linha `a = [1 2 3 4 5]` e o vetor coluna `b = [2 4 6 8]'`, tem-se:

```
>> a * b
ans =
    110
```

e, do mesmo modo, o *produto externo*

```
>> b * a
ans =
    2      4      6      8      10
    4      8     12     16     20
    6     12     18     24     30
    8     16     24     32     40
   10     20     30     40     50
```

Para melhor ilustrar a multiplicação de vetores e matrizes, primeiro redefina `a` e `b`:

```
>> a = [1 2 3];
e
>> b = [4 5 6]';
```

Agora, teste

```
>> a * A
ans =
    30      36      42
```

ou

```
>> A * b
ans =
    32
    77
   122
```

As matrizes não podem ser multiplicadas se as dimensões internas forem diferentes, o que pode ser ilustrado pelo exemplo a seguir. Teste

```
>> A * a
```

O MATLAB exibe automaticamente a mensagem de erro:

```
??? Error using ==> mtimes
Inner matrix dimensions must agree.
```

A multiplicação de matrizes é efetuada da mesma forma

```
>> A * A
ans =
    30      36      42
    66      81      96
   102     126     150
```

Operações mistas com escalares também são possíveis:

```
>> A/pi
ans =
    0.3183    0.6366    0.9549
    1.2732    1.5915    1.9099
    2.2282    2.5465    2.8648
```

Lembre sempre que, se possível, o MATLAB aplicará os operadores aritméticos simples no modo padrão de operações com vetores e matrizes.* Às vezes, pode ser necessário executar cálculos item por item em uma matriz ou vetor; felizmente, o MATLAB também fornece recursos para isso. Por exemplo,

```
>> A.^2
ans =
    30      36      42
    66      81      96
   102     126     150
```

resulta na multiplicação matricial de A por ela mesma.

E se quisermos o quadrado de cada elemento de A ? O resultado pode ser obtido com o comando

```
>> A.^2
ans =
    1      4      9
   16     25     36
   49     64     81
```

O $.$ que precede o operador $^$ significa que a operação deve ser executada elemento por elemento. O manual do MATLAB chama tais operações de *operações de arranjos*, também denominadas *operações elemento por elemento*.

O MATLAB possui um atalho útil para realização de cálculos já efetuados: as teclas de navegação (\uparrow) e (\downarrow) podem ser acionadas para recuperar comandos digitados anteriormente. Por exemplo, pressionando a tecla \uparrow (seta para cima), recupera-se a última linha digitada.

```
>> A.^2
```

Pressionando Enter, o cálculo é efetuado novamente, porém, também é possível editar a linha. Como um teste, altere-a para a linha a seguir e pressione Enter.

```
>> A.^3
ans =
    1      8      27
   64     125    216
  343     512    729
```

Utilizando a tecla \uparrow , é possível retornar a qualquer comando já digitado. Pressione essa tecla até retornar à linha

```
>> b * a
```

* N. de T.: Por modo padrão, entenda-se o modo como as operações são definidas do ponto de vista formal da matemática.

Também é possível recuperar o comando ao digitar `b` e pressionar a tecla \uparrow uma vez; então, o último comando iniciado pela letra `b` será automaticamente exibido. A tecla \uparrow possibilita uma maneira rápida de corrigir erros sem ser necessário redigitar toda a linha de comando.

2.4 UTILIZAÇÃO DE FUNÇÕES NATIVAS

O MATLAB e seus Toolboxes têm uma vasta coleção de funções nativas, que podem ser exploradas com o comando Help (Ajuda) do programa. Por exemplo, se quisermos aprender sobre a função `log`, digita-se

```
>> help log
LOG    Natural logarithm.
LOG(X) is the natural logarithm of the elements of X.
Complex results are produced if X is not positive.

See also LOG2, LOG10, EXP, LOGM.
```

Para uma lista de todas as funções elementares, digite

```
>> help elfun
```

Uma das propriedades importantes das funções nativas do MATLAB é que elas operam diretamente em quantidades vetoriais e matriciais. Por exemplo, digite

```
>> log(A)
ans =
    0      0.6931      1.0986
    1.3863     1.6094      1.7918
    1.9459     2.0794      2.1972
```

e perceba que a função logaritmo natural é aplicada em estilo arranjo, elemento por elemento, para a matriz `A` – vale dizer que a maioria das funções, tais como `sqrt`, `abs`, `sin`, `acos`, `tanh` e `exp`, opera em estilo arranjo. Certas funções, tais como a exponencial e a raiz quadrada, também possuem definições matriciais, que podem ser calculadas quando a letra `m` é acrescentada ao nome da função. Por exemplo,

```
>> sqrtm(A)
ans =
    0.4498 + 0.7623i    0.5526 + 0.2068i    0.6555 - 0.3487i
    1.0185 + 0.0842i    1.2515 + 0.0228i    1.4844 - 0.0385i
    1.5873 - 0.5940i    1.9503 - 0.1611i    2.3134 + 0.2717i
```

Há diversas funções para arredondamento. Considere, por exemplo, que entremos com um vetor:

```
>> E = [-1.6 -1.5 -1.4 1.4 1.5 1.6];
```

A função `round` arredonda os elementos de `E` para os inteiros mais próximos:

```
>> round(E)
ans =
    -2      -2      -1      1      2      2
```

A função `ceil` arredonda para os inteiros mais próximos em direção ao infinito positivo:

```
>> ceil(E)
ans =
-1      -1      -1      2      2      2
```

A função `floor` arredonda para baixo, para os inteiros mais próximos em direção ao infinito negativo:

```
>> floor(E)
ans =
-2      -2      -2      1      1      1
```

Existem também funções que realizam operações especiais com os elementos de matrizes e arranjos. A função `sum`, por exemplo, retorna a soma dos elementos:

```
>> F = [3 5 4 6 1];
>> sum(F)
ans =
19
```

De modo similar, deve ser óbvio o que acontece com os comandos a seguir:

```
>> min(F), max(F), mean(F), prod(F), sort(F)
ans =
1
ans =
6
ans =
3.8000
ans =
360
ans =
1      3      4      5      6
```

As funções são muito usadas na avaliação de fórmulas para uma série de argumentos. Lembre-se que a velocidade de um saltador de *bungee jumping* em queda livre pode ser calculada com [Equação (1.9)]:

$$v = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right)$$

onde v é a velocidade (m/s), g é a aceleração devida à gravidade ($9,81 \text{ m/s}^2$), m é a massa (kg), c_d é o coeficiente de arraste (kg/m) e t é o tempo (s).

Crie um vetor coluna t que contenha valores de 0 a 20 em passos de 2:

```
>> t = [0:2:20]'
t =
0
2
4
6
8
10
12
14
```

```
16  
18  
20
```

Verifique o número de elementos no arranjo t com a função `length`:

```
>> length(t)  
ans =  
11
```

Atribua valores aos parâmetros:

```
>> g = 9.81; m = 68.1; cd = 0.25;
```

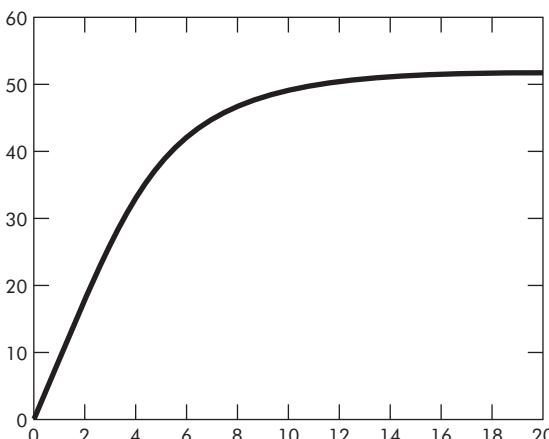
O MATLAB permite avaliar uma fórmula tal como $v = f(t)$, onde a fórmula é calculada para cada valor do arranjo t , e o resultado é atribuído para uma posição correspondente no arranjo v . Para o caso analisado,

```
>> v = sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t)  
v =  
0  
18.7292  
33.1118  
42.0762  
46.9575  
49.4214  
50.6175  
51.1871  
51.4560  
51.5823  
51.6416
```

2.5 GRÁFICOS

Com o MATLAB, é possível criar gráficos de forma rápida e conveniente. Para criar um gráfico dos arranjos t e v a partir dos dados anteriores, digite

```
>> plot(t, v)
```



É possível personalizar um pouco o gráfico com os comandos a seguir:

```
>> title('Gráfico de v versus t')
>> xlabel('Valores de t')
>> ylabel('Valores de v')
>> grid
```



O comando `plot` exibe, por padrão (default), uma fina linha sólida de cor azul. Caso o usuário decida plotar cada ponto com um símbolo (marcador), pode-se incluir um especificador delimitado por aspas simples no comando `plot`. A Tabela 2.2 mostra os especificadores disponíveis. Para um marcador do tipo círculos abertos, por exemplo, digite

```
>> plot(t, v, 'o')
```

É possível combinar diversos especificadores: caso se queira utilizar marcadores quadrados verdes, conectados por linhas tracejadas verdes, pode-se inserir o comando

```
>> plot(t, v, 's--g')
```

Também é possível controlar a largura da linha, assim como o tamanho do marcador e as cores das bordas e do preenchimento. O comando a seguir, por exemplo, utiliza uma linha mais larga (largura 2 pontos), tracejada e cor ciano, para conectar marcadores maiores (10 pontos), com forma de diamante, bordas pretas e preenchimento de cor magenta:

```
>> plot (x, y, '--dc', 'LineWidth', 2, ...
'MarkerSize', 10, ...
'MarkerEdgeColor', 'k', ...
'MarkerFaceColor', 'm')
```

TABELA 2.2 Especificadores de cores, símbolos (ou marcadores) e tipos de linha.

Cores		Símbolos		Tipos de linha	
Azul	b	Ponto	.	Sólida	-
Verde	g	Círculo	o	Pontilhada	:
Vermelho	r	Símbolo X	x	Traço-ponto	-.
Ciano	c	Mais	+	Tracejada	--
Magenta	m	Estrela	*		
Amarelo	y	Quadrado	s		
Preto	k	Diamante	d		
Branco	w	Triângulo (para baixo)	v		
		Triângulo (para cima)	^		
		Triângulo (para esquerda)	<		
		Triângulo (para direita)	>		
		Pentagrama (estrela de cinco pontas)	p		
		Hexagrama (estrela de seis pontas)	h		

Observe que a largura de linha padrão é um ponto; para os marcadores, o padrão são seis pontos com borda azul e sem preenchimento (cor branca).

O MATLAB permite exibir mais de um conjunto de dados no mesmo gráfico. Por exemplo, uma maneira alternativa de conectar cada marcador de ponto com uma linha reta seria digitar

```
>> plot(t, v, t, v, 'o')
```

Deve-se mencionar que, inevitavelmente, os gráficos anteriores são apagados toda vez que o comando `plot` é executado. O comando `hold on` mantém o gráfico corrente e todas as propriedades dos eixos, de modo que comandos gráficos adicionais possam ser acrescentados ao gráfico existente; já o comando `hold off` retorna ao modo padrão. Por exemplo, se tivéssemos digitado os comandos a seguir, o gráfico final exibiria apenas marcadores:

```
>> plot(t, v)
>> plot(t, v, 'o')
```

Os comandos a seguir, por sua vez, resultariam em um gráfico que exibiria tanto as linhas quanto os marcadores:

```
>> plot(t, v)
>> hold on
>> plot(t, v, 'o')
>> hold off
```

Além do `hold`, outro comando útil é o `subplot`, que permite dividir a janela do gráfico em subjanelas ou *painéis*, e tem a sintaxe:

```
subplot(m, n, p)
```

Esse comando divide a janela do gráfico em uma matriz *m* por *n*, com gráficos menores em cada célula da “matriz”, e seleciona o *p*-ésimo gráfico como o corrente.

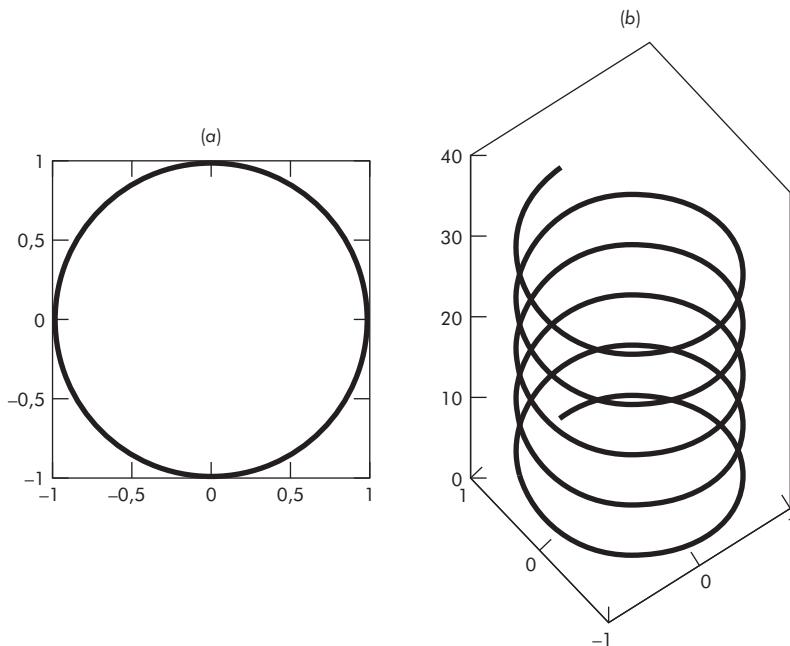


FIGURA 2.1 Gráfico de dois painéis de (a) um círculo bidimensional e (b) uma hélice tridimensional.

Podemos demonstrar o comando `subplot` por meio de um comando simples para gerar gráficos tridimensionais, o `plot3`, que tem a sintaxe:

```
plot3(x, y, z)
```

onde x , y e z são três vetores de mesmo comprimento. O resultado é uma linha no espaço tridimensional através dos pontos cujas coordenadas são os elementos de x , y e z .

O gráfico de uma hélice fornece um bom exemplo para ilustrar essa teoria. Primeiro, vamos traçar um círculo com a função bidimensional `plot` utilizando a representação paramétrica $x = \sin(t)$ e $y = \cos(t)$. Empregaremos o comando `subplot` para, em seguida, acrescentar o gráfico tridimensional.

```
>> t = 0:pi/50:10*pi;
>> subplot(1, 2, 1); plot(sin(t),cos(t))
>> axis square
>> title('(a)')
```

Como mostra a Figura 2.1a, o resultado é um círculo, que poderia ter sido distorcido se não tivéssemos usado o comando `axis square`.

Agora, vamos acrescentar a hélice ao painel da direita na janela do gráfico: para isso, empregaremos novamente uma representação paramétrica: $x = \sin(t)$ e $y = \cos(t)$ e $z = t$.

```
>> subplot(1, 2, 2); plot3(sin(t),cos(t),t);
>> title('(b)')
```

O resultado está ilustrado na Figura 2.1b. Você é capaz de dizer o que está acontecendo? À medida que o tempo evolui, as coordenadas x e y traçam a circunferência do círculo no plano x - y da mesma forma como o gráfico bidimensional; no entanto, a curva sobe à medida que a coordenada z aumenta linearmente com o tempo. O resultado líquido é uma mola característica ou uma escadaria em espiral na forma de hélice.

Existem outras características gráficas úteis do MATLAB – por exemplo, traçar objetos em vez de linhas, famílias de curvas, traçar gráficos no plano complexo, gráficos log-log e/ou semilog, gráficos de malhas tridimensionais e gráficos de curvas de nível. Alguns desses recursos serão descritos a seguir.

2.6 OUTROS RECURSOS

Do que foi visto até agora, focamos nos recursos do MATLAB que serão mais utilizados no restante deste livro. O leitor interessado em aprender mais sobre outras capacidades do software pode consultar alguns excelentes livros dedicados ao MATLAB (por exemplo, Attaway, 2009; Palm, 2007; Hanselman e Littlefield, 2005; e Moore, 2008).

O próprio MATLAB possui um extenso recurso de Help (Ajuda) que pode ser acessado no menu Help da command window e que oferece acesso a uma série de demonstrações bastante didáticas.

Como descrito neste capítulo, também há ajuda disponível em um modo interativo: basta digitar o comando `help`, seguido pelo nome da função ou do comando. Se o nome da função for desconhecido, pode-se utilizar o comando `lookfor` para procurar nos arquivos de Ajuda do MATLAB por ocorrências de texto (no caso, o texto deve ser digitado em inglês). Por exemplo, considere que se queira procurar todos os comandos e funções relacionados a logaritmos. Ao inserir o comando

```
>> lookfor logarithm
```

o MATLAB exibe todas as referências que incluem a palavra `logarithm`.

Por fim, pode-se obter ajuda no site da Math Works Inc., www.mathworks.com, onde há informações de produtos, grupos de discussão, livros, suporte técnico, além de muitos outros recursos úteis.

► 2.7 ESTUDO DE CASO ANÁLISE EXPLORATÓRIA DE DADOS

Contextualização. Os livros didáticos são repletos de fórmulas desenvolvidas no passado por cientistas e engenheiros renomados. Embora elas sejam de grande utilidade, engenheiros e cientistas muitas vezes devem completar essas relações coletando e analisando seus próprios dados, o que pode levar a uma nova fórmula. Porém, antes de se chegar a uma equação final, geralmente “brincamos” com os dados efetuando cálculos e traçando gráficos, a fim de obter uma visão dos padrões e mecanismos escondidos nos dados.

Neste estudo de caso, ilustraremos como o MATLAB facilita essa análise exploratória de dados ao estimar o coeficiente de arraste de um indivíduo em queda livre com base na Equação (2.1) e nos dados da Tabela 2.1. Além de simplesmente calcular o coeficiente

de arraste, utilizaremos também as capacidades gráficas do MATLAB para discernir padrões nos dados.

Solução. Pode-se entrar com os dados da Tabela 2.1 e com a aceleração da gravidade, como a seguir:

```
>> m=[83.6 60.2 72.1 91.1 92.9 65.3 80.9];
>> vt=[53.4 48.5 50.9 55.7 54 47.7 51.1];
>> g=9.81;
```

O coeficiente de arraste pode, então, ser calculado com a Equação (2.1). Como estamos realizando operações elemento por elemento nos vetores, é necessário incluir pontos antes dos operadores:

```
>> cd=g*m./vt.^2
cd =
0.2876    0.2511    0.2730    0.2881    0.3125    0.2815    0.3039
```

Podemos agora utilizar algumas funções nativas do MATLAB a fim de gerar estatísticas para os resultados:

```
>> cd_medio=mean(cd),cd_min=min(cd),cd_max=max(cd)
cd_medio =
0.2854
cd_min =
0.2511
cd_max =
0.3125
```

Assim, o valor médio é 0,2854 com um intervalo de 0,2511 a 0,3125 kg/m.

Agora vamos “brincar” com esses dados utilizando a Equação (2.1) para fazer uma predição da velocidade terminal com base no arraste médio:

```
>> vpred=sqrt(g*m/cd_medio)
vpred =
53.6065    45.4897    49.7831    55.9595    56.5096    47.3774    52.7338
```

Observe que não tivemos de usar pontos antes dos operadores nessa fórmula. Você entende por quê?

Podemos plotar esses valores *versus* as velocidades terminais reais medidas, e podemos também sobrepor uma reta que indique as previsões exatas (a reta 1:1) para auxiliar na avaliação dos resultados. Como será gerado um segundo gráfico, empregamos o comando subplot:

```
>> subplot(2,1,1); plot(vt,vpred,'o',vt,vt)
>> xlabel('Medido')
>> ylabel('Previsto')
>> title('Gráfico das velocidades previstas versus medidas')
```



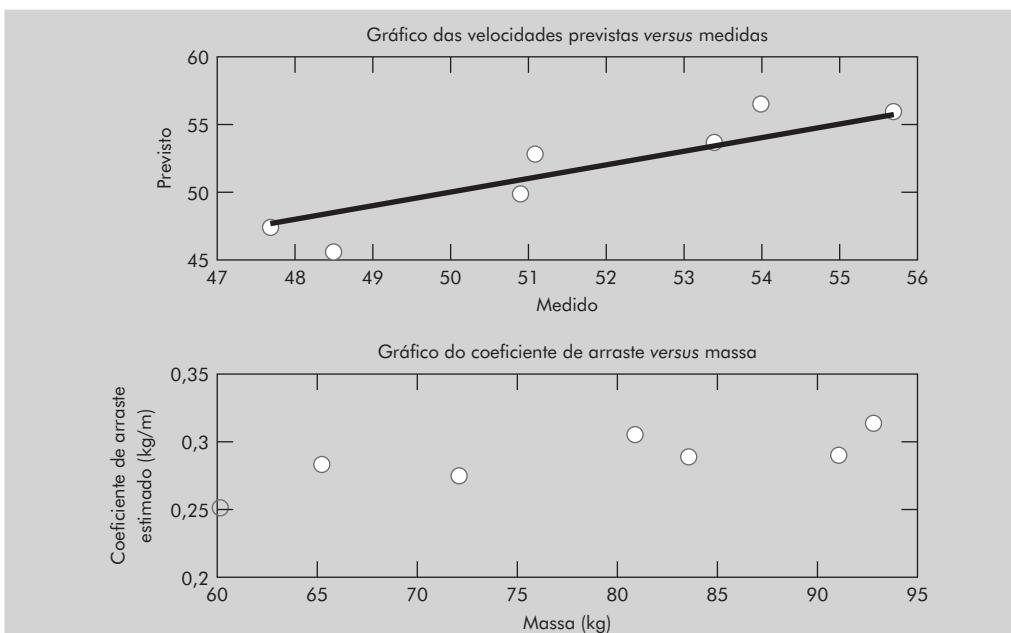


FIGURA 2.2 Dois gráficos criados com o MATLAB.

De acordo com o primeiro gráfico da Figura 2.2, como as previsões geralmente seguem a reta 1:1, você poderia inicialmente concluir que o coeficiente de arraste médio produz bons resultados. Observe, contudo, como o modelo tende a prever valores menores que os reais na faixa de baixas velocidades e prever valores maiores na faixa de altas velocidades. Isso sugere que, em vez de serem constantes, pode existir uma tendência nos coeficientes de arraste, o que pode ser verificado com um gráfico dos coeficientes de arraste *versus* a massa:

```
>> subplot(2,1,2); plot(m,cd,'o')
>> xlabel('Massa (kg)')
>> ylabel('Coeficiente de arraste estimado (kg/m)')
>> title('Gráfico do coeficiente de arraste versus massa')
```

O segundo gráfico da Figura 2.2 sugere que, em vez de ser constante, o coeficiente de arraste parece crescer à medida que a massa do saltador aumenta. Com base nesse resultado, você poderia concluir que seu modelo precisa ser melhorado; no mínimo, ele poderia motivá-lo a realizar experimentos adicionais com um maior número de saltadores para confirmar sua conclusão preliminar.

Além disso, o resultado poderia estimulá-lo a procurar a literatura de mecânica de fluidos e aprender mais sobre o coeficiente de arraste. Como descrito na Seção 1.4, você descobriria que o parâmetro c_d é, na realidade, um coeficiente de arraste concentrado que, com o arraste verdadeiro, inclui outros fatores tais como a área frontal do saltador e a densidade do ar,

$$c_d = \frac{C_D \rho A}{2} \quad (2.2)$$

onde C_D é um coeficiente de arraste adimensional, ρ é a densidade do ar (kg/m^3) e A é a área frontal (m^2), projetada sobre um plano normal à direção da velocidade.

Assumindo que as densidades fossem relativamente constantes durante a coleta dos dados (uma boa suposição se todos os saltadores tivessem pulado da mesma altura e no mesmo dia), a Equação (2.2) sugere que saltadores mais pesados devem ter áreas maiores – tal hipótese poderia ser comprovada por meio da medição das áreas frontais de indivíduos de diferentes massas.

►► PROBLEMAS

2.1 Use a função `linspace` para criar vetores idênticos aos abaixo criados com a notação dois-pontos:

$$(a) \quad t = 4:6:35$$

$$(b) \quad x = -4:2$$

2.2 Use a notação dois-pontos para criar vetores idênticos aos abaixo criados com a função `linspace`:

$$(a) \quad v = \text{linspace}(-2, 1.5, 8)$$

$$(b) \quad r = \text{linspace}(8, 4.5, 8)$$

2.3 O comando `linspace(a, b, n)` gera um vetor linha de n pontos igualmente espaçados entre a e b . Use a notação dois-pontos para escrever uma linha de comando alternativa a fim de gerar o mesmo vetor. Teste sua formulação para $a = -3$, $b = 5$, $n = 6$.

2.4 Considere que se entre com a matriz a seguir no MATLAB:

```
>> A=[3 2 1;0:0.5:1;linspace(6, 8, 3)]
```

(a) Escreva a matriz resultante.

(b) Use a notação dois-pontos para escrever uma única linha de comando do MATLAB a fim de multiplicar a segunda linha pela terceira coluna e atribuir o resultado à variável `c`.

2.5 A equação a seguir pode ser usada para calcular valores de y como uma função de x ,

$$y = be^{-ax} \sin(bx)(0,012x^4 - 0,15x^3 + 0,075x^2 + 2,5x)$$

onde a e b são parâmetros. Escreva a equação para ser calculada no MATLAB, onde $a = 2$, $b = 5$ e x é um vetor com valores de 0 a $\pi/2$ com incrementos de $\Delta x = \pi/40$. Empregue pontos seguidos dos operadores em sua formulação, de modo que ela produza um vetor para y . Calcule também o vetor $z = y^2$, onde cada elemento corresponda ao quadrado de cada elemento de y . Combine x , y e z em uma matriz w , em que cada coluna corresponda a uma variável, e exiba w utilizando o formato

`short g`. Além disso, gere um gráfico de y e z versus x com os eixos identificados e inclua uma legenda no gráfico (use a Ajuda do MATLAB para aprender como fazer isso). Para y , use uma linha traço-ponto vermelha com largura de 1,5 ponto com marcadores na forma de pentágonos de tamanho 14 pontos, sem preenchimento e bordas vermelhas. Para z , use uma linha sólida azul de largura-padrão com marcadores quadrados de tamanho-padrão, com preenchimento verde e bordas azuis.

2.6 Um circuito elétrico simples consistindo de um resistor, um capacitor e um indutor está ilustrado na Figura P2.6. A carga no capacitor $q(t)$ como uma função do tempo pode ser calculada como

$$q(t) = q_0 e^{-Rt/(2L)} \cos \left[\sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2} t \right]$$

onde t é o tempo, q_0 é a carga inicial, R é a resistência, L é a indutância e C é a capacitância. Use o MATLAB para gerar um gráfico dessa função de $t = 0$ a 0,8, dado que $q_0 = 10$, $R = 60$, $L = 9$ e $C = 0,00005$.

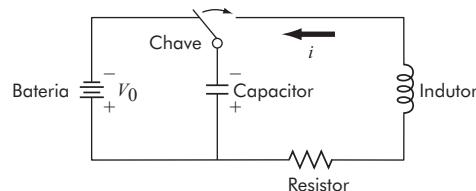


FIGURA P2.6

2.7 A função de densidade de probabilidade normal é uma curva em forma de sino que pode ser representada como

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$$

Use o MATLAB para gerar um gráfico dessa função de $z = -5$ até 5. Identifique as ordenadas como frequência e as abscissas como z .

2.8 Se uma força F (N) for aplicada para comprimir uma mola, seu deslocamento x (m) poderá ser modelado pela lei de Hooke,

$$F = kx$$

onde k é a constante da mola (N/m). A energia potencial armazenada na mola U (J) pode ser calculada como

$$U = \frac{1}{2}kx^2$$

Cinco molas foram testadas e os seguintes dados obtidos:

F, N	14	18	8	9	13
x, m	0,013	0,020	0,009	0,010	0,012

Use o MATLAB para armazenar F e x como vetores e, então, calcule vetores das constantes das molas e das energias potenciais. Use a função \max para determinar a energia potencial máxima.

2.9 A densidade da água doce pode ser calculada como uma função da temperatura com a seguinte equação cúbica

$$\rho = 5,5289 \times 10^{-8}T_C^3 - 8,5016 \times 10^{-6}T_C^2 + 6,5622 \times 10^{-5}T_C + 0,99987$$

onde ρ é a densidade (g/cm^3) e T_C é a temperatura ($^\circ\text{C}$). Use o MATLAB para gerar um vetor de temperaturas variando de 32 $^\circ\text{F}$ a 93,2 $^\circ\text{F}$, com incrementos de 3,6 $^\circ\text{F}$. Converta esse vetor para graus Celsius e, então, calcule um vetor de densidades baseado na fórmula cúbica. Crie um gráfico de ρ versus T_C . Lembre-se que $T_C = 5/9(T_F - 32)$.

2.10 A equação de Manning pode ser usada para calcular a velocidade da água em um canal aberto retangular,

$$U = \frac{\sqrt{S}}{n} \left(\frac{BH}{B+2H} \right)^{2/3}$$

onde U é a velocidade (m/s), S é a inclinação do canal, n é o coeficiente de rugosidade, B é a largura (m) e H é a profundidade (m). Os seguintes dados estão disponíveis para cinco canais:

n	S	B	H
0,035	0,0001	10	2
0,020	0,0002	8	1
0,015	0,0010	20	1,5
0,030	0,0007	24	3
0,022	0,0003	15	2,5

Armazene esses valores em uma matriz onde cada linha represente um dos canais e cada coluna represente um dos parâmetros. Escreva uma única linha de comando no MATLAB para calcular um vetor coluna que contenha as velocidades, com base nos valores na matriz de parâmetros.

2.11 É uma prática geral na engenharia e ciência plotar equações utilizando linhas e dados discretos com símbolos (marcadores). A tabela a seguir apresenta alguns dados da concentração (c) versus tempo (t) para a fotodegradação de bromo aquoso:

t, min	10	20	30	40	50	60
c, ppm	3,4	2,6	1,6	1,3	1,0	0,5

Esses dados podem ser descritos pela seguinte função:

$$c = 4,84e^{-0,034t}$$

Use o MATLAB para criar um gráfico que exiba tanto os dados (utilize marcadores com forma de diamante e preenchimento vermelho) quanto a função (utilize linha tracejada verde). Trace a função para $t = 0$ a 70 min.

2.12 A função `semilogy` funciona de maneira idêntica à função `plot`, com a exceção de que uma escala logarítmica (base 10) é usada para o eixo y . Use essa informação para plotar os dados e a função como descrito no Problema 2.11. Explique os resultados.

2.13 A tabela a seguir apresenta alguns dados de um túnel de vento para força (F) versus velocidade (v):

v, m/s	10	20	30	40	50	60	70	80
F, N	25	70	380	550	610	1220	830	1450

Esses dados podem ser descritos pela seguinte função:

$$F = 0,2741v^{1,9842}$$

Use o MATLAB para criar um gráfico que exiba tanto os dados (utilize marcadores circulares com preenchimento magenta) quanto a função (utilize linha traço-ponto preta). Trace a função para $v = 0$ a 100 m/s e identifique os eixos do gráfico.

2.14 A função `loglog` funciona de maneira idêntica à função `plot`, com a exceção de que escalas logarítmicas são usadas para ambos os eixos x e y . Use essa função para plotar os dados e a função como descrito no Problema 2.13. Explique os resultados.

2.15 A expansão em série de Maclaurin para o cosseno é:

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

Use o MATLAB para criar um gráfico do cosseno (linha sólida) com um gráfico da expansão em série (linha tracejada preta) até o termo $x^8/8!$. Use a função nativa `factorial` no cálculo da expansão em série, e para as abscissas utilize o intervalo de $x = 0$ até $3\pi/2$.

2.16 Você entrou em contato com os saltadores usados para gerar os dados na Tabela 2.1 e mediu suas áreas frontais. Os valores resultantes, que estão ordenados na mesma sequência que a dos valores correspondentes na Tabela 2.1, são

A, m² 0,455 0,402 0,452 0,486 0,531 0,475 0,487

- (a) Se a densidade do ar for $\rho = 1,223 \text{ kg/m}^3$, use o MATLAB para calcular valores do coeficiente de arraste adimensional C_D .
- (b) Para os valores resultantes da tabela calcule a média, o mínimo e o máximo.
- (c) Utilizando o comando `subplot`, faça um gráfico de A versus m (parte superior) e C_D versus m (parte inferior). Inclua identificações descriptivas para os eixos e títulos para os gráficos.

2.17 As seguintes equações paramétricas geram uma hélice cônica:

$$\begin{aligned} x &= t \cos(6t) \\ y &= t \sin(6t) \\ z &= t \end{aligned}$$

Calcule os valores de x , y e z para $t = 0$ a 6π com $\Delta t = \pi/64$. Use o comando `subplot` para gerar um gráfico bidimensional (linha sólida vermelha) de (x, y) na parte superior e um gráfico tridimensional (linha sólida ciano) de (x, y, z) na parte inferior. Identifique os eixos para ambos os gráficos.

2.18 O que será exibido após digitar os seguintes comandos de MATLAB?

- (a)

```
>> x = 5;
>> x ^ 3;
>> y = 8 - x
```
- (b)

```
>> q = 4:2:12;
>> r = [7 8 4; 3 6 -5];
>> sum(q) * r(2, 3)
```

2.19 A trajetória de um objeto pode ser modelada como

$$y = (\tan \theta_0)x - \frac{g}{2v_0^2 \cos^2 \theta_0} x^2 + y_0$$

onde y é a altura (m), θ_0 é o ângulo inicial (radianos), x é a distância horizontal (m), g é a aceleração da gravidade ($= 9,81 \text{ m/s}^2$), v_0 é a velocidade inicial (m/s) e y_0 é a altura inicial. Use o MATLAB para encontrar as trajetórias para $y_0 = 0$ e $v_0 = 28 \text{ m/s}$ para ângulos iniciais de 15° a 75° em incrementos de 15° . Empregue uma faixa de distâncias horizontais de $x = 0$ a 80 m em incrementos de 5 m . Os resultados devem ser reunidos em um arranjo onde a primeira dimensão (linhas) corresponda às distâncias e a segunda dimensão (colunas) corresponda aos diferentes ângulos iniciais. Use essa matriz para gerar um único gráfico das alturas *versus* as distâncias horizontais para cada um dos ângulos iniciais. Empregue uma legenda para distinguir os diferentes casos e dimunsione a escala do gráfico, utilizando o comando `axis`, de modo que a altura mínima seja zero.

2.20 A dependência da temperatura de reações químicas pode ser calculada com a *equação de Arrhenius*,

$$k = Ae^{-E/(RT_a)}$$

onde k é a constante de velocidade (s^{-1}), A é o fator pré-exponencial (ou frequência), E é a energia de ativação (J/mol), R é a constante dos gases [$8,314 \text{ J/(mol} \cdot \text{K)}$] e T_a é a temperatura absoluta. Um composto tem $E = 1 \times 10^5 \text{ J/mol}$ e $A = 7 \times 10^{16}$. Use o MATLAB para gerar valores da constante de velocidade para temperaturas variando de $253 \text{ a } 325 \text{ K}$. Utilize o comando `subplot` para gerar dois gráficos lado a lado de (a) k versus T_a (linha verde) e (b) $\log_{10} k$ (linha vermelha) versus $1/T_a$. Empregue a função `semilogy` para criar (b) e inclua identificações dos eixos e títulos para ambos os gráficos. Interprete seus resultados.

2.21 A Figura P2.21a mostra uma viga uniforme sujeita a uma carga distribuída de forma linearmente crescente. Como descrito na Figura P2.21b, a deflexão y (m) pode ser calculada com

$$y = \frac{w_0}{120EIL}(-x^5 + 2L^2x^3 - L^4x)$$

onde E é o módulo de elasticidade e I é o momento de inércia (m^4). Empregue essa equação e o cálculo para gerar gráficos no MATLAB das seguintes quantidades *versus* a distância ao longo da viga:

- (a) deslocamento (y),
- (b) inclinação [$\theta(x) = dy/dx$],
- (c) momento [$M(x) = EI d^2y/dx^2$],
- (d) cisalhamento [$V(x) = EI d^3y/dx^3$], e
- (e) carregamento [$w(x) = -EI d^4y/dx^4$].

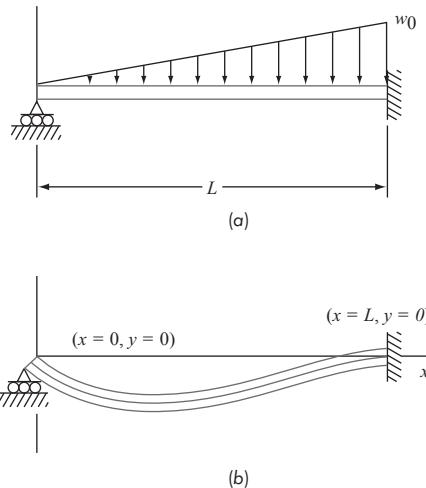


FIGURA P2.21

Use os seguintes parâmetros para seus cálculos: $L = 600 \text{ cm}$, $E = 50.000 \text{ kN/cm}^2$, $I = 30.000 \text{ cm}^4$, $w_0 = 2,5 \text{ kN/cm}$, $\Delta x = 10 \text{ cm}$. Empregue a função `subplot` para exibir todos os gráficos verticalmente na mesma janela na ordem (a) até (e). Inclua identificações para os eixos e use unidades consistentes com o MKS quando estiver desenvolvendo os gráficos.

2.22 A *equação da borboleta* é dada pelas seguintes equações paramétricas:

$$x = \sin(t) \left(e^{\cos t} - 2 \cos 4t - \sin^5 \frac{t}{12} \right)$$

$$y = \cos(t) \left(e^{\cos t} - 2 \cos 4t - \sin^5 \frac{t}{12} \right)$$

Gere valores de x e y para valores de t de 0 a 100 com $\Delta t = 1/16$ e construa gráficos de (a) x e y *versus* t e (b) y *versus* x . Use o comando `subplot` para organizar esses gráficos verticalmente e faça o gráfico em (b) quadrado. Inclua títulos e identificações dos eixos em ambos os gráficos e uma legenda para (a). Para (a), empregue uma linha tracejada para y de modo a distingui-la de x .

2.23 A curva da borboleta do Problema 2.22 também pode ser representada em coordenadas polares como

$$r = e^{\sin \theta} - 2 \cos(4\theta) - \sin^5 \left(\frac{2\theta - \pi}{24} \right)$$

Gere valores de r para valores de θ de 0 a 8π com $\Delta\theta = \pi/32$. Use a função do MATLAB `polar` para gerar um gráfico polar da curva da borboleta com uma linha tracejada vermelha. Empregue o Help do MATLAB para entender como gerar o gráfico.

3

Programação com MATLAB

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é aprender como escrever programas no MATLAB (arquivos-M) para implementar métodos numéricos. Após esta leitura completa, você será capaz de:

- ✓ Aprender como criar arquivos-M bem documentados na janela edit window e como chamá-los a partir da command window.
- ✓ Entender as diferenças entre scripts e funções.
- ✓ Saber como inserir comentários de ajuda nas funções.
- ✓ Definir arquivos-M de modo que eles solicitem informações aos usuários de forma interativa e exibam resultados na command window.
- ✓ Entender o papel das subfunções e como elas são acessadas.
- ✓ Aprender como criar arquivos de dados e recuperá-los.
- ✓ Aprender como escrever um arquivo-M claro e bem documentado empregando programação estruturada para implementar lógica e repetição.
- ✓ Reconhecer a diferença entre as construções if...elseif e switch.
- ✓ Reconhecer a diferença entre as construções for...end e while.
- ✓ Aprender como animar gráficos do MATLAB.
- ✓ Compreender o que é vetorização e quais as vantagens dela.
- ✓ Saber como usar funções anônimas para passar funções a arquivos-M do tipo "funções-funções".

VOCÊ TEM UM PROBLEMA

No Capítulo 1, usamos um balanço de forças para desenvolver um modelo matemático a fim de prever a velocidade de queda de um saltador de *bungee jumping*. Esse modelo resultou na seguinte equação diferencial:

$$\frac{dv}{dt} = g - \frac{c_d}{m}v|v|$$

Aprendemos também que essa equação poderia ser resolvida com o método de Euler:

$$v_{i+1} = v_i + \frac{dv_i}{dt} \Delta t$$

Essa equação pode ser usada repetidamente para calcular a velocidade como uma função do tempo; porém, para que o resultado seja o mais acurado possível, vários pequenos passos devem ser tomados, o que pode ser trabalhoso quando feito à mão, mas se torna muito mais fácil quando operado com o MATLAB.

Considerando tal raciocínio, este capítulo apresentará uma introdução de como programas desenvolvidos no MATLAB podem ser usados para obter determinadas soluções.

3.1 ARQUIVOS-M

Uma maneira comum de operar o MATLAB é inserir um comando por vez na command window. Os programas de MATLAB ou arquivos-M (*M-files*), no entanto, fornecem uma maneira alternativa – e mais dinâmica – de efetuar operações: um programa de MATLAB consiste em uma série de instruções ou comandos que podem ser executados todos de uma vez. Observe que a nomenclatura “arquivo-M” ou “M-file” vem do fato de os programas desenvolvidos no MATLAB serem salvos com a extensão `.m`. Tais arquivos podem ser de dois tipos: programas comuns (script files) ou funções (function files).

3.1.1 Programas comuns (script files)

Um *script file* (ou script ou simplesmente “programa” ou “rotina”) trata-se de uma série de comandos do MATLAB que são salvos em um arquivo e podem ser executados em mais de uma ocasião. Para usar o script, digite o nome do arquivo na command window ou selecione a opção **Run** no menu **Debug** da janela edit window.

EXEMPLO 3.1 Script file (ou programa)

Desenvolva um programa para calcular a velocidade do saltador de *bungee jumping* em queda livre para o caso de a velocidade inicial ser zero.

Solução. Abra o editor com a seleção do menu: **File**, **New**, **M-file** e digite as seguintes linhas de comando para calcular a velocidade do saltador de *bungee jumping* em queda livre em um instante de tempo específico [lembre-se da Equação (1.9)]:

```
g = 9.81; m = 68.1; t = 12; cd = 0.25;
v = sqrt(g * m / cd) * tanh(sqrt(g * cd / m) * t)
```

Salve o arquivo como `scriptdemo.m`. Retorne para a command window e digite

```
>> scriptdemo
```

O resultado será exibido como

```
v =
50.6175
```

Assim, o cálculo é exibido como se cada uma das linhas de comando tivesse sido digitada individualmente na command window.

Por fim, determine o valor de `g` digitando:

```
>> g
g =
9.8100
```

Note que, apesar de a variável `g` ter sido definida dentro do programa, ela mantém seu valor na área de trabalho (workspace) do MATLAB. Como veremos na próxima seção, isso é uma importante distinção entre os scripts e as funções.

3.1.2 Funções (function files)

As funções ou *function files* são arquivos-M que começam com a palavra `function`. Ao contrário dos scripts, as funções podem aceitar argumentos de entrada e retornar saídas, portanto, elas são análogas às funções definidas em linguagens de programação tais como Fortran, Visual Basic ou C.

A sintaxe para a função pode ser representada genericamente como

```
function var_saida = nome_funcao(lista_argumentos)
%Comentários_de_ajuda
declarações
var_saida = valor;
```

onde `var_saida` é o nome da variável de saída, `nome_funcao` é o nome da função, `lista_argumentos` é a lista de argumentos (de entrada) da função (isto é, valores separados por vírgulas que são passados para a função), `Comentários_de_ajuda` é um pequeno texto que fornece ao usuário informações sobre a função (esse texto pode ser acessado digitando-se `Help nome_funcao` na command window) e `declarações` são as declarações do MATLAB (linhas de comando) que calculam o `valor` atribuído à variável `var_saida`.

Além de descrever a função, a primeira linha do `Comentários_de_ajuda`, chamada *linha H1*, é a linha recuperada pelo comando `busca lookfor` (lembre-se da Seção 2.6); assim, é importante incluir nesta linha palavras-chave relacionadas à função.

O arquivo-M deve ser salvo como `nome_funcao.m`, para que a função possa ser executada digitando-se `nome_funcao` na command window, como ilustrado no exemplo a seguir. Observe que apesar de o MATLAB ser case sensitive, o sistema operacional do seu computador pode não ser; portanto, considere que seu sistema pode não diferenciar nomes de função como `queda_livre` e `Queda_livre`, distinguidos pelo MATLAB.

EXEMPLO 3.2 Function file ou função

Como no Exemplo 3.1, calcule a velocidade do saltador de *bungee jumping* em queda livre, mas agora use uma função para essa tarefa.

Solução. Digite os seguintes comandos no file editor:

```
function v = queda_livre(t,m,cd)
% queda_livre: velocidade do saltador com arraste de segunda ordem
% v=queda_livre(t,m,cd) calcula a velocidade de queda livre
% de um corpo com arraste de segunda ordem
% entrada:
%   t = tempo (s)
%   m = massa (kg)
```

```
% cd = coeficiente de arraste de segunda ordem (kg/m)
% saída:
% v = velocidade para baixo (m/s)
g = 9.81; % aceleração da gravidade
v = sqrt(g * m/cd) * tanh(sqrt(g * cd / m) * t);
```

Salve o arquivo como queda _ livre.m. Para chamar a função, retorne à command window e digite

```
>> queda _ livre(12,68.1,0.25)
```

O resultado será exibido como

```
ans =
50.6175
```

Uma vantagem da função é que ela pode ser chamada repetidamente para diferentes valores de argumentos. Suponha que se queira calcular a velocidade de um saltador de 100 kg após 8 s:

```
>> queda _ livre(8,100,0.25)
ans =
53.1878
```

Para acessar os comentários de ajuda da função, digite:

```
>> help queda _ livre
```

que resulta nos comentários a seguir:

```
queda _ livre: velocidade do saltador com arraste de segunda ordem
    v=queda _ livre(t,m,cd) calcula a velocidade de queda livre
                    de um corpo com arraste de segunda ordem
entrada:
    t = tempo (s)
    m = massa (kg)
    cd = coeficiente de arraste de segunda ordem (kg/m)
saída:
    v = velocidade para baixo (m/s)
```

Caso você esquecesse o nome dessa função, mas lembresse que ela envolvia um saltador de *bungee jumping*, você poderia digitar

```
>> lookfor saltador
```

e a seguinte informação seria exibida

```
queda _ livre - velocidade do saltador com arraste de segunda ordem
```

Observe que, no fim do exemplo anterior, se tivéssemos digitado

```
>> g
```

a seguinte mensagem teria sido exibida

```
??? Undefined function or variable 'g'.
```

Então, mesmo que g tenha um valor de 9,81 no arquivo-M da função, essa variável não tem um valor na área de trabalho do MATLAB. Como observado no fim do Exemplo 3.1, essa é uma diferença importante entre funções e scripts: enquanto as variáveis de uma função são ditas *locais* e são apagadas depois de executada a função, as variáveis em um script se mantêm mesmo depois de o programa ser executado.

Para casos em que as funções retornam mais que um resultado, as variáveis com os resultados são separadas por vírgulas e delimitadas por colchetes. Por exemplo, a função a seguir, `stats.m`, calcula a média e o desvio-padrão de um vetor:

```
function [media, desvio] = stats(x)
n = length(x);
media = sum(x)/n;
desvio = sqrt(sum( (x-media) .^2/(n-1) ) );
```

Um exemplo de como essa função pode ser aplicada é:

```
>> y = [8 5 10 12 6 7.5 4];
>> [m, d] = stats(y)
m =
7.5000
d =
2.8137
```

Embora também façamos uso de scripts, as funções serão nossa ferramenta de programação principal para o restante deste livro.

3.1.3 Subfunções

Funções podem chamar outras funções, e embora tais funções possam existir como arquivos-M separados, elas também podem estar contidas em um único arquivo-M. Por exemplo, a função no Exemplo 3.2 (sem os comentários) poderia ser dividida em duas funções e salva como um único arquivo-M:¹

```
function v = queda_livre_subfunc(t, m, cd)
v = vel(t, m, cd);
end

function v = vel(t, m, cd)
g = 9.81;
v = sqrt(g * m / cd)*tanh(sqrt(g * cd / m) * t);
end
```

Esse arquivo-M seria salvo como `queda_livre_subfunc.m`. Em tais casos, a primeira função é chamada *função principal* ou *primária*, e é a única função

¹ Observe que, embora as declarações `end` não sejam utilizadas para encerrar arquivos-M de uma única função, elas são incluídas quando subfunções demarcam as fronteiras entre a função principal e as subfunções.

acessível para a command window e para outras funções e scripts. Todas as outras funções (nesse caso, `vel`) são chamadas de *subfunções*.

Uma subfunção é acessível apenas na função principal e em outras subfunções no arquivo-M em que ela reside. Se executarmos a função `queda_livre_subfunc.m` na command window, o resultado será idêntico ao do Exemplo 3.2:

```
>> queda_livre_subfunc(12,68.1,0.25)
ans =
    50.6175
```

No entanto, se tentarmos executar a subfunção `vel`, uma mensagem de erro aparecerá:

```
>> vel(12,68.1,0.25)
??? Undefined function or method 'vel' for input arguments of
type 'double'.
```

3.2 ENTRADA E SAÍDA

Como visto na Seção 3.1, a informação é passada para a função por meio da lista de argumentos de entrada e a informação de saída é obtida chamando a função por seu nome. Duas outras funções nativas do MATLAB fornecem maneiras de inserir e exibir informações diretamente utilizando a command window.

A função `input`. Esta função solicita valores ao usuário diretamente da command window. Sua sintaxe é

```
n = input('mensagem_de_solicitacao')
```

A função exibe a `mensagem_de_solicitacao`, espera pelos dados digitados e, então, retorna o valor do teclado. Por exemplo,

```
m = input('Massa (kg): ')
```

Quando essa linha é executada, a seguinte mensagem aparece:

```
Massa (kg):
```

Se o usuário insere um valor, o número é atribuído à variável `m`.

A função `input` também pode responder a entrada do usuário como uma string. Para fazer isso, um '`s`' é acrescentado à lista de argumentos da função. Por exemplo,

```
nome = input('Entre com seu nome: ', 's')
```

A função `disp`. Esta função, que fornece uma maneira útil de exibir um valor, tem a seguinte sintaxe:

```
disp(valor)
```

onde `valor` é o valor a ser exibido. Ele pode ser uma constante numérica ou uma variável, ou ainda uma mensagem de texto delimitada por aspas simples; sua aplicação é ilustrada no exemplo a seguir.

EXEMPLO 3.3 Uma função interativa de MATLAB

Como no Exemplo 3.2, calcule a velocidade do saltador de *bungee jumping* em queda livre, mas agora use as funções `input` e `disp` para entrada/saída.

Solução. Digite os seguintes comandos no file editor:

```
function queda_livre_i
% queda_livre_i: velocidade do saltador interativa
% queda_livre_i cálculo interativo da velocidade de queda livre
% de um corpo com arraste de segunda ordem
g = 9.81; % aceleração da gravidade
m = input('Massa (kg): ');
cd = input('Coeficiente de arraste (kg/m): ');
t = input('Instante de tempo (s): ');
disp(' ')
disp('Velocidade (m/s):')
disp(sqrt(g * m / cd)*tanh(sqrt(g * cd / m) * t));
```

Salve o arquivo como `queda_livre_i.m`. Para chamar a função, retorne à command window e digite

```
>> queda_livre_i
Massa (kg): 68.1
Coeficiente de arraste (kg/m): 0.25
Instante de tempo (s): 12
Velocidade (m/s):
50.6175
```

A função `fprintf`. Com esta função é possível controlar melhor a exibição de informações. Uma representação simples de sua sintaxe é

```
fprintf('formato', x, ...)
```

onde `formato` é uma string que especifica como você deseja que o valor da variável `x` seja exibido. Vejamos alguns exemplos do uso dessa função.

Um exemplo simples seria exibir um valor com uma mensagem de texto. Por exemplo, considere que a variável `velocidade` tenha um valor de 50,6175. Para exibir o valor de oito dígitos com quatro dígitos à direita da vírgula decimal e, ainda, uma mensagem, a declaração – bem como a saída resultante – seria

```
>> fprintf('A velocidade é %8.4f m/s\n', velocidade)
A velocidade é 50.6175 m/s
```

Esse exemplo ilustra bem o funcionamento de uma string de formato. O MATLAB inicia da extremidade esquerda da string e exibe as palavras até detectar um dos símbolos: `%` ou `\;`; em nosso exemplo, ele encontra primeiro o símbolo `%` e reconhece que o texto a seguir é um código de formato. Como na Tabela 3.1, os *códigos de formato* permitem especificar se os valores numéricos são exibidos no formato inteiro, decimal ou científico. Depois de exibir o valor da variável `velocidade`, o MATLAB mostra as informações de caracteres (em nosso caso, as unidades: `m/s`)

até detectar o símbolo \, que informa ao MATLAB que o texto a seguir é um código de controle. Por meio dos *códigos de controle* (Tabela 3.1) é possível executar determinadas ações, como passar para a próxima linha. Se tivéssemos omitido o código \n no exemplo anterior, o prompt de comando apareceria no fim dos caracteres m/s em vez de na linha seguinte, como desejado.

TABELA 3.1 Alguns códigos de formato e de controle empregados com a função fprintf.

Código de formato	Descrição
%d	Formato inteiro
%e	Formato científico com e minúsculo
%E	Formato científico com E maiúsculo
%f	Formato de ponto fixo
%g	O formato mais compacto de %e ou %f
Código de controle	Descrição
\n	Inicia uma nova linha
\t	Tabulação horizontal

A função fprintf também pode ser usada para exibir diversos valores por linha com diferentes formatos. Por exemplo,

```
>> fprintf('%5d %10.3f %8.5e\n',100,2*pi,pi);
    100      6.283 3.14159e+000
```

Ela também pode ser utilizada para exibir vetores e matrizes. O arquivo-M a seguir define dois conjuntos de valores como vetores, que são combinados em uma matriz, exibida como uma tabela com cabeçalhos:

```
function fprintfdemo
x = [1 2 3 4 5];
y = [20.4 12.6 17.8 88.7 120.4];
z = [x;y];
fprintf('      x      y\n');
fprintf('%5d %10.3f\n',z);
```

O resultado ao executar essa função é:

```
>> fprintfdemo
      x      y
    1    20.400
    2    12.600
    3    17.800
    4    88.700
    5   120.400
```

3.2.1 Criando e acessando arquivos

O MATLAB tem a capacidade de ler e escrever arquivos de dados. A abordagem mais simples envolve um tipo especial de arquivo binário, chamado *MAT-file* ou *arquivo-MAT*, expressamente projetado para implementação no MATLAB. Tais arquivos são criados com o comando save e acessados por meio do comando load.

O comando `save` pode ser usado para gerar um arquivo-MAT que contenha toda a área de trabalho ou algumas variáveis selecionadas; sua sintaxe pode ser representada da seguinte forma:

```
save nome_arquivo var1 var2 ... varn
```

Esse comando cria um arquivo-MAT com o nome `nome_arquivo.mat`, que contém as variáveis `var1` até `varn`. Se as variáveis forem omitidas, todas as variáveis da área de trabalho serão salvas.

O comando `load` pode ser usado para recuperar o arquivo criado:

```
load nome_arquivo var1 var2 ... varn
```

Aqui foram recuperadas as variáveis `var1` até `varn` de `nome_arquivo.mat`. Assim como acontece com o comando `save`, se as variáveis forem omitidas, todas as variáveis do arquivo serão recuperadas.

Por exemplo, considere usar a Equação (1.9) para gerar velocidades para um conjunto de coeficientes de arraste:

```
>> g=9.81;m=80;t=5;
>> cd=[.25 .267 .245 .28 .273]';
>> v=sqrt(g*m ./cd).*tanh(sqrt(g*cd/m)*t);
```

Pode-se então criar um arquivo que contenha os valores dos coeficientes de arraste e as velocidades com

```
>> save velo_arraste v cd
```

Para ilustrar como os valores podem ser recuperados posteriormente, remova todas as variáveis da área de trabalho com o comando `clear`

```
>> clear
```

Neste ponto, se tentássemos exibir as velocidades, obteríamos o seguinte resultado:

```
>> v
??? Undefined function or variable 'v'.
```

No entanto, é possível recuperá-las digitando:

```
>> load velo_arraste
```

Agora as velocidades estão disponíveis, como pode ser verificado ao digitar

```
>> who
```

```
Your variables are:
```

```
cd      v
```

Embora os arquivos-MAT sejam úteis quando se trabalha exclusivamente no ambiente do MATLAB, para realizar uma interface entre o MATLAB e outros programas torna-se necessário criar arquivos de texto escritos no formato ASCII – uma abordagem comum em tais casos.

Arquivos ASCII podem ser gerados no MATLAB acrescentando-se `-ascii` ao comando `save`. Ao contrário dos arquivos-MAT, nos quais você poderia salvar todas as variáveis da área de trabalho, na forma aqui apresentada você salvaria uma única matriz retangular de valores. Por exemplo,

```
>> A=[5 7 9 2;3 6 3 9];
>> save simpmatrix.txt -ascii
```

Nesse caso, o comando `save` armazena os valores em `A` no formato ASCII de oito dígitos. Caso deseje armazenar os números com dupla precisão, crescente – `ascii -double`. Vale ressaltar que o arquivo pode ser acessado por outros programas, tais como processadores de planilhas e textos; por exemplo, se o arquivo salvo for aberto com um editor de textos, será visto desta forma:

```
5.000000e+000 7.000000e+000 9.000000e+000 2.000000e+000  
3.000000e+000 6.000000e+000 3.000000e+000 9.000000e+000
```

É também possível ler os valores de volta para o MATLAB com o comando `load`

```
>> load simpmatrix.txt
```

Como `simpmatrix.txt` não é um arquivo-MAT, o MATLAB cria um arranjo de dupla precisão nomeado conforme o `nome _ arquivo`:

```
>> simpmatrix  
simpmatrix =  
5 7 9 2  
3 6 3 9
```

Pode-se também utilizar o comando `load` como uma função e atribuir seus valores a uma variável, como em

```
>> A = load('simpmatrix.txt')
```

O que foi apresentado até aqui representa apenas uma pequena parte dos recursos de gerenciamento de arquivos do MATLAB. Outro importante assistente pode ser chamado por meio do menu do MATLAB: **File, Import Data**. Teste também o assistente de importação utilizando-o para abrir o arquivo `simpmatrix.txt`; além disso, consulte sempre o `help` para aprender mais sobre estes e outros recursos.

3.3 PROGRAMAÇÃO ESTRUTURADA

Os programas mais simples executam instruções sequencialmente, isto é, linha por linha, iniciando do topo da função e descendo até o fim. Como algumas sequências podem ser um tanto complicadas, todas as linguagens computacionais de alto nível incluem declarações que permitem aos programas tomar caminhos não sequenciais. Essas declarações podem ser classificadas como:

- ✓ *Decisões* (ou Seleção). A ramificação de fluxo com base em uma decisão.
- ✓ *Laços* (ou Repetição). O *looping* (ou enlaçamento) do fluxo para permitir que declarações sejam repetidas.

3.3.1 Decisões

A estrutura if. Esta estrutura permite executar um conjunto de declarações se uma condição lógica for verdadeira. Sua sintaxe genérica é,

```
if condição  
    declarações  
end
```

onde *condição* é uma expressão lógica que pode ser verdadeira ou falsa. Por exemplo, a seguir tem-se uma função simples que determina se uma nota escolar é suficiente ou não para a aprovação:

```
function classifcador(nota)
% classifcador(nota):
%     determina se a nota é suficiente ou não para passar.
% entrada:
%     nota = valor numérica da nota (0-100)
% saída:
% mensagem exibida
if nota >= 60
    disp('Nota suficiente para passar')
end
```

A linha de comando a seguir mostra o resultado:

```
>> classifcador(95.6)
Nota suficiente para passar
```

Para casos em que apenas uma declaração é executada, torna-se conveniente implementar a estrutura, chamada *if de única linha*,

```
if nota >=60, disp('Nota suficiente para passar'), end
```

Para casos em que mais de uma declaração é implementada, a estrutura if em múltiplas linhas é preferível, pois ela é mais fácil de ler.

A função Erro. O if de única linha pode ser muito útil em uma detecção de erros rudimentar, o que envolve a utilização da função error, com a seguinte sintaxe:

```
error(mensagem)
```

Quando essa função é encontrada, ela exibe a mensagem de texto *mensagem*, indica onde o erro ocorreu, encerra o programa e retorna para a command window. Ela poderia ser usada, por exemplo, para encerrar um programa a fim de evitar uma divisão por zero. A seguinte função ilustra como isso pode ser feito:

```
function f = errortest(x)
if x == 0, error('valor igual a zero encontrado'), end
f =1/x;
```

Se um argumento diferente de zero fosse usado, a divisão seria implementada com sucesso, como em

```
>> errortest(10)
ans =
0.1000
```

No entanto, para um argumento igual a zero, a função seria encerrada antes da divisão e a mensagem de texto seria exibida em letras vermelhas:

```
>> errortest(0)
??? Error using ==> errortest at 2
valor igual a zero encontrado
```

TABELA 3.2 Resumo de operadores relacionais no MATLAB.

Exemplo	Operador	Relação
<code>x == 0</code>	<code>==</code>	Igual
<code>unidade ~= 'm'</code>	<code>~=</code>	Diferente
<code>a < 0</code>	<code><</code>	Menor que
<code>s > t</code>	<code>></code>	Maior que
<code>3.9 <= a/3</code>	<code><=</code>	Menor ou igual a
<code>r >= 0</code>	<code>>=</code>	Maior ou igual a

Condições lógicas. A forma mais simples da *condição* é uma expressão relacional única que compara dois valores, como em

$valor_1 \text{ } relação \text{ } valor_2$

onde $valor_1$ e $valor_2$ são constantes, variáveis ou expressões e a *relação* é um dos operadores relacionais listados na Tabela 3.2.

O MATLAB também permite o teste de mais de uma condição lógica com o uso de operadores lógicos, como os seguintes:

- ✓ \sim (*Not* ou *Negação*). Efetua negação lógica em uma expressão.
 $\sim expressão$
Se a *expressão* for verdadeira, o resultado será falso; de forma análoga, se a *expressão* for falsa, o resultado será verdadeiro.
- ✓ $\&$ (*And* ou *E*). Efetua uma combinação lógica de duas expressões.
 $expressão_1 \& expressão_2$
Se ambas as $expressão_1$ e $expressão_2$ avaliadas forem verdadeiras, o resultado será verdadeiro. Se uma das *expressões* for falsa ou se ambas forem falsas, o resultado será falso.
- ✓ \mid (*Or* ou *Ou*). Efetua uma separação lógica de duas expressões.
 $expressão_1 \mid expressão_2$
Se uma das *expressões* for verdadeira ou se ambas forem verdadeiras, o resultado será verdadeiro.

A Tabela 3.3 resume todos os resultados possíveis para cada um desses operadores. Assim como para os operadores aritméticos, há uma ordem de prioridade para avaliação de operações lógicas; da mais alta para a mais baixa essa ordem é: \sim , $\&$ e \mid . Na escolha entre operadores de mesma prioridade, o MATLAB avalia os da esquerda para a direita, e, assim como com operadores aritméticos, parênteses podem ser usados para alterar a ordem de prioridade.

TABELA 3.3 Tabela verdadeiro/falso resumindo os resultados possíveis para operadores lógicos empregados no MATLAB. A ordem de prioridade dos operadores é mostrada no topo da tabela.

		Mais alta	→ Mais baixa	
x	y	$\sim x$	$x \& y$	$x \mid y$
T	T	F	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	F

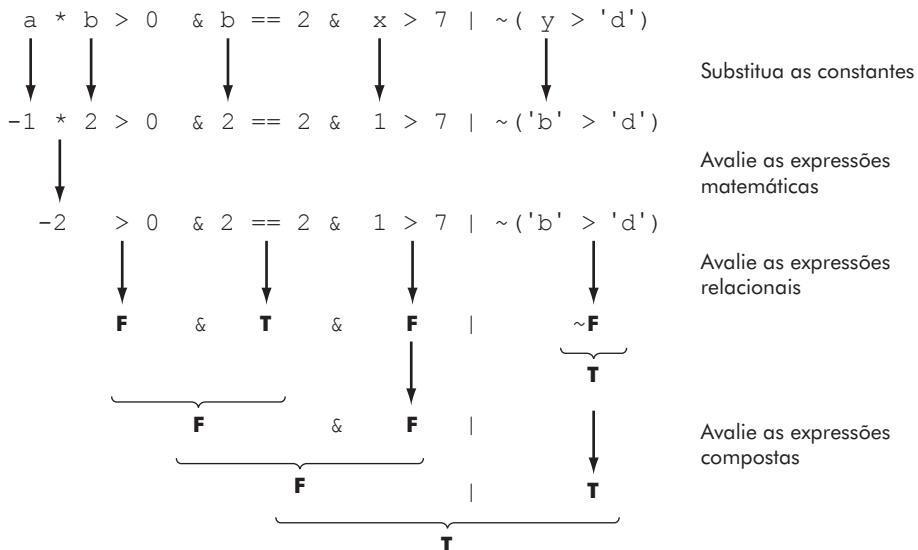


FIGURA 3.1 Avaliação passo a passo de uma decisão complexa.

Vamos analisar como o computador emprega as prioridades para avaliar uma expressão lógica. Se $a = -1$, $b = 2$, $x = 1$ e $y = 'b'$, avalie se a seguinte expressão é verdadeira ou falsa:

$$a * b > 0 \& b == 2 \& x > 7 \mid \sim(y > 'd')$$

Para facilitar, substitua os valores pelas variáveis:

$$-1 * 2 > 0 \& 2 == 2 \& 1 > 7 \mid \sim('b' > 'd')$$

A primeira coisa que o MATLAB faz é avaliar qualquer expressão matemática. Nesse exemplo há apenas uma: $-1 * 2$, portanto

$$-2 > 0 \& 2 == 2 \& 1 > 7 \mid \sim('b' > 'd')$$

Em seguida, avaliam-se as expressões relacionais

$$\begin{array}{ccccccc} -2 & > & 0 & \& 2 & == & 2 \\ F & \& \& \& T & \& \end{array} \& \begin{array}{ccccc} 1 & > & 7 & \mid & \sim \\ F & \& \& \& F \end{array}$$

Neste ponto, os operadores lógicos são avaliados por ordem de prioridade. Como o operador \sim tem prioridade, a última expressão ($\sim F$) é avaliada primeiro para fornecer

$$F \& T \& F \mid T$$

O operador $\&$ é avaliado em seguida. Como há dois, a regra “da esquerda para direita” é aplicada e a primeira expressão ($F \& T$) é avaliada:

$$F \& F \mid T$$

E o operador $\&$ novamente tem prioridade.

$$F \mid T$$

Por fim, o operador \mid é avaliado como verdadeiro. O processo completo é descrito na Figura 3.1.

A estrutura if...else. Esta estrutura permite executar um conjunto de declarações se uma expressão lógica for verdadeira e executar outro conjunto se a condição for falsa. Sua sintaxe genérica é

```
if condição
    declarações1
else
    declarações2
end
```

A estrutura if...elseif. Muitas vezes acontece de a opção falsa de uma estrutura if...else ser outra decisão. Esse tipo de estrutura geralmente ocorre quando temos mais de duas opções para resolver determinado problema. Para tais casos, uma forma especial de estrutura de decisão, a if...elseif, foi desenvolvida, e tem a seguinte sintaxe genérica:

```
if condição1
    declarações1
elseif condição2
    declarações2
elseif condição3
    declarações3
    .
    .
    .
else
    declarações-else
end
```

EXEMPLO 3.4 Estruturas if

Para um escalar, a função nativa sign do MATLAB retorna o sinal de seu argumento ($-1, 0, 1$). Veja a seguir um exemplo de seu funcionamento:

```
>> sign(25.6)
ans =
    1

>> sign(-0.776)
ans =
    -1

>> sign(0)
ans =
    0
```

Desenvolva uma função no MATLAB para efetuar a mesma função.

Solução. Primeiro, uma estrutura if pode ser usada para retornar 1 se o argumento é positivo:

```
function sinal = meu_sinal(x)
% meu_sinal(x) retorna 1 se x é maior que zero
if x > 0
    sinal = 1;
end
```

A função pode ser executada como

```
>> meu_sinal(25.6)
ans =
    1
```

Embora a função manipule números positivos corretamente, se ela for executada com um argumento negativo ou igual a zero, nada é exibido. Para corrigir parcialmente essa deficiência, uma estrutura `if...else` pode ser usada para exibir -1 se a condição for falsa:

```
function sinal = meu_sinal(x)
% meu_sinal(x) retorna 1 se x é maior que zero.
%           -1 se x é menor ou igual a zero.
if x > 0
    sinal = 1;
else
    sinal = -1;
end
```

A função pode ser executada como

```
>> meu_sinal(-0.776)
ans =
    -1
```

Embora agora os casos de números positivos e negativos sejam adequadamente tratados, -1 é retornado de forma incorreta se um argumento igual a zero for utilizado. Uma estrutura `if...elseif` pode ser usada para incorporar esse caso final:

```
function sinal = meu_sinal(x)
% meu_sinal(x) retorna 1 se x é maior que zero.
%           -1 se x é menor ou igual a zero.
%           0 se x é igual a zero.
if x > 0
    sinal = 1;
elseif x < 0
    sinal = -1;
else
    sinal = 0;
end
```

A função agora manipula todos os casos possíveis. Por exemplo,

```
>> meu_sinal(0)
ans =
    0
```

A estrutura switch. A estrutura `switch` é muito similar à estrutura `if...else`; no entanto, em vez de testar condições individuais, a ramificação se baseia no valor de uma única expressão de teste. Dependendo de seu valor, diferentes blocos de código podem ser implementados; além disso, um bloco opcional será implementado se a expressão não assumir nenhum dos valores prescritos. A estrutura `switch` tem a seguinte sintaxe genérica:

```
switch expressão _ teste
    case valor1
        declarações1
    case valor2
        declarações2
    .
    .
    .
    otherwise
        declarações _ otherwise
end
```

Como um exemplo, o código a seguir exibe uma mensagem que depende do valor da variável `conceito` (uma string).

```
conceito = 'B';
switch conceito
    case 'A'
        disp('Excelente')
    case 'B'
        disp('Bom')
    case 'C'
        disp('Regular')
    case 'D'
        disp('Ruim')
    case 'E'
        disp('Reprovado')
    otherwise
        disp('Péssimo!')
end
```

Quando esse código for executado, a mensagem “Bom” será exibida.

Lista de argumentos variável. O MATLAB permite que um número variável de argumentos seja passado para uma função, o que pode ser útil para a incorporação de valores-padrão em suas funções. Um *valor-padrão* ou *default* é um número automaticamente atribuído quando o usuário não o passa para a função.

Para ilustrar, lembre que no Exemplo 3.2 foi desenvolvida a função `queda_livre`, que tinha três argumentos:

```
v = queda _ livre(t, m, cd)
```

Embora um usuário obviamente tenha de especificar a massa e o instante e tempo, ele poderia não ter ideia de um coeficiente de arraste adequado. Seria interessante, portanto, que o programa fornecesse um valor, caso ele fosse omitido da lista de argumentos.

O MATLAB tem uma função chamada `nargin` que fornece o número de argumentos inseridos na função por um usuário. Ela pode ser usada com estruturas de decisão – como as construções `if` e `switch` – para incorporar valores-padrão, assim como mensagens de erro nas funções. O código a seguir ilustra como isso pode feito para a função `queda _ livre`:

```
function v = queda _ livre(t, m, cd)
% queda _ livre2: velocidade do saltador com arraste de se-
% gunda ordem
% v=queda _ livre(t,m,cd) calcula a velocidade de queda livre
%                 de um corpo com arraste de segunda ordem
% entrada:
%   t = tempo (s)
%   m = massa (kg)
%   cd = coeficiente de arraste de segunda ordem (kg/m)
% saída:
%   v = velocidade para baixo (m/s)
switch nargin
    case 0
        error('Deve-se entrar com o instante de tempo e a massa')
    case 1
        error('Deve-se entrar com a massa')
    case 2
        cd = 0.27;
end
g = 9.81;      % aceleração da gravidade
v = sqrt(g * m / cd)*tanh(sqrt(g * cd / m) * t);
```

Observe que foi utilizada uma estrutura `switch` para exibir mensagens de erro ou para definir o valor-padrão, dependendo do número de argumentos fornecidos pelo usuário. Confira os resultados nesta sessão do MATLAB:

```
>> queda _ livre2(12, 68.1, 0.25)
ans =
    50.6175
>> queda _ livre2(12, 68.1)
ans =
    48.8747
>> queda _ livre2(12)
??? Error using ==> queda _ livre2 at 15
Deve-se entrar com a massa
>> queda _ livre2()
??? Error using ==> queda _ livre2 at 13
Deve-se entrar com o instante de tempo e a massa
```

Observe que a função `nargin` se comporta de modo um pouco diferente quando ela é chamada na command window, onde deve-se incluir uma string como argumento especificando o nome da função para que ela retorne o número de argumentos na função. Por exemplo,

```
>> nargin('queda _ livre2')
ans =
    3
```

3.3.2 Laços

Como o próprio nome indica, os laços realizam operações repetidamente. Há dois tipos de laços, dependendo de como as repetições são encerradas: um *laço for* termina depois de um número específico de repetições, enquanto um *laço while* termina com base em uma condição lógica.

A estrutura for...end. Um laço *for* repete declarações um número específico de vezes, e sua sintaxe genérica é

```
for índice = início:passo:fim
    declarações
end
```

O laço *for* funciona da seguinte forma: o *índice* é uma variável definida com um valor inicial, *início*; o programa, então, compara a variável *índice* com o valor final desejado, *fim*. Se o *índice* for menor ou igual à variável *fim*, o programa executará as *declarações*. Quando ele chega à linha *end*, que caracteriza o fim do laço, a variável *índice* é incrementada pelo valor da variável *passo* e o programa retorna à declaração *for*. O processo continua até o *índice* tornar-se maior que o valor de *fim*; neste ponto, o laço é encerrado e o programa salta para a linha imediatamente após a declaração *end*.

Caso você deseje um incremento de 1 (como muitas vezes é o caso), o *passo* pode ser omitido. Por exemplo,

```
for i = 1:
    disp(i)
end
```

Quando esse código é executado, o MATLAB exibe, em sequência, 1, 2, 3, 4, 5, o que caracteriza o *passo* padrão (default) como 1.

O tamanho do *passo* pode ser alterado do padrão de 1 para qualquer outro valor numérico, que não precisa ser um inteiro, nem tem que ser positivo. Por exemplo, tamanhos de passo de 0.2, -1 ou -5 são todos aceitáveis.

Se um *passo* negativo for usado, o laço funcionará em uma “contagem regressiva”, no sentido inverso, o que inverte a lógica do laço. Assim, o valor de *fim* é menor que o de *início* e o laço é encerrado quando o *índice* é menor que o *fim*. Por exemplo,

```
for j = 10:-1:1
    disp(j)
end
```

Quando esse código é executado, o MATLAB exibe a clássica sequência de contagem regressiva: 10, 9, 8, 7, 6, 5, 4, 3, 2, 1.

EXEMPLO 3.5 Usando um laço for para calcular o fatorial

Desenvolva uma função no MATLAB para calcular o fatorial.²

$$0! = 1$$

$$1! = 1$$

$$2! = 1 \times 2 = 2$$

² Observe que o MATLAB tem uma função nativa `factorial` que efetua esse cálculo.

$$\begin{aligned}3! &= 1 \times 2 \times 3 = 6 \\4! &= 1 \times 2 \times 3 \times 4 = 24 \\5! &= 1 \times 2 \times 3 \times 4 \times 5 = 120\end{aligned}$$

.

.

Solução. Uma função simples que implementa esse cálculo pode ser desenvolvida como em

```
function fsaida = fatorial(n)
% fatorial(n):
%   Calcula o produto de todos os inteiros de 1 a n.
x = 1;
for i = 1:n
    x = x * i;
end
fsaida = x;
end
```

que pode ser executado como

```
>> fatorial(5)
ans =
    120
```

O laço executará 5 vezes (de 1 a 5). No fim do processo, a variável `x` terá o valor de 5! (ou seja, o fatorial de 5 ou $1 \times 2 \times 3 \times 4 \times 5 = 120$).

Observe o que acontece se $n = 0$; nesse caso, o laço `for` não executaria e obteríamos o resultado desejado, $0! = 1$.

Vetorização. Embora seja de fácil implementação e compreensão para o MATLAB, talvez o laço `for` não seja a maneira mais eficiente de repetir declarações um número específico de vezes. Por conta da capacidade do MATLAB de operar diretamente com arranjos, a *vetorização* fornece, em muitos casos, uma opção muito mais eficiente. Por exemplo, a seguinte estrutura de laço `for`

```
i = 0;
for t = 0:0.02:50
    i = i + 1;
    y(i) = cos(t);
end
```

pode ser representada na forma vetorializada como

```
t = 0:0.02:50;
y = cos(t);
```

É importante observar que, para códigos mais complexos, a forma como vetorizar o código pode não ser tão óbvia. Dito isso, a vetorização é recomendada sempre que possível.

Pré-alocação de memória. O MATLAB aumenta automaticamente o tamanho de arranjos toda vez que um novo elemento é adicionado, o que pode tornar o

processo demorado quando são realizadas ações tais como a adição de novos valores individualmente dentro de um laço. O código a seguir, por exemplo, define valores de elementos de y dependendo se os valores de t são ou não maiores que 1:

```
t = 0:.01:5;
for i = 1:length(t)
    if t(i)>1
        y(i) = 1/t(i);
    else
        y(i) = 1;
    end
end
```

Para esse caso, o MATLAB deve redimensionar y toda vez que um novo valor é determinado. O código a seguir pré-aloca a quantidade adequada de memória utilizando uma declaração vetorizada para atribuir 1's a y antes da entrada no laço.

```
t = 0:.01:5;
y = ones(size(t));
for i = 1:length(t)
    if t(i)>1
        y(i) = 1/t(i);
    end
end
```

Assim, além de o arranjo ser dimensionado apenas uma vez, a pré-alocação ajuda a reduzir a fragmentação de memória – o que também aumenta a eficiência.

A estrutura while. Um laço `while` repete enquanto uma condição lógica for verdadeira, e apresenta a seguinte sintaxe genérica:

```
while condição
    declarações
end
```

As *declarações* entre o `while` e o `end` serão repetidas enquanto a *condição* for verdadeira. Um exemplo simples é

```
x = 8
while x > 0
    x = x - 3;
    disp(x)
end
```

Quando esse código é executado, o resultado é

```
x =
8
5
2
-1
```

A estrutura while...break. Embora a estrutura `while` seja extremamente útil, o fato de que ela sempre se repete a menos que a condição testada no início seja falsa representa uma limitação. Por essa razão, algumas linguagens tais como Fortran 90

e Visual Basic têm estruturas especiais que permitem encerrar o laço sob uma condição verdadeira em qualquer lugar dentro do laço. Apesar de tais estruturas não estarem mais disponíveis no MATLAB, suas funcionalidades podem ser “imitadas” por uma versão especial do laço `while`. A sintaxe dessa versão, chamada *estrutura while...break*, pode ser escrita como

```
while (1)
    declarações
    if condição, break, end
    declarações
end
```

onde o comando `break` encerra a execução do laço. Assim, uma única linha `if` é usada para sair do laço se a condição de teste for verdadeira. Observe que, como mostrado, o `break` pode ser colocado no meio do laço (isto é, com `declarações` antes e depois dele), o que caracteriza o *laço com teste no meio*.

Se o problema exigir, pode-se colocar o `break` logo no início para criar um *laço com pré-teste*, como no exemplo:

```
while (1)
    if x < 0, break, end
    x = x - 5;
end
```

Observe como 5 é subtraído de `x` em cada iteração. Isso representa um mecanismo que finaliza o laço em algum momento. Todo laço de decisão deve ter tal mecanismo, caso contrário, ele se tornaria um *laço infinito*.

Como alternativa, pode-se colocar a declaração `if...break` no fim e criar um *laço com pós-teste*.

```
while (1)
    x = x - 5;
    if x < 0, break, end
end
```

Deve ficar claro que, na realidade, as três estruturas são a mesma coisa, isto é, dependendo do local onde a saída é colocada (início, meio ou fim), temos um laço com pré-teste, teste no meio ou pós-teste. Foi essa simplicidade que levou os cientistas da computação que desenvolveram o Fortran 90 e o Visual Basic a favorecer essa estrutura sobre outras formas de laços de decisão, tais como a estrutura `while` convencional.

O comando pause. Há vezes em que se deseja interromper um programa temporariamente. O comando `pause` gera um recurso para parar e esperar até que alguma tecla seja pressionada, como no caso de um usuário que queira examinar com calma uma sequência de gráficos. O código a seguir emprega um laço `for` para criar uma sequência de gráficos que podem ser vistos desta maneira:

```
for n = 3:10
    mesh(magic(n))
    pause
end
```

O comando `pause` também pode ser formulado como `pause(n)`, caso em que o procedimento será interrompido por `n` segundos. Esse recurso pode ser implementado com outras funções do MATLAB, como o comando `beep`, que faz o computador emitir um sinal sonoro, ou as funções `tic` e `toc`, que trabalham juntas para medir o tempo decorrido (o comando `tic` salva o tempo atual que o `toc` emprega posteriormente para exibir o tempo decorrido). O seguinte código confirma, então, que o comando `pause(n)` funciona como descrito, complementado com os efeitos sonoros:

```
tic  
beep  
pause(5)  
beep  
toc
```

Quando esse código é executado, o computador emite um sinal sonoro; cinco segundos depois, ele emite outro sinal sonoro e exibe a seguinte mensagem:

```
Elapsed time is 5.004661 seconds.
```

Caso o usuário tenha a necessidade de usar o comando `pause(inf)`, o MATLAB entrará em um laço infinito. Em tais casos, pode-se retornar ao prompt de comando digitando-se **Ctrl+c** ou **Ctrl+Break**.

Embora os exemplos anteriores possam parecer pouco significantes, os comandos são bastante úteis: por exemplo, `tic` e `toc` podem ser empregados para identificar as partes de um algoritmo que consomem a maior parte do tempo de execução. Além disso, as combinações de teclas **Ctrl+c** ou **Ctrl+Break** podem ser valiosas nos casos em que, acidentalmente, foi criado um laço infinito em algum programa.

3.3.3 Animação

Há duas maneiras simples de animar um gráfico no MATLAB. Primeiro, se os cálculos forem suficientemente rápidos, a função `plot` padrão pode ser usada para que a animação pareça mais suave. O fragmento de código a seguir indica como um laço `for` e funções gráficas padrão podem ser empregadas para animar um gráfico:

```
% cria uma animação com funções gráficas padrão  
for j=1:n  
    plot comandos  
end
```

Assim, como não foi incluído o comando `hold on`, o gráfico será atualizado em cada iteração do laço, o que pode resultar em uma mudança suave da imagem caso os comandos de formatação dos eixos sejam usados de forma criteriosa.

Segundo, há funções especiais, `getframe` e `movie`, que permitem capturar uma sequência de gráficos e, depois, reproduzi-los. Como o nome indica, a função `getframe` capture um instantâneo (*pixmap*) dos eixos ou figuras correntes, e é utilizada em um laço `for` para reunir quadros de filme reproduzidos posteriormente com a função `movie`, que tem a seguinte sintaxe:

```
movie(m, n, fps)
```

onde m é o vetor ou matriz que contém a sequência de quadros constituintes do filme, n é uma variável opcional que especifica quantas vezes o filme deve ser repetido (se ela for omitida, o filme será reproduzido uma vez) e fps é uma variável opcional que especifica a taxa de quadros do filme (se ela for omitida, o padrão será 12 quadros por segundo). O fragmento de código a seguir indica como um laço `for`, com as funções `getframe` e `movie`, pode ser empregado para criar um filme:

```
% cria uma animação com getframe e movie
for j=1:n
    plot comandos
    M(j) = getframe;
end
movie(M)
```

Cada vez que o laço é executado, a linha `plot comandos` cria uma versão atualizada de um gráfico, que é, então, armazenada no vetor M . Depois que o laço é encerrado, as n imagens são reproduzidas pela função `movie`.

EXEMPLO 3.6 Animação do movimento de um projétil

Considerando-se a falta de resistência do ar, as coordenadas cartesianas de um projétil lançado com uma velocidade inicial (v_0) e ângulo (θ_0) podem ser calculadas com

$$\begin{aligned}x &= v_0 \cos(\theta_0)t \\y &= v_0 \sin(\theta_0)t - 0,5gt^2\end{aligned}$$

onde $g = 9,81 \text{ m/s}^2$. Desenvolva um script para gerar um gráfico animado da trajetória do projétil dado que $v_0 = 5 \text{ m/s}$ e $\theta_0 = 45^\circ$.

Solução. Um script para gerar a animação pode ser escrito desta forma:

```
clc,clf,clear
g=9.81; theta0=45*pi/180; v0=5;
t(1)=0;x=0;y=0;
plot(x, y, 'o', 'MarkerFaceColor', 'b', 'MarkerSize', 8)
axis([0 3 0 0.8])
M(1)=getframe;
dt=1/128;
for j = 2:1000
    t(j)=t(j-1)+dt;
    x=v0*cos(theta0)*t(j);
    y=v0*sin(theta0)*t(j)-0.5*g*t(j)^2;
    plot(x, y, 'o', 'MarkerFaceColor', 'b', 'MarkerSize', 8)
    axis([0 3 0 0.8])
    M(j)=getframe;
    if y<=0, break, end
end
pause
movie(M,1)
```

Várias características desse script merecem ser mencionadas. Primeiro, observe que os intervalos dos eixos x e y foram fixados: se isso não for feito, as escalas dos eixos serão

redimensionadas em cada iteração do `for`, o que faz a animação ficar “pulando” (retire os comandos `axis` e veja o que acontece!). Segundo, o laço `for` é encerrado quando a altura do projétil y cai abaixo de zero.

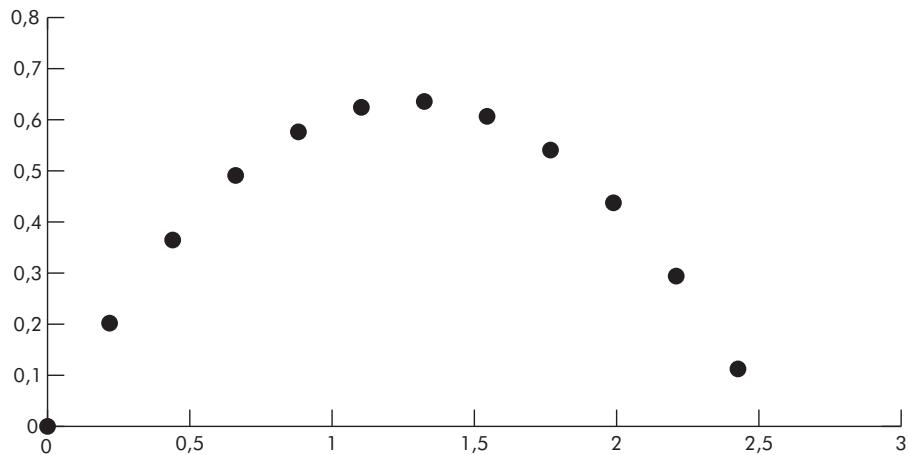


FIGURA 3.2 Gráfico da trajetória de um projétil.

Quando esse script é executado, duas animações são exibidas (foi colocado um `pause` entre elas): a primeira corresponde à geração sequencial dos quadros no laço e a segunda corresponde ao filme propriamente dito. Embora os resultados não possam ser mostrados neste livro, a trajetória para ambos os casos será semelhante à apresentada na Figura 3.2. Para ver a animação real, execute o código anterior no MATLAB.

3.4 ANINHAMENTO E INDENTAÇÃO

As estruturas podem ser “aninhadas” – ou colocadas – umas dentro das outras. Veja o exemplo a seguir, que ilustra o conceito.

EXEMPLO 3.7 Estruturas aninhadas

As raízes de uma equação quadrática

$$f(x) = ax^2 + bx + c$$

podem ser determinadas com a fórmula quadrática

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

Desenvolva uma função para implementar essa fórmula dados os valores dos coeficientes.

Solução. O projeto de cima para baixo (*top-down design*) fornece uma boa abordagem para projetar um algoritmo que calcule raízes; o processo envolve o desenvolvimento da

estrutura geral sem detalhes e, em seguida, o refinamento do algoritmo. Para iniciar, é necessário reconhecer que, dependendo se o parâmetro a for zero ou não, têm-se casos especiais (por exemplo, raízes únicas ou valores triviais) ou casos convencionais, que utilizam a fórmula quadrática. Essa primeira versão, um esboço geral, pode ser programada como

```
function raiz_quadrada(a, b, c)
% raiz_quadrada: raízes de uma equação quadrática
%   raiz_quadrada(a, b, c): raízes reais e complexas
%
%                           de uma equação quadrática
%
% entrada:
%
%   a = coeficiente de segunda ordem
%   b = coeficiente de primeira ordem
%   c = coeficiente de ordem zero
%
% saída:
%
%   r1 = parte real da primeira raiz
%   i1 = parte imaginária da primeira raiz
%   r2 = parte real da segunda raiz
%   i2 = parte imaginária da segunda raiz
if a == 0
    %casos especiais
else
    %fórmula quadrática
end
```

Em seguida, desenvolve-se o código refinado para tratar os casos especiais:

```
%casos especiais
if b ~= 0
    %raiz real única
    r1 = -c / b
else
    %solução trivial
    disp('Solução trivial. Tente novamente.')
end
```

Após isso, desenvolve-se o código refinado para tratar os casos convencionais que utilizam a fórmula quadrática:

```
%fórmula quadrática
d = b ^ 2 - 4 * a * c;
if d >= 0
    %raízes reais
    r1 = (-b + sqrt(d)) / (2 * a)
    r2 = (-b - sqrt(d)) / (2 * a)
else
    %raízes complexas
    r1 = -b / (2 * a)
    i1 = sqrt(abs(d)) / (2 * a)
    r2 = r1
    i2 = -i1
end
```

Por fim, pode-se simplesmente retornar esses blocos ao esboço geral da função para obter o resultado final:

```

function raiz_quadrada (a, b, c)
% raiz_quadrada: raízes de uma equação quadrática
%   raiz_quadrada (a, b, c): raízes reais e complexas
%                               de uma equação quadrática
%
% entrada:
%   a = coeficiente de segunda ordem
%   b = coeficiente de primeira ordem
%   c = coeficiente de ordem zero
%
% saída:
%   r1 = parte real da primeira raiz
%   i1 = parte imaginária da primeira raiz
%   r2 = parte real da segunda raiz
%   i2 = parte imaginária da segunda raiz
if a == 0
    %casos especiais
    if b ~= 0
        %raiz real única
        r1 = -c / b
    else
        %solução trivial
        disp('Solução trivial. Tente novamente.')
    end
else
    %fórmula quadrática
    d = b ^ 2 - 4 * a * c;      % discriminant
    if d >= 0
        %raízes reais
        r1 = (-b + sqrt(d)) / (2 * a)
        r2 = (-b - sqrt(d)) / (2 * a)
    else
        %raízes complexas
        r1 = -b / (2 * a)
        i1 = sqrt(abs(d)) / (2 * a)
        r2 = r1
        i2 = -i1
    end
end

```

Observe, na área destacada pelo sombreamento, como a indentação torna clara a estrutura lógica básica; note também como as estruturas são modulares. Veja a seguir uma sessão do MATLAB ilustrando como a função opera:

```

>> raiz_quadrada(1, 1, 1)
r1 =
-0.5000
i1 =
0.8660
r2 =
-0.5000
i2 =
-0.8660

```

```
>> raiz_quadrada(1, 5, 1)
r1 =
    -0.2087
r2 =
    -4.7913
>> raiz_quadrada(0, 5, 1)
r1 =
    -0.2000
>> raiz_quadrada(0, 0, 0)
Solução trivial. Tente novamente.
```

3.5 PASSANDO FUNÇÕES PARA ARQUIVOS-M

Grande parte deste livro aborda o desenvolvimento de funções para avaliar numericamente outras funções. Embora uma função personalizada possa ser desenvolvida para cada nova equação analisada, talvez seja melhor projetar uma função genérica e passar a equação que se deseja analisar como um argumento – no jargão do MATLAB, essas funções recebem um nome especial: *funções-funções* (*function functions*). Antes de descrever como elas funcionam, apresentaremos as funções anônimas, um meio prático para definir funções personalizadas pelo usuário sem a necessidade de desenvolver um arquivo-M completo.

3.5.1 Funções anônimas

As *funções anônimas* permitem criar funções simples sem desenvolver um arquivo-M e podem ser definidas na command window com a seguinte sintaxe:

```
fhandle = @(lista_de_argumentos) expressão
```

onde *fhandle* é o *function handle* ou identificador da função, que pode ser usado para chamar a função, *lista_de_argumentos* é uma lista de argumentos de entrada separados por vírgula a serem passados para a função e *expressão* é qualquer expressão válida do MATLAB. Por exemplo,

```
>> f1=@(x,y) x^2 + y^2;
```

Uma vez definidas na command window, tais funções podem ser usadas como qualquer outra função:

```
>> f1(3,4)
ans =
    25
```

Além dos argumentos em sua lista, uma função anônima pode incluir variáveis que existem na área de trabalho (workspace) onde ela foi criada. Por exemplo, pode-se criar uma função anônima $f(x) = 4x^2$ como

```
>> a = 4;
>> b = 2;
>> f2=@(x) a*x^b;
>> f2(3)
ans = 36
```

Observe que se, posteriormente, forem definidos novos valores para `a` e `b`, a função anônima não será alterada:

```
>> a = 3;
>> f2(3)
ans = 36
```

Assim, o identificador da função contém um “instantâneo” da função no momento em que ela foi criada. Caso se queira que as variáveis assumam novos valores, é necessário recriar a função – por exemplo, tendo alterado `a` para 3,

```
>> f2=@(x) a*x^b;
```

com o resultado

```
>> f2(3)
ans =
27
```

Deve-se notar que antes do MATLAB 7, as funções `inline` desempenhavam um papel similar ao das funções anônimas. Por exemplo, a função anterior, `f1`, poderia ser escrita como

```
>> f1=inline('x^2 + y^2', 'x', 'y');
```

Embora as funções `inline` estejam caindo em desuso em favor das funções anônimas, achamos que seria útil mencioná-las, pois alguns leitores podem ainda utilizar versões mais antigas. Para aprender mais sobre o uso e as limitações das funções `inline`, consulte a Ajuda do MATLAB.

3.5.2 Funções-funções

As *funções-funções* (*function functions*) são funções que operam em outras funções as quais são passadas para elas como argumentos de entrada. A função passada para a função-função é chamada de *função passada ou função importada*, como a função nativa `fplot`, que traça o gráfico de funções. Uma representação simples de sua sintaxe é

```
fplot(func,lims)
```

onde `func` é a função sendo traçada entre os limites do eixo `x` especificados por `lims = [xmin xmax]` – para esse caso, `func` é a função importada. Esta função é “inteligente”, pois analisa automaticamente a função e decide quantos valores usar, de modo que o gráfico exiba todas as características importantes da função.

A linha de comando a seguir é um exemplo de como a função `fplot` pode ser usada para traçar um gráfico da velocidade do saltador de *bungee jumping* em queda livre. A função pode ser criada com uma função anônima:

```
>> vel=@(t)...
sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
```

Pode-se, em seguida, gerar um gráfico de $t = 0$ a 12 s como

```
>> fplot(vel,[0 12])
```

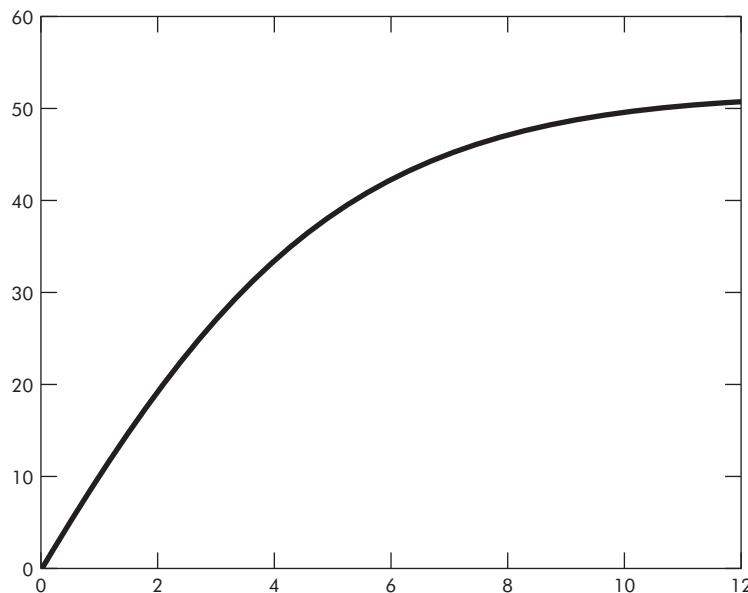


FIGURA 3.3 Gráfico da velocidade versus o tempo com a função `fplot`.

O resultado é exibido na Figura 3.3.

Observe que no restante deste livro, teremos muitas ocasiões para usar funções-funções nativas do MATLAB e desenvolver nossas próprias funções-funções, como no próximo exemplo.

EXEMPLO 3.8 Construindo e implementando uma função-função

Desenvolva uma função-função para determinar o valor médio de uma função ao longo de um intervalo. Ilustre seu uso para a velocidade do saltador de *bungee jumping* ao longo do intervalo de $t = 0$ a 12 s,

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right)$$

onde $g = 9,81$ m/s², $m = 68,1$ kg e $c_d = 0,25$ kg/m.

Solução. O valor médio da função pode ser calculado com comandos-padrão do MATLAB, como

```
>> t=linspace(0,12);
>> v=sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
>> mean(v)
ans =
36.0870
```

A inspeção de um gráfico da função (Figura 3.3) mostra que esse resultado é uma estimativa plausível da altura média da curva.

Podemos escrever um arquivo-M para efetuar os mesmos cálculos:

```
function f_media = func_media(a,b,n)
% func_media: altura média da função
% f_media=func_media(a, b, n): calcula o valor médio de uma função
%                               ao longo de um intervalo
% entrada:
% a = limite inferior do intervalo
% b = limite superior do intervalo
% n = número de subintervalos
% saída:
% f_media = valor médio da função
x = linspace(a,b,n);
y = func(x);
f_media = mean(y);
end

function f = func(t)
f=sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
end
```

Primeiro a função utiliza o comando `linspace` para gerar valores de `x` igualmente espaçados ao longo do intervalo; esses valores são, então, passados para uma subfunção `func` para gerar os valores de `y` correspondentes, e em seguida o valor médio é calculado. A função pode ser executada da command window como

```
>> func_media(0,12,60)
ans =
    36.0127
```

Vamos agora reescrever o arquivo-M de modo que, em vez de ser específico para `func`, ele avalie uma função qualquer chamada `f`, passada como um argumento de entrada:

```
function f_media = func_media(f,a,b,n)
% func_media: altura média da função
% f_media=func_media(f, a, b, n): calcula o valor médio de uma função
%                               ao longo de um intervalo
% entrada:
% f = função a ser avaliada
% a = limite inferior do intervalo
% b = limite superior do intervalo
% n = número de subintervalos
% saída:
% f_media = valor médio da função
x = linspace(a,b,n);
y = f(x);
f_media = mean(y);
end
```

Como removemos a subfunção `func`, essa versão é genérica e pode ser executada da command window como

```
>> vel=@(t)...
sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
>> func_media(vel,0,12,60)
```

```
ans =
36.0127
```

Para demonstrar sua natureza genérica, a função-função `func_media` pode ser aplicada a outro caso com a inserção de uma função diferente. Por exemplo, ela poderia ser usada para determinar o valor médio da função nativa `sin` entre 0 e 2π como

```
>> func_media(@sin,0,2*pi,180)
ans =
-6.3001e-017
```

Esse resultado faz sentido?

Pode-se perceber que a função-função `func_media` está agora projetada para avaliar qualquer expressão válida do MATLAB – você poderá vê-la em ação neste livro em vários contextos: desde a solução de equações não lineares até a solução de equações diferenciais.

3.5.3 Passando parâmetros

Vimos no Capítulo 1 que os termos em um modelo matemático podem ser divididos em variáveis dependentes e independentes, parâmetros e funções forçantes. Para o modelo do saltador de *bungee jumping*, a velocidade (v) é a variável dependente, o tempo (t) é a variável independente, a massa (m) e o coeficiente de arraste (c_d) são parâmetros e a constante da gravidade (g) é a função forçante. É comum investigar o comportamento de tais modelos realizando uma *análise de sensibilidade*, o que envolve a observação de como as variáveis dependentes mudam quando os parâmetros e as funções forçantes são variados.

No Exemplo 3.8, desenvolvemos uma função-função, `func_media`, que foi empregada para determinar o valor médio da velocidade do saltador de *bungee jumping* para o caso em que os parâmetros foram definidos como $m = 68,1 \text{ kg}$ e $c_d = 0,25 \text{ kg/m}$. Talvez alguém quisesse analisar a mesma função com parâmetros diferentes; poderíamos redigitar a função com novos parâmetros em cada caso, porém seria melhor apenas alterar os parâmetros.

Como descrito na Seção 3.5.1, é possível incorporar parâmetros em funções anônimas; em vez de digitar os valores numéricos, poderíamos fazer o seguinte:

```
>> m=68.1;c d=0.25;
>> vel=@(t) sqrt(9.81*m/cd)*tanh(sqrt(9.81*cd/m)*t);
>> func_media(vel,0,12,60)
ans =
36.0127
```

No entanto, se quisermos que os parâmetros assumam novos valores, deve-se recriar a função anônima.

O MATLAB oferece uma alternativa mais interessante, com a adição do termo `varargin` como último argumento de entrada da função-função. Além disso, cada vez que a função importada é chamada na função-função, o termo `varargin{:}` deve ser adicionado ao fim de sua lista de argumentos (observe as chaves). As linhas a seguir mostram como ambas as modificações podem ser implementadas para a função-função `func_media` (omitindo os comentários para fins de concisão):

```
function f_media = func_media(f,a,b,n,varargin)
x = linspace(a,b,n);
y = f(x,varargin{:});
f_media=mean(y);
```

Quando a função importada é definida, os parâmetros da equação devem ser adicionados ao fim da lista de argumentos, como no exemplo seguinte, que utiliza uma função anônima.

```
>> vel=@(t,m,cd) sqrt(9.81*m/cd)*tanh(sqrt(9.81*cd/m)*t);
```

Quando todas essas mudanças são feitas, a análise de diferentes parâmetros torna-se fácil. Para implementar o caso em que $m = 68,1 \text{ kg}$ e $c_d = 0,25 \text{ kg/m}$, pode-se entrar com o seguinte comando

```
>> func_media(vel,0,12,60,68.1,0.25)
ans =
36.0127
```

Um caso alternativo, por exemplo, $m = 100 \text{ kg}$ e $c_d = 0,28 \text{ kg/m}$, pode ser rapidamente gerado apenas alterando-se os argumentos:

```
>> func_media(vel,0,12,60,100,0.28)
ans =
38.9345
```

► 3.6 ESTUDO DE CASO VELOCIDADE DO SALTADOR DE BUNGEE JUMPING

Contextualização. Nesta seção, usaremos o MATLAB para resolver o problema do saltador de *bungee jumping* em queda livre colocado no início deste capítulo, o que envolve a resolução de

$$\frac{dv}{dt} = g - \frac{c_d}{m}v|v|$$

Lembre que, dada uma condição inicial para tempo e velocidade, o problema envolveu iterativamente a resolução da fórmula

$$v_{i+1} = v_i + \frac{dv_i}{dt} \Delta t$$

Lembre-se, também, que para obter o máximo de acurácia, devemos dar passos pequenos; portanto, aplicamos a fórmula repetidamente, saindo do instante de tempo inicial para atingir o valor no instante de tempo final, utilizando um algoritmo com base em um laço.

Solução. Considere que os cálculos sejam iniciados em $t = 0$ e que se deseje prever a velocidade em $t = 12 \text{ s}$ utilizando um passo de tempo de $\Delta t = 0,5 \text{ s}$. Nesse caso, a equação iterativa deveria ser aplicada 24 vezes, isto é,

$$n = \frac{12}{0,5} = 24$$

onde n é o número de iterações do laço. Como esse número é exato (isto é, a razão é um inteiro), pode-se utilizar um laço `for` como base para o algoritmo. A seguir, use um arquivo-M para fazer isso, incluindo uma subfunção que defina a equação diferencial:

```
function v_fm = velocidade1(dt, ti, tf, vi)
% velocidade1: Solução de Euler para velocidade do saltador
%   v_fm = velocidade1(dt, ti, tf, vi)
%       Resolução pelo método de Euler da velocidade
%       do saltador de bungee jumping
% entrada:
%   dt = passo de tempo (s)
%   ti = instante de tempo inicial (s)
%   tf = instante de tempo final (s)
%   vi = valor inicial da variável dependente (m/s)
% saída:
%   v_fm = velocidade em tf (m/s)
t = ti;
v = vi;
n = (tf - ti) / dt;
for i = 1:n
    dvdt = deriv(v);
    v = v + dvdt * dt;
    t = t + dt;
end
v_fm = v;
end
function dv=deriv(v)
dv = 9.81 - (0.25 / 68.1) * v*abs(v);
```

Essa função pode ser chamada da command window com o resultado:

```
>> velocidade1(0.5, 0, 12, 0)
ans =
    50.9259
```

Observe que o valor obtido da solução analítica é 50,6175 (Exemplo 3.1); pode-se, então, aplicar um valor bem menor a `dt` para obter um resultado numérico mais acurado:

```
>> velocidade1(0.001, 0, 12, 0)
ans =
    50.6181
```

Embora essa função seja simples de programar, ela não é infalível; ela não funcionará, por exemplo, se o intervalo de cálculo não for divisível pelo passo de tempo. Para abranger tais casos, um laço `while...break` pode substituído no local da área destacada (observe que os comentários foram omitidos para fins de concisão):

```

function v_fm = velocidade2(dt, ti, tf, vi)
t = ti;
v = vi;
h = dt;
while(1)
    if t + dt > tf, h = tf - t; end
    dvdt = deriv(v);
    v = v + dvdt * h;
    t = t + h;
    if t >= tf, break, end
end
v_fm = v;
end

function dv = deriv(v)
dv = 9.81 - (0.25 / 68.1) * v*abs(v);
end

```

Logo que se entra no laço `while`, é utilizada uma estrutura `if` de única linha para testar se a adição $t + dt$ nos levaria para fora do intervalo. Se não (o que seria o caso no início), nada é feito; se sim, encurta-se o intervalo – isto é, define-se o passo variável `h` para o restante do intervalo: $tf - t$, o que garante que o último passo caia exatamente em `tf`. Após implementar esse passo final, o laço será encerrado, pois a condição $t \geq tf$ será verdadeira.

Observe que antes de entrar no laço, atribuiu-se o valor do passo de tempo `dt` a outra variável `h`: essa *variável dummy (variável fictícia)* foi criada de modo que o código não altere o valor dado de `dt` se e quando o passo de tempo for encurtado. Isso é feito antecipando que poderia ser necessário usar o valor original de `dt` em outro lugar no caso de, por exemplo, esse código estar integrado a um programa maior.

Se executarmos essa nova versão, o resultado será o mesmo obtido com a versão baseada na estrutura de laço `for`:

```

>> velocidade2(0.5, 0, 12, 0)
ans =
50.9259

```

Além disso, pode-se usar um `dt` que não é (exatamente) divisível por `tf - ti`:

```

>> velocidade2(0.35, 0, 12, 0)
ans =
50.8348

```

Deve-se notar que o algoritmo ainda não é infalível. O usuário poderia, por engano, entrar com um passo maior que o intervalo de cálculo (por exemplo, $tf - ti = 5$ e `dt = 20`), então seria interessante incluir na função um trecho (ou trechos) de código destinado à detecção de erros e permitir que o usuário corrigisse o equívoco.

Como uma observação final, deve-se reconhecer que o código anterior não é genérico, isto é, ele foi projetado para resolver o problema específico da velocidade do saltador de *bungee jumping*. Uma versão mais genérica pode ser desenvolvida como

```
function y_fm = odesimp(dydt, dt, ti, tf, yi)
t = ti; y = yi; h = dt;
while(1)
    if t + dt > tf, h = tf - t; end
    y = y + dydt(y) * h;
    t = t + h;
    if t >= tf, break, end
end
y_fm = y;
```

Observe como foram retiradas as partes do algoritmo específicas para o exemplo do *bungee jumping* (incluindo a subfunção que define a equação diferencial), mantendo-se apenas as características essenciais da solução. Pode-se, então, utilizar esse programa para resolver o exemplo do saltador de *bungee jumping*, especificando-se a equação diferencial com uma função anônima e passando seu identificador (handle) para que a função-função `odesimp` gere a solução

```
>> dvdt=@(v) 9.81-(0.25/68.1)*v*abs(v);
>> odesimp(dvdt, 0.5, 0, 12, 0)
ans =
50.9259
```

Pode-se, agora, analisar uma função diferente sem a necessidade de entrar no arquivo-M e modificá-lo. Por exemplo, se $y = 10$ em $t = 0$, a equação diferencial $dy/dt = -0,1y$ tem a solução analítica $y = 10e^{-0,1t}$, portanto, a solução em $t = 5$ seria $y(5) = 10e^{-0,1(5)} = 6,0653$. A função-função `odesimp` pode ser usada para obter o mesmo resultado numericamente, como em

```
>> odesimp(@(y) -0.1*y, 0.005, 0, 5, 10)
ans =
6.0645
```

►► PROBLEMAS

3.1 A Figura P3.1 mostra um tanque cilíndrico de base cônica. Se o nível do líquido for muito baixo na parte cônica, o volume será simplesmente o volume cônico de líquido; se o nível de líquido estiver no meio da parte cilíndrica, o volume total de líquido incluirá a parte cônica preenchida e a parte parcialmente ocupada do cilindro.

Use estruturas de decisão para escrever um arquivo-M a fim de calcular o volume do tanque como

uma função dos valores dados de R e d . Desenvolva a função de modo que ela retorne o volume para todos os casos em que a profundidade é menor que $3R$. Retorne uma mensagem de erro (“Transbordo”) se você transbordou o tanque, isto é, $d > 3R$. Teste a função com os seguintes dados:

<i>R</i>	0,9	1,5	1,3	1,3
<i>d</i>	1	1,25	3,8	4,0

Observe que o raio do tanque é R .

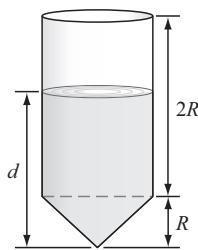


FIGURA P3.1

3.2 Uma quantidade de dinheiro P é investida em uma conta cujos juros são computados no final do período. O valor de retorno F fornecido por uma taxa de juros i , depois de n períodos, pode ser determinado pela seguinte fórmula:

$$F = P(1 + i)^n$$

Escreva uma função no MATLAB (arquivo-M) para calcular o valor de retorno do investimento para cada ano de 1 a n . Os argumentos de entrada da função devem incluir o investimento inicial P , a taxa de juros i (um valor decimal) e o número de anos n para o qual o valor de retorno deve ser calculado. A saída deve consistir em uma tabela com cabeçalhos e colunas para n e F . Execute o programa para $P = \text{R\$ } 100.000,00$; $i = 0,05$ e $n = 10$ anos.

3.3 Há fórmulas econômicas disponíveis para calcular pagamentos anuais de empréstimos. Considere que você tenha emprestado uma quantidade de dinheiro P e tenha concordado em pagar em n pagamentos anuais com uma taxa de juros i . A fórmula para calcular o pagamento anual A é

$$A = P \frac{i(1 + i)^n}{(1 + i)^n - 1}$$

Escreva uma função no MATLAB para calcular A . Teste-a com $P = \text{R\$ } 100.000,00$ e uma taxa de juros de 3,3% ($i = 0,033$). Calcule os resultados para $n = 1, 2, 3, 4$ e 5 anos e mostre o resultado em uma tabela com cabeçalhos e colunas para n e A .

3.4 A temperatura média por dia para uma determinada área pode ser calculada aproximadamente com a seguinte função

$$T = T_{\text{med}} + (T_{\text{max}} - T_{\text{med}}) \cos(\omega(t - t_{\text{max}}))$$

onde T_{med} é a temperatura média anual, T_{max} é a temperatura máxima, ω é a frequência de variação

anual ($= 2\pi/365$) e t_{max} são os dias de temperatura máxima ($\cong 205$ d). Veja os parâmetros para algumas cidades dos Estados Unidos na tabela a seguir:

Cidade	T_{med} (°C)	T_{max} (°C)
Miami, FL	22,1	28,3
Yuma, AZ	23,1	33,6
Bismarck, ND	5,2	22,1
Seattle, WA	10,6	17,6
Boston, MA	10,7	22,9

Desenvolva uma função no MATLAB que calcule a temperatura média entre dois dias do ano para uma cidade particular. Teste a função para (a) janeiro-fevereiro em Yuma, AZ ($t = 0$ a 59) e (b) julho-agosto em Seattle, WA ($t = 180$ a 242).

3.5 A função seno pode ser calculada pela seguinte série infinita:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

Desenvolva uma função no MATLAB para implementar essa fórmula de modo a calcular e exibir os valores de $\sin x$ assim que cada termo da série for acrescentado. Em outras palavras, calcule e imprima a sequência de valores para

$$\begin{aligned}\sin x &= x \\ \sin x &= x - \frac{x^3}{3!} \\ \sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} \\ &\vdots\end{aligned}$$

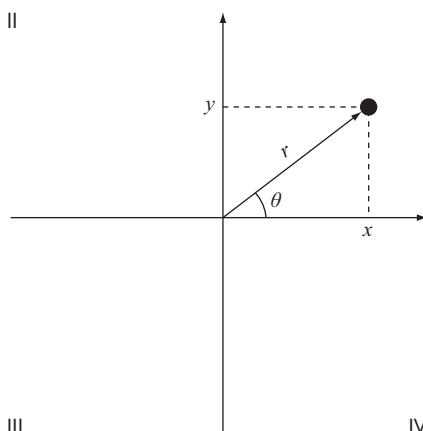
até o termo de ordem n escolhido. Para cada valor especificado, calcule e mostre o erro relativo percentual como

$$\% \text{erro} = \frac{\text{valor real} - \text{aproximação}}{\text{valor real}} \times 100\%$$

Como um teste, utilize o programa para calcular $\sin(0,9)$ empregando oito termos – isto é, até o termo $x^{15}/15!$.

3.6 Para especificar a localização de um ponto com relação à origem do espaço bidimensional (Figura P3.6) são necessárias duas distâncias:

- ✓ As distâncias horizontal e vertical (x, y) em coordenadas cartesianas.
- ✓ O raio e o ângulo (r, θ) em coordenadas polares.

**FIGURA P3.6**

É relativamente simples calcular as coordenadas cartesianas (x, y) com base nas coordenadas polares (r, θ), mas o inverso não é tão simples. O raio pode ser calculado pela seguinte fórmula:

$$r = \sqrt{x^2 + y^2}$$

Se as coordenadas estiverem no primeiro e no quarto quadrantes (isto é, $x > 0$), então uma fórmula simples pode ser utilizada para calcular θ :

$$\theta = \tan^{-1}\left(\frac{y}{x}\right)$$

A dificuldade surge para os demais casos. A tabela a seguir sintetiza as possibilidades:

x	y	θ
<0	>0	$\tan^{-1}(y/x) + \pi$
<0	<0	$\tan^{-1}(y/x) - \pi$
<0	=0	π
=0	>0	$\pi/2$
=0	<0	$-\pi/2$
=0	=0	0

Escreva um arquivo-M bem estruturado utilizando a estrutura `if...elseif` para calcular r e θ como uma função de x e y e expresse os resultados finais para θ em graus. Teste seu programa para avaliar os seguintes casos:

x	y	r	θ
2	0		
2	1		
0	3		
-3	1		
-2	0		
-1	-2		
0	0		
0	-2		
2	2		

3.7 Desenvolva um arquivo-M para determinar as coordenadas polares como descrito no Problema 3.6. No entanto, em vez de projetar a função para avaliar um único caso, passe vetores de x e y . A função deve exibir os resultados como uma tabela com colunas para x, y, r e θ . Teste o programa para os casos delineados no Problema 3.6.

3.8 Desenvolva uma função no MATLAB para a qual é inserida uma nota em valor numérico de 0 a 100 e que retorna uma nota em caracteres (conceito) segundo o esquema:

Letra	Critério
A	$90 \leq \text{valor numérico} \leq 100$
B	$80 \leq \text{valor numérico} < 90$
C	$70 \leq \text{valor numérico} < 80$
D	$60 \leq \text{valor numérico} < 70$
F	valor numérico < 60

A primeira linha da função deve ser

```
function nota = conceito(valor_numérico)
```

Desenvolva a função de modo que ela exiba uma mensagem de erro e seja encerrada no caso em que o usuário entra com um `valor_numérico` menor que 0 ou maior que 100. Teste sua função com 89,9999; 90; 45 e 120.

3.9 A Equação de Manning pode ser usada para calcular a velocidade da água em um canal aberto retangular,

$$U = \frac{\sqrt{S}}{n} \left(\frac{B H}{B + 2H} \right)^{2/3}$$

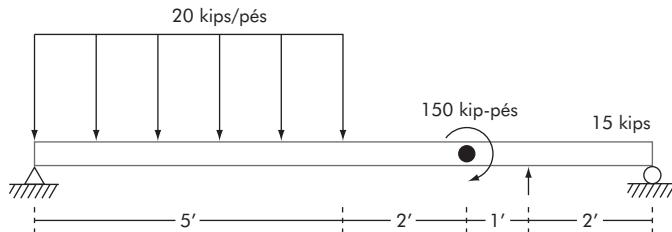


FIGURA P3.10

onde U é a velocidade (m/s), S é a inclinação do canal, n é o coeficiente de rugosidade, B é a largura (m) e H é a profundidade (m). Os seguintes dados estão disponíveis para cinco canais:

n	S	B	H
0,036	0,0001	10	2
0,020	0,0002	8	1
0,015	0,0012	20	1,5
0,030	0,0007	25	3
0,022	0,0003	15	2,6

Escreva uma função no MATLAB que calcule a velocidade para cada um desses canais. Insira esses valores em uma matriz em que cada coluna represente um parâmetro e cada linha represente um canal. O arquivo-M deve exibir os dados de entrada com a velocidade calculada na forma de tabela, onde a velocidade é a quinta coluna. Inclua cabeçalhos na tabela para identificar as colunas.

3.10 Uma viga de suporte simples está carregada como ilustrado na Figura P3.10. Usando funções de singularidade, o deslocamento ao longo da viga pode ser expresso pela equação:

$$u_y(x) = \frac{-5}{6}[(x - 0)^4 - (x - 5)^4] + \frac{15}{6}(x - 8)^3 + 75(x - 7)^2 + \frac{57}{6}x^3 - 238,25x$$

Por definição, a função de singularidade pode ser expressa como segue:

$$(x - a)^n = \begin{cases} (x - a)^n & \text{quando } x > a \\ 0 & \text{quando } x \leq a \end{cases}$$

Desenvolva uma função no MATLAB que trace um gráfico do deslocamento (linha tracejada) versus a distância ao longo da viga, x . Observe que $x = 0$ na extremidade esquerda da viga.

3.11 O volume V de líquido em um cilindro horizontal oco de raio r e comprimento L é relacionado à profundidade do líquido h por

$$V = \left[r^2 \cos^{-1} \left(\frac{r - h}{r} \right) - (r - h)\sqrt{2rh - h^2} \right] L$$

Desenvolva uma função no MATLAB para criar um gráfico do volume versus a profundidade. Aqui estão as primeiras linhas:

```
function cilindro(r, L, titulo_grafco)
% volume de um cilindro horizontal
% entradas:
% r = raio
% L = comprimento
% titulo_grafco = string contendo o
% título do gráfico
```

Teste seu programa com

```
>> cilindro(3, 5...
'Volume versus profundidade para um
tanque...
cilíndrico horizontal')
```

3.12 Desenvolva uma versão vetorizada do seguinte código:

```
t_inicio=0; t_fm=20; ni=8;
t(1)=t_inicio;
y(i)=12+6*cos(2*pi*t(i)/(t_fim-t_
inicio));
for i=2:ni+1
    t(i)=t(i-1)+(t_fm-t_inicio)/ni;
    y(i)=10 + 5*cos(2*pi*t(i)/...
        (t_fm-t_inicio));
end
```

3.13 O método babilônico, um antigo sistema para aproximação da raiz quadrada de qualquer número positivo a , pode ser formulado como

$$x = \frac{x + a/x}{2}$$

Escreva uma função bem estruturada no MATLAB com base na estrutura de laço `while...break` para implementar esse algoritmo. Use indentação adequada, de modo que a estrutura do código fique clara. Em cada passo, estime o erro em sua aproximação como

$$\varepsilon = \left| \frac{x_{\text{novo}} - x_{\text{velho}}}{x_{\text{novo}}} \right|$$

Repita o laço até que ε seja menor ou igual a um valor específico e projete seu programa de modo que ele retorne o resultado e o erro. Certifique-se de que ele possa avaliar a raiz quadrada de números iguais ou menores que zero. Para o último caso, exiba o resultado como um número imaginário, por exemplo, a raiz quadrada de -4 seria retornada como $2i$. Teste seu programa avaliando $a = 0, 2, 10$ e -4 para $\varepsilon = 1 \times 10^{-4}$.

3.14 Funções definidas por partes às vezes são úteis quando a relação entre uma variável dependente e uma variável independente não pode ser adequadamente representada por uma única equação. Por exemplo, a velocidade de um foguete poderia ser descrita por

$$v(t) = \begin{cases} 10t^2 - 5t & 0 \leq t \leq 8 \\ 624 - 3t & 8 \leq t \leq 16 \\ 36t + 12(t - 16)^2 & 16 \leq t \leq 26 \\ 2136e^{-0.1(t-26)} & t > 26 \\ 0 & \text{caso contrário} \end{cases}$$

Desenvolva uma função no MATLAB para calcular v como uma função de t . Em seguida, desenvolva um script que utilize essa função para gerar um gráfico de v versus t para $t = -5$ a 50 .

3.15 Desenvolva uma função no MATLAB chamada `arredonda` para arredondar um número x para um número especificado de n dígitos decimais. A primeira linha da função deve ser definida como

```
function xr = arredonda(x, n)
```

Teste sua função arredondando cada um dos números a seguir para dois dígitos decimais: $x = 477,9587; -477,9587; 0,125; 0,135; -0,125$ e $-0,135$.

3.16 Desenvolva uma função no MATLAB para determinar os dias decorridos em um ano. A primeira linha da função deve ser definida como,

```
function nd = dias(mês, dia, bissexto)
```

onde mês é o mês (1–12), dia é o dia (1–31) e bissexto deve ser 0 para ano não bissexto e 1 para ano bissexto. Teste sua função para 1 de janeiro de 1997, 29 de fevereiro de 2004, 1 de março de 2001, 21 de junho de 2004 e 31 de dezembro de 2008. Dica: uma boa maneira para fazer isso é combinar as estruturas `for` e `switch`.

3.17 Desenvolva uma função no MATLAB para determinar os dias decorridos em um ano. A primeira linha da função deve ser definida como

```
function nd = dias(mês, dia, ano)
```

onde mês é o mês (1–12), dia é o dia (1–31) e ano é o ano. Teste sua função para 1 de janeiro de 1997, 29 de fevereiro de 2004, 1 de março de 2001, 21 de junho de 2004 e 31 de dezembro de 2008.

3.18 Desenvolva uma função-função no MATLAB que retorne a diferença entre o valor máximo e mínimo da função importada, dado um intervalo da variável independente. A função-função deve gerar um gráfico da função importada para o intervalo. Teste seu programa para os seguintes casos:

- (a) $f(t) = 8e^{-0.25t}\sin(t - 2)$ para $t = 0$ a 6π .
- (b) $f(x) = e^{4x}\sin(1/x)$ para $x = 0,01$ a $0,2$.
- (c) A função nativa `humps` para $x = 0$ a 2 .

3.19 Modifique a função `odesimp` desenvolvida no fim da Seção 3.6 de modo que se possa passar os argumentos da função importada. Teste-a para o seguinte caso:

```
>> dvdt=@(v,m,cd) 9.81-(cd/m)*v^2;
>> odesimp(dvdt,0.5,0,12,-10,70,0.23)
```

3.20 Um vetor em coordenadas cartesianas pode ser pensado como uma representação de valores ao longo dos eixos x , y e z multiplicados por um vetor unitário (i , j , k). Para tais casos, o produto escalar de dois desses vetores $\{a\}$ e $\{b\}$ corresponde ao produto de seus módulos e do cosseno do ângulo entre eles, como em

$$\{a\} \cdot \{b\} = ab \cos\theta$$

O produto vetorial produz outro vetor, $\{c\} = \{a\} \times \{b\}$, que é perpendicular ao plano definido por $\{a\}$ e $\{b\}$ tal que seu sentido é especificado pela regra da mão direita. Desenvolva uma função no MATLAB que tenha como argumentos de entrada dois vetores em coordenadas cartesianas e retorne θ , $\{c\}$ e o módulo de $\{c\}$, e gere um gráfico tridimensional dos três vetores $\{a\}$, $\{b\}$ e $\{c\}$ com suas origens no zero. Use linhas tracejadas para $\{a\}$ e $\{b\}$

e uma linha sólida para $\{c\}$ e teste sua função para os casos a seguir:

- (a) $a = [6 \ 4 \ 2]; b = [2 \ 6 \ 4];$
- (b) $a = [3 \ 2 \ -6]; b = [4 \ -3 \ 1];$
- (c) $a = [2 \ -2 \ 1]; b = [4 \ 2 \ -4];$
- (d) $a = [-1 \ 0 \ 0]; b = [0 \ -1 \ 0];$

3.21 Com base no Exemplo 3.6, desenvolva um script para produzir uma animação de uma bola quicando, sendo que $v_0 = 5$ m/s e $\theta_0 = 50^\circ$. Para fazer isso, você deve ser capaz de prever exatamente quando a bola atinge o chão. Nesse ponto, o sentido muda (o novo ângulo será igual ao negativo do ângulo no impacto), e a velocidade decrescerá em módulo para refletir as perdas de energia causadas pela colisão da bola com o chão. A alteração na velocidade pode ser quantificada pelo *coeficiente de restituição* C_r , que é igual à razão entre a velocidade depois e a velocidade antes do impacto. Para o presente caso, use um valor de $C_r = 0,8$.

3.22 Desenvolva uma função para produzir a animação de uma partícula se movendo em um círculo em coordenadas cartesianas com base em coordenadas radiais. Assuma um raio constante, r , e permita o ângulo, θ , aumentar de 0 a 2π em incrementos iguais. As primeiras linhas da função devem ser

```
function fasor(r, nt, nm)
% função para mostrar a órbita de um fasor
% r = raio
% nt = número de incrementos para theta
% nm = número de filmes (movies)
```

Teste sua função com

```
fasor(1, 256, 10)
```

3.23 Desenvolva um script para produzir um filme para o gráfico da borboleta do Problema 2.22. Use uma partícula localizada nas coordenadas x - y para visualizar como o gráfico evolui no tempo.

4

Erros de arredondamento e de truncamento

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é familiarizar o leitor com as principais fontes de erros envolvidas em métodos numéricos. Após esta leitura completa, você será capaz de:

- ✓ Entender a diferença entre acurácia e precisão.
- ✓ Aprender como quantificar o erro.
- ✓ Aprender como usar estimativas de erro para decidir quando encerrar um cálculo iterativo.
- ✓ Entender como erros de arredondamento ocorrem em computadores digitais por conta de sua capacidade limitada para representar números.
- ✓ Entender por que existem limites de faixa e de precisão na representação de números em ponto flutuante.
- ✓ Reconhecer que os erros de truncamento ocorrem quando formulações matemáticas exatas são representadas por aproximações.
- ✓ Aprender como usar a série de Taylor para estimar erros de truncamento.
- ✓ Escrever aproximações por diferença finita progressiva, regressiva e centrada para primeira e segunda derivadas.
- ✓ Reconhecer que esforços para minimizar erros de truncamento podem, às vezes, aumentar os erros de arredondamento.

VOCÊ TEM UM PROBLEMA

No Capítulo 1, desenvolvemos um modelo numérico para calcular a velocidade de um saltador de *bungee jumping*. Para resolver o problema com um computador, você teve de aproximar a derivada da velocidade com uma diferença finita.

$$\frac{dv}{dt} \cong \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}$$

Assim, a solução resultante não é exata, isto é, ela tem um erro.

Além disso, o computador que você utiliza para obter a solução é uma ferramenta imperfeita, limitada em sua capacidade de representar os valores e a precisão de números. Consequentemente, a própria máquina fornece resultados que contêm erros.

Assim, tanto a aproximação matemática como o computador fazem a predição de seu modelo ser incerta. Como lidar com essa incerteza? É possível compreender,

quantificar e controlar esses erros de modo a obter resultados aceitáveis? Este capítulo apresenta uma introdução a abordagens e conceitos que engenheiros e cientistas utilizam para lidar com esse dilema.

4.1 ERROS

Engenheiros e cientistas constantemente têm de trabalhar e alcançar objetivos com base em informações incertas, mas ainda que a perfeição seja uma meta louvável, ela raramente é atingida (se alguma vez já foi). Por exemplo, embora o modelo desenvolvido a partir da segunda lei de Newton seja uma aproximação excelente, ele nunca prediria exatamente, na prática, a queda do saltador, pois uma variedade de fatores, como vento e pequenas variações na resistência do ar, causaria desvios no valor previsto. Se esses desvios fossem sistematicamente para cima ou para baixo, então poderia ser necessário desenvolver um novo modelo; entretanto, se eles fossem distribuídos aleatoriamente e agrupados em torno da previsão, então os desvios poderiam ser considerados desprezíveis, e o modelo, adequado.

Este capítulo aborda os tópicos básicos relacionados à identificação, quantificação e minimização desses erros. As Seções 4.2 e 4.3 tratam das duas principais formas de erros numéricos: o erro de arredondamento (causado pelas aproximações dos computadores) e o erro de truncamento (pelas aproximações matemáticas). Este capítulo também descreve como estratégias para reduzir o erro de truncamento algumas vezes provocam o aumento desses erros. Por fim, são discutidos brevemente os erros que não estão associados diretamente aos próprios métodos numéricos, como enganos, erros na formulação ou no modelo e incerteza dos dados.

4.1.1 Acurácia e precisão

Os erros associados tanto aos cálculos quanto às medidas podem ser caracterizados com relação à sua acurácia e precisão. A *acurácia* se refere a quão próximo o valor calculado ou medido está do valor verdadeiro, enquanto a *precisão* se refere a quão próximos os valores individuais calculados ou medidos estão uns dos outros.

Esses conceitos podem ser ilustrados graficamente usando-se uma analogia da prática de tiro ao alvo. Os buracos de bala em cada alvo na Figura 4.1 representam as previsões de uma técnica numérica, enquanto a mosca (centro do alvo) representa a verdade. A *inacurácia* (também chamada de *viés*) é definida como um desvio sistemático da verdade; assim, embora os tiros na Figura 4.1c estejam agrupados mais juntos do que aqueles na Figura 4.1a, os dois casos são igualmente inacurados, porque ambos estão centrados no quadrante superior esquerdo do alvo. As *imprecisões* (também chamadas *incertezas*), por outro lado, referem-se à intensidade do espalhamento; portanto, embora as Figuras 4.1b e d sejam igualmente exatas (isto é, centradas na mosca), a última é mais precisa porque os tiros estão mais agrupados.

Os métodos numéricos deveriam ser acurados (ou sem viés) e precisos o suficiente para permitir a realização de projetos e a resolução de problemas de engenharia. Neste livro será usado o termo coletivo *erro* para representar tanto a inacurácia quanto a imprecisão; com esses conceitos como base, é possível discutir os fatores que contribuem para o erro nos cálculos numéricos.

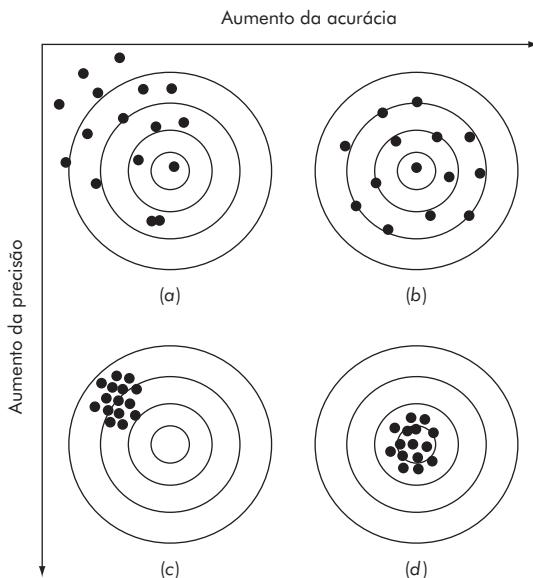


FIGURA 4.1 Um exemplo do tiro ao alvo que ilustra os conceitos de acurácia e precisão. (a) Inacurado e impreciso, (b) acurado e impreciso, (c) inacurado e preciso, (d) acurado e preciso.

4.1.2 Definições de erro

Os erros numéricos são causados pelo uso de aproximações para representar operações e quantidades matemáticas exatas. Para tais erros, a relação entre o resultado exato, ou verdadeiro, e a aproximação pode ser formulada como

$$\text{Valor verdadeiro} = \text{aproximação} + \text{erro} \quad (4.1)$$

Rearranjando a Equação (4.1), conclui-se que o erro numérico é igual à discrepância entre o valor verdadeiro e a aproximação, ou seja,

$$E_t = \text{valor verdadeiro} - \text{aproximação} \quad (4.2)$$

onde E_t é usado para designar o valor exato do erro. O subscrito t é incluído para designar que esse é o erro “verdadeiro” (*true*, em inglês), o que contrasta com outros casos, como será descrito em breve, em que uma estimativa “aproximada” do erro deve ser empregada. Observe que o erro verdadeiro é geralmente expresso como um valor absoluto e referido como o *erro absoluto*.

Um defeito dessa definição é que ela não leva em conta a ordem de grandeza do valor que está sendo examinado, ou seja, um erro de um centímetro é muito mais significativo quando se mede um rebite do que uma ponte. Uma forma de considerar o valor das quantidades que estão sendo calculadas é normalizar o erro com o valor verdadeiro, como em

$$\text{Erro relativo fracionário verdadeiro} = \frac{\text{valor verdadeiro} - \text{aproximação}}{\text{valor verdadeiro}}$$

O erro relativo também pode ser multiplicado por 100% para expressá-lo como

$$\varepsilon_t = \frac{\text{valor verdadeiro} - \text{aproximação}}{\text{valor verdadeiro}} \cdot 100\% \quad (4.3)$$

onde ε_t designa o erro relativo percentual verdadeiro.

Considere que você tenha a tarefa de medir os comprimentos de uma ponte e de um rebite e que obteve 9.999 e 9 cm, respectivamente; se os valores verdadeiros são 10.000 e 10 cm, respectivamente, o erro em ambos os casos é 1 cm. Entretanto, os erros relativos percentuais podem ser calculados com a Equação (4.3), resultando em 0,01% e 10%, respectivamente. Assim, embora ambas as medidas tenham um erro de 1 cm, a conclusão é de que foi feito um trabalho adequado na medição da ponte, enquanto a estimativa para o rebite deixou muito a desejar.

Observe que, para as Equações (4.2) e (4.3), E e ε têm o subscrito t para indicar que o erro está normalizado com relação ao valor verdadeiro. Para o exemplo do rebite e da ponte, esse valor foi fornecido; entretanto, nas situações reais, tais informações raramente estão disponíveis. Para os métodos numéricos, o valor verdadeiro será conhecido apenas ao lidar com funções que podem ser resolvidas analiticamente, como quando se quer investigar o comportamento teórico de uma técnica particular para sistemas simples. Nas aplicações do mundo real, entretanto, não se conhecerá *a priori* a resposta verdadeira. Para tais situações, uma alternativa é normalizar o erro usando a melhor estimativa disponível do valor verdadeiro, isto é, a aproximação propriamente dita, como em

$$\varepsilon_a = \frac{\text{erro aproximado}}{\text{aproximação}} \cdot 100\% \quad (4.4)$$

onde o subscrito a significa que o erro foi normalizado por um valor aproximado. Observe também que, para aplicações do mundo real, a Equação (4.2) não pode ser usada para calcular o termo do erro no numerador da Equação (4.4). Um dos desafios dos métodos numéricos é determinar estimativas de erro quando se desconhece o valor verdadeiro. Por exemplo, certos métodos numéricos usam uma *abordagem iterativa* para calcular as respostas, ou seja, uma aproximação atual é feita com base em uma aproximação prévia. Esse processo é realizado repetidamente, ou iterativamente, para se calcular com sucesso (é o que se espera) aproximações cada vez melhores, assim, o erro é estimado como a diferença entre as aproximações prévia e atual, e sua versão percentual é determinada de acordo com

$$\varepsilon_a = \frac{\text{aproximação atual} - \text{aproximação prévia}}{\text{aproximação atual}} \cdot 100\% \quad (4.5)$$

Essa e outras abordagens para expressar erros serão mais bem discutidas em capítulos subsequentes.

Os sinais nas Equações (4.2) até (4.5) podem ser tanto positivos quanto negativos. Se a aproximação for maior que o valor verdadeiro (ou a aproximação prévia for maior que a aproximação atual), o erro será negativo; se a aproximação for menor que o valor verdadeiro, o erro será positivo. Além disso, para as Equações (4.3) a (4.5), o denominador pode ser menor que zero, o que também pode levar a um erro negativo. Ao se realizar cálculos normalmente não há preocupações com o sinal do

erro, mas interesse em saber se o valor absoluto percentual é menor que uma tolerância percentual pré-especificada ε_s . Portanto, em geral, é útil usar o valor absoluto da Equação (4.5); para tais casos, os cálculos são repetidos até que

$$|\varepsilon_a| < \varepsilon_s \quad (4.6)$$

Essa relação é chamada de *critério de parada*: se ela for satisfeita, supõe-se que o resultado esteja dentro do nível aceitável pré-especificado ε_s . Observe que aqui serão utilizados quase exclusivamente valores absolutos quando falarmos de erros relativos.

Também é conveniente relacionar esses erros ao número de algarismos significativos na aproximação. Segundo Scarborough (1966), se o seguinte critério for satisfeito, é possível ter certeza de que o resultado é correto até *pelo menos n* algarismos significativos.

$$\varepsilon_s = (0,5 \times 10^{2-n})\% \quad (4.7)$$

EXEMPLO 4.1 Estimativa de erro para métodos iterativos

Em matemática, as funções geralmente podem ser representadas por séries infinitas. Por exemplo, a função exponencial pode ser calculada usando-se

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} \quad (\text{E4.1.1})$$

Portanto, conforme mais termos forem adicionados à sequência, a estimativa se tornará cada vez mais próxima do valor verdadeiro de e^x . A Equação (E.4.1.1) é chamada de *expansão em série de Maclaurin*.

Começando com a versão mais simples, $e^x = 1$, some um termo de cada vez para estimar $e^{0.5}$. Depois que cada termo for adicionado, calcule o erro verdadeiro e o erro relativo percentual aproximado com as Equações (4.3) e (4.5), respectivamente, e observe que o valor verdadeiro é $e^{0.5} = 1,648721 \dots$ Adicione termos até que o valor absoluto do erro estimado aproximado ε_a esteja dentro do critério de erro pré-especificado ε_s que garanta três algarismos significativos.

Solução. Inicialmente, a Equação (4.7) pode ser empregada para determinar o critério de erro que garante o resultado correto até pelos menos três algarismos significativos:

$$\varepsilon_s = (0,5 \times 10^{2-3})\% = 0,05\%$$

Portanto, serão adicionados termos à série até que ε_a caia abaixo desse nível.

A primeira estimativa é igual à Equação (E4.1.1), com um único termo; assim a primeira estimativa é igual a 1. A segunda estimativa é então gerada somando-se o segundo termo, como em

$$e^x = 1 + x$$

ou, para $x = 0,5$

$$e^{0.5} = 1 + 0,5 = 1,5$$

Isso representa um erro relativo percentual verdadeiro de [Equação (4.3)]

$$\varepsilon_t = \left| \frac{1,648721 - 1,5}{1,648721} \right| \times 100\% = 9,02\%$$

A Equação (4.5) pode ser usada para determinar uma estimativa aproximada do erro, como em

$$\varepsilon_a = \left| \frac{1,5 - 1}{1,5} \right| \times 100\% = 33,3\%$$

Como ε_a não é menor que o valor pedido de ε_s , os cálculos continuariam com a soma de um outro termo, $x^2/2!$, repetindo-se os cálculos de erro até que $|\varepsilon_a| < \varepsilon_s$. Todos esses cálculos podem ser resumidos em

Termos	Resultados	ε_r (%)	ε_a (%)
1	1	39,3	
2	1,5	9,02	33,3
3	1,625	1,44	7,69
4	1,645833333	0,175	1,27
5	1,648437500	0,0172	0,158
6	1,648697917	0,00142	0,0158

Portanto, depois que seis termos são incluídos, o erro aproximado cai abaixo de $\varepsilon_s = 0,05\%$ e os cálculos são encerrados; entretanto, observe que, em vez de três algarismos significativos, os resultados são exatos até cinco! Isso acontece porque, nesse caso, tanto a Equação (4.5) quanto a (4.7) são conservadoras – ou seja, garantem que o resultado seja tão satisfatório quanto elas especificam. Embora isso não seja a regra para a Equação (4.5), costuma ser verdade na maioria das vezes.

4.1.3 Algoritmo computacional para cálculos iterativos

Muitos dos métodos numéricos descritos no restante deste texto envolvem cálculos iterativos do tipo ilustrado no Exemplo 4.1: todos eles implicam na solução de um problema matemático pelo cálculo de sucessivas aproximações para a solução, partindo de uma aproximação inicial.

A implementação computacional de tais soluções iterativas envolve laços. Como vimos na Seção 3.3.2, eles podem ser basicamente de dois tipos: laços controlados por contador (termina depois de um número específico de repetições) e laços controlados por uma condição (termina com base em uma condição lógica). A maior parte das soluções iterativas utiliza laços condicionais; assim, em vez de empregar um número pré-especificado de iterações, o processo é repetido até que um erro aproximado fique abaixo de um critério de parada, como no Exemplo 4.1.

Ao fazer isso para o mesmo problema do Exemplo 4.1, a expansão em série pode ser expressa como

$$e^x \cong \sum_{i=0}^n \frac{x^n}{n!}$$

Uma função do MATLAB que implementa essa fórmula é apresentada na Figura 4.2 e tem os seguintes parâmetros de entrada: o valor a ser avaliado (x), o critério de

parada para o erro (`es`) e o valor máximo permitido de iterações (`maxit`). Se o usuário omitir qualquer um dos dois últimos parâmetros, a função atribui valores-padrão predefinidos (`default`).

A função inicializa três variáveis: (a) `iter`, que mantém o controle do número de iterações, (b) `sol`, que armazena a estimativa atual da solução, e (c) a variável `ea`, que armazena o erro relativo percentual aproximado. Observe que a variável `ea` é inicializada com um valor igual a 100 para garantir que o laço seja executado ao menos uma vez.

Essas inicializações são seguidas por um laço condicional que implementa efetivamente o cálculo iterativo. Antes de gerar uma nova solução, o valor prévio, `sol`, é atribuído à variável `sol_previa`; após isso, um novo valor de `sol` é calculado e o contador de iterações é incrementado. Se o novo valor de `sol` for diferente de zero, o erro relativo percentual, `ea`, é determinado. Os critérios de parada são, então, testados: se ambos forem falsos, o laço é repetido; se um deles for verdadeiro, o laço é interrompido e a solução final é retornada como parâmetro de saída da função.

```
function [fx,ea,iter] = IterMeth(x,es,maxit)
% Série de Maclaurin da função exponencial
% [fx,ea,iter] = IterMeth(x,es,maxit)
% entrada:
%   x = valor no qual a série é avaliada
%   es = critério de parada (padrão = 0.0001)
%   maxit = número máximo de iterações (default = 50)
% saída:
%   fx = valor estimado
%   ea = erro relativo aproximado (%)
%   iter = número de iterações

% padrões:
if nargin<2|isempty(es),es=0.0001;end
if nargin<3|isempty(maxit),maxit=50;end
% inicialização
iter = 1; sol = 1; ea = 100;
% cálculo iterativo
while (1)
    sol_previa = sol;
    sol = sol + x ^ iter / factorial(iter);
    iter = iter + 1;
    if sol~=0
        ea=abs((sol - sol_previa)/sol)*100;
    end
    if ea<=es | iter>=maxit,break,end
end
fx = sol;
end
```

FIGURA 4.2 Função do MATLAB para resolver um cálculo iterativo. Esse código está configurado para avaliar a expansão em série de Maclaurin para a função e^x , como descrito no Exemplo 4.1.

Quando a função é executada, ela gera uma estimativa para a função exponencial, retornada com o erro aproximado e o número de iterações. Por exemplo, e^1 pode ser avaliado como

```
>> format long  
>> [valor_aprox, ea, iter] = IterMeth(1,1e-6,100)  
valor_aprox =  
    2.718281826198493  
ea =  
    9.216155641522974e-007  
iter =  
    12
```

Pode-se ver que, após 12 iterações, obtém-se o resultado de 2,7182818 com um erro aproximado estimado de $9,2162 \times 10^{-7}$ %. O resultado pode ser verificado utilizando-se a função nativa `exp` para calcular diretamente o valor exato e o erro relativo percentual verdadeiro,

```
>> valor_verdadeiro=exp(1)  
valor_verdadeiro =  
    2.718281828459046  
>> et=abs((valor_verdadeiro-valor_aprox)/valor_verda-  
deiro)*100  
et =  
    8.316108397236229e-008
```

Assim como no Exemplo 4.1, obtemos o resultado desejável de que o erro verdadeiro é menor que o erro aproximado.

4.2 ERROS DE ARREDONDAMENTO

Os *erros de arredondamento* surgem porque os computadores digitais não podem representar algumas quantidades de forma exata na solução de problemas científicos e de engenharia, levando a resultados errados. Em certos casos, eles podem conduzir os cálculos a instabilidades numéricas, fornecendo resultados claramente incorretos ou *mal condicionados*. Pior ainda: os erros de arredondamento podem levar a discrepâncias sutis difíceis de detectar.

Há dois aspectos principais sobre os erros de arredondamento envolvidos em cálculos numéricos:

1. Os computadores digitais têm limites de tamanho e precisão em sua capacidade para representar números.
2. Certas manipulações numéricas são altamente sensíveis a erros de arredondamento, o que pode resultar tanto de considerações matemáticas quanto do modo como o computador realiza as operações aritméticas.

4.2.1 Representação dos números no computador

Os erros numéricos de arredondamento estão diretamente relacionados à maneira como os números são armazenados no computador. A unidade fundamental na qual a informação é representada é chamada de *palavra*, uma entidade que consiste em uma sequência de dígitos binários (*binary digits*), ou *bits*. Os números são armazenados em

uma ou mais palavras – para entender como isso é feito, primeiro é necessário revisar alguns conceitos relacionados aos sistemas numéricos.

Um *sistema numérico* é meramente uma convenção para representar quantidades e como são 10 dedos na mão e 10 dedos no pé, o sistema numérico com o qual temos mais familiaridade é o sistema *decimal* ou o sistema na *base 10*. A base é o número usado como referência para construir o sistema. O sistema na base 10 usa os 10 algarismos – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 – para representar os números. Por si sós, esses algarismos são satisfatórios para contar de 0 a 9.

Para quantidades maiores, combinações desses algarismos básicos devem ser usadas, com a posição ou o *valor do lugar* especificando a ordem de grandeza. O algarismo mais à direita em um número completo representa um número entre 0 e 9; o segundo algarismo a partir da direita representa um múltiplo de 10; o terceiro algarismo a partir da direita representa um múltiplo de 100 e assim por diante. Por exemplo, no número 8642,9 há oito grupos de 1000, seis grupos de 100, quatro grupos de 10, dois grupos de 1 e nove grupos de 0,1, ou

$$(8 \times 10^3) + (6 \times 10^2) + (4 \times 10^1) + (2 \times 10^0) + (9 \times 10^{-1}) = 8642,9$$

Esse tipo de representação é chamada *notação posicional*.

Como o sistema decimal é tão familiar, em geral não se percebe que existem alternativas; por exemplo, se os seres humanos tivessem oito dedos na mão e oito dedos no pé, sem dúvida teriam desenvolvido um sistema *octogonal* ou uma representação na *base 8*. Nesse mesmo sentido, o computador é como um animal de dois dedos limitado a dois estados – ou 0 ou 1. Isso está relacionado ao fato de que as unidades lógicas primárias dos computadores digitais são componentes elétricas ligadas ou desligadas; portanto, os números em um computador são representados com um sistema *binário* ou na *base 2*. Da mesma forma como no sistema decimal, as quantidades podem ser representadas usando-se a notação posicional, por exemplo: o número binário 101,1 é equivalente a $(1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (1 \times 2^{-1}) = 4 + 0 + 1 + 0,5 = 5,5$ no sistema decimal.

Representação inteira. Agora que foi revisado como o número na base 10 pode ser representado na forma binária, é simples conceber como os inteiros são representados em um computador. A abordagem mais direta, chamada de *método dos valores com sinal*, utiliza o primeiro bit de uma palavra para indicar o sinal, com um 0 para positivo e um 1 para negativo, já os bits restantes são usados para armazenar o número. Por exemplo, o valor inteiro 173 é representado em binário como 10101101:

$$(10101101)_2 = 2^7 + 2^5 + 2^3 + 2^2 + 2^0 = 128 + 32 + 8 + 4 + 1 = (173)_{10}$$

Portanto, o binário equivalente de -173 seria armazenado em um computador de 16 bits, como mostra a Figura 4.3.

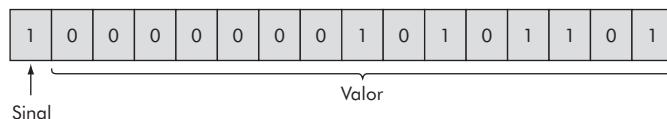


FIGURA 4.3 A representação binária de um inteiro decimal -173 em um computador de 16 bits usando-se o método do valor com sinal.

Se esse esquema for empregado, haverá claramente uma faixa limitada de inteiros que poderão ser representados. Novamente assumindo um tamanho de palavra de 16 bits, se um bit é usado para o sinal, os 15 bits restantes podem representar inteiros binários de 0 a 111111111111111. O limite superior pode ser convertido para um número decimal inteiro, como em $(1 \times 2^{14}) + (1 \times 2^{13}) + \dots + (1 \times 2^1) + (1 \times 2^0) = 32.767$. Observe que essa expressão pode ser calculada simplesmente como $2^{15} - 1$, portanto, uma palavra no computador de 16 bits pode armazenar inteiros decimais que variam de -32.768 a 32.767.

Além disso, como o zero já foi definido como 0000000000000000, é redundante usar o número 1000000000000000 para definir um “menos zero”; assim, ele é normalmente empregado para representar um número negativo adicional: -32.768, e o intervalo é de -32.768 a 32.767. Para uma palavra de n -bits, a faixa seria de -2^{n-1} a $2^{n-1} - 1$, portanto, inteiros de 32 bits variariam de -2.147.483.648 a 2.147.483.647.

Observe que, embora ele forneça uma boa maneira de ilustrar o assunto, o método dos valores com sinal não é usado para representar inteiros nos computadores convencionais. Uma abordagem preferível, chamada de técnica do *complemento de 2*, incorpora diretamente o sinal no valor absoluto do número em vez de fornecer um bit separado para representar mais ou menos; independentemente, a faixa de números ainda é a mesma que para o método dos valores com sinal descrito acima.

Tudo isso serve para ilustrar como os computadores digitais são limitados em sua capacidade de representar inteiros, isto é, os números acima ou abaixo de um intervalo não podem ser representados. Uma limitação mais séria diz respeito ao armazenamento e à manipulação de quantidades fracionárias, como descrito a seguir.

Representação em ponto flutuante. As quantidades fracionárias costumam ser representadas em computadores usando-se o *formato de ponto flutuante*, que é muito parecido com a notação científica e é expresso como

$$\pm s \times b^e$$

onde s é o *significando* (ou *mantissa*), b é a base do sistema numérico que está sendo usado e e é o expoente.

Antes de ser expresso nessa forma, o número é *normalizado* movendo-se a casa decimal de maneira que apenas um dígito significativo esteja à esquerda da vírgula decimal, o que é feito de modo que a memória do computador não seja desperdiçada no armazenamento de zeros sem utilidade. Por exemplo, um valor como 0,005678 poderia ser representado de forma pouco eficiente (isto é, com desperdício de memória), como $0,005678 \times 10^0$; no entanto, a normalização produziria $5,678 \times 10^{-3}$, que elimina os zeros sem utilidade.

Antes de descrever a implementação da base 2 usada em computadores, vamos explorar as implicações dessa representação em ponto flutuante: por exemplo, quais seriam as consequências do fato de que a mantissa e o expoente devem ser limitados a um número finito de bits a fim de que possam ser armazenados no computador? Como no próximo exemplo, uma boa maneira de fazer isso é no contexto de nosso familiar mundo decimal de base 10.

EXEMPLO 4.2 Implicações da representação em ponto flutuante

Considere que haja um computador de base 10 hipotético com um tamanho de palavra de cinco dígitos e que um dígito é usado para o sinal, dois para o expoente e dois para a mantissa. Para simplificar, assuma que um dos dígitos do expoente é usado para seu sinal, deixando um único dígito para seu valor.

Solução. Uma representação geral do número que segue a normalização seria

$$s_1 d_1.d_2 \times 10^{s_0 d_0}$$

onde s_0 e s_1 são os sinais, d_0 é o valor do expoente e d_1 e d_2 são os valores dos dígitos da mantissa.

Agora vamos “brincar” com esse sistema: primeiro, qual é a maior quantidade positiva que pode ser representada? Claramente, essa quantidade corresponderia a ambos os sinais como positivos e todos os dígitos de valores definidos para o maior valor possível na base 10, isto é, 9:

$$\text{Maior valor} = +9,9 \times 10^{+9}$$

Então, o maior número possível seria um pouco menor que 10 bilhões – que, embora possa parecer um número grande, não é assim tão grande. Por exemplo, esse computador seria incapaz de representar uma constante bastante usada como o número de Avogadro ($6,022 \times 10^{23}$).

Nesse mesmo sentido, o menor número positivo possível seria

$$\text{Menor valor} = +1,0 \times 10^{-9}$$

Novamente, embora esse valor possa parecer bem pequeno, ele não poderia ser utilizado para representar uma quantidade como a constante de Planck ($6,626 \times 10^{-34}$ J·s).

É também possível obter valores negativos similares. As faixas resultantes são exibidas na Figura 4.4: note que os números extensos positivos e negativos fora da faixa causariam um *erro de overflow*, enquanto quantidades muito pequenas gerariam um “buraco” causado pelo *underflow* entre o zero e os menores números – positivo e negativo – representados; nesse caso, quantidades muito pequenas seriam convertidas para zero.

Observe que os limites do intervalo mostrado na Figura 4.4 são basicamente determinados pelo expoente: por exemplo, se a mantissa é acrescida de um dígito, o valor máximo aumenta levemente para $9,99 \times 10^9$, já um aumento de um dígito no expoente eleva o máximo em 90 ordens de grandeza para $9,9 \times 10^{99}$!

No entanto, quando se trata de precisão, a situação se inverte: enquanto a mantissa exerce um papel de menor importância na definição dos limites do intervalo, na definição da precisão ela tem um efeito significativo, o que é ilustrado no presente exemplo em que a mantissa foi limitada a dois dígitos. Como ilustrado na Figura 4.5, assim como há um “buraco” no zero, também há “buracos” entre valores.

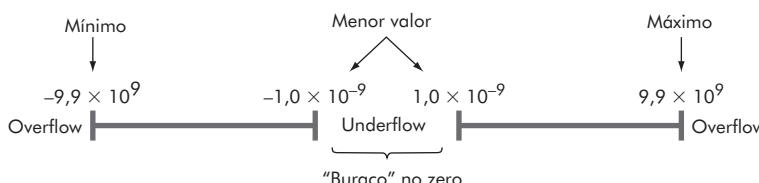


FIGURA 4.4 A linha de números com as faixas possíveis correspondentes ao esquema hipotético de representação em ponto flutuante na base 10, descrito no Exemplo 4.2.

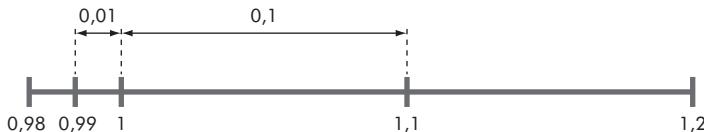


FIGURA 4.5 A menor porção da linha de números correspondente ao esquema hipotético de representação em ponto flutuante na base 10, descrito no Exemplo 4.2. Os números indicam valores que podem ser representados de forma exata. Todas as quantidades que caírem nos “buracos” entre esses valores exibirão algum erro de arredondamento.

Por exemplo, um número racional simples com uma quantidade finita de dígitos como $2^{-5} = 0,03125$ teria de ser armazenado como $3,1 \times 10^{-2}$ ou 0,031, causando um *erro de arredondamento*. Para esse caso, ele representa um erro relativo de

$$\frac{0,03125 - 0,031}{0,03125} = 0,008$$

Enquanto números como 0,03125 podem ser armazenados exatamente expandindo-se os dígitos da mantissa, quantidades com dígitos infinitos devem ser sempre aproximadas. Por exemplo, uma constante geralmente usada como o π ($=3,14159\dots$) teria de ser representada como $3,1 \times 10^0$ ou 3,1, com o erro relativo a seguir:

$$\frac{3,14159 - 3,1}{3,14159} = 0,0132$$

Embora a adição de dígitos à mantissa possa melhorar a aproximação, tais quantidades sempre terão algum erro de arredondamento quando armazenadas em um computador.

Outro efeito mais sutil da representação em ponto flutuante é ilustrado na Figura 4.5. Observe como o intervalo entre números aumenta à medida que se move entre ordens de grandeza: para números com um expoente de -1 (isto é, entre 0,1 e 1), o espaçamento é 0,01; passando para faixa de 1 a 10, o espaçamento aumenta para 0,1. Isso significa não só que o erro de arredondamento de um número será proporcional ao seu módulo, mas também que o erro relativo terá um limite superior. Para este exemplo, o erro relativo máximo seria 0,05, valor chamado de *épsilon da máquina* (ou precisão da máquina).

Como ilustrado no Exemplo 4.2, o fato de tanto o expoente quanto a mantissa serem finitos significa que existem limites de faixa e de precisão na representação em ponto flutuante. Agora, vamos examinar como quantidades em ponto flutuante são realmente representadas em um computador real usando a base 2 ou números binários.

Primeiro, considere a normalização: como números binários consistem exclusivamente de 0s e 1s, há um “bônus” quando eles são normalizados, isto é, o bit à esquerda da vírgula binária será sempre 1. Isso significa que esse bit à esquerda não precisa ser armazenado, portanto, um binário não nulo pode ser expresso em ponto flutuante como

$$\pm(1 + f) \times 2^e$$

onde f é a *mantissa* (isto é, a parte fracionária do significando). Por exemplo, se o número binário 1101,1 fosse normalizado, o resultado seria $1,1011 \times (2)^{-3}$ ou $(1 + 0,1011) \times (2)^{-3}$, assim, apesar de o número original ter cinco bits significativos, é necessário armazenar apenas os quatro bits fracionários: 0,1011.

Por padrão, o MATLAB adotou o *formato de precisão dupla do IEEE*¹, em que oito bytes (64 bits) são usados para representar números em ponto flutuante. Como indicado na Figura 4.6, um bit é reservado para o sinal do número. Do mesmo modo como inteiros são armazenados, o expoente e seu sinal são armazenados em 11 bits, e 52 bits são reservados para a mantissa; no entanto, por conta da normalização, são 53 os bits que podem efetivamente ser armazenados.

Assim como no Exemplo 4.2, isso significa que os números terão uma faixa e uma precisão limitadas, entretanto, como o formato do IEEE usa muito mais bits, o sistema numérico resultante pode ser usado para fins práticos.

Faixa. De forma semelhante à maneira como inteiros são armazenados, os 11 bits utilizados para o expoente resultam em uma faixa de -1022 a 1023. O maior número positivo pode ser representado em binário como

$$\text{Maior valor} = +1,1111\dots1111 \times 2^{+1.023}$$

onde os 52 bits na mantissa são todos 1. Como o significando é aproximadamente 2 (ele é na verdade $2 - 2^{-52}$), o maior valor é, portanto, $2^{1.024} = 1,7977 \times 10^{308}$. De forma similar, o menor número positivo pode ser representado como

$$\text{Menor número} = +1,0000\dots0000 \times 2^{-1.022}$$

que pode ser transformado em um valor na base 10, ou seja, $2^{-1.022} = 2,2251 \times 10^{-308}$.

Precisão. Os 52 bits usados para a mantissa correspondem a 15 ou 16 dígitos na base 10, de forma que a constante π seria expressa como

```
>> format long
>> pi
ans =
3.141592653589793
```

Observe que o épsilon da máquina é $2^{-52} = 2,2204 \times 10^{-16}$.

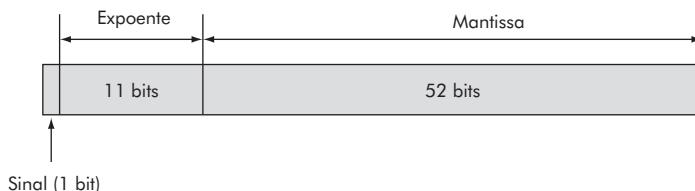


FIGURA 4.6 A maneira como um número em ponto flutuante é armazenado em uma palavra de 8 bytes no formato de precisão dupla do IEEE.

¹ N. de T.: IEEE – Institute of Electrical and Electronics Engineers ou Instituto de Engenheiros Eletricistas e Eletrônicos.

O MATLAB tem uma série de funções nativas relacionadas à sua representação numérica interna: a função *realmax*, por exemplo, exibe o maior número real positivo:

```
>> format long
>> realmax
ans =
1.797693134862316e+308
```

Os números resultantes de cálculos que excederem esse valor, definidos no MATLAB como infinito (*inf*), criam um *overflow*. A função *realmin* exibe o menor número real positivo:

```
>> realmin
ans =
2.225073858507201e-308
```

Os números menores que esse valor criam um *underflow* e, no MATLAB, são definidos como zero. Finalmente, a função *eps* exibe o épsilon da máquina:

```
>> eps
ans =
2.220446049250313e-016
```

4.2.2 Manipulação aritmética dos números no computador

Assim como as limitações de um sistema numérico no computador, as manipulações aritméticas reais envolvendo esses números também podem resultar em erros de arredondamento. Para entender como isso ocorre, vamos examinar como o computador realiza adição e subtração simples.

Para ilustrar os efeitos dos erros de arredondamento em adição e subtração simples será usado um sistema numérico na base 10 normalizado, por conta de sua familiaridade – ainda que outras bases numéricas pudessem se comportar de forma similar. Para simplificar a discussão, será empregado um computador decimal hipotético com uma mantissa de quatro dígitos e um expoente de um dígito.

Quando dois números em ponto flutuante são somados, a mantissa com o número de menor expoente é modificada de forma que os expoentes sejam os mesmos, o que tem o efeito de alinhar as vírgulas decimais. Por exemplo, suponha que você queira somar $1,557 + 0,04341$; o computador expressaria os números como $0,1557 \times 10^1 + 0,004341 \times 10^1$ e as mantissas seriam somadas para fornecer $0,160041 \times 10^1$. Agora, como esse computador hipotético apenas carrega uma mantissa de quatro dígitos, o número excedente de dígitos é cortado e o resultado é $0,1600 \times 10^1$. Observe como os dois últimos algarismos do segundo número (41) que foram deslocados para a direita foram essencialmente perdidos nos cálculos.

A subtração é feita de forma idêntica à soma, exceto pelo fato de que o sinal do subtraendo é trocado. Considere que estejamos subtraindo 26,86 de 36,41, isto é,

$$\begin{array}{r} 0,3641 \times 10^2 \\ - 0,2686 \times 10^2 \\ \hline 0,0955 \times 10^2 \end{array}$$

Para esse caso, o resultado não está normalizado e é necessário deslocar a vírgula uma casa para a direita a fim de fornecer $0,9550 \times 10^1 = 9,550$. Observe que o

zero adicionado no final da mantissa não é significativo, mas apenas um apêndice para preencher um espaço vazio criado pelo deslocamento. Caso os números fossem muito próximos, seriam produzidos resultados ainda mais drásticos, como

$$\begin{array}{r} 0,7642 \times 10^3 \\ - 0,7641 \times 10^3 \\ \hline 0,0001 \times 10^3 \end{array}$$

que deveria ser convertido para $0,1000 \times 10^0 = 0,1000$. Logo, nesse caso, três zeros não significativos são adicionados.

A subtração de dois números praticamente iguais é chamada de *cancelamento na subtração*, termo que se refere ao erro de arredondamento induzido quando se subtraem dois números quase iguais em ponto flutuante. Ele é um exemplo clássico de que a forma como os computadores manipulam um problema matemático pode levar a problemas numéricos. Entre outros cálculos que podem causar problemas estão:

Cálculos grandes. Para chegar ao resultado final, certos métodos necessitam de um grande número de cálculos aritméticos, geralmente interdependentes – isto é, os últimos cálculos dependem do resultado dos primeiros. Assim, mesmo que os erros de arredondamento individuais sejam pequenos, o efeito cumulativo sobre um grande número de cálculos pode ser significativo. Um caso muito simples envolve a soma de um número redondo na base 10 que não é redondo na base 2. Suponha que a seguinte função em MATLAB seja implementada:

```
function soma = soma_demo()
s = 0
for i = 1:10000
    s = s + 0.0001;
end
soma = s;
```

Quando essa função é executada o resultado é

```
>> format long
>> soma_demo
ans =
0.9999999999999906
```

O comando `format long` exibe o resultado numérico na representação de 15 dígitos significativos utilizada pelo MATLAB. Talvez você esperasse que a soma fosse igual a 1 (e realmente deveria ser!); no entanto, embora 0,0001 seja um número redondo satisfatório na base 10, ele não pode ser expresso exatamente na base 2, de forma que a soma resulta em um número ligeiramente diferente de 1. Deve-se enfatizar que o MATLAB possui funcionalidades projetadas para minimizar esses erros. Por exemplo, suponha que você defina um vetor como em

```
>> format long
>> s = [0:0.0001:1];
```

Para esse caso, em vez de ser igual a 0.9999999999999906, o último elemento do vetor será exatamente igual a 1, como pode ser verificado em:

```
>> s(10001)
ans =
1
```

Adicionando um número grande e um número pequeno. Suponha que seja somado um número pequeno, 0,0010, a um número grande, 4000, usando-se um computador hipotético com quatro algarismos na mantissa e um algarismo no expoente. O número pequeno é modificado de forma que o seu expoente coincida com o grande,

$$\begin{array}{r} 0,4000 \times 10^4 \\ 0,0000001 \times 10^4 \\ \hline 0,4000001 \times 10^4 \end{array}$$

que é truncado para $0,4000 \times 10^4$ – portanto, a soma poderia muito bem não ter sido feita! Esse tipo de erro pode ocorrer em cálculos de uma série infinita. Os termos iniciais de tais séries são em geral relativamente grandes se comparados aos termos posteriores, assim, depois que uns poucos termos foram somados, somam-se quantidades muito pequenas a uma quantidade grande. Uma forma de atenuar esse tipo de erro é somar as séries, em uma ordem crescente em vez de decrescente; dessa forma, cada novo termo será de valor absoluto comparável à soma acumulada.

Borrão. O borrão ocorre sempre que termos individuais em uma soma são maiores que a soma propriamente dita, como em uma série com sinais mistos.

Produtos internos. Pelo que foi visto até aqui, deve ficar claro que algumas séries infinitas são mais suscetíveis a erros de arredondamento. Felizmente o cálculo de séries não é uma operação tão comum quanto o cálculo do produto escalar, como em

$$\sum_{i=1}^n x_i y_i = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$$

Essa operação é muito usada, em especial na solução de equações algébricas lineares simultâneas. Tais somas são suscetíveis a erros de arredondamento, assim em geral é melhor calculá-las com precisão dupla, como é feito automaticamente no MATLAB.

4.3 ERROS DE TRUNCAMENTO

Os *erros de truncamento* resultam do uso de uma aproximação no lugar de um procedimento matemático exato. Por exemplo, no Capítulo 1 aproximamos a derivada da velocidade de um saltador de *bungee jumping* em queda livre por uma equação de diferenças divididas da forma [Equação (1.11)]

$$\frac{dv}{dt} \cong \frac{\Delta v}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} \quad (4.8)$$

Um erro de truncamento foi introduzido na solução numérica porque a equação de diferenças apenas aproxima o valor verdadeiro da derivada (lembre-se da Figura 1.3). Para melhor compreender tais erros, será mostrada agora uma formulação matemática amplamente usada nos métodos numéricos para expressar uma função de forma aproximada – a série de Taylor.

4.3.1 A série de Taylor

O teorema de Taylor e sua fórmula associada, a série de Taylor, são de grande valia no estudo dos métodos numéricos. O *teorema de Taylor* afirma que qualquer função suave pode ser aproximada por um polinômio, então a *série de Taylor* fornece um meio para expressar essa ideia matematicamente de forma que gere resultados práticos.

Uma maneira útil de compreender a série de Taylor é construí-la termo a termo, como ao prever um valor de uma função em um ponto em termos do valor da função e suas derivadas em outro ponto.

Considere que você esteja com os olhos vendados e seja levado para um local ao lado de uma montanha de frente para um declive (Figura 4.7); vamos chamar sua localização horizontal de x_i e sua distância vertical com relação à base da montanha de $f(x_i)$. Você tem a tarefa de prever a altura em uma posição x_{i+1} , que está a uma distância h de você.

Você é colocado em uma plataforma completamente horizontal, de modo que não tem ideia de que a montanha está inclinada para baixo e longe de você. Nesse ponto, qual seria seu melhor palpite da altura em x_{i+1} ? Se você pensar bem, (lembre que você não tem ideia alguma do que está à sua frente), o melhor palpite seria a mesma altura de onde você está agora! Você poderia expressar essa previsão matematicamente como

$$f(x_{i+1}) \cong f(x_i) \quad (4.9)$$

Essa relação, chamada de *aproximação de ordem zero*, indica que o valor de f no novo ponto é o mesmo que seu valor no antigo ponto. Esse resultado faz sentido do ponto de vista intuitivo, pois se x_i e x_{i+1} estiverem perto um do outro, é provável que o novo valor seja parecido com o anterior.

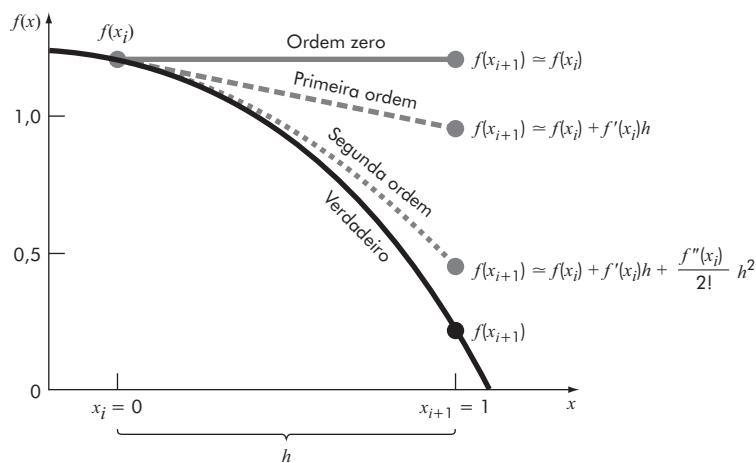


FIGURA 4.7 A aproximação de $f(x) = -0,1x^4 - 0,15x^3 - 0,5x^2 - 0,25x + 1,2$ em $x = 1$ por expansões em série de Taylor de ordem zero, de primeira ordem e de segunda ordem.

A Equação (4.9) fornece uma estimativa perfeita se a função que estiver sendo aproximada for de fato constante. Entretanto, se a função variar de alguma forma no intervalo, termos adicionais da série de Taylor serão necessários para fornecer uma estimativa melhor. No nosso problema, a estimativa perfeita seria fornecida apenas se você estivesse em um platô completamente plano.

Você agora pode sair da plataforma e ficar sobre a superfície da montanha com uma perna posicionada à sua frente e a outra atrás: imediatamente, você sente que seu pé à frente está em um nível mais baixo do que seu pé atrás. De fato, você pode obter uma estimativa quantitativa da inclinação medindo a diferença na elevação e dividindo-a pela distância entre seus pés.

Com essa informação adicional, você tem como prever melhor a altura em $f(x_{i+1})$: basta usar a inclinação estimada para projetar uma linha reta saindo de x_i . Essa predição pode ser expressa matematicamente por

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)h \quad (4.10)$$

Essa relação é chamada *aproximação de primeira ordem*, porque o termo adicional de primeira ordem consiste em uma inclinação $f'(x_i)$ multiplicada por h , a distância entre x_i e x_{i+1} . Assim, a expressão está agora na forma de uma linha reta capaz de prever um aumento ou diminuição da função entre x_i e x_{i+1} .

Embora a Equação (4.10) possa prever uma variação, ela é exata apenas para retas ou tendências *lineares*, assim, para obter uma previsão melhor, precisamos adicionar mais termos à nossa equação. Então, agora é permitido permanecer sobre a montanha e tirar duas medidas: primeiro, meça a inclinação atrás de você mantendo um pé fixo em x_i e movendo o outro para trás uma distância Δx – vamos chamar essa inclinação de $f'_b(x_i)$. Depois, você mede a inclinação à sua frente mantendo um pé fixo em x_i e movendo o outro para frente uma distância Δx – vamos chamar essa inclinação de $f'_f(x_i)$. Imediatamente você percebe que a inclinação atrás é mais suave que a inclinação à sua frente, o que mostra claramente que a queda em altura está “acelerando” para baixo à sua frente. Assim, as chances são de que $f(x_i)$ seja ainda mais baixo que sua previsão linear anterior.

Como você deve estar pensando, vamos adicionar um termo de segunda ordem à nossa equação e transformá-la em uma parábola, com a série de Taylor

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 \quad (4.11)$$

Para utilizar essa fórmula, é necessária uma estimativa da segunda derivada. As duas últimas inclinações determinadas podem ser usadas para estimá-la como

$$f''(x_{i+1}) \cong \frac{f'_f(x_i) - f'_b(x_i)}{\Delta x} \quad (4.12)$$

Assim, a segunda derivada é meramente uma derivada de uma derivada; nesse caso, a taxa de variação da inclinação.

Antes de continuar, analise cuidadosamente a Equação (4.11): observe que todos os valores com subscrito i representam valores estimados; isto é, eles são números. Consequentemente, as únicas incógnitas são os valores na posição de previsão x_{i+1} , o que torna esta uma equação quadrática da forma

$$f(h) \cong a_2h^2 + a_1h + a_0$$

Portanto, pode-se perceber que a série de Taylor de segunda ordem aproxima a função com um polinômio de segunda ordem.

Poderíamos continuar adicionando mais derivadas para capturar mais informações da curvatura da função, chegando à expansão completa em série de Taylor

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \dots + \frac{f^{(n)}(x_i)}{n!}h^n + R_n \quad (4.13)$$

Observe que, como a Equação (4.13) é uma série infinita, o sinal de igual substitui o sinal de aproximação usado nas Equações (4.9) a (4.11). O resto é incluído para representar todos os termos de $n + 1$ até infinito,

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}h^{n+1} \quad (4.14)$$

onde o subscrito n indica que é o resto para a aproximação de ordem n e ξ é um valor de x localizado em algum ponto entre x_i e x_{i+1} .

Agora está claro porque o teorema de Taylor estabelece que qualquer função suave pode ser aproximada por um polinômio e como a série de Taylor expressa essa ideia matematicamente.

Em geral, a expansão em série de Taylor de ordem n será exata para um polinômio de grau n . Para outras funções diferenciáveis (portanto, contínuas), como exponenciais e funções senoidais, um número finito de termos não resultará em uma estimativa exata. Cada termo adicional irá contribuir com alguma melhora, ainda que pequena, para a aproximação, comportamento que será ilustrado no Exemplo 4.3. A série fornecerá um valor exato apenas se um número finito de termos for somado.

Embora o que foi dito seja verdadeiro, o valor prático da expansão em série de Taylor é que, na maioria dos casos, a inclusão de apenas poucos termos resultará em uma aproximação que estará suficientemente perto do valor verdadeiro para fins práticos. A avaliação de quantos termos são necessários para ficar “suficientemente próximo” é baseada no resto da expansão; lembre-se de que o resto tem a forma geral da Equação (4.14). Essa relação tem dois grandes defeitos: primeiro, ξ não é conhecido de fato, mas está em algum lugar entre x_i e x_{i+1} ; segundo, para se calcular a Equação (4.14) é preciso determinar a $(n+1)$ -ésima derivada de $f(x)$, ou seja, é necessário conhecer $f'(x)$. Entretanto, se $f(x)$ fosse conhecida, não seria preciso fazer a expansão em série de Taylor no contexto atual!

Apesar desse dilema, a Equação (4.14) ainda é útil para compreender os erros de truncamento, porque, *de fato*, há controle sobre o termo h na equação. Em outras palavras, é possível escolher quão longe de x se quer calcular $f(x)$ e controlar o número de termos incluídos na expansão. Consequentemente, a Equação (4.14) em geral é expressa como

$$R_n = O(h^{n+1})$$

onde a nomenclatura $O(h^{n+1})$ significa que o erro de truncamento é da ordem de h^{n+1} , ou seja, o erro é proporcional ao tamanho do passo h elevado à $(n+1)$ -ésima potência. Embora essa aproximação não tenha nenhuma implicação no valor da derivada que multiplica h^{n+1} , ela é extremamente útil para analisar o erro comparativo de métodos numéricos com base em expansões em série de Taylor. Por exemplo: se o

erro for $O(h)$, dividir o tamanho do passo por dois fará com que o erro seja dividido por dois; contudo, se o erro for $O(h^2)$, dividir o tamanho do passo por dois fará com que o erro seja dividido por quatro.

Em geral, pode-se supor que o erro de truncamento diminui com a adição de termos na série de Taylor. Em muitos casos, se h for suficientemente pequeno, o primeiro e alguns outros termos de baixa ordem geralmente darão conta de uma porcentagem desproporcionalmente alta dos erros, assim, apenas poucos termos serão necessários para se obter uma estimativa adequada. Essa propriedade é ilustrada no exemplo a seguir.

EXEMPLO 4.3 Aproximação de uma função com uma expansão em série de Taylor

Use expansões em série de Taylor com $n = 0$ até 6 para aproximar $f(x) = \cos x$ em $x_i = \pi/3$ com base no valor de $f(x)$ e de suas derivadas em $x_i = \pi/4$. Observe que isso significa que $h = \pi/3 - \pi/4 = \pi/12$.

Solução. O conhecimento da função verdadeira permite determinar o valor correto de $f(\pi/3) = 0,5$. A aproximação de ordem zero é [Equação (4.9)]

$$f\left(\frac{\pi}{3}\right) \cong \cos\left(\frac{\pi}{4}\right) = 0,707106781$$

o que representa um erro percentual relativo de

$$\varepsilon_t = \left| \frac{0,5 - 0,707106781}{0,5} \right| 100\% = 41,4\%$$

Para a aproximação de primeira ordem, adiciona-se o termo da primeira derivada, $f'(x) = -\sin x$:

$$f\left(\frac{\pi}{3}\right) \cong \cos\left(\frac{\pi}{4}\right) - \sin\left(\frac{\pi}{4}\right)\left(\frac{\pi}{12}\right) = 0,521986659$$

que tem $|\varepsilon_t| = 4,40\%$. Para a aproximação de segunda ordem, adiciona-se o termo da segunda derivada, $f''(x) = -\cos x$:

$$f\left(\frac{\pi}{3}\right) \cong \cos\left(\frac{\pi}{4}\right) - \sin\left(\frac{\pi}{4}\right)\left(\frac{\pi}{12}\right) - \frac{\cos(\pi/4)}{2}\left(\frac{\pi}{12}\right)^2 = 0,497754491$$

com $|\varepsilon_t| = 0,449\%$ – logo, a inclusão de um termo adicional melhorou a estimativa. Pode-se continuar o processo e listar os resultados, como na tabela a seguir:

Ordem n	$f^{(n)}(x)$	$f(\pi/3)$	$ \varepsilon_t $
0	$\cos x$	0,707106781	41,4
1	$-\sin x$	0,521986659	4,40
2	$-\cos x$	0,497754491	0,449
3	$\sin x$	0,499869147	$2,62 \times 10^{-2}$
4	$\cos x$	0,500007551	$1,51 \times 10^{-3}$
5	$-\sin x$	0,500000304	$6,08 \times 10^{-5}$
6	$-\cos x$	0,499999988	$2,44 \times 10^{-6}$

Observe que as derivadas nunca se anulam, como seria o caso para um polinômio, portanto, cada termo adicional melhora a estimativa. Entretanto, observe também que a maior parte da melhora acontece nos termos iniciais, assim, quando se tiver somado o terceiro termo, o erro terá sido reduzido a 0,026%, o que significa que foram obtidos 99,974% do valor verdadeiro. Embora a adição de mais termos reduza mais ainda o erro, a melhora se torna desprezível.

4.3.2 O resto na expansão em série de Taylor

Antes de ilustrar como a série de Taylor é usada para estimar erros numéricicos, é preciso explicar porque o argumento ξ foi incluído na Equação (4.14). Faremos isso de forma simples, com base em uma interpretação um pouco mais visual.

Considere que a expansão em série de Taylor [Equação (4.13)] seja truncada depois do termo de ordem zero para se obter

$$f(x_{i+1}) \equiv f(x_i)$$

Uma descrição visual dessa previsão de ordem zero é mostrada na Figura 4.8: o resto – ou erro – dessa previsão, também apresentado na figura, consiste na série infinita dos termos truncados:

$$R_0 = f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f^{(3)}(x_i)}{3!}h^3 + \dots$$

É óbvia a inconveniência de lidar com o resto nessa forma de série infinita, portanto, uma simplificação seria truncar o próprio resto, como em

$$R_0 \equiv f'(x_i)h \quad (4.15)$$

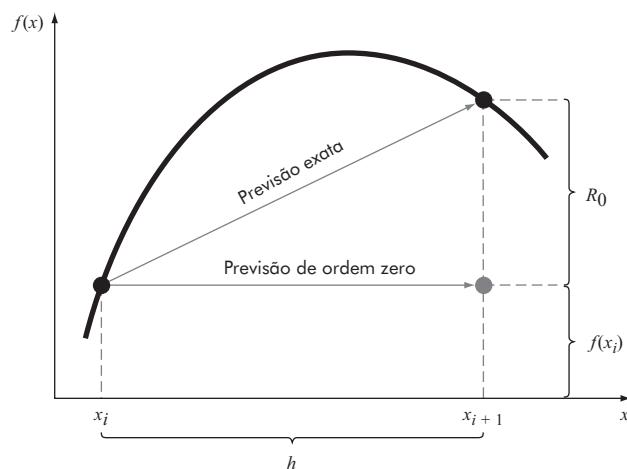


FIGURA 4.8 Descrição gráfica da previsão de ordem zero da série de Taylor e do resto.

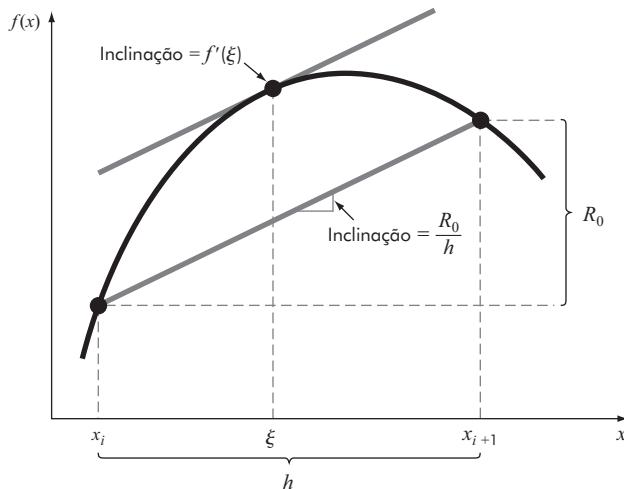


FIGURA 4.9 Descrição gráfica do teorema do valor médio para as derivadas.

Embora, como afirmado na seção anterior, as derivadas de ordem mais baixa sejam em geral responsáveis por uma parte maior do resto do que os termos de ordem mais alta, esse resultado ainda não é exato por causa dos termos de segunda ordem e de ordens mais altas desprezados. Essa “inexatidão” é representada pelo símbolo de igualdade aproximada (\equiv) usado na Equação (4.15).

Vejamos uma simplificação alternativa, que transforma a aproximação em uma equivalência com base em uma observação gráfica. Como na Figura 4.9, o *teorema do valor médio para derivadas* afirma que, se uma função $f(x)$ e sua primeira derivada forem contínuas em um intervalo de x_i a x_{i+1} , então pelo menos um ponto na função tem uma inclinação, denotada por $f'(\xi)$, paralela à reta que liga $f(x_i)$ e $f(x_{i+1})$; o parâmetro ξ designa o valor de x onde essa inclinação ocorre (Figura 4.9). Uma ilustração física desse teorema é que, se você viajar entre dois pontos com uma dada velocidade média, existirá pelo menos um instante durante o percurso no qual você estará se movendo com a velocidade média.

Usando esse teorema, é simples perceber que, como ilustrado pela Figura 4.9, a inclinação $f'(\xi)$ é igual à elevação R_0 dividida pelo passo h , ou

$$f'(\xi) = \frac{R_0}{h}$$

que pode ser reorganizado para fornecer

$$R_0 = f'(\xi) h \quad (4.16)$$

Assim, deduz-se a versão para ordem zero da Equação (4.14). As versões em ordem superior são apenas uma extensão lógica do raciocínio usado para deduzir a Equação (4.16); já a versão de primeira ordem é

$$R_1 = \frac{f''(\xi)}{2!} h^2 \quad (4.17)$$

Nesse caso, o valor de ξ está de acordo com o valor de x correspondente à segunda derivada que torna a Equação (4.17) exata. Versões similares de ordem superior podem ser deduzidas a partir da Equação (4.14).

4.3.3 Usando a série de Taylor para fazer uma estimativa do erro de truncamento

Embora a série de Taylor seja extremamente útil na estimativa dos erros de truncamento em todo este livro, talvez não fique claro como a expansão pode realmente ser aplicada aos métodos numéricos. De fato, já fizemos isso no exemplo do saltador de *bungee jumping* em queda livre: lembre que o objetivo dos Exemplos 1.1 e 1.2 era prever a velocidade como uma função do tempo, isto é, o interesse era determinar $v(t)$. Como especificado pela Equação (4.13), $v(t)$ pode ser expandida em uma série de Taylor:

$$v(t_{i+1}) = v(t_i) + v'(t_i)(t_{i+1} - t_i) + \frac{v''(t_i)}{2!}(t_{i+1} - t_i)^2 + \dots + R_n$$

Agora, trunca-se a série depois do termo da primeira derivada

$$v(t_{i+1}) = v(t_i) + v'(t_i)(t_{i+1} - t_i) + R_1 \quad (4.18)$$

A Equação (4.18) pode ser reescrita como

$$v'(t_i) = \underbrace{\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i}}_{\substack{\text{Aproximação} \\ \text{de primeira ordem}}} - \underbrace{\frac{R_1}{t_{i+1} - t_i}}_{\substack{\text{Erro de} \\ \text{truncamento}}} \quad (4.19)$$

A primeira parte da Equação (4.19) é exatamente a mesma relação que foi usada para aproximar a derivada no Exemplo 1.2 [Equação (1.11)], entretanto, com a série de Taylor, obteve-se agora uma estimativa para o erro de truncamento associado a essa aproximação da derivada. Usando as Equações (4.14) e (4.19), obtém-se

$$\frac{R_1}{t_{i+1} - t_i} = \frac{v''(\xi)}{2!}(t_{i+1} - t_i)$$

ou

$$\frac{R_1}{t_{i+1} - t_i} = O(t_{i+1} - t_i)$$

Assim, a estimativa da derivada [Equação (1.11)] ou a primeira parte da Equação (4.19) tem um erro de truncamento da ordem de $t_{i+1} - t_i$, ou seja, o erro na aproximação da derivada deveria ser proporcional ao tamanho do passo. Consequentemente, se o tamanho do passo for dividido por dois, deve-se esperar que o erro na derivada também seja dividido por dois.

4.3.4 Derivação numérica

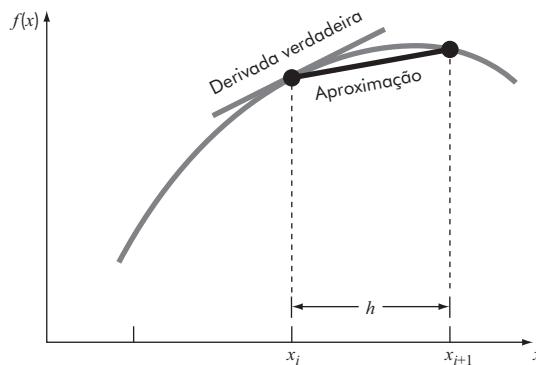
A Equação (4.19) tem uma designação formal nos métodos numéricos – ela é chamada *diferença dividida finita* e pode ser representada em geral por

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} + O(x_{i+1} - x_i) \quad (4.20)$$

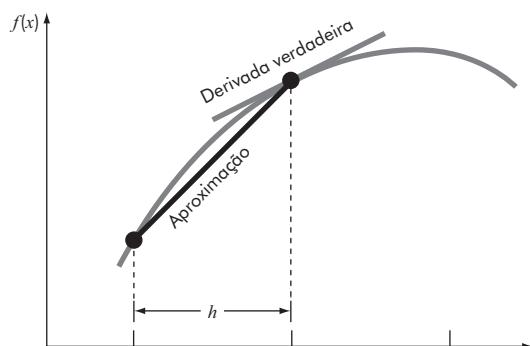
ou

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \quad (4.21)$$

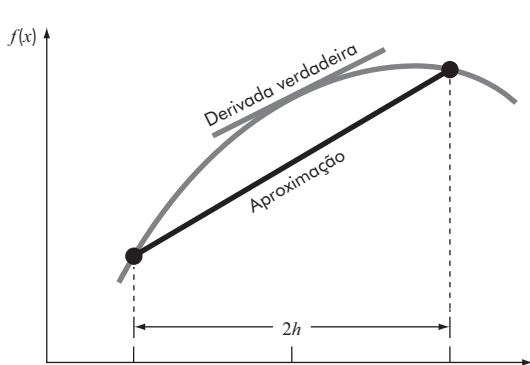
onde h é chamado de tamanho do passo, isto é, o comprimento do intervalo no qual a aproximação é feita, $x_{i+1} - x_i$. Ela é chamada de diferença “progressiva” porque utiliza os dados em i e $i + 1$ para estimar a derivada (Figura 4.10a).



(a)



(b)



(c)

FIGURA 4.10 Descrição gráfica das aproximações da primeira derivada por diferenças divididas finitas (a) progressiva, (b) regressiva e (c) centrada.

Essa diferença dividida progressiva é apenas uma das muitas que podem ser deduzidas a partir da série de Taylor para aproximar derivadas numericamente, por exemplo, aproximações com diferenças regressivas e centradas da primeira derivada podem ser deduzidas de forma similar à dedução da Equação (4.19). A primeira utiliza valores em x_{i-1} e x_i (Figura 4.10b), enquanto a última utiliza valores igualmente espaçados em torno do ponto no qual a derivada será estimada (Figura 4.10c). Aproximações mais acuradas da primeira derivada podem ser deduzidas incluindo-se termos de ordem mais alta na série de Taylor – na verdade todas essas versões também podem ser deduzidas para derivadas de segunda e terceira ordens ou superiores, conforme mostram os tópicos a seguir.

Aproximação da primeira derivada por diferença regressiva. A série de Taylor pode ser expandida regressivamente para calcular um valor anterior com base no valor atual, como em

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \dots \quad (4.22)$$

Truncando essa equação depois da primeira derivada e a reorganizando, obtém-se

$$f'(x_i) \cong \frac{f(x_i) - f(x_{i-1})}{h} \quad (4.23)$$

onde o erro é $O(h)$.

Aproximação da primeira derivada por diferença centrada. Uma terceira forma de aproximar a primeira derivada é subtrair a Equação (4.22) da expansão em série de Taylor progressiva:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \dots \quad (4.24)$$

para obter

$$f(x_{i+1}) = f(x_{i-1}) + 2f'(x_i)h + 2\frac{f^{(3)}(x_i)}{3!}h^3 + \dots$$

que pode ser reescrita como

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{f^{(3)}(x_i)}{6}h^2 + \dots$$

ou

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - O(h^2) \quad (4.25)$$

A Equação (4.25) é uma representação da primeira derivada como *diferença centrada*: observe que o erro de truncamento é da ordem de h^2 , em oposição às aproximações progressiva e regressiva, que eram da ordem de h . Consequentemente, pela análise da série de Taylor é possível inferir que a diferença centrada é uma representação mais acurada da derivada (Figura 4.10c). Por exemplo, se o tamanho do passo fosse dividido por dois usando-se uma diferença progressiva ou regressiva, o erro de truncamento seria aproximadamente dividido por dois, enquanto para a diferença centrada o erro seria dividido por quatro.

EXEMPLO 4.4 Aproximação de derivadas por diferenças divididas finitas

Use aproximações por diferenças progressiva e regressiva de $O(h)$ e uma aproximação por diferença centrada $O(h^2)$ para fazer uma estimativa da primeira derivada de

$$f(x) = -0,1x^4 - 0,15x^3 - 0,5x^2 - 0,25x + 1,2$$

em $x = 0,5$ usando um tamanho de passo $h = 0,5$. Repita os cálculos usando $h = 0,25$. Observe que a derivada pode ser calculada diretamente por

$$f'(x) = -0,4x^3 - 0,45x^2 - 1,0x - 0,25$$

e pode ser usada para calcular o valor verdadeiro como sendo $f'(0,5) = -0,9125$.

Solução. Para $h = 0,5$, a função pode ser empregada para determinar

$$\begin{aligned} x_{i-1} &= 0 & f(x_{i-1}) &= 1,2 \\ x_i &= 0,5 & f(x_i) &= 0,925 \\ x_{i+1} &= 1,0 & f(x_{i+1}) &= 0,2 \end{aligned}$$

Com esses valores pode-se calcular a diferença dividida progressiva [Equação (4.21)],

$$f'(0,5) \cong \frac{0,2 - 0,925}{0,5} = -1,45 \quad |\varepsilon_t| = 58,9\%$$

a diferença dividida regressiva [Equação (4.23)],

$$f'(0,5) \cong \frac{0,925 - 1,2}{0,5} = -0,55 \quad |\varepsilon_t| = 39,7\%$$

e a diferença dividida centrada [Equação (4.25)]

$$f'(0,5) \cong \frac{0,2 - 1,2}{1,0} = -1,0 \quad |\varepsilon_t| = 9,6\%$$

Para $h = 0,25$,

$$\begin{aligned} x_{i-1} &= 0,25 & f(x_{i-1}) &= 1,10351563 \\ x_i &= 0,5 & f(x_i) &= 0,925 \\ x_{i+1} &= 0,75 & f(x_{i+1}) &= 0,63632813 \end{aligned}$$

que podem ser usadas para calcular a diferença dividida progressiva,

$$f'(0,5) \cong \frac{0,63632813 - 0,925}{0,25} = -1,155 \quad |\varepsilon_t| = 26,5\%$$

a diferença dividida regressiva,

$$f'(0,5) \cong \frac{0,925 - 1,10351563}{0,25} = -0,714 \quad |\varepsilon_t| = 21,7\%$$

e a diferença dividida centrada

$$f'(0,5) \cong \frac{0,63632813 - 1,10351563}{0,5} = -0,934 \quad |\varepsilon_t| = 2,4\%$$

Para ambos os tamanhos de passo, a aproximação por diferença centrada é mais acurada que as aproximações progressiva e regressiva. Além disso, como previsto pelo teorema de Taylor, dividir o tamanho do passo por dois divide o erro aproximadamente por dois nas diferenças progressiva e regressiva e por quatro na diferença centrada.

Aproximações por diferenças finitas de derivadas superiores. Além das primeiras derivadas, a expansão em série pode ser usada para deduzir estimativas numéricas das derivadas mais altas. Para fazer isso, escreve-se uma expansão em série de Taylor progressiva para $f(x_{i+2})$ em termos de $f(x_i)$:

$$f(x_{i+2}) = f(x_i) + f'(x_i)(2h) + \frac{f''(x_i)}{2!}(2h)^2 + \dots \quad (4.26)$$

A Equação (4.24) pode ser multiplicada por 2 e subtraída da Equação (4.23) para fornecer

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

que pode ser reescrita como

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h) \quad (4.27)$$

Essa relação, chamada *segunda diferença dividida finita progressiva*, possui manipulações similares que podem ser usadas para deduzir uma versão regressiva:

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

Uma aproximação para a segunda derivada por diferença dividida centrada pode ser obtida somando-se as Equações (4.22) e (4.24) e reorganizando o resultado para fornecer

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

Como nas aproximações da primeira derivada, o caso centrado é mais acurado. Observe que a versão centrada pode ser expressa também por

$$f''(x_i) \cong \frac{\frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f(x_i) - f(x_{i-1})}{h}}{h}$$

Assim, da mesma forma como a derivada segundas é uma derivada da derivada, a aproximação pela segunda diferença dividida é a diferença de duas diferenças divididas [lembre-se da Equação (4.12)].

4.4 ERRO NUMÉRICO TOTAL

O *erro numérico total* é a soma dos erros de truncamento e de arredondamento. Em geral, a única maneira de minimizar os erros de arredondamento é acrescentar algoritmos significativos ao computador; além disso, observa-se que tais erros podem *aumentar* por causa de cancelamentos na subtração ou de um aumento do número de cálculos na análise. O Exemplo 4.4, entretanto, mostrou que o erro de truncamento pode ser reduzido com a diminuição do passo; como esse tipo de ação pode levar a um cancelamento na subtração ou a um aumento nos cálculos, os erros de truncamento *diminuirão*, enquanto os erros de arredondamento *aumentarão*.

Encara-se então o seguinte dilema: a estratégia para diminuir uma componente do erro total leva a um aumento da outra componente. É perfeitamente possível diminuir o tamanho do passo para minimizar os erros de truncamento, mas, ao fazer isso, os erros de arredondamento começam a dominar a solução e o erro total cresce (Figura 4.11). Um desafio é determinar um tamanho de passo apropriado para determinado cálculo; o ideal seria escolher um tamanho de passo maior para diminuir o número de operações e os erros de arredondamento sem pagar o preço de grandes erros de truncamento. Se o erro total for como mostra a Figura 4.11, o desafio é identificar o ponto de retorno diminuído onde os erros de arredondamento começam a neutralizar as vantagens da redução do tamanho do passo.

Pelo fato de o MATLAB utilizar de 15 a 16 dígitos de precisão, tais situações são relativamente incomuns, mas podem ocorrer algumas vezes e sugerem um tipo de “princípio da incerteza numérico” que coloca um limite absoluto na acurácia obtida com o uso de um certo método numérico computadorizado. Um caso desse tipo é explorado na seção a seguir.

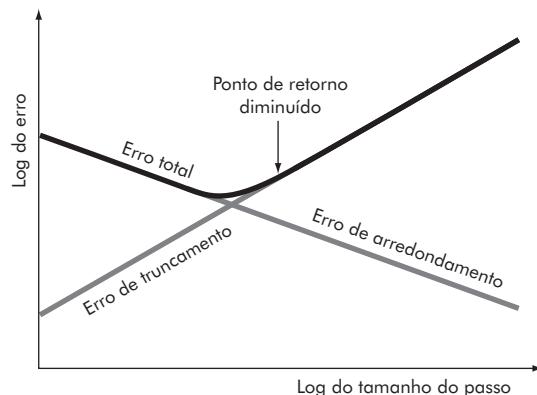


FIGURA 4.11 Uma descrição gráfica do balanço entre erros de arredondamento e de truncamento que ocorrem algumas vezes na utilização de um método numérico. É mostrado o ponto de retorno diminuído, onde o erro de arredondamento começa a neutralizar as vantagens da redução do tamanho do passo.

4.4.1 Análise de erro de derivação numérica

Como descrito na Seção 4.3.4, uma aproximação por diferença centrada para a primeira derivada pode ser escrita como (Equação 4.25)

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{f^{(3)}(\xi)}{6}h^2$$

Valor verdadeiro	Aproximação por diferença finita	Erro de arredondamento	Erro de truncamento
---------------------	--	---------------------------	------------------------

(4.28)

Assim, se os dois valores da função no numerador da aproximação por diferença finita não têm erro de arredondamento, o único erro se deve ao truncamento. Entretanto, como estamos usando computadores digitais, os valores da função incluem um erro de arredondamento, como em

$$\begin{aligned} f(x_{i-1}) &= \tilde{f}(x_{i-1}) + e_{i-1} \\ f(x_{i+1}) &= \tilde{f}(x_{i+1}) + e_{i+1} \end{aligned}$$

onde os \tilde{f} 's são os valores arredondados da função e os e 's são os erros de arredondamento associados. A substituição desses valores na Equação (4.28) fornece

$$f'(x_i) = \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} + \frac{e_{i+1} - e_{i-1}}{2h} - \frac{f^{(3)}(\xi)}{6}h^2$$

Valor verdadeiro	Aproximação por diferença finita	Erro de arredondamento	Erro de truncamento
---------------------	--	---------------------------	------------------------

Pode-se perceber que o erro total da aproximação por diferença finita consiste em um erro de arredondamento que diminui com o tamanho do passo e um erro de truncamento que aumenta com o tamanho do passo.

Assumindo que o valor absoluto de cada componente do erro de arredondamento tenha um limite superior de ε , o valor máximo possível da diferença $e_{i+1} - e_{i-1}$ será 2ε . Considere também que a terceira derivada tenha um valor absoluto máximo de M . Um limite superior do valor absoluto do erro total pode, portanto, ser representado como

$$\text{Erro total} = \left| f'(x_i) - \frac{\tilde{f}(x_{i+1}) - \tilde{f}(x_{i-1})}{2h} \right| \leq \frac{\varepsilon}{h} + \frac{h^2 M}{6} \quad (4.29)$$

Um tamanho de passo ótimo pode ser determinado derivando-se a Equação (4.29), igualando o resultado a zero e resolvendo para

$$h_{optm} = \sqrt[3]{\frac{3\varepsilon}{M}} \quad (4.30)$$

EXEMPLO 4.5 Erros de arredondamento e truncamento na derivação numérica

No Exemplo 4.4 foi utilizada uma aproximação por diferença centrada de $O(h^2)$ para estimar a primeira derivada da função a seguir em $x = 0,5$,

$$f(x) = -0,1x^4 - 0,15x^3 - 0,5x^2 - 0,25x + 1,2$$

Efetue o mesmo cálculo iniciando com $h = 1$ e divida progressivamente o tamanho do passo por um fator de 10 para demonstrar como o erro de arredondamento torna-se dominante à medida que o tamanho do passo é reduzido. Relacione os resultados obtidos à Equação (4.30), lembrando que o valor verdadeiro da derivada é $-0,9125$.

Solução. A função do MATLAB a seguir pode ser implementada para efetuar os cálculos e traçar os resultados. Observe que tanto a função como a sua derivada são argumentos de entrada da função do MATLAB.

```
function deriv_erro(func, dfunc, x, n)
format long
df_verdadeiro=dfunc(x);
h=1;
H(1)=h;
D(1)=(func(x+h)-func(x-h))/(2*h);
E(1)=abs(df_verdadeiro-D(1));
for i=2:n
    h=h/10;
    H(i)=h;
    D(i)=(func(x+h)-func(x-h))/(2*h);
    E(i)=abs(df_verdadeiro-D(i));
end
L=[H' D' E'];
fprintf(' tamanho do passo      diferença fnita      erro verdadeiro\n');
fprintf('%16.10f %20.14f %19.13f\n',L);
loglog(H,E), xlabel('Tamanho do passo'), ylabel('Erro')
title('Gráfico do Erro versus Tamanho do Passo')
format short
```

O código anterior pode ser executado por meio dos seguintes comandos:

```
>> ff=@(x) -0.1*x^4-0.15*x^3-0.5*x^2-0.25*x+1.2;
>> df=@(x) -0.4*x^3-0.45*x^2-x-0.25;
>> deriv_erro(ff,df,0.5,11)
    tamanho do passo      diferença fnita      erro verdadeiro
    1.0000000000          -1.262500000000000          0.350000000000000
    0.1000000000          -0.916000000000000          0.003500000000000
    0.0100000000          -0.912535000000000          0.000035000000000
    0.0010000000          -0.91250035000001          0.0000003500000
    0.0001000000          -0.91250000349985          0.0000000034998
    0.0000100000          -0.9125000003318           0.0000000000332
    0.0000010000          -0.9125000000542           0.0000000000054
    0.0000001000          -0.91249999945031          0.0000000005497
    0.0000000100          -0.91250000333609          0.0000000033361
    0.0000000010          -0.91250001998944          0.0000000199894
    0.0000000001          -0.91250007550059          0.0000000755006
```

**FIGURA 4.12**

Como mostra a Figura 4.12, os resultados estão de acordo com o esperado. Em um primeiro momento, o erro de arredondamento é mínimo e a estimativa é dominada pelo erro de truncamento, portanto, como na Equação (4.29), o erro total cai por um fator de 100 cada vez que o passo é dividido por 10. Entretanto, iniciando em aproximadamente $h = 0,0001$, pode-se ver que o erro de arredondamento começa a ser significativo e reduz a taxa na qual o erro diminui, com erro mínimo alcançado em $h = 10^{-6}$. Além desse ponto, o erro aumenta à medida que o erro de arredondamento predomina.

Como estamos trabalhando com uma função facilmente diferenciável, pode-se também investigar se os resultados são consistentes com a Equação (4.30). Primeiramente, é possível estimar M pela avaliação da derivada terceira da função, como em

$$M = |f^{(3)}(0,5)| = |-2,4(0,5) - 0,9| = 2,1$$

Como o MATLAB tem uma precisão de cerca de 15 a 16 dígitos na base 10, uma estimativa aproximada do limite superior do erro de arredondamento seria cerca de $\varepsilon = 0,5 \times 10^{-16}$, já a substituição desses valores na Equação (4.30) fornece

$$h_{otm} = \sqrt[3]{\frac{3(0,5 \times 10^{-16})}{2,1}} = 4,3 \times 10^{-6}$$

que é da mesma ordem de grandeza do resultado 1×10^{-6} obtido com o MATLAB.

4.4.2 Controle dos erros numéricos

Na maioria dos casos práticos, não se sabe exatamente qual o erro associado aos métodos numéricos – a não ser quando se obtém a solução exata, o que torna a

aproximação numérica desnecessária. Então, na maior parte das aplicações científicas e de engenharia, é necessário contentar-se com alguma estimativa do erro nos cálculos.

Não existe uma abordagem geral e sistemática que obtenha estimativas dos erros numéricos para todos os problemas; em muitos casos, as estimativas de erros têm base na experiência e no julgamento do engenheiro ou cientista.

Embora a análise de erro seja até certo ponto uma arte, sugerimos aqui diversas diretrizes práticas de programação: primeiro e principalmente, deve-se evitar subtrair dois números quase iguais, a fim de evitar perda de significado. Algumas vezes, o profissional pode reorganizar ou reformular o problema para não causar o cancelamento da subtração; caso isso não seja possível, ele pode usar aritmética com precisão estendida. Além disso, ao somar ou subtrair números, é melhor ordená-los e trabalhar com os menores primeiro, o que evita perda de significado.

Também é possível prever os erros numéricos totais usando formulações teóricas, como a série de Taylor, ferramenta básica na análise tanto dos erros de arredondamento quanto dos de truncamento. A previsão do erro numérico total é muito complicada, mesmo para problemas de tamanho moderado, e tende a ser pessimista; logo, costuma ser usada apenas para tarefas em pequena escala.

A tendência é ir em frente com os cálculos numéricos e tentar estimar a acurácia dos resultados, o que pode ser feito com a substituição dos resultados na equação original para verificar se ela realmente é satisfeita.

Finalmente, é preciso estar preparado para fazer experiências numéricas a fim de aumentar a percepção dos erros numéricos e de possíveis problemas mal condicionados, o que pode envolver a repetição de cálculos com tamanho de passo ou método diferentes e a comparação dos resultados. Pode-se empregar análise de sensibilidade para ver como a solução varia quando os parâmetros do modelo ou os valores de entrada variam. É possível também testar algoritmos numéricos diferentes que têm fundamentos teóricos distintos, são baseados em estratégias computacionais diferentes ou têm propriedades de convergência e características de estabilidade distintas.

Quando os resultados de cálculos numéricos são extremamente cruciais e podem provocar a perda de vidas humanas ou ter ramificações econômicas graves, deve-se tomar cuidados especiais, como envolver dois ou mais grupos independentes para resolver o mesmo problema, de modo que os seus resultados possam ser comparados.

O papel dos erros será um tópico de investigação e análise em todas as seções deste livro, em partes específicas destinadas ao assunto.

4.5 ENGANOS, ERROS DE FORMULAÇÃO E INCERTEZA NOS DADOS

Embora as fontes de erros a seguir não estejam diretamente ligadas à maioria dos métodos deste livro, algumas vezes elas têm grande impacto no sucesso do esforço de modelagem, por isso deve-se tê-las em mente ao aplicar as técnicas numéricas no contexto de problemas do mundo real.

4.5.1 Enganos

Todos estamos familiarizados com erros grosseiros ou enganos. Nos primeiros anos de uso dos computadores, resultados numéricos errados podiam às vezes ser

atribuídos ao mau funcionamento do próprio computador, mas hoje essa fonte de erro é altamente improvável e a maioria dos enganos deve ser atribuída à imperfeição humana.

Os enganos podem ocorrer em qualquer estágio do processo de modelagem matemática e podem contribuir para todas as outras componentes do erro. É possível evitá-los apenas por meio do conhecimento profundo dos princípios fundamentais e do cuidado na concepção de uma solução para o problema.

Os enganos são, em geral, desconsiderados na discussão dos métodos numéricos – o que, sem dúvida, decorre do fato de que os erros são em certa medida inevitáveis. Existem, entretanto, formas de minimizar sua ocorrência, como os bons hábitos de programação que foram esboçados no Capítulo 3 e formas simples de verificar se um método numérico está funcionando apropriadamente. Em todo o livro serão discutidas maneiras de verificar os resultados dos cálculos numéricos.

4.5.2 Erros de formulação

Os *erros de formulação* ou *do modelo* se relacionam com vieses que podem ser atribuídos a modelos matemáticos incompletos. Um exemplo de um erro de formulação constantemente negligenciado é o fato de a segunda lei de Newton não levar em conta efeitos relativísticos – o que não afeta a solução do Exemplo 1.1, pois esses erros são mínimos nas escalas de tempo e espaço associadas com o problema do saltador de *bungee jumping* em queda livre.

Contudo, considere que a resistência do ar não seja proporcional ao quadrado da velocidade de queda, como na Equação (1.7), mas que seja relacionada à velocidade e a outros fatores de um modo diferente; se esse fosse o caso, as soluções analítica e numérica obtidas no Capítulo 1 estariam erradas por conta do erro de formulação. É necessário estar ciente desses problemas e perceber que, ao se trabalhar com um modelo precariamente concebido, nenhum método fornecerá resultados adequados.

4.5.3 Incerteza nos dados

Os erros muitas vezes são objeto de análise por causa de incertezas nos dados físicos nos quais um modelo se baseia. Por exemplo, considere que se queira testar o modelo do saltador de *bungee jumping* fazendo um indivíduo pular repetidas vezes e medindo-se sua velocidade depois de um intervalo de tempo especificado. Sem dúvida tais medidas seriam incertas, já que o saltador cairia mais rapidamente durante alguns saltos do que em outros. Esses erros podem exibir tanto falta de acurácia quanto imprecisão. Se os instrumentos consistentemente subestimarem ou superestimarem a velocidade, o profissional estará lidando com um aparelho não acurado ou com viés; entretanto, se as medidas forem aleatoriamente altas e baixas, trata-se de uma questão de precisão.

Os erros nas medidas podem ser quantificados com o resumo dos dados em uma ou mais estatísticas bem escolhidas que transfiram o máximo possível de informação em relação às características específicas dos dados. Essas estimativas descritivas são frequentemente escolhidas para representar (1) a posição do centro da distribuição de dados e (2) o grau de dispersão dos dados. Dessa forma, essas estimativas fornecem uma medida do viés e da imprecisão, respectivamente. Retornaremos ao tópico de caracterização das incertezas dos dados quando discutirmos regressão na Parte 4.

Embora seja necessário estar ciente dos enganos, erros de formulação e incerteza nos dados, os métodos numéricos usados para construir modelos podem ser estudados, na maior parte, independentemente de tais erros. Na maior parte deste livro, a suposição é de que não foram cometidos erros grosseiros, o modelo é seguro e as medidas estão livres de erro, o que torna possível estudar os métodos numéricos sem fatores complicadores.

►► PROBLEMAS

4.1 O método babilônico, um antigo sistema para aproximação da raiz quadrada de qualquer número positivo a , pode ser formulado como

$$x = \frac{x + a/x}{2}$$

Escreva uma função bem estruturada para implementar esse algoritmo, com base no algoritmo esboçado na Figura 4.2.

4.2 Converta os seguintes números na base 2 para a base 10: (a) 1011001, (b) 0,01011, e (c) 110,01001.

4.3 Converta os seguintes números na base 8 para a base 10: 61.565 e 2,71.

4.4 Para computadores, o épsilon da máquina ε também pode ser pensado como o menor número que, quando adicionado a 1, fornece um número maior que 1. Um algoritmo que se baseia nessa ideia pode ser desenvolvido como

Passo 1: Defina $\varepsilon = 1$.

Passo 2: Se $1 + \varepsilon$ for menor que ou igual a 1, então vá para o Passo 5; caso contrário, vá para o Passo 3.

Passo 3: $\varepsilon = \varepsilon/2$.

Passo 4: Retorne para o Passo 2.

Passo 5: $\varepsilon = 2 \times \varepsilon$

Escreva um programa no MATLAB com base nesse algoritmo para determinar o épsilon da máquina. Valide o resultado obtido comparando-o com o valor calculado com a função nativa `eps`.

4.5 De modo similar ao Problema 4.4, desenvolva um programa no MATLAB para determinar o menor número real positivo usado no MATLAB. Tome como base para seu algoritmo a noção de que o computador será incapaz de distinguir de forma confiável entre zero e uma quantidade menor que esse número. Observe que o resultado obtido será diferente do valor calculado com a função nativa `realmin`. Desafio: Investigue os resultados tomando o logaritmo na base 2 do número gerado por seu código e daquele obtido com a função `realmin`.

4.6 Embora não seja geralmente usado, o MATLAB permite que os números sejam expressos com precisão simples. Cada valor é armazenado em 4 bytes, com 1 bit para o sinal, 23 bits para a mantissa e 8 bits para o expoente. Determine o menor e o maior número em ponto flutuante, assim como o épsilon da máquina para a representação com precisão simples – observe que o expoente varia de -126 até 127.

4.7 Para o computador de base 10 do Exemplo 4.2, prove que o épsilon da máquina é 0,05.

4.8 A derivada de $f(x) = 1/(1 - 3x^2)$ é dada por

$$\frac{6x}{(1 - 3x^2)^2}$$

Você espera ter dificuldades calculando essa função em $x = 0,577$? Tente isso usando aritmética com três e quatro algarismos significativos, com truncamento.

4.9 (a) Calcule o polinômio

$$y = x^3 - 7x^2 + 8x - 0,35$$

em $x = 1,37$. Use aritmética com três algarismos significativos e truncamento, e calcule o erro relativo percentual.

(b) Repita (a), mas expresse y como

$$y = ((x - 7)x + 8)x - 0,35$$

Calcule o erro e compare com a parte (a).

4.10 A seguinte série infinita pode ser usada para aproximar e^x :

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

- (a) Demonstre que essa expansão em série de Maclaurin é um caso especial da expansão em série de Taylor [Equação (4.13)] com $x_i = 0$ e $h = x$.
- (b) Use a série de Taylor para fazer uma estimativa de $f(x) = e^{-x}$ em $x_{i+1} = 1$ para $x_i = 0,25$.

Use as versões de ordem zero, um, dois e três e calcule $|\varepsilon_i|$ para cada caso.

4.11 A expansão em série de Maclaurin para $\cos x$ é

$$\cos x = 1 - \frac{x^2}{2} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

Começando com a versão mais simples, $\cos x = 1$, some termos um a um para estimar $\cos(\pi/4)$. Depois que cada novo termo for somado, calcule os erros relativos percentuais verdadeiro e aproximado. Utilize sua calculadora de bolso ou o MATLAB para determinar o valor verdadeiro e some termos até que o valor absoluto da estimativa aproximada de erro fique abaixo de um critério de erro de até dois algarismos significativos.

4.12 Faça os mesmos cálculos do Problema 4.11, mas use a expansão em série de Maclaurin para o $\sin x$ para estimar $\sin(\pi/4)$

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

4.13 Use a expansão em série de Taylor de ordem zero até três para prever $f(3)$ para

$$f(x) = 25x^3 - 6x^2 + 7x - 88$$

utilizando o ponto-base $x = 1$. Calcule o erro relativo percentual ε_i para cada aproximação.

4.14 Demonstre que a Equação (4.11) é exata para todos os valores de x se $f(x) = ax^2 + bx + c$.

4.15 Use a expansão em série de Taylor de ordem zero até quatro para prever $f(2)$ para $f(x) \ln x$ utilizando o ponto-base $x = 1$. Calcule o erro relativo percentual ε_i para cada aproximação e discuta o significado dos resultados.

4.16 Use aproximações por diferenças progressiva e regressiva de $O(h)$ e uma aproximação por diferença centrada de $O(h^2)$ para estimar a primeira derivada da função examinada no Problema 4.13. Calcule a derivada em $x = 2$ usando um tamanho de passo $h = 0,25$. Compare seus resultados com o valor verdadeiro da derivada e interprete-os com base no resto da expansão em série de Taylor.

4.17 Use uma aproximação por diferença centrada de $O(h^2)$ para estimar a segunda derivada da função examinada no Problema 4.13. Faça o cálculo em $x = 2$ utilizando tamanhos de passo de $h = 0,2$ e $0,1$ e compare suas estimativas com o valor verdadeiro da segunda derivada. Interprete seus resultados com base no termo de resto da expansão em série de Taylor.

4.18 Se $|x| < 1$, é sabido que

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \dots$$

Repeta o Problema 4.11 para essa série para $x = 0,1$.

4.19 Para calcular as coordenadas espaciais de um planeta, é necessário calcular a função

$$f(x) = x - 1 - 0,5 \sin x$$

Tome o ponto-base $a = x_i = \pi/2$ no intervalo $[0, \pi]$. Determine a expansão em série de Taylor de ordem mais alta que resulta em um erro máximo de 0,015 no intervalo especificado. O erro é igual ao valor absoluto da diferença entre a função dada e a expansão em série de Taylor específica. (*Sugestão:* resolva graficamente.)

4.20 Considere a função $f(x) = x^3 - 2x + 4$ no intervalo $[-2, 2]$ com $h = 0,25$. Use aproximações por diferenças finitas progressivas, regressivas e centradas para a primeira e a segunda derivadas de modo a ilustrar graficamente qual aproximação é a mais acurada. Trace o gráfico de todas as três aproximações por diferenças finitas da primeira derivada junto com a derivada teórica, e faça o mesmo para a segunda derivada.

4.21 Demonstre a Equação (4.30).

4.22 Repita o Exemplo 4.5, mas para $f(x) = \cos(x)$ em $x = \pi/6$.

4.23 Repita o Exemplo 4.5, mas para a diferença finita progressiva [Equação (4.21)].

4.24 Um exemplo comum em que ocorre cancelamento na subtração envolve a determinação das raízes de uma equação quadrática ou uma parábola que usa a fórmula quadrática:

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

Para os casos nos quais $b^2 \gg 4ac$, a diferença no numerador pode ser muito pequena e podem ocorrer erros de arredondamento. Para tais casos, usa-se uma formulação alternativa a fim de minimizar o cancelamento na subtração:

$$x = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

Use aritmética com cinco algarismos significativos e truncamento para determinar as raízes da equação a seguir, empregando ambas as versões da fórmula quadrática.

$$x^2 - 5.000,002x + 10$$



Parte 2

Raízes e otimização

2.1 VISÃO GERAL

Há muito tempo, você aprendeu a usar a fórmula quadrática

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (\text{PT2.1})$$

para resolver

$$f(x) = ax^2 + bx + c = 0 \quad (\text{PT2.2})$$

Os valores calculados com a Equação (PT2.1) são chamados “raízes” da Equação (PT2.2) e representam os valores de x que tornam a Equação (PT2.2) igual a zero – por essa razão, as raízes são às vezes chamadas de *zeros* da equação.

Embora a fórmula quadrática seja muito cômoda para resolver a Equação (PT2.2), existem muitas outras funções para as quais as raízes não podem ser determinadas tão facilmente. Antes do aparecimento dos computadores digitais, existiam diversas maneiras de determinar as raízes de tais equações. Embora algumas equações – como a Equação (PT2.1) – pudesse ser resolvidas de modo direto, muitas outras não podiam; nesses casos, a única alternativa era uma técnica de solução aproximada.

Um método para obter uma solução aproximada é traçar o gráfico da função e determinar onde ele cruza o eixo dos x – esse ponto, que representa um valor de x para o qual $f(x) = 0$, é a raiz. Embora os métodos gráficos sejam úteis para obter estimativas grosseiras, eles são limitados por conta da falta de precisão. Uma abordagem alternativa é a “técnica” de *tentativa e erro*, que consiste em “chutar” um valor x e avaliar se $f(x)$ é zero. Caso não seja (como é quase sempre o caso), é feito um outro “chute”, e $f(x)$ é novamente calculado para determinar se o novo valor fornece uma estimativa melhor para a raiz. O processo é repetido até que seja feito um “chute” que resulte em um $f(x)$ que esteja próximo de zero.

Esses métodos casuais são obviamente inefficientes e inadequados às necessidades práticas da ciência e da engenharia. Os métodos numéricos representam alternativas também aproximadas, mas que empregam estratégias sistemáticas para mirar na raiz verdadeira. Como será visto nas próximas páginas, a combinação de métodos sistemáticos e computadores torna a solução de muitos problemas aplicados de raízes de equações uma tarefa simples.

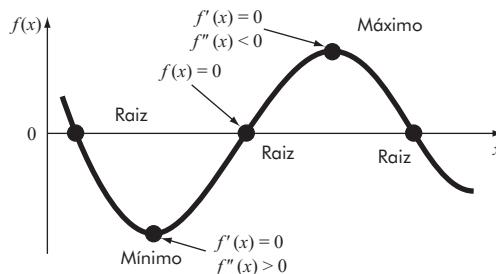


FIGURA PT2.1 Uma função de uma única variável ilustrando a diferença entre raízes e pontos ótimos.

Além das raízes, outro aspecto interessante para engenheiros e cientistas são os valores mínimos e máximos de funções; a determinação desses valores ótimos é chamada de *otimização*. Como você aprendeu no cálculo, tais soluções podem ser obtidas analiticamente determinando-se o valor no qual a função é “achatada”; isto é, onde sua derivada é zero. Embora as soluções analíticas sejam às vezes viáveis, a maior parte dos problemas de otimização necessita de soluções numéricas e computacionais. De um ponto de vista numérico, os métodos de otimização são similares, em essência, aos métodos de localização de raízes, ou seja, ambos envolvem a estimativa e a busca de um ponto em uma função. A diferença fundamental entre os dois tipos de problema é ilustrada na Figura PT2.1: a localização de raízes envolve a busca dos locais onde a função é igual a zero, enquanto a otimização envolve a busca dos pontos extremos da função (um mínimo ou um máximo).

2.2 ORGANIZAÇÃO DA PARTE 2

Os primeiros dois capítulos desta parte são dedicados à localização de raízes. O Capítulo 5 se concentra nos *métodos intervalares*, que começam com aproximações que delimitam ou contêm a raiz e depois diminuem sistematicamente a largura do intervalo – dois exemplos desses métodos são a *bissecção* e a *falsa posição*. Os métodos gráficos são usados para fornecer uma percepção visual das técnicas. Você verá como formulações de erro podem ajudá-lo a determinar quanto esforço computacional é necessário para fazer estimar uma raiz até um nível de precisão pré-especificado.

O Capítulo 6 aborda os *métodos abertos*, que também envolvem iterações sistemáticas de tentativa e erro, mas não exigem aproximações iniciais que delimitem a raiz. Descobriremos que, em geral, esses métodos são computacionalmente mais eficientes do que os métodos intervalares, mas nem sempre funcionam. Ilustraremos vários métodos abertos, como a *iteração de ponto fixo*, o *Newton-Raphson* e o da *secante*.

Seguindo a descrição desses métodos abertos individuais, discutiremos uma abordagem híbrida chamada *método de localização de raízes de Brent*, que analisa o grau de confiabilidade dos métodos intervalares enquanto explora a velocidade dos métodos abertos – essa abordagem forma a base para a função do MATLAB de localização de raízes, `fzero`. Depois de ilustrar como a função `fzero` pode ser usada para solucionar problemas científicos e de engenharia, o Capítulo 6 termina com uma breve discussão sobre métodos especiais dedicados à localização de raízes.

de *polinômios* – descrevemos, em especial, as excelentes capacidades nativas do MATLAB para essa tarefa.

O Capítulo 7 trata do tema *otimização*: primeiro, descrevemos dois métodos intervalares, *busca da seção áurea* e *interpolação quadrática*, para localização do ótimo de uma função de uma única variável. Após isso, discutimos uma abordagem híbrida que combina a busca da seção áurea e a interpolação quadrática; essa abordagem, novamente atribuída a Brent, forma a base da função do MATLAB de localização do mínimo de uma função unidimensional: `fminbnd`. Depois de ilustrar e descrever a função `fminbnd`, a última parte do capítulo fornece uma descrição breve da otimização de funções multidimensionais, com ênfase na descrição e ilustração do uso dos recursos do MATLAB nesta área, em especial a função nativa `fminsearch`. O capítulo termina com um exemplo de como o MATLAB pode ser empregado para resolver problemas de otimização na engenharia e na ciência.

Raízes: métodos intervalares

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar os métodos intervalares para calcular a raiz de uma equação não linear simples. Após esta leitura completa, você será capaz de:

- ✓ Entender o que são problemas de raízes e onde eles ocorrem na engenharia e na ciência.
- ✓ Aprender como determinar uma raiz graficamente.
- ✓ Entender o método da busca incremental e suas deficiências.
- ✓ Aprender como resolver um problema de raízes com o método da bissecção.
- ✓ Compreender como estimar o erro da bissecção e por que ele difere do erro estimado para outros tipos de algoritmos de localização de raízes.
- ✓ Entender o método da falsa posição e como ele difere da bissecção.

VOCÊ TEM UM PROBLEMA

Segundo estudos médicos, as chances de um saltador de *bungee jumping* sofrer uma lesão significativa nas vértebras aumentam bastante se a velocidade de queda livre exceder 36 m/s após 4 s de queda livre. Seu chefe na empresa de *bungee jumping* solicita que você determine a massa na qual esse critério é excedido, dado um coeficiente de arraste de 0,25 kg/m.

Você já deve saber que a solução analítica a seguir pode ser usada para prever a velocidade de queda como uma função do tempo

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) \quad (5.1)$$

Repare, porém, que você não pode manipular essa equação para resolvê-la explicitamente para m (tente o quanto puder!) – ou seja, você não pode isolar a massa no lado esquerdo da equação.

Uma maneira alternativa de olhar para o problema envolve a subtração de $v(t)$ de ambos os lados para fornecer uma nova função:

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) - v(t) \quad (5.2)$$

Agora, podemos ver que a resposta para o problema é o valor de m que torna a função igual a zero, portanto, chamamos isso de um problema de “raízes”. Esse capítulo explicará como o computador é usado para obter tais soluções.

5.1 RAÍZES EM ENGENHARIA E CIÊNCIA

Embora apareçam em outros contextos de problemas, as raízes de equações em geral ocorrem na área de projetos de engenharia. A Tabela 5.1 lista diversos princípios fundamentais rotineiramente usados no trabalho de projeto. Como introduzido no Capítulo 1, as equações ou os modelos matemáticos deduzidos a partir desses princípios são usados para prever variáveis dependentes como função das variáveis independentes, funções forçantes e parâmetros. Observe que, em cada caso, as variáveis dependentes refletem o estado ou o desempenho do sistema, enquanto os parâmetros representam suas propriedades ou composição.

Um exemplo desse modelo é a equação deduzida da segunda lei de Newton para a velocidade do saltador de *bungee jumping*. Se os parâmetros são conhecidos, a Equação (5.1) pode ser usada para prever a velocidade do saltador. Tais cálculos podem ser feitos de modo direto porque v é expressa *explicitamente* como uma função dos parâmetros do modelo, isto é, ela está isolada em um lado do sinal de igualdade.

Entretanto, retornando ao problema do início do capítulo, considere que temos de determinar a massa para que um saltador com um dado coeficiente de arraste atinja uma certa velocidade em um determinado intervalo de tempo. Embora a Equação (5.1) forneça uma representação matemática da inter-relação entre as variáveis e os parâmetros do modelo, ela não pode ser resolvida explicitamente para determinar a massa; em tais casos, dizemos que m está *implícito*.

Tal dilema é real, pois muitos problemas de projeto de engenharia envolvem a especificação das propriedades ou da composição de um sistema (como representado por seus parâmetros) para garantir que funcionem da maneira desejada (como representado por suas variáveis). Assim, esses problemas frequentemente exigem a determinação de parâmetros implícitos.

A solução do dilema é fornecida pelos métodos numéricos para raízes de equações. Para resolver o problema usando métodos numéricos, é conveniente reescrever a Equação (5.1) subtraindo-se a variável dependente v de ambos os lados da equação para obter a Equação (5.2). O valor de m que torna $f(m) = 0$ é, portanto, a raiz da equação, e também representa a massa que resolve o problema de projeto.

TABELA 5.1 Princípios fundamentais usados em problemas e projetos de engenharia.

Princípio fundamental	Variáveis dependentes	Variáveis independentes	Parâmetros
Balanço de calor	Temperatura	Tempo e posição	Propriedades térmicas do material e geometria do sistema
Balanço de massa	Concentração ou quantidade de massa	Tempo e posição	Comportamento químico do material, coeficientes de transferência de massa e geometria do sistema
Balanço de força	Intensidade e direção das forças	Tempo e posição	Resistência do material, propriedades estruturais e geometria do sistema
Balanço de energia	Mudanças nas energias cinética e potencial	Tempo e posição	Propriedades térmicas, massa do material, geometria do sistema
Leis de movimento de Newton	Aceleração, velocidade ou posição	Tempo e posição	Massa do material, geometria do sistema e parâmetros dissipativos tais como atrito ou arraste
Leis de Kirchhoff	Correntes e Tensões	Tempo	Propriedades elétricas (resistência, capacidade e indutância)

As páginas seguintes trazem uma variedade de métodos numéricos e gráficos para a determinação de raízes de relações como a Equação (5.2). Essas técnicas podem ser aplicadas a problemas confrontados na prática da engenharia e nas ciências de forma geral.

5.2 MÉTODOS GRÁFICOS

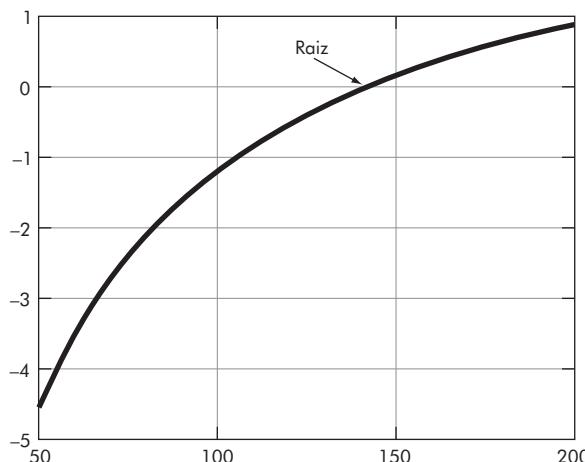
Um método simples para obter uma estimativa da raiz da equação $f(x) = 0$ é fazer um gráfico da função e observar onde ela corta o eixo dos x , ponto que representa o valor de x para o qual $f(x) = 0$ e fornece uma aproximação grosseira da raiz.

EXEMPLO 5.1 A abordagem gráfica

Use a abordagem gráfica para determinar a massa do saltador de *bungee jumping* com um coeficiente de arraste de 0,25 kg/m para ter uma velocidade de 36 m/s após 4 s de queda livre, com a aceleração da gravidade a 9,81 m/s².

Solução. Os seguintes comandos em uma sessão do MATLAB geram um gráfico da Equação (5.2) versus a massa:

```
>> cd = 0.25; g = 9.81; v = 36; t = 4;
>> m = linspace(50,200);
>> fm = sqrt(g*m/cd).*tanh(sqrt(g*cd./m)*t)-v;
>> plot(m,fm), grid
```



A função cruza o eixo m entre 140 e 150 kg (uma inspeção visual do gráfico fornece uma estimativa grosseira da raiz de 145 kg). A validade da estimativa gráfica pode ser verificada pela substituição na Equação (5.2), o que fornece

```
>> sqrt(g*145/cd).*tanh(sqrt(g*cd./145)*t)-v
ans =
0.0456
```

que está próximo de zero. A solução gráfica também pode ser verificada pela substituição na Equação (5.1) com os parâmetros desse exemplo, o que dá

```
>> sqrt(g*145/cd).*tanh(sqrt(g*cd./145)*t)
ans =
36.0456
```

que está muito próximo da velocidade de queda desejada de 36 m/s.

Apesar de as técnicas gráficas terem valor prático limitado por não serem precisas, os métodos gráficos podem ser usados para obter estimativas grosseiras das raízes, usadas como aproximações iniciais para os métodos numéricos discutidos neste capítulo.

Além de fornecer estimativas grosseiras para as raízes, as interpretações gráficas são ferramentas importantes para se entender as propriedades das funções e antecipar as “armadilhas” dos métodos numéricos. Por exemplo, a Figura 5.1 mostra diversas maneiras como as raízes podem ocorrer (ou não) em um intervalo determinado por uma extremidade inferior x_l e por uma extremidade superior x_u . A Figura 5.1b descreve o caso no qual uma única raiz foi delimitada por valores positivos e negativos de $f(x)$, entretanto, a Figura 5.1d, onde $f(x_l)$ e $f(x_u)$ também estão em lados opostos ao eixo x , mostra três raízes ocorrendo nesse intervalo. Em geral, se $f(x_l)$ e $f(x_u)$ têm sinais opostos, existe um número ímpar de raízes no intervalo. Como indicado pelas Figuras 5.2a e c, se $f(x_l)$ e $f(x_u)$ têm o mesmo sinal, ou não existe nenhuma raiz ou existe um número par de raízes entre os valores.

Embora essas generalizações sejam quase sempre verdadeiras, há casos em que elas não funcionam, por exemplo, as funções que são tangentes ao eixo x (Figura 5.2a) ou que são funções descontínuas (Figura 5.2b) podem violar esses princípios. Um exemplo de uma função tangente ao eixo é a equação cúbica $f(x) = (x - 2)(x - 2)(x - 4)$. Observe que $x = 2$ torna dois termos nesse polinômio iguais a zero e note que, matematicamente, $x = 2$ é chamado de *raiz múltipla*. Apesar de elas estarem fora do escopo deste livro, existem técnicas especiais expressamente concebidas para localizar raízes múltiplas (Chapra e Canale, 2010).

A existência de casos do tipo mostrado na Figura 5.2 torna difícil desenvolver algoritmos computacionais gerais que garantam a localização de todas as raízes em um intervalo. Entretanto, quando usados em conjunto com abordagens gráficas, os métodos descritos nas seções seguintes tornam-se bastante úteis para resolver problemas de raízes de equações confrontados por engenheiros, cientistas e matemáticos aplicados.

5.3 MÉTODOS INTERVALARES E ESTIMATIVAS INICIAIS

Se você precisasse resolver um problema de raízes em uma época anterior à computação, provavelmente teria utilizado a técnica de “tentativa e erro” para obter a raiz, isto é, você faria suposições (aproximações ou “chutes”) repetidamente até que o valor da função fosse o mais próximo possível de zero. Esse processo foi bastante facilitado com o advento de ferramentas de software, como as planilhas. Ao permitir que vários “chutes” sejam feitos rapidamente, tais ferramentas podem tornar a abordagem de tentativa e erro aplicável a alguns problemas.

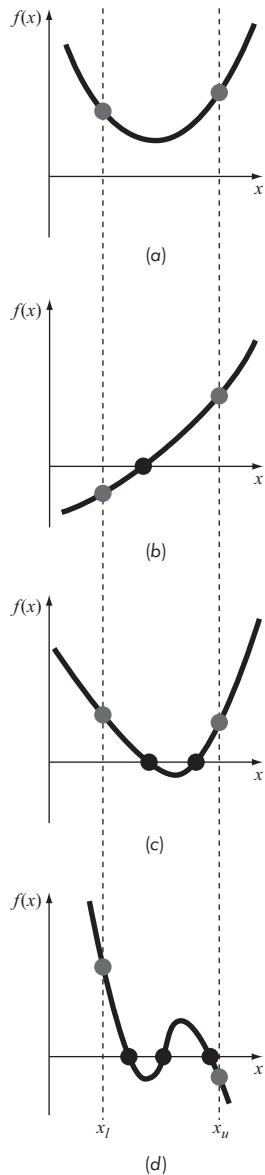


FIGURA 5.1 Ilustração de diversas formas gerais nas quais uma raiz pode ocorrer em um intervalo determinado por uma extremidade inferior x_l e uma extremidade superior x_u . As partes (a) e (c) indicam que, se $f(x_l)$ e $f(x_u)$ tiverem o mesmo sinal, ou não existirão raízes ou existirá um número par de raízes no intervalo. As partes (b) e (d) indicam que, se a função tiver sinais diferentes nas extremidades, existirá um número ímpar de raízes no intervalo.

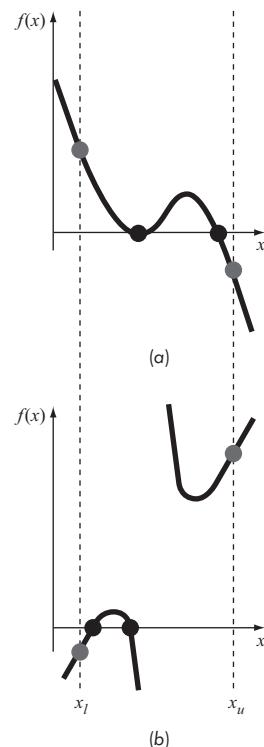


FIGURA 5.2 Ilustração de algumas exceções aos casos gerais mostrados na Figura 5.1. (a) Raízes múltiplas que ocorrem quando a função é tangente ao eixo x : embora as extremidades tenham sinais opostos, há um número par de intercepções com o eixo nesse intervalo. (b) Funções descontínuas nas quais as extremidades de sinais opostos cercam um número par de raízes. É necessário usar estratégias especiais para determinar as raízes nesses casos.

Para muitos outros problemas, porém, é preferível ter métodos que forneçam a resposta correta automaticamente. É interessante notar que, assim como a abordagem por tentativa e erro, esses métodos necessitam de uma “aproximação” inicial para, em seguida, buscar sistematicamente a raiz de uma forma iterativa.

As duas principais classes de métodos disponíveis são distinguidas pelo tipo de aproximação inicial:

- ✓ *Métodos intervalares.* Como o nome indica, eles são baseados em duas aproximações iniciais que “delimitam” a raiz – isto é, estão uma de cada lado da raiz.
- ✓ *Métodos abertos.* Esses métodos podem envolver uma ou mais aproximações iniciais, mas não é necessário que elas delimitem a raiz.

Para problemas bem condicionados, os métodos intervalares sempre funcionam, porém convergem lentamente (isto é, eles costumam gastar mais iterações para chegar à resposta). Já os métodos abertos não funcionam sempre (isto é, eles podem divergir), porém, quando funcionam, geralmente convergem de forma mais rápida.

Em ambos os casos são necessárias aproximações iniciais, que podem surgir naturalmente do contexto físico analisado. No entanto, em outros casos, boas estimativas iniciais podem não ser óbvias, sendo necessário o uso de abordagens automatizadas para obter aproximações. A seção a seguir descreve uma dessas abordagens, a busca incremental.

5.3.1 Busca incremental

Ao aplicar a técnica gráfica no Exemplo 5.1, observou-se que $f(x)$ mudou de sinal em lados opostos da raiz. Em geral, se $f(x)$ for real e contínua no intervalo de x_l a x_u e $f(x_l)$ e $f(x_u)$ tiverem sinais opostos, isto é,

$$f(x_l) f(x_u) < 0 \quad (5.3)$$

então existe pelo menos uma raiz real entre x_l a x_u .

Os *métodos de busca incremental* tiram vantagem dessa observação localizando um intervalo no qual a função muda de sinal. Um problema para esse método é a escolha do comprimento do incremento: se o comprimento for pequeno demais, a busca pode gastar muito tempo; se o comprimento for grande demais, existe a possibilidade de que as raízes próximas possam ser perdidas (Figura 5.3). O problema é agravado pela possível existência de raízes múltiplas.

Um programa pode ser desenvolvido¹ no MATLAB para implementar uma busca incremental a fim de localizar as raízes de uma função `func` dentro do intervalo de `xmin` a `xmax` (Figura 5.4). Um argumento opcional `ns` permite que o usuário especifique o número de divisões (subintervalos) dentro do intervalo; se `ns` for omitido, ele será automaticamente definido como 50. Um laço `for` é usado para percorrer cada subintervalo. Ao ocorrer uma mudança de sinal, os limites superior e inferior do subintervalo são armazenados no arranjo `xb`.

¹ Essa função é uma versão modificada de um programa em MATLAB originalmente apresentado por Recktenwald (2000).

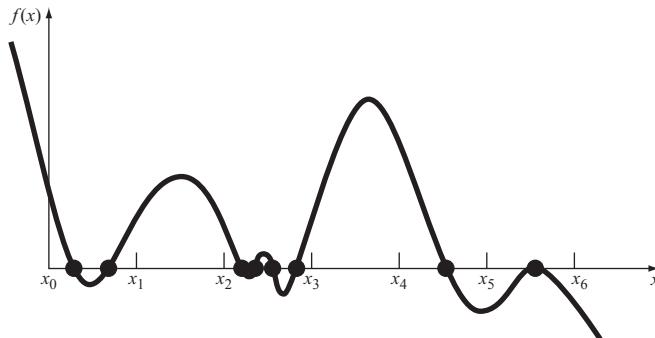


FIGURA 5.3 Casos em que raízes podem ser perdidas porque o comprimento do incremento do método de busca é muito grande. Observe que a última raiz à direita é múltipla e seria perdida independentemente do comprimento do incremento.

```

function xb = busca_inc(func,xmin,xmax,ns)
% busca_inc: busca incremental para localização de raízes
%   xb = busca_inc(func,xmin,xmax,ns):
%       encontra os subintervalos de x que contêm mudanças de sinal
%       de uma função em um intervalo
% entrada:
%   func = nome de uma função
%   xmin, xmax = extremidades do intervalo
%   ns = número de subintervalos (padrão = 50)
% saída:
%   xb(k,1) é o limite inferior da k-ésima mudança de sinal
%   xb(k,2) é o limite superior da k-ésima mudança de sinal
%   Se nenhum subintervalo foi encontrado, xb = [].
if nargin < 3, error('são necessários pelo menos 3 argumentos'), end
if nargin < 4, ns = 50; end %se ns é vazio, ele é definido como 50
% Busca incremental
x = linspace(xmin,xmax,ns);
f = func(x);
nb = 0; xb = []; %xb é nulo a menos que seja detectada mudança de sinal
for k = 1:length(x)-1
    if sign(f(k)) ~= sign(f(k+1)) %verifica se há mudança de sinal
        nb = nb + 1;
        xb(nb,1) = x(k);
        xb(nb,2) = x(k+1);
    end
end
if isempty(xb) %exibe que nenhum subintervalo foi encontrado
    disp('nenhum subintervalo encontrado')
    disp('verifique o intervalo ou aumente ns')
else
    disp('número de subintervalos:') %exibe o número de subintervalos
    disp(nb)
end

```

FIGURA 5.4 Função do MATLAB para implementar uma busca incremental.

EXEMPLO 5.2 Busca incremental

Use a função do `busca_inc` do MATLAB (Figura 5.4) para identificar os subintervalos nos quais ocorre mudança de sinal dentro do intervalo $[3, 6]$ para a função

$$f(x) = \sin(10x) + \cos(3x) \quad (5.4)$$

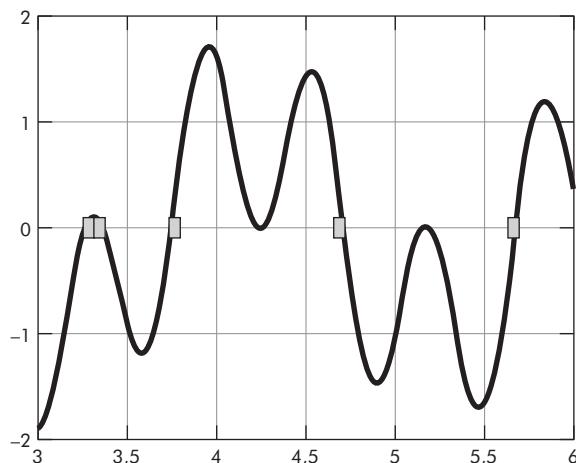
Solução. A sessão do MATLAB que utiliza o número-padrão de subintervalos (50) é

```
>> busca_inc(@(x) sin(10*x)+cos(3*x), 3, 6)
número de subintervalos:
```

5

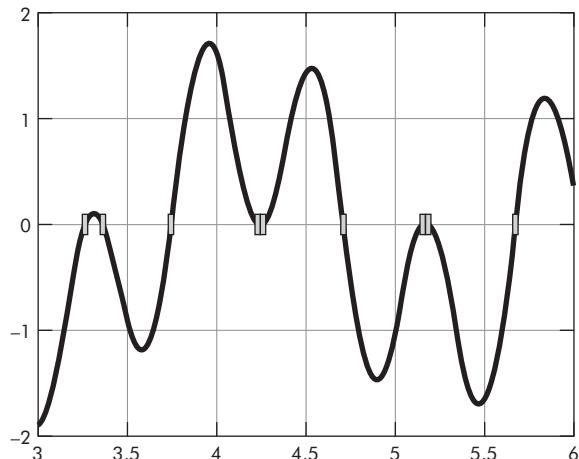
```
ans =
    3.2449    3.3061
    3.3061    3.3673
    3.7347    3.7959
    4.6531    4.7143
    5.6327    5.6939
```

Um gráfico da Equação (5.4) com a localização das raízes é mostrado a seguir.



Embora tenham sido detectadas cinco mudanças de sinal, como os subintervalos são muito grandes, a função não detectou possíveis raízes em $x \approx 4,25$ e $5,2$. Aparentemente, essas possíveis raízes podem ser duplas (já que o gráfico parece tangenciar o eixo x próximo a elas); no entanto, utilizando a ferramenta de *zoom in*, fica claro que cada uma representa duas raízes reais muito próximas umas das outras. A função pode ser executada novamente com mais subintervalos. Conforme mostrado a seguir, utilizando $ns = 100$, todas as nove mudanças de sinal são localizadas.

```
>> busca_inc(@(x) sin(10*x)+cos(3*x), 3, 6, 100)
número de subintervalos:
9
ans =
    3.2424    3.2727
    3.3636    3.3939
    3.7273    3.7576
    4.2121    4.2424
    4.2424    4.2727
    4.6970    4.7273
    5.1515    5.1818
    5.1818    5.2121
    5.6667    5.6970
```



O exemplo anterior ilustra que métodos do tipo força bruta, como a busca incremental, não são infalíveis, por isso seria sensato complementar tais técnicas automáticas com outras informações que auxiliem na localização das raízes. Essas informações podem ser encontradas com o uso de gráficos e a compreensão do problema físico do qual a equação se originou.

5.4 O MÉTODO DA BISSEÇÃO

O *método da bissecção* é uma variação do método de busca incremental no qual o intervalo é sempre dividido na metade. Se uma função muda de sinal em um intervalo, calcula-se o valor da função em seu ponto médio; a posição da raiz é então determinada como o ponto médio do subintervalo no qual a mudança de sinal ocorre. O subintervalo torna-se, então, o intervalo para a próxima iteração, e o processo é repetido até que a raiz seja conhecida com precisão necessária. Veja uma descrição gráfica do método na Figura 5.5 e uma demonstração dos cálculos realmente envolvidos no método no exemplo a seguir.

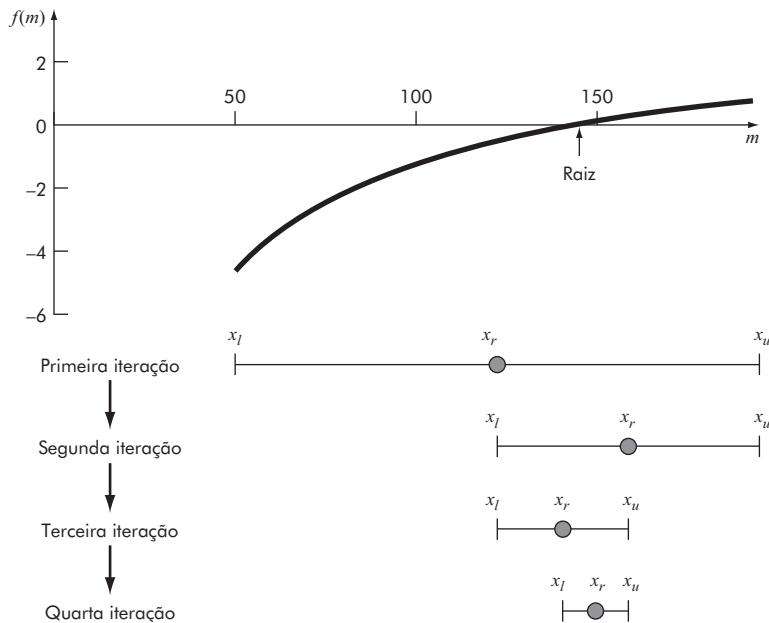


FIGURA 5.5 Descrição gráfica do método da bissecção. Esse gráfico mostra as primeiras quatro iterações para o Exemplo 5.3.

EXEMPLO 5.3 O método da bissecção

Use bissecção para resolver o mesmo problema tratado graficamente no Exemplo 5.1.

Solução. O primeiro passo na bissecção é estimar dois valores para a incógnita (no presente problema, m) que forneçam valores de $f(m)$ com sinais diferentes. A partir da solução gráfica no Exemplo 5.1, pode-se ver que a função muda de sinal entre 50 e 200. O gráfico obviamente sugere aproximações iniciais ótimas (sejam, por exemplo, 140 e 150), mas, para propósitos ilustrativos, vamos supor que não temos acesso ao gráfico e foram feitas aproximações conservativas, portanto, a aproximação inicial da raiz x_r é o ponto médio do intervalo

$$x_r = \frac{50 + 200}{2} = 125$$

Observe que o valor exato da raiz é 142,7376, o que significa que o valor de 125 calculado aqui tem um erro percentual verdadeiro de

$$|\varepsilon_t| = \left| \frac{142,7376 - 125}{142,7376} \right| \times 100\% = 12,43\%$$

A seguir, calcula-se o produto do valor da função no extremo inferior e no ponto médio:

$$f(50)f(125) = -4,579(-0,409) = 1,871$$

que é muito maior que zero e, portanto, não ocorre troca de sinal entre a extremidade inferior e o ponto médio. Consequentemente, a raiz deve estar localizada no intervalo superior entre 125 e 200 e criamos um novo intervalo, redefinindo a extremidade inferior como 125.

Neste ponto, o novo intervalo se estende de $x_l = 125$ até $x_u = 200$. Uma estimativa revisada da raiz pode, então, ser calculada por

$$x_r = \frac{125 + 200}{2} = 162,5$$

que representa um erro percentual verdadeiro de $|\varepsilon_t| = 13,85\%$. O processo pode ser repetido para se obter estimativas refinadas, por exemplo:

$$f(125)f(162,5) = -0,409(0,359) = -0,147$$

Portanto, a raiz está agora no intervalo inferior, entre 125 e 162,5. A extremidade superior é redefinida como 162,5 e a estimativa da raiz para a terceira iteração é calculada como

$$x_r = \frac{125 + 162,5}{2} = 143,75$$

que representa um erro relativo percentual de $\varepsilon_t = 0,709\%$. O método pode ser repetido até que o resultado seja acurado o suficiente para satisfazer suas necessidades.

Terminamos o Exemplo 5.3 com a afirmação de que se poderia continuar com o método para obter uma estimativa mais refinada da raiz. Agora é preciso desenvolver um critério objetivo para decidir quando parar o método.

Uma sugestão inicial poderia ser parar os cálculos quando o erro cair abaixo de um determinado nível. Como ilustração, no Exemplo 5.3, o erro relativo verdadeiro caiu de 12,43 para 0,709% durante os cálculos. A decisão pode ser parar quando o erro cair abaixo de, por exemplo, 0,5%. Essa estratégia é falha porque os erros estimados no exemplo foram baseados no conhecimento da raiz verdadeira da função – o que não seria o caso em situações reais, nas quais a raiz é desconhecida.

É necessária, portanto, uma estimativa de erro que não dependa do conhecimento prévio dessa raiz; pode-se fazer isso estimando-se um erro relativo percentual como em [lembre-se da Equação (4.5)],

$$|\varepsilon_a| = \left| \frac{x_r^{\text{novo}} - x_r^{\text{velho}}}{x_r^{\text{novo}}} \right| 100\% \quad (5.5)$$

onde x_r^{novo} é a raiz da iteração atual e x_r^{velho} é a raiz da iteração prévia. Quando ε_a torna-se menor que um critério de parada pré-especificado ε_s , param-se os cálculos.

EXEMPLO 5.4 Estimativa de erro para a bissecção

Continue o Exemplo 5.3 até que o erro aproximado fique abaixo do critério de parada $\varepsilon_s = 0,5\%$, usando a Equação (5.5) para calcular os erros.

Solução. Os resultados das primeiras duas iterações no Exemplo 5.3 foram 125 e 162,5. A substituição desses valores na Equação (5.5) fornece

$$|\varepsilon_a| = \left| \frac{162,5 - 125}{162,5} \right| 100\% = 23,08\%$$

Lembre que o erro relativo percentual verdadeiro para a estimativa da raiz de 162,5 era 13,85%; logo, ε_a é maior que ε_t , comportamento manifestado nas outras iterações:

Iteração	x_l	x_u	x_r	$ \varepsilon_a (\%)$	$ \varepsilon_t (\%)$
1	50	200	125		12,43
2	125	200	162,5	23,08	13,85
3	125	162,5	143,75	13,04	0,71
4	125	143,75	134,375	6,98	5,86
5	134,375	143,75	139,0625	3,37	2,58
6	139,0625	143,75	141,4063	1,66	0,93
7	141,4063	143,75	142,5781	0,82	0,11
8	142,5781	143,75	143,1641	0,41	0,30

Assim, depois de oito iterações, $|\varepsilon_a|$ finalmente fica abaixo de $\varepsilon_s = 0,5\%$, e os cálculos podem parar.

Esses resultados estão resumidos na Figura 5.6. A natureza “irregular” do erro verdadeiro deve-se ao fato de que, na bissecção, a raiz verdadeira pode estar em qualquer lugar no intervalo que a cerca. Os erros verdadeiro e aproximado estão bem distantes quando o intervalo está centrado na raiz verdadeira, e bem próximos quando a raiz verdadeira está próxima de qualquer uma das extremidades.

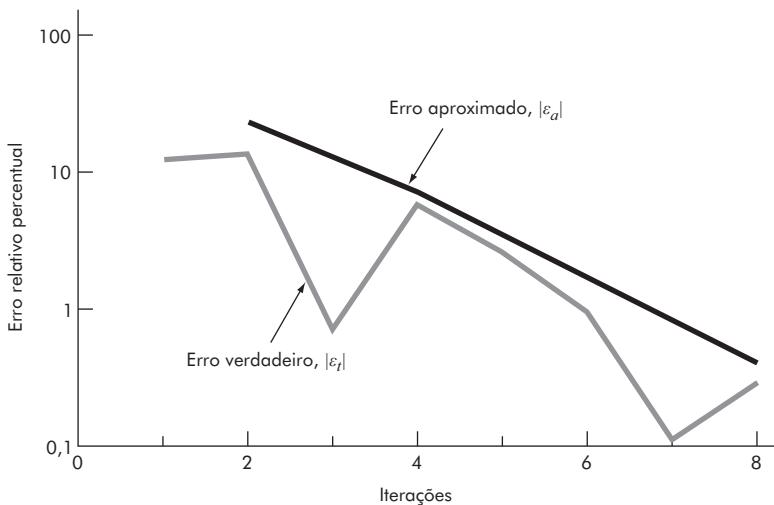


FIGURA 5.6 Erros para o método da bissecção: os erros verdadeiro e estimado são mostrados em função do número de iterações.

Embora o erro aproximado não forneça uma estimativa exata do erro verdadeiro, a Figura 5.6 sugere que $|\varepsilon_a|$ captura a tendência geral para baixo de $|\varepsilon_t|$. Além disso, o gráfico mostra que $|\varepsilon_a|$ é sempre maior do que $|\varepsilon_t|$; logo, quando $|\varepsilon_a|$ cai abaixo

de $|\varepsilon_s|$, podemos parar os cálculos com a confiança de que a raiz é conhecida e tão acurada quanto o nível aceitável pré-especificado.

Ainda que seja perigoso tirar conclusões gerais de um único exemplo, vemos que $|\varepsilon_a|$ será sempre maior que $|\varepsilon_s|$ para o método da bissecção, o que ocorre porque, cada vez que uma raiz é localizada usando a bissecção com $x_r = (x_l + x_u)/2$, sabemos que a raiz verdadeira está em algum ponto em um intervalo de comprimento $\Delta x = x_u - x_l$; assim, a raiz deve estar a $\pm \Delta x/2$ de nossa estimativa. Por exemplo, quando o Exemplo 5.4 foi parado, poderíamos fazer a afirmação definitiva de que

$$x_r = 143,1641 \pm \frac{143,7500 - 142,5781}{2} = 143,1641 \pm 0,5859$$

Em essência, a Equação (5.5) fornece um limitante superior para o erro verdadeiro, e para que esse limitante fosse excedido, a raiz verdadeira deveria estar fora do intervalo que a cerca – o que, por definição, nunca poderia ocorrer no método da bissecção. Embora a bissecção seja em geral mais lenta que outros métodos, a simplicidade de sua análise de erro é certamente um aspecto positivo em relação a outras técnicas para localizar raízes.

Outro benefício do método da bissecção é que o número de iterações necessárias para se chegar a um erro absoluto pode ser calculado antes de começar os cálculos. Isso pode ser visto reconhecendo-se que, antes de começar a técnica, o erro absoluto é

$$E_a^0 = x_u^0 - x_l^0 = \Delta x^0$$

onde o sobrescrito indica a iteração. Então, antes de começar o método, estamos na “iteração zero”; depois da primeira iteração, o erro se torna

$$E_a^1 = \frac{\Delta x^0}{2}$$

Como cada iteração sucessiva divide o erro por dois, uma fórmula geral que relaciona o erro e o número de iterações, n , é

$$E_a^n = \frac{\Delta x^0}{2^n}$$

Se $E_{a,d}$ é o erro desejado, essa equação pode ser resolvida por²

$$n = \frac{\log(\Delta x^0 / E_{a,d})}{\log 2} = \log_2 \left(\frac{\Delta x^0}{E_{a,d}} \right) \quad (5.6)$$

Vamos testar a fórmula. Para o Exemplo 5.4, o intervalo inicial era $\Delta x_0 = 200 - 50 = 150$; depois de seis iterações, o erro absoluto era

$$E_a = \frac{|143,7500 - 142,5781|}{2} = 0,5859$$

² O MATLAB possui a função `log2`, que permite calcular diretamente o logaritmo na base 2. Se a calculadora de bolso ou a linguagem de computador que você estiver utilizando não incluir o logaritmo na base 2 como uma função intrínseca, a equação a seguir mostra uma maneira acessível de calculá-lo: em geral, $\log_b(x) = \log(x)/\log(b)$.

Podemos substituir esses valores na Equação (5.6) para obter

$$n = \log_2(150/0,5859) = 8$$

Logo, se soubermos antecipadamente que um erro abaixo de 0,5859 é aceitável, a fórmula diz que seis iterações fornecem o resultado procurado.

Embora se tenha enfatizado o uso do erro relativo por razões óbvias, há casos em que – talvez pelo conhecimento do contexto do problema – se poderá especificar um erro absoluto. Nesses casos, a bissecção, com a Equação (5.6), pode fornecer um algoritmo útil na localização das raízes.

5.4.1 Programa em MATLAB: bissec

Veja uma função em MATLAB que implementa o método da bissecção na Figura 5.7: os argumentos de entrada são a função (`func`) com as aproximações inferior (`x1`) e superior (`xu`). Um critério de parada opcional (`es`) e um número máximo de iterações (`maxit`) são argumentos de entrada opcionais. A função verifica, primeiro, se há argumentos de entrada suficientes e se no intervalo entre as aproximações (`x1` e `xu`) iniciais existe mudança de sinal da função; se não, uma mensagem de erro é exibida e a função é encerrada. Ela também atribui valores-padrão às variáveis `maxit` e `es`, se nenhum valor for definido pelo usuário. Em seguida, um laço `while...break` é empregado para implementar o algoritmo da bissecção até o erro aproximado cair abaixo de `es` ou o número de iterações exceder `maxit`.

A função pode ser utilizada para resolver o problema colocado no início do capítulo. Lembre-se de que era necessário determinar a massa para a qual a velocidade de um saltador de *bungee jumping* em queda livre excede 36 m/s após 4 s de queda, dado um coeficiente de arraste de 0,25 kg/m, portanto, deve-se encontrar a raiz de

$$f(m) = \sqrt{\frac{9,81m}{0,25}} \tanh\left(\sqrt{\frac{9,81(0,25)}{m}} 4\right) - 36$$

No Exemplo 5.1 foi gerado um gráfico dessa função *versus* a massa e estimou-se que a raiz estava entre 140 e 150 kg. A função `bissec` da Figura 5.7 pode ser usada para determinar a raiz conforme mostrado a seguir:

```
>> fm=@(m) sqrt(9.81*m/0.25)*tanh(sqrt(9.81*0.25/m)*4)-36;
>> [massa fx ea iter]=bissec(fm,40,200)

massa =
    142.7377
fx =
    4.6089e-007
ea =
    5.3450e-005
iter =
    21
```

Assim, um resultado de $m = 142,7377$ kg é obtido após 21 iterações com um erro relativo aproximado de $\varepsilon_a = 0,00005345\%$ e um valor de função próximo de zero.

```

function [raiz,fx,ea,iter]=bissec(func,xl,xu,es,maxit,varargin)
% bissec: localização de raízes pelo método da bissecção
% [raiz,fx,ea,iter]=bisect(func,xl,xu,es,maxit,p1,p2,...):
%   usa o método da bissecção para encontrar a raiz de func
% entrada:
%   func = nome da função
%   xl, xu = aproximações inferior e superior
%   es = erro relativo desejado (padrão = 0,0001%)
%   maxit = número máximo de iterações permitidas (padrão = 50)
%   p1,p2,... = parâmetros adicionais usados por func
% saída:
%   raiz = raiz real
%   fx = valor da função em raiz
%   ea = erro relativo aproximado (%)
%   iter = número de iterações

if nargin<3,error('são necessários pelo menos 3 argumentos de
entrada'),end
test = func(xl,varargin{:})*func(xu,varargin{:});
if test>0,error('não há mudança de sinal'),end
if nargin<4|isempty(es), es=0.0001;end
if nargin<5|isempty(maxit), maxit=50;end
iter = 0; xr = xl; ea = 100;
while (1)
    xr_velho = xr;
    xr = (xl + xu)/2;
    iter = iter + 1;
    if xr ~= 0,ea = abs((xr - xr_velho)/xr) * 100;end
    test = func(xl,varargin{:})*func(xr,varargin{:});
    if test < 0
        xu = xr;
    elseif test > 0
        xl = xr;
    else
        ea = 0;
    end
    if ea <= es | iter >= maxit,break,end
end
raiz = xr; fx = func(xr, varargin{:});

```

FIGURA 5.7 Função do MATLAB para implementar o método da bissecção.

5.5 O MÉTODO DA FALSA POSIÇÃO

O *método da falsa posição* (também chamado de método da interpolação linear) é outro método intervalar bem conhecido, bastante similar ao método da bissecção, com a exceção de que ele utiliza uma estratégia diferente para chegar à sua nova estimativa da raiz. Em vez de dividir o intervalo de x_l a x_u em duas partes iguais, ele localiza a raiz ligando $f(x_l)$ e $f(x_u)$ por uma reta (Figura 5.8). A interseção dessa reta com o eixo x representa uma estimativa melhorada da raiz, ou seja, a forma da função

influencia a nova estimativa da raiz. Usando triângulos semelhantes, a intersecção da reta com o eixo x pode ser estimada como (ver Chapra e Canala, 2010, para detalhes)

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)} \quad (5.7)$$

Essa é a *fórmula da falsa posição*. O valor de x_r calculado com a Equação (5.7) substitui então qualquer uma das duas aproximações iniciais, x_l ou x_u , que forneça um valor da função com o mesmo sinal que $f(x_r)$; dessa forma, os valores de x_l e x_u sempre delimitam a raiz verdadeira. O processo é repetido até que a raiz seja estimada adequadamente, e o algoritmo é idêntico ao da bissecção (Figura 5.7), com a diferença de que a Equação (5.7) é utilizada para calcular a estimativa da raiz.

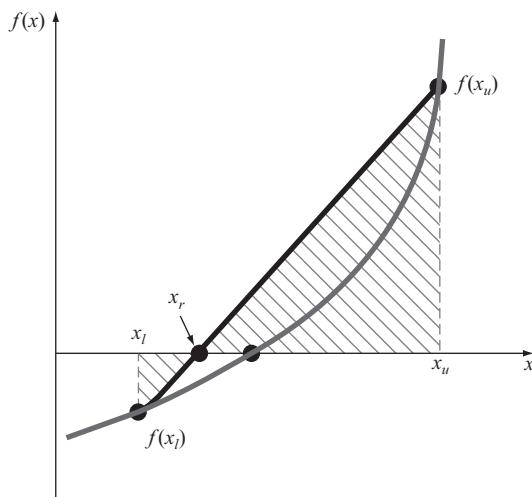


FIGURA 5.8 O método da falsa posição.

EXEMPLO 5.5 O método da falsa posição

Use o método da falsa posição para resolver o mesmo problema abordado graficamente e com o método da bissecção nos Exemplos 5.1 e 5.3.

Solução. Como no Exemplo 5.3, comece os cálculos com aproximações $x_l = 50$ e $x_u = 200$. Primeira iteração:

$$x_l = 50 \quad f(x_l) = -4,579387$$

$$x_u = 200 \quad f(x_u) = 0,860291$$

$$x_r = 200 - \frac{0,860291(50 - 200)}{-4,579387 - 0,860291} = 176,2773$$

que tem um erro relativo verdadeiro de 23,5%.

Segunda iteração:

$$f(x_l) f(x_r) = -2,592732$$

A raiz, portanto, está no primeiro subintervalo e x_r se torna a aproximação superior da próxima iteração, $x_u = 176,2773$,

$$\begin{aligned}x_l &= 50 & f(x_l) &= -4,579387 \\x_u &= 176,2773 & f(x_u) &= 0,566174 \\x_r &= 176,2773 - \frac{0,566174(50 - 176,2773)}{-4,579387 - 0,566174} & &= 162,3828\end{aligned}$$

que tem erros relativos verdadeiro e aproximado de 13,76% e 8,56%, respectivamente. Caso deseje refinar a estimativa da raiz, é possível fazer iterações adicionais.

Embora o método da falsa posição geralmente tenha um melhor desempenho que a bissecção, há casos em que seu desempenho é deficiente ou inferior ao da bissecção, como no exemplo a seguir.

EXEMPLO 5.6 Um caso em que a bissecção é preferível à falsa posição

Use a bissecção e a falsa posição para localizar a raiz de

$$f(x) = x^{10} - 1$$

entre $x = 0$ e $1,3$.

Solução. Usando a bissecção, os resultados podem ser resumidos por

Iteração	x_l	x_u	x_r	ε_a (%)	ε_t (%)
1	0	1,3	0,65	100,0	35
2	0,65	1,3	0,975	33,3	2,5
3	0,975	1,3	1,1375	14,3	13,8
4	0,975	1,1375	1,05625	7,7	5,6
5	0,975	1,05625	1,015625	4,0	1,6

Depois de cinco iterações, o erro verdadeiro foi reduzido para menos de 2%. Na falsa posição é obtida uma saída muito diferente:

Iteração	x_l	x_u	x_r	ε_a (%)	ε_t (%)
1	0	1,3	0,09430		90,6
2	0,09430	1,3	0,18176	48,1	81,8
3	0,18176	1,3	0,26287	30,9	73,7
4	0,26287	1,3	0,33811	22,3	66,2
5	0,33811	1,3	0,40788	17,1	59,2

Depois de cinco iterações, o erro verdadeiro diminuiu para cerca de 59%. É possível avaliar melhor tais resultados analisando-se o gráfico da função. Como na Figura 5.9, a curva viola a hipótese básica na qual a falsa posição está baseada – isto é, se $f(x_l)$ está muito mais próxima de zero do que $f(x_u)$, então a raiz está mais próxima de x_l do que de x_u (lembre-se da Figura 5.8). Devido à forma da função presente, o contrário é verdadeiro.

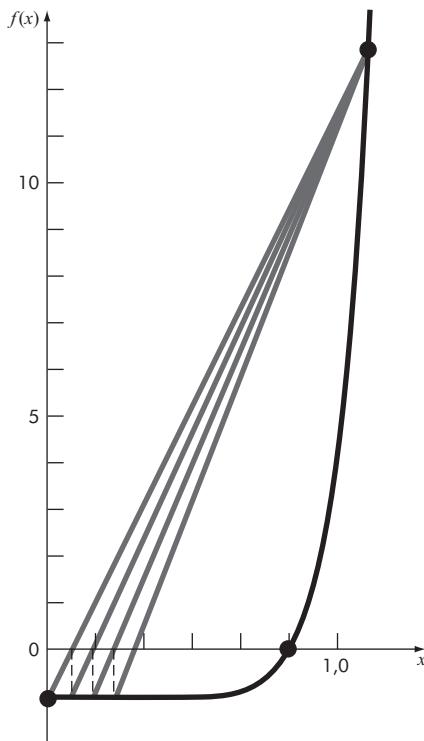


FIGURA 5.9 Gráfico de $f(x) = x^{10} - 1$, ilustrando a convergência lenta do método da falsa posição.

O exemplo anterior ilustra que, com relação aos métodos de localização de raízes, generalizações não costumam ser aceitas. Embora um método como a falsa posição seja quase sempre superior à bissecção, invariavelmente há casos que violam essas conclusões gerais. Portanto, além de se usar a Equação (5.5), os resultados deveriam sempre ser verificados substituindo a estimativa da raiz na equação original e verificando se o resultado está próximo de zero.

O exemplo também ilustra uma grande fraqueza do método da falsa posição: ele é unilateral, ou seja, conforme as iterações continuam, uma das extremidades do intervalo tenderá a permanecer fixa – o que pode levar à convergência insatisfatória, particularmente para funções com curvatura significativa. Possíveis soluções para essa desvantagem estão disponíveis em outras referências (Chapra e Canale, 2010).

► 5.6 ESTUDO DE CASO GASES DE EFEITO ESTUFA E ÁGUA DA CHUVA

Contextualização. Sabe-se, de acordo com estudos e documentos, que os níveis atmosféricos de vários dos chamados gases “de efeito estufa” têm aumentado nos últimos 50 anos. A Figura 5.10 mostra dados da pressão parcial do dióxido de carbono (CO_2) coletados em Mauna Loa, Havaí, de 1958 até 2008. A tendência nesses dados pode ser ajustada com um polinômio quadrático,³

$$p_{\text{CO}_2} = 0,012226(t - 1983)^2 + 1,418542(t - 1983) + 342,38309$$

onde p_{CO_2} = a pressão parcial do CO_2 (ppm). Esses dados indicam que os níveis aumentaram um pouco mais de 22% durante o período, passando de 316 para 386 ppm.

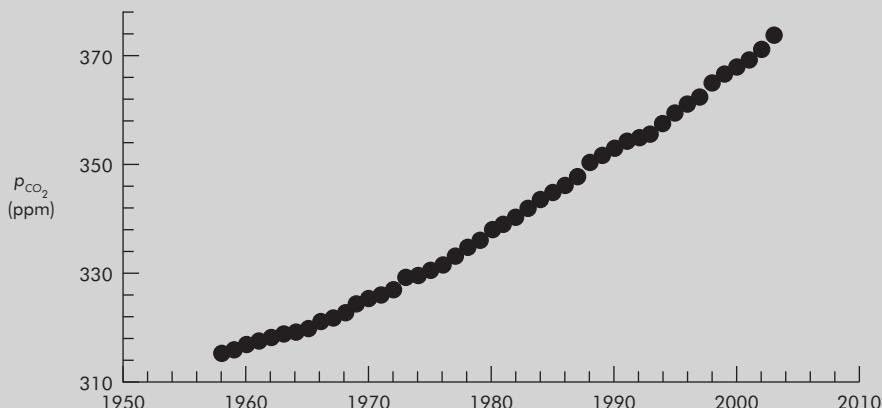


FIGURA 5.10 Média anual de pressões parciais do dióxido de carbono atmosférico (ppm) medidas em Mauna Loa, Havaí.

Uma questão que pode ser avaliada é como essa tendência está afetando o pH da água da chuva. Sabe-se que, fora das áreas urbanas e industriais, o dióxido de carbono é o principal determinante do pH da chuva, que a partir da medida da atividade dos íons de hidrogênio, determina sua acidez ou alcalinidade. Para soluções aquosas diluídas, ele pode ser calculado como

$$\text{pH} = -\log_{10}[\text{H}^+] \quad (5.8)$$

onde $[\text{H}^+]$ é a concentração molar dos íons de hidrogênio.

As seguintes cinco equações governam a química da água da chuva:

$$K_1 = 10^6 \frac{[\text{H}^+][\text{HCO}_3^-]}{K_H p_{\text{CO}_2}} \quad (5.9)$$

$$K_2 = \frac{[\text{H}^+][\text{CO}_3^{2-}]}{[\text{HCO}_3^-]} \quad (5.10)$$

$$K_w = [\text{H}^+][\text{OH}^-] \quad (5.11)$$

³ Na Parte 4 aprenderemos como determinar tais polinômios.

$$c_T = \frac{K_H p_{CO_2}}{10^6} + [HCO_3^-] + [CO_3^{2-}] \quad (5.12)$$

$$0 = [HCO_3^-] + 2[CO_3^{2-}] + [OH^-] - [H^+] \quad (5.13)$$

onde K_H é a constante de Henry e K_1 , K_2 e K_w são coeficientes de equilíbrio. As cinco incógnitas são C_T = carbono inorgânico total, $[HCO_3^-]$ = bicarbonato, $[CO_3^{2-}]$ = carbonato, $[H^+]$ = íon de hidrogênio e $[OH^-]$ = íon de hidroxila. Observe como a pressão parcial do CO_2 aparece nas Equações (5.9) e (5.12).

Use essas equações para calcular o pH da água da chuva, dado que $K_H = 10^{-1,46}$, $K_1 = 10^{-6,3}$, $K_2 = 10^{-10,3}$ e $K_w = 10^{-14}$ e compare os resultados em 1958, quando a pressão parcial do CO_2 (p_{CO_2}) era 315, e em 2008, quando ela era 386 ppm. Ao selecionar o método numérico para seus cálculos, considere os seguintes aspectos:

- ✓ O pH da chuva em áreas virgens sempre está entre 2 e 12.
- ✓ O pH pode ser medido apenas com duas casas de precisão decimal.

Solução. Há várias maneiras de resolver esse sistema de cinco equações: uma delas é eliminar as incógnitas combinando-as para produzir uma única função que dependa apenas de $[H^+]$. Para fazer isso, primeiro resolva as Equações (5.9) e (5.10) para

$$[HCO_3^-] = \frac{K_1}{10^6[H^+]} K_H p_{CO_2} \quad (5.14)$$

$$[CO_3^{2-}] = \frac{K_2[HCO_3^-]}{[H^+]} \quad (5.15)$$

Substitua a Equação (5.14) na Equação (5.15)

$$[CO_3^{2-}] = \frac{K_2 K_1}{10^6[H^+]^2} K_H p_{CO_2} \quad (5.16)$$

As Equações (5.14) e (5.16) podem ser substituídas com a Equação (5.11) na Equação (5.13) para fornecer

$$0 = \frac{K_1}{10^6[H^+]} K_H p_{CO_2} + 2\frac{K_2 K_1}{10^6[H^+]^2} K_H p_{CO_2} + \frac{K_w}{[H^+]} - [H^+] \quad (5.17)$$

Embora não seja imediatamente aparente, esse resultado é um polinômio de terceira ordem em $[H^+]$; assim, sua raiz pode ser usada para calcular o pH da água da chuva.

Agora, devemos decidir qual método numérico empregar para obter a solução. Há duas razões pelas quais o método da bissecção seria uma boa escolha: primeiro, o fato de que o pH está sempre dentro do intervalo de 2 a 12 nos fornece duas boas aproximações iniciais; segundo, como o pH pode apenas ser medido com duas casas de precisão decimal, um erro absoluto de $E_{a,d} = \pm 0,005$ ainda é aceitável. Lembre-se de que dados o intervalo inicial e o erro desejado, pode-se calcular o número de iterações *a priori*. Substituindo os valores presentes na Equação (5.6), temos

```
>> dx=12-2;
>> Ead=0.005;
>> n=log2(dx/Ead)
n =
    10.9658
```

Onze iterações da bissecção produzirão a precisão desejada.

Antes de implementar a bissecção, devemos expressar a Equação (5.17) como uma função, mas como ela é relativamente complicada, vamos armazená-la como uma função do MATLAB:

```

function f = fpH(pH,pCO2)
K1=10^-6.3; K2=10^-10.3; Kw=10^-14;
KH=10^-1.46;
H=10^-pH;
f=K1/(1e6*H)*KH*pCO2+2*K2*K1/(1e6*H)*KH*pCO2+Kw/H-H;

```

Podemos utilizar a função do MATLAB apresentada na Figura 5.7 para obter a solução, conforme mostrado a seguir. Observe que o valor do erro relativo desejado foi definido em um nível muito baixo ($\varepsilon_a = 1 \times 10^{-8}$), de modo que o limite de iterações (maxit) é alcançado primeiro e, assim, são implementadas exatamente 11 iterações.

```
>> [pH1958 fx ea iter]=bissec(@fpH,2,12,1e-8,11,315)
pH1958 =
    5.6279
fx =
    -2.7163e-008
ea =
    0.0868
iter =
    11
```

Assim, o pH é calculado como 5,6279, com um erro relativo de 0,0868%. Podemos ter certeza de que o resultado arredondado 5,63 está correto com duas casas decimais, o que pode ser verificado executando novamente a função com mais iterações. Por exemplo, definindo `maxit` como 50 produz:

```
>> [pH1958 fx ea iter] = bissec(@fpH,2,12,1e-8,50,315)
pH1958 =
    5.6304
fx =
    1.6150e-015
ea =
    5.1690e-009
iter =
    35
```

Para 2008 o resultado é:

```
>> [pH2008 fx ea iter]=bissec(@fpH,2,12,1e-8,50,386)
pH2008 =
    5.5864
fx =
    3.2926e-015
ea =
    5.2098e-009
iter =
    35
```



Curiosamente, os resultados indicam que o crescimento de 22,5% nos níveis de CO₂ na atmosfera produziu uma queda de apenas 0,78% no pH – ainda que isso seja verdadeiro, lembre-se de que o pH representa uma escala logarítmica, como definido pela Equação (5.8). Consequentemente, uma queda unitária no pH representa um aumento de uma ordem de grandeza (isto é, 10 vezes) nos íons de hidrogênio. A concentração pode ser calculada como [H⁺] = 10^{-pH} e sua variação percentual pode ser determinada da seguinte forma:

```
>> ((10^-pH2008-10^-pH1958)/10^-pH1958)*100
ans =
10.6791
```

Portanto, a concentração de íons de hidrogênio aumentou cerca de 10,7%.

Há bastante controvérsia relacionada às tendências dos gases de efeito estufa. O debate questiona se o aumento desses gases está contribuindo ou não para o aquecimento global; entretanto, independentemente das últimas implicações, é preocupante perceber que algo tão grande como a nossa atmosfera mudou tanto durante um período relativamente curto de tempo. Esse estudo de caso ilustra como os métodos numéricos e o MATLAB podem ser empregados para analisar e interpretar tais tendências. Nos próximos anos, engenheiros e cientistas podem usar tais ferramentas para compreender melhor tais fenômenos e ajudar a racionalizar o debate.

►► PROBLEMAS

5.1 Use a bissecção para determinar o coeficiente de arraste necessário de modo que um saltador de *bungee jumping* de 80 kg tenha uma velocidade de 36 m/s após 4 s de queda livre. Observação: A aceleração da gravidade é 9,81 m/s². Comece com as estimativas iniciais de $x_l = 0,1$ e $x_u = 0,2$, e efetue os cálculos até que o erro relativo aproximado caia abaixo de 2%.

5.2 Desenvolva sua própria função do MATLAB para a bissecção em um modo similar à Figura 5.7. Porém, em vez utilizar o número máximo de iterações e a Equação (5.5), empregue a Equação (5.6) como seu critério de parada. Certifique-se de arredondar o resultado da Equação (5.6) para o próximo inteiro mais elevado. Teste sua função por meio da resolução do Problema 5.1 utilizando $E_{a,d} = 0,0001$.

5.3 Repita o Problema 5.1, mas use o método da falsa posição para obter a solução.

5.4 Desenvolva uma função em MATLAB para o método da falsa posição. Teste-a resolvendo o Problema 5.1.

5.5 (a) Determine as raízes de $f(x) = -12 - 21x + 18x^2 - 2,75x^3$ graficamente. Além disso, determine a primeira raiz da função com (b) bissecção e (c) falsa posição. Para (b) e (c) use as aproximações iniciais $x_l = -1$ e $x_u = 0$ e um critério de parada de 1%.

5.6 Localize a primeira raiz não trivial de $\sin(x) = x^2$, onde x está em radianos. Use uma técnica gráfica

e a bissecção com o intervalo inicial de 0,5 a 1. Faça os cálculos até que ε_a seja menor que $\varepsilon_s = 2\%$.

5.7 Determine a raiz real positiva de $\ln(x^2) = 0,7$ (a) graficamente, (b) utilizando três iterações do método da bissecção, com aproximações iniciais $x_l = 0,5$ e $x_u = 2$, e (c) usando três iterações do método da falsa posição, com as mesmas aproximações iniciais utilizadas em (b).

5.8 A concentração de saturação do oxigênio dissolvido em água fresca pode ser calculada pela equação

$$\ln o_{sf} = -139,34411 + \frac{1,575701 \times 10^5}{T_a} - \frac{6,642308 \times 10^7}{T_a^2} + \frac{1,243800 \times 10^{10}}{T_a^3} - \frac{8,621949 \times 10^{11}}{T_a^4}$$

onde o_{sf} é a concentração de saturação do oxigênio dissolvido em água fresca a 1 atm (mg/L) e T_a é a temperatura absoluta (K). Lembre-se que $T_a = T + 273,15$, onde T é a temperatura (°C). De acordo com essa equação, a saturação diminui com o aumento da temperatura. Para águas naturais típicas em climas temperados, a equação pode ser usada para determinar a variação da concentração de oxigênio de 14,621 mg/L a 0 °C até 6,949 mg/L a 35 °C. Dado um valor da concentração de oxigênio, essa

fórmula e o método da bissecção podem ser usados para determinar a temperatura em °C.

- Se as aproximações iniciais forem tomadas como 0 e 35 °C, quantas iterações da bissecção seriam necessárias para determinar a temperatura até um erro absoluto de 0,05 °C?
- Com base em (a) desenvolva e teste uma função em MATLAB de bissecção para determinar T como uma função de uma dada concentração de oxigênio. Teste sua função para $o_{sf} = 8, 10$ e 14 mg/L . Verifique os resultados.

5.9 Uma viga está sob carga, como mostra a Figura P5.9. Use o método da bissecção para determinar a posição dentro da viga onde não existe nenhum momento.

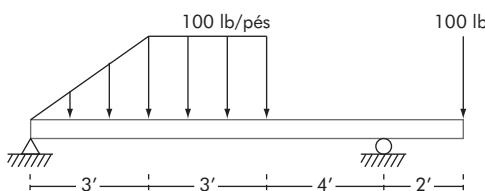


FIGURA P5.9

5.10 Há água escorrendo em um canal trapezoidal a uma vazão de $Q = 20 \text{ m}^3/\text{s}$. A profundidade crítica y para tal canal deve satisfazer a equação

$$0 = 1 - \frac{Q^2}{gA_c^3}B$$

onde $g = 9,81 \text{ m}^2/\text{s}^2$, A_c é a área da seção transversal (m^2) e B é a largura do canal na superfície (m). Para esse caso, a largura e a área transversal podem ser relacionadas à profundidade y por

$$B = 3 + y$$

e

$$A_c = 3y + \frac{y^2}{2}$$

Encontre a profundidade crítica utilizando (a) o método gráfico, (b) a bissecção e (c) a falsa posição. Para (b) e (c) use aproximações iniciais de $x_l = 0,5$ e $x_u = 2,5$ e itere até que o erro aproximado fique abaixo de 1% ou que o número de iterações ultrapasse 10. Discuta os resultados.

5.11 O modelo de Michaelis-Menten descreve a cinética de reações mediadas por enzimas

$$\frac{dS}{dt} = -v_m \frac{S}{k_s + S}$$

onde S é a concentração de substrato (mols/L), v_m é a taxa máxima de absorção (mols/L/d) e k_s é a cons-

tante de meia saturação, que é o nível de substrato no qual a absorção é metade do máximo (mols/L). Se o nível inicial de substrato em $t = 0$ é S_0 , essa equação diferencial pode ser resolvida para

$$S = S_0 - v_m t + k_s \ln(S_0/S)$$

Desenvolva um programa no MATLAB que gere um gráfico de S versus t para o caso em que $S_0 = 8 \text{ mols/L}$, $v_m = 0,7 \text{ mols/L/d}$ e $k_s = 2,5 \text{ mols/L}$.

5.12 Uma reação química reversível



pode ser caracterizada pela relação de equilíbrio

$$K = \frac{c_c}{c_a^2 c_b}$$

onde o símbolo c_i representa a concentração do constituinte i . Considere que definamos uma variável x que representa o número de mols de C que são produzidos. A conservação da massa pode ser usada para reformular a relação de equilíbrio como

$$K = \frac{(c_{c,0} + x)}{(c_{a,0} - 2x)^2(c_{b,0} - x)}$$

onde o subscrito 0 indica a concentração inicial de cada constituinte. Se $K = 0,016$, $c_{a,0} = 42$, $c_{b,0} = 28$ e $c_{c,0} = 4$, determine o valor de x .

- Obtenha a solução graficamente.
- Com base em (a), resolva para determinar a raiz com aproximações iniciais de $x_l = 0$ e $x_u = 20$ até $\varepsilon_s = 0,5\%$. Escolha a bissecção ou a falsa posição para obter a solução. Justifique sua escolha.

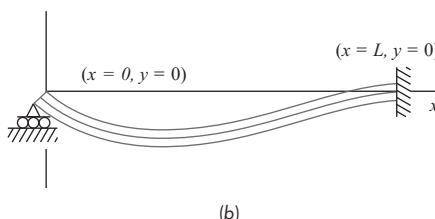
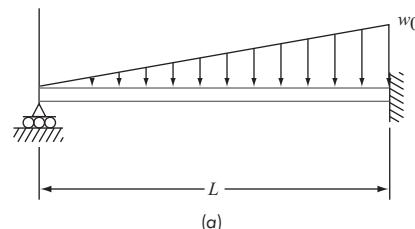


FIGURA P5.13

5.13 A Figura P5.13a mostra uma viga uniforme sujeita a uma carga distribuída de forma linearmente crescente. A equação para a curva elástica resultante é (ver Figura P5.13b)

$$y = \frac{w_0}{120EI}(-x^5 + 2L^2x^3 - L^4x) \quad (\text{P5.13})$$

Use a bissecção para determinar o ponto de deflexão máxima (isto é, o valor de x onde $dy/dx = 0$). A seguir, substitua esse valor na Equação (P5.13) para determinar o valor da deflexão máxima. Use os seguintes valores dos parâmetros nos seus cálculos: $L = 600$ cm, $E = 50.000$ kN/cm², $I = 30.000$ cm⁴ e $w_0 = 2,5$ kN/cm.

5.14 Você comprou um veículo de R\$ 35.000,00 sem entrada e pagando R\$ 8.500,00 por ano por 7 anos. Use a função *bissec* da Figura 5.7 para determinar a taxa de juros que você está pagando. Empregue aproximações iniciais para a taxa de juros de 0,01 e 0,3 e um critério de parada de 0,00005. A fórmula que relaciona o valor atual P , os pagamentos anuais A , o número de anos n e a taxa de juros i é

$$A = P \frac{i(1+i)^n}{(1+i)^n - 1}$$

5.15 Muitos campos da engenharia necessitam de estimativas acuradas da população. Por exemplo, os engenheiros de transporte podem achar necessário determinar separadamente a tendência de crescimento da população de uma cidade e do subúrbio adjacente. A população da área urbana está diminuindo com o tempo de acordo com

$$P_u(t) = P_{u,\max} e^{-k_u t} + P_{u,\min}$$

enquanto a população suburbana está crescendo de acordo com

$$P_s(t) = \frac{P_{s,\max}}{1 + [P_{s,\max}/P_0 - 1]e^{-k_s t}}$$

onde $P_{u,\max}$, k_u , $P_{s,\max}$, P_0 e k_s são parâmetros determinados empiricamente. Determine o tempo e os valores correspondentes de $P_u(t)$ e $P_s(t)$ quando os subúrbios forem 20% maiores que a cidade. Os valores dos parâmetros são $P_{u,\max} = 80.000$ pessoas, $k_u = 0,05/\text{ano}$, $P_{u,\min} = 110.000$ pessoas, $P_{s,\max} = 320.000$ pessoas, $P_0 = 10.000$ pessoas e $k_s = 0,09/\text{ano}$. Para obter suas soluções, use os métodos (a) gráfico e (b) falsa posição.

5.16 A resistividade ρ do silício dopado é baseada na carga q de um elétron, na densidade eletrônica n e na mobilidade do elétron μ . A densidade

eletrônica é dada em termos da densidade da dopagem N e da densidade de transporte intrínseca n_i . A mobilidade do elétron é descrita pela temperatura T , pela temperatura de referência T_0 e pela mobilidade de referência μ_0 . As equações necessárias para o cálculo da resistividade são

$$\rho = \frac{1}{qn\mu}$$

onde

$$n = \frac{1}{2} \left(N + \sqrt{N^2 + 4n_i^2} \right) \text{ e } \mu = \mu_0 \left(\frac{T}{T_0} \right)^{-2,42}$$

Determine N dados $T_0 = 300$ K, $T = 1.000$ K, $\mu_0 = 1.360$ cm² (V s)⁻¹, $q = 1,7 \times 10^{-19}$ C, $n_i = 6,21 \times 10^9$ cm⁻³ e uma resistividade desejada de $\rho = 6,5 \times 10^6$ V s cm/C. Empregue aproximações iniciais de $N = 0$ e $2,5 \times 10^{10}$. Use o método (a) da bissecção e (b) da falsa posição.

5.17 Uma carga total Q está uniformemente distribuída ao redor de um condutor circular de raio a . Uma carga q está localizada a uma distância x do centro do anel (Figura P5.17). A força exercida na carga pelo anel é dada por

$$F = \frac{1}{4\pi e_0} \frac{qQx}{(x^2 + a^2)^{3/2}}$$

onde $e_0 = 8,9 \times 10^{-12}$ C²/(N m²). Encontre a distância x onde a força é 1,25 N se q e Q são 2×10^{-5} C para um anel de raio 0,85 m.

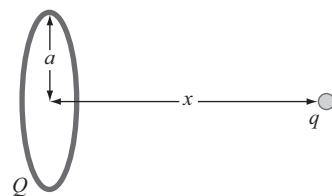


FIGURA P5.17

5.18 Para o escoamento de fluidos em tubos, o atrito é descrito por um número adimensional, o *fator de atrito de Fanning f*. O fator de atrito de Fanning depende de diversos parâmetros relacionados ao tamanho do tubo e ao fluido, os quais podem ser representados por uma outra quantidade adimensional, o *número de Reynolds Re*. Uma fórmula que prevê f dado Re é a *equação de von Karman*:

$$\frac{1}{\sqrt{f}} = 4 \log_{10} \left(Re \sqrt{f} \right) - 0,4$$

Para o escoamento turbulento, os valores típicos do número de Reynolds estão entre 10.000 e 500.000 e

os do fator de atrito de Fanning estão entre 0,001 e 0,01. Desenvolva uma função que use a bissecção para determinar f dado um valor fornecido pelo usuário de Re entre 2.500 e 1.000.000, de forma que ela garanta que o erro absoluto no resultado seja $E_{a,d} < 0,000005$.

5.19 Os engenheiros mecânicos, da mesma forma que a maioria dos outros engenheiros, utilizam a termodinâmica frequentemente nos seus trabalhos. O seguinte polinômio pode ser usado para relacionar o calor específico do ar seco à pressão nula, c_p kJ/(kg K), à temperatura (K):

$$c_p = 0,99403 + 1,671 \times 10^{-4} T + 9,7215 \times 10^{-8} T^2 - 9,5838 \times 10^{-11} T^3 + 1,9520 \times 10^{-14} T^4$$

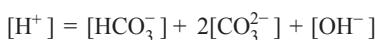
Faça um gráfico de c_p versus T em uma faixa de 0 a 1200 K e, então, use o método da bissecção para determinar a temperatura que corresponde a um calor específico de 1,1 kJ/(kg K).

5.20 A velocidade para cima de um foguete pode ser calculada pela seguinte fórmula:

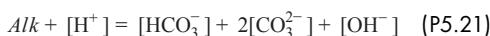
$$v = u \ln \frac{m_0}{m_0 - qt} - gt$$

onde v é a velocidade para cima, u é a velocidade na qual o combustível é expelido em relação ao foguete, m_0 é a massa inicial do foguete no instante $t = 0$, q é a taxa de consumo do combustível, e g é a aceleração da gravidade para baixo (suposta constante e igual a $9,81 \text{ m/s}^2$). Se $u = 1800 \text{ m/s}$, $m_0 = 160.000 \text{ kg}$ e $q = 2600 \text{ kg/s}$, calcule o tempo no qual $v = 750 \text{ m/s}$. (Sugestão: t está em algum ponto entre 10 e 50 s.) Determine seu resultado de modo que ele esteja a menos de 1% do valor verdadeiro. Verifique sua resposta.

5.21 Embora não tenha sido mencionado na Seção 5.6, a Equação (5.13) é uma expressão de *eletroneutralidade* – isto é, as cargas positivas e negativas devem se equilibrar. Isso pode ser visto mais claramente expressando-a como



Em outras palavras, as cargas positivas devem ser iguais às cargas negativas. Assim, quando você calcula o pH de um corpo de água natural, como um lago, você deve também ter em conta outros íons que podem estar presentes. Para o caso em que esses íons se originam de sais não reativos, os valores líquidos (ou totais) de cargas negativas menos cargas positivas devidos a esses íons são agrupados em uma grandeza chamada *alcalinidade*, e a equação é reformulada como



onde Alk é a alcalinidade (eq/L). Por exemplo, a alcalinidade de um dado lago é aproximadamente $0,4 \times 10^{-3}$ eq/L. Efetue os mesmos cálculos como na

Seção 5.6 para calcular o pH desse lago em 2008. Assuma que, assim como as gotas de chuva, o lago está em equilíbrio com o CO_2 atmosférico, porém tenha em conta a alcalinidade como na Equação (P5.21).

5.22 De acordo com o princípio de Arquimedes, a força de *empuxo* é igual ao peso do fluido deslocado por uma porção submersa do objeto. Para a esfera apresentada na Figura P5.22, use a bissecção para determinar a altura, h , da porção que está acima da água. Empregue os seguintes valores para seus cálculos: $r = 1 \text{ m}$, ρ_s = densidade da esfera = 200 kg/m^3 e ρ_w = densidade da água = 1.000 kg/m^3 . Observe que o volume da porção da esfera acima da água pode ser calculado com

$$V = \frac{\pi h^2}{3}(3r - h)$$

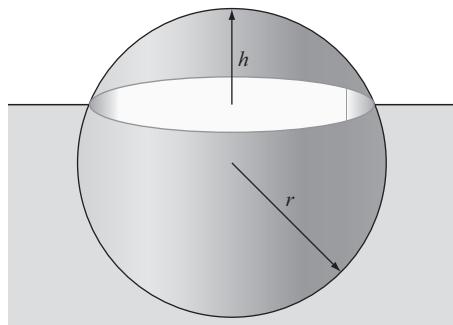


FIGURA P5.22

5.23 Efetue os mesmos cálculos como no Problema 5.22, porém para o *tronco* de um cone como ilustrado na Figura P5.23. Empregue os seguintes valores para seus cálculos: $r_1 = 0,5 \text{ m}$, $r_2 = 1 \text{ m}$, $h = 1 \text{ m}$, ρ_f = densidade do tronco = 200 kg/m^3 e ρ_w = densidade da água = 1.000 kg/m^3 . Observe que o volume de um tronco é dado por

$$V = \frac{\pi h}{3}(r_1^2 + r_2^2 + r_1 r_2)$$

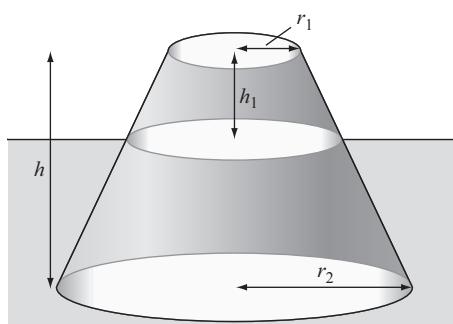


FIGURA P5.23

6

Raízes: métodos abertos

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é familiarizar o leitor com os métodos abertos para encontrar a raiz de uma única equação não linear. Após esta leitura completa, você será capaz de:

- ✓ Reconhecer a diferença entre os métodos intervalares e os métodos abertos para localização de raiz.
- ✓ Compreender o método de iteração de ponto fixo e como se pode avaliar suas características de convergência.
- ✓ Aprender como resolver um problema de raízes com o método de Newton-Raphson e apreciar o conceito de convergência quadrática.
- ✓ Saber como implementar os métodos da secante e da secante modificada.
- ✓ Entender como o método de Brent combina os métodos intervalares confiáveis com os métodos abertos rápidos para localizar raízes de uma maneira robusta e eficiente.
- ✓ Aprender como usar a função `fzero` do MATLAB para estimar raízes.
- ✓ Aprender como manipular e determinar raízes de polinômios com o MATLAB.

Nos métodos intervalares do Capítulo 5, a raiz era localizada dentro de um intervalo prescrito por um limitante inferior e um superior. As aplicações repetidas desses métodos sempre resultam em estimativas mais próximas do valor verdadeiro da raiz. Tais métodos são ditos *convergentes* porque se aproximam da verdade à medida que os cálculos prosseguem (Figura 6.1a).

Já os *métodos abertos* descritos neste capítulo são baseados em fórmulas que exigem apenas um único valor inicial de x ou dois valores iniciais que não delimitam necessariamente a raiz. Como tal, algumas vezes *divergem* ou se afastam da raiz verdadeira à medida que os cálculos prosseguem (Figura 6.1b). Entretanto, quando os métodos abertos convergem (Figura 6.1c), em geral o fazem muito mais rapidamente do que os métodos intervalares. Iniciaremos nossa discussão sobre as técnicas abertas com uma versão simples, que será útil para ilustrar sua forma geral e também o conceito de convergência.

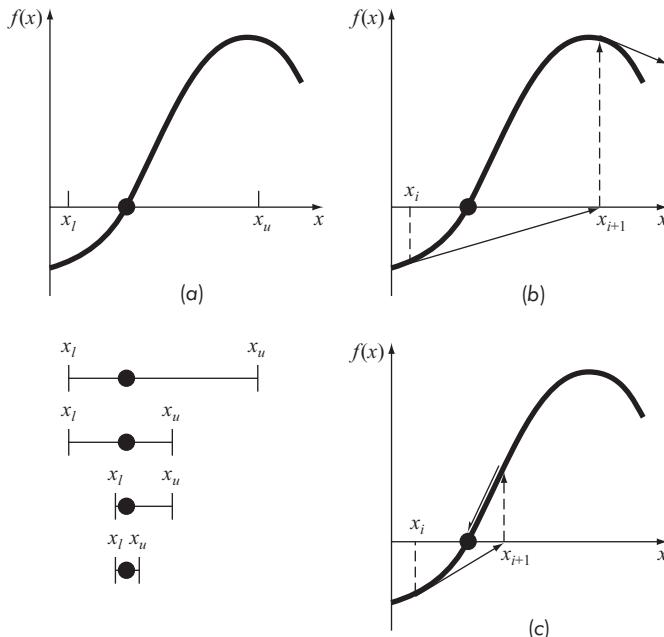


FIGURA 6.1 Descrição gráfica da diferença fundamental entre os métodos (a) intervalares e (b) e (c) abertos para a localização de raízes. Em (a), que é o método da bissecção, a raiz está restrita ao interior do intervalo definido por x_l e x_u . Já para o método aberto descrito em (b) e (c), é usada uma fórmula para avançar de x_i a x_{i+1} de forma iterativa. Assim, o método pode ou (b) divergir ou (c) convergir rapidamente, dependendo da forma da função e do valor da aproximação inicial.

6.1 ITERAÇÃO DE PONTO FIXO SIMPLES

Como já mencionado, os métodos abertos usam uma fórmula para prever a raiz. Tal fórmula pode ser deduzida para a *iteração de ponto fixo* (ou, como também é chamada, *iteração de um ponto, substituições sucessivas ou aproximações sucessivas*) reescrevendo a função $f(x) = 0$ de modo que x esteja isolado no lado esquerdo da equação:

$$x = g(x) \quad (6.1)$$

Pode-se conseguir essa transformação por manipulação algébrica ou simplesmente somando x em ambos os lados da equação original.

A Equação (6.1) é útil porque fornece uma fórmula para prever um novo valor de x em função de um velho valor de x . Portanto, dada uma aproximação inicial para a raiz x_i , a Equação (6.1) pode ser usada para calcular uma nova estimativa x_{i+1} expressa pela fórmula iterativa:

$$x_{i+1} = g(x_i) \quad (6.2)$$

Como nas outras fórmulas iterativas deste livro, o erro aproximado para essa equação pode ser determinado usando-se o estimador de erro:

$$\varepsilon_a = \left| \frac{x_{i+1} - x_i}{x_{i+1}} \right| 100\% \quad (6.3)$$

EXEMPLO 6.1 Iteração de ponto fixo simples

Use a iteração de ponto fixo simples para localizar a raiz de $f(x) = e^{-x} - x$.

Solução. A função pode ser separada diretamente e expressa na forma da Equação (6.2) como

$$x_{i+1} = e^{-x_i}$$

Começando com uma aproximação inicial $x_0 = 0$, essa equação iterativa pode ser aplicada para calcular:

i	x_i	$ \varepsilon_a (\%)$	$ \varepsilon_i (\%)$	$ \varepsilon_i / \varepsilon_{i-1} $
0	0,0000		100,000	
1	1,0000	100,000	76,322	0,763
2	0,3679	171,828	35,135	0,460
3	0,6922	46,854	22,050	0,628
4	0,5005	38,309	11,755	0,533
5	0,6062	17,447	6,894	0,586
6	0,5454	11,157	3,835	0,556
7	0,5796	5,903	2,199	0,573
8	0,5601	3,481	1,239	0,564
9	0,5711	1,931	0,705	0,569
10	0,5649	1,109	0,399	0,566

Assim, cada iteração traz o valor estimado para mais perto do valor verdadeiro: 0,56714329.

Observe que o erro relativo percentual verdadeiro para cada iteração do Exemplo 6.1 é aproximadamente proporcional (para esse caso, por um fator de cerca de 0,5 a 0,6) ao erro da iteração anterior. Essa propriedade, chamada de *convergência linear*, é característica da iteração de ponto fixo.

Além de comentarmos sobre a “taxa” de convergência, nesse ponto devemos comentar sobre a “possibilidade” de convergência. Os conceitos de convergência e divergência podem ser descritos graficamente. Lembre-se de que na Seção 5.2 tratamos o gráfico de uma função para visualizar sua estrutura e seu comportamento. Tal abordagem é usada na Figura 6.2a para a função $f(x) = e^{-x} - x$. Uma abordagem gráfica alternativa é separar a equação em duas componentes, como em

$$f_1(x) = f_2(x)$$

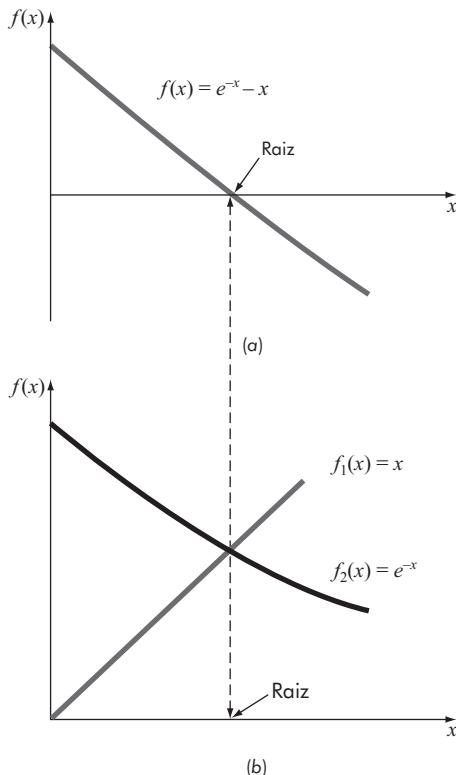


FIGURA 6.2 Dois métodos gráficos alternativos para determinar as raízes de $f(x) = e^{-x} - x$.
(a) Raiz em um ponto onde o gráfico cruza o eixo x ; (b) raiz na interseção das funções componentes.

Então, as duas equações

$$y_1 = f_1(x) \quad (6.4)$$

e

$$y_2 = f_2(x) \quad (6.5)$$

podem ser traçadas separadamente (Figura 6.2b). Os valores de x que correspondem às interseções dessas funções representam as raízes de $f(x) = 0$.

O método das duas curvas pode agora ser usado para ilustrar a convergência e a divergência de ponto fixo. A Equação (6.1) pode ser reescrita como um par de equações $y_1 = x$ e $y_2 = g(x)$, que podem então ser traçadas separadamente. Como ocorreu com as Equações (6.4) e (6.5), as raízes de $f(x) = 0$ correspondem aos valores da abscissa na interseção das duas curvas. A função $y_1 = x$ e quatro formas diferentes para $y_2 = g(x)$ são mostradas na Figura 6.3.

No primeiro caso (Figura 6.3a), a aproximação inicial x_0 é usada para determinar o ponto correspondente na curva $y_2[x_0, g(x_0)]$. O ponto $[x_1, y_1]$ é localizado movendo-se horizontalmente para a esquerda até a curva y_1 . Esses movimentos são equivalentes à primeira iteração no método do ponto fixo:

$$x_1 = g(x_0)$$

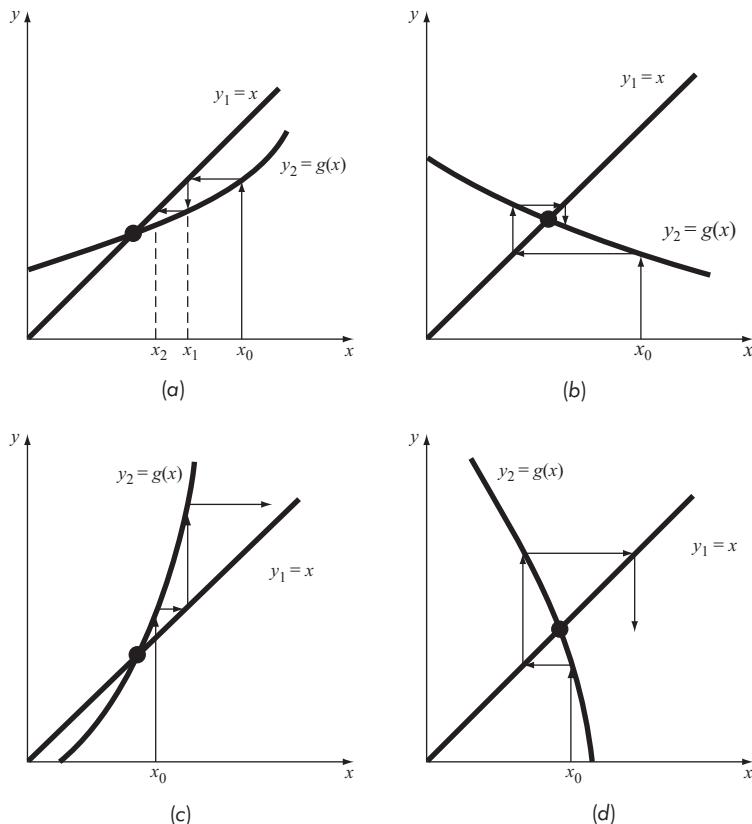


FIGURA 6.3 Descrição gráfica da convergência (a) e (b) e divergência (c) e (d) da iteração de ponto fixo simples. Os gráficos (a) e (c) são chamados padrões monótonos, enquanto (b) e (d) são chamados padrões oscilantes ou espirais. Observe que a convergência ocorre quando $|g'(x)| < 1$.

Portanto, em ambas as equações e no gráfico, um valor inicial x_0 é usado para se obter uma estimativa de x_1 . A próxima iteração consiste na movimentação para $[x_1, g(x_1)]$ e então para $[x_2, y_2]$. Essa iteração é equivalente à equação

$$x_2 = g(x_1)$$

A solução na Figura 6.3a é *convergente* porque as estimativas de x se aproximam da raiz a cada iteração. O mesmo é verdadeiro para a Figura 6.3b. Entretanto, esse não é o caso para as Figuras 6.3c e d, nas quais as iterações divergem da raiz.

Uma dedução teórica pode ser usada para compreender o processo de forma intuitiva. Como descrito em Chapra e Canale (2010), pode-se verificar que o erro para qualquer iteração é linearmente proporcional ao erro da iteração anterior multiplicado pelo valor absoluto da inclinação de g :

$$E_{i+1} = g'(\xi) E_i$$

Consequentemente, se $|g'| < 1$, os erros diminuem a cada iteração. Para $|g'| > 1$ os erros crescem. Observe também que, se a derivada for positiva, os erros serão

positivos e, portanto, terão o mesmo sinal (Figuras 6.3a e c). Se a derivada for negativa, os erros mudarão o sinal em cada iteração (Figuras 6.3b e d).

6.2 O MÉTODO DE NEWTON-RAPHSON

Talvez a fórmula mais utilizada para localizar uma raiz seja a o *método de Newton-Raphson* (Figura 6.4). Se a aproximação inicial da raiz for x_i , pode-se estender uma reta tangente a partir do ponto $[x_i, f(x_i)]$. O ponto onde essa tangente cruza o eixo x normalmente representa uma estimativa melhorada da raiz.

O método de Newton-Raphson pode ser deduzido com base em sua interpretação geométrica. Como na Figura 6.4, a primeira derivada em x é equivalente à inclinação:

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}}$$

que pode ser reorganizada para fornecer

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (6.6)$$

que é chamada de *fórmula de Newton-Raphson*.

EXEMPLO 6.2 Método de Newton-Raphson

Use o método de Newton-Raphson para estimar a raiz de $f(x) = e^{-x} - x$, utilizando uma aproximação inicial de $x_0 = 0$.

Solução. A primeira derivada da função pode ser calculada como

$$f'(x) = -e^{-x} - 1$$

que pode ser substituída junto com a função original na Equação (6.6) para fornecer

$$x_{i+1} = x_i - \frac{e^{-x_i} - x_i}{-e^{-x_i} - 1}$$

Começando com uma aproximação inicial de $x_0 = 0$, essa equação iterativa pode ser aplicada para calcular

i	x_i	$ \varepsilon_i , \%$
0	0	100
1	0,500000000	11,8
2	0,566311003	0,147
3	0,567143165	0,0000220
4	0,567143290	<10 ⁻⁸

Assim, a aproximação converge rapidamente para a raiz verdadeira. Observe que o erro relativo percentual verdadeiro em cada iteração diminui muito mais rapidamente do que no método da iteração de ponto fixo simples (compare com o Exemplo 6.1).

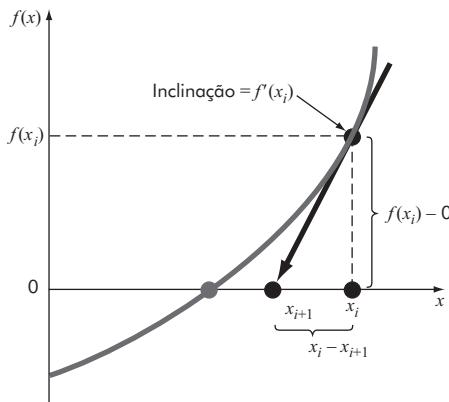


FIGURA 6.4 Descrição gráfica do método de Newton-Raphson. A tangente à função em x_i [isto é, $f'(x)$] é prolongada até o eixo x para fornecer uma estimativa da raiz em x_{i+1} .

Assim como outros métodos de localização de raízes, a Equação (6.3) pode ser usada como critério de parada. Além disso, uma análise teórica (Chapra e Canale, 2010) fornece uma visão sobre a taxa de convergência, como expresso por

$$E_{t,i+1} = \frac{-f''(x_r)}{2f'(x_r)} E_{t,i}^2 \quad (6.7)$$

Dessa forma, o erro deve ser aproximadamente proporcional ao quadrado do erro anterior, ou seja, o número de algarismos significativos de acurácia dobra a cada iteração. Esse comportamento é chamado de *convergência quadrática* e é uma das principais razões para a popularidade do método.

Embora o método de Newton-Raphson seja em geral muito eficiente, há situações nas quais tem um desempenho insatisfatório. Um caso especial – raízes múltiplas – é discutido na referência Chapra e Canale (2010). Entretanto, mesmo no tratamento de raízes simples podem aparecer dificuldades, como no próximo exemplo.

EXEMPLO 6.3 Exemplo de uma função que converge lentamente com Newton-Raphson

Determine a raiz positiva de $f(x) = x^{10} - 1$ utilizando o método de Newton-Raphson e uma aproximação inicial de $x = 0,5$.

Solução. A fórmula de Newton-Raphson para esse caso é

$$x_{i+1} = x_i - \frac{x_i^{10} - 1}{10x_i^9}$$

que pode ser usada para calcular

i	x_i	$ \varepsilon_a , \%$
0	0,5	
1	51,65	99,032
2	46,485	11,111
3	41,8365	11,111
4	37,65285	11,111
•		
•		
•		
40	1,002316	2,130
41	1,000024	0,229
42	1	0,002

Assim, depois da primeira previsão fraca, a técnica está convergindo para a raiz verdadeira de 1, mas a uma taxa muito lenta.

Por que isso acontece? Como mostra a Figura 6.5, um gráfico simples das primeiras iterações pode fornecer uma visão do problema. Observe como a primeira aproximação está em uma região onde a inclinação é próxima de zero. Dessa forma, a primeira iteração afasta a solução da aproximação inicial e a lança para um novo valor ($x = 51,65$) onde $f(x)$ tem um valor extremamente elevado. A solução, então, caminha com dificuldade por mais de 40 iterações até convergir para a raiz com a acurácia adequada.

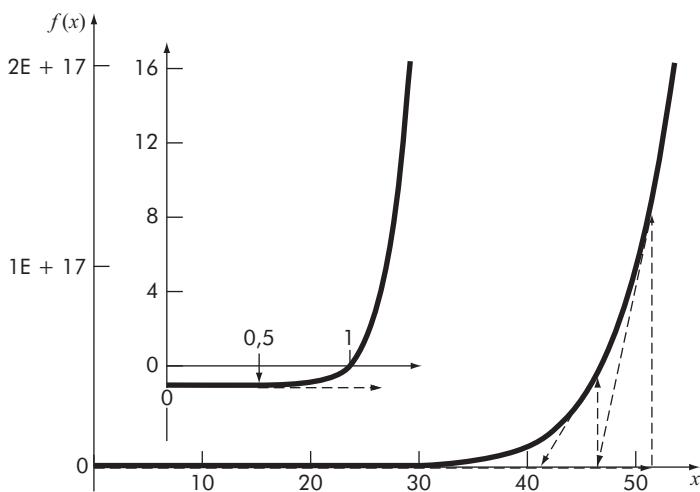


FIGURA 6.5 Descrição gráfica do método de Newton-Raphson para um caso com convergência lenta. A seta na aproximação inicial ($x_0 = 0,5$) mostra como uma inclinação próxima de zero inicialmente joga a solução para longe da raiz. Depois disso, a solução converge muito lentamente para a raiz.

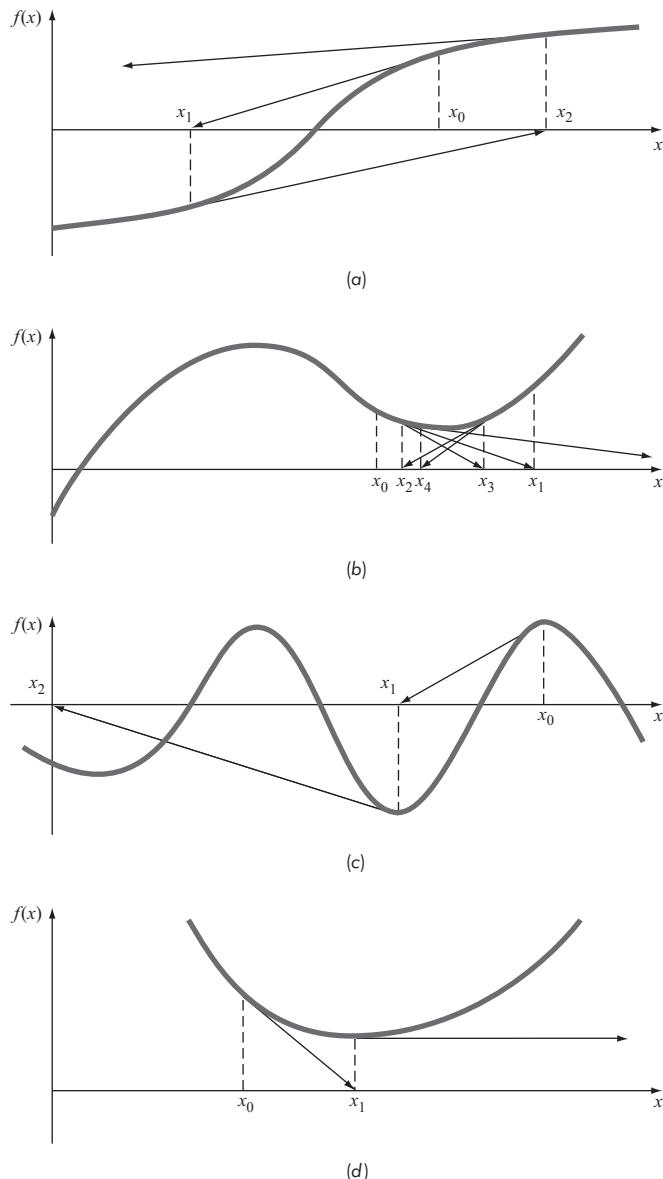


FIGURA 6.6 Quatro casos nos quais o método de Newton-Raphson exibe convergência insatisfatória.

Além da convergência lenta em razão da natureza da função, outras dificuldades podem aparecer, como ilustra a Figura 6.6. Por exemplo, a Figura 6.6a descreve o caso no qual ocorre um ponto de inflexão [isto é, $f''(x) = 0$] na vizinhança de uma raiz. Observe que as iterações que começam em x_0 se afastam progressivamente da raiz. A Figura 6.6b ilustra a tendência da técnica de Newton-Raphson em oscilar em torno de uma posição de máximo ou de mínimo. Tais oscilações podem persistir, ou, como na Figura 6.6b, atinge-se uma inclinação próxima a zero, em que a solução é

afastada da área de interesse. A Figura 6.6c mostra como uma aproximação inicial que esteja perto de uma raiz pode pular para uma posição longe por diversas raízes. Essa tendência de se afastar da área de interesse ocorre porque foram encontradas inclinações próximas de zero. Obviamente, uma inclinação nula [$f'(x) = 0$] é, na verdade, um desastre, porque provoca divisão por zero na fórmula de Newton-Raphson [Equação (6.6)]. Como ilustrado na Figura 6.6d, isso significa que a solução dispara horizontalmente e nunca atinge o eixo x .

Logo, não existe nenhum critério de convergência geral para o Newton-Raphson. Sua convergência depende da natureza da função e da precisão da aproximação inicial. A única solução é ter uma aproximação inicial que esteja “suficientemente” perto da raiz. E, para algumas funções, nenhuma aproximação funcionará. Boas aproximações dependem em geral do conhecimento das condições do problema físico ou de artifícios como gráficos que forneçam informações sobre o comportamento da solução. A falta de um critério de convergência geral também sugere que um bom software deveria ser projetado para reconhecer convergência lenta ou divergência.

6.2.1 Função do MATLAB: newtraph

Um algoritmo para o método de Newton-Raphson pode ser facilmente desenvolvido (Figura 6.7). Observe que o programa deve ter acesso à função (`func`) e à sua primeira derivada (`dfunc`). Isso pode ser feito simplesmente pela inclusão de funções personalizadas para calcular essas quantidades. De forma alternativa, como no algoritmo da Figura 6.7, os dois parâmetros citados (função e sua derivada) podem ser passados para a função como argumentos de entrada.

Depois de digitado e salvo o código da função, ela pode ser usada para calcular raízes. Por exemplo, para a função simples $x^2 - 9$, a raiz pode ser determinada como em

```
>> newtraph(@(x) x^2-9,@(x) 2*x,5)
ans =
    3
```

EXEMPLO 6.4 O método de Newton-Raphson para o problema do saltador de *bungee jumping*

Use a função do MATLAB da Figura 6.7 para determinar a massa do saltador de *bungee jumping* com um coeficiente de arraste de 0,25 kg/m para atingir uma velocidade de 36 m/s após 4 s de queda livre. A aceleração da gravidade é 9,81 m/s².

Solução. A função a ser avaliada é

$$f(m) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}t\right) - v(t) \quad (\text{E6.4.1})$$

Para aplicar o método de Newton-Raphson, a derivada dessa função deve ser determinada com relação à incógnita, m :

$$\frac{df(m)}{dm} = \frac{1}{2}\sqrt{\frac{g}{mc_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}t\right) - \frac{g}{2m}t \operatorname{sech}^2\left(\sqrt{\frac{gc_d}{m}}t\right) \quad (\text{E6.4.2})$$

É importante mencionar que, embora essa derivada não seja, em princípio, difícil de calcular, envolve um pouco de concentração e esforço para chegar ao resultado final.

As duas fórmulas podem agora ser usadas em conjunto com a função newtraph para determinar a raiz

```
>> y = @(m) sqrt(9.81*m/0.25)*tanh(sqrt(9.81*0.25/m)*4)-36;
>> dy = @(m) 1/2*sqrt(9.81/(m*0.25))*tanh((9.81*0.25/m)...
    ^^(1/2)*4)-9.81/(2*m)*sech(sqrt(9.81*0.25/m)*4)^2;
>> newtraph(y,dy,140,0.00001)

ans =
142.7376
```

```
function [raiz,ea,iter]=newtraph(func,dfunc,xr,es,maxit,varargin)
% newtraph: localização de raízes pelo método de Newton-Raphson
% [raiz,ea,iter]=newtraph(func,dfunc,xr,es,maxit,p1,p2,...):
% usa o método de Newton-Raphson para encontrar a raiz de func
% entrada:
%   func = nome da função
%   dfunc = nome da derivada da função
%   xr = aproximação inicial
%   es = erro relativo desejado (padrão = 0,0001%)
%   maxit = número máximo de iterações permitidas (padrão = 50)
%   p1,p2,... = parâmetros adicionais usados por func
% saída:
%   raiz = raiz real
%   ea = erro relativo aproximado (%)
%   iter = número de iterações

if nargin<3,error('são necessários pelo menos 3 argumentos de
entrada'),end
if nargin<4|isempty(es),es=0.0001;end
if nargin<5|isempty(maxit),maxit=50;end
iter = 0;
while (1)
    x_velho = xr;
    xr = xr - func(xr)/dfunc(xr);
    iter = iter + 1;
    if xr ~= 0, ea = abs((xr - x_velho)/xr) * 100; end
    if ea <= es | iter >= maxit, break, end
end
raiz = xr;
```

FIGURA 6.7 Função do MATLAB para implementar o método de Newton-Raphson.

6.3 O MÉTODO DA SECANTE

Como no Exemplo 6.4, um problema em potencial na implementação do método de Newton-Raphson é o cálculo de sua derivada. Embora isso não seja inconveniente

para polinômios e muitas outras funções, há certas funções cujas derivadas podem ser de cálculos extremamente difícil ou inconveniente. Para esses casos, a derivada pode ser aproximada por uma diferença dividida regressiva:

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

Essa aproximação pode ser substituída na Equação (6.6) para fornecer a seguinte equação iterativa:

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)} \quad (6.8)$$

A Equação (6.8) é a fórmula para o *método da secante*. Observe que a abordagem exige duas estimativas iniciais de x . No entanto, como não é exigido que $f(x)$ mude de sinal entre as estimativas, ele não é classificado como um método intervalar.

Em vez de usar dois valores arbitrários para estimar a derivada, uma abordagem alternativa envolve uma pequena perturbação da variável independente para estimar $f'(x)$,

$$f'(x_i) \cong \frac{f(x_i + \delta x_i) - f(x_i)}{\delta x_i}$$

onde δ é uma pequena fração de perturbação. Essa aproximação pode ser substituída na Equação (6.6) para fornecer a seguinte equação iterativa:

$$x_{i+1} = x_i - \frac{\delta x_i f(x_i)}{f(x_i + \delta x_i) - f(x_i)} \quad (6.9)$$

Essa é a fórmula do chamado *método da secante modificado*. Como mostrado no exemplo a seguir, ela fornece um meio interessante de alcançar a eficiência do Newton-Raphson sem ter de calcular a derivada.

EXEMPLO 6.5 O método da secante modificado

Use o método da secante modificado para determinar a massa do saltador de *bungee jumping* com um coeficiente de arraste de 0,25 kg/m a fim de atingir uma velocidade de 36 m/s após 4 s de queda livre. A aceleração da gravidade é 9,81 m/s². Use uma aproximação inicial de 50 kg e um valor de 10⁻⁶ para a fração de perturbação.

Solução. Inserindo os parâmetros na Equação (6.9), temos

Primeira iteração:

$$x_0 = 50 \qquad f(x_0) = -4,57938708$$

$$x_0 + \delta x_0 = 50,00005 \qquad f(x_0 + \delta x_0) = -4,579381118$$

$$\begin{aligned} x_1 &= 50 - \frac{10^{-6}(50)(-4,57938708)}{-4,579381118 - (-4,57938708)} \\ &= 88,39931(|\varepsilon_t| = 38,1\%; |\varepsilon_a| = 43,4\%) \end{aligned}$$

Segunda iteração:

$$\begin{aligned}x_1 &= 88,39931 & f(x_1) &= -1,69220771 \\x_1 + \delta x_1 &= 88,39940 & f(x_1 + \delta x_1) &= -1,692203516 \\x_2 &= 88,39931 - \frac{10^{-6}(88,39931)(-1,69220771)}{-1,692203516 - (-1,69220771)} \\&= 124,08970 (|\varepsilon_t| = 13,1\%; |\varepsilon_a| = 28,76\%) \end{aligned}$$

Continuando os cálculos, os seguintes valores são obtidos:

i	x_i	$ \varepsilon_t , \%$	$ \varepsilon_a , \%$
0	50,0000	64,971	
1	88,3993	38,069	43,438
2	124,0897	13,064	28,762
3	140,5417	1,538	11,706
4	142,7072	0,021	1,517
5	142,7376	$4,1 \times 10^{-6}$	0,021
6	142,7376	$3,4 \times 10^{-12}$	$4,1 \times 10^{-6}$

A escolha de um valor adequado para δ não é automática. Se δ for muito pequeno, o método pode ser sobrecarregado de erros de arredondamento causados pelo cancelamento da subtração no denominador da Equação (6.9). Se for grande demais, a técnica pode tornar-se ineficiente e mesmo divergente. Entretanto, se escolhido de forma correta, ele fornece uma boa alternativa para os casos nos quais o cálculo da derivada é difícil e encontrar duas aproximações é inconveniente.

Além disso, em seu sentido mais geral, uma função de uma única variável é simplesmente uma entidade que retorna um único valor em troca de valores que lhe foram enviados. Considerando essa visão, as funções não são sempre fórmulas simples como as equações de uma única linha resolvidas nos exemplos anteriores neste capítulo. Por exemplo, uma função pode consistir em várias linhas de código que podem demandar uma quantidade significativa de tempo de execução para serem avaliadas, e pode, em alguns casos, representar um programa de computador independente; para esses casos, os métodos da secante e da secante modificada são de grande valor.

6.4 MÉTODO DE BRENT

Não seria interessante ter uma abordagem híbrida, que combinasse a confiabilidade dos métodos intervalares com a velocidade dos métodos abertos? O *método de Brent para localização de raízes* é um algoritmo inteligente que faz isso pela aplicação de um método aberto rápido sempre que possível, porém revertendo para um método intervalar confiável se necessário. Essa abordagem foi desenvolvida por Richard Brent (1973) com base em um algoritmo anterior de Theodorus Dekker (1969).

A técnica intervalar é o método da bissecção, enquanto dois diferentes métodos abertos são empregados. O primeiro é o método da secante, descrito na Seção 6.3; o segundo é a interpolação quadrática inversa, como será explicado a seguir.

6.4.1 Interpolação quadrática inversa

A *interpolação quadrática inversa* é similar, em essência, ao método da secante. Como ilustra a Figura 6.8a, o método da secante é baseado no cálculo de uma reta que passa através de duas aproximações. A interseção dessa reta com o eixo x representa a nova estimativa da raiz; por essa razão, ele é algumas vezes chamado de *método da interpolação linear*.

Considere que haja agora três pontos. Nesse caso, poderíamos determinar uma função quadrática de x que passa pelos três pontos (Figura 6.8b). Assim como no método da secante linear, a interseção dessa parábola com o eixo x representaria a nova estimativa da raiz. E, como ilustra a Figura 6.8b, a utilização de uma curva em vez de uma reta geralmente produz uma estimativa melhor.

Embora isso pareça representar uma grande melhoria, a abordagem tem uma falha fundamental: é possível que a parábola não intercepte o eixo x , como, por exemplo, quando a parábola resultante tem raízes complexas. Essa situação é ilustrada pela parábola, $y = f(x)$, na Figura 6.9.

A dificuldade pode ser contornada empregando interpolação quadrática inversa. Isto é, em vez de utilizar uma parábola em x , os pontos podem ser ajustados com uma parábola em y . Isso equivale a inverter os eixos e criar uma parábola “lateral” [a curva $x = f(y)$, na Figura 6.9].

Se os três pontos são designados como (x_{i-2}, y_{i-2}) , (x_{i-1}, y_{i-1}) e (x_i, y_i) , uma função quadrática de y que passa pelos pontos pode ser gerada como

$$g(y) = \frac{(y - y_{i-1})(y - y_i)}{(y_{i-2} - y_{i-1})(y_{i-2} - y_i)}x_{i-2} + \frac{(y - y_{i-2})(y - y_i)}{(y_{i-1} - y_{i-2})(y_{i-1} - y_i)}x_{i-1} + \frac{(y - y_{i-2})(y - y_{i-1})}{(y_i - y_{i-2})(y_i - y_{i-1})}x_i \quad (6.10)$$

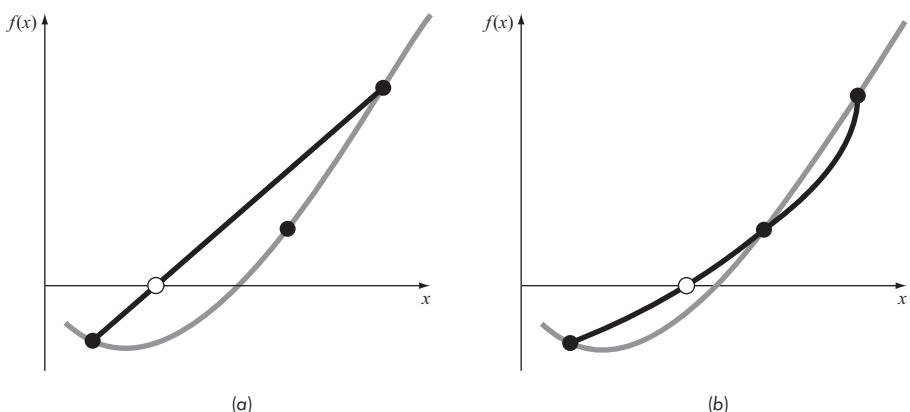


FIGURA 6.8 Comparação entre (a) o método da secante e (b) a interpolação quadrática inversa. Observe que a abordagem em (b) é chamada “inversa” porque a função quadrática é escrita em y em vez de em x .

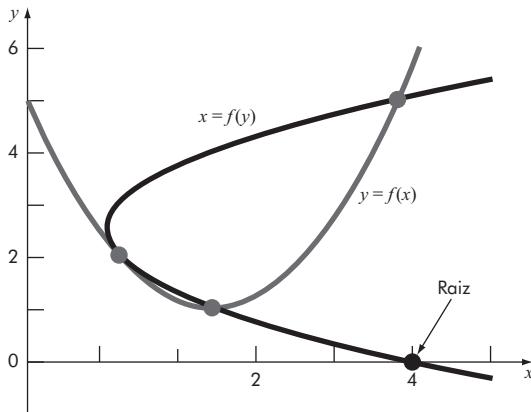


FIGURA 6.9 Duas parábolas ajustadas a três pontos. A parábola escrita como uma função de x , $y = f(x)$ tem raízes complexas e, portanto, não intercepta o eixo x . Já se as variáveis são invertidas, e a parábola deduzida como $x = f(y)$, a função intercepta o eixo x .

Como será detalhado na Seção 18.2, essa forma é chamada de *polinômio de Lagrange*. A raiz, x_{i+1} , corresponde a $y = 0$, que, quando substituída na Equação (6.10), fornece

$$\begin{aligned} x_{i+1} = & \frac{y_{i-1}y_i}{(y_{i-2} - y_{i-1})(y_{i-2} - y_i)}x_{i-2} + \frac{y_{i-2}y_i}{(y_{i-1} - y_{i-2})(y_{i-1} - y_i)}x_{i-1} \\ & + \frac{y_{i-2}y_{i-1}}{(y_i - y_{i-2})(y_i - y_{i-1})}x_i \end{aligned} \quad (6.11)$$

Como mostrado na Figura 6.9, essa parábola “lateral” sempre intercepta o eixo x .

EXEMPLO 6.6 Interpolação quadrática inversa

Desenvolva equações quadráticas em x e em y para os seguintes pontos representados na Figura 6.9: $(1, 2)$, $(2, 1)$ e $(4, 5)$. Para a primeira, $y = f(x)$, utilize a fórmula quadrática para ilustrar que as raízes são complexas. Para a segunda, $x = g(y)$, use a interpolação quadrática inversa [Equação (6.11)] para determinar a estimativa da raiz.

Solução. Pela inversão de x 's e y 's, a Equação (6.10) pode ser usada para gerar uma função quadrática em x , como

$$f(x) = \frac{(x-2)(x-4)}{(1-2)(1-4)}2 + \frac{(x-1)(x-4)}{(2-1)(2-4)}1 + \frac{(x-1)(x-2)}{(4-1)(4-2)}5$$

ou, agrupando os termos comuns,

$$f(x) = x^2 - 4x + 5$$

Essa equação foi usada para gerar a parábola $y = f(x)$ (Figura 6.9). A fórmula quadrática pode ser usada para verificar que as raízes para esse caso são complexas

$$x = \frac{4 \pm \sqrt{(-4)^2 - 4(1)(5)}}{2} = 2 \pm i$$

A Equação (6.10) pode ser usada para gerar a função quadrática em y , como

$$g(y) = \frac{(y-1)(y-5)}{(2-1)(2-5)}1 + \frac{(y-2)(y-5)}{(1-2)(1-5)}2 + \frac{(y-2)(y-1)}{(5-2)(5-1)}4$$

ou, agrupando os termos comuns,

$$g(y) = 0,5x^2 - 2,5x + 4$$

Por fim, a Equação (6.11) pode ser usada para determinar as raízes, como

$$x_{i+1} = \frac{-1(-5)}{(2-1)(2-5)}1 + \frac{-2(-5)}{(1-2)(1-5)}2 + \frac{-2(-1)}{(5-2)(5-1)}4 = 4$$

Antes de avançar para o algoritmo de Brent, é importante mencionar mais um caso em que a interpolação quadrática inversa não funciona. Se os três valores de y não são distintos (isto é, $y_{i-2} = y_{i-1}$ ou $y_{i-1} = y_i$), uma função quadrática inversa não existe; nesses casos, portanto, utiliza-se o método da secante. Assim, se chegarmos a uma situação em que os valores de y não são distintos, é sempre possível reverter para o método da secante, menos eficiente, para gerar uma raiz utilizando dois dos pontos. Se $y_{i-2} = y_{i-1}$, utiliza-se o método da secante com x_{i-1} e x_i . Se $y_{i-1} = y_i$, utiliza-se x_{i-2} e x_{i-1} .

6.4.2 Algoritmo do método de Brent

A ideia geral por trás do *método de Brent para localização de raízes* é usar, sempre que possível, um dos métodos abertos rápidos. Quando esses métodos produzem um resultado inaceitável (isto é, uma estimativa da raiz que saia do intervalo), o algoritmo reverte para o método da bissecção, mais conservativo. Embora a bissecção possa ser mais lenta, gera uma estimativa que certamente cai dentro do intervalo. Esse processo é, então, repetido até que a raiz seja localizada dentro de uma tolerância aceitável. Como poderia ser esperado, a bissecção costuma dominar o processo no início, mas, à medida que se aproxima da raiz, a técnica muda para os métodos abertos mais rápidos.

A Figura 6.10 apresenta uma função com base no programa de MATLAB desenvolvido por Cleve Moler (2004), a qual representa uma versão simplificada da função `fzero`, que é a função de localização de raízes utilizada internamente no MATLAB. Por essa razão, chamamos essa versão simplificada de `fzerosimp`. Observe que ela requer outra função `f` que contém a equação para a qual a raiz está sendo avaliada.

A função `fzerosimp` possui dois argumentos de entrada que correspondem às duas aproximações iniciais que devem delimitar a raiz. Uma vez passados esses argumentos, as três variáveis que definem o intervalo de busca (`a`, `b`, `c`) são inicializadas e `f` é calculada nas extremidades.

```

function b = fzerosimp(xl,xu)
a = xl; b = xu; fa = f(a); fb = f(b);
c = a; fc = fa; d = b - c; e = d;
while (1)
    if fb == 0, break, end
    if sign(fa) == sign(fb) %Se necessário, os pontos são rearranjados
        a = c; fa = fc; d = b - c; e = d;
    end
    if abs(fa) < abs(fb)
        c = b; b = a; a = c;
        fc = fb; fb = fa; fa = fc;
    end
    m = 0.5*(a - b); %Teste de finalização e possível saída
    tol = 2 * eps * max(abs(b), 1);
    if abs(m) <= tol | fb == 0.
        break
    end
    %Escolhe entre os métodos abertos e a bissecção
    if abs(e) >= tol & abs(fc) > abs(fa)
        s = fb/fc;
        if a == c %Método da secante
            p = 2*m*s;
            q = 1 - s;
        else %Interpolação quadrática inversa
            q = fc/fa; r = fb/fa;
            p = s * (2*m*q * (q - r) - (b - c)*(r - 1));
            q = (q - 1)*(r - 1)*(s - 1);
        end
        if p > 0, q = -q; else p = -p; end;
        if 2*p < 3*m*q - abs(tol*q) & p < abs(0.5*e*q)
            e = d; d = p/q;
        else
            d = m; e = m;
        end
    else %Bissecção
        d = m; e = m;
    end
    c = b; fc = fb;
    if abs(d) > tol, b=b+d; else b=b-sign(b-a)*tol; end
    fb = f(b);
end

```

FIGURA 6.10 Função que implementa o método de Brent para localização de raízes com base no programa de MATLAB desenvolvido por Cleve Moler (2005).

Em seguida, o laço principal é implementado. Se necessário, os três pontos são rearranjados para satisfazer às condições necessárias para o algoritmo funcionar de forma eficiente. Nesse ponto, se os critérios de parada são satisfeitos, o laço é encerrado. Caso contrário, uma estrutura de decisão escolhe entre os três métodos e

verifica se o resultado é aceitável. Então, uma sessão final calcula f no novo ponto e o laço é repetido. Uma vez que os critérios de parada sejam satisfeitos, o laço é encerrado e a estimativa final da raiz é retornada.

6.5 FUNÇÃO DO MATLAB: `fzero`

A função `fzero` é projetada para encontrar uma raiz real de uma única equação. Uma representação simples de sua sintaxe é

```
fzero(funcao, x0)
```

onde `funcao` é o nome da função sendo avaliada e `x0` é a aproximação inicial.

Observe que duas aproximações que delimitam a raiz podem ser passadas como um vetor:

```
fzero(funcao, [x0 x1])
```

onde `x0` e `x1` são aproximações que delimitam um intervalo em que ocorre mudança de sinal.

Apresentamos a seguir uma sessão do MATLAB que calcula a raiz de uma equação quadrática simples: $x^2 - 9$. Claramente existem duas raízes, uma em -3 e outra em 3 . Para encontrar a raiz negativa:

```
>> x = fzero(@(x) x^2-9, -4)
x =
-3
```

Se quisermos encontrar a raiz positiva, utilizamos uma aproximação próxima dela:

```
>> x = fzero(@(x) x^2-9, 4)
x =
3
```

Se definirmos a aproximação inicial igual a zero, a função retorna a raiz negativa:

```
>> x = fzero(@(x) x^2-9, 0)
x =
-3
```

Se quisermos garantir que encontraremos a raiz positiva, podemos entrar com duas aproximações, como em

```
>> x = fzero(@(x) x^2-9, [0 4])
x =
3
```

E se uma mudança de sinal não ocorre entre as duas aproximações, uma mensagem de erro é exibida

```
>> x = fzero(@(x) x^2-9, [-4 4])
??? Error using ==> fzero at 293
The function values at the interval endpoints must differ in sign.
```

A função `fzero` funciona como descrito a seguir. Se uma única aproximação inicial é passada, ela primeiro efetua uma busca para identificar uma mudança de

sinal. Essa busca difere da busca incremental descrita na Seção 5.3.1, pois a busca implementada na função `fzero` começa na aproximação inicial e, então, dá passos cada vez maiores tanto na direção positiva quanto na direção negativa até uma mudança de sinal ser detectada.

Em seguida, os métodos rápidos (secante ou interpolação quadrática inversa) são usados, a menos que um resultado inaceitável ocorra (por exemplo, a estimativa da raiz sair do intervalo delimitador). Se um resultado inaceitável acontece, a bissecção é implementada até um resultado aceitável ser obtido com um dos métodos rápidos. Como já mencionado, a bissecção domina o processo no início, mas, à medida que se aproxima da raiz, a técnica muda para os métodos mais rápidos.

Uma representação mais completa da sintaxe da função `fzero` pode ser escrita como

```
[x,fx] = fzero(funcão,x0,opcoes,p1,p2,...)
```

onde `[x,fx]` é um vetor que contém a raiz, `x`, a função calculada na raiz, `fx`, `opcoes`, é uma estrutura de dados criada pela função `optimset` e `p1, p2,...` são quaisquer parâmetros que a função requer. Caso você deseje passar parâmetros (`p1, p2,...`), mas não queira usar as `opcoes`, passe um vetor vazio `[]` em seu lugar.

A função `optimset` tem a sintaxe

```
opcoes = optimset('par1',val1,'par2',val2,...)
```

onde o parâmetro `par1` tem o valor `val1`, o parâmetro `par2` tem o valor `val2` e assim por diante. Uma lista completa de todos os parâmetros possíveis pode ser obtida ao digitar `optimset` no prompt de comando do MATLAB. Os parâmetros geralmente usados com a função `fzero` são

`display`: Quando definido como 'iter' exibe um registro detalhado de todas as iterações.

`tolx`: Um escalar positivo que define uma tolerância de terminação para `x`.

EXEMPLO 6.7 As funções `fzero` e `optimset`

Lembre-se de que no Exemplo 6.3 foi encontrada a raiz positiva de $f(x) = x^{10} - 1$ utilizando o método de Newton-Raphson com uma aproximação inicial de 0,5. Resolva o mesmo problema com as funções `optimset` e `fzero`.

Solução. A seguinte sessão do MATLAB pode ser implementada para solucionar o problema:

```
>> options = optimset('display','iter');
>> [x,fx] = fzero(@(x) x^10-1,0.5,options)
Func-count      x          f(x)      Procedure
    1            0.5        -0.999023    initial
    2            0.485858     -0.999267    search
    3            0.514142     -0.998709    search
    4            0.48           -0.999351    search
    5            0.52           -0.998554    search
    6            0.471716     -0.999454    search
```

```

.
.
.

23      0.952548      -0.385007      search
24      -0.14          -1            search
25      1.14          2.70722       search

Looking for a zero in the interval [-0.14, 1.14]
26      0.205272      -1            interpolation
27      0.672636      -0.981042     bisection
28      0.906318      -0.626056     bisection
29      1.02316       0.257278      bisection
30      0.989128      -0.103551     interpolation
31      0.998894      -0.0110017    interpolation
32      1.00001       7.68385e-005   interpolation
33      1              -3.83061e-007  interpolation
34      1              -1.3245e-011   interpolation
35      1              0              interpolation

Zero found in the interval: [-0.14, 1.14].
x =
1
fx =
0

```

Assim, após 25 iterações de busca, a função `fzero` encontra uma mudança de sinal. Após isso, usa interpolação e bissecção até se aproximar suficientemente da raiz, de modo que a interpolação assume o controle e rapidamente converge para a raiz.

Considere que se deseja utilizar uma tolerância menos rigorosa. A função `optimset` pode ser usada para definir uma tolerância máxima menor, o que resulta em uma estimativa da raiz menos acurada:

```

>> opcoes = optimset ('tolx', 1e-3);
>> [x,fx] = fzero(@(x) x^10-1,0.5,opcoes)
x =
1.0009
fx =
0.0090

```

6.6 POLINÔMIOS

Os polinômios são um tipo especial de equação algébrica não linear da forma geral

$$f_n(x) = a_1x^n + a_2x^{n-1} + \cdots + a_{n-1}x^2 + a_nx + a_{n+1} \quad (6.12)$$

onde n é a ordem do polinômio e os a 's são coeficientes constantes. Em muitos (mas não todos) casos, os coeficientes são reais; nesses casos, as raízes podem ser reais e/ou complexas. Em geral, um polinômio de ordem n terá n raízes.

Os polinômios têm muitas aplicações na engenharia e na ciência: uma delas é o uso no ajuste de curvas. Entretanto, uma de suas aplicações mais interessantes e

poderosas é na caracterização de sistemas dinâmicos – e, em particular, de sistemas lineares, como reatores, dispositivos mecânicos, estruturas e circuitos elétricos.

6.6.1 Função do MATLAB: `roots`

Se você estiver tratando de um problema no qual deve determinar uma única raiz de um polinômio, as técnicas anteriores, como a bissecção e o método de Newton-Raphson, podem ser úteis. No entanto, em muitos casos, os engenheiros desejam determinar todas as raízes, tanto as reais quanto as complexas. Infelizmente, técnicas simples como bissecção e Newton-Raphson não estão disponíveis para determinação de todas as raízes de polinômios de ordem mais elevada. O MATLAB, entretanto, tem um excelente recurso interno, a função `roots`, para executar essa tarefa.

A função `roots` tem a sintaxe

```
x = roots(c)
```

onde x é um vetor coluna que contém as raízes e c é um vetor linha que contém os coeficientes do polinômio.

Então, como a função `a roots` funciona? O MATLAB é muito eficiente em encontrar os autovalores de uma matriz. Consequentemente, a abordagem é remodelar a tarefa de cálculo de raízes como um problema de autovalores, assunto que será abordado mais adiante; por enquanto será fornecida apenas uma visão geral do tema.

Considere o seguinte polinômio

$$a_1x^5 + a_2x^4 + a_3x^3 + a_4x^2 + a_5x + a_6 = 0 \quad (6.13)$$

Dividindo por a_1 e rearranjando, temos

$$x^5 = -\frac{a_2}{a_1}x^4 - \frac{a_3}{a_1}x^3 - \frac{a_4}{a_1}x^2 - \frac{a_5}{a_1}x - \frac{a_6}{a_1}$$

Uma matriz especial pode ser construída com a utilização dos coeficientes do lado direito da equação como a primeira linha e com 1's e 0's escritos para as outras linhas conforme a seguir:

$$\begin{bmatrix} -a_2/a_1 & -a_3/a_1 & -a_4/a_1 & -a_5/a_1 & -a_6/a_1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.14)$$

A Equação (6.14) é chamada *matriz companheira* do polinômio; seus autovalores são as raízes do polinômio. Assim, o algoritmo da função `roots` consiste, fundamentalmente, na criação da matriz companheira e, após, na utilização das funcionalidades poderosas do MATLAB de determinação de autovalores para calcular as raízes. As aplicações da função `roots`, com outras funções de manipulação de polinômios relacionadas, são descritas no exemplo a seguir.

É importante ressaltar que a função `roots` tem uma função inversa chamada `poly`, que, quando passados os valores das raízes, retorna os coeficientes do polinômio. Sua sintaxe é

$$c = \text{poly}(r)$$

onde r é um vetor coluna que contém as raízes e c é um vetor linha que contém os coeficientes do polinômio.

EXEMPLO 6.8 Usando o MATLAB para manipular e determinar as raízes de polinômios

Use a seguinte equação para verificar como o MATLAB pode ser empregado para manipular polinômios:

$$f_5(x) = x^5 - 3,5x^4 + 2,75x^3 + 2,125x^2 - 3,875x + 1,25 \quad (\text{E6.8.1})$$

Observe que esse polinômio tem três raízes reais: 0,5, -1,0 e 2; e duas raízes complexas: $1 \pm 0,5i$.

Solução. Os polinômios são inseridos no MATLAB armazenando-se seus coeficientes como um vetor. Por exemplo, digitando a linha a seguir, os coeficientes são armazenados no vetor a :

```
>> a = [1 -3.5 2.75 2.125 -3.875 1.25];
```

Pode-se então prosseguir para manipular o polinômio. Por exemplo, é possível calculá-lo em $x=1$ digitando

```
>> polyval(a, 1)
```

com o resultado $1(1)^5 - 3,5(1)^4 + 2,75(1)^3 + 2,125(1)^2 - 3,875(1) + 1,25 = -0,25$:

```
ans =
-0.2500
```

Pode-se criar um polinômio quadrático que tenha raízes correspondentes a duas das raízes originais da Equação (E6.8.1): 0,5 e -1. Esse polinômio quadrático é $(x - 0,5)(x + 1) = x^2 + 0,5x - 0,5$, e pode ser inserido no MATLAB como o vetor b :

```
>> b = [1 .5 -.5]
b =
1.0000 0.5000 -0.5000
```

Observe que a função `poly` pode ser usada para executar a mesma tarefa, como em

```
>> b = poly([0.5 -1])
b =
1.0000 0.5000 -0.5000
```

Pode-se dividir o polinômio original por esse polinômio por meio de

```
>> [q, r] = deconv(a, b)
```

com o resultado sendo um quociente (um polinômio de grau três, q) e um resto (r)

```
q =
1.0000 -4.0000 5.2500 -2.5000
r =
0 0 0 0 0 0
```

Como o polinômio é um divisor perfeito, o polinômio resto tem coeficientes nulos. Agora, as raízes do polinômio quociente podem ser determinadas por

```
>> x = roots(q)
```

com o resultado esperado de que as raízes restantes do polinômio original [Equação (E6.8.1)] sejam encontradas

```
x =
2.0000
1.0000 + 0.5000i
1.0000 - 0.5000i
```

Pode-se agora multiplicar q por b para reencontrar o polinômio original:

```
>> a = conv(q, b)
a =
1.0000 -3.5000 2.7500 2.1250 -3.8750 1.2500
```

Todas as raízes do polinômio podem ser determinadas por

```
>> x = roots(a)
x =
2.0000
-1.0000
1.0000 + 0.5000i
1.0000 - 0.5000i
0.5000
```

Por fim, pode-se retornar novamente ao polinômio original utilizando a função `poly`:

```
>> a = poly(x)
a =
1.0000 -3.5000 2.7500 2.1250 -3.8750 1.2500
```

► 6.7 ESTUDO DE CASO ATRITO DA TUBULAÇÃO

Contextualização. A determinação do escoamento (fluxo) de fluidos através de canais e tubulações tem grande relevância em muitas áreas da engenharia e da ciência. Entre as aplicações típicas em engenharia estão o fluxo de líquido e gases através de dutos e sistemas de refrigeração. Já os cientistas estão interessados em tópicos que vão desde o fluxo nos vasos sanguíneos para transmissão de nutrientes até o sistema vascular de uma planta.

A resistência ao fluxo em tais canais é parametrizada por um número adimensional chamado *fator de atrito*, f . Para fluxo turbulento, a *equação de Colebrook* fornece meios para calcular o fator de atrito,

$$0 = \frac{1}{\sqrt{f}} + 2,0 \log \left(\frac{\varepsilon}{3,7D} + \frac{2,51}{Re \sqrt{f}} \right) \quad (6.15)$$

onde ε é a rugosidade (m), D é o diâmetro (m) e Re é o *número de Reynolds*:

$$Re = \frac{\rho V D}{\mu}$$

sendo que ρ é a densidade do fluido (kg/m^3), V é sua velocidade (m/s) e μ é a viscosidade dinâmica ($\text{N} \cdot \text{s/m}^2$). Além de aparecer na Equação (6.15), o número de Reynolds também serve como um critério para saber se o fluxo é turbulento ($\text{Re} > 4000$).

Neste estudo de caso, ilustraremos como os métodos numéricos abordados nesta parte do livro podem ser empregados para determinar f para o fluxo de ar através de uma tubulação lisa e fina. Para esse caso, os parâmetros são $\rho = 1,23 \text{ kg/m}^3$, $\mu = 1,79 \times 10^{-5} \text{ N} \cdot \text{s/m}^2$, $D = 0,005 \text{ m}$, $V = 40 \text{ m/s}$ e $\varepsilon = 0,0015 \text{ mm}$. Observe que os fatores de atrito variam de cerca de 0,008 a 0,08, e que uma formulação explícita chamada *equação de Swamee-Jain* fornece uma estimativa aproximada:

$$f = \frac{1,325}{\left[\ln \left(\frac{\varepsilon}{3,7D} + \frac{5,74}{\text{Re}^{0,9}} \right) \right]^2} \quad (6.16)$$

Solução. O número de Reynolds pode ser calculado como

$$\text{Re} = \frac{\rho V D}{\mu} = \frac{1,23(40)0,005}{1,79 \times 10^{-5}} = 13.743$$

Esse valor, com os outros parâmetros, pode ser substituído na Equação (6.15) para fornecer

$$g(f) = \frac{1}{\sqrt{f}} + 2,0 \log \left(\frac{0,0000015}{3,7(0,005)} + \frac{2,51}{13.743 \sqrt{f}} \right)$$

Antes de determinar a raiz, é aconselhável fazer um gráfico da função para estimar as aproximações iniciais e antecipar possíveis dificuldades. Isso pode ser feito facilmente com o MATLAB:

```
>> rho=1.23; mu=1.79e-5; D=0.005; V=40; e=0.0015/1000;
>> Re=rho*V*D/mu;
>> g=@(f) 1/sqrt(f)+2*log10(e/(3.7*D))+2.51/(Re*sqrt(f));
>> fplot(g,[0.008 0.08]), grid, xlabel('f'), ylabel('g(f)')
```

Como mostra a Figura 6.11, a raiz está localizada próximo de 0,03.

Como são fornecidas aproximações iniciais ($x_l = 0,008$ e $x_u = 0,08$), qualquer um dos métodos intervalares do Capítulo 5 poderiam ser usados. Por exemplo, a função *bissec* apresentada na Figura 5.7 fornece um valor de $f = 0,0289678$ com um erro relativo percentual de $5,926 \times 10^{-5}$ em 22 iterações. O método da falsa posição produz um resultado de precisão similar em 26 iterações; assim, embora tais métodos forneçam o resultado correto, são pouco eficientes. Isso não seria importante para uma única aplicação, porém poderia se tornar relevante se muitas avaliações fossem feitas.

Poderíamos tentar alcançar um melhor desempenho aplicando um método aberto. Como a Equação (6.15) é simples de diferenciar, o método de Newton-Raphson é um bom candidato. Por exemplo, usando uma aproximação inicial na extremidade inferior do intervalo ($x_0 = 0,008$), a função *newraph* apresentada na Figura 6.7 converge rapidamente:

```
>> dg=@(f) -2/log(10)*1.255/Re*f^(-3/2)/(e/D/3.7 ...
+2.51/Re/sqrt(f))-0.5/f^(3/2);
>> [f ea iter]=newraph(g,dg,0.008)
```



```
f =
0.02896781017144
ea =
6.870124190058040e-006
iter =
6
```

Porém, quando a aproximação inicial é definida na extremidade superior do intervalo ($x_0 = 0,08$), a rotina diverge,

```
>> [f ea iter]=newtraph(g,dg,0.08)
f =
NaN +
NaNi
```

Como mostra a Figura 6.11, isso ocorre porque a inclinação da função na aproximação inicial faz a primeira iteração pular para um valor negativo. Execuções adicionais da função demonstram que, para esse caso, a convergência ocorre apenas quando a aproximação inicial é menor que cerca de 0,066.

Então, podemos perceber que, embora o método de Newton-Raphson seja muito eficiente, ele requer uma boa estimativa inicial. Para a equação de Colebrook, uma boa estratégia poderia ser empregar a equação de Swamee-Jain [Equação (6.16)] para fornecer uma aproximação inicial como em

```
>> fSJ=1.325/log(e/(3.7*D)+5.74/Re^0.9)^2
fSJ =
0.02903099711265
>> [f ea iter]=newtraph(g,dg,fSJ)
f =
0.02896781017144
ea =
8.510189472800060e-010
iter =
3
```

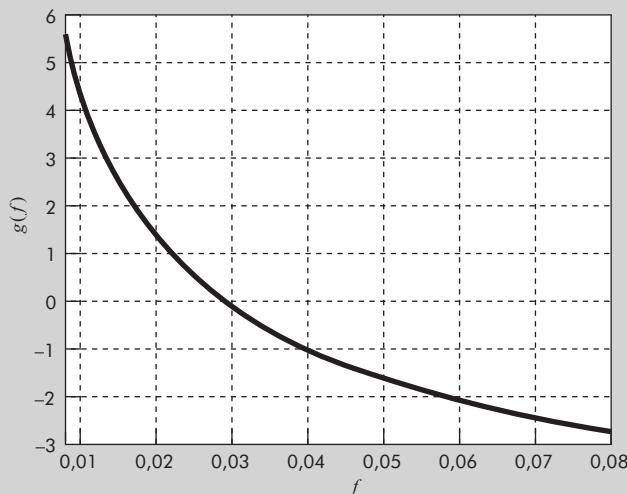


FIGURA 6.11

Além das funções desenvolvidas por nós mesmos, podemos usar a função nativa `fzero` do MATLAB. No entanto, assim como o método de Newton-Raphson, a função `fzero` também pode divergir quando utilizada com uma única aproximação inicial. Entretanto, nesse caso, aproximações na extremidade inferior do intervalo causam problemas. Por exemplo,

```
>> fzero(g, 0.008)
Exiting fzero: aborting search for an interval containing a sign
change because complex function value encountered ...
during search.
(Function value at -0.0028 is -4.92028-20.2423i.)
Check function or try again with a different starting value.
ans =
NaN
```

Se as iterações são exibidas usando a função `optimset` (lembre-se do Exemplo 6.7), percebe-se que um valor negativo ocorre durante a fase de busca antes de uma mudança de sinal ser detectada, e então a rotina aborta. Porém, para aproximações iniciais superiores a cerca de 0,016, a rotina funciona muito bem; é o caso da aproximação de 0,08 que causou problemas para o método de Newton-Raphson, no qual a função `fzero` teve uma boa aplicação:

```
>> fzero(g, 0.08)
ans =
0.02896781017144
```

Como uma última avaliação, vamos verificar se a iteração de ponto fixo simples converge. A versão mais fácil e simples envolve a solução para o primeiro f na Equação (6.15):

$$f_{i+1} = \frac{0,25}{\left(\log \left(\frac{\varepsilon}{3,7D} + \frac{2,51}{\operatorname{Re} \sqrt{f_i}} \right) \right)^2} \quad (6.17)$$

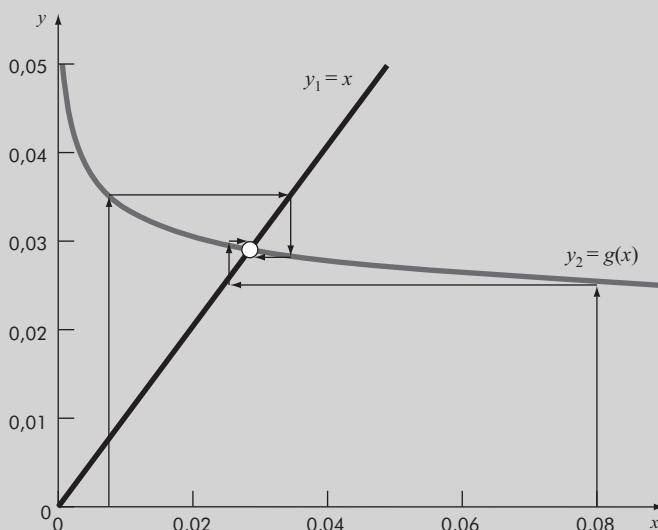


FIGURA 6.12

O gráfico das duas curvas associado à função analisada indica um resultado surpreendente (Figura 6.12). Lembre-se de que a iteração de ponto fixo converge quando a curva y_2 tem uma inclinação relativamente “plana” (isto é, $|g'(\xi)| < 1$). Como indicado pela Figura 6.12, o fato de a curva y_2 ser bastante plana no intervalo de $f = 0,008$ até $0,08$ significa não apenas que a iteração de ponto fixo converge, mas que converge muito rapidamente. De fato, para aproximações iniciais em qualquer lugar entre $0,008$ e $0,08$, a iteração de ponto fixo produz previsões com erros relativos percentuais menores que $0,008\%$ em seis ou menos iterações. Assim, essa abordagem simples, que requer apenas uma aproximação e não demanda o cálculo da derivada, funciona muito bem para esse caso particular.

A mensagem a ser guardada deste estudo de caso é que mesmo softwares profissionais poderosos como o MATLAB não são sempre infalíveis. Além disso, normalmente não há um método único que funcione bem para todos os problemas. Usuários sofisticados compreendem os pontos fortes e fracos das técnicas numéricas disponíveis, e compreendem o suficiente da teoria fundamental de modo que podem de fato tratar situações em que um método não funciona adequadamente.

►► PROBLEMAS

6.1 Use a iteração de ponto fixo simples para localizar a raiz de

$$f(x) = \sin(\sqrt{x}) - x$$

Use a aproximação inicial $x_0 = 0,5$ e itere até que $\varepsilon_a \leq 0,01\%$. Verifique que o processo é linearmente convergente, como descrito no final da Seção 6.1.

6.2 Use (a) a iteração de ponto fixo e (b) o método de Newton-Raphson para determinar a raiz de $f(x) = -0,9x^2 + 1,7x + 2,5$ utilizando $x_0 = 0,5$. Faça os cálculos até que ε_a seja menor do que $\varepsilon_s = 0,01\%$. Faça também uma verificação do erro na sua resposta final.

6.3 Determine a maior raiz real de $f(x) = x^3 - 6x^2 + 11x - 6,1$:

(a) Graficamente.

(b) Usando o método de Newton-Raphson (três iterações, $x_i = 3,5$).

(c) Usando o método da secante (três iterações, $x_{i-1} = 2,5$ e $x_i = 3,5$).

(d) Usando o método da secante modificado (três iterações, $x_i = 3,5$, $\delta = 0,01$).

(e) Determine todas as raízes com o MATLAB.

6.4 Determine a menor raiz positiva de $f(x) = 7 \operatorname{sen}(x)e^{-x} - 1$:

(a) Graficamente.

(b) Usando o método de Newton-Raphson (três iterações, $x_i = 0,3$).

(c) Usando o método da secante (três iterações, $x_{i-1} = 0,5$ e $x_i = 0,4$).

(d) Usando o método da secante modificado (três iterações, $x_i = 0,3$, $\delta = 0,01$).

6.5 Use (a) o método de Newton-Raphson e (b) o método da secante modificado ($\delta = 0,05$) para determinar uma raiz de $f(x) = x^5 - 16,05x^4 + 88,75x^3 - 192,0375x^2 + 116,35x + 31,6875$ utilizando uma aproximação inicial de $x_0 = 0,5825$ e $\varepsilon_s = 0,01\%$. Explique seus resultados.

6.6 Desenvolva uma função no MATLAB para o método da secante. Com as duas aproximações iniciais, passe a função como um argumento e teste a função resolvendo o Problema 6.3.

6.7 Desenvolva uma função no MATLAB para o método da secante modificado. Com a aproximação inicial e a fração de perturbação, passe a função como um argumento. Teste a função resolvendo o Problema 6.3.

6.8 Diferencie a Equação (E6.4.1) para obter a Equação (E6.4.2).

6.9 Empregue o método de Newton-Raphson para determinar uma raiz real de $f(x) = -2 + 6x - 4x^2 + 0,5x^3$, utilizando uma aproximação inicial de (a) 4,5 e (b) 4,43. Discuta e use métodos analíticos e gráficos para explicar quaisquer peculiaridades em seus resultados.

6.10 O método babilônico, um antigo método para aproximação da raiz quadrada de qualquer número positivo a , pode ser formulado como

$$x_{i+1} = \frac{x_i + a/x_i}{2}$$

Mostre que essa fórmula é baseada no algoritmo de Newton-Raphson.

6.11 (a) Aplique o método de Newton-Raphson para a função $f(x) = \operatorname{tanh}(x^2 - 9)$ a fim de avaliar sua raiz real conhecida em $x = 3$. Use uma aproximação inicial de $x_0 = 3,2$ e tome um mínimo de três iterações. **(b)** O método convergiu para sua raiz real? Esboce o gráfico com os resultados identificados para cada iteração.

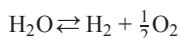
6.12 O polinômio $f(x) = 0,0074x^4 - 0,284x^3 + 3,355x^2 - 12,183x + 5$ tem uma raiz real entre 15 e 20. Aplique o método de Newton-Raphson para essa função utilizando uma aproximação inicial de $x_0 = 16,15$. Explique seus resultados.

6.13 Os engenheiros mecânicos, da mesma forma que a maioria dos outros engenheiros, utilizam frequentemente a termodinâmica nos seus trabalhos. O seguinte polinômio pode ser usado para relacionar o calor específico a pressão nula do ar seco, c_p kJ/(kg K), à temperatura (K):

$$c_p = 0,99403 + 1,671 \times 10^{-4} T + 9,7215 \times 10^{-8} T^2 - 9,5838 \times 10^{-11} T^3 + 1,9520 \times 10^{-14} T^4$$

Escreva um programa no MATLAB **(a)** para traçar um gráfico de c_p versus T em uma faixa de 0 a 1.200 K, e **(b)** para determinar a temperatura que corresponde a um calor específico de 1,1 kJ/(kg K), empregando as funções polinomiais do MATLAB.

6.14 Em um processo de engenharia química, o vapor de água (H_2O) é aquecido a uma temperatura suficientemente alta para que uma parte significativa da água se dissocie, ou se quebre, para formar oxigênio (O_2) e hidrogênio (H_2):



Se for assumido que essa é a única reação envolvida, a fração molar x de H_2O que se dissocia pode ser representada por

$$K = \frac{x}{1-x} \sqrt{\frac{2p_t}{2+x}} \quad (\text{P6.14.1})$$

onde K é a constante de equilíbrio da reação e p_t é a pressão total da mistura. Se $p_t = 3$ atm e $K = 0,05$, determine o valor de x que satisfaz a Equação (P6.14.1).

6.15 A equação de estado de Redlich-Kwong é dada por

$$p = \frac{RT}{v-b} - \frac{a}{v(v+b)\sqrt{T}}$$

onde R é a constante universal dos gases [= 0,518 kJ/(kg K)], T é a temperatura absoluta (K), p é a

pressão absoluta (kPa) e v é o volume de um kg de gás (m³/kg). Os parâmetros a e b são calculados por

$$a = 0,427 \frac{R^2 T_c^{2,5}}{p_c} \quad b = 0,0866 R \frac{T_c}{p_c}$$

onde p_c é a pressão crítica (kPa) e T_c é a temperatura crítica (K). Como engenheiro químico, foi-lhe pedido para determinar a quantidade de gás metano ($p_c = 4.600$ kPa e $T_c = 191$ K) que pode ser mantido em um tanque de 3 m³ a uma temperatura de -40 °C, com uma pressão de 65.000 kPa. Use um método para localizar raízes de sua escolha para calcular v e a seguir determinar a massa de metano contida no tanque.

6.16 O volume V de líquido em um cilindro horizontal oco de raio r e comprimento L se relaciona com a profundidade do líquido h por

$$V = \left[r^2 \cos^{-1} \left(\frac{r-h}{r} \right) - (r-h)\sqrt{2rh-h^2} \right] L$$

Determine h , dado que $r = 2$ m, $L = 5$ m e $V = 8$ m³.

6.17 Um cabo catenário é um cabo que está pendurado entre dois pontos que não estão na mesma reta vertical. Como mostra a Figura P6.17a, ele não está sujeito a nenhuma carga além do seu próprio peso, portanto, seu peso age como uma carga uniforme por unidade de comprimento ao longo do cabo, w (N/m). Um diagrama de corpo livre de uma seção AB é apresentado na Figura P6.17b, onde T_A e T_B são as forças de tensão nas extremidades. Com base nos balanços de força horizontal e vertical, é possível deduzir a seguinte equação diferencial para modelar o cabo:

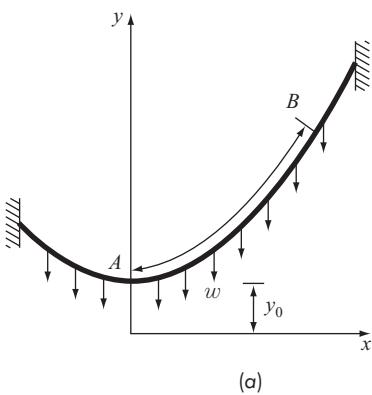
$$\frac{d^2y}{dx^2} = \frac{w}{T_A} \sqrt{1 + \left(\frac{dy}{dx} \right)^2}$$

O cálculo pode ser usado para resolver essa equação a fim de determinar a altura y do cabo como uma função da distância x :

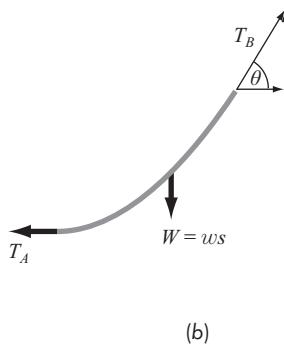
$$y = \frac{T_A}{w} \cosh \left(\frac{w}{T_A} x \right) + y_0 - \frac{T_A}{w}$$

- (a)** Use um método numérico para calcular um valor para o parâmetro T_A , dados os valores para os parâmetros $w = 10$ e $y_0 = 5$, tal que o cabo tenha uma altura de $y = 15$ em $x = 50$.
- (b)** Faça um gráfico de y versus x para $x = -50$ até 100.

6.18 Uma corrente oscilante em um circuito elétrico é descrita por $I = 9e^{-t} \operatorname{sen}(2\pi t)$, onde t está em segundos. Determine todos os valores de t tais que $I = 3,5$.



(a)



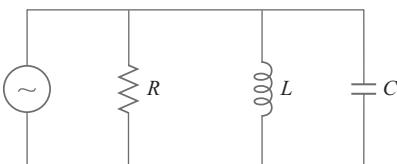
(b)

FIGURA P6.17

6.19 A Figura P6.19 mostra um circuito com um resistor, um indutor e um capacitor em paralelo. As leis de Kirchhoff podem ser usadas para expressar a impedância do sistema como

$$\frac{1}{Z} = \sqrt{\frac{1}{R^2} + \left(\omega C - \frac{1}{\omega L}\right)^2}$$

onde Z é a impedância (Ω) e ω é a frequência angular. Encontre a frequência ω que resulta em uma impedância de 100Ω utilizando a função `fzero` com aproximações iniciais de 1 e 1.000, e considerando os seguintes parâmetros: $R = 225 \Omega$, $C = 0,6 \times 10^{-6} F$ e $L = 0,5 H$.

**FIGURA P6.19**

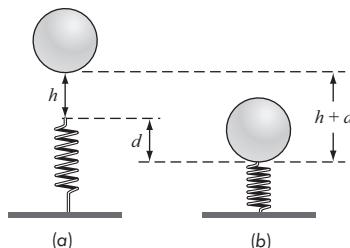
6.20 Os sistemas mecânicos reais podem envolver a deflexão de molas não lineares. Na Figura P6.20, uma massa m é abandonada de uma distância h acima de uma mola não linear. A força de resistência F da mola é dada por

$$F = -(k_1 d + k_2 d^{3/2})$$

A conservação da energia pode ser usada para mostrar que

$$0 = \frac{2k_2 d^{5/2}}{5} + \frac{1}{2} k_1 d^2 - mgd - mgh$$

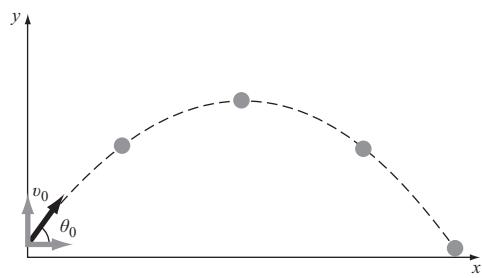
Determine d , dados os seguintes valores dos parâmetros: $k_1 = 40.000 \text{ g/s}^2$, $k_2 = 40 \text{ g/(s}^2 \text{ m}^3)$, $m = 95 \text{ g}$, $g = 9,81 \text{ m/s}^2$ e $h = 0,43 \text{ m}$.

**FIGURA P6.20**

6.21 Os engenheiros aeroespaciais algumas vezes calculam a trajetória de projéteis como foguetes. Um problema relacionado trata da trajetória de uma bola lançada. A trajetória da bola é definida pelas coordenadas (x, y) , como mostra a Figura P6.21. A trajetória pode ser modelada por

$$y = (\tan \theta_0)x - \frac{g}{2v_0^2 \cos^2 \theta_0} x^2 + y_0$$

Encontre o ângulo inicial apropriado θ_0 , se a velocidade inicial for $v_0 = 30 \text{ m/s}$ e a distância do receptor x for 90 m. Observe que a bola deixa a mão do lançador a uma elevação de $y_0 = 1,8 \text{ m}$ e o receptor a recebe a 1 m.

**FIGURA P6.21**

6.22 Você está projetando um tanque esférico (Figura P6.22) para armazenar água para uma pequena vila em um país em desenvolvimento. O volume de líquido que ele pode armazenar pode ser calculado por

$$V = \pi h^2 \frac{[3R - h]}{3}$$

onde V é o volume (m^3), h é a profundidade da água no tanque (m) e R é o raio do tanque (m).

Se $R = 3$ m, até qual profundidade o tanque deve ser enchido para que armazene 30 m^3 ? Use três iterações do método numérico mais adequado para determinar sua resposta. Determine o erro relativo aproximado depois de cada iteração. Forneça também uma justificativa para sua escolha do método. Informações adicionais: (a) Para os métodos intervalares, aproximações iniciais de 0 e R irão delimitar uma raiz única para esse problema. (b) Para os métodos abertos, uma aproximação inicial de R será sempre convergente.

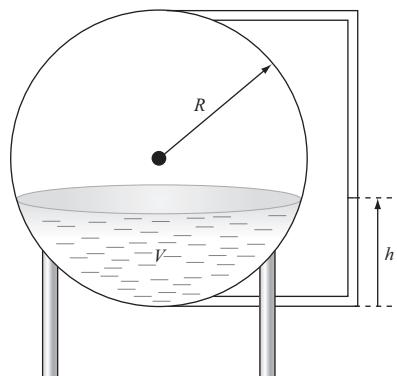


FIGURA P6.22

6.23 Realize as mesmas operações do MATLAB efetuadas no Exemplo 6.8 para manipular e encontrar todas as raízes do polinômio

$$f_5(x) = (x + 2)(x + 5)(x - 6)(x - 4)(x - 8)$$

6.24 Na análise de sistemas de controle são desenvolvidas funções de transferência que relacionam matematicamente a dinâmica da entrada de um sistema com sua saída. Uma função de transferência para um sistema de posicionamento de um robô é dada por

$$G(s) = \frac{C(s)}{N(s)} = \frac{s^3 + 9s^2 + 26s + 24}{s^4 + 15s^3 + 77s^2 + 153s + 90}$$

onde $G(s)$ é o ganho do sistema, $C(s)$ é a saída do sistema, $N(s)$ é a entrada do sistema e s é a frequência complexa da transformada de Laplace. Use o MATLAB para encontrar as raízes do numerador e do denominador e fatore-os na forma

$$G(s) = \frac{(s + a_1)(s + a_2)(s + a_3)}{(s + b_1)(s + b_2)(s + b_3)(s + b_4)}$$

onde a_i e b_i são as raízes do numerador e do denominador, respectivamente.

6.25 A equação de Manning pode ser escrita para um canal aberto retangular como

$$Q = \frac{\sqrt{S}(BH)^{5/3}}{n(B + 2H)^{2/3}}$$

onde Q é o fluxo (m^3/s), S é a inclinação (m/m), H é a profundidade (m) e n = o coeficiente de rugosidade de Manning. Desenvolva um esquema de iteração de ponto fixo para resolver essa equação para H , dados $Q = 5$, $S = 0,0002$, $B = 20$ e $n = 0,03$. Efetue o cálculo até que ε_a seja menor que $\varepsilon_s = 0,05\%$. Prove que seu esquema converge para todas as aproximações iniciais maiores ou iguais a zero.

6.26 Veja se você pode desenvolver uma função infalível para calcular o fator de atrito baseada na equação de Colebrook, como descrita na Seção 6.7. Sua função deve retornar um resultado preciso para um número de Reynolds variando de 4000 a 10^7 e para razão ε/D variando de $0,00001$ a $0,05$.

6.27 Use o método de Newton-Raphson para encontrar a raiz de

$$f(x) = e^{-0,5x}(4 - x) - 2$$

Empregue aproximações iniciais de (a) 2, (b) 6 e (c) 8. Explique seus resultados.

6.28 Dado

$$f(x) = -2x^6 - 1,5x^4 + 10x + 2$$

Use uma técnica de localização de raízes para determinar o máximo dessa função. Efetue iterações até que o erro relativo aproximado caia abaixo de 5%. Se você aplicar um método intervalar, use aproximações iniciais de $x_l = 0$ e $x_u = 1$. Se aplicar o método de Newton-Raphson ou o método da secante modificado, use uma aproximação inicial de $x_i = 1$. Se aplicar o método da secante, use aproximações iniciais de $x_{i-1} = 0$ e $x_i = 1$. Assumindo que convergência não é um inconveniente, escolha a técnica mais adequada para esse problema. Justifique sua escolha.

6.29 Determine a raiz da equação a seguir, cuja derivada é fácil de ser calculada.

$$e^{0.5x} = 5 - 5x$$

Escolha a melhor técnica numérica, justifique sua escolha e, após, aplique a técnica para determinar a raiz. Observe que, para aproximações iniciais positivas, todas as técnicas – exceto a iteração de ponto fixo – convergirão. Efetue iterações até que o erro relativo aproximado caia abaixo de 2%. Se você aplicar um método intervalar, use aproximações iniciais de $x_l = 0$ e $x_u = 2$. Se aplicar o método de Newton-Raphson ou o método da secante modificado, use uma aproximação inicial de $x_i = 0,7$. Se você aplicar o método da secante, use aproximações iniciais de $x_{i-1} = 0$ e $x_i = 2$.

6.30 (a) Desenvolva uma função do MATLAB para implementar o método de localização de raízes de Brent. Tome como base para sua função o código da Figura 6.10, porém com o início da função modificado para

```
function [b,fb] = fzeronova (f,xl,xu,
varargin)
% fzeronova: localização de raízes
```

pelo método de Brent

```
% [b,fb] = fzeronova(f,xl,xu,p1,p2,...):
% usa o método de Brent para encontrar
a raiz de f
% entrada:
% f = nome da função
% xl, xu = aproximações inferior e
superior
% p1,p2,... = parâmetros adicionais
usados por f
% saída:
% b = raiz real
% fb = valor da função na raiz
```

Faça as modificações apropriadas de modo que a função trabalhe conforme descrito nas linhas de documentação anteriores (linhas de comentário da função). Seu código deve ser capaz de verificar erros, de modo a garantir que os três argumentos necessários da função (f, xl, xu) sejam passados e que as aproximações iniciais delimitem uma raiz.

(b) Teste sua função utilizando-a para calcular a raiz da função do Exemplo 5.6 usando

```
>> [x,fx] = fzeronova(@(x,n) x^n-1,
0,1,3,10)
```

Otimização

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar uma introdução de como a otimização pode ser usada para determinar mínimos e máximos tanto de funções unidimensionais quanto de funções multidimensionais. Após esta leitura completa, você será capaz de:

- ✓ Entender por que e onde a otimização ocorre na solução de problemas científicos e de engenharia.
- ✓ Reconhecer a diferença entre otimização unidimensional e multidimensional.
- ✓ Distinguir entre ótimos globais e locais.
- ✓ Aprender como reformular um problema de maximização de modo que ele possa ser resolvido com um algoritmo de minimização.
- ✓ Definir a razão áurea e entender por que ela torna a otimização unidimensional eficiente.
- ✓ Localizar o ótimo de uma função de única variável com a busca da seção áurea.
- ✓ Localizar o ótimo de uma função de única variável com a interpolação quadrática.
- ✓ Aprender como aplicar a função `fminbnd` para determinar o mínimo de uma função unidimensional.
- ✓ Desenvolver gráficos de curvas de nível e de superfície no MATLAB para visualizar funções bidimensionais.
- ✓ Aprender como aplicar a função `fminsearch` para determinar o mínimo de uma função multidimensional.

VOCÊ TEM UM PROBLEMA

Um objeto (corpo) como um saltador de *bungee jumping* pode ser projetado para cima com uma velocidade específica. Se ele é submetido a um arraste linear, sua altitude como uma função do tempo pode ser calculada como

$$z = z_0 + \frac{m}{c} \left(v_0 + \frac{mg}{c} \right) \left(1 - e^{-(c/m)t} \right) - \frac{mg}{c} t \quad (7.1)$$

onde z é a altitude (m) acima da superfície da terra (definida como $z = 0$), z_0 é a altitude inicial (m), m é a massa (kg), c é um coeficiente de arraste linear (kg/s), v_0 é a velocidade inicial (m/s) e t é o tempo (s). Observe que, para essa formulação, uma velocidade positiva é considerada com o sentido para cima. Dados os seguintes valores dos parâmetros: $g = 9,81 \text{ m/s}^2$, $z_0 = 100 \text{ m}$, $v_0 = 55 \text{ m/s}$, $m = 80 \text{ kg}$, $c = 15 \text{ kg/s}$, a Equação (7.1) pode ser usada para calcular a altitude do saltador. Como mostra a Figura 7.1, o saltador atinge uma elevação máxima de aproximadamente 190 m em cerca de $t = 4 \text{ s}$.

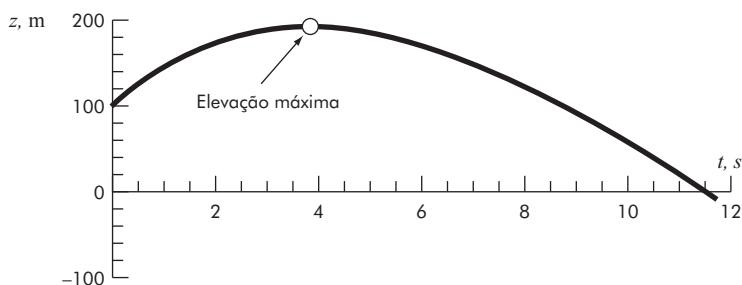


FIGURA 7.1 Elevação como uma função do tempo para um objeto inicialmente projetado para cima com uma velocidade inicial.

Considere que você receba a tarefa de determinar o tempo exato da elevação máxima. Neste capítulo, apresentaremos uma introdução de como o computador é utilizado para fazer tais determinações de valores extremos, chamadas de otimização.

7.1 INTRODUÇÃO E CONTEXTUALIZAÇÃO

No sentido mais geral, a otimização é o processo de criação de algo que é o mais eficiente possível. Engenheiros devem continuamente projetar dispositivos e produtos que desempenhem tarefas de forma eficiente com o menor custo. Assim, esses profissionais estão sempre confrontando problemas de otimização que buscam equilibrar o desempenho e as limitações, enquanto os cientistas têm interesse em fenômenos ótimos, que vão desde a elevação máxima de projéteis até a energia livre mínima.

De uma perspectiva matemática, a otimização trata da determinação dos máximos e mínimos de uma função que depende de uma ou mais variáveis. O objetivo é determinar os valores das variáveis que produzem máximos ou mínimos para a função. Em seguida, esses valores podem ser substituídos de volta na função para calcular os valores ótimos.

Embora essas soluções possam algumas vezes ser obtidas analiticamente, a maioria dos problemas práticos de otimização requer soluções numéricas computacionais. De um ponto de vista numérico, a otimização é similar, em essência, aos métodos de localização de raízes abordados nos Capítulos 5 e 6. Isto é, ambos envolvem a estimativa e a busca de um ponto em uma função. A diferença fundamental entre os dois tipos de problema é ilustrada na Figura 7.2. A localização de raízes envolve a busca dos locais onde a função é igual a zero, já a otimização envolve a busca dos pontos extremos da função (um mínimo ou um máximo).

Como mostra a Figura 7.2, os pontos ótimos são pontos onde a curva é achata-dada. Em termos matemáticos, isso corresponde ao valor de x onde a derivada $f'(x)$ é igual a zero. A segunda derivada, $f''(x)$, por sua vez, indica quando o ponto ótimo é um mínimo ou um máximo: se $f''(x) < 0$, o ponto é um máximo; se $f''(x) > 0$, o ponto é um mínimo.

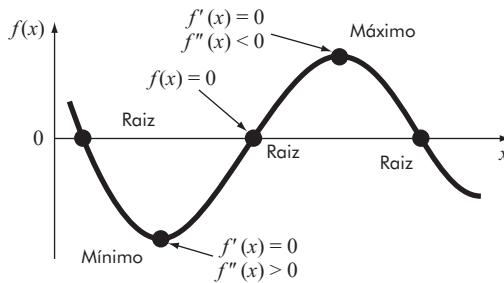


FIGURA 7.2 Função de uma única variável que ilustra a diferença entre raízes e pontos ótimos.

Agora, compreendendo a relação entre as raízes e os pontos ótimos, poderíamos sugerir uma possível estratégia para encontrar os pontos ótimos de uma função; por exemplo, podemos diferenciar a função e localizar a raiz (isto é, o zero) da nova função. De fato, alguns métodos de otimização fazem exatamente isso pela resolução do problema de raízes: $f'(x) = 0$.

EXEMPLO 7.1 Determinação dos pontos ótimos analiticamente pela localização das raízes

Determine o tempo e o valor da elevação máxima com base na Equação (7.1). Use os seguintes parâmetros em seus cálculos: $g = 9,81 \text{ m/s}^2$, $z_0 = 100 \text{ m}$, $v_0 = 55 \text{ m/s}$, $m = 80 \text{ kg}$, $c = 15 \text{ kg/s}$.

Solução. A Equação (7.1) pode ser diferenciada para fornecer

$$\frac{dz}{dt} = v_0 e^{-(c/m)t} - \frac{mg}{c} (1 - e^{-(c/m)t}) \quad (\text{E7.1.1})$$

Observe que, como $v = dz/dt$, essa é, na realidade, a equação para a velocidade. A máxima elevação ocorre no valor de t que leva essa equação para zero; assim, o problema equivale à determinação da raiz. Para esse caso, isso pode ser feito igualando a derivada a zero e resolvendo a Equação (E.7.1.1) analiticamente para

$$t = \frac{m}{c} \ln \left(1 + \frac{cv_0}{mg} \right)$$

Substituindo os parâmetros, temos

$$t = \frac{80}{15} \ln \left(1 + \frac{15(55)}{80(9,81)} \right) = 3,83166 \text{ s}$$

Esse valor, junto aos parâmetros, pode então ser substituído na Equação (7.1) para calcular a elevação máxima como

$$z = 100 + \frac{80}{15} \left(50 + \frac{80(9,81)}{15} \right) \left(1 - e^{-(15/80)3,83166} \right) - \frac{80(9,81)}{15} (3,83166) = 192,8609 \text{ m}$$

Pode-se verificar que o resultado é um máximo pela diferenciação da Equação (E7.1.1) para obter a segunda derivada

$$\frac{d^2z}{dt^2} = -\frac{c}{m}v_0e^{-(c/m)t} - ge^{-(c/m)t} = -9,81 \frac{m}{s^2}$$

O fato de a segunda derivada ser negativa nos informa que temos um máximo. Além disso, o resultado é fisicamente consistente, uma vez que a aceleração no corpo deveria ser devida apenas à força da gravidade na elevação máxima, quando a velocidade vertical (e, portanto, o arraste) é zero.

Embora uma solução analítica tenha sido possível para esse caso, o mesmo resultado poderia ser obtido utilizando os métodos de localização de raízes descritos nos Capítulos 5 e 6. Isso será deixado como exercício para o leitor.

Apesar da possibilidade de se abordar a otimização como um problema de raízes, uma variedade de métodos numéricos diretos para otimização está disponível, tanto para problemas unidimensionais quanto multidimensionais. Como o nome indica, os problemas unidimensionais envolvem funções que dependem de uma única variável dependente. Como na Figura 7.3a, a busca, então, consiste em subir ou descer picos ou vales. Já os problemas multidimensionais envolvem funções que dependem de duas ou mais variáveis dependentes. Seguindo o mesmo raciocínio, uma otimização bidimensional pode ser novamente visualizada como buscando picos ou vales (Figura 7.3b). Entretanto, do mesmo modo como em uma trilha real, não se limita a andar em uma única direção; em vez disso, a topografia deve ser examinada a fim de se alcançar a meta de forma eficiente.

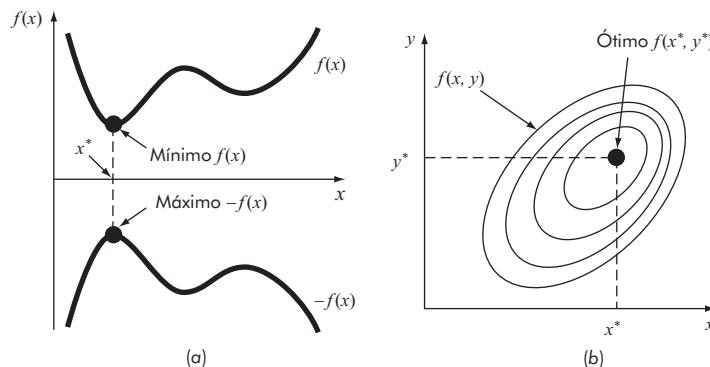


FIGURA 7.3 (a) Otimização unidimensional. A figura também ilustra como a minimização de $f(x)$ é equivalente à maximização de $-f(x)$.
 (b) Otimização bidimensional. Observe que essa figura pode ser usada para representar a maximização (os contornos aumentam em elevação até o máximo, como em uma montanha) ou a minimização (os contornos decrescem em elevação até o mínimo, como em um vale).

Por fim, o processo de busca pelo máximo comparado com a busca pelo mínimo é basicamente idêntico, porque o mesmo valor x^* tanto minimiza $f(x)$ quanto maximiza $-f(x)$. Essa equivalência é ilustrada graficamente para uma função unidimensional na Figura 7.3a.

Na próxima seção, descreveremos algumas das abordagens mais comuns para a otimização unidimensional. Em seguida, apresentaremos uma breve descrição sobre o emprego do MATLAB para determinar os valores ótimos para funções multidimensionais.

7.2 OTIMIZAÇÃO UNIDIMENSIONAL

Esta seção descreverá técnicas para encontrar o mínimo ou o máximo de uma função de uma única variável $f(x)$. Uma imagem útil com relação a isso é a “montanha-russa” unidimensional – como a função descrita na Figura 7.4. Nos Capítulos 5 e 6, vimos que a localização das raízes era complicada pelo fato de que podiam ocorrer várias raízes para uma única função. De modo similar, podem ocorrer tanto ótimos locais quanto globais na otimização.

Um *ótimo global* representa a melhor solução possível. Um *ótimo local*, embora não represente a melhor solução, é melhor que seus vizinhos imediatos. Os casos que incluem ótimos locais são chamados *multimodais*, nos quais quase sempre estaremos interessados em encontrar o ótimo global. Além disso, devemos ser cuidadosos para não confundir um resultado local com o ponto ótimo global.

Assim como a localização de raízes, a otimização em uma dimensão pode ser dividida nos métodos intervalares e abertos. Como descrito na próxima seção, a busca da razão áurea é um exemplo de método intervalar bastante similar, em essência, ao método da bissecção para localização de raízes, o qual é seguido por uma abordagem intervalar um pouco mais sofisticada – a interpolação parabólica. Após, mostra-se como esses dois métodos são combinados e implementados com a função `fminbnd` do MATLAB.

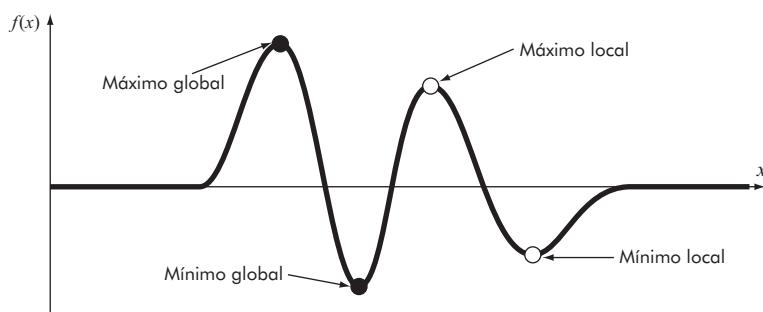


FIGURA 7.4 Uma função que se aproxima assintoticamente de zero em mais e menos infinito tem dois pontos de máximos e dois pontos de mínimos na vizinhança da origem. Os dois pontos à direita são pontos locais, enquanto os dois à esquerda são globais.

7.2.1 Busca da razão áurea

Em muitas culturas são atribuídas qualidades mágicas a certos números. Por exemplo, no Ocidente, são expressões comuns “7 da sorte” e “Sexta-feira 13”. Além dessas quantidades supersticiosas, existem vários números bem conhecidos que têm propriedades matemáticas interessantes e poderosas, de tal forma que podem ser verdadeiramente chamados “mágicos”. Os mais comuns desses números são a razão entre a circunferência de um círculo e seu diâmetro, π , e a base do logaritmo natural, e .

Embora não seja tão conhecida, a *razão áurea* certamente deveria ser incluída no panteão dos números notáveis. Essa quantidade, normalmente representada pela letra Grega ϕ (pronuncia-se “fí”), foi originalmente definida por Euclides (300 a.C.) em função do seu papel na construção do pentagrama ou estrela de cinco pontas. Como mostra a Figura 7.5, segundo definição de Euclides: “Diz-se que uma linha é dividida em média e extrema razão quando toda a linha está para o maior segmento, assim como o maior segmento está para o menor.”

O valor real da razão áurea pode ser deduzido expressando a definição de Euclides como

$$\frac{\ell_1 + \ell_2}{\ell_1} = \frac{\ell_1}{\ell_2} \quad (7.2)$$

Multiplicando por ℓ_1/ℓ_2 e agrupando os termos comuns, temos

$$\phi^2 - \phi - 1 = 0 \quad (7.3)$$

onde $\phi = \ell_1/\ell_2$. A raiz positiva dessa equação é a razão áurea:

$$\phi = \frac{1 + \sqrt{5}}{2} = 1,61803398874989\dots \quad (7.4)$$

A razão áurea, considerada por muito tempo esteticamente agradável nas culturas ocidentais, surge em uma variedade de contextos, incluindo a biologia; e esta, para nossos propósitos, fornece a base para a busca daquela, como um método simples e de uso geral para determinação do ponto ótimo de uma função de única variável.

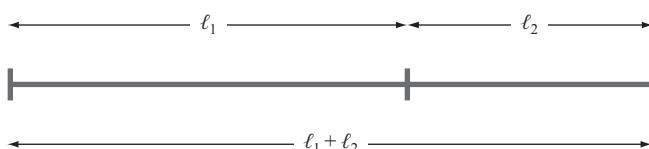


FIGURA 7.5 A definição de Euclides para a razão áurea é baseada na divisão de uma linha em dois segmentos, de modo que a razão de toda a linha para o maior segmento seja igual à razão do segmento maior para o menor. Essa razão é conhecida como a razão áurea.

A busca da razão áurea é similar à abordagem da bissecção para localizar raízes, descrita no Capítulo 5. Lembre-se de que a bissecção dependia da definição de um intervalo, especificado por uma aproximação inferior (x_l) e por uma aproximação superior (x_u), que delimitava uma única raiz. A presença de uma raiz entre esses limites era verificada determinando-se que $f(x_l)$ e $f(x_u)$ tinham sinais diferentes. A raiz era então estimada como o ponto médio desse intervalo:

$$x_r = \frac{x_l + x_u}{2} \quad (7.5)$$

O passo final na iteração da bissecção envolvia a determinação de um intervalo menor, que era feito substituindo-se qualquer uma das extremidades x_l ou x_u cujo valor da função tivesse o mesmo sinal que $f(x_r)$. Uma vantagem dessa abordagem era que o novo valor x_r substituía uma das extremidades antigas.

Agora, considere que, em vez de uma raiz, estejamos interessados em determinar o mínimo de uma função unidimensional. Como na bissecção, pode-se começar definindo um intervalo que contenha uma única resposta. Assim, o intervalo deve conter um único mínimo e, portanto, é chamado *unimodal*. Pode-se adotar a mesma nomenclatura para a bissecção, onde x_l e x_u definem as extremidades inferior e superior, respectivamente, de um intervalo. No entanto, em contraste com a bissecção, é necessária uma nova estratégia para achar o mínimo nesse intervalo: em vez de usar um único valor intermediário (o que é suficiente para determinar uma mudança de sinal e, consequentemente, um zero), serão necessários dois valores intermediários da função para detectar a ocorrência do mínimo.

A chave para tornar a abordagem mais eficiente é uma escolha judiciosa dos pontos intermediários. Como na bissecção, o objetivo é minimizar cálculos da função substituindo os antigos valores por novos valores: para a bissecção, isso foi realizado pela escolha do ponto médio; para a busca da seção áurea, os dois pontos intermediários são escolhidos de acordo com a razão áurea:

$$x_1 = x_l + d \quad (7.6)$$

$$x_2 = x_u - d \quad (7.7)$$

onde

$$d = (\phi - 1)(x_u - x_l) \quad (7.8)$$

A função é calculada nesses dois pontos interiores. Podem ocorrer dois resultados:

1. Se, como na Figura 7.6a, $f(x_1) < f(x_2)$, então $f(x_1)$ é o mínimo, e o domínio de x à esquerda de x_2 , de x_l a x_2 , pode ser eliminado porque não contém o mínimo. Nesse caso, x_2 se torna o novo valor de x_l para a próxima rodada.
2. Se $f(x_2) < f(x_1)$, então $f(x_2)$ é o mínimo, e o domínio de x à direita de x_1 , de x_1 a x_u , seria eliminado. Nesse caso, x_1 se torna o novo x_u para a próxima rodada.

Agora, destaca-se o real benefício do uso da razão áurea: como os valores originais de x_1 e x_2 foram escolhidos usando-se a razão áurea, não é preciso recalcular todos os valores da função para a próxima iteração. Por exemplo, no caso ilustrado na Figura 7.6, o antigo x_1 se torna o novo x_2 . Isso significa que já se tem o valor para o novo $f(x_2)$, pois é o mesmo que o valor da função no antigo x_1 .

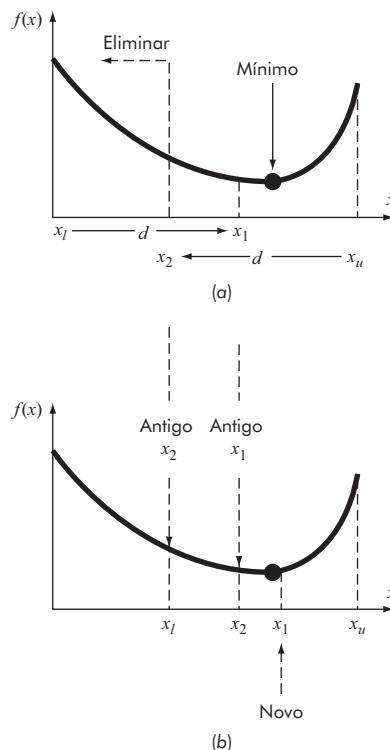


FIGURA 7.6 (a) O passo inicial do algoritmo de busca da razão áurea envolve a escolha de dois pontos interiores segundo a razão áurea. (b) O segundo passo envolve a definição de um novo intervalo que inclui o ponto ótimo.

Para completar o algoritmo, é necessário apenas determinar o novo x_1 , o que é feito com a Equação (7.6) com d calculado com a Equação (7.8) com base nos valores novos de x_l e x_u . Uma abordagem similar seria usada para o caso alternativo em que o ponto ótimo cai no subintervalo da esquerda. Para esse caso, o novo x_2 seria calculado com a Equação (7.7).

À medida as iterações são repetidas, o intervalo que contém o ponto extremo é reduzido rapidamente: a cada rodada o intervalo é reduzido por um fator de $\phi - 1$ (cerca de 61,8%). Isso significa que, depois de 10 rodadas, o intervalo foi encolhido de cerca de 0,618¹⁰ ou 0,008 ou 0,8% do seu tamanho inicial. Depois de 20 rodadas, chega a 0,0066%. O resultado não é tão positivo quanto a redução obtida com a bissecção, mas esse é um problema muito mais difícil.

EXEMPLO 7.2 Busca da razão áurea

Use a busca da razão áurea para achar o mínimo de

$$f(x) = \frac{x^2}{10} - 2 \operatorname{sen} x$$

no intervalo entre $x_l = 0$ e $x_u = 4$.

Solução. A razão áurea é usada para criar os dois pontos interiores:

$$d = 0,61803(4 - 0) = 2,4721$$

$$x_1 = 0 + 2,4721 = 2,4721$$

$$x_2 = 4 - 2,4721 = 1,5279$$

A função pode ser calculada nos pontos interiores:

$$f(x_2) = \frac{1,5279^2}{10} - 2 \operatorname{sen}(1,5279) = -1,7647$$

$$f(x_1) = \frac{2,4721^2}{10} - 2 \operatorname{sen}(2,4721) = -0,6300$$

Uma vez que $f(x_2) < f(x_1)$, a melhor estimativa do mínimo nesse ponto é aquela que está localizada em $x = 1,5279$ com um valor de $f(x) = -1,7647$. Além disso, sabe-se também que o mínimo está no intervalo definido por x_l , x_2 e x_u . Assim, para a próxima iteração, a extremidade inferior permanece em $x_l = 0$ e x_1 torna-se a extremidade superior, isto é, $x_u = 2,4721$. Além disso, o valor antigo x_2 torna-se o novo x_1 , isto é, $x_1 = 1,5279$, e não é preciso recalcular $f(x_1)$ porque este foi determinado na iteração prévia como $f(1,5279) = -1,7647$.

Tudo o que resta fazer é usar as Equações (7.8) e (7.7) para calcular os novos valores de d e x_2 :

$$d = 0,61803(2,4721 - 0) = 1,5279$$

$$x_2 = 2,4721 - 1,5279 = 0,9443$$

O valor da função em x_2 é $f(0,9443) = -1,5310$. Como esse valor é menor que o valor da função em x_1 , o mínimo é $f(1,5279) = -1,7647$, e está no intervalo prescrito por x_2 , x_1 e x_u . O processo pode ser repetido, com os resultados tabulados a seguir:

i	x_l	$f(x_l)$	x_2	$f(x_2)$	x_1	$f(x_1)$	x_u	$f(x_u)$	d
1	0	0	1,5279	-1,7647	2,4721	-0,6300	4,0000	3,1136	2,4721
2	0	0	0,9443	-1,5310	1,5279	-1,7647	2,4721	-0,6300	1,5279
3	0,9443	-1,5310	1,5279	-1,7647	1,8885	-1,5432	2,4721	-0,6300	0,9443
4	0,9443	-1,5310	1,3050	-1,7595	1,5279	-1,7647	1,8885	-1,5432	0,5836
5	1,3050	-1,7595	1,5279	-1,7647	1,6656	-1,7136	1,8885	-1,5432	0,3607
6	1,3050	-1,7595	1,4427	-1,7755	1,5279	-1,7647	1,6656	-1,7136	0,2229
7	1,3050	-1,7595	1,3901	-1,7742	1,4427	-1,7755	1,5279	-1,7647	0,1378
8	1,3901	-1,7742	1,4427	-1,7755	1,4752	-1,7732	1,5279	-1,7647	0,0851

Observe que o mínimo corrente é destacado para cada iteração. Após a oitava iteração, o mínimo ocorre em $x = 1,4427$ com o valor da função de $-1,7755$, portanto, o resultado converge para o valor verdadeiro $-1,7757$ em $x = 1,4276$.

Lembre-se de que, para a bissecção (Seção 5.4), pode ser calculado um limite superior exato para o erro a cada iteração. Usando um raciocínio análogo, um limite superior para a busca da razão áurea pode ser deduzido da seguinte forma: uma vez que uma iteração é completada, o ponto ótimo cairá em um de dois intervalos. Se x_2

for o valor ótimo da função, ele estará no intervalo inferior (x_l, x_2, x_1); se x_1 for o valor ótimo da função, ele estará no intervalo superior (x_2, x_1, x_u). Como os pontos interiores são simétricos, qualquer um dos casos pode ser usado para definir o erro.

Olhando para o intervalo superior (x_2, x_1, x_u), se o valor verdadeiro estiver na extremidade esquerda, a distância máxima do valor estimado será

$$\begin{aligned}\Delta x_a &= x_1 - x_2 \\ &= x_l + (\phi - 1)(x_u - x_l) - x_u + (\phi - 1)(x_u - x_l) \\ &= (x_l - x_u) + 2(\phi - 1)(x_u - x_l) \\ &= (2\phi - 3)(x_u - x_l)\end{aligned}$$

ou $0,2361(x_u - x_l)$. Se o valor verdadeiro estiver na extremidade direita, a distância máxima do valor estimado será

$$\begin{aligned}\Delta x_b &= x_u - x_1 \\ &= x_u - x_l - (\phi - 1)(x_u - x_l) \\ &= (x_u - x_l) - (\phi - 1)(x_u - x_l) \\ &= (2 - \phi)(x_u - x_l)\end{aligned}$$

ou $0,3820(x_u - x_l)$. Portanto, esse caso representa o erro máximo. Esse resultado pode então ser normalizado para o valor ótimo para a iteração, x_{otm} , o que fornece

$$\varepsilon_a = (2 - \phi) \left| \frac{x_u - x_l}{x_{\text{otm}}} \right| \times 100\% \quad (7.9)$$

Essa estimativa proporciona uma base para o término das iterações.

Uma função do MATLAB para a busca da razão áurea para minimização é apresentada na Figura 7.7. A função retorna a localização do mínimo, o valor da função, o erro aproximado e o número de iterações, e pode ser usada para resolver o problema do Exemplo 7.1.

```
>> g=9.81; v0=55; m=80; c=15; z0=100;
>> z=@(t) -(z0+m/c*(v0+m*g/c)*(1-exp(-c/m*t))-m*g/c*t);
>> [xmin,fmin,ea]=aureamin(z,0,8)
xmin =
    3.8317
fmin =
   -192.8609
ea =
    6.9356e-005
```

Observe que, como esse é um problema de maximização, entrou-se com o negativo da Equação (7.1). Consequentemente, f_{min} corresponde à altura máxima de 192,8609.

Você deve estar imaginando por que foi enfatizado o número reduzido de cálculos da função na busca da razão áurea. Naturalmente, para resolver uma única otimização, o ganho em velocidade seria desprezível. Entretanto, existem dois importantes contextos nos quais a minimização do número de cálculos pode ser importante, explicitados a seguir:

```

function [x,fx,ea,iter]=aureamin(f,xl,xu,es,maxit,varargin)
% aureamin: busca da razão áurea para minimização
% [x,fx,ea,iter]=goldmin(f,xl,xu,es,maxit,p1,p2,...):
% usa a busca da razão áurea para encontrar o mínimo de f
% entrada:
% f = nome da função
% xl, xu = aproximações inferior e superior
% es = erro relativo desejado (padrão = 0.0001%)
% maxit = número máximo de iterações permitidas (padrão = 50)
% p1,p2,... = parâmetros adicionais usados por f
% saída:
% x = localização do mínimo
% fx = valor da função no mínimo
% ea = erro relativo aproximado (%)
% iter = número de iterações

if nargin<3,error('s  o necess  rios pelo menos 3 argumentos de entrada'),end
if nargin<4|isempty(es), es=0.0001;end
if nargin<5|isempty(maxit), maxit=50;end
phi=(1+sqrt(5))/2;
iter=0;
while(1)
    d = (phi-1)*(xu - xl);
    xl = xl + d;
    x2 = xu - d;
    if f(x1,varargin{:}) < f(x2,varargin{:})
        xopt = xl;
        xl = x2;
    else
        xopt = x2;
        xu = xl;
    end
    iter=iter+1;
    if xopt~=-0, ea = (2 - phi) * abs((xu - xl) / xopt) * 100;end
    if ea <= es | iter >= maxit,break,end
end
x=xopt;fx=f(xopt,varargin{:});

```

FIGURA 7.7 Fun  o do MATLAB para determinar o m  imo de uma fun  o com a busca da se  o ´urea.

1. Muitos c  culos. Em certos casos, o algoritmo de busca da raz  o ´urea pode ser parte de um c  culo muito maior, o que talvez exija mais chamadas dos algoritmos. Portanto, manter o c  culo da fun  o em um n  mero m  imo pode trazer grandes dividendos para esses casos.
2. C  culos demorados. Por raz  es pedag  icas, foram utilizadas fun  es simples na maioria dos exemplos.  preciso entender que uma fun  o pode ser muito complexa e levar muito tempo para ser calculada. Por exemplo, a otimiza  o pode ser utilizada para estimar os par  metros de um modelo que consiste em um sistema de equa  es diferenciais; nesses casos, a “fun  o” envolve a integra  o do modelo, que consome tempo. Qualquer m  todo que minimize esses c  culos torna-se vantajoso.

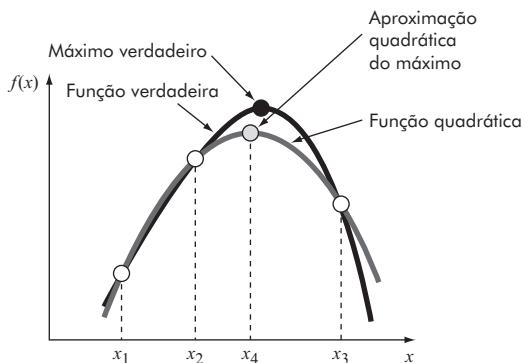


FIGURA 7.8 Descrição gráfica da interpolação quadrática.

7.2.2 Interpolação quadrática

A interpolação quadrática (ou parabólica) aproveita-se do fato de que um polinômio de segundo grau frequentemente fornece uma boa aproximação para a forma de $f(x)$ nas proximidades do ponto ótimo (Figura 7.8).

Assim como existe somente uma reta ligando dois pontos, há apenas uma função quadrática ou parábola ligando três pontos. Então, se houver três pontos que conjuntamente delimitam um ponto ótimo, pode-se ajustar uma parábola aos três pontos, e é possível, desse modo, derivá-la, igualar o resultado a zero e resolver para determinar uma estimativa do ponto ótimo x . Pode-se mostrar mediante um pouco de manipulação algébrica que o resultado é

$$x_4 = x_2 - \frac{1}{2} \frac{(x_2 - x_1)^2 [f(x_2) - f(x_3)] - (x_2 - x_3)^2 [f(x_2) - f(x_1)]}{(x_2 - x_1) [f(x_2) - f(x_3)] - (x_2 - x_3) [f(x_2) - f(x_1)]} \quad (7.10)$$

onde x_1, x_2 e x_3 são as aproximações iniciais e x_4 é o valor de x que corresponde ao valor ótimo da função quadrática ajustada às aproximações.

EXEMPLO 7.3 Interpolação quadrática

Use a interpolação quadrática para aproximar o ponto de mínimo de

$$f(x) = \frac{x^2}{10} - 2 \operatorname{sen} x$$

com aproximações iniciais $x_1 = 0, x_2 = 1$ e $x_3 = 4$.

Solução. Os valores da função nas três aproximações são dados por

$$\begin{array}{ll} x_1 = 0 & f(x_1) = 0 \\ x_2 = 1 & f(x_2) = -1,5829 \\ x_3 = 4 & f(x_3) = 3,1136 \end{array}$$

o que, substituindo na Equação (7.10), fornece

$$x_4 = 1 - \frac{1}{2} \frac{(1-0)^2 [-1,5829 - 3,1136] - (1-4)^2 [-1,5829 - 0]}{(1-0)[-1,5829 - 3,1136] - (1-4)[-1,5829 - 0]} = 1,5055$$

que tem o valor da função de $f(1,5055) = -1,7691$.

Em seguida, uma estratégia parecida àquela da busca da razão áurea pode ser empregada para determinar qual ponto deve ser descartado. Uma vez que o valor da função para o novo ponto é menor do que para o ponto intermediário (x_2) e o novo valor de x está à direita do ponto intermediário, a aproximação inferior (x_0) é descartada. Portanto, para a próxima iteração:

$$\begin{array}{ll} x_1 = 1 & f(x_1) = -1,5829 \\ x_2 = 1,5055 & f(x_2) = -1,7691 \\ x_3 = 4 & f(x_3) = 3,1136 \end{array}$$

o que pode ser substituído na Equação (7.10) para fornecer

$$x_4 = 1,5055 - \frac{1}{2} \frac{(1,5055 - 1)^2 [-1,7691 - 3,1136] - (1,5055 - 4)^2 [-1,7691 - (-1,5829)]}{(1,5055 - 1)[-1,7691 - 3,1136] - (1,5055 - 4)[-1,7691 - (-1,5829)]} = 1,4903$$

que tem o valor da função de $f(1,4903) = -1,7714$. O processo pode ser repetido, com os resultados tabulados a seguir:

i	x_1	$f(x_1)$	x_2	$f(x_2)$	x_3	$f(x_3)$	x_4	$f(x_4)$
1	0,0000	0,0000	1,0000	-1,5829	4,0000	3,1136	1,5055	-1,7691
2	1,0000	-1,5829	1,5055	-1,7691	4,0000	3,1136	1,4903	-1,7714
3	1,0000	-1,5829	1,4903	-1,7714	1,5055	-1,7691	1,4256	-1,7757
4	1,0000	-1,5829	1,4256	-1,7757	1,4903	-1,7714	1,4266	-1,7757
5	1,4256	-1,7757	1,4266	-1,7757	1,4903	-1,7714	1,4275	-1,7757

Assim, em cinco iterações, o resultado converge rapidamente para o valor verdadeiro de -1,7757 em $x = 1,4276$.

7.2.3 Função do MATLAB: `fminbnd`

Vimos na Seção 6.4 a descrição do método de Brent para localização de raízes, que combinava vários métodos de determinação de raízes em um único algoritmo que equilibrava confiabilidade e eficiência. Em função dessas qualidades, esse algoritmo forma a base para a função nativa do MATLAB `fzero`.

Brent também desenvolveu uma abordagem similar para a minimização unidimensional que forma a base da função `fminbnd` do MATLAB. Ela combina a busca da seção áurea, lenta e confiável, com a interpolação quadrática, mais rápida, porém não confiável. A primeira tentativa do algoritmo é a interpolação quadrática, aplicada continuamente enquanto resultados aceitáveis forem obtidos. Caso contrário, a busca da seção áurea é utilizada para obter os resultados.

Uma expressão simples de sua sintaxe é

```
[xmin, fval] = fminbnd(funcao,x1,x2)
```

onde x e $fval$ são, respectivamente, a localização e o valor do mínimo, $funcao$ é o nome da função sendo avaliada e $x1$ e $x2$ são as extremidades do intervalo em que a busca é realizada.

As linhas de código a seguir apresentam uma sessão do MATLAB simples que usa a função `fminbnd` para resolver o problema do Exemplo 7.1.

```
>> g=9.81; v0=55; m=80; c=15; z0=100;
>> z=@(t) -(z0+m/c*(v0+m*g/c)*(1-exp(-c/m*t))-m*g/c*t);
>> [x,f]=fminbnd(z,0,8)

x =
    3.8317
f =
   -192.8609
```

De modo análogo à função `fzero`, parâmetros opcionais podem ser especificados utilizando a função `optimset`. Por exemplo, os detalhes do cálculo podem ser exibidos:

```
>> opcoes = optimset('display','iter');
>> fminbnd(z,0,8,opcoes)
      Func-count      x          f(x)      Procedure
            1    3.05573    -189.759    initial
            2    4.94427    -187.19     golden
            3    1.88854    -171.871     golden
            4    3.87544    -192.851    parabolic
            5    3.85836    -192.857    parabolic
            6    3.83332    -192.861    parabolic
            7    3.83162    -192.861    parabolic
            8    3.83166    -192.861    parabolic
            9    3.83169    -192.861    parabolic

Optimization terminated:
the current x satisfies the termination criteria using
OPTIONS.TolX of 1.000000e-004
ans =
    3.8317
```

Assim, depois de três iterações, o método muda da busca da razão áurea para a interpolação parabólica, e, após oito iterações, o mínimo é determinado com uma tolerância de 0,0001.

7.3 OTIMIZAÇÃO MULTIDIMENSIONAL

Além das funções unidimensionais, a otimização também aborda funções multidimensionais. Na Figura 7.3a, nossa imagem pictórica de uma busca unidimensional era uma montanha-russa; para o caso bidimensional, a imagem tornou-se a de montanhas e vales (Figura 7.3b). Como no exemplo a seguir, os recursos gráficos do MATLAB fornecem meios práticos para visualizar tais funções.

EXEMPLO 7.4 Visualizando uma função bidimensional

Use os recursos gráficos do MATLAB para exibir a seguinte função e estimar visualmente seu mínimo no intervalo $-2 \leq x_1 \leq 0$ e $0 \leq x_2 \leq 3$:

$$f(x_1, x_2) = 2 + x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

Solução. O seguinte programa de MATLAB gera gráficos de curvas de nível e de malhas da função:

```
x=linspace(-2,0,40); y=linspace(0,3,40);
[X,Y] = meshgrid(x,y);
Z=2+X-Y+2*X.^2+2*X.*Y+Y.^2;
subplot(1,2,1);
cs=contour(X,Y,Z); clabel(cs);
xlabel('x _ 1'); ylabel('x _ 2');
title('(a) Contour plot'); grid;
subplot(1,2,2);
cs=surfc(X,Y,Z);
zmin=foor(min(Z));
zmax=ceil(max(Z));
xlabel('x _ 1'); ylabel('x _ 2'); zlabel('f(x _ 1,x _ 2)');
title('(b) Mesh plot');
```

Como ilustrado na Figura 7.9, ambos os gráficos indicam que a função tem um valor mínimo de cerca de $f(x_1, x_2) = 0$ a 1, localizado em aproximadamente $x_1 = -1$ e $x_2 = 1,5$.

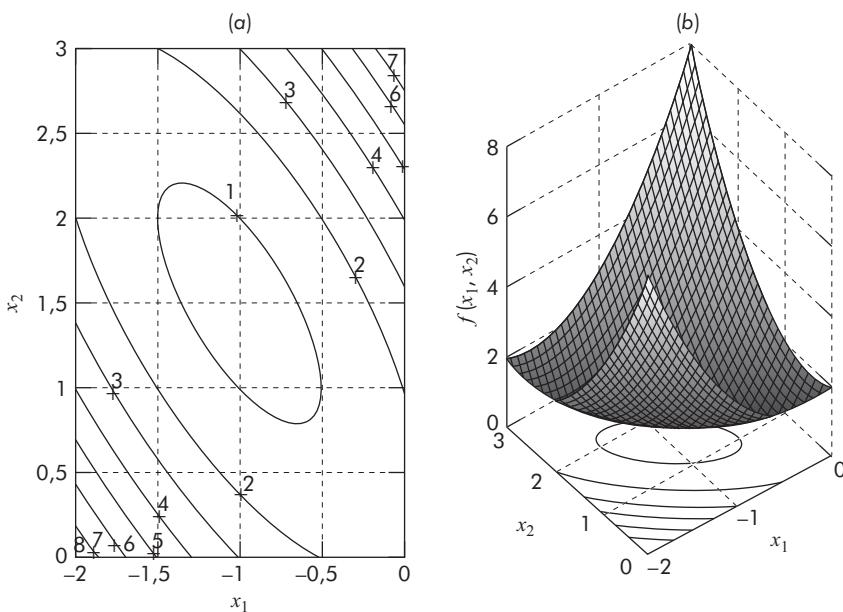


FIGURA 7.9 Gráficos (a) de curvas de nível e (b) de malhas de uma função bidimensional.

As técnicas para otimização multidimensional sem restrições podem ser classificadas de diversas maneiras. Para o propósito da presente discussão, elas serão divididas dependendo do fato de exigirem ou não o cálculo da derivada: aquelas que exigem derivadas são chamadas *métodos gradientes* ou de *declive* (ou *active*); as que não exigem o cálculo da derivada são chamadas de *não gradientes* ou *métodos diretos*. Como descrito a seguir, a função nativa do MATLAB `fminsearch` é um método direto.

7.3.1 Função do MATLAB: `fminsearch`

O MATLAB padrão possui uma função, `fminsearch`, que pode ser usada para determinar o mínimo de uma função multidimensional. Ela é baseada no método de Nelder-Mead, um método direto de busca que usa apenas valores da função (não exige derivadas) e lida com funções objetivo não suaves. Uma expressão simples de sua sintaxe é

```
[xmin, fval] = fminsearch(funcão,x0)
```

onde `xmin` e `fval` são a localização e o valor do mínimo, `funcão` é o nome da função avaliada e `x0` é a aproximação inicial. Observe que `x0` pode ser um escalar, um vetor ou uma matriz.

As linhas de código a seguir apresentam uma sessão do MATLAB simples que usa a função `fminsearch` para a função traçada no Exemplo 7.4:

```
>> f=@(x) 2+x(1)-x(2)+2*x(1)^2+2*x(1)*x(2)+x(2)^2;
>> [x,fval]=fminsearch(f,[-0.5,0.5])
x =
    -1.0000    1.5000
fval =
    0.7500
```

► 7.4 ESTUDO DE CASO EQUILÍBRIO E ENERGIA POTENCIAL MÍNIMA

Contextualização. Como mostra a Figura 7.10a, uma mola sem carga pode ser conectada a um suporte preso a uma parede. Quando uma força horizontal é aplicada, a mola se estende. O deslocamento está relacionado à força pela *lei de Hooke*, $F = kx$. A *energia potencial* do estado deformado consiste da diferença entre a energia de deformação da mola e o trabalho realizado pela força:

$$PE(x) = 0,5kx^2 - Fx \quad (7.11)$$

A Equação (7.11) define uma parábola. Considerando que a energia potencial estará em um mínimo no equilíbrio, a solução para o deslocamento pode ser vista como um problema de otimização unidimensional. Como a derivada da equação é fácil de ser calculada, pode-se resolver para o deslocamento como $x = F/k$. Por exemplo, se $k = 2 \text{ N/cm}$ e $F = 5 \text{ N}$, $x = 5 \text{ N}/(2 \text{ N/cm}) = 2,5 \text{ cm}$.



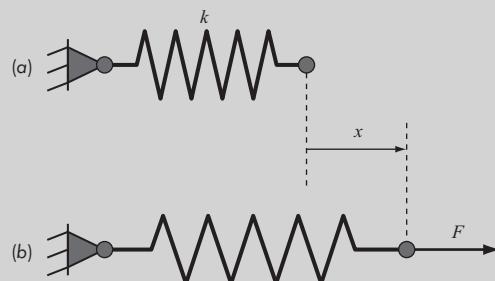


FIGURA 7.10 (a) Mola sem carga conectada a um suporte de parede. (b) A aplicação de uma força horizontal estende a mola, sendo que a relação entre a força e o deslocamento é descrita pela lei de Hooke.

Um caso bidimensional mais interessante é mostrado na Figura 7.11: nesse sistema, há dois graus de liberdade, de modo que ele pode se mover horizontalmente e verticalmente. Do mesmo modo que abordamos o sistema unidimensional, as deformações de equilíbrio são os valores de x_1 e x_2 que minimizam a energia potencial:

$$\begin{aligned} PE(x_1, x_2) = & 0,5k_a \left(\sqrt{x_1^2 + (L_a - x_2)^2} - L_a \right)^2 \\ & + 0,5k_b \left(\sqrt{x_1^2 + (L_b + x_2)^2} - L_b \right)^2 - F_1 x_1 - F_2 x_2 \end{aligned} \quad (7.12)$$

Se os parâmetros são $k_a = 9$ N/cm, $k_b = 2$ N/cm, $L_a = 10$ cm, $L_b = 10$ cm, $F_1 = 2$ N e $F_2 = 4$ N, utilize o MATLAB para calcular o deslocamento e a energia potencial.

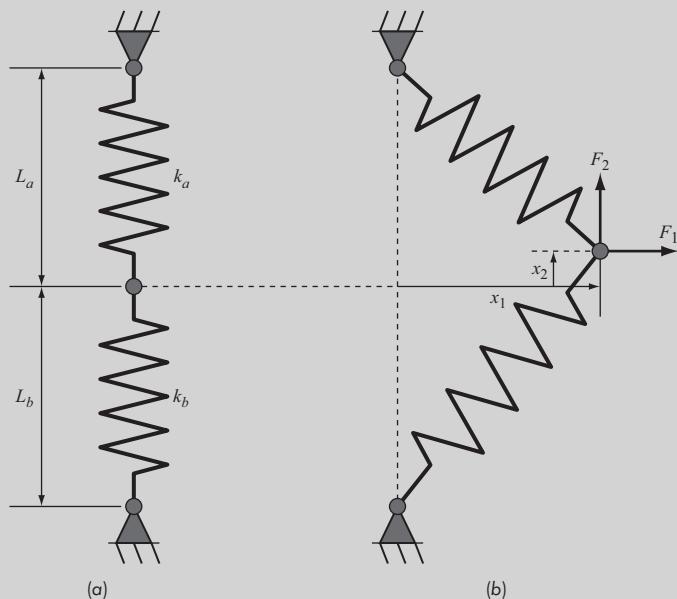


FIGURA 7.11 Sistema de duas molas: (a) sem carga e (b) com carga.

Solução. Uma função do MATLAB pode ser implementada para armazenar a função da energia potencial:

```
function p=PE(x, ka, kb, La, Lb, F1, F2)
PEa=0.5*ka*(sqrt(x(1)^2+(La-x(2))^2)-La)^2;
PEb=0.5*kb*(sqrt(x(1)^2+(Lb+x(2))^2)-Lb)^2;
W=F1*x(1)+F2*x(2);
p=PEa+PEb-W;
```

A solução pode ser obtida com a função `fminsearch`:

```
>> ka=9; kb=2; La=10; Lb=10; F1=2; F2=4;
>> [x, f] = fminsearch(@PE, [-0.5, 0.5], [], ka, kb, La, Lb, F1, F2)
x =
    4.9523 1.2769
f =
    -9.6422
```

Assim, no equilíbrio, a energia potencial é $-9,6422 \text{ N} \cdot \text{cm}$. O ponto de conexão (ponto de aplicação da força) está localizado 4,9523 cm à direita e 1,2759 acima de sua posição original.

►► PROBLEMAS

7.1 Efetue três iterações do método de Newton-Raphson para determinar a raiz da Equação (E7.1.1). Use os valores dos parâmetros do Exemplo 7.1 juntamente com uma aproximação inicial de $t = 3 \text{ s}$.

7.2 Dada a fórmula

$$f(x) = -x^2 + 8x - 12$$

- (a) Determine o máximo e o valor correspondente de x para essa função analiticamente (isto é, utilizando derivação).
- (b) Verifique que a Equação (7.10) produz os mesmos resultados com base nas aproximações iniciais de $x_1 = 0$, $x_2 = 2$ e $x_3 = 6$.

7.3 Considere a seguinte função:

$$f(x) = 3 + 6x + 5x^2 + 3x^3 + 4x^4$$

Localize o mínimo pelo cálculo da raiz da derivada dessa função. Use a bissecção com aproximações iniciais de $x_l = -2$ e $x_u = 1$.

7.4 Dado

$$f(x) = -1,5x^6 - 2x^4 + 12x$$

- (a) Faça um gráfico da função.

(b) Use métodos analíticos para provar que a função é côncava para todos os valores de x .

(c) Derive a função e então use um método de localização de raízes para determinar o máximo de $f(x)$ e o valor correspondente de x .

7.5 Ache o valor de x que maximiza a função $f(x)$ no Problema 7.4 utilizando a busca da razão áurea. Empregue aproximações iniciais de $x_l = 0$ e $x_u = 2$ e efetue três iterações.

7.6 Repita o Problema 7.5, porém utilize interpolação quadrática. Empregue aproximações iniciais de $x_1 = 0$, $x_2 = 1$ e $x_3 = 2$ e efetue três iterações.

7.7 Empregue os seguintes métodos para encontrar o máximo de

$$f(x) = 4x - 1,8x^2 + 1,2x^3 - 0,3x^4$$

- (a) Busca da razão áurea ($x_l = -2$, $x_u = 4$, $\varepsilon_s = 1\%$).
- (b) Interpolação quadrática ($x_1 = 1,75$, $x_2 = 2$, $x_3 = 2,5$, iterações = 5).

7.8 Considere a seguinte função

$$f(x) = x^4 + 2x^3 + 8x^2 + 5x$$

Use métodos analíticos e gráficos para mostrar que a função tem um mínimo para algum valor de x no intervalo $-2 \leq x \leq 1$.

7.9 Empregue os seguintes métodos para encontrar o mínimo da função do Problema 7.8:

- Busca da razão áurea ($x_1 = -2, x_u = 1, \varepsilon_s = 1\%$).
- Interpolação quadrática ($x_1 = -2, x_2 = -1, x_3 = 1$, iterações = 5).

7.10 Considere a seguinte função:

$$f(x) = 2x + \frac{3}{x}$$

Faça 10 iterações da interpolação quadrática para localizar o mínimo e comente sobre a convergência dos seus resultados ($x_1 = 0,1; x_2 = 0,5; x_3 = 5$).

7.11 Desenvolva um único programa no MATLAB para (a) gerar gráficos de curvas de nível e de malhas (similares aos do Exemplo 7.4) em uma mesma janela (*subplots*) para o campo de temperaturas a seguir:

$$T(x, y) = 2x^2 + 3y^2 - 4xy - y - 3x$$

e (b) determinar o mínimo com a função `fminsearch`.

7.12 A superfície de um aquífero de águas subterrâneas é descrita por

$$h(x, y) = \frac{1}{1 + x^2 + y^2 + x + xy}$$

Desenvolva um único programa no MATLAB para (a) gerar gráficos de curvas de nível e de malhas (similares aos do Exemplo 7.4) em uma mesma janela (*subplots*) da função anterior e (b) determinar o máximo com a função `fminsearch`.

7.13 O interesse recente no ciclismo competitivo e recreativo tem obrigado os engenheiros a direcionar suas habilidades para o projeto e teste de *mountain bikes* (Figura P7.13a). Considere que você foi escalado para prever o deslocamento horizontal e vertical do quadro da bicicleta em resposta a uma força. Assuma que as forças que você deve analisar possam ser simplificadas, como descrito na Figura P7.13b. Você está interessado em testar a resposta do suporte a uma força exercida em uma direção qualquer, designada pelo ângulo θ . Os parâmetros para o problema são E = módulo de Young = 2×10^{11} Pa, A = área da seção transversal = $0,0001 \text{ m}^2$, w = largura = $0,44 \text{ m}$, ℓ = comprimento = $0,56 \text{ m}$ e h = altura = $0,5 \text{ m}$. Os deslocamentos x e y podem ser calculados determinando-se os valores que resultam na energia potencial mínima. Determine os deslocamentos para uma força de 10.000 N e um intervalo de θ 's de 0° (horizontal) a 90° (vertical).

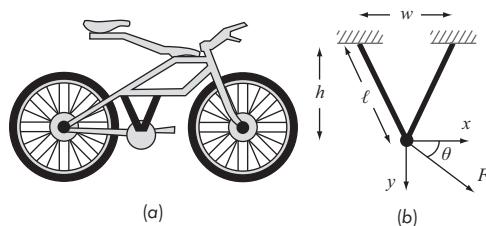


FIGURA P7.13 (a) Uma *mountain bike* junto do (b) diagrama de corpo livre para uma parte da estrutura.

7.14 À medida que uma corrente elétrica circula através de um fio (Figura P7.14), o calor gerado pela resistência é conduzido através de uma camada de isolamento e então transferido para o ar circundante, essencialmente, por convecção. A temperatura de regime permanente do fio pode ser calculada como

$$T = T_{\text{ar}} + \frac{q}{2\pi} \left[\frac{1}{k} \ln \left(\frac{r_w + r_i}{r_w} \right) + \frac{1}{h} \frac{1}{r_w + r_i} \right]$$

Determine a espessura do isolamento r_i (m) que minimiza a temperatura do fio dados os seguintes parâmetros: q = taxa de geração de calor = 75 W/m , r_w = raio do fio = 6 mm , k = condutividade térmica do isolamento = $0,17 \text{ W/(m K)}$, h = coeficiente de transferência de calor por convecção = $12 \text{ W(m}^2 \text{ K)}$ e T_{ar} = temperatura do ar = 293 K .

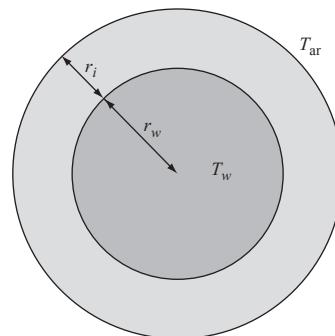


FIGURA P7.14 Seção transversal de um fio isolado.

7.15 Desenvolva um programa no MATLAB que seja expressamente projetado para localizar um máximo com a busca da razão áurea. Em outras palavras, desenvolva-o para que encontre diretamente o máximo em vez de encontrar o mínimo de $-f(x)$. A função deve ter as seguintes propriedades:

- ✓ Iterar até que o erro relativo f que abaixo de um critério de parada ou exceda um número máximo de iterações.
- ✓ Retorne tanto o ótimo de x quanto o valor de $f(x)$.

Teste seu programa com o mesmo problema do Exemplo 7.1.

7.16 Desenvolva um programa no MATLAB para localizar um mínimo com a busca da seção áurea. Em vez de utilizar o máximo de iterações e a Equação (7.9) como critérios de parada, determine o número de iterações necessárias para atingir uma tolerância desejada. Teste seu programa resolvendo o Exemplo 7.2 utilizando $E_{a,d} = 0,0001$.

7.17 Desenvolva um programa no MATLAB para implementar a interpolação quadrática para localizar um mínimo. A função deve ter as seguintes propriedades:

- ✓ Ser baseada em duas aproximações iniciais e fazer o programa gerar o terceiro valor inicial no ponto médio do intervalo.
- ✓ Verificar se as aproximações delimitam um máximo. Em caso negativo, a função não deve implementar o algoritmo e deve retornar uma mensagem de erro.
- ✓ Iterar até que o erro relativo f que abaixo de um critério de parada ou exceda um número máximo de iterações.
- ✓ Retornar tanto o ponto ótimo x quanto o valor de $f(x)$.

Teste seu programa com o mesmo problema do Exemplo 7.3.

7.18 São feitas medidas de pressão em certos pontos atrás de um aerofólio durante um certo tempo. Os dados se ajustam à curva $y = 6 \cos x - 1,5 \sin x$ de $x = 0$ a 6 s. Use quatro iterações do método da razão áurea para achar a pressão mínima. Adote $x_l = 2$ e $x_u = 4$.

7.19 A trajetória de uma bola pode ser calculada como

$$y = (\tan \theta_0)x - \frac{g}{2v_0^2 \cos^2 \theta_0}x^2 + y_0$$

onde y é a altura (m), θ_0 é o ângulo inicial (radianos), v_0 é a velocidade inicial (m/s), $g = 9,81$ m/s² é a aceleração da gravidade e y_0 é a altura inicial (m). Use a busca da razão áurea para determinar a altura máxima, dados $y_0 = 1$ m, $v_0 = 25$ m/s e $\theta_0 = 50^\circ$. Itere até que o erro aproximado caia

abaixo de $\varepsilon_s = 1\%$ usando aproximações iniciais $x_l = 0$ e $x_u = 60$ m.

7.20 A deflexão de uma viga uniforme submetida a uma carga crescente e linearmente distribuída pode ser escrita como

$$y = \frac{w_0}{120EI}(-x^5 + 2L^2x^3 - L^4x)$$

Dado que $L = 600$ cm, $E = 50.000$ kN/cm², $I = 30.000$ cm⁴ e $w_0 = 2,5$ kN/cm, determine o ponto de máxima deflexão (*a*) graficamente, (*b*) usando a busca da razão áurea até que o erro aproximado fique abaixo de $\varepsilon_s = 1\%$, com aproximações iniciais $x_l = 0$ e $x_u = L$.

7.21 Um objeto com uma massa de 90 kg é arremessado para cima a partir de um ponto na superfície da Terra, com uma velocidade de 60 m/s. Se o objeto está submetido a um arraste linear ($c = 15$ kg/s), use a busca da razão áurea para determinar qual a altura máxima que o objeto atinge.

7.22 A distribuição normal é uma curva em forma de sino definida por

$$y = e^{-x^2}$$

Use a busca da razão áurea para determinar a posição do ponto de inflexão nessa curva para x positivo.

7.23 Use a função `fminsearch` para determinar o máximo de

$$f(x, y) = 2y^2 - 2,25xy - 1,75y + 1,5x^2$$

7.24 Use a função `fminsearch` para determinar o máximo de

$$f(x, y) = 4x + 2y + x^2 - 2x^4 + 2xy - 3y^2$$

7.25 Dada a seguinte função:

$$f(x, y) = -8x + x^2 + 12y + 4y^2 - 2xy$$

Determine o mínimo (*a*) graficamente, (*b*) numericamente com a função `fminsearch`, e (*c*) substitua o resultado de (*b*) de volta na função para determinar o mínimo de $f(x, y)$.

7.26 A taxa de crescimento específico de uma cultura que produz um antibiótico é uma função da concentração de alimento c :

$$g = \frac{2c}{4 + 0,8c + c^2 + 0,2c^3}$$

Como mostra a Figura P7.26, o crescimento tende a zero em concentrações muito baixas em decorrência da limitação de alimento, e também tende a zero em concentrações muito altas por causa do

efeito de toxinas (toxidades). Encontre o valor de c no qual o crescimento é máximo.

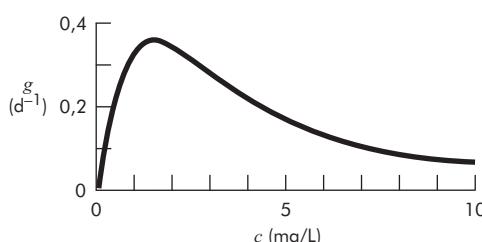


FIGURA P7.26 Taxa de crescimento específico de uma cultura que produz antibiótico versus a concentração de alimento.

7.27 Um composto A será convertido em B em um reator tanque agitado. O produto B e o produto A não processado são purificados em uma unidade de separação. O produto A não processado é reciclado para o reator. Um engenheiro de processo descobriu que o custo inicial do sistema é função da taxa de conversão x_A . Encontre a taxa de conversão que resultará no mais baixo custo do sistema. C é uma constante de proporcionalidade.

$$\text{Custo} = C \left[\left(\frac{1}{(1-x_A)^2} \right)^{0,6} + 6 \left(\frac{1}{x_A} \right)^{0,6} \right]$$

7.28 Um modelo de elementos finitos de uma viga em balanço sujeita a carregamento e momentos (Figura P7.28) é dado pela otimização de

$$f(x, y) = 5x^2 - 5xy + 2,5y^2 - x - 1,5y$$

onde x é o deslocamento da extremidade e y é o momento da extremidade. Encontre os valores de x e y que minimizam $f(x, y)$.

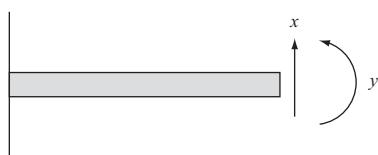


FIGURA P7.28 Uma viga em balanço.

7.29 O modelo de Streeter-Phelps pode ser usado para calcular a concentração de oxigênio dissolvido em um rio abaixo de um ponto de descarga de esgotos (Figura P7.29),

$$o = o_s - \frac{k_d L_o}{k_d + k_s - k_a} (e^{-k_a t} - e^{-(k_d + k_s)t}) - \frac{S_b}{k_a} (1 - e^{-k_a t}) \quad (\text{P7.29})$$

onde o é a concentração de oxigênio dissolvido (mg/L), o_s é a concentração de saturação do oxigênio (mg/L), t é o tempo de percurso (d), L_o é a concentração da demanda bioquímica de oxigênio (BOD) no ponto de mistura (mg/L), k_d é a taxa de decomposição de BOD (d⁻¹), k_s é a taxa de depósito de BOD (d⁻¹), k_a é a taxa de rearejamento (d⁻¹) e S_b é a demanda de oxigênio sedimentado (mg/L/d).

Como mostra a Figura P7.29, a Equação (P7.29) produz uma queda do oxigênio que atinge um nível crucial mínimo o_c para um tempo de percurso t_c abaixo do ponto de descarga. Esse ponto é chamado “crucial” porque representa a posição na qual o biosistema que depende de oxigênio (como os peixes) seria mais pressionado. Determine o tempo crucial de percurso e a concentração crítica, dados os seguintes valores:

$$\begin{aligned} o_s &= 10 \text{ mg/L} & k_d &= 0,1 \text{ d}^{-1} & k_a &= 0,6 \text{ d}^{-1} \\ k_s &= 0,05 \text{ d}^{-1} & L_o &= 50 \text{ mg/L} & S_b &= 1 \text{ mg/L/d} \end{aligned}$$

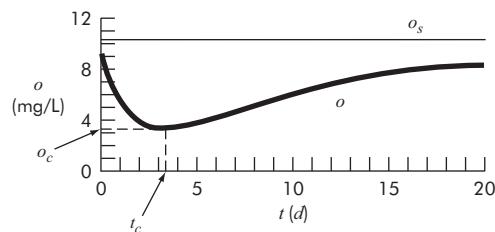


FIGURA P7.29 “Afundamento” de oxigênio dissolvido abaixo de um ponto de descarga de esgotos em um rio.

7.30 A distribuição bidimensional da concentração de poluentes em um canal pode ser descrita por

$$\begin{aligned} c(x, y) &= 7,9 + 0,13x + 0,21y - 0,05x^2 \\ &\quad - 0,016y^2 - 0,007xy \end{aligned}$$

Determine a localização exata do pico de concentração dados a função e o conhecimento de que o pico fica entre os limites $-10 \leq x \leq 10$ e $0 \leq y \leq 20$.

7.31 Uma carga total Q está uniformemente distribuída em um condutor na forma de um anel de raio a . Uma carga q está situada a uma distância x do centro do anel (Figura P7.31). A força exercida na carga pelo anel é dada por

$$F = \frac{1}{4\pi e_0} \frac{q Q x}{(x^2 + a^2)^{3/2}}$$

onde $e_0 = 8,85 \times 10^{-12} \text{ C}^2/(\text{N m}^2)$, $q = Q = 2 \times 10^{-5} \text{ C}$ e $a = 0,9 \text{ m}$. Determine a distância x onde a força é máxima.

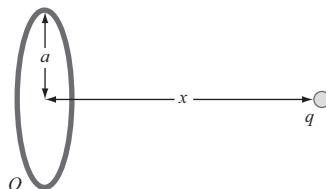


FIGURA P7.31

7.32 O torque transmitido a um motor de indução é uma função do escorregamento, s , entre a velocidade de rotação do campo do estator e a velocidade do rotor, com o escorregamento definido como

$$S = \frac{n - n_R}{n}$$

onde n é a velocidade de rotação do campo do estator em rotações por segundo e n_R é a velocidade do rotor. As leis de Kirchhoff podem ser usadas para mostrar que o torque (expresso em uma forma adimensional) e o escorregamento estão relacionados por

$$T = \frac{15s(1-s)}{(1-s)(4s^2 - 3s + 4)}$$

A Figura P7.32 mostra essa função. Use um método numérico para determinar o escorregamento no qual ocorre o torque máximo.

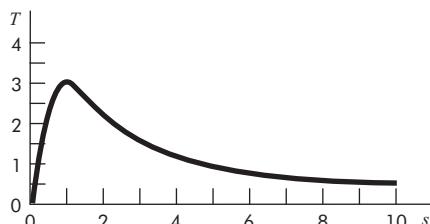


FIGURA P7.32 Torque transmitido a um motor de indução em função do escorregamento.

7.33 O arraste total de um aerofólio pode ser estimado por

$$D = 0,01\sigma V^2 + \frac{0,95}{\sigma} \left(\frac{W}{V} \right)^2$$

Atrito Sustentacão

onde D é o arraste, σ é a razão da densidade do ar entre a altitude de voo e o nível do mar, W é o peso e V é a velocidade. Como mostra a Figura P7.33, os dois fatores que contribuem para o arraste são afetados de forma diferente por um aumento de velocidade. Enquanto o arraste por atrito aumenta com a velocidade, o arraste decorrente da sustentação decresce. A combinação desses dois fatores leva a um arraste mínimo.

- (a) Se $\sigma = 0,6$ e $W = 16.000$, determine o arraste mínimo e a velocidade na qual ele ocorre.

(b) Desenvolva uma análise de sensibilidade para determinar como esse ponto ótimo varia em resposta a uma variação de $W = 12.000$ a 20.000 com $\sigma = 0,6$.

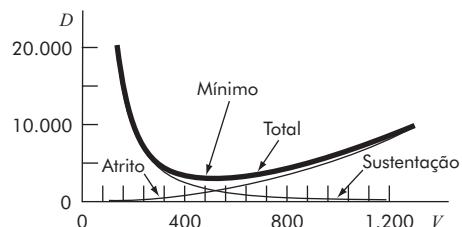


FIGURA P7.33 Gráfico do arraste versus a velocidade para um aerofólio.

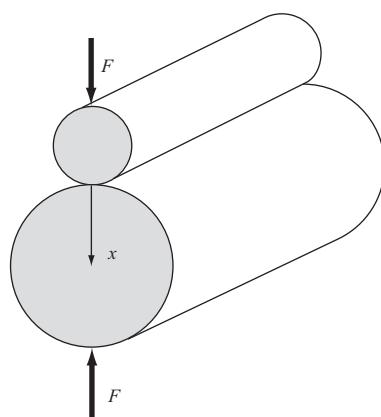


FIGURA P7-34 Rolamentos de rolos.

7.34 Os rolamentos de rolos cilíndricos da Figura P7.34 estão sujeitos à falha por fadiga causada por grandes cargas de contato F . Pode-se mostrar que o problema de determinar a localização da tensão máxima ao longo do eixo x é equivalente a maximizar a função

$$f(x) = \frac{0,4}{\sqrt{1+x^2}} - \sqrt{1+x^2} \left(1 - \frac{0,4}{1+x^2}\right) + x$$

Encontre o x que maximiza $f(x)$.

7.35 De uma maneira semelhante ao estudo de caso descrito na Seção 7.4, desenvolva a função de energia potencial para o sistema ilustrado na Figura P7.35. Desenvolva gráficos de curvas de nível e de superfícies no MATLAB. Minimize a função de energia potencial para determinar os deslocamentos de equilíbrio x_1 e x_2 , dada a força $F = 100$ N e os parâmetros $k_a = 20$ e $k_b = 15$ N/m.

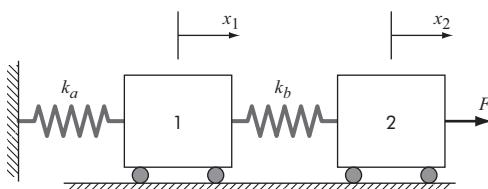


FIGURA P7.35 Duas massas sem atrito conectadas a uma parede por um par de molas elásticas lineares.

7.36 Como um engenheiro agrícola, você deve projetar um canal trapezoidal aberto para transportar água de irrigação (Figura P7.36). Determine as dimensões ótimas para minimizar o perímetro molhado para uma área da seção transversal de 50 m^2 . As dimensões relativas são universais?

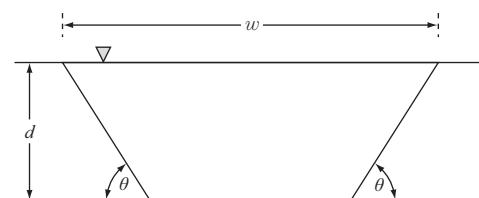


FIGURA P7.36

7.37 Use a função `fminsearch` para determinar o comprimento da menor escada que chega, a

partir do solo e apoiada no muro, à parede da construção (Figura P7.37). Faça um teste para o caso em que $h = d = 4$ m.

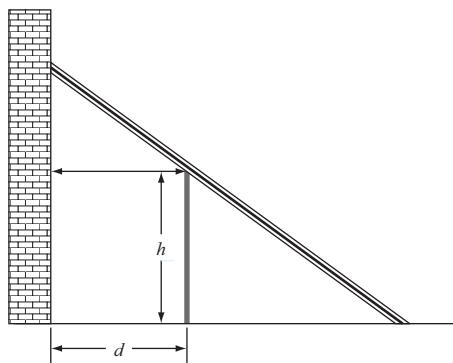


FIGURA P7.37 Escada encostada em um muro e tocando uma parede.

7.38 O comprimento da maior escada que pode passar pela quina detalhada na Figura P7.38 pode ser determinado calculando o valor de θ , que minimiza a seguinte função:

$$L(\theta) = \frac{w_1}{\sin \theta} + \frac{w_2}{\sin(\pi - \alpha - \theta)}$$

Para o caso em que $w_1 = w_2 = 2$ m, use um método numérico descrito neste capítulo (incluindo os recursos nativos do MATLAB) para desenvolver um gráfico de L versus um intervalo de α 's de 45 a 135°.

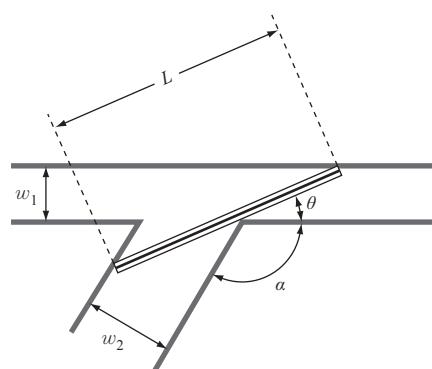
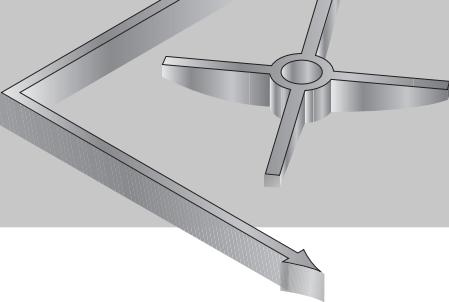


FIGURA P7.38 Escada passando por uma quina formada por dois corredores.



Parte 3

Sistemas lineares

3.1 VISÃO GERAL

O que são equações algébricas lineares?

Na Parte 2, foi determinado o valor de x que satisfaz uma única equação, $f(x) = 0$. Agora, será abordada a determinação dos valores x_1, x_2, \dots, x_n que satisfazem simultaneamente um conjunto de equações:

$$\begin{aligned}f_1(x_1, x_2, \dots, x_n) &= 0 \\f_2(x_1, x_2, \dots, x_n) &= 0 \\&\vdots \\&\vdots \\f_n(x_1, x_2, \dots, x_n) &= 0\end{aligned}$$

Esses sistemas podem ser ou lineares ou não lineares. Na Parte 3, trataremos basicamente das *equações algébricas lineares* que têm a forma geral

$$\begin{aligned}a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\&\vdots \\&\vdots \\a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n\end{aligned}\tag{PT3.1}$$

onde os a 's são coeficientes constantes, os b 's são constantes, os x 's são as incógnitas e n é o número de equações. Todas as outras equações são não lineares.

Equações algébricas lineares e a engenharia e a ciência

Muitas das equações fundamentais da engenharia e da ciência são baseadas em leis de conservação. Algumas quantidades familiares que se submetem a tais leis são massa, energia e momento. Em termos matemáticos, esses princípios levam às equações de continuidade e de balanço que relacionam o comportamento do sistema, representado por seus níveis ou respostas das quantidades que estão sendo modeladas, às propriedades ou características do sistema e aos estímulos externos ou funções forçantes (perturbações) agindo sobre o sistema.

Como exemplo, o princípio de conservação de massa pode ser usado para formular um modelo para uma série de reatores químicos (Figura PT3.1a). Para esse caso, a quantidade que está sendo modelada é a massa do composto químico em cada reator. As propriedades do sistema são as características da reação do composto químico, o tamanho dos reatores e as vazões. As funções forçantes são as taxas de alimentação dos compostos químicos no sistema.

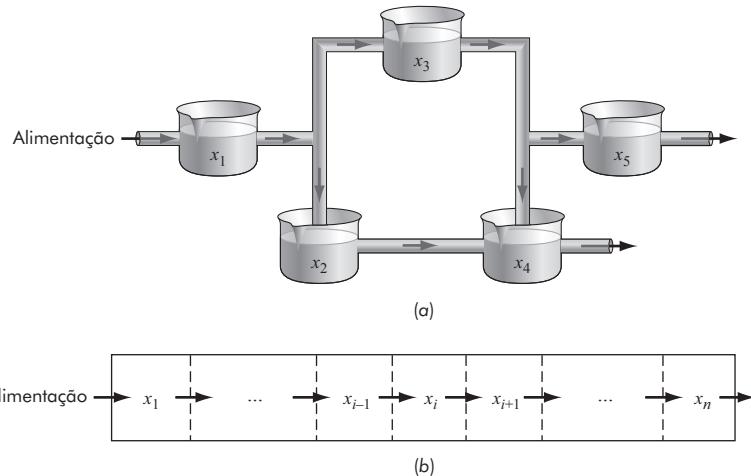


FIGURA PT3.1 Dois tipos de sistemas que podem ser modelados usando equações algébricas lineares: (a) sistemas de variáveis concentradas que envolvem componentes finitas acopladas e (b) sistemas de variáveis distribuídas que envolvem um contínuo.

Quando estudamos raízes de equações, vimos como os sistemas com uma única componente resultam em uma única equação, que pode ser resolvida usando-se as técnicas de localização de raízes. Sistemas com componentes múltiplas resultam em um conjunto de equações matemáticas acopladas, que devem ser resolvidas simultaneamente. As equações são acopladas porque as partes individuais do sistema são influenciadas pelas outras partes. Por exemplo, na Figura PT3.1a, o reator 4 recebe entradas dos reatores 2 e 3. Consequentemente, sua resposta depende das quantidades dos compostos químicos nesses outros reatores.

Quando essas dependências são expressas matematicamente, as equações resultantes são, muitas vezes, da forma algébrica linear da Equação (PT3.1). Os x 's são em geral medidas das intensidades das respostas de componentes individuais. Usando a Figura PT3.1a como exemplo, x_1 poderia indicar a quantidade de massa no primeiro reator, x_2 poderia indicar a quantidade no segundo reator e assim por diante. Os a 's representam as propriedades e as características que produzem as interações entre as componentes. Por exemplo, os a 's para a Figura PT3.1a poderiam refletir as vazões em massa entre os reatores. Por fim, os b 's representam as funções forçantes que agem sobre o sistema, como a taxa de alimentação na Figura PT3.1a.

Problemas com várias componentes do tipo citado surgem de modelos matemáticos de variáveis tanto concentradas (macro) quanto distribuídas (micro). *Problemas de variáveis concentradas* envolvem componentes finitas acopladas. Os três saltadores de bungee jumping interconectados, descritos no início do Capítulo 8, são um sistema concentrado. Outros exemplos incluem treliças, reatores e circuitos elétricos.

Reciprocamente, *problemas de variáveis distribuídas* tentam descrever detalhes espaciais de sistemas em uma base contínua ou semicontínua. A distribuição de compostos químicos ao longo do comprimento de um reator retangular alongado (Figura PT3.1b) é um exemplo de um modelo de variável contínua. As equações

diferenciais deduzidas das leis de conservação especificam a distribuição das variáveis dependentes para tais sistemas e podem ser resolvidas numericamente convertendo-as em um sistema equivalente de equações algébricas simultâneas.

A solução desses conjuntos de equações representa uma área de aplicação importante na engenharia para os métodos que serão apresentados nos capítulos seguintes. Essas equações são acopladas, pois as variáveis em uma posição dependem das variáveis nas regiões adjacentes. Por exemplo, a concentração no meio do reator na Figura PT3.1b é uma função da concentração nas regiões adjacentes. Exemplos similares podem ser desenvolvidos para distribuições espaciais de temperatura, momento ou eletricidade.

Além dos sistemas físicos, as equações algébricas lineares simultâneas também aparecem em uma variedade de contextos de problemas matemáticos. Eles ocorrem quando exigimos que funções matemáticas satisfaçam várias condições simultâneas. Cada condição resulta em uma equação que tem coeficientes conhecidos e variáveis desconhecidas. As técnicas discutidas nesta parte podem ser utilizadas para determinar as incógnitas quando as equações forem lineares e algébricas. Algumas técnicas numéricas amplamente utilizadas que empregam equações simultâneas são a análise de regressão e a interpolação por splines.

3.2 ORGANIZAÇÃO DA PARTE 3

Devido à sua importância na formulação e resolução de equações algébricas lineares, o Capítulo 8 fornece uma breve visão geral de *álgebra matricial* e, além de abordar os princípios básicos de representação e manipulação de matrizes, também descreve como matrizes são manipuladas no MATLAB.

O Capítulo 9 é dedicado à técnica mais fundamental para resolver sistemas algébricos lineares: *eliminação de Gauss*. Antes de uma discussão detalhada dessa técnica, uma seção preliminar trata dos métodos simples para resolver sistemas pequenos. Essas técnicas são apresentadas para fornecer uma percepção visual e o motivo pelo qual um dos métodos – a eliminação de incógnitas – representa a base da eliminação de Gauss.

Em seguida, a eliminação de Gauss “ingênua” é discutida. Começamos com essa versão simplificada porque ela permite que a técnica fundamental seja elaborada, sem detalhes complicados. Então, nas seções subsequentes, serão discutidos problemas potenciais dessa abordagem ingênua e apresentadas diversas modificações para minimizar e contornar esses problemas. O foco da discussão será o processo de permitar linhas, ou o *pivotamento parcial*. O capítulo termina com uma breve descrição de métodos eficientes para resolução de *matrizes tridiagonais*.

O Capítulo 10 ilustra como a eliminação de Gauss pode ser formulada como uma *decomposição LU*, o que é valioso para os casos em que muitos vetores do lado direito precisam ser calculados. O capítulo finaliza com uma breve explicação de como o MATLAB resolve sistemas lineares.

O Capítulo 11 começa com uma descrição de como a decomposição *LU* pode ser empregada para calcular de forma eficiente a *matriz inversa*, que tem uma grande utilidade na análise de relações do tipo estímulo-resposta de sistemas físicos, e, na sequência, dedica-se ao importante conceito de condicionamento de matrizes.

O número de condicionamento é introduzido como uma medida dos erros de arredondamento que podem ocorrer ao se resolver matrizes mal condicionadas.

O Capítulo 12 trata de técnicas de solução iterativas, que são similares, em essência, aos métodos de aproximação de raízes de equações que foram discutidos no Capítulo 6. Isto é, elas envolvem a escolha de uma solução aproximada e a iteração que obterá uma estimativa refinada. A ênfase é dada ao *método de Gauss-Seidel*, embora seja descrita uma abordagem alternativa, o *método de Jacobi*. O capítulo termina com uma breve descrição de como *equações simultâneas não lineares* podem ser resolvidas.

Por fim, o Capítulo 13 é dedicado aos problemas de *autovalores*, que têm relevância matemática geral, bem como muitas aplicações em engenharia e em ciência. São descritos dois métodos simples, assim como os recursos do MATLAB para determinação de autovalores e *autovetores*. Em termos de aplicações, focamos no uso dos autovalores e dos autovetores para estudar as vibrações e oscilações de sistemas mecânicos e estruturas.

8

Equações algébricas lineares e matrizes

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é familiarizar o leitor com as equações algébricas lineares e suas relações com as matrizes e com a álgebra matricial. Após esta leitura completa, você será capaz de:

- ✓ Entender a notação matricial.
- ✓ Identificar os seguintes tipos de matrizes: identidade, diagonal, simétrica, triangular e tridiagonal.
- ✓ Aprender como efetuar uma multiplicação matricial e avaliar quando ela é possível.
- ✓ Aprender como representar um sistema de equações algébricas lineares na forma matricial.
- ✓ Aprender como resolver equações algébricas lineares no MATLAB aplicando a divisão à esquerda ou a inversão de matriz.

VOCÊ TEM UM PROBLEMA

Suponha que três saltadores de *bungee jumping* estão conectados por cordas. A Figura 8.1a mostra-os sendo mantidos no lugar verticalmente, de modo que cada corda está totalmente estendida, porém não esticada. Podemos definir três distâncias: x_1 , x_2 e x_3 , como medidas para baixo a partir de cada uma de suas posições não esticadas. Depois que eles são lançados, a gravidade atua e os saltadores chegam, por fim, às posições de equilíbrio mostradas na Figura 8.1b.

Considere que seja solicitado a você calcular o deslocamento de cada um dos saltadores. Se admitirmos que cada corda se comporta como uma mola linear e segue a lei de Hooke, diagramas de corpo livre podem ser desenvolvidos para cada saltador como detalhado na Figura 8.2.

Utilizando a segunda lei de Newton, o balanço de forças pode ser escrito para cada saltador como:

$$\begin{aligned}m_1 \frac{d^2x_1}{dt^2} &= m_1g + k_2(x_2 - x_1) - k_1x_1 \\m_2 \frac{d^2x_2}{dt^2} &= m_2g + k_3(x_3 - x_2) + k_2(x_1 - x_2) \\m_3 \frac{d^2x_3}{dt^2} &= m_3g + k_3(x_2 - x_3)\end{aligned}\tag{8.1}$$

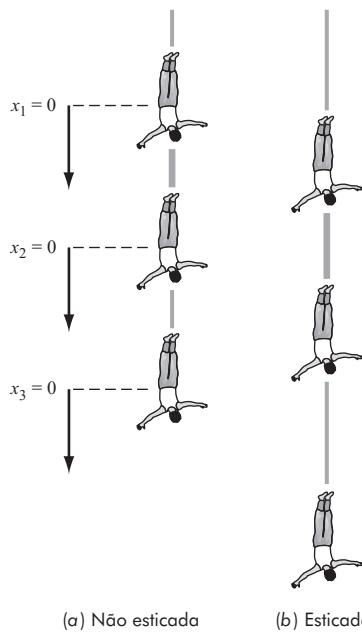


FIGURA 8.1 Três indivíduos conectados por cordas de *bungee jumping*.

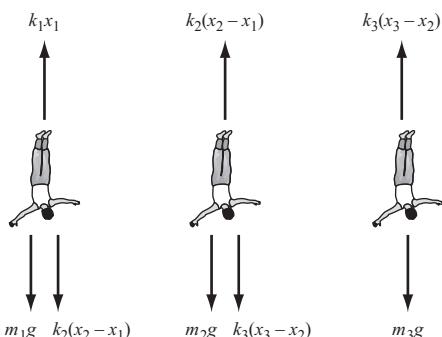


FIGURA 8.2 Diagramas de corpo livre.

onde m_i é a massa do saltador i (kg), t é o tempo (s), k_j é a constante da mola para a corda j (N/m), x_i é o deslocamento do saltador i medido para baixo a partir da posição de equilíbrio (m) e g é a aceleração da gravidade ($9,81 \text{ m/s}^2$). Como estamos interessados na solução de estado estacionário (ou regime permanente), a segunda derivada pode ser definida como zero. Agrupando os termos comuns, temos

$$\begin{aligned} (k_1 + k_2)x_1 - k_2x_2 &= m_1g \\ -k_2x_1 + (k_2 + k_3)x_2 - k_3x_3 &= m_2g \\ -k_3x_2 + k_3x_3 &= m_3g \end{aligned} \quad (8.2)$$

Portanto, o problema se reduz à solução de um sistema de três equações simultâneas para os três deslocamentos desconhecidos. Como utilizamos uma lei (física) linear para as cordas, as equações resultantes são equações algébricas lineares. Nos Capítulos 8 a 12 será apresentada uma introdução de como o MATLAB é usado para resolver tais sistemas de equações.

8.1 ÁLGEBRA MATRICIAL: UMA VISÃO GERAL

O conhecimento de matrizes é essencial para compreender a solução de sistemas de equações algébricas lineares. As seções seguintes descrevem como as matrizes fornecem uma maneira concisa para representar e manipular equações algébricas lineares.

8.1.1 Notação matricial

Uma *matriz* consiste em um arranjo retangular de elementos representados por um único símbolo. Como mostra a Figura 8.3, $[A]$ é a notação abreviada para a matriz, e a_{ij} designa um *elemento* individual da matriz.

Um conjunto horizontal de elementos é chamado de *linha*, e um conjunto vertical é chamado de *coluna*. O primeiro índice, i , sempre representa o número da linha na qual o elemento está. O segundo índice, j , designa a coluna. Por exemplo, o elemento a_{23} está na linha 2 e coluna 3.

A matriz na Figura 8.3 tem m linhas e n colunas, com dimensão m por n (ou $m \times n$), e é chamada de uma matriz m por n .

As matrizes com número de linhas $m = 1$, tais como

$$[b] = [b_1 \ b_2 \ \cdots \ b_n]$$

são chamadas *vetores linha*. Observe que, para simplificar, o primeiro índice de cada elemento foi omitido. É importante ressaltar que, algumas vezes, é desejável empregar uma notação abreviada especial para diferenciar uma matriz linha de outros tipos de matrizes. Uma forma de fazer isso é utilizar colchetes abertos em cima, como em $\lceil b \rfloor$.¹

As matrizes com número de colunas $n = 1$, como

$$[c] = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \quad (8.3)$$

são chamadas de *vetores coluna*. Para simplificar, o segundo índice é omitido. Como nos vetores linha, há ocasiões nas quais é necessário empregar uma notação abreviada especial para diferenciar uma matriz coluna de outros tipos de matrizes. Uma forma de fazer isso é usar chaves, como em $\{c\}$.

Coluna 3
↓
 $[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$ ← Linha 2

FIGURA 8.3 Uma matriz.

¹ Além de colchetes especiais, utilizamos letras minúsculas e maiúsculas para diferenciar vetores (minúsculas) e matrizes (maiúsculas).

As matrizes em que $m = n$ são chamadas *matrizes quadradas*. Por exemplo, uma matriz 3×3 é

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

A diagonal que consiste nos elementos a_{11} , a_{22} e a_{33} é chamada *diagonal principal* da matriz.

As matrizes quadradas são particularmente importantes quando se resolvem conjuntos de equações lineares simultâneas. Para tais sistemas, o número de equações (correspondentes às linhas) e o número de incógnitas (correspondentes às colunas) devem ser iguais, para que seja possível uma solução única. Consequentemente, são encontradas matrizes quadradas de coeficientes quando lidamos com tais sistemas, para as quais há algumas formas especiais que são importantes e devem ser observadas.

Uma *matriz simétrica* é aquela cujas linhas são iguais às colunas – isto é, $a_{ij} = a_{ji}$ para todos os i 's e j 's. Por exemplo,

$$[A] = \begin{bmatrix} 5 & 1 & 2 \\ 1 & 3 & 7 \\ 2 & 7 & 8 \end{bmatrix}$$

é uma matriz 3×3 simétrica.

Uma *matriz diagonal* é uma matriz quadrada cujos elementos fora da diagonal principal são iguais a zero, como em

$$[A] = \begin{bmatrix} a_{11} & & \\ & a_{22} & \\ & & a_{33} \end{bmatrix}$$

Observe que, quando blocos grandes de elementos são nulos, eles são deixados em branco.

Uma *matriz identidade* é uma matriz diagonal na qual todos os elementos da diagonal principal são iguais a 1, como em

$$[I] = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$$

A matriz identidade tem propriedades similares à unidade. Isto é,

$$[A][I] = [I][A] = [A]$$

Uma *matriz triangular superior* é uma matriz em que todos os elementos abaixo da diagonal principal são zero, como em

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & a_{22} & a_{23} \\ & & a_{33} \end{bmatrix}$$

Uma *matriz triangular inferior* é uma matriz em que todos os elementos acima da diagonal principal são zero, como em

$$[A] = \begin{bmatrix} a_{11} & & \\ a_{21} & a_{22} & \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Uma *matriz de banda* tem todos os elementos iguais a zero, com exceção de uma faixa centrada na diagonal principal:

$$[A] = \begin{bmatrix} a_{11} & a_{12} & & \\ a_{21} & a_{22} & a_{23} & \\ & a_{32} & a_{33} & a_{34} \\ & & a_{43} & a_{44} \end{bmatrix}$$

A matriz anterior tem uma largura da banda de 3 e tem um nome especial – *matriz tridiagonal*.

8.1.2 Regras de operações matriciais

Agora que ficou claro o que se quer definir por matriz, pode-se apresentar certas operações que governam seu uso. Duas matrizes m por n são iguais se, e somente se, todos os elementos da primeira forem iguais a todos elementos na segunda, isto é, $[A] = [B]$ se $a_{ij} = b_{ij}$ para todo i e j .

A *adição* de duas matrizes, $[A]$ e $[B]$, é feita somando-se os elementos correspondentes em cada matriz. Os elementos da matriz resultante $[C]$ são calculados por

$$c_{ij} = a_{ij} + b_{ij}$$

para $i = 1, 2, \dots, m$ e $j = 1, 2, \dots, n$. Analogamente, a subtração de duas matrizes, $[E]$ menos $[F]$, é obtida subtraindo-se os termos correspondentes, como em

$$d_{ij} = e_{ij} - f_{ij}$$

para $i = 1, 2, \dots, m$ e $j = 1, 2, \dots, n$. Segue diretamente dessas definições que a adição e a subtração podem ser efetuadas apenas entre matrizes com as mesmas dimensões.

A adição e a subtração são comutativas:

$$[A] + [B] = [B] + [A]$$

e associativas:

$$([A] + [B]) + [C] = [A] + ([B] + [C])$$

A multiplicação de uma matriz $[A]$ por um escalar g é obtida multiplicando-se todo elemento de $[A]$ por g . Por exemplo, para uma matriz 3×3 :

$$[D] = g[A] = \begin{bmatrix} ga_{11} & ga_{12} & ga_{13} \\ ga_{21} & ga_{22} & ga_{23} \\ ga_{31} & ga_{32} & ga_{33} \end{bmatrix}$$

O produto de duas matrizes é representado por $[C] = [A][B]$, em que os elementos de $[C]$ são definidos como

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \quad (8.4)$$

onde n é o número de colunas de $[A]$ e o número de linhas de $[B]$. Isto é, o elemento c_{ij} é obtido somando-se o produto dos elementos individuais da i -ésima linha da primeira matriz, nesse caso $[A]$, com a j -ésima coluna da segunda matriz, $[B]$. A Figura 8.4 mostra como as linhas e colunas se alinham em uma multiplicação de matrizes.

De acordo com essa definição, a multiplicação de matrizes pode ser efetuada apenas quando a primeira matriz tem tantas colunas quanto o número de linhas da segunda matriz. Então, se $[A]$ é uma matriz m por n , $[B]$ poderia ser uma matriz n por l . Nesse caso, a matriz resultante $[C]$ terá dimensão m por l . Entretanto, se $[B]$ fosse uma matriz m por l , a multiplicação não poderia ser efetuada. A Figura 8.5 apresenta uma forma fácil de verificar quando duas matrizes podem ser multiplicadas.

Se as dimensões das matrizes forem adequadas, a multiplicação de matrizes será *associativa*:

$$([A][B])[C] = [A]([B][C])$$

e *distributiva*:

$$[A]([B] + [C]) = [A][B] + [A][C]$$

ou

$$([A] + [B])[C] = [A][C] + [B][C]$$

Porém, em geral, a multiplicação não é *comutativa*:

$$[A][B] \neq [B][A]$$

Isto é, a ordem na multiplicação da matriz é importante.

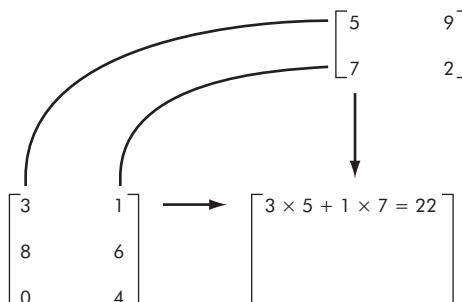


FIGURA 8.4 Descrição visual de como as linhas e colunas se alinham em uma multiplicação de matrizes.

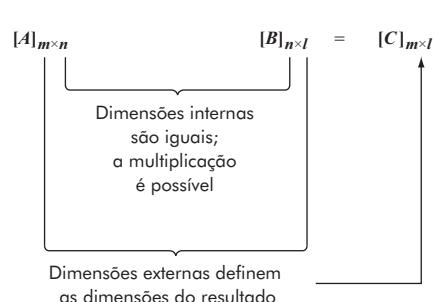


FIGURA 8.5 A multiplicação matricial pode ser efetuada somente se as dimensões internas forem iguais.

Apesar de a multiplicação ser possível, a divisão de matrizes não é uma operação definida. Porém, se a matriz $[A]$ for quadrada e não singular, existe uma outra matriz $[A]^{-1}$, chamada a *inversa* de $[A]$, tal que

$$[A][A]^{-1} = [A]^{-1}[A] = [I]$$

Assim, a multiplicação de uma matriz pela sua inversa é o análogo da divisão, no sentido de que um número dividido por ele mesmo é igual a 1, isto é, a multiplicação de uma matriz pela sua inversa resulta na matriz identidade.

A inversa de uma matriz 2×2 pode ser representada simplesmente por

$$[A]^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

Fórmulas semelhantes para matrizes de dimensão maior são muito mais complexas. No Capítulo 11 serão apresentadas técnicas para calcular a inversa de tais matrizes por meio de métodos numéricos e do computador.

A *transposta* de uma matriz implica transformar suas linhas em colunas e suas colunas em linhas. Por exemplo, para a matriz 3×3 :

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

a transposta, designada por $[A]^T$, é definida por

$$[A]^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}$$

Em outras palavras, o elemento a_{ij} da transposta é igual ao elemento a_{ji} da matriz original.

A transposta tem uma variedade de funções na álgebra das matrizes e permite que um vetor coluna seja escrito como um vetor linha, o que representa uma vantagem simples. Por exemplo, se

$$\{c\} = \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \end{Bmatrix}$$

então

$$\{c\}^T = [c_1 \quad c_2 \quad c_3]$$

Além disso, tem inúmeras aplicações matemáticas.

Uma *matriz de permutação* (também chamada *matriz de transposição*) é uma matriz identidade com linhas e colunas permutadas. Por exemplo, a matriz a seguir é uma matriz de permutação que é construída trocando-se a primeira e a terceira linhas e colunas de uma matriz identidade 3×3 :

$$[P] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

A multiplicação à esquerda de uma matriz $[A]$ por essa matriz, como em $[P][A]$, trocará as linhas correspondentes de $[A]$. A multiplicação à direita, como em $[A][P]$, trocará as colunas correspondentes. Um exemplo de multiplicação à esquerda é:

$$[P][A] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & -7 & 4 \\ 8 & 3 & -6 \\ 5 & 1 & 9 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 9 \\ 8 & 3 & -6 \\ 2 & -7 & 4 \end{bmatrix}$$

A última manipulação de matrizes que terá utilidade na nossa discussão é o *aumento*. A matriz é aumentada pela adição de uma coluna (ou colunas) à matriz original. Por exemplo, suponha que tenhamos uma matriz de coeficientes 3×3 . Você poderia aumentar essa matriz $[A]$ com uma matriz identidade 3×3 para produzir uma matriz de dimensão 3×6 .

$$\left[\begin{array}{ccc|ccc} a_{11} & a_{11} & a_{11} & 1 & 0 & 0 \\ a_{21} & a_{21} & a_{21} & 0 & 1 & 0 \\ a_{31} & a_{31} & a_{31} & 0 & 0 & 1 \end{array} \right]$$

Tal expressão tem utilidade quando é preciso efetuar um conjunto de operações idênticas em duas matrizes. Assim, pode-se efetuar as operações na matriz aumentada em vez de fazê-lo nas duas matrizes individuais.

EXEMPLO 8.1 Manipulando matrizes no MATLAB

O exemplo a seguir ilustra como uma variedade de manipulações matriciais são implementadas com o MATLAB. Este exemplo é mais bem aproveitado se executado como um exercício no computador.

Solução. Crie uma matriz 3×3 :

```
>> A = [1 5 6; 7 4 2; -3 6 7]
A =
    1     5     6
    7     4     2
   -3     6     7
```

A transposta de $[A]$ pode ser obtida utilizando-se o operador `'`:

```
>> A'
ans =
    1     7    -3
    5     4     6
    6     2     7
```

Agora, criaremos uma outra matriz 3×3 com base em linhas individuais. Primeiro, são criados três vetores linha:

```
>> x = [8 6 9];
>> y = [-5 8 1];
>> z = [4 8 2];
```

Depois, podemos combiná-los para formar a matriz:

```
>> B = [x; y; z]
```

B =

8	6	9
-5	8	1
4	8	2

Podemos somar [A] e [B]:

```
>> C = A+B
```

C =

9	11	15
2	12	3
1	14	9

Então, podemos subtrair [B] de [C] para chegar novamente à [A]:

```
>> A = C-B
```

A =

1	5	6
7	4	2
-3	6	7

Como suas dimensões internas são iguais, [A] e [B] podem ser multiplicadas

```
>> A*B
```

ans =

7	94	26
44	90	71
-26	86	-7

Observe que [A] e [B] também podem ser multiplicadas, elemento por elemento, incluindo um ponto com o operador de multiplicação como em

```
>> A.*B
```

ans =

8	30	54
-35	32	2
-12	48	14

Uma matriz 2×3 pode ser definida

```
>> D = [1 4 3; 5 8 1];
```

Se tentarmos multiplicar [A] por [D], uma mensagem de erro é exibida

```
>> A*D
```

```
??? Error using ==> mtimes
```

```
Inner matrix dimensions must agree.
```

No entanto, se invertermos a ordem de multiplicação de modo que as dimensões internas sejam iguais, a multiplicação matricial será executada

```
>> D*A  
ans =  
    20     39     35  
    58     63     53
```

A inversa da matriz pode ser calculada com a função `inv`:

```
>> AI = inv(A)  
AI =  
    0.2462     0.0154    -0.2154  
   -0.8462     0.3846     0.6154  
    0.8308    -0.3231    -0.4769
```

Para verificar se esse resultado está correto, a inversa pode ser multiplicada pela matriz original para fornecer a matriz identidade:

```
>> A*AI  
ans =  
    1.0000    -0.0000    -0.0000  
    0.0000     1.0000    -0.0000  
    0.0000    -0.0000     1.0000
```

A função `eye` pode ser usada para gerar uma matriz identidade:

```
>> I = eye(3)  
I =  
    1     0     0  
    0     1     0  
    0     0     1
```

Podemos definir uma matriz de permutação para trocar a primeira e a terceira linhas e colunas de uma matriz 3×3 como

```
>> P = [0 0 1; 0 1 0; 1 0 0]  
P =  
    0     0     1  
    0     1     0  
    1     0     0
```

Podemos, então, trocar as linhas:

```
>> PA = P*A  
PA =  
    -3      6      7  
     7      4      2  
     1      5      6
```

ou trocar as colunas:

```
>> AP = A*P
AP =
 6   5   1
 2   4   7
 7   6  -3
```

Finalmente, matrizes podem ser aumentadas simplesmente como em

```
>> Aum = [A  I]
Aum =
 1   5   6   1   0   0
 7   4   2   0   1   0
 -3   6   7   0   0   1
```

Observe que as dimensões de uma matriz podem ser determinadas utilizando-se a função `size`:

```
>> [n,m] = size(Aum)
n =
 3
m =
 6
```

8.1.3 Representando equações algébricas lineares na forma matricial

É importante ressaltar que as matrizes fornecem uma notação concisa para representar equações lineares simultâneas. Por exemplo, um conjunto de equações lineares 3×3 ,

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \tag{8.5}$$

pode ser expresso como

$$[A]\{x\} = \{b\} \tag{8.6}$$

onde $[A]$ é a matriz dos coeficientes

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$\{b\}$ é o vetor coluna das constantes

$$\{b\}^T = [b_1 \quad b_2 \quad b_3]$$

e $\{x\}$ é o vetor coluna das incógnitas:

$$\{x\}^T = [x_1 \quad x_2 \quad x_3]$$

Recorde a definição de multiplicação de matrizes [Equação (8.4)] para se convencer de que as Equações (8.5) e (8.6) são equivalentes. Além disso, observe que a Equação (8.6) é uma multiplicação de matrizes válida, pois o número de colunas, n , da primeira matriz $[A]$ é igual ao número de linhas, n , da segunda matriz $\{x\}$.

Esta parte do livro é dedicada a resolver a Equação (8.6) para determinar $\{x\}$. Uma maneira formal de obter a solução usando álgebra matricial é multiplicar cada lado da equação pela inversa de $[A]$ para obter

$$[A]^{-1}[A]\{x\} = [A]^{-1}\{b\}$$

Como $[A]^{-1}[A]$ é igual à matriz identidade, a equação se torna

$$\{x\} = [A]^{-1}\{b\} \quad (8.7)$$

Então, a equação foi resolvida e $\{x\}$ foi determinado. Esse é um outro exemplo de como a inversa desempenha um papel na álgebra de matrizes semelhante à divisão. Deve-se observar que esse não é um método muito eficiente para resolver um sistema de equações. Portanto, outras abordagens serão empregadas nos algoritmos numéricos. Entretanto, como discutido na Seção 11.1.2, a própria matriz inversa tem grande importância na análise de tais sistemas.

É importante observar que sistemas com mais equações (linhas) que incógnitas (colunas), $m > n$, são chamados *sobre determinados*. Um exemplo típico é a regressão por mínimos quadrados em que uma equação com n coeficientes é ajustada a m pontos (x, y) . Já os sistemas com menos equações que incógnitas, $m < n$, são chamados *subdeterminados*, e um exemplo típico destes é a otimização numérica.

8.2 RESOLVENDO EQUAÇÕES ALGÉBRICAS LINEARES COM O MATLAB

O MATLAB fornece dois meios diretos para resolver sistemas de equações algébricas lineares, sendo que o mais eficiente é o emprego do operador barra invertida, ou “divisão à esquerda”, como em

```
>> x = A\b
```

O segundo é usar a inversão da matriz:

```
>> x = inv(A)*b
```

Como indicado no fim da Seção 8.1.3, a solução pela matriz inversa é menos eficiente que utilizar a divisão à esquerda. Ambas as opções são ilustradas no exemplo a seguir.

EXEMPLO 8.2 Resolvendo o problema do saltador de *bungee jumping* com o MATLAB

Use o MATLAB para resolver o problema do *bungee jumping* descrito no início desse capítulo. Os parâmetros para o problema são

Saltador	Massa (kg)	Constante da mola (N/m)	Comprimento da corda não esticada (m)
Superior (1)	60	50	20
Do meio (2)	70	100	20
Inferior (3)	80	50	20

Solução. Substituindo esses valores de parâmetros na Equação (8.2), obtemos

$$\begin{bmatrix} 150 & -100 & 0 \\ -100 & 150 & -50 \\ 0 & -50 & 50 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 588,6 \\ 686,7 \\ 784,8 \end{Bmatrix}$$

Inicie o MATLAB e entre com a matriz de coeficientes e com o vetor à direita da equação anterior:

```
>> K = [ 150 -100 0; -100 150 -50; 0 -50 50]
K =
    150   -100      0
   -100    150   -50
       0    -50     50
>> mg = [ 588.6; 686.7; 784.8]
mg =
    588.6000
    686.7000
    784.8000
```

A utilização da divisão à esquerda fornece

```
>> x = K\mg
x =
    41.2020
    55.9170
    71.6130
```

Alternativamente, a multiplicação da inversa da matriz de coeficientes pelo vetor do lado direito fornece o mesmo resultado:

```
>> x = inv(K) * mg
x =
    41.2020
    55.9170
    71.6130
```

Como os saltadores foram conectados por cordas de 20 m, suas posições iniciais em relação à plataforma são

```
>> xi = [20; 40; 60];
```

Assim, suas posições finais podem ser calculadas como

```
>> xf = x+xi
```

```
xf =
61.2020
95.9170
131.6130
```

Os resultados apresentados na Figura 8.6 fazem sentido. A primeira corda é a mais esticada, pois tem a menor constante da mola e está sujeita ao maior peso (todos os três saltadores). Observe que a segunda e a terceira cordas estão estendidas aproximadamente na mesma proporção. Como a segunda corda está sujeita ao peso de dois saltadores, poder-se-ia esperar que se estendesse mais que a terceira. No entanto, como a segunda corda é mais rígida (isto é, tem uma constante da mola mais elevada), ela se estende menos que o esperado com base no peso que ela suporta.

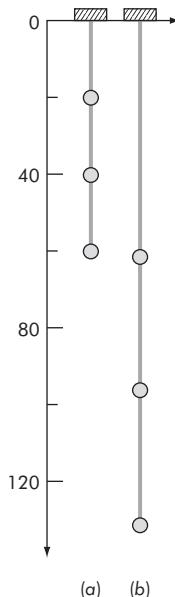


FIGURA 8.6 Posições de três indivíduos conectados por cordas de *bungee jumping*. (a) Não esticadas e (b) esticadas.

►8.3 ESTUDO DE CASO CORRENTES E TENSÕES EM CIRCUITOS

Contextualização. Vimos no Capítulo 1 (Tabela 1.1) o resumo de alguns modelos e leis de conservação associadas que têm papel importante na engenharia. Como na Figura 8.7, cada modelo representa um sistema de elementos em interação. Consequentemente, balanços de regime permanente derivados das leis de conservação produzem sistemas de equações simultâneas. Em muitos casos, tais sistemas são lineares e, assim, podem ser expressos na forma matricial. O estudo de caso presente concentra-se em uma aplicação: análise de circuitos.

Um problema comum em engenharia elétrica envolve a determinação das correntes e tensões em vários pontos de circuitos resistivos. Esses problemas são resolvidos aplicando-se as *leis de Kirchhoff das correntes e das tensões*. A *lei das correntes* estabelece que a soma algébrica de todas as correntes incidindo em um nó deve ser zero (Figura 8.8a), ou

$$\sum i = 0 \quad (8.8)$$

onde todas as correntes que entram no nó são consideradas de sinal positivo (aqueles que saem do nó têm, portanto, sinal negativo). A lei das correntes é uma aplicação do princípio de *conservação da carga* (ver Tabela 1.1).

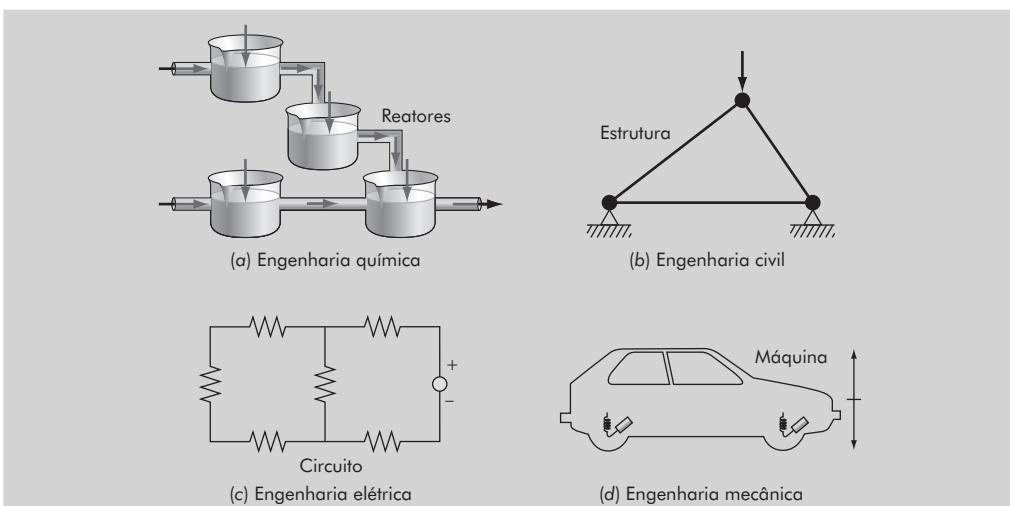


FIGURA 8.7 Sistemas de engenharia que, em regime permanente ou estado estacionário, podem ser modelados com equações algébricas lineares.

A *lei das tensões* especifica que a soma algébrica das diferenças de potencial (ou seja, quedas de tensão) em qualquer caminho fechado (ou laço) deve ser igual a zero. Para um circuito resistivo, isso é expresso como

$$\sum \xi - \sum iR = 0 \quad (8.9)$$

onde ξ é a fem (força eletromotriz) das fontes de tensão e R é a resistência de qualquer resistor no caminho fechado. Observe que o segundo termo é derivado da *lei de Ohm* (Figura 8.8b), que estabelece que a queda de tensão através de um resistor ideal é igual ao produto de sua resistência pela corrente que circula por ele. A lei de Kirchhoff das tensões é uma expressão da *conservação da energia*.

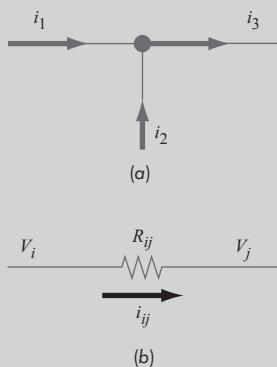


FIGURA 8.8 Representação esquemática da (a) lei de Kirchhoff das correntes e (b) lei de Ohm.

Solução. A aplicação dessas leis resulta em sistemas de equações algébricas lineares simultâneas, porque os vários laços dentro de um circuito estão interconectados. Por exemplo, considere o circuito mostrado na Figura 8.9. As correntes associadas com esse circuito são desconhecidas tanto em valor quanto em sentido. Isso não representa grande dificuldade, pois pode-se simplesmente assumir um sentido (arbitrário) para cada corrente. Se a solução resultante da lei de Kirchhoff é negativa, então o sentido real de circulação da corrente é contrário ao assumido arbitrariamente. Por exemplo, a Figura 8.10 mostra algumas correntes assumidas.

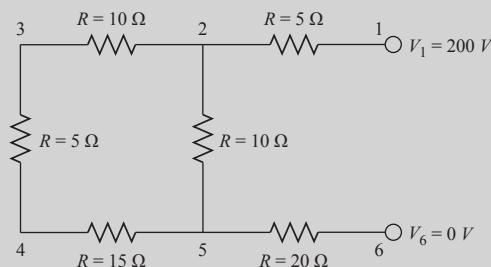


FIGURA 8.9 Circuito resistivo para ser resolvido por meio de equações algébricas lineares simultâneas.

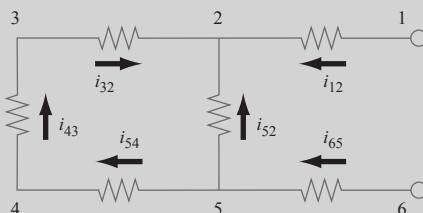


FIGURA 8.10 Sentidos assumidos para as correntes.

Dadas essas considerações, a lei de Kirchhoff das correntes é aplicada a cada nó para produzir

$$i_{12} + i_{52} + i_{32} = 0$$

$$i_{65} - i_{52} - i_{54} = 0$$

$$i_{43} - i_{32} = 0$$

$$i_{54} - i_{43} = 0$$

A aplicação da lei das tensões a cada um dos dois laços fornece

$$-i_{54}R_{54} - i_{43}R_{43} - i_{32}R_{32} + i_{52}R_{52} = 0$$

$$-i_{65}R_{65} - i_{52}R_{52} + i_{12}R_{12} - 200 = 0$$

ou, substituindo as resistências da Figura 8.9 e trazendo as constantes para o lado direito,

$$-15i_{54} - 5i_{43} - 10i_{32} + 10i_{52} = 0$$

$$-20i_{65} - 10i_{52} + 5i_{12} = 200$$

Portanto, o problema equivale à resolução de seis equações com seis correntes desconhecidas, e que podem ser expressas na forma matricial como

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 10 & -10 & 0 & -15 & -5 \\ 5 & -10 & 0 & -20 & 0 & 0 \end{bmatrix} \begin{Bmatrix} i_{12} \\ i_{52} \\ i_{32} \\ i_{65} \\ i_{54} \\ i_{43} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 200 \end{Bmatrix}$$

Embora a resolução desse sistema seja pouco prática manualmente, ele é facilmente solucionado com a utilização do MATLAB. A solução é

```
>> A=[1 1 1 0 0 0
0 -1 0 1 -1 0
0 0 -1 0 0 1
0 0 0 0 1 -1
0 10 -10 0 -15 -5
5 -10 0 -20 0 0];
>> b=[0 0 0 0 200]';
>> corrente=A\b

corrente =
6.1538
-4.6154
-1.5385
-6.1538
-1.5385
-1.5385
```

Assim, com a interpretação adequada dos sinais do resultado, as correntes e tensões do circuito são apresentadas na Figura 8.11. As vantagens de utilizar o MATLAB para problemas desse tipo devem estar evidentes.

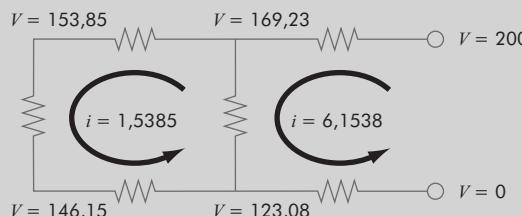


FIGURA 8.11 A solução para as correntes e tensões obtida usando o MATLAB.

►►PROBLEMAS

8.1 Dada uma matriz quadrada $[A]$, escreva uma única linha de comando do MATLAB que cria uma nova matriz $[Aum]$ que consista na matriz original $[A]$ aumentada por uma matriz identidade $[I]$.

8.2 Uma série de matrizes são definidas como

$$\begin{aligned} [A] &= \begin{bmatrix} 4 & 5 \\ 1 & 2 \\ 5 & 6 \end{bmatrix} & [B] &= \begin{bmatrix} 4 & 3 & 7 \\ 1 & 2 & 6 \\ 2 & 0 & 4 \end{bmatrix} \\ \{C\} &= \begin{bmatrix} 2 \\ 6 \\ 1 \end{bmatrix} & [D] &= \begin{bmatrix} 5 & 4 & 3 & -7 \\ 2 & 1 & 7 & 5 \end{bmatrix} \\ [E] &= \begin{bmatrix} 1 & 5 & 6 \\ 7 & 1 & 3 \\ 4 & 0 & 6 \end{bmatrix} \\ [F] &= \begin{bmatrix} 2 & 0 & 1 \\ 1 & 7 & 4 \end{bmatrix} & [G] &= [8 \quad 6 \quad 4] \end{aligned}$$

Responda às seguintes questões em relação a essas matrizes:

- (a) Quais são as dimensões das matrizes?
- (b) Identifique as matrizes quadrada, coluna e linha.
- (c) Quais são os valores dos elementos: $a_{12}, b_{23}, d_{32}, e_{22}, f_{12}, g_{12}$?
- (d) Efetue as seguintes operações:

- (1) $[E] + [B]$
- (2) $[A] + [F]$
- (3) $[B] - [E]$
- (4) $7 \times [B]$
- (5) $\{C\}^T$
- (6) $[E] \times [B]$
- (7) $[B] \times [E]$
- (8) $[D]^T$
- (9) $[G] \times \{C\}$
- (10) $[I] \times [B]$
- (11) $[E]^T \times [E]$
- (12) $\{C\}^T \times \{C\}$

8.3 Escreva o seguinte conjunto de equações na forma matricial:

$$\begin{aligned} 50 &= 5x_3 - 6x_2 \\ 2x_2 + 7x_3 + 30 &= 0 \\ x_1 - 7x_3 &= 50 - 3x_2 + 5x_1 \end{aligned}$$

Use o MATLAB para calcular o valor das incógnitas e também a transposta e a inversa da matriz dos coeficientes.

8.4 Três matrizes são definidas como

$$[A] = \begin{bmatrix} 6 & -1 \\ 12 & 7 \\ -5 & 3 \end{bmatrix} \quad [B] = \begin{bmatrix} 4 & 0 \\ 0,6 & 8 \end{bmatrix} \quad [C] = \begin{bmatrix} 1 & -2 \\ -6 & 1 \end{bmatrix}$$

- (a) Efetue todas as multiplicações que podem ser calculadas entre pares dessas matrizes.

- (b) Justifique por que os pares restantes não podem ser multiplicados.

- (c) Use os resultados de (a) para ilustrar por que a ordem de multiplicação é importante.

8.5 Resolva o seguinte sistema com o MATLAB

$$\begin{bmatrix} 3+2i & 4 \\ -i & 1 \end{bmatrix} \begin{Bmatrix} z_1 \\ z_2 \end{Bmatrix} = \begin{Bmatrix} 2+i \\ 3 \end{Bmatrix}$$

8.6 Desenvolva, depure e teste seu próprio programa em MATLAB para multiplicar duas matrizes – isto é, $[X] = [Y][Z]$, onde $[Y]$ é m por n e $[Z]$ é n por p . Empregue laços `for...end` para implementar a multiplicação e inclua em seu código armadilhas de erro para sinalizar casos incorretos (por exemplo, matrizes com dimensões internas diferentes). Teste seu programa com as matrizes do Problema 8.4.

8.7 Desenvolva, depure e teste seu próprio programa em MATLAB para fornecer a transposta de uma matriz. Empregue laços `for...end` para implementar a transposta. Teste seu programa com as matrizes do Problema 8.4.

8.8 Desenvolva, depure e teste seu próprio programa em MATLAB para mudar as linhas de uma matriz usando uma matriz de permutação. As primeiras linhas da função devem ser como a seguir:

```
function B = permute(A,11,12)
% Permut: Troca linhas de uma matriz A
% com a matriz de permutação
% B = permute(A,11,12)
% entrada:
% A = matriz original
% 11, 12 = linhas a serem permutadas
% saída:
% B = matriz com as linhas trocadas
```

Inclua em seu código armadilhas para entradas incorretas (por exemplo, o usuário especifica linhas que excedem as dimensões da matriz original).

8.9 Cinco reatores acoplados por tubos estão ilustrados na Figura P8.9. A taxa do fluxo de massa através de cada tubo é calculada como o produto do fluxo (Q) pela concentração (c). Em regime permanente (estado estacionário), o fluxo de massa para dentro e para fora de cada reator deve ser igual. Por exemplo, para o primeiro reator, um *balanço de massa* pode ser escrito como

$$Q_{01}c_{01} + Q_{31}c_3 = Q_{15}c_1 + Q_{12}c_1$$

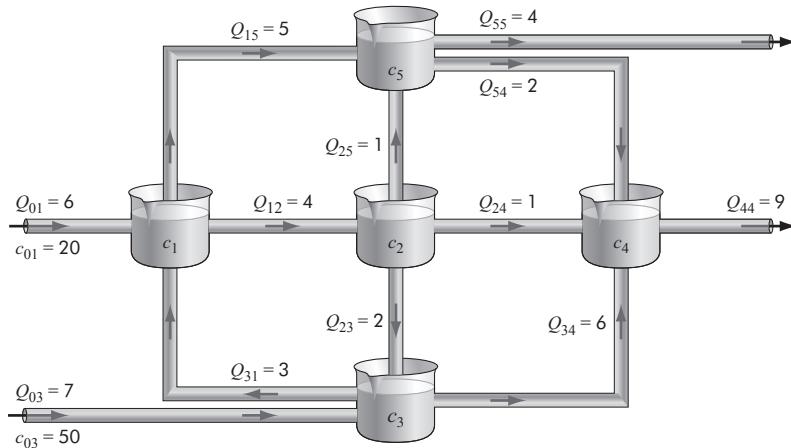


FIGURA P8.9

Escreva balanços de massa para os reatores restantes na Figura P8.9 e expresse as equações na forma matricial. Em seguida, use o MATLAB para determinar as concentrações em cada reator.

8.10 Um problema importante em engenharia estrutural é determinar as forças em uma treliça estaticamente determinada (Figura P8.10). Esse tipo de estrutura pode ser descrito por um sistema de equações lineares acopladas derivado dos balanços de forças. A soma das forças na direção horizontal e vertical deve ser zero em cada nó, pois o sistema está em repouso. Assim, para o nó 1:

$$\begin{aligned}\sum F_H &= 0 = -F_1 \cos 30^\circ + F_3 \cos 60^\circ + F_{1,h} \\ \sum F_V &= 0 = -F_1 \sin 30^\circ - F_3 \sin 60^\circ + F_{1,v}\end{aligned}$$

para o nó 2:

$$\begin{aligned}\sum F_H &= 0 = F_2 + F_1 \cos 30^\circ + F_{2,h} + H_2 \\ \sum F_V &= 0 = F_1 \sin 30^\circ + F_{2,v} + V_2\end{aligned}$$

para o nó 3:

$$\begin{aligned}\sum F_H &= 0 = -F_2 - F_3 \cos 60^\circ + F_{3,h} \\ \sum F_V &= 0 = F_3 \sin 60^\circ + F_{3,v} + V_3\end{aligned}$$

onde $F_{i,h}$ é a força externa horizontal aplicada ao nó i (onde uma força positiva age da esquerda para a direita) e $F_{i,v}$ é a força externa vertical aplicada ao nó i (onde uma força positiva age para cima).

Assim, nesse problema, a força de 2.000 N para baixo no nó 1 corresponde a $F_{1,v} = -2.000$ N. Para esse caso, todas as outras $F_{i,v}$'s e $F_{i,h}$'s são nulas.

Expresse esse conjunto de equações algébricas lineares na forma matricial e, então, use o MATLAB para calcular as incógnitas.

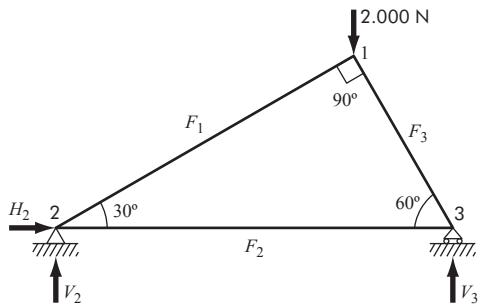


FIGURA P8.10

8.11 Considere o sistema de quatro massas e três molas da Figura P8.11. A determinação das equações de movimento a partir de $\sum F_x = m a_x$, para cada massa, usando seu diagrama de corpo livre, resulta nas seguintes equações diferenciais:

$$\ddot{x}_1 + \left(\frac{k_1 + k_2}{m_1} \right) x_1 - \left(\frac{k_2}{m_1} \right) x_2 = 0$$

$$\ddot{x}_2 - \left(\frac{k_2}{m_2} \right) x_1 + \left(\frac{k_2 + k_3}{m_2} \right) x_2 - \left(\frac{k_3}{m_2} \right) x_3 = 0$$

$$\ddot{x}_3 - \left(\frac{k_3}{m_3} \right) x_2 + \left(\frac{k_3 + k_4}{m_3} \right) x_3 = 0$$

onde $k_1 = k_4 = 10$ N/m, $k_2 = k_3 = 40$ N/m e $m_1 = m_2 = m_3 = 1$ kg. As três equações podem ser escritas na forma matricial:

$0 = \{\text{Vetor de aceleração}\} + [\text{matriz } k/m]\{\text{Vetor de deslocamento } x\}$

Em um instante de tempo específico em que $x_1 = 0,05$ m, $x_2 = 0,04$ m e $x_3 = 0,03$ m, isso forma uma matriz tridiagonal. Use o MATLAB para determinar a aceleração de cada massa.

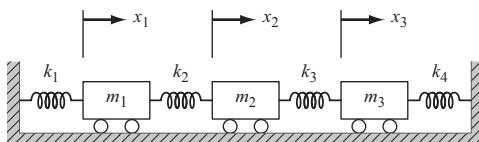


FIGURA P8.11

8.12 Efetue os mesmos cálculos como no Exemplo 8.2, mas use cinco saltadores com as seguintes características:

Saltador	Massa (kg)	Constante da mola (N/m)	Comprimento da corda não esticada (m)
1	65	80	10
2	75	40	10
3	60	70	10
4	75	100	10
5	90	20	10

8.13 Três massas estão suspensas verticalmente por uma série de molas idênticas, onde a massa 1 está na parte superior e a massa 3, na inferior. Se $g = 9,81$ m/s², $m_1 = 2$ kg, $m_2 = 2,5$ kg, $m_3 = 3$ kg e k 's = 15 kg/s², use o MATLAB para calcular os deslocamentos x .

8.14 Efetue os mesmos cálculos como na Seção 8.3, porém para o circuito da Figura P8.14.

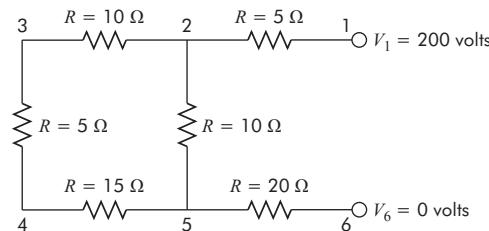


FIGURA P8.14

8.15 Efetue os mesmos cálculos como na Seção 8.3, porém para o circuito da Figura P8.15.

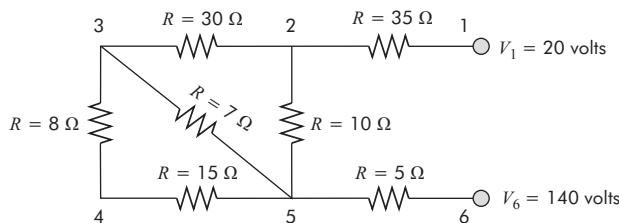


FIGURA P8.15

Eliminação de Gauss

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é descrever o algoritmo da eliminação de Gauss para solução de equações algébricas lineares. Após esta leitura completa, você será capaz de:

- ✓ Aprender como resolver conjuntos pequenos de equações lineares com o método gráfico e com a regra de Cramer.
- ✓ Entender como implementar a eliminação progressiva e a substituição regressiva como na eliminação de Gauss.
- ✓ Entender como contar os flops para avaliar a eficiência de um algoritmo.
- ✓ Entender os conceitos de singularidade e mau condicionamento.
- ✓ Entender como o pivotamento parcial é implementado e como difere do pivotamento completo.
- ✓ Aprender como calcular o determinante como parte do algoritmo da eliminação de Gauss com pivotamento parcial.
- ✓ Reconhecer como a estrutura de banda de um sistema tridiagonal pode ser explorada a fim de obter soluções extremamente eficientes.

No fim do Capítulo 8, afirmamos que o MATLAB fornece dois métodos simples e diretos para resolver sistemas de equações algébricas lineares: a divisão à esquerda,

```
>> x = A\b
```

e a inversão da matriz,

```
>> x = inv(A)*b
```

Os Capítulos 9 e 10 fornecem informações sobre como essas soluções são obtidas. Esse material é incluído para fornecer uma visão de como o MATLAB opera; e também tem a intenção de mostrar como o usuário pode desenvolver seus próprios algoritmos em ambientes computacionais que não têm os recursos nativos do MATLAB.

A técnica descrita neste capítulo é chamada de eliminação de Gauss, pois envolve combinar equações para eliminar variáveis. Apesar de ser um dos métodos mais antigos para resolver equações simultâneas, mantém-se como um dos algoritmos mais importantes em uso hoje, e é a base da resolução de equações lineares em muitos pacotes de software populares, incluindo o MATLAB.

9.1 RESOLUÇÃO DE UM NÚMERO PEQUENO DE EQUAÇÕES

Antes de prosseguir para a eliminação de Gauss, serão descritos vários métodos que são apropriados para resolver um conjunto pequeno de equações simultâneas ($n \leq 3$) e que não exigem um computador. Eles incluem o método gráfico, a regra de Cramer e a eliminação de variáveis.

9.1.1 Método gráfico

Uma solução gráfica para duas equações pode ser obtida esboçando-se seus gráficos em coordenadas cartesianas, com um eixo correspondente a x_1 e o outro a x_2 . Como estamos lidando com sistemas lineares, cada equação é uma reta. Por exemplo, suponha que tenhamos as seguintes equações:

$$\begin{aligned} 3x_1 + 2x_2 &= 18 \\ -x_1 + 2x_2 &= 2 \end{aligned}$$

Considerando que x_1 é a abscissa, podemos resolver cada uma dessas equações para x_2 :

$$\begin{aligned} x_2 &= -\frac{3}{2}x_1 + 9 \\ x_2 &= \frac{1}{2}x_1 + 1 \end{aligned}$$

Assim, as equações estão agora na forma de equações de reta – isto é, $x_2 = (\text{inclinação})x_1 + \text{interseção com o eixo}$. Quando essas retas são traçadas, os valores de x_1 e x_2 na interseção das retas representam a solução (Figura 9.1). Nesse caso, a solução é $x_1 = 4$ e $x_2 = 3$.

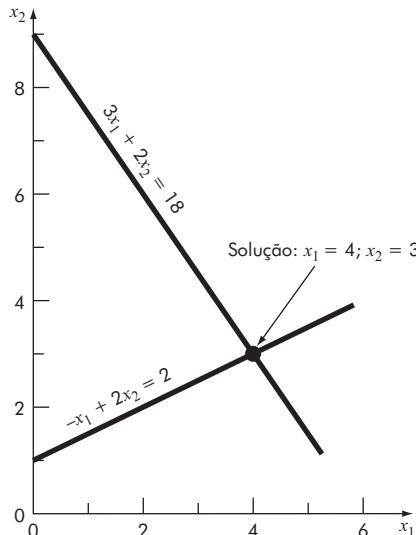


FIGURA 9.1 Solução gráfica de um conjunto de duas equações algébricas lineares simultâneas. A interseção das retas representa a solução.

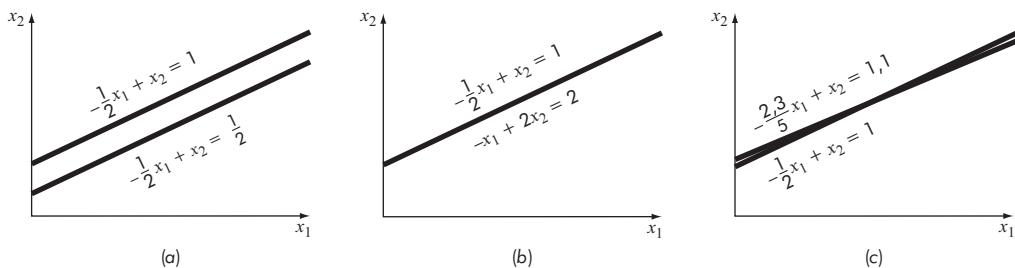


FIGURA 9.2 Descrição gráfica de sistemas singulares e mal condicionados: (a) nenhuma solução, (b) infinitas soluções e (c) sistemas mal condicionados, nos quais as inclinações são tão próximas que o ponto de interseção é difícil de detectar visualmente.

Para três equações simultâneas, cada equação seria representada por um plano em um sistema de coordenadas tridimensional. O ponto no qual os planos se interceptam irá representar a solução. Para mais de três equações, os métodos gráficos não funcionam e, consequentemente, têm pouco valor prático para resolver equações simultâneas. No entanto, às vezes são úteis para visualizar as propriedades das soluções.

Por exemplo, a Figura 9.2 descreve três casos que podem ser problemáticos quando se resolve um conjunto de equações lineares. A Figura 9.2a mostra o caso em que duas equações representam retas paralelas. Para tais situações, não há solução, pois as retas não se cruzam. A Figura 9.2b mostra o caso em que as duas retas são coincidentes. Para tais situações, há um número infinito de soluções. Ambos os tipos de sistemas são chamados de *singulares*.

Há, ainda, sistemas que estão muito próximos de serem singulares (Figura 9.2c) os chamados *mal condicionados*, que também podem causar problemas. Em termos gráficos, isso significa que é difícil identificar o ponto exato no qual as retas se interceptam, e que também causarão problemas quando forem encontrados durante as soluções numéricas de equações lineares, porque são extremamente sensíveis a erros de arredondamento.

9.1.2 Determinantes e a regra de Cramer

A regra de Cramer é outra técnica de resolução que é melhor adequada a um pequeno número de equações. Antes de descrever esse método, será apresentado brevemente o conceito de determinantes, que é usado para implementar a regra de Cramer. Além disso, o determinante é relevante para avaliar o mau condicionamento de uma matriz.

Determinantes. O determinante pode ser ilustrado para um conjunto de três equações:

$$[A]\{x\} = \{b\}$$

onde $[A]$ é a matriz dos coeficientes

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

O determinante D desse sistema é formado a partir dos coeficientes de $[A]$ e representado como

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Apesar de o determinante D e a matriz de coeficientes $[A]$ serem compostos dos mesmos elementos, são conceitos matemáticos completamente diferentes. Por essa razão, são distinguidos visualmente usando-se colchetes para delimitar a matriz e linhas retas para delimitar o determinante. Ao contrário da matriz, o determinante é um único número. Por exemplo, o valor do determinante para duas equações simultâneas

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

é calculado por

$$D = a_{11}a_{22} - a_{12}a_{21}$$

Para o caso de terceira ordem, o determinante pode ser calculado como

$$D = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \quad (9.1)$$

onde os determinantes 2×2 são chamados de *menores*.

EXEMPLO 9.1 Determinantes

Calcule os valores dos determinantes dos sistemas representados nas Figuras 9.1 e 9.2.

Solução. Para a Figura 9.1:

$$D = \begin{vmatrix} 3 & 2 \\ -1 & 2 \end{vmatrix} = 3(2) - 2(-1) = 8$$

Para a Figura 9.2a:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -\frac{1}{2} & 1 \end{vmatrix} = -\frac{1}{2}(1) - 1\left(\frac{-1}{2}\right) = 0$$

Para a Figura 9.2b:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -1 & 2 \end{vmatrix} = -\frac{1}{2}(2) - 1(-1) = 0$$

Para a Figura 9.2c:

$$D = \begin{vmatrix} -\frac{1}{2} & 1 \\ -\frac{2,3}{5} & 1 \end{vmatrix} = -\frac{1}{2}(1) - 1\left(\frac{-2,3}{5}\right) = -0,04$$

No exemplo anterior, os sistemas singulares têm determinantes nulos. Além disso, os resultados sugerem que o sistema que é quase singular (Figura 9.2c) tem um determinante próximo de zero. Essas ideias serão examinadas mais a fundo em nossa discussão subsequente sobre mau condicionamento no Capítulo 11.

Regra de Cramer. Essa regra estabelece que cada incógnita em um sistema de equações algébricas lineares pode ser expressa como uma fração de dois determinantes com denominador D e com numerador obtido a partir de D , trocando-se a coluna de coeficientes da incógnita em questão pelas constantes b_1, b_2, \dots, b_n . Por exemplo, para três equações, x_1 seria calculado como

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}}{D}$$

EXEMPLO 9.2 Regra de Cramer

Use a regra de Cramer para resolver

$$0,3x_1 + 0,52x_2 + x_3 = -0,01$$

$$0,5x_1 + x_2 + 1,9x_3 = 0,67$$

$$0,1x_1 + 0,3x_2 + 0,5x_3 = -0,44$$

Solução. O determinante D pode ser definido como [Equação (9.1)]:

$$D = 0,3 \begin{vmatrix} 1 & 1,9 \\ 0,3 & 0,5 \end{vmatrix} - 0,52 \begin{vmatrix} 0,5 & 1,9 \\ 0,1 & 0,5 \end{vmatrix} + 1 \begin{vmatrix} 0,5 & 1 \\ 0,1 & 0,3 \end{vmatrix} = -0,0022$$

A solução pode ser calculada como

$$x_1 = \frac{\begin{vmatrix} -0,01 & 0,52 & 1 \\ 0,67 & 1 & 1,9 \\ -0,44 & 0,3 & 0,5 \end{vmatrix}}{-0,0022} = \frac{0,03278}{-0,0022} = -14,9$$

$$x_2 = \frac{\begin{vmatrix} 0,3 & -0,01 & 1 \\ 0,5 & 0,67 & 1,9 \\ 0,1 & -0,44 & 0,5 \end{vmatrix}}{-0,0022} = \frac{0,0649}{-0,0022} = -29,5$$

$$x_3 = \frac{\begin{vmatrix} 0,3 & 0,52 & -0,01 \\ 0,5 & 1 & 0,67 \\ 0,1 & 0,3 & -0,44 \end{vmatrix}}{-0,0022} = \frac{-0,04356}{-0,0022} = 19,8$$

A função det. O determinante pode ser calculado diretamente no MATLAB com a função `det`. Por exemplo, usando o sistema do exemplo anterior:

```
>> A=[0.3 0.52 1; 0.5 1 1.9; 0.1 0.3 0.5];
```

```

>> D=det(A)
D =
-0.0022

A regra de Cramer pode ser aplicada para calcular  $x_1$  como em

>> A(:,1)=[-0.01;0.67;-0.44]

A =
-0.0100    0.5200    1.0000
 0.6700    1.0000    1.9000
-0.4400    0.3000    0.5000
>> x1=det(A)/D
x1 =
-14.9000

```

Para mais de três equações, a regra de Cramer torna-se impraticável, pois à medida que o número de equações cresce, os determinantes demoram mais para ser calculados à mão (ou pelo computador). Consequentemente, são usadas alternativas mais eficientes, algumas baseadas na última técnica de solução não computacional abordada na Seção 9.1.3 – a eliminação de variáveis.

9.1.3 A eliminação de variáveis

A eliminação de variáveis pela combinação de equações é uma abordagem algébrica que pode ser ilustrada para um conjunto de duas equações:

$$a_{11}x_1 + a_{12}x_2 = b_1 \quad (9.2)$$

$$a_{21}x_1 + a_{22}x_2 = b_2 \quad (9.3)$$

A estratégia básica é multiplicar as equações por constantes de modo que uma das variáveis seja eliminada quando as duas equações são combinadas. O resultado é uma única equação, que pode ser resolvida determinando a variável restante. Então, esse valor pode ser substituído em qualquer uma das equações originais para calcular a outra variável.

Por exemplo, a Equação (9.2) pode ser multiplicada por a_{21} , e a Equação (9.3) por a_{11} para fornecer

$$a_{21}a_{11}x_1 + a_{21}a_{12}x_2 = a_{21}b_1 \quad (9.4)$$

$$a_{11}a_{21}x_1 + a_{11}a_{22}x_2 = a_{11}b_2 \quad (9.5)$$

Subtrair a Equação (9.4) da Equação (9.5) irá, portanto, eliminar o termo x_1 da equação, produzindo

$$a_{11}a_{22}x_2 - a_{21}a_{12}x_2 = a_{11}b_2 - a_{21}b_1$$

que pode ser resolvida por

$$x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{21}a_{12}} \quad (9.6)$$

A Equação (9.6) pode então ser substituída na Equação (9.2), a qual é resolvida por

$$x_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{21}a_{12}} \quad (9.7)$$

Observe que as Equações (9.6) e (9.7) seguem diretamente a regra de Cramer, que estabelece que:

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{21}a_{12}}$$

$$x_2 = \frac{\begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{21}a_{12}}$$

A eliminação de variáveis pode ser estendida a sistemas com mais do que duas ou três equações. Porém, os numerosos cálculos necessários para sistemas maiores tornam o método extremamente tedioso para ser implementado à mão. Mas, como descrito na Seção 9.2, a técnica pode ser formalizada e rapidamente programada para o computador.

9.2 ELIMINAÇÃO DE GAUSS INGÊNUA

Na Seção 9.1.3, a eliminação de variáveis foi usada para resolver duas equações simultâneas. O processo consiste em dois passos (ver Figura 9.3):

- As equações são manipuladas para eliminar uma das variáveis das equações. O resultado desse passo de eliminação foi a obtenção de uma equação com uma variável.
- Consequentemente, ela pode ser resolvida diretamente e o resultado substituído regressivamente em uma das equações originais para determinar a variável remanescente.

Essa abordagem básica pode ser estendida a conjuntos maiores de equações desenvolvendo-se um esquema sistemático ou um algoritmo para eliminar variáveis e substituí-las regressivamente. A eliminação de Gauss é o mais básico desses esquemas.

Esta seção inclui técnicas sistemáticas para eliminação progressiva e substituição regressiva que compõem a eliminação de Gauss e apesar de serem hipoteticamente adequadas para implementação em computadores, algumas modificações serão necessárias para se obter um algoritmo confiável. Em particular, o programa de computador deve evitar a divisão por zero. O método a seguir é chamado de eliminação de Gauss “ingênuo” porque não evita esse problema. A Seção 9.3 tratará das características adicionais necessárias para um programa de computador efetivo.

A abordagem é planejada para resolver um conjunto arbitrário de n equações:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \quad (9.8a)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \quad (9.8b)$$

$$\vdots \qquad \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n \quad (9.8c)$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array} \right] \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{(a) Eliminação progressiva}$$

↓

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a'_{22} & a'_{23} & & b'_2 \\ a''_{33} & & & b''_3 \end{array} \right] \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{(b) Substituição regressiva}$$

↓

$$\begin{aligned} x_3 &= b''_3/a''_{33} \\ x_2 &= (b'_2 - a'_{23}x_3)/a'_{22} \\ x_1 &= (b_1 - a_{13}x_3 - a_{12}x_2)/a_{11} \end{aligned}$$

FIGURA 9.3 As duas fases da eliminação de Gauss: (a) eliminação progressiva e (b) substituição regressiva.

De forma semelhante à solução de duas equações, a técnica para n equações consiste em duas fases: a eliminação de variáveis e a solução por meio da substituição regressiva.

Eliminação Progressiva de Variáveis. A primeira fase é projetada para reduzir o conjunto de equações a um sistema triangular superior (Figura 9.3a). O passo inicial será eliminar a variável x_1 da segunda até a n -ésima equação. Para isso, multiplique a Equação (9.8a) por a_{21}/a_{11} para obter

$$a_{21}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \frac{a_{21}}{a_{11}}a_{13}x_3 + \cdots + \frac{a_{21}}{a_{11}}a_{1n}x_n = \frac{a_{21}}{a_{11}}b_1 \quad (9.9)$$

Essa equação pode ser subtraída da Equação (9.8b) para se obter

$$\left(a_{22} - \frac{a_{21}}{a_{11}}a_{12} \right)x_2 + \cdots + \left(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n} \right)x_n = b_2 - \frac{a_{21}}{a_{11}}b_1$$

ou

$$a'_{22}x_2 + \cdots + a'_{2n}x_n = b'_2$$

onde as linhas indicam que os elementos foram modificados de seus valores originais.

O processo é repetido para as equações restantes. Por exemplo, a Equação (9.8a) pode ser multiplicada por a_{31}/a_{11} e o resultado subtraído da terceira equação. Repetindo o procedimento para as equações restantes, obtém-se o seguinte sistema modificado:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \quad (9.10a)$$

$$a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n = b'_2 \quad (9.10b)$$

$$a'_{32}x_2 + a'_{33}x_3 + \cdots + a'_{3n}x_n = b'_3 \quad (9.10c)$$

$$\vdots \qquad \vdots$$

$$a'_{n2}x_2 + a'_{n3}x_3 + \cdots + a'_{nn}x_n = b'_n \quad (9.10d)$$

Nos passos precedentes, a Equação (9.8a) é chamada de *equação de pivô* e a_{11} é chamado de *coeficiente* ou *elemento pivô*. Observe que o processo de multiplicar a primeira linha por a_{21}/a_{11} é equivalente a dividi-la por a_{11} e multiplicá-la por a_{21} . Às vezes, essa operação de divisão é chamada de *normalização*. Fazemos essa distinção porque um elemento pivô nulo pode interferir na normalização, já que provoca uma divisão por zero. Voltaremos a esse assunto importante depois de completar a descrição da eliminação de Gauss ingênua.

O próximo passo é eliminar x_2 das Equações (9.10c) até (9.10d), multiplicando a Equação (9.14b) por a'_{32}/a'_{22} e subtraindo o resultado da Equação (9.10c). Faça uma eliminação semelhante nas equações restantes para obter

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

$$a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n = b'_2$$

$$a''_{33}x_3 + \cdots + a''_{3n}x_n = b''_3$$

$$\vdots \qquad \vdots$$

$$a''_{n3}x_3 + \cdots + a''_{nn}x_n = b''_n$$

onde as linhas duplas indicam que os elementos foram modificados duas vezes.

O processo pode continuar usando-se as equações pivô remanescentes. A última manipulação na sequência é usar a $(n - 1)$ -ésima equação para eliminar o termo x_{n-1} da n -ésima equação. Nesse ponto, o sistema terá sido transformado em um sistema triangular superior.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \quad (9.11a)$$

$$a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n = b'_2 \quad (9.11b)$$

$$a''_{33}x_3 + \cdots + a''_{3n}x_n = b''_3 \quad (9.11c)$$

$$\ddots \qquad \vdots$$

$$a_{nn}^{(n-1)}x_n = b_n^{(n-1)} \quad (9.11d)$$

Substituição Regressiva. A Equação (9.11d) pode agora ser resolvida para determinar x_n :

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}} \quad (9.12)$$

Esse resultado pode ser substituído de volta na $(n - 1)$ -ésima equação para determinar x_{n-1} . O processo, que é repetido para determinar os x 's remanescentes, pode ser representado pela seguinte equação:

$$x_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)}x_j}{a_{ii}^{(i-1)}} \quad \text{para } i = n-1, n-2, \dots, 1 \quad (9.13)$$

EXEMPLO 9.3 Eliminação de Gauss Ingênua

Use a eliminação de Gauss para resolver

$$3x_1 - 0,1x_2 - 0,2x_3 = 7,85 \quad (\text{E9.3.1})$$

$$0,1x_1 + 7x_2 - 0,3x_3 = -19,3 \quad (\text{E9.3.2})$$

$$0,3x_1 - 0,2x_2 + 10x_3 = 71,4 \quad (\text{E9.3.3})$$

Solução. A primeira parte do processo é a eliminação progressiva. Multiplique a Equação (E9.3.1) por 0,1/3 e subtraia o resultado da Equação (E9.3.2) para obter

$$7,00333x_2 - 0,293333x_3 = -19,5617$$

Então, multiplique a Equação (E9.3.1) por 0,3/3 e subtraia da Equação (E9.3.3) para eliminar x_1 . Depois dessas operações, o conjunto de equações é

$$3x_1 - 0,1x_2 - 0,2x_3 = 7,85 \quad (\text{E9.3.4})$$

$$7,00333x_2 - 0,293333x_3 = -19,5617 \quad (\text{E9.3.5})$$

$$-0,190000x_2 + 10,0200x_3 = 70,6150 \quad (\text{E9.3.6})$$

Para completar a eliminação progressiva, x_2 deve ser removida da Equação (E9.3.6). Para isso, multiplique a Equação (E9.3.5) por $-0,190000/7,00333$ e subtraia o resultado da Equação (E9.3.6), o que elimina x_2 da terceira equação e reduz o sistema a um sistema triangular superior, como em

$$3x_1 - 0,1x_2 - 0,2x_3 = 7,85 \quad (\text{E9.3.7})$$

$$7,00333x_2 - 0,293333x_3 = -19,5617 \quad (\text{E9.3.8})$$

$$10,0120x_3 = 70,0843 \quad (\text{E9.3.9})$$

Agora, é possível resolver essas equações por substituição regressiva. Primeiro, a Equação (E9.3.9) pode ser resolvida para determinar

$$x_3 = \frac{70,0843}{10,0120} = 7,00003$$

Depois esse resultado pode ser substituído na Equação (E9.3.8), que pode ser resolvida por

$$x_2 = \frac{-19,5617 + 0,293333(7,00003)}{7,00333} = -2,50000$$

Por fim, $x_3 = 7,00003$ e $x_2 = -2,50000$ podem ser substituídos na Equação (E9.3.7), a qual pode ser resolvida por

$$x_1 = \frac{7,85 + 0,1(-2,50000) + 0,2(7,00003)}{3} = 3,00000$$

Embora exista um pequeno erro de arredondamento, os resultados estão muito próximos da solução exata de $x_1 = 3$, $x_2 = -2,5$ e $x_3 = 7$, o que pode ser verificado substituindo-se os resultados no conjunto original de equações

$$3(3) - 0,1(-2,5) - 0,2(7,00003) = 7,84999 \cong 7,85$$

$$0,1(3) + 7(-2,5) - 0,3(7,00003) = -19,30000 = -19,3$$

$$0,3(3) - 0,2(-2,5) + 10(7,00003) = 71,4003 \cong 71,4$$

```

function x = GaussIngenua(A,b)
% GaussIngenua: eliminação de Gauss ingênua
%   x = GaussIngenua(A,b): eliminação de Gauss sem pivotamento.
%   entrada:
%     A = matriz dos coeficientes
%     b = vetor do lado direito
%   saída:
%     x = vetor solução
[m,n] = size(A);
if m~=n, error('A matriz A deve ser quadrada'); end
nb = n+1;
Aum = [A b];
% eliminação progressiva
for k = 1:n-1
    for i = k+1:n
        fator = Aum(i,k)/Aum(k,k);
        Aum(i,k:nb) = Aum(i,k:nb)-fator*Aum(k,k:nb);
    end
end
% substituição regressiva
x = zeros(n,1);
x(n) = Aum(n,nb)/Aum(n,n);
for i = n-1:-1:1
    x(i) = (Aum(i,nb)-Aum(i,i+1:n)*x(i+1:n))/Aum(i,i);
end

```

FIGURA 9.4 Função do MATLAB para implementar a eliminação de Gauss ingênua.

9.2.1 Função do MATLAB: GaussIngenua

Uma função do MATLAB que implementa a eliminação de Gauss ingênua é apresentada na Figura 9.4. Observe que a matriz de coeficientes, A , e o vetor do lado direito, b , estão combinados em uma matriz aumentada, A_{um} . Assim, as operações são efetuadas em A_{um} em vez de separadamente em A e b .

Dois laços aninhados fornecem uma representação concisa da fase de eliminação progressiva. Um laço externo move-se para baixo na matriz de uma linha pivô para a próxima. O laço interno move-se para baixo da linha pivô para cada uma das linhas subsequentes onde a eliminação é efetuada. Finalmente, a eliminação efetiva é representada por uma única linha que aproveita os recursos do MATLAB para realizar operações matriciais.

A fase de substituição regressiva segue diretamente das Equações (9.12) e (9.13). Novamente, os recursos do MATLAB destinados a realizar operações matriciais permitem que a Equação (9.13) seja programada como uma única linha.

9.2.2 Contagem de operações

O tempo de execução da eliminação de Gauss depende da quantidade de *operações com ponto flutuante* (ou *flops*) envolvidos no algoritmo. Em computadores modernos usando coprocessadores matemáticos, o tempo consumido para efetuar uma

adição/subtração ou uma multiplicação/divisão é mais ou menos o mesmo; portanto, a contagem dessas operações fornece informação sobre quais partes do algoritmo consomem mais tempo e sobre como o tempo de computação cresce à medida que o sistema fica maior.

Antes de analisar a eliminação de Gauss ingênua, é preciso definir algumas quantidades que facilitam a contagem de operações:

$$\sum_{i=1}^m cf(i) = c \sum_{i=1}^m f(i) \quad \sum_{i=1}^m f(i) + g(i) = \sum_{i=1}^m f(i) + \sum_{i=1}^m g(i) \quad (9.14a,b)$$

$$\sum_{i=1}^m 1 = 1 + 1 + 1 + \dots + 1 = m \quad \sum_{i=k}^m 1 = m - k + 1 \quad (9.14c,d)$$

$$\sum_{i=1}^m i = 1 + 2 + 3 + \dots + m = \frac{m(m+1)}{2} = \frac{m^2}{2} + O(m) \quad (9.14e)$$

$$\sum_{i=1}^m i^2 = 1^2 + 2^2 + 3^2 + \dots + m^2 = \frac{m(m+1)(2m+1)}{6} = \frac{m^3}{3} + O(m^2) \quad (9.14f)$$

onde $O(m^n)$ significa “termos de ordem m^n e menores.”

Agora, examinaremos o algoritmo de eliminação de Gauss ingênua (Figura 9.4) em detalhes. O processo deve ser iniciado com a contagem dos flops no estágio de eliminação. Na primeira passagem pelo laço externo, $k = 1$. Então, os limites no laço interno são de $i = 2$ até n . De acordo com a Equação (9.14d), isso significa que o número de iterações do laço interno será

$$\sum_{i=2}^n 1 = n - 2 + 1 = n - 1 \quad (9.15)$$

Para cada uma dessas iterações, há uma operação de divisão para calcular o fator. A próxima linha executa uma multiplicação e uma subtração para cada elemento da coluna de 2 até nb . Como $nb = n + 1$, indo de 2 até nb resulta em n multiplicações e n subtrações. Juntamente com a única divisão, isso equivale a $n + 1$ multiplicações/divisões e n adições/subtrações para cada iteração do laço interno. O total para a primeira passagem pelo laço externo é, portanto, $(n-1)(n+1)$ multiplicações/divisões e $(n-1)n$ adições/subtrações.

Um raciocínio semelhante pode ser usado para estimar os flops para as iterações subsequentes do laço externo. Isso pode ser resumido por

Laço externo k	Laço interno i	Flops adição/ subtração	Flops multiplicação/ divisão
1	2, n	$(n-1)n$	$(n-1)(n+1)$
2	3, n	$(n-2)(n-1)$	$(n-2)n$
\vdots	\vdots		
k	$k+1, n$	$(n-k)(n+1-k)$	$(n-k)(n+2-k)$
\vdots	\vdots		
$n-1$	n, n	$(1)(2)$	$(1)(3)$

Então, o total de flops de adição/subtração para a eliminação pode ser calculado por

$$\sum_{k=1}^{n-1} (n-k)(n+1-k) = \sum_{k=1}^{n-1} [n(n+1) - k(2n+1) + k^2] \quad (9.16)$$

ou

$$n(n+1) \sum_{k=1}^{n-1} 1 - (2n+1) \sum_{k=1}^{n-1} k + \sum_{k=1}^{n-1} k^2 \quad (9.17)$$

Aplicando algumas das relações da Equação (9.14), obtemos

$$[n^3 + O(n)] - [n^3 + O(n^2)] + \left[\frac{1}{3}n^3 + O(n^2) \right] = \frac{n^3}{3} + O(n) \quad (9.18)$$

Uma análise similar para os flops de multiplicação/divisão resulta em

$$[n^3 + O(n^2)] - [n^3 + O(n)] + \left[\frac{1}{3}n^3 + O(n^2) \right] = \frac{n^3}{3} + O(n^2) \quad (9.19)$$

Somando esses resultados, tem-se

$$\frac{2n^3}{3} + O(n^2) \quad (9.20)$$

Assim, o número total de flops é igual a $2n^3/3$ mais uma componente adicional proporcional a termos de ordem n^2 e menores. O resultado é escrito dessa forma porque, à medida que n fica maior, o $O(n^2)$ e termos menores ficam desprezíveis. Pode-se então concluir, justificadamente, que para n grande, o esforço envolvido na eliminação progressiva converge para $2n^3/3$.

Como apenas um único laço foi utilizado, a substituição regressiva é muito mais simples de ser avaliada. O número de flops de adição/subtração é igual a $n(n-1)/2$. Devido à divisão extra anterior ao laço, o número de flops de multiplicação/divisão é $n(n+1)/2$. Isso pode ser somado para se obter um total de

$$n^2 + O(n) \quad (9.21)$$

Assim, o esforço total na eliminação de Gauss ingênua pode ser representado por

$$\underbrace{\frac{2n^3}{3} + O(n^2)}_{\text{Eliminação progressiva}} + \underbrace{n^2 + O(n)}_{\text{Substituição regressiva}} \xrightarrow{\text{quando } n \text{ cresce}} \frac{2n^3}{3} + O(n^2) \quad (9.22)$$

Duas conclusões gerais úteis podem ser tiradas dessa análise:

1. À medida que o sistema fica maior, o tempo de computação cresce bastante. Como mostra a Tabela 9.1, a quantidade de flops cresce quase três ordens de grandeza para cada aumento no número de equações.
2. A maior parte do esforço vem do passo de eliminação. Assim, os esforços para tornar o algoritmo mais eficiente devem concentrar-se nesse passo.

TABELA 9.1 Número de flops para a eliminação de Gauss ingênua.

<i>n</i>	Eliminação	Substituição regressiva	Total de flops	$2n^3/3$	Porcentagem devida à eliminação
10	705	100	805	667	87,58%
100	671.550	10.000	681.550	666.667	98,53%
1.000	$6,67 \times 10^8$	1×10^6	$6,68 \times 10^8$	$6,67 \times 10^8$	99,85%

9.3 PIVOTAMENTO

A principal razão para que a técnica anterior seja chamada de “ingênua” é que tanto durante a fase de eliminação quanto durante a de substituição é possível que ocorra uma divisão por zero. Por exemplo, se for usada a eliminação de Gauss ingênua para resolver

$$\begin{aligned} 2x_2 + 3x_3 &= 8 \\ 4x_1 + 6x_2 + 7x_3 &= -3 \\ 2x_1 - 3x_2 + 6x_3 &= 5 \end{aligned}$$

a normalização da primeira linha iria envolver a divisão por $a_{11} = 0$. Problemas também podem surgir quando o elemento pivô é muito próximo, em vez de exatamente igual a zero, porque, se a ordem de grandeza do elemento pivô é pequena comparada a dos outros elementos, então podem ocorrer erros de arredondamento.

Dessa forma, antes que cada linha seja normalizada, é vantajoso determinar o maior coeficiente disponível na coluna abaixo do elemento pivô. As linhas podem então ser trocadas, de modo que o maior coeficiente seja o elemento pivô. Isso é chamado de *pivotamento parcial*.

Se procurarmos o maior elemento também nas colunas, além de nas linhas, e então o trocarmos, o processo é chamado de *pivotamento completo*, o que é raramente usado, porque a maior parte das melhorias é obtida simplesmente com o pivotamento parcial. Além disso, trocar colunas muda a ordem dos x 's e, consequentemente, acrescenta uma complexidade significativa, normalmente, injustificada ao programa de computador.

O próximo exemplo ilustra as vantagens do pivotamento parcial, que, além de evitar a divisão por zero, também minimiza os erros de arredondamento; assim, serve como um remédio parcial para o mau condicionamento.

EXEMPLO 9.4 Pivotamento parcial

Use a eliminação de Gauss para resolver

$$\begin{aligned} 0,0003x_1 + 3,0000x_2 &= 2,0001 \\ 1,0000x_1 + 1,0000x_2 &= 1,0000 \end{aligned}$$

Observe que, dessa forma, o primeiro elemento pivô, $a_{11} = 0,0003$, é muito próximo de zero. Então, repita os cálculos, mas use pivotamento parcial, trocando a ordem das equações. As soluções exatas são $x_1 = 1/3$ e $x_2 = 2/3$.

Solução. Multiplicando a primeira equação por $1/(0,0003)$, obtém-se

$$x_1 + 10.000x_2 = 6667$$

que pode ser usado para eliminar x_1 da segunda equação:

$$-9.999x_2 = -6.666$$

a qual pode ser resolvida a fim de obter $x_2 = 2/3$. Esse resultado pode ser substituído na primeira equação para calcular x_1 :

$$x_1 = \frac{2,0001 - 3(2/3)}{0,0003} \quad (\text{E9.4.1})$$

Entretanto, devido ao cancelamento na subtração, o resultado é muito sensível ao número de algarismos significativos nos cálculos:

Algarismos significativos	x_2	x_1	Valor absoluto do erro relativo percentual para x_1
3	0,667	-3,33	1.099
4	0,6667	0,0000	100
5	0,66667	0,30000	10
6	0,666667	0,330000	1
7	0,6666667	0,3330000	0,1

Observe como a solução para x_1 é altamente dependente do número de algarismos significativos. Isso ocorre porque, na Equação (E9.4.1), estamos subtraindo dois números quase iguais. No entanto, se as equações forem resolvidas na ordem inversa, a linha com elemento pivô maior é normalizada. As equações são

$$\begin{aligned} 1,0000x_1 + 1,0000x_2 &= 1,0000 \\ 0,0003x_1 + 3,0000x_2 &= 2,0001 \end{aligned}$$

A eliminação e a substituição resultam em $x_2 = 2/3$. Para números diferentes de algarismos significativos, x_1 pode ser calculado da primeira equação, como em

$$x_1 = \frac{1 - (2/3)}{1}$$

Esse caso é muito menos sensível ao número de algarismos significativos nos cálculos:

Algarismos significativos	x_2	x_1	Valor absoluto do erro relativo percentual para x_1
3	0,667	0,333	0,1
4	0,6667	0,3333	0,01
5	0,66667	0,33333	0,001
6	0,666667	0,333333	0,0001
7	0,6666667	0,3333333	0,0000

Assim, a estratégia com pivotamento é muito mais satisfatória.

9.3.1 Função do MATLAB: GaussPivot

Uma função do MATLAB que implementa a eliminação de Gauss com pivotamento é apresentada na Figura 9.5. Ela é idêntica à função para a eliminação de Gauss

ingênuas apresentadas anteriormente, na Seção 9.2.1, com exceção da parte em negrito, que implementa o pivotamento parcial.

Observe que a função nativa do MATLAB `max` é usada para determinar o maior coeficiente disponível na coluna abaixo do elemento pivô. A função `max` tem a sintaxe:

```
[y,i] = max(x)
```

onde `y` é o maior elemento no vetor `x` e `i` é o índice correspondente a esse elemento.

```
function x = GaussPivot(A,b)
% GaussPivot: eliminação de Gauss com pivotamento
%   x = GaussPivot(A,b): eliminação de Gauss com pivotamento.
% entrada:
%   A = matriz dos coeficientes
%   b = vetor do lado direito
% saída:
%   x = vetor solução

[m,n]=size(A);
if m~=n, error('A matriz A deve ser quadrada'); end
nb=n+1;
Aum=[A b];
% eliminação progressiva
for k = 1:n-1
    % pivotamento parcial
    [maior,i]=max(abs(Aum(k:n,k)));
    ipr=i+k-1;
    if ipr~k
        Aum([k,ipr],:)=Aum([ipr,k],:);
    end
    for i = k+1:n
        fator=Aum(i,k)/Aum(k,k);
        Aum(i,k:nb)=Aum(i,k:nb)-fator*Aum(k,k:nb);
    end
end
% substituição regressiva
x=zeros(n,1);
x(n)=Aum(n,nb)/Aum(n,n);
for i = n-1:-1:1
    x(i)=(Aum(i,nb)-Aum(i,i+1:n)*x(i+1:n))/Aum(i,i);
end
```

FIGURA 9.5 Função do MATLAB para implementar a eliminação de Gauss com pivotamento.

9.3.2 Cálculo de determinantes usando eliminação de Gauss

No fim da Seção 9.1.2, foi sugerido que o cálculo do determinante a partir da expansão em menores era pouco prático para conjuntos grandes de equações. No entanto, como o determinante tem utilidade na avaliação do condicionamento do sistema, seria útil ter um método prático para calculá-lo.

Felizmente, a eliminação de Gauss fornece uma forma simples de fazer isso. O método se baseia no fato de que o determinante de uma matriz triangular pode ser calculado simplesmente como o produto dos elementos da diagonal

$$D = a_{11}a_{22}a_{33} \dots a_{nn}$$

A validade dessa afirmação pode ser ilustrada para um sistema 3×3 :

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{vmatrix}$$

onde o determinante pode ser calculado como [lembre-se da Equação (9.1)]:

$$D = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ 0 & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} 0 & a_{23} \\ 0 & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} 0 & a_{22} \\ 0 & 0 \end{vmatrix}$$

ou, calculando-se os menores:

$$D = a_{11}a_{22}a_{33} - a_{12}(0) + a_{13}(0) = a_{11}a_{22}a_{33}$$

Lembre-se de que o passo de eliminação progressiva na eliminação de Gauss resulta em um sistema triangular superior. Como o valor do determinante não muda no processo de eliminação progressiva, o determinante pode ser calculado simplesmente no fim desse processo por

$$D = a_{11}a'_{22}a''_{33} \dots a_{nn}^{(n-1)}$$

onde os subscritos significam o número de vezes que os elementos foram modificados pelo processo de eliminação. Assim, podemos aproveitar o esforço que já foi gasto para reduzir o sistema a uma forma triangular e, de brinde, obter uma estimativa simples do determinante.

Há uma pequena modificação nessa abordagem quando o programa utiliza pivotamento parcial. Nesse caso, o determinante troca de sinal toda vez que uma linha é pivotada. Uma forma de representar isso é modificar o cálculo do determinante como em

$$D = a_{11}a'_{22}a''_{33} \dots a_{nn}^{(n-1)}(-1)^p$$

onde p indica o número de vezes que as linhas foram pivotadas. Essa modificação pode ser incorporada de modo simples a um programa; meramente mantenha controle do número de pivotamentos que ocorreram durante os cálculos.

9.4 SISTEMAS TRIDIAGONAIS

Certas matrizes têm uma estrutura particular que pode ser explorada para desenvolver esquemas eficientes de solução. Por exemplo, uma matriz de banda é uma matriz quadrada que tem todos os elementos iguais a zero, com exceção de uma faixa centralizada na diagonal principal.

Um sistema *tridiagonal* tem uma largura de faixa de 3 e pode ser expresso no caso geral como

$$\left[\begin{array}{ccc|c} f_1 & g_1 & & r_1 \\ e_2 & f_2 & g_2 & r_2 \\ e_3 & f_3 & g_3 & r_3 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ e_{n-1} & f_{n-1} & g_{n-1} & r_{n-1} \\ e_n & f_n & & r_n \end{array} \right] \left\{ \begin{array}{l} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_{n-1} \\ x_n \end{array} \right\} = \left\{ \begin{array}{l} r_1 \\ r_2 \\ r_3 \\ \vdots \\ \vdots \\ r_{n-1} \\ r_n \end{array} \right\} \quad (9.23)$$

Observe que trocamos nossa notação para os coeficientes a 's e b 's para e 's, f 's, g 's e r 's a fim de evitar o armazenamento de grandes quantidades de zeros inúteis na matriz quadrada dos a 's. Essa modificação, que tem o objetivo de economizar espaço, é vantajosa, pois o algoritmo resultante requer menos memória de computador.

Um algoritmo para resolver tais sistemas pode ser diretamente derivado da eliminação de Gauss – isto é, utilizando eliminação progressiva e substituição regressiva. No entanto, como a maior parte dos elementos da matriz já é igual a zero, muito menos esforço é despendido do que para uma matriz cheia. Essa eficiência é ilustrada no exemplo a seguir.

EXEMPLO 9.5 Solução de um sistema tridiagonal

Resolva o seguinte sistema tridiagonal:

$$\left[\begin{array}{cccc} 2,04 & -1 & & \\ -1 & 2,04 & -1 & \\ & -1 & 2,04 & -1 \\ & & -1 & 2,04 \end{array} \right] \left\{ \begin{array}{l} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right\} = \left\{ \begin{array}{l} 40,8 \\ 0,8 \\ 0,8 \\ 200,8 \end{array} \right\}$$

Solução. De modo semelhante à eliminação de Gauss, o primeiro passo envolve a transformação da matriz para a forma triangular superior. Isso é feito multiplicando a primeira equação pelo fator e_2/f_1 e subtraindo o resultado da segunda equação, o que cria um zero no lugar de e_2 e transforma os outros coeficientes para os seguintes novos valores:

$$\begin{aligned} f_2 &= f_2 - \frac{e_2}{f_1} g_1 = 2,04 - \frac{-1}{2,04}(-1) = 1,550 \\ r_2 &= r_2 - \frac{e_2}{f_1} r_1 = 0,8 - \frac{-1}{2,04}(40,8) = 20,8 \end{aligned}$$

Observe que g_2 não é modificado porque o elemento acima dele na primeira linha é zero. Depois de efetuar cálculos similares para a terceira e quarta linhas, o sistema é transformado para a forma triangular superior

$$\left[\begin{array}{cccc} 2,04 & -1 & & \\ 1,550 & -1 & & \\ 1,395 & -1 & & \\ & 1,323 & & \end{array} \right] \left\{ \begin{array}{l} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right\} = \left\{ \begin{array}{l} 40,8 \\ 20,8 \\ 14,221 \\ 210,996 \end{array} \right\}$$

Agora, a substituição regressiva pode ser aplicada para gerar a solução final:

$$\begin{aligned}x_4 &= \frac{r_4}{f_4} = \frac{210,996}{1,323} = 159,480 \\x_3 &= \frac{r_3 - g_3 x_4}{f_3} = \frac{14,221 - (-1)159,480}{1,395} = 124,538 \\x_2 &= \frac{r_2 - g_2 x_3}{f_2} = \frac{20,800 - (-1)124,538}{1,550} = 93,778 \\x_1 &= \frac{r_1 - g_1 x_2}{f_1} = \frac{40,800 - (-1)93,778}{2,040} = 65,970\end{aligned}$$

9.4.1 Função do MATLAB: Tridiag

Uma função do MATLAB que resolve um sistema de equações tridiagonal é apresentada na Figura 9.6. Observe que o algoritmo não inclui pivotamento parcial. Embora o pivotamento seja necessário algumas vezes, a maior parte dos sistemas tridiagonais rotineiramente resolvidos em engenharia e na ciência não necessitam dele.

Lembre-se que o esforço computacional para a eliminação de Gauss era proporcional a n^3 . Devido à sua esparsidade, o esforço envolvido na solução de sistemas tridiagonais é proporcional a n . Consequentemente, o algoritmo na Figura 9.6 executa muito mais rápido que a eliminação de Gauss, particularmente para grandes sistemas.

```
function x = Tridiag(e,f,g,r)
% Tridiag: Solução de sistemas de equações tridiagonais
%   x = Tridiag(e,f,g,r): Solução de sistemas de equações
%   tridiagonais.
%   entrada:
%     e = vetor subdiagonal
%     f = vetor diagonal
%     g = vetor superdiagonal
%     r = vetor do lado direito
%   saída:
%     x = vetor solução
n=length(f);
% eliminação progressiva
for k = 2:n
    factor = e(k)/f(k-1);
    f(k) = f(k) - factor*g(k-1);
    r(k) = r(k) - factor*r(k-1);
end
% substituição regressiva
x(n) = r(n)/f(n);
for k = n-1:-1:1
    x(k) = (r(k)-g(k)*x(k+1))/f(k);
end
```

FIGURA 9.6 Função do MATLAB para resolver um sistema tridiagonal.

►9.5 ESTUDO DE CASO MODELO DE UMA HASTE AQUECIDA

Contextualização. Equações algébricas lineares podem surgir na modelagem de sistemas distribuídos. Por exemplo, a Figura 9.7 mostra uma haste longa e fina posicionada entre duas paredes mantidas em temperaturas constantes. Há um fluxo de calor através da haste (entre as paredes) assim como entre a haste e o ar circundante. No caso de estado estacionário (regime permanente), uma equação diferencial baseada na conservação do calor pode ser escrita para tal sistema como

$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0 \quad (9.24)$$

onde T é a temperatura ($^{\circ}\text{C}$), x é a distância ao longo da haste (m), h' é o coeficiente de transferência de calor entre a barra e o ar circundante (m^{-2}) e T_a é a temperatura do ar ($^{\circ}\text{C}$).

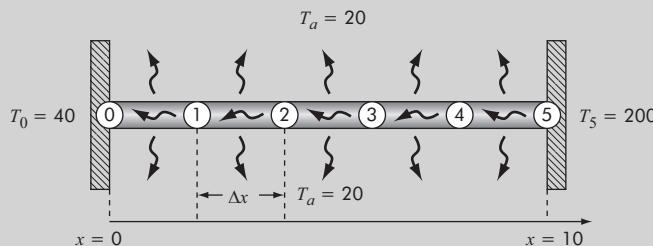


FIGURA 9.7 Haste uniforme não isolada posicionada entre duas paredes de temperaturas constantes, mas diferentes. A representação por diferenças finitas usa quatro nós interiores.

Dados os valores dos parâmetros, das funções forçantes e as condições de contorno, o cálculo pode ser usado para determinar uma solução analítica. Por exemplo, se $h' = 0,01$, $T_a = 20$, $T(0) = 40$ e $T(10) = 200$, a solução é

$$T = 73,4523e^{0,1x} - 53,4523e^{-0,1x} + 20 \quad (9.25)$$

Embora o cálculo forneça uma solução para esse caso, isso nem sempre é possível para todos os problemas desse tipo. Em tais casos, os métodos numéricos fornecem uma alternativa viável. Aqui, usaremos diferenças finitas divididas para transformar essa equação diferencial em um sistema tridiagonal de equações algébricas lineares, que pode ser imediatamente resolvido utilizando os métodos numéricos descritos neste capítulo.

Solução. A Equação (9.24) pode ser transformada em um conjunto de equações algébricas lineares pela consideração da haste como consistindo de uma série de nós. Por exemplo, a haste na Figura 9.7 está dividida em seis nós equidistantes. Como a haste tem um comprimento igual a 10 (unidades de comprimento), o espaçamento entre nós é $\Delta x = 2$.

O cálculo foi necessário para resolver a Equação (9.24) porque ela inclui uma derivada segunda. Como visto na Seção 4.3.4, aproximações por diferenças finitas divididas fornecem meios para transformar derivadas em formulações algébricas. Por exemplo, a derivada segunda em cada nó pode ser aproximada como

$$\frac{d^2T}{dx^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}$$

onde T_i denota a temperatura no nó i . Essa aproximação pode ser substituída na Equação (9.24) para fornecer

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + h'(T_a - T_i) = 0$$

Agrupando os termos comuns e substituindo os parâmetros, tem-se

$$-T_{i-1} + 2,04T_i - T_{i+1} = 0,8 \quad (9.26)$$

Assim, a Equação (9.24) foi transformada de uma equação diferencial em uma equação algébrica. Agora, a Equação (9.26) pode ser aplicada para cada um dos nós interiores:

$$\begin{aligned} -T_0 + 2,04T_1 - T_2 &= 0,8 \\ -T_1 + 2,04T_2 - T_3 &= 0,8 \\ -T_2 + 2,04T_3 - T_4 &= 0,8 \\ -T_3 + 2,04T_4 - T_5 &= 0,8 \end{aligned} \quad (9.27)$$

Os valores das temperaturas fixas das extremidades, $T_0 = 40$ e $T_5 = 200$, podem ser substituídos e movidos para o lado direito. O resultado final são quatro equações com quatro incógnitas expressas na forma matricial como

$$\begin{bmatrix} 2,04 & -1 & 0 & 0 \\ -1 & 2,04 & -1 & 0 \\ 0 & -1 & 2,04 & -1 \\ 0 & 0 & -1 & 2,04 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 40,8 \\ 0,8 \\ 0,8 \\ 200,8 \end{Bmatrix} \quad (9.28)$$

Então, nossa equação diferencial original foi convertida em um sistema de equações algébricas lineares equivalentes. Consequentemente, podemos usar as técnicas descritas neste capítulo para calcular as temperaturas. Por exemplo, utilizando o MATLAB

```
>> A=[2.04 -1 0 0
     -1 2.04 -1 0
      0 -1 2.04 -1
      0 0 -1 2.04];
>> b=[40.8 0.8 0.8 200.8]';
>> T=(A\b)'
T =
    65.9698    93.7785   124.5382   159.4795
```

Um gráfico também pode ser desenvolvido comparando esses resultados com a solução analítica obtida com a Equação (9.25),

```
>> T=[40 T 200];
>> x=[0:2:10];
>> xanal=[0:10];
>> TT=@(x) 73.4523*exp(0.1*x)-53.4523* ...
    exp(-0.1*x)+20;
>> Tanal=TT(xanal);
>> plot(x,T,'o',xanal,Tanal)
```

Como ilustrado na Figura 9.8, os resultados numéricos estão muito próximos daqueles obtidos com o cálculo (solução analítica).

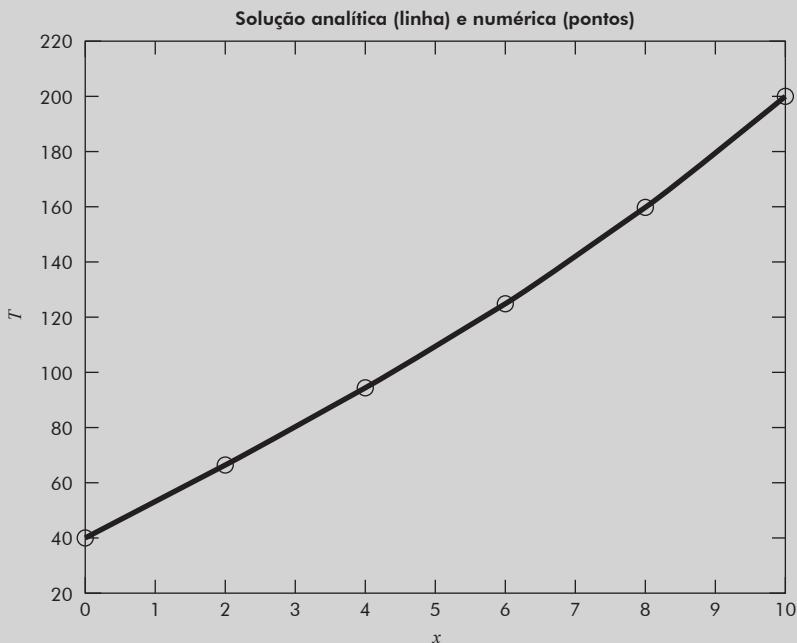


FIGURA 9.8 Um gráfico da temperatura versus a distância ao longo da haste aquecida. Ambas as soluções analítica (linha) e numérica (pontos) são apresentadas.

Além de ser um sistema linear, observe que a Equação (9.28) também é tridiagonal. Então, pode-se usar um esquema de solução eficiente como a função do MATLAB apresentada na Figura 9.6 para obter a solução:

```
>> e=[0 -1 -1 -1];
>> f=[2.04 2.04 2.04 2.04];
>> g=[-1 -1 -1 0];
>> r=[40.8 0.8 0.8 200.8];
>> Tridiag(e,f,g,r)
ans =
    65.9698    93.7785   124.5382   159.4795
```

O sistema é tridiagonal porque cada nó depende apenas dos nós adjacentes. Como eles foram enumerados sequencialmente, as equações resultantes são tridiagonais. Tais casos geralmente ocorrem na resolução de equações diferenciais baseadas em leis de conservação.

►► PROBLEMAS

9.1 Determine o número total de flops como uma função do número de equações n para o algoritmo tridiagonal (Figura 9.6).

9.2 Use o método gráfico para resolver

$$2x_1 - 6x_2 = -18$$

$$-x_1 + 8x_2 = 40$$

Verifique seus resultados substituindo-os de volta nas equações.

9.3 Dado o sistema de equações

$$0,77x_1 + x_2 = 14,25$$

$$1,2x_1 + 1,7x_2 = 20$$

(a) Resolva graficamente e verifique seus resultados substituindo-os de volta nas equações.

(b) Com base na solução gráfica, o que você espera do condicionamento do sistema?

(c) Calcule o determinante.

9.4 Dado o sistema de equações

$$2x_2 + 5x_3 = 1$$

$$2x_1 + x_2 + x_3 = 1$$

$$3x_1 + x_2 = 2$$

(a) Calcule o determinante.

(b) Use a regra de Cramer para calcular os x 's.

(c) Utilize a eliminação de Gauss com pivoteamento parcial para calcular os x 's. Como parte dos cálculos, calcule o determinante e verifique o valor obtido em (a).

(d) Substitua seus resultados de volta nas equações originais e cheque sua solução.

9.5 Dadas as equações

$$0,5x_1 - x_2 = -9,5$$

$$1,02x_1 - 2x_2 = -18,8$$

(a) Resolva graficamente.

(b) Calcule o determinante.

(c) Com base em (a) e (b), o que você espera do condicionamento do sistema?

(d) Resolva por eliminação de variáveis.

(e) Resolva novamente, mas com a_{11} levemente modificado para 0,52. Interprete seus resultados.

9.6 Dadas as equações

$$10x_1 + 2x_2 - x_3 = 27$$

$$-3x_1 - 5x_2 + 2x_3 = -61,5$$

$$x_1 + x_2 + 6x_3 = -21,5$$

(a) Resolva por eliminação de Gauss ingênua. Mostre todos os passos dos cálculos.

(b) Substitua seus resultados nas equações originais para verificar suas respostas.

9.7 Dadas as equações

$$2x_1 - 6x_2 - x_3 = -38$$

$$-3x_1 - x_2 + 7x_3 = -34$$

$$-8x_1 + x_2 - 2x_3 = -20$$

(a) Resolva por eliminação de Gauss com pivoteamento parcial. Como parte dos cálculos, use os elementos da diagonal para calcular o determinante. Mostre todos os passos dos cálculos.

(b) Substitua seus resultados nas equações originais para verificar suas respostas.

9.8 Efetue os mesmos cálculos como no Exemplo 9.5, porém para o sistema tridiagonal:

$$\begin{bmatrix} 0,8 & -0,4 & & \\ -0,4 & 0,8 & -0,4 & \\ & -0,4 & 0,8 & \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 41 \\ 25 \\ 105 \end{Bmatrix}$$

9.9 A Figura P9.9 mostra três reatores interligados por tubos. Como indicado, a taxa de transferência de produtos químicos através de cada tubo é igual a uma vazão (Q , com unidade de metros cúbicos por segundo) multiplicada pela concentração do reator do qual o fluxo se origina (c , com unidades em miligramas por metro cúbico). Se o sistema estiver em um estado estacionário, a transferência para dentro de cada reator vai balancear a transferência para fora. Deduza equações de balanço de massa para os reatores e resolva as três equações lineares algébricas simultâneas para suas concentrações.

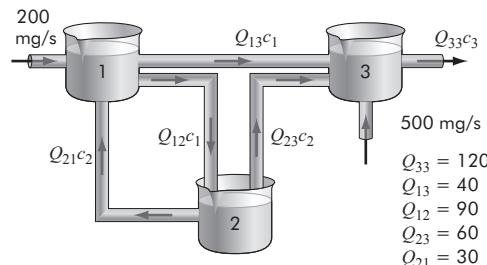


FIGURA P9.9 Três reatores interligados por tubos. A taxa de transferência de massa através de cada tubo é igual ao produto da vazão Q pela concentração c do reator do qual o fluxo se origina.

9.10 Um engenheiro civil envolvido em uma construção precisa de 4.800, 5.800 e 5.700 m³ de areia, cascalho fino e cascalho grosso, respectivamente, para um projeto de construção. Existem três minas de onde esses materiais podem ser obtidos. A composição dessas minas é

	Areia %	Cascalho fino %	Cascalho grosso %
Mina 1	52	30	18
Mina 2	20	50	30
Mina 3	25	20	55

Quantos metros cúbicos devem ser minerados de cada mina para atender as necessidades do engenheiro?

9.11 Um engenheiro eletricista supervisiona a produção de três tipos de componentes elétricos. Três tipos de material – metal, plástico e borracha – são necessários para a produção. As quantidades necessárias para produzir cada componente são:

Compo- nente	Metal (g/compo- nente)	Plástico (g/compo- nente)	Borracha (g/compo- nente)
1	15	0,25	1,0
2	17	0,33	1,2
3	19	0,42	1,6

Se um total de 2,12, 0,0434 e 0,164 kg de metal, plástico e borracha, respectivamente, estiver disponível a cada dia, quantos componentes poderão ser produzidos por dia?

9.12 Como descrito na Seção 9.4, equações algébricas lineares podem surgir na solução de equações diferenciais. Por exemplo, a equação diferencial a seguir resulta do balanço de massa em estado estacionário para o fluxo de um produto químico em um canal unidimensional:

$$0 = D \frac{d^2c}{dx^2} - U \frac{dc}{dx} - kc$$

onde c é a concentração, t é o tempo, x é a distância, D é o coeficiente de difusão, U é a velocidade do fluido e k é a taxa de decaimento de primeira ordem. Converta essa equação diferencial para um sistema equivalente de equações algébricas simultâneas. Dados $D = 2$, $U = 1$, $k = 0,2$, $c(0) = 80$ e $c(10) = 10$, resolva essas equações de $x = 0$ a 10 e faça um gráfico da concentração versus distância.

9.13 Um processo de extração em estágios é descrito na Figura P9.13. Em tais sistemas, uma corrente contendo uma fração de peso y_{entrada} de um produto químico entra a partir da esquerda a uma vazão em massa de F_1 . Simultaneamente, um solvente carregando uma fração de peso $x_{\text{entrad}}a$ do mesmo produto químico entra pela direita a uma vazão de F_2 . Assim, no estágio i , o balanço de massas pode ser representado por

$$F_1 y_{i-1} + F_2 x_{i+1} = F_1 y_i + F_2 x_i \quad (\text{P9.13a})$$

Em cada estágio, supõe-se que foi estabelecido um equilíbrio entre y_i e x_i como em

$$K = \frac{x_i}{y_i} \quad (\text{P9.13b})$$

onde K é chamado de coeficiente de distribuição. Pode-se isolar os x_i na Equação (P9.13b) e substituí-los na Equação (P9.13a) para obter

$$y_{i-1} - \left(1 + \frac{F_2}{F_1} K\right) y_i + \left(\frac{F_2}{F_1} K\right) y_{i+1} = 0 \quad (\text{P9.13c})$$

Se $F_1 = 400 \text{ kg/h}$, $y_{\text{entrada}} = 0,1$, $F_2 = 800 \text{ kg/h}$, $x_{\text{entrad}}a = 0$ e $K = 5$, determine os valores de $y_{\text{saída}}$ e $x_{\text{saída}}$ se um reator de cinco estágios for usado. Observe que a Equação (P9.13c) deve ser modificada para levar em conta as frações de peso dos fluxos de entrada quando aplicada ao primeiro e último estágios.

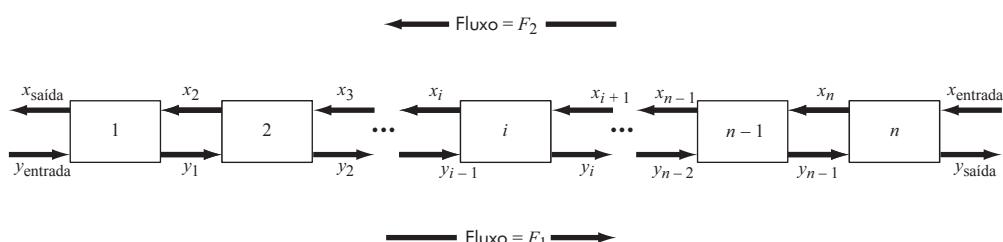


FIGURA P9.13 Processo de extração em estágios.

9.14 Uma bomba peristáltica produz um fluxo unitário (Q) de um fluido altamente viscoso. A rede está descrita na Figura P9.14. Cada seção de tubo tem os mesmos comprimentos e diâmetros. Os balanços de massa e energia mecânica podem ser simplificados para se obter os escoamentos em cada tubo. Resolva o sistema de equações a seguir para obter o fluxo de cada corrente.

$$\begin{aligned} Q_3 + 2Q_4 - 2Q_2 &= 0 & Q_1 &= Q_2 + Q_3 \\ Q_5 + 2Q_6 - 2Q_4 &= 0 & Q_3 &= Q_4 + Q_5 \\ 3Q_7 - 2Q_6 &= 0 & Q_5 &= Q_6 + Q_7 \end{aligned}$$

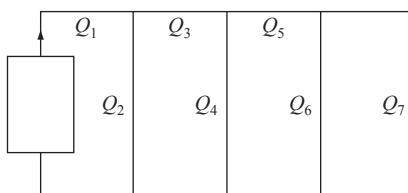


FIGURA P9.14

9.15 Uma treliça está carregada como mostra a Figura P9.15. Usando o seguinte conjunto de equações, determine as dez incógnitas: AB , BC , AD , BD , CD , DE , CE , A_x , A_y e E_y .

$$\begin{aligned} A_x + AD &= 0 \\ A_y + AB &= 0 \\ 74 + BC + (3/5)BD &= 0 \\ -AB - (4/5)BD &= 0 \\ -BC + (3/5)CE &= 0 \\ -24 - CD - (4/5)CE &= 0 \\ -AD + DE - (3/5)BD &= 0 \\ CD + (4/5)BD &= 0 \\ -DE - (3/5)CE &= 0 \\ E_y + (4/5)CE &= 0 \end{aligned}$$

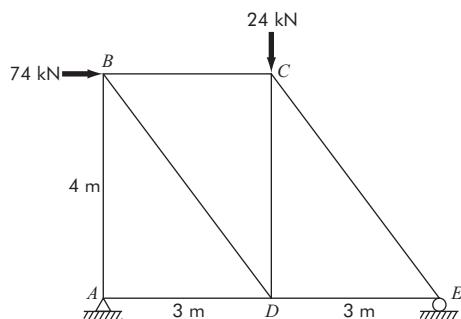


FIGURA P9.15

9.16 Um sistema *pentadiagonal* com uma largura de faixa de 5 pode ser expresso de forma genérica como

$$\left[\begin{array}{ccccccccc} f_1 & g_1 & h_1 & & & & & & \\ e_2 & f_2 & g_2 & h_2 & & & & & \\ d_3 & e_3 & f_3 & g_3 & h_3 & & & & \\ & & & \ddots & & & & & \\ & & & & \ddots & & & & \\ & & & & & d_{n-1} & e_{n-1} & f_{n-1} & g_{n-1} \\ & & & & & & d_n & e_n & f_n \end{array} \right]$$

$$\times \left\{ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{array} \right\} = \left\{ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{n-1} \\ r_n \end{array} \right\}$$

Desenvolva uma função no MATLAB para resolver tais sistemas sem pivotamento de uma maneira eficiente e similar ao algoritmo usado para as matrizes tridiagonais na Seção 9.4. Teste sua função para o seguinte caso:

$$\left[\begin{array}{ccccc} 8 & -2 & -1 & 0 & 0 \\ -2 & 9 & -4 & -1 & 0 \\ -1 & -3 & 7 & -1 & -2 \\ 0 & -4 & -2 & 12 & -5 \\ 0 & 0 & -7 & -3 & 15 \end{array} \right] \left\{ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \right\} = \left\{ \begin{array}{c} 5 \\ 2 \\ 1 \\ 1 \\ 5 \end{array} \right\}$$

9.17 Desenvolva uma função no MATLAB baseada na Figura 9.5 para implementar a eliminação de Gauss com pivotamento parcial. Modifique a função de modo que ela calcule e retorne o determinante (com o sinal correto), e detecte se o sistema é singular com base em um determinante próximo de zero. Para este último, defina “próximo de zero” como sendo quando o valor absoluto do determinante for inferior a uma dada tolerância. Quando isso ocorrer, projete a função de modo que uma mensagem de erro seja exibida e a função encerrada. A primeira linha da função deve ser:

```
function [x,D] = GaussPivotNova(A,b,tol)
```

onde D é o determinante e tol é a tolerância. Teste sua função para o Problema 9.5 com $tol = 1 \times 10^{-5}$.

10

Decomposição LU

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é familiarizar o leitor com a decomposição *LU*.¹ Após esta leitura completa, você será capaz de:

- ✓ Entender que a decomposição *LU* envolve a decomposição da matriz de coeficientes em duas matrizes triangulares que podem ser usadas de forma eficiente para avaliar diferentes vetores do lado direito.
- ✓ Aprender como expressar a eliminação de Gauss como uma decomposição *LU*.
- ✓ Aprender como avaliar múltiplos vetores do lado direito, dada uma decomposição *LU*.
- ✓ Reconhecer que o método de Cholesky fornece uma maneira eficiente para decompor uma matriz simétrica e que a matriz triangular resultante e a sua transposta podem ser usadas para avaliar vetores do lado direito de maneira eficiente.
- ✓ Entender em linhas gerais o que acontece quando o operador barra invertida do MATLAB é usado para resolver sistemas lineares.

Como descrito no Capítulo 9, a eliminação de Gauss é projetada para resolver sistemas de equações algébricas lineares:

$$[A]\{x\} = \{b\} \quad (10.1)$$

Embora certamente represente um método seguro de resolver esses sistemas, ela se torna ineficiente ao se resolver equações com os mesmos coeficientes $[A]$, mas com diferentes constantes do lado direito $\{b\}$.

Lembre-se de que a eliminação de Gauss envolve dois passos: a eliminação progressiva e a substituição regressiva (Figura 9.3). Como descrito na Seção 9.2.2, a eliminação progressiva compreende o maior volume de esforço computacional, o que é particularmente verdadeiro para grandes sistemas de equações.

Os métodos de decomposição *LU* separam a eliminação da matriz $[A]$, que consome tempo, das manipulações do lado direito $\{b\}$. Assim, uma vez que $[A]$ tenha sido “decomposta”, vetores múltiplos à direita podem ser calculados de maneira eficiente.

¹ No jargão de métodos numéricos, os termos “decomposição” e “fatoração” são sinônimos. Embora a documentação do MATLAB empregue o termo *fatoração LU* (*LU factorization*), adotou-se neste capítulo o termo *decomposição LU* conforme outra obra já traduzida do mesmo autor deste livro (*Métodos Numéricos para Engenharia*, Chapra & Canale, 5^a edição, McGraw-Hill, 2008). Observe que a expressão *fatorização LU* é frequentemente usada para descrever a mesma abordagem.

É interessante observar que a própria eliminação de Gauss pode ser escrita como uma decomposição *LU*. Antes de mostrar como isso pode ser feito, será fornecida uma visão matemática geral da estratégia da decomposição.

10.1 UMA VISÃO GERAL DA DECOMPOSIÇÃO *LU*

De modo semelhante à eliminação de Gauss, a decomposição *LU* requer pivotamento, a fim de se evitar divisões por zero. Entretanto, para simplificar a descrição a seguir, a questão do pivotamento será omitida, e, além disso, estará limitada a um conjunto de três equações simultâneas. Os resultados podem ser estendidos diretamente a sistemas de n dimensões.

A Equação (10.1) pode ser reorganizada como

$$[A]\{x\} - \{b\} = 0 \quad (10.2)$$

Considere que a Equação (10.2) possa ser escrita como um sistema triangular superior. Por exemplo, para um sistema 3×3 :

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} \quad (10.3)$$

Perceba que isso é parecido com os cálculos que ocorrem no primeiro passo da eliminação de Gauss; ou seja, a eliminação é usada para reduzir o sistema a uma forma triangular superior. A Equação (10.3) também pode ser escrita em notação matricial e reorganizada para fornecer

$$[U]\{x\} - \{d\} = 0 \quad (10.4)$$

Agora, suponha que exista uma matriz triangular inferior com 1's na diagonal,

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \quad (10.5)$$

que tem a propriedade de que, quando o primeiro membro da Equação (10.4) for multiplicado à esquerda por ela, o resultado seja o primeiro membro da Equação (10.2). Isto é,

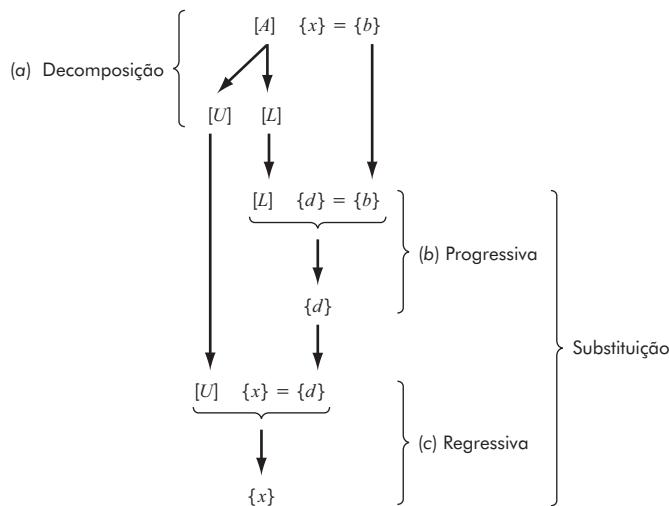
$$[L]\{[U]\{x\} - \{d\}\} = [A]\{x\} - \{b\} \quad (10.6)$$

Se essa equação for válida para todo $\{x\}$, segue das regras de multiplicação de matrizes que

$$[L][U] = [A] \quad (10.7)$$

e

$$[L]\{d\} = \{b\} \quad (10.8)$$

**FIGURA 10.1** Os passos da decomposição LU.

Uma estratégia de dois passos (ver Figura 10.1) para obter soluções pode basear-se nas Equações (10.3), (10.7) e (10.8):

1. Passo da decomposição LU. $[A]$ é fatorada ou “decomposta” em matrizes triangulares inferiores $[L]$ e superiores $[U]$.
2. Passo de substituição. $[L]$ e $[U]$ são usadas para determinar uma solução $\{x\}$ para um lado direito $\{b\}$. Esse passo consiste em duas etapas. Primeiro, a Equação (10.8) é usada para gerar um vetor intermediário $\{d\}$ por substituição progressiva. A seguir, o resultado é substituído na Equação (10.3), que pode ser resolvida por substituição regressiva, determinando-se $\{x\}$.

Agora, vejamos como a eliminação de Gauss pode ser implementada dessa forma.

10.2 A VERSÃO DA ELIMINAÇÃO DE GAUSS POR DECOMPOSIÇÃO LU

Embora à primeira vista pareça não estar relacionada com a decomposição LU, a eliminação de Gauss pode ser usada para decompor $[A]$ em $[L]$ e $[U]$. Isso pode ser visto facilmente para $[U]$, que é um produto direto da eliminação progressiva. Lembrase de que o passo de eliminação progressiva tem por objetivo reduzir a matriz original dos coeficientes $[A]$ para a forma

$$[U] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \quad (10.9)$$

que está na forma triangular superior desejada.

Ainda que possa não ser tão aparente, a matriz $[L]$ também é produzida durante essa etapa. Isso pode ser prontamente ilustrado para um sistema de três equações,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

A primeira etapa da eliminação de Gauss é multiplicar a linha 1 pelo fator [lembre-se da Equação (9.9)]

$$f_{21} = \frac{a_{21}}{a_{11}}$$

e subtrair o resultado da segunda linha para eliminar a_{21} . Analogamente, a linha 1 é multiplicada por

$$f_{31} = \frac{a_{31}}{a_{11}}$$

e o resultado é subtraído da terceira linha para eliminar a_{31} . O passo final é multiplicar a segunda linha modificada por

$$f_{32} = \frac{a'_{32}}{a'_{22}}$$

e subtrair o resultado da terceira linha para eliminar a'_{32} .

Agora, considere que foram meramente efetuadas todas essas manipulações na matriz $[A]$. Claramente, se não se quer modificar as equações, também se deve fazer o mesmo com o lado direito $\{b\}$. Mas não há nenhuma razão que obrigue a efetuar essas operações simultaneamente. Assim, pode-se salvar os f 's e manipular $\{b\}$ depois.

Onde serão armazenados os fatores f_{21} , f_{31} e f_{32} ? Lembre-se de que a ideia por trás da eliminação era criar zeros em a_{21} , a_{31} e a_{32} . Dessa forma, é possível armazenar f_{21} em a_{21} , f_{31} em a_{31} e f_{32} em a_{32} . Depois da eliminação, a matriz $[A]$ poderia, portanto, ser escrita como

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ f_{21} & a'_{22} & a'_{23} \\ f_{31} & f_{32} & a''_{33} \end{bmatrix} \quad (10.10)$$

Essa matriz, de fato, representa um armazenamento eficiente da decomposição LU de $[A]$,

$$[A] \rightarrow [L][U] \quad (10.11)$$

onde

$$[U] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \quad (10.12)$$

e

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ f_{21} & 1 & 0 \\ f_{31} & f_{32} & 1 \end{bmatrix} \quad (10.13)$$

O exemplo a seguir confirma que $[A] = [L][U]$.

EXEMPLO 10.1 Decomposição LU com eliminação de Gauss

Obtenha uma decomposição LU com base na eliminação de Gauss efetuada anteriormente no Exemplo 9.3.

Solução. No Exemplo 9.3, a eliminação de Gauss foi utilizada para resolver um conjunto de equações algébricas lineares que tinha a seguinte matriz de coeficientes:

$$[A] = \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0,1 & 7 & -0,3 \\ 0,3 & -0,2 & 10 \end{bmatrix}$$

Após a eliminação progressiva, foi obtida a seguinte matriz triangular superior:

$$[U] = \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0 & 7,00333 & -0,293333 \\ 0 & 0 & 10,0120 \end{bmatrix}$$

Os fatores empregados para obter a matriz triangular superior podem ser reunidos em uma matriz triangular inferior. Os elementos a_{21} e a_{31} foram eliminados usando-se os fatores

$$f_{21} = \frac{0,1}{3} = 0,0333333 \quad f_{31} = \frac{0,3}{3} = 0,1000000$$

e o elemento a_{32} foi eliminado usando-se o fator

$$f_{32} = \frac{-0,19}{7,00333} = -0,0271300$$

Assim, a matriz triangular inferior é

$$[L] = \begin{bmatrix} 1 & 0 & 0 \\ 0,0333333 & 1 & 0 \\ 0,100000 & -0,0271300 & 1 \end{bmatrix}$$

Consequentemente, a decomposição LU é

$$[A] = [L][U] = \begin{bmatrix} 1 & 0 & 0 \\ 0,0333333 & 1 & 0 \\ 0,100000 & -0,0271300 & 1 \end{bmatrix} \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0 & 7,00333 & -0,293333 \\ 0 & 0 & 10,0120 \end{bmatrix}$$

Esse resultado pode ser verificado efetuando-se a multiplicação de $[L][U]$ para obter

$$[L][U] = \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0,0999999 & 7 & -0,3 \\ 0,3 & -0,2 & 9,99996 \end{bmatrix}$$

cujas pequenas discrepâncias se devem a erros de arredondamento.

Depois que a matriz é decomposta, pode-se gerar uma solução para um valor particular do vetor do lado direito $\{b\}$. Isso é feito em duas etapas. Na primeira, uma substituição progressiva é executada, resolvendo-se a Equação (10.8) para $\{d\}$. É importante reconhecer que isso significa simplesmente efetuar as manipulações da eliminação em $\{b\}$. Assim, ao final desse passo, o lado direito estará no mesmo

estado em que estaria se tivesse sido efetuada a eliminação progressiva em $[A]$ e $\{b\}$ simultaneamente.

O passo de substituição progressiva pode ser representado de modo conciso por

$$d_i = b_i - \sum_{j=1}^{i-1} l_{ij} d_j \quad \text{para } i = 1, 2, \dots, n$$

A segunda etapa, então, resume-se simplesmente em implementar a substituição regressiva para resolver a Equação (10.3). Mais uma vez, é importante perceber que isso é idêntico à fase de substituição regressiva da eliminação de Gauss convencional [compare com as Equações (9.12) e (9.13)]:

$$\begin{aligned} x_n &= d_n / u_{nn} \\ x_i &= \frac{d_i - \sum_{j=i+1}^n u_{ij} x_j}{u_{ii}} \quad \text{para } i = n-1, n-2, \dots, 1 \end{aligned}$$

EXEMPLO 10.2 Os passos de substituição

Complete o problema iniciado no Exemplo 10.1 determinando a solução final por substituição progressiva e regressiva.

Solução. Como mencionado, o objetivo da substituição progressiva é impor as manipulações da eliminação, que foram aplicadas anteriormente a $[A]$, no vetor do lado direito, $\{b\}$. Lembre-se de que o sistema que está sendo resolvido é

$$\left[\begin{array}{ccc} 3 & -0,1 & -0,2 \\ 0,1 & 7 & -0,3 \\ 0,3 & -0,2 & 10 \end{array} \right] \left\{ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right\} = \left\{ \begin{array}{c} 7,85 \\ -19,3 \\ 71,4 \end{array} \right\}$$

e que a fase de eliminação progressiva da eliminação de Gauss convencional resultou em

$$\left[\begin{array}{ccc} 3 & -0,1 & -0,2 \\ 0 & 7,00333 & -0,29333 \\ 0 & 0 & 10,0120 \end{array} \right] \left\{ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right\} = \left\{ \begin{array}{c} 7,85 \\ -19,5617 \\ 70,0843 \end{array} \right\}$$

A fase de substituição progressiva é implementada aplicando-se a Equação (10.8):

$$\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0,0333333 & 1 & 0 \\ 0,100000 & -0,0271300 & 1 \end{array} \right] \left\{ \begin{array}{c} d_1 \\ d_2 \\ d_3 \end{array} \right\} = \left\{ \begin{array}{c} 7,85 \\ -19,3 \\ 71,4 \end{array} \right\}$$

ou multiplicando o lado esquerdo:

$$\begin{aligned} d_1 &= 7,85 \\ 0,0333333d_1 + d_2 &= -19,3 \\ 0,100000d_1 - 0,0271300d_2 + d_3 &= 71,4 \end{aligned}$$

Pode-se resolver a primeira equação para determinar $d_1 = 7,85$, que pode ser substituído na segunda equação para encontrar

$$d_2 = -19,3 - 0,0333333(7,85) = -19,5617$$

Ambos, d_1 e d_2 , podem então ser substituídos na terceira equação para fornecer

$$d_3 = 71,4 - 0,1(7,85) + 0,02713(-19,5617) = 70,0843$$

Assim,

$$\{d\} = \begin{Bmatrix} 7,85 \\ -19,5617 \\ 70,0843 \end{Bmatrix}$$

Esse resultado pode ser substituído na Equação (10.3), $[U]\{x\} = \{d\}$:

$$\begin{bmatrix} 3 & -0,1 & -0,2 \\ 0 & 7,00333 & -0,293333 \\ 0 & 0 & 10,0120 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7,85 \\ -19,5617 \\ 70,0843 \end{Bmatrix}$$

que pode ser resolvida por substituição regressiva (ver Exemplo 9.3 para detalhes) para determinar a solução final:

$$\{x\} = \begin{Bmatrix} 3 \\ -2,5 \\ 7,00003 \end{Bmatrix}$$

10.2.1 Decomposição LU com pivotamento

Assim como para a eliminação de Gauss padrão, o pivotamento parcial é necessário para obter uma solução confiável com a decomposição LU. Uma maneira de fazer isso envolve a utilização de uma matriz de permutação (lembre-se da Seção 8.1.2). A abordagem consiste dos seguintes passos:

1. Eliminação. A decomposição LU com pivotamento de uma matriz $[A]$ pode ser representada na forma matricial como

$$[P][A] = [L][U]$$

A matriz triangular superior, $[U]$, é gerada por eliminação com pivotamento parcial, enquanto armazenam-se os fatores de multiplicação em $[L]$ e emprega-se a matriz de permutação, $[P]$, para manter o controle das linhas pivotadas.

2. Substituição progressiva. As matrizes $[L]$ e $[P]$ são usadas para efetuar o passo de eliminação com pivotamento em $\{b\}$, de modo a gerar o vetor do lado direito intermediário, $\{d\}$. Esse passo pode ser representado de forma concisa como a solução da seguinte formulação matricial:

$$[L]\{d\} = [P]\{b\}$$

3. Substituição regressiva. A solução final é gerada da mesma forma como foi feito anteriormente para a eliminação de Gauss. Esse passo também pode ser representado de forma concisa como a solução da formulação matricial:

$$[U]\{x\} = \{d\}$$

Essa abordagem é ilustrada no exemplo a seguir.

EXEMPLO 10.3 Decomposição LU com pivotamento

Calcule a decomposição LU e encontre a solução para o mesmo sistema analisado no Exemplo 9.4.

$$\begin{bmatrix} 0,0003 & 3,0000 \\ 1,0000 & 1,0000 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 2,0001 \\ 1,0000 \end{Bmatrix}$$

Solução. Antes da eliminação, define-se a matriz de permutação inicial:

$$[P] = \begin{bmatrix} 1,0000 & 0,0000 \\ 0,0000 & 1,0000 \end{bmatrix}$$

Nota-se imediatamente que o pivotamento é necessário, de modo que, antes do passo de eliminação, trocam-se as linhas:

$$[A] = \begin{bmatrix} 1,0000 & 1,0000 \\ 0,0003 & 3,0000 \end{bmatrix}$$

Ao mesmo tempo, mantém-se o controle do pivô trocando as linhas da matriz de permutação:

$$[P] = \begin{bmatrix} 0,0000 & 1,0000 \\ 1,0000 & 0,0000 \end{bmatrix}$$

Em seguida, elimina-se a_{21} subtraindo o fator $l_{21} = a_{21}/a_{11} = 0,0003/1 = 0,0003$ da segunda linha de A . Ao fazer isso, calculamos que o novo valor de $a'_{22} = 3 - 0,0003(1) = 2,9997$. Assim, o passo de eliminação está completo com o resultado:

$$[U] = \begin{bmatrix} 1 & 1 \\ 0 & 2,9997 \end{bmatrix} \quad [L] = \begin{bmatrix} 1 & 0 \\ 0,0003 & 1 \end{bmatrix}$$

Antes de implementar a substituição progressiva, a matriz de permutação é usada para reordenar o vetor do lado direito para refletir os pivôs como em

$$[P]\{b\} = \begin{bmatrix} 0,0000 & 1,0000 \\ 1,0000 & 0,0000 \end{bmatrix} \begin{Bmatrix} 2,0001 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2,0001 \end{Bmatrix}$$

Em seguida, a substituição progressiva é aplicada como em

$$\begin{bmatrix} 1 & 0 \\ 0,0003 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 2,0001 \end{Bmatrix}$$

que pode ser resolvido para determinar $d_1 = 1$ e $d_2 = 2,0001 - 0,0003(1) = 1,9998$. Neste ponto, o sistema é

$$\begin{bmatrix} 1 & 1 \\ 0 & 2,9997 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 1,9998 \end{Bmatrix}$$

A aplicação da substituição regressiva fornece o resultado final:

$$x_2 = \frac{1,9998}{2,9997} = 0,66667$$

$$x_1 = \frac{1 - 1(0,66667)}{1} = 0,33333$$

O algoritmo de decomposição *LU* exige o mesmo número de flops que a eliminação de Gauss. A única diferença é que um pouco menos de esforço é gasto na fase de decomposição, já que as operações não são aplicadas ao lado direito. Reciprocamente, a fase de substituição gasta um pouco mais de esforço.

10.2.2 Função do MATLAB: `lu`

O MATLAB tem uma função nativa chamada `lu`, que gera a decomposição *LU*. Ela tem a seguinte sintaxe básica:

```
[L, U] = lu(X)
```

onde `L` e `U` são as matrizes triangulares inferior e superior, respectivamente, derivadas da decomposição *LU* da matriz `X`. Observe que essa função usa pivotamento parcial para evitar a divisão por zero. O exemplo a seguir mostra como ela pode ser empregada para gerar tanto a decomposição quanto uma solução para o mesmo problema que foi resolvido nos Exemplos 10.1 e 10.2.

EXEMPLO 10.4 Decomposição *LU* com o MATLAB

Use o MATLAB para calcular a decomposição *LU* e encontrar a solução para o mesmo sistema linear analisado nos Exemplos 10.1 e 10.2:

$$\begin{bmatrix} 3 & -0,1 & -0,2 \\ 0,1 & 7 & -0,3 \\ 0,3 & -0,2 & 10 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7,85 \\ -19,3 \\ 71,4 \end{Bmatrix}$$

Solução. A matriz dos coeficientes e o vetor do lado direito podem ser definidos na forma-padrão como

```
>> A = [3 -.1 -.2;.1 7 -.3;.3 -.2 10];
>> b = [7.85; -19.3; 71.4];
```

Em seguida, a decomposição *LU* pode ser calculada com

```
>> [L, U] = lu(A)

L =
    1.0000      0      0
    0.0333    1.0000      0
    0.1000   -0.0271    1.0000

U =
    3.0000   -0.1000   -0.2000
        0     7.0033   -0.2933
        0         0    10.0120
```

Esse é o mesmo resultado que foi obtido manualmente no Exemplo 10.1. Pode-se verificar que o resultado está correto calculando a matriz original como

```
>> L*U
ans =
```

3.0000	-0.1000	-0.2000
0.1000	7.0000	-0.3000
0.3000	-0.2000	10.0000

Para gerar a solução, primeiro calcula-se

```
>> d = L\b
d =
7.8500
-19.5617
70.0843
```

E, depois, utiliza-se esse resultado para calcular a solução

```
>> x = U\d
x =
3.0000
-2.5000
7.0000
```

Esses resultados estão em conformidade com aqueles obtidos manualmente no Exemplo 10.2.

10.3 DECOMPOSIÇÃO DE CHOLESKY

Lembre-se de que, no Capítulo 8, foi mostrado que uma matriz simétrica é aquela na qual $a_{ij} = a_{ji}$ para todo i e j ; ou seja, $[A] = [A]^T$. Estes sistemas ocorrem geralmente no contexto tanto de problemas de matemática quanto de engenharia, e para eles estão disponíveis técnicas de solução especiais. Elas oferecem vantagens computacionais, pois apenas metade do armazenamento é necessária, e, na maioria dos casos, apenas a metade do tempo de computação para sua solução.

Uma das abordagens mais populares envolve a *decomposição de Cholesky*, algoritmo baseado no fato de que uma matriz simétrica pode ser decomposta em

$$[A] = [U]^T[U] \quad (10.14)$$

Isto é, os fatores triangulares resultantes são transpostos um do outro.

Os termos da Equação (10.14) podem ser multiplicados e igualados. O resultado é expresso de forma simples por relações de recorrência. Para a i -ésima linha,

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2} \quad (10.15)$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}}{u_{ii}} \quad \text{para } j = i+1, \dots, n \quad (10.16)$$

EXEMPLO 10.5 Decomposição de Cholesky

Aplique a decomposição de Cholesky à matriz simétrica

$$[A] = \begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix}$$

Solução. Para a primeira linha ($i = 1$), a Equação (10.15) é empregada para calcular

$$u_{11} = \sqrt{a_{11}} = \sqrt{6} = 2,44949$$

Em seguida, a Equação (10.16) pode ser usada para determinar

$$u_{12} = \frac{a_{12}}{u_{11}} = \frac{15}{2,44949} = 6,123724$$

$$u_{13} = \frac{a_{13}}{u_{11}} = \frac{55}{2,44949} = 22,45366$$

Para a segunda linha ($i = 2$):

$$u_{22} = \sqrt{a_{22} - u_{12}^2} = \sqrt{55 - (6,123724)^2} = 4,1833$$

$$u_{23} = \frac{a_{23} - u_{12}u_{13}}{u_{22}} = \frac{225 - 6,123724(22,45366)}{4,1833} = 20,9165$$

Para a terceira linha ($i = 3$):

$$u_{33} = \sqrt{a_{33} - u_{13}^2 - u_{23}^2} = \sqrt{979 - (22,45366)^2 - (20,9165)^2} = 6,110101$$

Assim, a decomposição de Cholesky fornece

$$[U] = \begin{bmatrix} 2,44949 & 6,123724 & 22,45366 \\ & 4,1833 & 20,9165 \\ & & 6,110101 \end{bmatrix}$$

A validade dessa decomposição pode ser verificada substituindo-se a matriz L e sua transposta na Equação (10.14), para ver se seu produto fornece a matriz original $[A]$. Isso é deixado como exercício.

Depois de obter a decomposição, ela pode ser usada para determinar uma solução para um vetor do lado direito $\{b\}$ de um modo similar à decomposição LU . Primeiro, um vetor intermediário $\{d\}$ é criado pela resolução de

$$[U]^T \{d\} = \{b\} \quad (10.17)$$

Em seguida, a solução final pode ser obtida pela resolução de

$$[U]\{x\} = \{d\} \quad (10.18)$$

10.3.1 Função do MATLAB: chol

O MATLAB tem uma função nativa chamada `chol`, que gera a decomposição de Cholesky. Ela tem a seguinte sintaxe básica:

$$U = \text{chol}(X)$$

onde U é a matriz triangular superior, de modo que $U^*U = X$. O exemplo a seguir mostra como essa função pode ser empregada para gerar tanto a decomposição quanto a solução para a mesma matriz analisada no exemplo anterior.

EXEMPLO 10.6 Decomposição de Cholesky com o MATLAB

Use o MATLAB para obter a decomposição de Cholesky para a mesma matriz analisada no Exemplo 10.5.

$$[A] = \begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix}$$

Também obtenha uma solução para um vetor do lado direito, que é a soma das linhas de $[A]$. Observe que, para esse caso, a resposta será um vetor de 1's.

Solução. A matriz é inicializada na forma-padrão como

```
>> A = [6 15 55; 15 55 225; 55 225 979];
```

O vetor do lado direito, que é a soma das linhas de $[A]$, pode ser gerado como

```
>> b = [sum(A(1,:)); sum(A(2,:)); sum(A(3,:))]

b =
    76
    295
   1259
```

Em seguida, a decomposição de Cholesky pode ser calculada com

```
>> U = chol(A)

U =
    2.4495      6.1237     22.4537
        0      4.1833     20.9165
        0          0      6.1101
```

Pode-se verificar que isso está correto calculando a matriz original como

```
>> U'*U

ans =
    6.0000      15.0000     55.0000
   15.0000      55.0000    225.0000
   55.0000     225.0000    979.0000
```

Para gerar a solução, primeiro calcula-se

```
>> d = U'\b  
d =  
    31.0269  
    25.0998  
    6.1101
```

E, depois, utiliza-se esse resultado para calcular a solução final

```
>> x = U\d  
x =  
    1.0000  
    1.0000  
    1.0000
```

10.4 DIVISÃO À ESQUERDA DO MATLAB

A divisão à esquerda foi introduzida anteriormente sem nenhuma explicação de como ela funciona. Agora que temos algum conhecimento de técnicas de solução matriciais, podemos fornecer uma descrição simplificada de sua operação.

Quando se implementa a divisão à esquerda com o operador barra invertida, o MATLAB chama um algoritmo altamente sofisticado para obter a solução. Em essência, o MATLAB examina a estrutura da matriz dos coeficientes e, então, implementa um método ótimo para obter a solução. Embora os detalhes estejam fora do escopo deste livro, uma visão geral simplificada pode ser descrita.

A princípio, o MATLAB verifica se a matriz $[A]$ está em um formato em que uma solução pode ser obtida com a eliminação de Gauss completa. Isso inclui sistemas que são (a) esparsos e de banda, (b) triangulares (ou facilmente transformáveis para a forma triangular), ou (c) simétricos. Se qualquer um desses casos é detectado, a solução é obtida com as técnicas eficientes que estão disponíveis para tais sistemas. Algumas dessas técnicas incluem solucionadores de matrizes de banda, substituição progressiva e regressiva e decomposição de Cholesky.

Se nenhuma dessas soluções simplificadas é possível e a matriz é quadrada², uma decomposição triangular geral é calculada por eliminação de Gauss com pivotamento parcial e a solução é obtida com substituição.

² Deve-se notar que no caso em que $[A]$ não é quadrada, uma solução por mínimos quadrados é obtida com uma abordagem chamada *decomposição QR*.

►►PROBLEMAS

10.1 Determine o total de flops como uma função do número de equações n para as fases de (a) decomposição, (b) substituição progressiva, e (c) substituição regressiva da versão da eliminação de Gauss por decomposição LU .

10.2 Use as regras de multiplicação matricial para provar que as Equações (10.7) e (10.8) seguem da Equação (10.6).

10.3 Utilize a eliminação de Gauss ingênua para decompor o seguinte sistema de acordo com a descrição dada na Seção 10.2:

$$\begin{aligned} 7x_1 + 2x_2 - 3x_3 &= -12 \\ 2x_1 + 5x_2 - 3x_3 &= -20 \\ x_1 - x_2 - 6x_3 &= -26 \end{aligned}$$

Após isso, multiplique as matrizes $[L]$ e $[U]$ resultantes para verificar que $[A]$ é produzida.

10.4 (a) Use decomposição LU para resolver o sistema de equações do Problema 10.3. Mostre todos os passos nos cálculos. (b) Também resolva o sistema para um vetor do lado direito alternativo

$$\{b\}^T = [12 \quad 18 \quad -6]$$

10.5 Resolva o sistema de equações a seguir utilizando decomposição LU com pivotamento parcial:

$$\begin{aligned} 2x_1 - 6x_2 - x_3 &= -38 \\ -3x_1 - x_2 + 6x_3 &= -34 \\ -8x_1 + x_2 - 2x_3 &= -40 \end{aligned}$$

10.6 Desenvolva sua própria função do MATLAB para determinar a decomposição LU de uma matriz quadrada sem pivotamento parcial. Isto é, desenvolva uma função que tenha como entrada a matriz quadrada e retorne as matrizes triangulares $[L]$ e $[U]$. Teste sua função utilizando-a para resolver o sistema do Problema 10.3. Confirme que sua função está funcionando adequadamente verificando que $[L][U] = [A]$ e utilizando a função nativa `lu`.

10.7 Confirme a validade da decomposição de Cholesky do Exemplo 10.5 substituindo os resultados na Equação (10.14) para verificar que o produto de $[U]^T$ e $[U]$ produz $[A]$.

10.8 (a) Efetue manualmente a decomposição de Cholesky do seguinte sistema simétrico

$$\begin{bmatrix} 8 & 20 & 16 \\ 20 & 80 & 50 \\ 16 & 50 & 60 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 100 \\ 250 \\ 100 \end{Bmatrix}$$

(b) Verifique seu cálculo com a função do MATLAB `chol`. (c) Utilize os resultados da decompo-

sição $[U]$ para determinar a solução para o vetor do lado direito.

10.9 Desenvolva sua própria função do MATLAB para determinar a decomposição de Cholesky de uma matriz simétrica sem pivotamento. Isto é, desenvolva uma função que tenha como entrada a matriz simétrica e retorne a matriz $[U]$. Teste sua função utilizando-a para resolver o sistema do Problema 10.8 e use a função nativa `chol` para confirmar que sua função está funcionando adequadamente.

10.10 Resolva o seguinte conjunto de equações empregando decomposição LU com pivotamento:

$$\begin{aligned} 3x_1 - 2x_2 + x_3 &= -10 \\ 2x_1 + 6x_2 - 4x_3 &= 44 \\ -8x_1 - 2x_2 + 5x_3 &= -26 \end{aligned}$$

10.11 (a) Determine manualmente a decomposição LU com pivotamento para a matriz a seguir e cheque seus resultados verificando que $[L][U] = [A]$.

$$\begin{bmatrix} 8 & 5 & 1 \\ 3 & 7 & 4 \\ 2 & 3 & 9 \end{bmatrix}$$

(b) Empregue o resultado de (a) para calcular o determinante.

(c) Repita (a) e (b) utilizando o MATLAB.

10.12 Use a decomposição LU a seguir para (a) calcular o determinante e (b) resolver $[A]\{x\} = \{b\}$ com $\{b\}^T = [-10 \quad 50 \quad -26]$.

$$[A] = [L][U] = \begin{bmatrix} 1 & & \\ 0,6667 & 1 & \\ -0,3333 & -0,3636 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} 3 & -2 & 1 \\ 7,3333 & -4,6667 & \\ & 3,6364 & \end{bmatrix}$$

10.13 Use a decomposição de Cholesky para determinar $[U]$ de modo que

$$[A] = [U]^T[U] = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

10.14 Calcule a decomposição de Cholesky para

$$[A] = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 16 \end{bmatrix}$$

Os seus resultados fazem sentido em termos das Equações (10.15) e (10.16)?

Matriz inversa e condicionamento

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é mostrar como calcular a matriz inversa e ilustrar como ela pode ser usada para analisar sistemas lineares complexos que ocorrem na engenharia e na ciência. Além disso, é descrito um método para avaliar a sensibilidade de uma solução matricial a erros de arredondamento. Após esta leitura completa, você será capaz de:

- ✓ Aprender como determinar a matriz inversa de modo eficiente com base na decomposição LU .
- ✓ Entender como a matriz inversa pode ser usada para avaliar as características estímulo-resposta de sistemas de engenharia.
- ✓ Entender o significado de normas de matrizes e de vetores e como elas são calculadas.
- ✓ Aprender como usar normas para calcular o número de condicionamento da matriz.
- ✓ Entender como o valor do número de condicionamento pode ser usado para estimar a precisão das soluções de sistemas de equações algébricas lineares.

11.1 A MATRIZ INVERSA

Na discussão sobre operações com matrizes (Seção 8.1.2), foi introduzida a noção de que, se a matriz $[A]$ é quadrada, existe outra matriz $[A]^{-1}$, chamada inversa de $[A]$, para a qual

$$[A][A]^{-1} = [A]^{-1}[A] = [I] \quad (11.1)$$

Agora, o foco será em como calcular essa inversa numericamente. A seguir, será explorado como isso pode ser usado para análises de engenharia.

11.1.1 Calculando a inversa

A inversa pode ser calculada coluna a coluna, gerando soluções para vetores unitários como constantes do lado direito. Por exemplo, se o vetor do lado direito tem um 1 na primeira posição e zeros no resto,

$$\{b\} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \quad (11.2)$$

a solução resultante será a primeira coluna da matriz inversa. Analogamente, se um vetor unitário com um 1 na segunda linha é usado

$$\{b\} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \quad (11.3)$$

o resultado será a segunda coluna da matriz inversa.

A melhor forma de implementar tais cálculos é com uso do algoritmo de decomposição *LU*. Lembre-se de que uma das grandes vantagens da decomposição *LU* é que ela fornece uma forma eficaz de calcular diversos vetores do lado direito. Assim, é ideal para manipular os vários vetores unitários necessários para calcular a inversa.

EXEMPLO 11.1 Inversão de matrizes

Utilize a decomposição *LU* para determinar a inversa da matriz para o sistema do Exemplo 10.1:

$$[A] = \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0,1 & 7 & -0,3 \\ 0,3 & -0,2 & 10 \end{bmatrix}$$

Lembre-se de que a decomposição resultou nas matrizes triangulares superior e inferior a seguir:

$$[U] = \begin{bmatrix} 3 & -0,1 & -0,2 \\ 0 & 7,00333 & -0,293333 \\ 0 & 0 & 10,0120 \end{bmatrix} \quad [L] = \begin{bmatrix} 1 & 0 & 0 \\ 0,0333333 & 1 & 0 \\ 0,100000 & -0,0271300 & 1 \end{bmatrix}$$

Solução. A primeira coluna da matriz inversa pode ser determinada efetuando-se o procedimento de solução da substituição progressiva com um vetor unitário (como o 1 na primeira linha) como vetor do lado direito. Dessa forma, o sistema triangular inferior pode ser definido como [lembre-se da Equação (10.8)]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0,0333333 & 1 & 0 \\ 0,100000 & -0,0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

e resolvido com substituição progressiva para determinar $\{d\}^T = [1 \ -0,03333 \ -0,1009]$. Esse vetor pode ser usado como o lado direito do sistema triangular superior [lembre-se da Equação (10.3)]:

$$\begin{bmatrix} 3 & -0,1 & -0,2 \\ 0 & 7,00333 & -0,293333 \\ 0 & 0 & 10,0120 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 1 \\ -0,03333 \\ -0,1009 \end{Bmatrix}$$

que pode ser resolvido por substituição regressiva para determinar $\{x\}^T = [0,33249 \ -0,00518 \ -0,01008]$, que é a primeira coluna da matriz inversa:

$$[A]^{-1} = \begin{bmatrix} 0,33249 & 0 & 0 \\ -0,00518 & 0 & 0 \\ -0,01008 & 0 & 0 \end{bmatrix}$$

Para determinar a segunda coluna, a Equação (10.8) é formulada como

$$\begin{bmatrix} 1 & 0 & 0 \\ 0,0333333 & 1 & 0 \\ 0,100000 & -0,0271300 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}$$

Isso pode ser usado para determinar $\{d\}$, e os resultados podem ser utilizados na Equação (10.3) para determinar $\{x\}^T = [0,004944 \quad 0,142903 \quad 0,00271]$, que é a segunda coluna da matriz inversa:

$$[A]^{-1} = \begin{bmatrix} 0,33249 & 0,004944 & 0 \\ -0,00518 & 0,142903 & 0 \\ -0,01008 & 0,002710 & 0 \end{bmatrix}$$

Por fim, o procedimento de substituição progressiva e regressiva pode ser implementado com $\{b\}^T = [0 \quad 0 \quad 1]$ para determinar $\{x\}^T = [0,006798 \quad 0,004183 \quad 0,09988]$, que é a última coluna da matriz inversa:

$$[A]^{-1} = \begin{bmatrix} 0,33249 & 0,004944 & 0,006798 \\ -0,00518 & 0,142903 & 0,004183 \\ -0,01008 & 0,002710 & 0,099880 \end{bmatrix}$$

A validade desse resultado é verificada por $[A][A]^{-1} = [I]$.

11.1.2 Cálculos de respostas a estímulos

Como discutido na Seção 3.1 da Parte 3, muitos dos sistemas de equações lineares provenientes da engenharia e da ciência são derivados das leis de conservação. A expressão matemática dessas leis é alguma forma de equação de balanço para garantir que uma propriedade particular – massa, força, calor, momento, carga elétrica – seja conservada. Para um balanço de forças em uma estrutura, as propriedades poderiam ser as componentes horizontais ou verticais das forças agindo sobre cada nó da estrutura. Para um balanço de massas, as propriedades poderiam ser a massa em cada reator do processo químico. Outras áreas da engenharia e da ciência vão dar origem a exemplos semelhantes.

Uma única equação de balanço pode ser escrita para cada parte do sistema, resultando em um conjunto de equações que define o comportamento da propriedade para o sistema inteiro. Essas equações são inter-relacionadas, ou acopladas, no sentido de que cada equação pode incluir uma ou mais das variáveis das outras equações. Em muitos casos, esses sistemas são lineares e, portanto, da forma exata tratada neste capítulo:

$$[A]\{x\} = \{b\} \quad (11.4)$$

Agora, para equações de balanço, os termos da Equação (11.4) têm uma interpretação física definida. Por exemplo, os elementos de $\{x\}$ são os níveis da propriedade sendo balanceada para cada parte do sistema. Em um balanço de forças em uma estrutura, representam as forças horizontais e verticais em cada membro. Para o balanço de massas, são as massas de cada composto químico em cada reator. Em qualquer dos casos, eles representam o *estado ou a resposta* do sistema, que estamos tentando determinar.

O vetor $\{b\}$ do lado direito contém aqueles elementos do balanço que são independentes do comportamento do sistema – isto é, que são constantes. Em muitos problemas, eles representam as *funções forçantes* ou os *estímulos externos* que conduzem o sistema.

Por fim, a matriz $[A]$ dos coeficientes normalmente contém os *parâmetros* que expressam como as partes do sistema *interagem* ou são acopladas. Consequentemente, a Equação (11.4) pode ser expressa como

$$\{\text{Interações}\}\{\text{respostas}\} = \{\text{estímulos}\}$$

Como visto nos capítulos anteriores, há diversas formas de resolver a Equação (11.4). No entanto, usar a matriz inversa leva a um resultado particularmente interessante: a solução forma pode ser escrita como

$$\{x\} = [A]^{-1} \{b\}$$

ou (lembrando a definição de multiplicação de matrizes da Seção 8.1.2)

$$\begin{aligned} x_1 &= a_{11}^{-1} b_1 + a_{12}^{-1} b_2 + a_{13}^{-1} b_3 \\ x_2 &= a_{21}^{-1} b_1 + a_{22}^{-1} b_2 + a_{23}^{-1} b_3 \\ x_3 &= a_{31}^{-1} b_1 + a_{32}^{-1} b_2 + a_{33}^{-1} b_3 \end{aligned}$$

Portanto, vê-se que a própria matriz inversa, além de fornecer uma solução, tem propriedades extremamente úteis, isto é, cada um dos seus elementos representa a resposta de uma parte do sistema a um estímulo unitário de qualquer outra parte do sistema.

Observe que essas formulações são lineares e, portanto, valem a superposição e a proporcionalidade. A *superposição* significa que, se um sistema está sujeito a diversos estímulos diferentes (os b 's), as respostas podem ser calculadas individualmente, e os resultados podem ser somados para se obter uma resposta total. A *proporcionalidade* significa que a multiplicação do estímulo por uma quantidade faz a resposta a esse estímulo ser multiplicada por essa mesma quantidade. Logo, o coeficiente a_{11}^{-1} é uma constante de proporcionalidade que dá o valor de x_1 devido a um nível unitário de b_1 . Esse resultado é independente dos efeitos de b_2 e b_3 em x_1 , os quais são refletidos nos coeficientes a_{12}^{-1} e a_{13}^{-1} , respectivamente. Portanto, chega-se à conclusão geral de que o elemento a_{ij}^{-1} da matriz inversa representa o valor de x_i causado por uma quantidade unitária de b_j .

Usando o exemplo da estrutura, o elemento a_{ij}^{-1} da matriz inversa representa a força no membro i decorrente de uma força externa unitária no nó j . Mesmo para sistemas pequenos, tal comportamento das interações estímulo-resposta individuais não seria intuitivamente óbvio. Assim, a matriz inversa fornece uma técnica poderosa para entender as inter-relações de partes componentes de sistemas complicados.

EXEMPLO 11.2 Analisando o problema do saltador de bungee jumping

No início do Capítulo 8, definimos um problema envolvendo três indivíduos suspensos verticalmente e conectados por cordas. Foi derivado um sistema de equações algébricas lineares baseadas nos balanços de forças para cada saltador

$$\begin{bmatrix} 150 & -100 & 0 \\ -100 & 150 & -50 \\ 0 & -50 & 50 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 588,6 \\ 686,7 \\ 784,8 \end{Bmatrix}$$

No Exemplo 8.2, utilizou-se o MATLAB para resolver esse sistema e determinar as posições verticais dos saltadores (os x 's). Neste exemplo, use o MATLAB para calcular a matriz inversa e interprete o que ela significa.

Solução. Inicialize o MATLAB e entre com a matriz de coeficientes:

```
>> K = [150 -100 0; -100 150 -50; 0 -50 50];
```

A inversa pode ser calculada como

```
>> KI = inv(K)
```

```
KI =
0.0200 0.0200 0.0200
0.0200 0.0300 0.0300
0.0200 0.0300 0.0500
```

Cada elemento da matriz inversa, k_{ij}^{-1} , representa a variação vertical na posição (em metros) do saltador i devido à variação unitária na força (em Newtons) aplicada ao saltador j .

Observe que os números na primeira coluna ($j = 1$) indicam que a posição de todos os três saltadores aumentaria em 0,02 m se a força sobre o primeiro saltador fosse aumentada em 1 N. Isso faz sentido, pois a força adicional alongaria apenas a primeira corda desse valor (0,02 m).

Já os números na segunda coluna ($j = 2$) indicam que a aplicação de uma força de 1 N ao segundo saltador moveria o primeiro saltador 0,02 m para baixo, enquanto o segundo e o terceiro moveriam 0,03 m. A elongação de 0,02 m do primeiro saltador faz sentido, porque a primeira corda é submetida a uma força extra de 1 N independentemente de a força ser aplicada ao primeiro ou segundo saltador. Entretanto, para o segundo saltador a elongação é agora 0,03 m, porque, junto da primeira corda, a segunda corda também alonga devido à força adicional. E, obviamente, o terceiro saltador apresenta a mesma translação que o segundo saltador, já que não há força adicional sobre a terceira corda que os conecta.

Como esperado, a terceira coluna ($j = 3$) indica que a aplicação de uma força de 1 N ao terceiro saltador resulta, para o primeiro e segundo saltadores, na mesma variação da posição vertical que ocorreu quando a força foi aplicada ao segundo saltador. Porém, devido à elongação adicional da terceira corda, o terceiro saltador é movido um pouco mais para baixo.

Os princípios da superposição e proporcionalidade podem ser demonstrados utilizando a inversa para determinar o quanto longe o terceiro saltador se moveria para baixo se forças adicionais de 10, 50 e 20 N fossem aplicadas ao primeiro, segundo e terceiro saltadores, respectivamente. Isso pode ser feito simplesmente pela utilização dos elementos apropriados da terceira linha da inversa para calcular,

$$\Delta x_3 = k_{31}^{-1} \Delta F_1 + k_{32}^{-1} \Delta F_2 + k_{33}^{-1} \Delta F_3 = 0,02(10) + 0,03(50) + 0,05(20) = 2,7 \text{ m}$$

11.2 ANÁLISE DE ERRO E CONDICIONAMENTO DO SISTEMA

Além das aplicações científicas e de engenharia, a inversa também fornece formas de discernir se um sistema é mal condicionado. Três métodos diretos podem ser concebidos para essa finalidade:

1. Faça uma mudança de escala na matriz de coeficientes $[A]$ de modo que o maior elemento em cada linha seja 1. Inverta a matriz na nova escala e, se existirem elementos de $[A]^{-1}$ que sejam várias ordens de grandeza maiores que 1, é provável que o sistema seja mal condicionado.

2. Multiplique a inversa pela matriz de coeficientes original e verifique se o resultado está próximo da matriz identidade. Se não, isso indica mau condicionamento.
3. Inverta a matriz invertida e verifique se o resultado está suficientemente próximo da matriz de coeficientes original. Caso contrário, novamente isso pode indicar que o sistema é mal condicionado.

Apesar de esses métodos poderem indicar mau condicionamento, seria preferível obter um único número que servisse como um indicador do problema. As tentativas de formular tal número de condicionamento da matriz são baseadas no conceito matemático de norma.

11.2.1 Normas de vetores e de matrizes

Uma *norma* é uma função a valores reais que fornece uma medida do tamanho ou “comprimento” de entidades matemáticas com muitas componentes, como vetores e matrizes.

Um exemplo simples é um vetor no espaço euclidiano tridimensional (Figura 11.1) que pode ser representado como

$$[F] = [a \ b \ c]$$

onde a , b e c são as distâncias ao longo dos eixos x , y e z , respectivamente. O comprimento desse vetor – isto é, a distância das coordenadas $(0, 0, 0)$ até (a, b, c) – pode ser simplesmente calculado como

$$\|F\|_e = \sqrt{a^2 + b^2 + c^2}$$

onde a nomenclatura $\|F\|_e$ indica que esse comprimento se chamada *norma euclidiana* de $[F]$.

De forma semelhante, para um vetor de dimensão n , $[X] = [x_1 \ x_2 \ \dots \ x_n]$, uma norma euclidiana pode ser calculada como

$$\|X\|_e = \sqrt{\sum_{i=1}^n x_i^2}$$

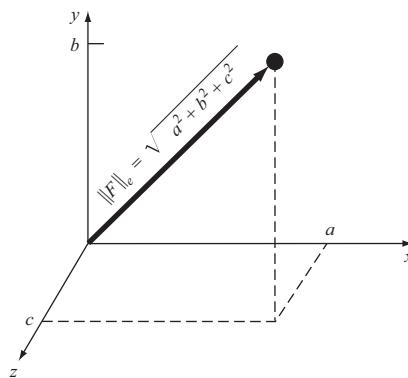


FIGURA 11.1 Representação gráfica de um vetor no espaço euclidiano.

O conceito pode ser estendido para uma matriz $[A]$, como em

$$\|A\|_f = \sqrt{\sum_{i=1}^n \sum_{j=1}^n a_{i,j}^2} \quad (11.5)$$

que recebe um nome especial – *norma de Frobenius*. Assim como as outras normas de vetores, ela fornece um único valor para quantificar o “tamanho” de $[A]$.

Deve-se observar que há alternativas às normas euclidiana e de Frobenius. Para vetores, há alternativas chamadas de normas- p que podem ser representadas genericamente por

$$\|X\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

Pode-se perceber que a norma euclidiana e a norma-2 ($p = 2$), $\|X\|_2$, são idênticas para vetores.

Outros exemplos importantes são ($p = 1$)

$$\|X\|_1 = \sum_{i=1}^n |x_i|$$

que representa a norma como a soma dos valores absolutos dos elementos. Outra é o módulo máximo ou norma uniforme do vetor ($p = \infty$),

$$\|X\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

que define a norma como o elemento com o maior valor absoluto.

Usando uma abordagem semelhante, pode-se desenvolver normas para as matrizes. Por exemplo,

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$$

Ou seja, a soma dos valores absolutos dos coeficientes é calculada para cada coluna e a maior dessas somas é tomada como a norma. Isso é chamado de *norma da soma das colunas*.

Uma construção semelhante pode ser feita para as linhas, resultando em uma *norma da soma das linhas* ou *norma uniforme da matriz*:

$$\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

Deve-se observar que, em contraste com os vetores, a norma-2 e a norma de Frobenius de uma matriz não são iguais. Enquanto a norma de Frobenius $\|A\|_f$ pode ser facilmente determinada pela Equação (11.5), a norma-2 de uma matriz $\|A\|_2$ é calculada como

$$\|A\|_2 = (\mu_{\max})^{1/2}$$

onde μ_{\max} é o maior autovalor de $[A]^T[A]$. No Capítulo 13 iremos aprender mais sobre autovalores. No momento, o ponto importante é que $\|A\|_2$, ou *norma espectral*, é a norma mínima e, portanto, fornece a medida mais justa do tamanho (Ortega, 1972).

11.2.2 Número de condicionamento da matriz

Depois de apresentado o conceito de norma, podemos usá-lo para definir

$$\text{Cond}[A] = \|A\| \cdot \|A^{-1}\|$$

onde $\text{Cond}[A]$ é chamado *número de condicionamento da matriz*. Observe que, para uma matriz $[A]$, esse número será maior ou igual a 1. Pode-se mostrar (Ralston e Rabinowitz, 1978; Gerald e Wheatley, 1989) que

$$\frac{\|\Delta X\|}{\|X\|} \leq \text{Cond}[A] \frac{\|\Delta A\|}{\|A\|}$$

Isto é, o erro relativo da norma da solução calculada pode ser tão grande quanto o erro relativo da norma dos coeficientes de $[A]$ multiplicado pelo número de condicionamento. Por exemplo, se os coeficientes da matriz $[A]$ são conhecidos com uma precisão de t dígitos (ou seja, os erros de arredondamento são da ordem de 10^{-t}) e $\text{Cond}[A] = 10^c$, a solução $[X]$ pode ser válida para apenas $t - c$ dígitos (erros de arredondamento $\approx 10^{c-t}$).

EXEMPLO 11.3 Determinação do condicionamento de uma matriz

A matriz de Hilbert, que é notoriamente mal condicionada, pode ser representada genericamente como

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{bmatrix}$$

Use a norma da soma das linhas para fazer uma estimativa do número de condicionamento da matriz de Hilbert 3×3 :

$$[A] = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$$

Solução. Pode-se normalizar a matriz de modo que o elemento máximo em cada linha seja 1:

$$[A] = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 1 & \frac{2}{3} & \frac{1}{2} \\ 1 & \frac{3}{4} & \frac{3}{5} \end{bmatrix}$$

Somando cada uma dessas linhas, temos 1,833, 2,1667 e 2,35. Assim, a terceira linha tem a maior soma, e a norma da soma das linhas é

$$\|A\|_{\infty} = 1 + \frac{3}{4} + \frac{3}{5} = 2,35$$

A inversa da matriz normalizada pode ser calculada por

$$[A]^{-1} = \begin{bmatrix} 9 & -18 & 10 \\ -36 & 96 & -60 \\ 30 & -90 & 60 \end{bmatrix}$$

Observe que os elementos dessa matriz são maiores do que os elementos da matriz original. Isso também se reflete na sua norma da soma das linhas, que é calculada como

$$\|A^{-1}\|_\infty = |-36| + |96| + |-60| = 192$$

Logo, o número de condicionamento pode ser calculado por

$$\text{Cond}[A] = 2,35(192) = 451,2$$

O fato de o número de condicionamento ser consideravelmente maior do que a unidade sugere que o sistema é mal condicionado. A extensão do mau condicionamento pode ser quantificada calculando-se $c = \log 451,2 = 2,65$. Assim, os três últimos dígitos significativos da solução podem exibir erros de arredondamento. Observe que tais estimativas quase sempre superestimam o erro real; no entanto, são úteis ao alertarem para a possibilidade de que os erros de arredondamento possam ser significativos.

11.2.3 Normas e número de condicionamento no MATLAB

O MATLAB tem funções nativas para calcular tanto a norma quanto o número de condicionamento:

```
>> norm(X, p)
e
>> cond(X, p)
```

onde X é o vetor ou a matriz e p designa o tipo de norma ou número de condicionamento (1 , 2 , inf ou ' fro '). Observe que a função `cond` é equivalente a

```
>> norm(X, p) * norm(inv(X), p)
```

Também, observe que se p é omitido, ele é definido automaticamente como 2 (padrão).

EXEMPLO 11.4 Determinação do condicionamento de uma matriz com o MATLAB

Use o MATLAB para calcular tanto a norma quanto o número de condicionamento para a matriz de Hilbert 3×3 analisada no Exemplo 11.3:

$$[A] = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 1 & \frac{2}{3} & \frac{1}{2} \\ 1 & \frac{3}{4} & \frac{3}{5} \end{bmatrix}$$

- (a) Como no Exemplo 11.3, a princípio calcule as versões da soma das linhas ($p = \text{inf}$).
- (b) Calcule também os números de condicionamento baseados nas normas de Frobenius ($p = \text{'fro'}$) e espectral ($p = 2$).

Solução. (a) Primeiro, entre com a matriz:

```
>> A = [ 1 1/2 1/3; 1 2/3 1/2; 1 3/4 3/5];
```

Então, a norma da soma das linhas e o número de condicionamento podem ser calculados como

```
>> norm(A, inf)
ans =
    2.3500
>> cond(A, inf)
ans =
    451.2000
```

Esses resultados correspondem àqueles que foram calculados manualmente no Exemplo 11.3.

(b) Os números de condicionamento baseados nas normas de Frobenius e espectral são

```
>> cond(A, 'fro')
ans =
    368.0866
>> cond(A)
ans =
    366.3503
```

► 11.3 ESTUDO DE CASO POLUIÇÃO DO AR DE INTERIORES

Contextualização. Como o nome sugere, a poluição do ar de interiores refere-se à contaminação do ar em ambientes fechados, como casas, escritórios, áreas de serviço, etc. Considere que você esteja projetando um sistema de ventilação para um restaurante localizado ao lado de uma rodovia.

Como mostra a Figura 11.2, as áreas de alimentação do restaurante consistem em duas salas, uma para fumantes e outra para crianças, e uma sala alongada. As salas 1 e 3 têm fontes de monóxido de carbono da área dos fumantes e de uma grelha defeituosa, respectivamente. Além disso, as salas 1 e 2 recebem monóxido de carbono pelas entradas de ar que, infelizmente, estão posicionadas ao lado da rodovia.

Escreva o balanço de massa no estado estacionário para cada área e resolva as equações algébricas lineares resultantes para determinar a concentração de monóxido de carbono em cada área. E, também, calcule a matriz inversa e use-a para analisar como as várias fontes afetam a área das crianças. Por exemplo, determine qual porcentagem de monóxido de carbono na área das crianças é decorrente (1) da área dos fumantes, (2) da grelha e (3) da entrada de ar de ventilação. Calcule ainda a melhoria na concentração da área das crianças se a carga de monóxido de carbono for reduzida eliminando-se a área de fumantes e concertando-se a grelha. Por fim, analise como a concentração na área das crianças varia se uma tela for construída de modo que a mistura entre as áreas 2 e 4 seja reduzida para $5 \text{ m}^3/\text{h}$.

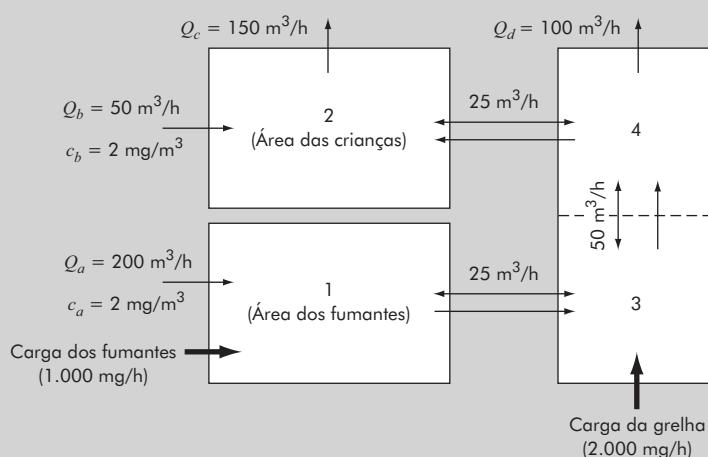


FIGURA 11.2 Visão superior dos ambientes de um restaurante. As flechas em um único sentido representam o fluxo volumétrico de ar, enquanto as flechas em duplo sentido representam a mistura por difusão. As cargas dos fumantes e da grelha adicionam monóxido de carbono ao sistema, mas fluxo de ar desprezível.

Solução. O balanço de massa pode ser escrito para cada área. Por exemplo, para a área dos fumantes (sala 1), o balanço pode ser escrito como

$$0 = W_{\text{fumantes}} + Q_a c_a - Q_a c_1 + E_{13}(c_3 - c_1)$$

(Carga) + (Entrada) - (Saída) + (Mistura)

Balanços similares podem ser escritos para as outras salas:

$$0 = Q_b c_b + (Q_a - Q_d) c_4 - Q_c c_2 + E_{24}(c_4 - c_2)$$

$$0 = W_{\text{grelha}} + Q_a c_1 + E_{13}(c_1 - c_3) + E_{34}(c_4 - c_3) - Q_a c_3$$

$$0 = Q_a c_3 + E_{34}(c_3 - c_4) + E_{24}(c_2 - c_4) - Q_a c_4$$

A substituição dos valores dos parâmetros fornece o sistema final de equações:

$$\begin{bmatrix} 225 & 0 & -25 & 0 \\ 0 & 175 & 0 & -125 \\ -225 & 0 & 275 & -50 \\ 0 & -25 & -250 & 275 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 1400 \\ 100 \\ 2000 \\ 0 \end{Bmatrix}$$

O MATLAB pode ser usado para gerar a solução. A princípio, podemos calcular a inversa. Observe que o formato “short g” foi usado para obter cinco dígitos significativos de precisão:

```
>> format short g
>> A=[225 0 -25 0
0 175 0 -125
-225 0 275 -50
0 -25 -250 275];
>> AI=inv(A)
```

```

AI =
0.0049962    1.5326e-005    0.00055172    0.00010728
0.0034483    0.0062069    0.0034483    0.0034483
0.0049655    0.00013793    0.0049655    0.00096552
0.0048276    0.00068966    0.0048276    0.0048276

```

A solução pode, então, ser gerada como

```

>> b=[1400 100 2000 0]';
>> c=AI*b
c =
8.0996
12.345
16.897
16.483

```

Assim, obtemos o resultado surpreendente que a área de fumantes tem os menores níveis de monóxido de carbono. As maiores concentrações ocorrem nas salas 3 e 4, sendo que a área 2 tem um nível intermediário. Esses resultados ocorrem porque (a) o monóxido de carbono é conservativo e (b) as únicas saídas de ar estão para fora das salas 2 e 4 (Q_c e Q_d). A sala 3 tem o pior resultado porque não apenas recebe a carga da grelha defeituosa, mas também recebe o efluente da sala 1.

Embora a análise anterior seja interessante, o real poder de sistemas lineares vem da utilização dos elementos da matriz inversa para compreender como as partes do sistema interagem. Por exemplo, os elementos da matriz inversa podem ser usados para determinar a porcentagem de monóxido de carbono na área das crianças devido a cada fonte:

Os fumantes:

$$c_{2,\text{fumantes}} = a_{21}^{-1} W_{\text{fumantes}} = 0,0034483(1.000) = 3,4483$$

$$\%_{\text{fumantes}} = \frac{3,4483}{12,345} \times 100\% = 27,93\%$$

A grelha:

$$c_{2,\text{grelha}} = a_{23}^{-1} W_{\text{grelha}} = 0,0034483(2.000) = 6,897$$

$$\%_{\text{grelha}} = \frac{6,897}{12,345} \times 100\% = 55,87\%$$

As entradas de ar:

$$c_{2,\text{entradas}} = a_{21}^{-1} Q_a c_a + a_{22}^{-1} Q_b c_b = 0,0034483(200)2 + 0,0062069(50)2$$

$$= 1,37931 + 0,62069 = 2$$

$$\%_{\text{entradas}} = \frac{2}{12,345} \times 100\% = 16,20\%$$

A grelha defeituosa é claramente a fonte mais significativa.

A inversa também pode ser empregada para determinar o impacto das soluções propostas, tais como a eliminação da área de fumantes e o concerto da grelha defeituosa.

Como o modelo é linear, o princípio da superposição é válido e os resultados podem ser determinados individualmente e somados:

$$\begin{aligned}\Delta c_2 &= a_{21}^{-1} \Delta W_{\text{fumantes}} + a_{23}^{-1} \Delta W_{\text{grelha}} = 0,0034483(-1000) + 0,0034483(-2000) \\ &= -3,4483 - 6,8966 = -10,345\end{aligned}$$

Observe que o mesmo cálculo poderia ser feito no MATLAB como em

```
>> AI(2,1)*(-1000)+AI(2,3)*(-2000)
ans =
-10.345
```

A implementação de ambas as soluções propostas reduziria a concentração em $10,345 \text{ mg/m}^3$. O resultado levaria a concentração da área das crianças para $12,345 - 10,345 = 2 \text{ mg/m}^3$. Isso faz sentido, pois na ausência das cargas dos fumantes e da grelha, as únicas fontes são as entradas de ar que valem 2 mg/m^3 .

Como todos os cálculos anteriores envolvem a variação das funções forçantes, não foi necessário recalcular a solução. No entanto, se a mistura entre a área das crianças e a sala 4 é diminuída, a matriz é alterada

$$\begin{bmatrix} 225 & 0 & -25 & 0 \\ 0 & 155 & 0 & -105 \\ -225 & 0 & 275 & -50 \\ 0 & -5 & -250 & 255 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 1400 \\ 100 \\ 2000 \\ 0 \end{Bmatrix}$$

Os resultados para esse caso envolvem uma nova solução. Utilizando o MATLAB, o resultado é

$$\begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 8,1084 \\ 12,0800 \\ 16,9760 \\ 16,8800 \end{Bmatrix}$$

Portanto, a estratégia proposta (redução da mistura entre as salas 2 e 4) reduziria a concentração da área das crianças em apenas $0,265 \text{ mg/m}^3$.

►► PROBLEMAS

11.1 Determine a matriz inversa para o seguinte sistema:

$$\begin{aligned}10x_1 + 2x_2 - x_3 &= 27 \\ -3x_1 - 6x_2 + 2x_3 &= -61,5 \\ x_1 + x_2 + 5x_3 &= -21,5\end{aligned}$$

Cheque seu resultado verificando que $[A][A]^{-1} = [I]$.
Não use uma estratégia de pivotamento.

11.2 Determine a matriz inversa para o seguinte sistema:

$$\begin{aligned}-8x_1 + x_2 - 2x_3 &= -20 \\ 2x_1 - 6x_2 - x_3 &= -38 \\ -3x_1 - x_2 + 7x_3 &= -34\end{aligned}$$

11.3 O seguinte sistema de equações é projetado para determinar as concentrações (os c 's em g/m^3)

em uma série de reatores acoplados como função da quantidade de entrada de massa em cada reator (o lado direito está em g/dia),

$$\begin{aligned} 15c_1 - 3c_2 - c_3 &= 4.000 \\ -3c_1 + 18c_2 - 6c_3 &= 1.500 \\ -4c_1 - c_2 + 12c_3 &= 2.400 \end{aligned}$$

- (a) Determine a matriz inversa.
- (b) Use a inversa para determinar a solução.
- (c) Determine quanto o fluxo de entrada de massa no reator 3 deve ser aumentado para induzir um aumento de 10 g/m³ na concentração do reator 1.
- (d) Qual será a redução da concentração no reator 3 se a taxa de entrada de massa nos reatores 1 e 2 for reduzida para 500 e 250 g/dia, respectivamente?

11.4 Determine a matriz inversa para o sistema descrito no Problema 8.9. Use a matriz inversa para determinar a concentração no reator 5 se as concentrações de entrada são alteradas para $c_{01} = 10$ e $c_{03} = 20$.

11.5 Determine a matriz inversa para o sistema descrito no Problema 8.10. Use a matriz inversa para determinar a força nos três membros (F_1 , F_2 e F_3) se a carga vertical no nó 1 é dobrada para $F_{1,v} = -4.000$ N e a carga horizontal de $F_{3,h} = -1.000$ N é aplicada ao nó 3.

11.6 Determine $\|A\|_F$, $\|A\|_1$ e $\|A\|_\infty$ para

$$[A] = \begin{bmatrix} 8 & 2 & -10 \\ -9 & 1 & 3 \\ 15 & -1 & 6 \end{bmatrix}$$

Antes de determinar as normas, normalize a matriz fazendo o maior elemento de cada linha igual a um.

11.7 Determine as normas de Frobenius e da soma das linhas para os sistemas dos Problemas 11.2 e 11.3.

11.8 Use o MATLAB para determinar o número de condicionamento espectral para o sistema a seguir. Não normalize o sistema.

$$\begin{bmatrix} 1 & 4 & 9 & 16 & 25 \\ 4 & 9 & 16 & 25 & 36 \\ 9 & 16 & 25 & 36 & 49 \\ 16 & 25 & 36 & 49 & 64 \\ 25 & 36 & 49 & 64 & 81 \end{bmatrix}$$

Calcule o número de condicionamento com base na norma da soma das linhas.

11.9 Além da matriz de Hilbert, há outras matrizes que são inherentemente mal condicionadas. Uma delas é a *matriz de Vandermonde*, que tem a seguinte forma:

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix}$$

- (a) Determine o número de condicionamento com base na norma da soma das linhas para o caso em que $x_1 = 4$, $x_2 = 2$ e $x_3 = 7$.
- (b) Use o MATLAB para calcular os números de condicionamento espectral e de Frobenius.

11.10 Use o MATLAB para determinar o número de condicionamento espectral para uma matriz de Hilbert de dimensão 10×10 . Quantos dígitos de precisão espera-se que sejam perdidos devido ao mau condicionamento? Determine a solução para esse sistema para o caso em que cada elemento do vetor do lado direito $\{b\}$ consista na soma dos coeficientes em sua linha; ou seja, resolva para o caso em que todas as incógnitas devem ser exatamente iguais a um. Compare os erros resultantes com aqueles esperados com base no número de condicionamento.

11.11 Repita o Problema 11.10, mas para o caso de uma matriz de Vandermonde de dimensão 6×6 (ver Problema 11.9) em que $x_1 = 4$, $x_2 = 2$, $x_3 = 7$, $x_4 = 10$, $x_5 = 3$ e $x_6 = 5$.

11.12 O Baixo Rio Colorado consiste em uma série de quatro reservatórios, como mostra a Figura P11.12.

Os balanços de massas podem ser escritos para cada reservatório, e o seguinte conjunto de equações algébricas lineares simultâneas é obtido:

$$\begin{bmatrix} 13,422 & 0 & 0 & 0 \\ -13,422 & 12,252 & 0 & 0 \\ 0 & -12,252 & 12,377 & 0 \\ 0 & 0 & -12,377 & 11,797 \end{bmatrix}$$

$$\times \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 750,5 \\ 300 \\ 102 \\ 30 \end{Bmatrix}$$

onde o vetor à direita consiste nas cargas de cloreto em cada um dos quatro lagos e c_1 , c_2 , c_3 e c_4 são iguais à concentração resultante de cloreto nos lagos Powell, Mead, Mohave e Havasu, respectivamente.

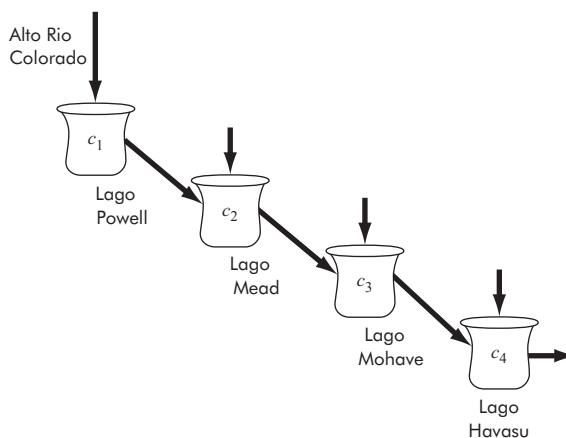


FIGURA P11.12 O Baixo Rio Colorado.

- (a) Use a inversão de matrizes para determinar a concentração em cada um dos quatro lagos.
 (b) Qual deve ser a redução da carga no lago Powell para que a concentração de cloreto no lago Havasu seja 75?
 (c) Usando a norma da soma das colunas, calcule o número de condicionamento e quantos algarismos suspeitos serão gerados ao resolver esse sistema.

11.13 (a) Determine a matriz inversa e o número de condicionamento para a seguinte matriz:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

(b) Repita (a) mas mude a_{33} levemente para 9,1.

11.14 A interpolação polinomial consiste na determinação do único polinômio de $(n - 1)$ -ésima ordem que se ajusta a n pontos. Tais polinômios têm a seguinte forma geral

$$f(x) = p_1x^{n-1} + p_2x^{n-2} + \dots + p_{n-1}x + p_n \quad (\text{P11.14})$$

onde os p 's são coeficientes constantes. Uma maneira simples de calcular os coeficientes é gerar n equações algébricas lineares que possam ser resolvidas simultaneamente para determinação dos coeficientes. Considere que se queira determinar os coeficientes do polinômio de quarta ordem $f(x) = p_1x^4 + p_2x^3 + p_3x^2 + p_4x + p_5$ que passa pelos seguintes cinco pontos: (200; 0,746), (250; 0,675), (300; 0,616), (400; 0,525) e (500; 0,457). Cada um desses pares pode ser substituído na Equação (P11.14)

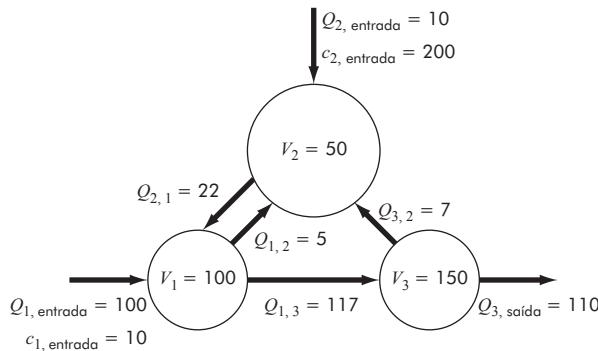
para produzir um sistema de cinco equações com cinco incógnitas (os p 's). Use essa abordagem para determinar os coeficientes. Além disso, determine e interprete o número de condicionamento.

11.15 Um componente químico flui entre três reatores, como ilustra a Figura P11.15. O balanço de massa no estado estacionário pode ser escrito para uma substância que reage com uma cinética de primeira ordem. Por exemplo, o balanço de massa para o reator 1 é

$$Q_{1,\text{entrada}}c_{1,\text{entrada}} - Q_{1,2}c_1 - Q_{1,3}c_1 + Q_{2,1}c_2 - kV_1c_1 = 0 \quad (\text{P11.15})$$

onde $Q_{1,\text{entrada}}$ é o fluxo volumétrico de entrada para o reator 1 (m^3/min), $c_{1,\text{entrada}}$ é a concentração do fluxo de entrada para o reator 1 (g/m^3), Q_{ij} é o fluxo do reator i para o reator j (m^3/min), c_i é a concentração do reator i (g/m^3), k é uma taxa de decaimento de primeira ordem ($/\text{min}$) e V_i é o volume do reator i (m^3).

- (a) Escreva os balanços de massas para os reatores 2 e 3.
 (b) Se $k = 0,1/\text{min}$, escreva os balanços de massas para todos os três reatores como um sistema de equações algébricas lineares.
 (c) Calcule a decomposição LU para esse sistema.
 (d) Use a decomposição LU para calcular a matriz inversa.
 (e) Use a matriz inversa para responder às seguintes perguntas: (i) Quais são as concentrações estacionárias para os três reatores? (ii) Se a concentração do fuxo de entrada

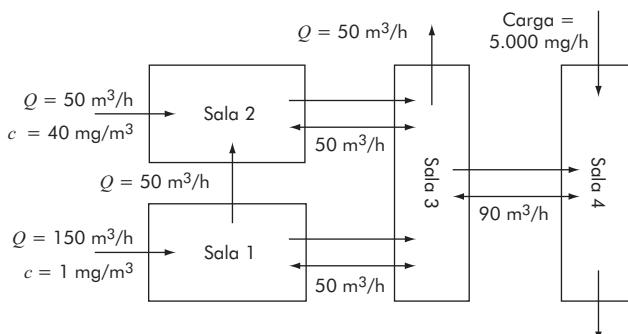
**FIGURA P11.15**

para o reator 2 é definida como zero, qual é a redução resultante na concentração do reator 1? (iii) Se a concentração do fluxo de entrada para o reator 1 é dobrada, e a concentração do fluxo de entrada para o reator 2 é reduzida à metade, qual é a concentração do reator 3?

- 11.16** Como descrito nos Exemplos 8.2 e 11.2, use a matriz inversa para responder os seguintes itens:
- Determine a variação na posição do primeiro saltador, se a massa do terceiro é aumentada para 100 kg.
 - Que força deve ser aplicada ao terceiro saltador de modo que a sua posição seja 140 m?

11.17 Determine a matriz inversa para o circuito elétrico formulado na Seção 8.3. Use a inversa para determinar a nova corrente entre os nós 2 e 5 (i_{52}), se uma tensão de 200 V é aplicada ao nó 6 e a tensão no nó 1 é reduzida à metade.

- 11.18** (a) Utilizando a mesma abordagem descrita na Seção 11.3, desenvolva um balanço de massas em estado estacionário para a configuração de salas apresentada na Figura P11.18.
 (b) Determine a matriz inversa e use-a para calcular as concentrações resultantes nas salas.
 (c) Use a matriz inversa para determinar qual deve ser a redução da carga da sala 4 para manter uma concentração de 20 mg/m³ na sala 2.

**FIGURA P11.18**

12

Métodos iterativos

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é familiarizar o leitor com os métodos iterativos para resolução de equações simultâneas. Após esta leitura completa, você será capaz de:

- ✓ Entender a diferença entre os métodos de Gauss-Seidel e de Jacob.
- ✓ Aprender como avaliar a dominância da diagonal e entender o que ela significa.
- ✓ Reconhecer como o relaxamento pode ser usado para melhorar a convergência de métodos iterativos.
- ✓ Entender como resolver sistemas de equações não lineares com os métodos da substituição sucessiva e de Newton-Raphson.

Os métodos iterativos ou de aproximação fornecem uma alternativa aos métodos de eliminação descritos até agora e são semelhantes às técnicas desenvolvidas para obter as raízes de uma única equação nos Capítulos 5 e 6. Aquelas abordagens consistiam em escolher um valor e então usar um método sistemático para obter uma estimativa refinada da raiz. Como esta parte do livro trata de um problema similar – obter valores que satisfaçam simultaneamente um conjunto de equações – pode-se suspeitar que tais métodos de aproximações são úteis neste contexto. Neste capítulo, apresentaremos abordagens para resolução de equações simultâneas lineares e não lineares.

12.1 SISTEMAS LINEARES: GAUSS-SEIDEL

O *método de Gauss-Seidel* é o método iterativo mais frequentemente usado para resolução de equações algébricas lineares. Considere que tenhamos um conjunto de n equações:

$$[A]\{x\} = \{b\}$$

e que, para sermos concisos, nos limitemos a um conjunto de 3×3 equações. Se os elementos da diagonal forem todos não nulos, é possível isolar x_1 na primeira equação, x_2 na segunda e x_3 na terceira pra obter

$$x_1^j = \frac{b_1 - a_{12}x_2^{j-1} - a_{13}x_3^{j-1}}{a_{11}} \quad (12.1a)$$

$$x_2^j = \frac{b_2 - a_{21}x_1^j - a_{23}x_3^{j-1}}{a_{22}} \quad (12.1b)$$

$$x_3^j = \frac{b_3 - a_{31}x_1^j - a_{32}x_2^j}{a_{33}} \quad (12.1c)$$

onde j e $j - 1$ representam a iteração atual e a anterior.

Para iniciar o processo de resolução, devem ser feitas aproximações iniciais para os x 's. Uma forma simples de obtê-las é supor que são todas nulas. Esses zeros podem ser substituídos na Equação (12.1a), que pode ser usada para calcular um novo valor para $x_1 = b_1/a_{11}$ e substitui-se este valor de x_1 junto com a aproximação anterior nula para x_3 na Equação (12.1b), para calcular um novo valor para x_2 . O processo é repetido para a Equação (12.1c), para se calcular uma nova estimativa para x_3 . Então, volta-se para a primeira equação e o procedimento inteiro é repetido até que a solução converja para valores suficientemente próximos dos valores verdadeiros. A convergência pode ser verificada usando-se o critério de que para todo i ,

$$\varepsilon_{a,i} = \left| \frac{x_i^j - x_i^{j-1}}{x_i^j} \right| \times 100\% \leq \varepsilon_s \quad (12.2)$$

EXEMPLO 12.1 Método de Gauss-Seidel

Use o método de Gauss-Seidel para obter a solução para

$$\begin{aligned} 3x_1 - 0,1x_2 - 0,2x_3 &= 7,85 \\ 0,1x_1 + 7x_2 - 0,3x_3 &= -19,3 \\ 0,3x_1 - 0,2x_2 + 10x_3 &= 71,4 \end{aligned}$$

Observe que a solução é $x_1 = 3$, $x_2 = -2,5$ e $x_3 = 7$.

Solução. Em cada uma das equações, isole a variável na diagonal:

$$x_1 = \frac{7,85 + 0,1x_2 + 0,2x_3}{3} \quad (\text{E12.1.1})$$

$$x_2 = \frac{-19,3 - 0,1x_1 + 0,3x_3}{7} \quad (\text{E12.1.2})$$

$$x_3 = \frac{71,4 - 0,3x_1 + 0,2x_2}{10} \quad (\text{E12.1.3})$$

Considerando que x_2 e x_3 são iguais a zero, a Equação (E12.1.1) pode ser usada para calcular

$$x_1 = \frac{7,85 + 0,1(0) + 0,2(0)}{3} = 2,616667$$

Esse valor, junto com o valor suposto de $x_3 = 0$, pode ser substituído na Equação (E12.1.2) para calcular

$$x_2 = \frac{-19,3 - 0,1(2,616667) + 0,3(0)}{7} = -2,794524$$

A primeira iteração é completada substituindo-se os valores calculados para x_1 e x_2 na Equação (E12.1.3) para obter

$$x_3 = \frac{71,4 - 0,3(2,616667) + 0,2(-2,794524)}{10} = 7,005610$$

Para a segunda iteração, o mesmo processo é repetido para calcular

$$x_1 = \frac{7,85 + 0,1(-2,794524) + 0,2(7,005610)}{3} = 2,990557$$

$$x_2 = \frac{-19,3 - 0,1(2,990557) + 0,3(7,005610)}{7} = -2,499625$$

$$x_3 = \frac{71,4 - 0,3(2,990557) + 0,2(-2,499625)}{10} = 7,000291$$

O método está, portanto, convergindo para a verdadeira solução. Iterações adicionais podem ser aplicadas para melhorar as respostas; porém, em um problema real, a resposta verdadeira não seria conhecida *a priori*. Consequentemente, a Equação (12.2) fornece um meio de estimar o erro. Por exemplo, para x_1 :

$$\varepsilon_{a,1} = \left| \frac{2,990557 - 2,616667}{2,990557} \right| \times 100\% = 12,5\%$$

Para x_2 e x_3 , as estimativas de erro são $\varepsilon_{a,2} = 11,8\%$ e $\varepsilon_{a,3} = 0,076\%$. Observe que, quando determinamos raízes de uma única equação, as formulações como a Equação (12.2) em geral fornecem uma estimativa conservadora de convergência. Assim, quando elas são satisfeitas, garantem que o resultado é conhecido pelo menos dentro da tolerância especificada por ε_s .

À medida que cada novo valor de x é calculado pelo método de Gauss-Seidel, ele é imediatamente usado na próxima equação para se determinar outro valor de x . Dessa forma, se a solução estiver convergindo, a melhor estimativa disponível será empregada. Uma abordagem alternativa, chamada *iteração de Jacob*, utiliza uma tática um pouco diferente: em vez de usar os últimos x 's disponíveis, usa a Equação (12.1) para calcular um conjunto de novos x 's com base no conjunto de antigos x 's. Logo, conforme novos valores são gerados, eles não são imediatamente usados, mas guardados para a próxima iteração.

A diferença entre o método de Gauss-Seidel e a iteração de Jacob é descrita na Figura 12.1. Embora haja certos casos para os quais o método de Jacob é útil, a utilização das melhores estimativas disponíveis no método de Gauss-Seidel em geral o torna o preferido.

12.1.1 Convergência e dominância da diagonal

Observe que o método de Gauss-Seidel é similar à técnica de iteração de ponto fixo simples, que foi usada na Seção 6.1 para determinar as raízes de uma única equação. Lembre-se de que a iteração de ponto fixo simples às vezes era não convergente; isto é, à medida que as iterações avançavam, a resposta movia-se para longe do resultado correto.

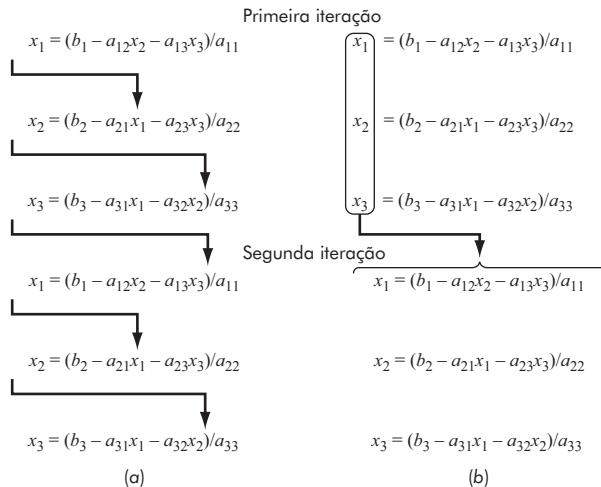


FIGURA 12.1 Descrição gráfica da diferença entre (a) o método de Gauss-Seidel e (b) a iteração de Jacob para resolver equações algébricas lineares simultâneas.

Embora o método de Gauss-Seidel também possa divergir, como ele é projetado para sistemas lineares, sua capacidade para convergir é muito mais previsível que para a iteração de ponto fixo de equações não lineares. Pode-se mostrar que se a condição a seguir é satisfeita, o método de Gauss-Seidel irá convergir:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| \quad (12.3)$$

Isto é, o valor absoluto do coeficiente da diagonal em cada equação deve ser maior que a soma dos valores absolutos dos outros coeficientes na equação. Tais sistemas são chamados *diagonal dominantes*. Esse critério é suficiente, mas não necessário para a convergência, ou seja, embora o método possa às vezes funcionar se a Equação (12.3) não for satisfeita, a convergência é garantida se essa condição for satisfeita. Felizmente, muitos problemas de importância prática na engenharia e na ciência satisfazem tal exigência. Portanto, o método de Gauss-Seidel representa uma abordagem viável para resolução de diversos problemas na engenharia e na ciência.

12.1.2 Função do MATLAB: GaussSeidel

Antes de desenvolver um algoritmo, vamos reescrever o Gauss-Seidel em uma forma que seja compatível com os recursos do MATLAB de efetuar operações com matrizes. Isso é feito expressando a Equação (12.1) como

$$\begin{aligned} x_1^{\text{novo}} &= \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2^{\text{velho}} - \frac{a_{13}}{a_{11}}x_3^{\text{velho}} \\ x_2^{\text{novo}} &= \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1^{\text{novo}} - \frac{a_{23}}{a_{22}}x_3^{\text{velho}} \\ x_3^{\text{novo}} &= \frac{b_3}{a_{33}} - \frac{a_{31}}{a_{33}}x_1^{\text{novo}} - \frac{a_{32}}{a_{33}}x_2^{\text{novo}} \end{aligned}$$

Observe que a solução pode ser expressa de forma concisa na forma matricial como

$$\{x\} = \{d\} - [C]\{x\} \quad (12.4)$$

onde

$$\{d\} = \begin{cases} b_1/a_{11} \\ b_2/a_{22} \\ b_3/a_{33} \end{cases}$$

e

$$[C] = \begin{bmatrix} 0 & a_{12}/a_{11} & a_{13}/a_{11} \\ a_{21}/a_{22} & 0 & a_{23}/a_{22} \\ a_{31}/a_{33} & a_{32}/a_{33} & 0 \end{bmatrix}$$

Uma função do MATLAB para implementar a Equação (12.4) é apresentada na Figura 12.2.

12.1.3 Relaxamento

O relaxamento representa uma pequena modificação no método de Gauss-Seidel e foi criado para melhorar a convergência. Após cada novo valor de x ser calculado usando-se a Equação (12.1), esse valor é modificado por uma média ponderada dos resultados da iteração anterior e da atual:

$$x_i^{\text{novo}} = \lambda x_i^{\text{novo}} + (1 - \lambda)x_i^{\text{velho}} \quad (12.5)$$

onde λ é um fator de peso para o qual deve ser escolhido um valor entre 0 e 2.

Se $\lambda = 1$, $(1 - \lambda)$ é igual a 0, e o resultado não é modificado. No entanto, se a λ for atribuído um valor entre 0 e 1, o resultado é uma média ponderada do resultado atual e do anterior. Esse tipo de modificação é chamado *sub-relaxamento*. Ele é em geral usado para que um sistema não convergente passe a convergir, ou para apressar a convergência, amortecendo as oscilações.

Para valores de λ entre 1 e 2, um peso extra é colocado no valor atual. Nesse caso, existe uma suposição explícita de que o novo valor está se movendo na direção correta da solução verdadeira, mas a taxa muito lenta. Assim, o peso extra λ se presta a melhorar a estimativa, aproximando-a do valor verdadeiro. Portanto, esse tipo de modificação, que é chamado de *sobrerrelaxamento*, é implementado para acelerar a convergência de um sistema já convergente. Essa abordagem é também chamada de *sobrerrelaxamento simultâneo* ou *sucessivo* ou *SOR*.

A escolha de um valor adequado para λ depende criticamente do problema tratado e é, muitas vezes, determinada de maneira empírica, sendo frequentemente desnecessária para uma única solução de um conjunto de equações. Contudo, se o sistema que está sendo estudado deve ser resolvido repetidamente, a eficiência introduzida por uma boa escolha de λ pode ser extremamente importante. Bons exemplos são os sistemas muito grandes de equações algébricas lineares que aparecem na resolução de equações diferenciais parciais em uma variedade de contextos de problemas científicos e de engenharia.

```

function x = GaussSeidel(A,b,es,maxit)
% GaussSeidel: Método de Gauss-Seidel
%   x = GaussSeidel(A,b): Gauss-Seidel sem relaxamento
%   entrada:
%     A = matriz dos coeficientes
%     b = vetor do lado direito
%     es = critério de parada (padrão = 0.00001%)
%     maxit = número máximo de iterações (padrão = 50)
%   output:
%     x = solution vector
if nargin<2,error('são necessários pelo menos 2 argumentos de
entrada'),end
if nargin<4|isempty(maxit),maxit=50;end
if nargin<3|isempty(es),es=0.00001;end
[m,n] = size(A);
if m~=n, error('A matriz A deve ser quadrada'); end
C = A;
for i = 1:n
    C(i,i) = 0;
    x(i) = 0;
end
x = x';
for i = 1:n
    C(i,1:n) = C(i,1:n)/A(i,i);
end
for i = 1:n
    d(i) = b(i)/A(i,i);
end
iter = 0;
while (1)
    x_velho = x;
    for i = 1:n
        x(i) = d(i)-C(i,:)*x;
        if x(i) ~= 0
            ea(i) = abs((x(i) - x_velho(i))/x(i)) * 100;
        end
    end
    iter = iter+1;
    if max(ea)<=es | iter >= maxit, break, end
end

```

FIGURA 12.2 Função do MATLAB para implementar o método de Gauss-Seidel.

EXEMPLO 12.2 Método de Gauss-Seidel com relaxamento

Resolva o sistema a seguir com o método de Gauss-Seidel utilizando sobrerelaxamento ($\lambda = 1,2$) e um critério de parada de $\varepsilon_s = 10\%$:

$$\begin{aligned} -3x_1 + 12x_2 &= 9 \\ 10x_1 - 2x_2 &= 8 \end{aligned}$$

Solução. Rearrange as equações de modo que elas sejam diagonalmente dominantes e isole x_1 na primeira equação e x_2 na segunda:

$$\begin{aligned} x_1 &= \frac{8 + 2x_2}{10} = 0,8 + 0,2x_2 \\ x_2 &= \frac{9 + 3x_1}{12} = 0,75 + 0,25x_1 \end{aligned}$$

Primeira iteração: Utilizando aproximações iniciais de $x_1 = x_2 = 0$, pode-se calcular x_1 :

$$x_1 = 0,8 + 0,2(0) = 0,8$$

Antes de calcular x_2 , aplica-se o fator de relaxamento ao resultado para x_1 :

$$x_{1,r} = 1,2(0,8) - 0,2(0) = 0,96$$

O subscrito r é usado para indicar que esse é o valor “relaxado”. Esse resultado é, então, usado para calcular x_2 :

$$x_2 = 0,75 + 0,25(0,96) = 0,99$$

Aplica-se o fator de relaxamento a esse resultado para fornecer

$$x_{2,r} = 1,2(0,99) - 0,2(0) = 1,188$$

Neste ponto, poder-se-ia calcular os erros estimados com a Equação (12.2). Porém, como assumiu-se aproximações iniciais iguais a zero, os erros para ambas as variáveis seria 100%.

Segunda iteração: Utilizando o mesmo procedimento como para a primeira iteração, a segunda iteração produz:

$$x_1 = 0,8 + 0,2(1,188) = 1,0376$$

$$x_{1,r} = 1,2(1,0376) - 0,2(0,96) = 1,05312$$

$$\varepsilon_{a,1} = \left| \frac{1,05312 - 0,96}{1,05312} \right| \times 100\% = 8,84\%$$

$$x_2 = 0,75 + 0,25(1,05312) = 1,01328$$

$$x_{2,r} = 1,2(1,01328) - 0,2(1,188) = 0,978336$$

$$\varepsilon_{a,2} = \left| \frac{0,978336 - 1,188}{0,978336} \right| \times 100\% = 21,43\%$$

Como agora temos valores diferentes de zero obtidos da primeira iteração, podemos determinar estimativas aproximadas do erro à medida que cada novo valor é calculado. Neste ponto, embora a estimativa do erro para a primeira variável tenha caído abaixo do critério de parada de 10%, o mesmo critério ainda não foi satisfeito para a segunda variável. Portanto, deve-se implementar outra iteração.

Terceira iteração:

$$x_1 = 0,8 + 0,2(0,978336) = 0,995667$$

$$x_{1,r} = 1,2(0,995667) - 0,2(1,05312) = 0,984177$$

$$\varepsilon_{a,1} = \left| \frac{0,984177 - 1,05312}{0,984177} \right| \times 100\% = 7,01\%$$

$$x_2 = 0,75 + 0,25(0,984177) = 0,996044$$

$$x_{2,r} = 1,2(0,996044) - 0,2(0,978336) = 0,999586$$

$$\varepsilon_{a,2} = \left| \frac{0,999586 - 0,978336}{0,999586} \right| \times 100\% = 2,13\%$$

Neste ponto, podemos finalizar os cálculos, pois ambas as estimativas de erro caíram abaixo do critério de parada de 10%. Os resultados neste momento, $x_1 = 0,984177$ e $x_2 = 0,999586$, estão convergindo para a solução exata de $x_1 = x_2 = 1$.

12.2 SISTEMAS NÃO LINEARES

O sistema a seguir é formado por um conjunto de duas equações simultâneas não lineares com duas incógnitas:

$$x_1^2 + x_1 x_2 = 10 \quad (12.6a)$$

$$x_2 + 3x_1 x_2^2 = 57 \quad (12.6b)$$

Ao contrário dos sistemas lineares, cujas equações representam retas (lembre-se da Figura 9.1), os gráficos associados a essas equações (não lineares) são curvas de x_2 versus x_1 . Como mostra a Figura 12.3, a solução é a interseção das curvas.

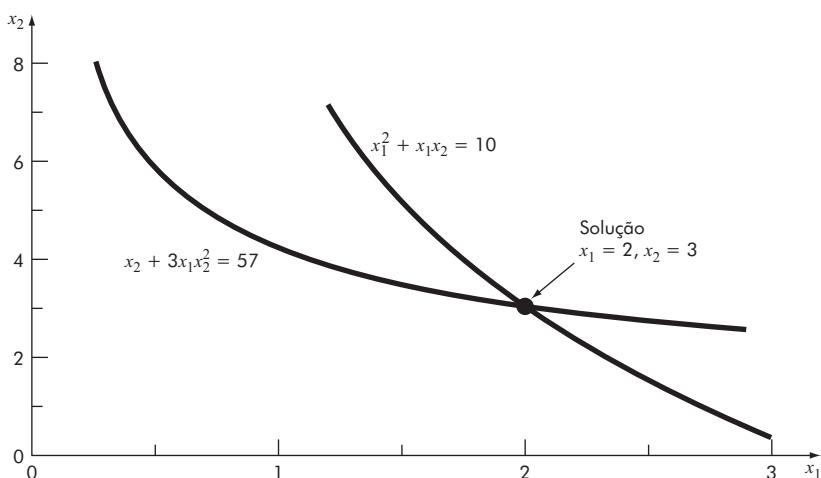


FIGURA 12.3 Descrição gráfica da solução de duas equações simultâneas não lineares.

Assim como foi feito na determinação de raízes para equações não lineares individuais, tais sistemas de equações podem ser expressos genericamente como

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \tag{12.7}$$

Portanto, a solução são os valores dos x 's que tornam as equações iguais a zero.

12.2.1 Substituição sucessiva

Uma abordagem simples para resolução da Equação (12.7) é usar a mesma estratégia empregada para a iteração de ponto fixo e o método de Gauss-Seidel, isto é, cada uma das equações não lineares pode ser resolvida para uma das variáveis (incógnitas). Essas equações podem, então, ser implementadas iterativamente para calcular novos valores que, espera-se, convergirão para as soluções. Essa abordagem, que é chamada *substituição sucessiva*, é ilustrada no exemplo a seguir.

EXEMPLO 12.3 Substituição sucessiva para um sistema não linear

Use a substituição sucessiva para determinar as raízes da Equação (12.6). Observe que um par correto de raízes é $x_1 = 2$ e $x_2 = 3$. Inicie os cálculos com aproximações de $x_1 = 1,5$ e $x_2 = 3,5$.

Solução. A Equação (12.6a) pode ser resolvida para

$$x_1 = \frac{10 - x_2^2}{x_2} \tag{E12.3.1}$$

e a Equação (12.6b) pode ser resolvida para

$$x_2 = 57 - 3x_1x_2^2 \tag{E12.3.2}$$

Com base nas aproximações iniciais, a Equação (E12.3.1) pode ser usada para determinar um novo valor de x_1 :

$$x_1 = \frac{10 - (1,5)^2}{3,5} = 2,21429$$

Esse resultado e o valor inicial de $x_2 = 3,5$ podem ser substituídos na Equação (E12.3.2) para determinar um novo valor de x_2 :

$$x_2 = 57 - 3(2,21429)(3,5)^2 = -24,37516$$

Considerando os resultados, a abordagem parece estar divergindo. Esse comportamento é ainda mais pronunciado na segunda iteração:

$$x_1 = \frac{10 - (2,21429)^2}{-24,37516} = -0,20910$$

$$x_2 = 57 - 3(-0,20910)(-24,37516)^2 = 429,709$$

Obviamente, a abordagem está deteriorando.

Agora, vamos repetir os cálculos, mas com as equações originais definidas em uma forma diferente. Por exemplo, uma solução alternativa da Equação (12.6a) é

$$x_1 = \sqrt{10 - x_1 x_2}$$

e da Equação (12.6b) é

$$x_2 = \sqrt{\frac{57 - x_2}{3x_1}}$$

Agora, os resultados são mais satisfatórios:

$$x_1 = \sqrt{10 - 1,5(3,5)} = 2,17945$$

$$x_2 = \sqrt{\frac{57 - 3,5}{3(2,17945)}} = 2,86051$$

$$x_1 = \sqrt{10 - 2,17945(2,86051)} = 1,94053$$

$$x_2 = \sqrt{\frac{57 - 2,86051}{3(1,94053)}} = 3,04955$$

Assim, a abordagem está convergindo para os valores verdadeiros de $x_1 = 2$ e $x_2 = 3$.

O exemplo anterior ilustra a maior deficiência da substituição sucessiva – ou seja, a convergência geralmente depende da maneira como as equações são formuladas. Além disso, mesmo naqueles casos em que a convergência é possível, a abordagem pode divergir se as aproximações iniciais não estiverem suficientemente próximas da solução verdadeira. Esses critérios são tão restritivos que a iteração de ponto fixo tem utilidade limitada para resolução de sistemas não lineares.

12.2.2 Newton-Raphson

Do mesmo modo que a iteração de ponto fixo pode ser usada para resolver sistemas de equações não lineares, outros métodos abertos de localização de raízes, tal como o método de Newton-Raphson, podem ser usados para o mesmo propósito. Lembre-se de que este método baseava-se no emprego da derivada (isto é, da inclinação) de uma função para fazer uma estimativa da sua interseção com o eixo da variável independente – isto é, a raiz. No Capítulo 6, usou-se uma dedução geométrica para calcular essa estimativa. Uma alternativa é deduzi-la a partir de uma expansão em série de Taylor de primeira ordem:

$$f(x_{i+1}) = f(x_i) + (x_{i+1} - x_i) f'(x_i) \quad (12.8)$$

onde x_i é a aproximação inicial da raiz e x_{i+1} é o ponto em que a tangente intercepta o eixo x . Nesta interseção, $f(x_{i+1})$, por definição, é igual a zero, e a Equação (12.8) pode ser rearranjada para fornecer

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (12.9)$$

que é a forma para uma única equação do método de Newton-Raphson.

A forma para várias equações é deduzida de maneira idêntica. Entretanto, uma série de Taylor para diversas variáveis deve ser usada para representar o fato de que mais de uma variável independente contribui para a determinação da raiz. Para o caso de duas variáveis, a série de Taylor pode ser escrita para cada equação não linear como

$$f_{1,i+1} = f_{1,i} + (x_{1,i+1} - x_{1,i}) \frac{\partial f_{1,i}}{\partial x_1} + (x_{2,i+1} - x_{2,i}) \frac{\partial f_{1,i}}{\partial x_2} \quad (12.10a)$$

$$f_{2,i+1} = f_{2,i} + (x_{1,i+1} - x_{1,i}) \frac{\partial f_{2,i}}{\partial x_1} + (x_{2,i+1} - x_{2,i}) \frac{\partial f_{2,i}}{\partial x_2} \quad (12.10b)$$

Da mesma forma como para a versão para uma única equação, a estimativa da raiz corresponde aos valores de x_1 e x_2 nos quais $f_{1,i+1}$ e $f_{2,i+1}$ são iguais a zero. Para tal situação, a Equação (12.10) pode ser reorganizada para fornecer

$$\frac{\partial f_{1,i}}{\partial x_1} x_{1,i+1} + \frac{\partial f_{1,i}}{\partial x_2} x_{2,i+1} = -f_{1,i} + x_{1,i} \frac{\partial f_{1,i}}{\partial x_1} + x_{2,i} \frac{\partial f_{1,i}}{\partial x_2} \quad (12.11a)$$

$$\frac{\partial f_{2,i}}{\partial x_1} x_{1,i+1} + \frac{\partial f_{2,i}}{\partial x_2} x_{2,i+1} = -f_{2,i} + x_{1,i} \frac{\partial f_{2,i}}{\partial x_1} + x_{2,i} \frac{\partial f_{2,i}}{\partial x_2} \quad (12.11b)$$

Já que todos os valores subscritos com i são conhecidos (eles correspondem à última aproximação), as únicas incógnitas são $x_{1,i+1}$ e $x_{2,i+1}$. Portanto, a Equação (12.11) é um conjunto de duas equações lineares com duas incógnitas. Consequentemente, manipulações algébricas (por exemplo, a regra de Cramer) podem ser usadas para determinar

$$x_{1,i+1} = x_{1,i} - \frac{f_{1,i} \frac{\partial f_{2,i}}{\partial x_2} - f_{2,i} \frac{\partial f_{1,i}}{\partial x_2}}{\frac{\partial f_{1,i}}{\partial x_1} \frac{\partial f_{2,i}}{\partial x_2} - \frac{\partial f_{1,i}}{\partial x_2} \frac{\partial f_{2,i}}{\partial x_1}} \quad (12.12a)$$

$$x_{2,i+1} = x_{2,i} - \frac{f_{2,i} \frac{\partial f_{1,i}}{\partial x_1} - f_{1,i} \frac{\partial f_{2,i}}{\partial x_1}}{\frac{\partial f_{1,i}}{\partial x_1} \frac{\partial f_{2,i}}{\partial x_2} - \frac{\partial f_{1,i}}{\partial x_2} \frac{\partial f_{2,i}}{\partial x_1}} \quad (12.12b)$$

O denominador de cada uma dessas equações é chamado formalmente de determinante da *matriz Jacobiana* do sistema.

A Equação (12.12) é a versão para duas equações do método de Newton-Raphson. Como no exemplo a seguir, ele pode ser usado iterativamente para aproximar as raízes de duas equações simultâneas.

EXEMPLO 12.4 Newton-Raphson para um sistema não linear

Use o método de Newton-Raphson para equações múltiplas para determinar as raízes da Equação (12.6). Inicie os cálculos com as aproximações $x_1 = 1,5$ e $x_2 = 3,5$.

Solução. Determine primeiro as derivadas parciais e calcule-as nas aproximações iniciais de x_1 e x_2 :

$$\begin{aligned}\frac{\partial f_{1,0}}{\partial x_1} &= 2x_1 + x_2 = 2(1,5) + 3,5 = 6,5 & \frac{\partial f_{1,0}}{\partial x_2} &= x_1 = 1,5 \\ \frac{\partial f_{2,0}}{\partial x_1} &= 3x_2^2 = 3(3,5)^2 = 36,75 & \frac{\partial f_{2,0}}{\partial x_2} &= 1 + 6x_1x_2 = 1 + 6(1,5)(3,5) = 32,5\end{aligned}$$

Portanto, o determinante da matriz Jacobiana para a primeira iteração é:

$$6,5(32,5) - 1,5(36,75) = 156,125$$

Os valores das funções podem ser calculados nas aproximações iniciais como

$$\begin{aligned}f_{1,0} &= (1,5)^2 + 1,5(3,5) - 10 = -2,5 \\ f_{2,0} &= 3,5 + 3(1,5)(3,5)^2 - 57 = 1,625\end{aligned}$$

Esses valores podem ser substituídos na Equação (12.12) para fornecer

$$\begin{aligned}x_1 &= 1,5 - \frac{-2,5(32,5) - 1,625(1,5)}{156,125} = 2,03603 \\ x_2 &= 3,5 - \frac{1,625(6,5) - (-2,5)(36,75)}{156,125} = 2,84388\end{aligned}$$

Desse modo, os resultados estão convergindo para os valores verdadeiros $x_1 = 2$ e $x_2 = 3$. O cálculo pode ser repetido até que uma acurácia aceitável seja obtida.

Quando o método de Newton-Raphson para equações múltiplas funciona, ele exibe a mesma convergência quadrática rápida como a versão para uma única equação. No entanto, da mesma forma como na substituição sucessiva, a abordagem de Newton-Raphson pode divergir se as aproximações iniciais não estiverem suficientemente próximas das raízes verdadeiras. Enquanto os métodos gráficos podiam ser usados para se encontrar boas aproximações para o caso de uma única equação, nenhum processo simples está disponível para a versão com diversas equações. Embora existam algumas abordagens avançadas para se obter primeiras estimativas aceitáveis, em geral as aproximações iniciais precisam ser obtidas com base na tentativa e erro e no conhecimento do sistema físico que está sendo modelado.

A abordagem de Newton-Raphson para duas equações pode ser generalizada para resolver n equações simultâneas. Para isso, a Equação (12.11) pode ser escrita para a k -ésima equação como

$$\begin{aligned}\frac{\partial f_{k,i}}{\partial x_1}x_{1,i+1} + \frac{\partial f_{k,i}}{\partial x_2}x_{2,i+1} + \cdots + \frac{\partial f_{k,i}}{\partial x_n}x_{n,i+1} &= -f_{k,i} + x_{1,i}\frac{\partial f_{k,i}}{\partial x_1} + x_{2,i}\frac{\partial f_{k,i}}{\partial x_2} \\ &\quad + \cdots + x_{n,i}\frac{\partial f_{k,i}}{\partial x_n}\end{aligned}\tag{12.13}$$

onde o primeiro índice, k , representa a equação ou a variável, e o segundo índice denota se o valor ou a função em questão é o valor atual (i) ou o valor seguinte ($i + 1$). Observe que as únicas incógnitas na Equação (12.13) são os termos $x_{k,i+1}$ no lado esquerdo da equação. Todas as outras quantidades estão localizadas no valor atual (i) e, portanto, são conhecidas em qualquer iteração. Consequentemente, o conjunto de equações representado de forma genérica pela Equação (12.13) (isto é, com $k = 1, 2, \dots, n$) constitui um conjunto de equações lineares simultâneas, que pode ser resolvido numericamente pelos métodos de eliminação elaborados nos capítulos anteriores.

A notação matricial pode ser usada para expressar a Equação (12.13) de forma concisa como

$$[J]\{x_{i+1}\} = -\{f\} + [J]\{x_i\} \quad (12.14)$$

onde as derivadas parciais avaliadas em i são escritas como a *matriz Jacobiana* das derivadas parciais:

$$[J] = \begin{bmatrix} \frac{\partial f_{1,i}}{\partial x_1} & \frac{\partial f_{1,i}}{\partial x_2} & \cdots & \frac{\partial f_{1,i}}{\partial x_n} \\ \frac{\partial f_{2,i}}{\partial x_1} & \frac{\partial f_{2,i}}{\partial x_2} & \cdots & \frac{\partial f_{2,i}}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_{n,i}}{\partial x_1} & \frac{\partial f_{n,i}}{\partial x_2} & \cdots & \frac{\partial f_{n,i}}{\partial x_n} \end{bmatrix} \quad (12.15)$$

Os valores inicial e final são expressos na forma vetorial como

$$\{x_i\}^T = [x_{1,i} \ x_{2,i} \ \cdots \ x_{n,i}]$$

e

$$\{x_{i+1}\}^T = [x_{1,i+1} \ x_{2,i+1} \ \cdots \ x_{n,i+1}]$$

Por fim, os valores da função em i podem ser expressos por

$$\{f\}^T = [f_{1,i} \ f_{2,i} \ \cdots \ f_{n,i}]$$

A Equação (12.14) pode ser resolvida usando-se uma técnica como a eliminação de Gauss, que pode ser repetida iterativamente para se obter estimativas refinadas de forma parecida com o caso das duas equações abordado no Exemplo 12.4.

Uma visão sobre a solução pode ser obtida resolvendo-se a Equação (12.14) com inversão de matriz. Lembre-se de que a versão para uma única equação do método de Newton-Raphson é

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (12.16)$$

Se a Equação (12.14) é resolvida multiplicando-a pela inversa da matriz Jacobiana, o resultado é

$$\{x_{i+1}\} = \{x_i\} - [J]^{-1} \{f\} \quad (12.17)$$

A comparação das Equações (12.16) e (12.17) ilustra claramente os paralelos entre as duas equações. Em essência, a matriz Jacobiana é análoga à derivada de uma função de diversas variáveis.

Tais cálculos matriciais podem ser implementados de forma bastante eficiente no MATLAB, o que pode ser ilustrado utilizando o MATLAB para duplicar os cálculos do Exemplo 12.4. Depois de definir as aproximações iniciais, pode-se calcular a matriz Jacobiana e os valores da função como

```
>> x=[1.5; 3.5];
>> J=[2*x(1)+x(2) x(1); 3*x(2)^2 1+6*x(1)*x(2)]
J =
    6.5000    1.5000
   36.7500   32.5000
>> f=[x(1)^2+x(1)*x(2)-10; x(2)+3*x(1)*x(2)^2-57]
f =
    -2.5000
    1.6250
```

Então, pode-se implementar a Equação (12.17) para produzir as aproximações melhoradas

```
>> x=x-J\f
x =
    2.0360
    2.8439
```

Embora possamos continuar as iterações no modo de comando, uma alternativa melhor é expressar o algoritmo como uma função do MATLAB. Como mostra a Figura 12.4, o algoritmo calcula os valores da função e a Jacobiana em um dado valor de x . Em seguida, ele chama essa função e implementa a Equação (12.17) em um modo iterativo. A função itera até atingir um limite superior de iterações (`maxit`) ou um erro relativo percentual especificado (`es`).

Deve-se notar que há duas deficiências na abordagem anterior. A primeira é que a Equação (12.15) é, algumas vezes, inconveniente de ser calculada. Assim, variações do método de Newton-Raphson foram desenvolvidas para contornar esse problema. Como esperado, a maioria delas é baseada em aproximações por diferenças finitas das derivadas parciais que compõem $[J]$. A segunda deficiência do método de Newton-Raphson para equações múltiplas é que, em geral, são necessárias estimativas excelentes para os valores iniciais para garantir a convergência. Como esses valores são geralmente difíceis de obter, foram desenvolvidas outras abordagens alternativas, que são mais lentas do que o método de Newton-Raphson, mas que têm um comportamento de convergência melhor. Uma abordagem comum é reformular o sistema não linear como uma única equação:

$$F(x) = \sum_{i=1}^n [f_i(x_1, x_2, \dots, x_n)]^2$$

onde $f_i(x_1, x_2, \dots, x_n)$ é o i -ésimo membro do sistema original da Equação (12.7). Os valores de x que minimizam essa função também representam a solução do sistema não linear, portanto, técnicas de otimização não lineares podem ser empregadas para obter soluções.

```

function [x,f,ea,iter]=newtmult(func,x0,es,maxit,varargin)
% newtmult: Método de Newton-Raphson para solução de sistemas
não lineares
% [x,f,ea,iter]=newtmult(func,x0,es,maxit,p1,p2,...):
% usa o método de Newton-Raphson para encontrar as raízes de
% um sistema de equações não lineares
% entrada:
% func = nome da função que retorna f e J
% x0 = aproximação inicial
% es = erro relativo percentual desejado (padrão = 0.0001%)
% maxit = número máximo de iterações permitidas (padrão = 50)
% p1,p2,... = parâmetros adicionais usados pela função
% saída:
% x = vetor das raízes
% f = vetor das funções calculadas nas raízes
% ea = erro relativo percentual aproximado (%)
% iter = número de iterações
if nargin<2,error('são necessários pelo menos 2 argumentos de
entrada'),end
if nargin<3|isempty(es),es=0.0001;end
if nargin<4|isempty(maxit),maxit=50;end
iter = 0;
x=x0;
while (1)
    [J,f]=func(x,varargin{:});
    dx=J\f;
    x=x-dx;
    iter = iter + 1;
    ea=100*max(abs(dx./x));
    if iter>=maxit|ea<=es, break, end
end

```

FIGURA 12.4 Função do MATLAB para implementar o método de Newton-Raphson para sistemas de equações não lineares.

► 12.3 ESTUDO DE CASO REAÇÕES QUÍMICAS

Contextualização. Sistemas de equações não lineares ocorrem frequentemente na caracterização de reações químicas. Por exemplo, as reações químicas a seguir ocorrem em um sistema fechado:



No equilíbrio, elas podem ser caracterizadas por

$$K_1 = \frac{c_c}{c_a^2 c_b} \quad (12.20)$$



$$K_2 = \frac{c_c}{c_a c_d} \quad (12.21)$$

onde a nomenclatura c_i representa a concentração do constituinte i . Se x_1 e x_2 são os números de mols de C que são produzidos devido a primeira e segunda reações, respectivamente, formule as relações de equilíbrio como um par de duas equações simultâneas não lineares. Se $K_1 = 4 \times 10^{-4}$, $K_2 = 3,7 \times 10^{-2}$, $c_{a,0} = 50$, $c_{b,0} = 20$, $c_{c,0} = 5$ e $c_{d,0} = 10$, empregue o método de Newton-Raphson para resolver essas equações.

Solução. Utilizando a estequiometria das Equações (12.18) e (12.19), a concentração de cada constituinte pode ser representada em termos de x_1 e x_2 como

$$c_a = c_{a,0} - 2x_1 - x_2 \quad (12.22)$$

$$c_b = c_{b,0} - x_1 \quad (12.23)$$

$$c_c = c_{c,0} + x_1 + x_2 \quad (12.24)$$

$$c_d = c_{d,0} - x_2 \quad (12.25)$$

onde o subscrito 0 designa a concentração inicial de cada constituinte. Esses valores podem ser substituídos nas Equações (12.20) e (12.21) para fornecer

$$K_1 = \frac{(c_{c,0} + x_1 + x_2)}{(c_{a,0} - 2x_1 - x_2)^2(c_{b,0} - x_1)}$$

$$K_2 = \frac{(c_{c,0} + x_1 + x_2)}{(c_{a,0} - 2x_1 - x_2)(c_{d,0} - x_2)}$$

Dados os valores dos parâmetros, essas são duas equações não lineares com duas incógnitas.

$$f_1(x_1, x_2) = \frac{5 + x_1 + x_2}{(50 - 2x_1 - x_2)^2(20 - x_1)} - 4 \times 10^{-4} \quad (12.26)$$

$$f_2(x_1, x_2) = \frac{(5 + x_1 + x_2)}{(50 - 2x_1 - x_2)(10 - x_2)} - 3,7 \times 10^{-2} \quad (12.27)$$

Para usar o método de Newton-Raphson, deve-se determinar a Jacobiana tomando as derivadas parciais das Equações (12.26) e (12.27). Embora isso seja possível, o cálculo das derivadas consome muito tempo. Uma alternativa é representá-las por diferenças finitas em um modo similar à abordagem utilizada para o método da secante modificada na Seção 6.3. Por exemplo, as derivadas parciais que compreendem a matriz Jacobiana podem ser calculadas como

$$\frac{\partial f_1}{\partial x_1} = \frac{f_1(x_1 + \delta x_1, x_2) - f_1(x_1, x_2)}{\delta x_1} \quad \frac{\partial f_1}{\partial x_2} = \frac{f_1(x_1, x_2 + \delta x_2) - f_1(x_1, x_2)}{\delta x_2}$$

$$\frac{\partial f_2}{\partial x_1} = \frac{f_2(x_1 + \delta x_1, x_2) - f_2(x_1, x_2)}{\delta x_1} \quad \frac{\partial f_2}{\partial x_2} = \frac{f_2(x_1, x_2 + \delta x_2) - f_2(x_1, x_2)}{\delta x_2}$$

Essas relações podem, então, ser expressas como uma função do MATLAB para calcular tantos os valores da função quanto a Jacobina como



```

function [J,f]=jfreact(x,varargin)
del=0.000001;
df1dx1=(u(x(1)+del*x(1),x(2))-u(x(1),x(2)))/(del*x(1));
df1dx2=(u(x(1),x(2)+del*x(2))-u(x(1),x(2)))/(del*x(2));
df2dx1=(v(x(1)+del*x(1),x(2))-v(x(1),x(2)))/(del*x(1));
df2dx2=(v(x(1),x(2)+del*x(2))-v(x(1),x(2)))/(del*x(2));
J=[df1dx1 df1dx2;df2dx1 df2dx2];
f1=u(x(1),x(2));
f2=v(x(1),x(2));
f=[f1;f2];

function f=u(x,y)
f = (5 + x + y) / (50 - 2 * x - y) ^ 2 / (20 - x) - 0.0004;

function f=v(x,y)
f = (5 + x + y) / (50 - 2 * x - y) / (10 - y) - 0.037;

```

A função `newtmult` (Figura 12.4) pode em seguida ser empregada para determinar as raízes, dadas as aproximações iniciais $x_1 = x_2 = 3$:

```

>> format short e, x0=[3; 3];
>> [x,f,ea,iter]=newtmult(@jfreact,x0)

x =
    3.3366e+000
    2.6772e+000

f =
    -7.1286e-017
    8.5973e-014

ea =
    5.2237e-010

iter =
    4

```

Depois de quatro iterações, uma solução de $x_1 = 3,3366$ e $x_2 = 2,6772$ é obtida. Esses valores podem então ser substituídos nas Equações (12.22) até (12.25) para calcular as concentrações de equilíbrio para os quatro constituintes.

$$\begin{aligned}
 c_a &= 50 - 2(3,3366) - 2,6772 = 40,6496 \\
 c_b &= 20 - 3,3366 = 16,6634 \\
 c_c &= 5 + 3,3366 + 2,6772 = 11,0138 \\
 c_d &= 10 - 2,6772 = 7,3228
 \end{aligned}$$

►►PROBLEMAS

12.1 Resolva o sistema a seguir utilizando três iterações com o método de Gauss-Seidel com sobrerelaxamento ($\lambda = 1,25$). Se necessário, reorganize as equações e mostre todos os passos em sua solução, incluindo suas estimativas de erro. Ao finalizar os cálculos, determine o erro verdadeiro de seus resultados finais.

$$\begin{aligned} 3x_1 + 8x_2 &= 11 \\ 6x_1 - x_2 &= 5 \end{aligned}$$

12.2 (a) Use o método de Gauss-Seidel para resolver o sistema a seguir até que o erro relativo percentual caia abaixo de $\varepsilon_s = 5\%$:

$$\left[\begin{array}{ccc} 0,8 & -0,4 & \\ -0,4 & 0,8 & -0,4 \\ -0,4 & 0,8 & \end{array} \right] \left\{ \begin{array}{l} x_1 \\ x_2 \\ x_3 \end{array} \right\} = \left\{ \begin{array}{l} 41 \\ 25 \\ 105 \end{array} \right\}$$

(b) Repita (a) mas use sobrerelaxamento com $\lambda = 1,2$.

12.3 Use o método de Gauss-Seidel para resolver o seguinte sistema até que o erro relativo percentual caia abaixo de $\varepsilon_s = 5\%$:

$$\begin{aligned} 10x_1 + 2x_2 - x_3 &= 27 \\ -3x_1 - 6x_2 + 2x_3 &= -61,5 \\ x_1 + x_2 + 5x_3 &= -21,5 \end{aligned}$$

12.4 Repita o Problema 12.3, mas utilize a iteração de Jacob.

12.5 O sistema de equações a seguir é projetado para determinar concentrações (os c 's em g/m³) em uma série de reatores acoplados como uma função da quantidade de entrada de massa para cada reator (o lado direito das equações em g/dia):

$$\begin{aligned} 15c_1 - 3c_2 - c_3 &= 3.800 \\ -3c_1 + 18c_2 - 6c_3 &= 1.200 \\ -4c_1 - c_2 + 12c_3 &= 2.350 \end{aligned}$$

Resolva esse problema com o método de Gauss-Seidel para $\varepsilon_s = 5\%$.

12.6 Use o método de Gauss-Seidel (a) sem relaxamento e (b) com relaxamento ($\lambda = 1,2$) para resolver o sistema a seguir com uma tolerância de $\varepsilon_s = 5\%$. Se necessário, rearranje as equações para garantir a convergência.

$$\begin{aligned} 2x_1 - 6x_2 - x_3 &= -38 \\ -3x_1 - x_2 + 7x_3 &= -34 \\ -8x_1 + x_2 - 2x_3 &= -20 \end{aligned}$$

12.7 Dos três conjuntos de equações lineares a seguir, identifique o(s) conjunto(s) que você não poderia resolver utilizando um método iterativo

como o de Gauss-Seidel. Mostre, utilizando qualquer número de iterações, que necessariamente sua solução não converge. Estabeleça de modo claro seu critério de convergência (como você sabe que a solução não está convergindo).

Conjunto 1	Conjunto 2	Conjunto 3
$8x + 3y + z = 12$	$x + y + 5z = 7$	$-x + 3y + 5z = 7$
$-6x + 7z = 1$	$x + 4y - z = 4$	$-2x + 4y - 5z = -3$
$2x + 4y - z = 5$	$3x + y - z = 3$	$2y - z = 1$

12.8 Determine a solução das equações não lineares simultâneas

$$\begin{aligned} y &= -x^2 + x + 0,5 \\ y + 5xy &= x^2 \end{aligned}$$

Use o método de Newton-Raphson e empregue aproximações iniciais de $x = y = 1,2$.

12.9 Determine a solução das equações não lineares simultâneas:

$$\begin{aligned} x^2 &= 5 - y^2 \\ y + 1 &= x^2 \end{aligned}$$

- (a) Graficamente.
- (b) Pelo método da substituição sucessiva, utilizando aproximações iniciais de $x = y = 1,5$.
- (c) Pelo método de Newton-Raphson, utilizando aproximações iniciais de $x = y = 1,5$.

12.10 A Figura P12.10 mostra um processo de trocas químicas que consiste em uma série de reatores nos quais um gás, fluindo da esquerda para a direita, passa sobre um líquido fluindo da direita para a esquerda. A transferência de um produto químico do gás para o líquido se dá a uma taxa proporcional à diferença de concentração entre o gás e o líquido em cada reator. No estado estacionário, um balanço de massas para o primeiro reator pode ser escrito para o gás como

$$Q_{GC_{G0}} - Q_{GC_{G1}} + D(c_{L1} - c_{G1}) = 0$$

e para o líquido como

$$Q_{LC_{L2}} - Q_{LC_{L1}} + D(c_{G1} - c_{L1}) = 0$$

onde Q_G e Q_L são as vazões do gás e do líquido, respectivamente, e D é a taxa de troca entre gás e líquido. Balanços semelhantes podem ser escritos para os outros reatores. Use o método de Gauss-Seidel sem relaxamento para calcular as concentrações dados os seguintes valores: $Q_G = 2$, $Q_L = 1$, $D = 0,8$, $c_{G0} = 100$, $c_{L6} = 10$.

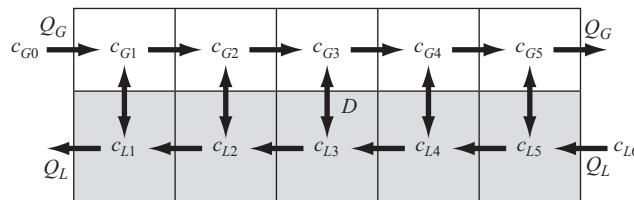


FIGURA P12.10

12.11 A distribuição estacionária de temperatura em uma placa aquecida pode ser modelada pela *equação de Laplace*:

$$0 = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

Se a placa for representada por uma série de nós (Figura P12.11), pode-se substituir as segundas derivadas por diferenças divididas finitas centrais, o que resulta em um sistema de equações algébricas lineares. Use o método de Gauss-Seidel para determinar as temperaturas nos nós na Figura P12.11.

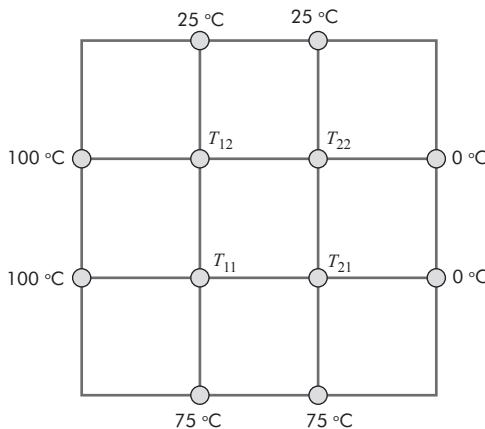


FIGURA P12.11

12.12 Desenvolva sua própria função no MATLAB para o método de Gauss-Seidel sem relaxamento com base na Figura 12.2, mas altere a primeira linha de modo que a função retorne o erro aproximado e o número de iterações:

```
function [x,ea,iter] = ...
GaussSeidel(A,b,es,maxit)
```

Teste sua função duplicando o Exemplo 12.1 e, em seguida, use-a para resolver o Problema 12.2a.

12.13 Desenvolva sua própria função no MATLAB para o método de Gauss-Seidel com relaxamento. A primeira linha da função deve ser da seguinte forma:

```
function [x,ea,iter] = ...
GaussSeidelR(A,b,lambda,es,maxit)
```

Quando o usuário não entrar com um valor de λ , deve-se definir o valor-padrão como $\lambda = 1$. Teste sua função duplicando o Exemplo 12.2 e, em seguida, use-a para resolver o Problema 12.2b.

12.14 Desenvolva sua própria função no MATLAB para o método de Newton-Raphson para sistemas de equações não lineares, baseando-se na Figura 12.4. Teste sua função resolvendo o Exemplo 12.4 e, em seguida, use-a para resolver o Problema 12.8.

13

Autovalores

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é introduzir os autovalores. Após esta leitura completa, você será capaz de:

- ✓ Entender a definição matemática de autovalores e autovetores.
- ✓ Entender a interpretação física de autovalores e autovetores no contexto de sistemas de engenharia que vibram ou oscilam.
- ✓ Aprender como implementar o método polinomial.
- ✓ Aprender como implementar o método da potência para avaliar o menor e maior autovalores e seus respectivos autovetores.
- ✓ Aprender como usar e interpretar a função `eig` do MATLAB.

VOCÊ TEM UM PROBLEMA

No início do Capítulo 8 utilizamos a segunda lei de Newton e balanços de força para prever as posições de equilíbrio de três saltadores de *bungee jumping* conectados por cordas. Admitido que as cordas se comportavam como molas ideais (isto é, seguiam a lei de Hooke), a solução em estado estacionário foi reduzida à solução de um sistema de equações algébricas lineares (lembre-se da Equação 8.1 e do Exemplo 8.2). Na mecânica, isso é chamado de problema de *estática*.

Agora, vamos analisar um problema de *dinâmica* envolvendo o mesmo sistema, ou seja, estudaremos o movimento dos saltadores como uma função do tempo. Para isso, suas condições iniciais (isto é, suas posições e velocidades) devem ser especificadas, o que pode ser feito de acordo com os valores de equilíbrio calculados no Exemplo 8.2. Se, então, definirmos as velocidades iniciais iguais a zero, nada aconteceria, porque o sistema estaria em equilíbrio.

Considerando que aqui o interesse é examinar a dinâmica do sistema, devemos definir valores para as condições iniciais que induzem movimento. Assim, embora as posições iniciais dos saltadores sejam definidas como os valores de equilíbrio, e a velocidade inicial do saltador do meio seja definida igual a zero, podemos definir as velocidades iniciais dos saltadores superior e inferior como alguns valores reconhecidamente extremos. Isto é, impomos uma velocidade para baixo de 200 m/s ao saltador 1 e uma velocidade para cima de 100 m/s ao saltador 3. (Dica de segurança: não tente fazer isso em casa!) Então, usamos o MATLAB para resolver a equação diferencial [Equação (8.1)] para gerar as posições e velocidades resultantes como uma função do tempo.¹

¹ Mostraremos como isso é feito quando abordarmos as equações diferenciais ordinárias na Parte 6.

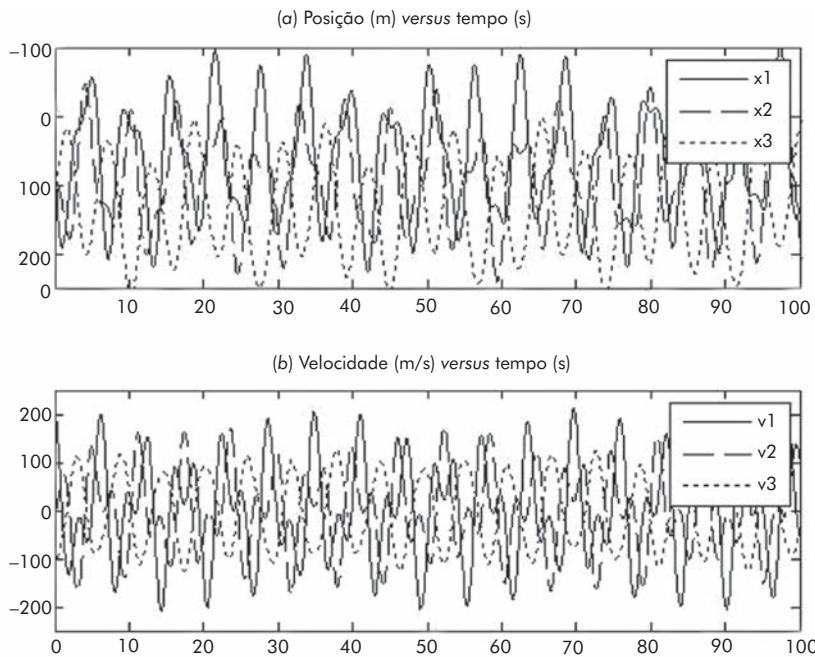


FIGURA 13.1 As (a) posições e (b) velocidades versus o tempo para o sistema de três saltadores de bungee jumping interconectados do Exemplo 8.2.

De acordo com a Figura 13.1, o resultado é que os saltadores oscilam descontroladamente. Como não há forças de atrito (por exemplo, sem força de arraste do ar ou de amortecimento da mola), eles balançam para cima e para baixo em torno de suas posições de equilíbrio de uma maneira persistente que, pelo menos visualmente, beira o caótico. Uma inspeção cuidadosa das trajetórias individuais sugere que pode haver algum padrão para as oscilações. Por exemplo, a distância entre os picos e vales poderia ser constante. Mas, quando visto como uma série temporal, é difícil perceber se existe qualquer coisa sistemática e previsível acontecendo.

Neste capítulo, lidaremos com uma abordagem que extrairá algo fundamental desse comportamento aparentemente caótico. Isso implica a determinação dos *autovalores*, ou *valores característicos*, para tais sistemas, o que envolve a formulação e resolução de sistemas de equações algébricas lineares de uma maneira que difere do que foi feito até este ponto. Para isso, vamos a princípio descrever exatamente o que se entende por autovalores de um ponto de vista matemático.

13.1 FUNDAMENTOS MATEMÁTICOS

Os Capítulos 8 a 12 abordaram métodos para resolução de conjuntos de equações algébricas lineares da forma geral

$$[A]\{x\} = \{b\} \quad (13.1)$$

Tais sistemas são chamados *não homogêneos* devido à presença do vetor $\{b\}$ no lado direito da igualdade. Se as equações que fazem parte de tal sistema forem linearmente independentes (isto é, tiverem um determinante não nulo), elas terão uma única solução. Ou seja, existe um único conjunto de valores de x que balanceará as equações. Como visto na Seção 9.1.1, para duas equações com duas incógnitas, a solução pode ser visualizada como a interseção de duas retas representadas pelas equações (lembre-se da Figura 9.1).

Em contraste, um sistema algébrico linear *homogêneo* tem o lado direito igual a zero:

$$[A]\{x\} = 0 \quad (13.2)$$

Essa equação sugere que a única solução possível seria o caso trivial para o qual todos os x 's = 0. Graficamente isso corresponderia a duas retas que se interceptam na origem (no zero).

Embora isso seja verdade, problemas de autovalores associados à engenharia normalmente são da seguinte forma geral:

$$[[A] - \lambda[I]]\{x\} = 0 \quad (13.3)$$

onde o parâmetro λ é chamado de *autovalor*. Assim, em vez de definir os x 's iguais a zero, pode-se determinar o valor de λ que leva o lado esquerdo para zero. Uma maneira de fazer isso tem base no fato de que o determinante da matriz deve ser igual a zero para que soluções não triviais sejam possíveis:

$$|[A] - \lambda[I]| = 0 \quad (13.4)$$

Expandindo o determinante, obtemos um polinômio em λ , que é chamado de *polinômio característico*. As raízes desse polinômio são as soluções para os autovalores.

Para compreender melhor esses conceitos, é útil examinar o caso de duas equações:

$$\begin{aligned} (a_{11} - \lambda)x_1 + a_{12}x_2 &= 0 \\ a_{21}x_1 + (a_{22} - \lambda)x_2 &= 0 \end{aligned} \quad (13.5)$$

Expandindo o determinante da matriz de coeficientes, tem-se

$$\begin{vmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{vmatrix} = \lambda^2 - (a_{11} + a_{22})\lambda - a_{12}a_{21} \quad (13.6)$$

que é o *polinômio característico*. A fórmula quadrática pode então ser usada para determinar os dois autovalores:

$$\lambda_1 = \frac{(a_{11} - a_{22})^2 \pm \sqrt{(a_{11} - a_{22})^2 - 4a_{12}a_{21}}}{2} \quad (13.7)$$

Esses são os valores que resolvem a Equação (13.5). Antes de prosseguir, vamos nos convencer de que essa abordagem (que, a propósito, é chamada de *método polinomial*) é correta.

EXEMPLO 13.1 O método polinomial

Use o método polinomial para calcular os autovalores para o seguinte sistema homogêneo:

$$\begin{aligned} (10 - \lambda)x_1 - 5x_2 &= 0 \\ -5x_1 + (10 - \lambda)x_2 &= 0 \end{aligned}$$

Solução. Antes de determinar a solução correta, vamos investigar o caso em que se tem um autovalor incorreto. Por exemplo, se $\lambda = 3$, a equação torna-se

$$\begin{aligned} 7x_1 - 5x_2 &= 0 \\ -5x_1 + 7x_2 &= 0 \end{aligned}$$

Traçando essas equações, têm-se duas retas que se interceptam na origem (Figura 13.2a). Assim, a solução é o caso trivial em que $x_1 = x_2 = 0$.

Para determinar os autovalores corretos, pode-se expandir o determinante para obter o polinômio característico:

$$\begin{vmatrix} 10 - \lambda & -5 \\ -5 & 10 - \lambda \end{vmatrix} = \lambda^2 - 20\lambda + 75$$

que pode ser resolvido para

$$\frac{\lambda_1 = 20 \pm \sqrt{20^2 - 4(1)75}}{2} = 15, 5$$

Portanto, os autovalores para esse sistema são 15 e 5.

Pode-se agora substituir cada um desses valores de volta no sistema e examinar o resultado. Para $\lambda_1 = 15$, obtém-se:

$$\begin{aligned} -5x_1 - 5x_2 &= 0 \\ -5x_1 - 5x_2 &= 0 \end{aligned}$$

Desse modo, um autovalor correto torna as duas equações idênticas (Figura 13.2b). Em essência, à medida que se move em direção a um autovalor correto, as duas retas giram até se encontrarem uma em cima da outra. Matematicamente, isso significa que há um número infinito de soluções. Porém, resolvendo cada uma das equações produz-se um resultado interessante: todas as soluções têm a propriedade de que $x_1 = -x_2$. Embora à primeira vista isso pareça trivial, na realidade é bastante interessante e mostra que a razão das incógnitas é uma constante. Esse resultado pode ser expresso na forma vetorial como

$$\{x\} = \left\{ \begin{array}{c} -1 \\ 1 \end{array} \right\}$$

que é chamado de *autovetor* correspondente ao autovalor $\lambda = 15$.

De modo análogo, substituindo o segundo autovalor, $\lambda_2 = 5$, tem-se:

$$\begin{aligned} 5x_1 - 5x_2 &= 0 \\ -5x_1 + 5x_2 &= 0 \end{aligned}$$

Novamente, o autovalor torna as duas equações idênticas (Figura 13.2b) e pode-se ver que a solução para esse caso corresponde a $x_1 = x_2$ e o autovetor é

$$\{x\} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\}$$

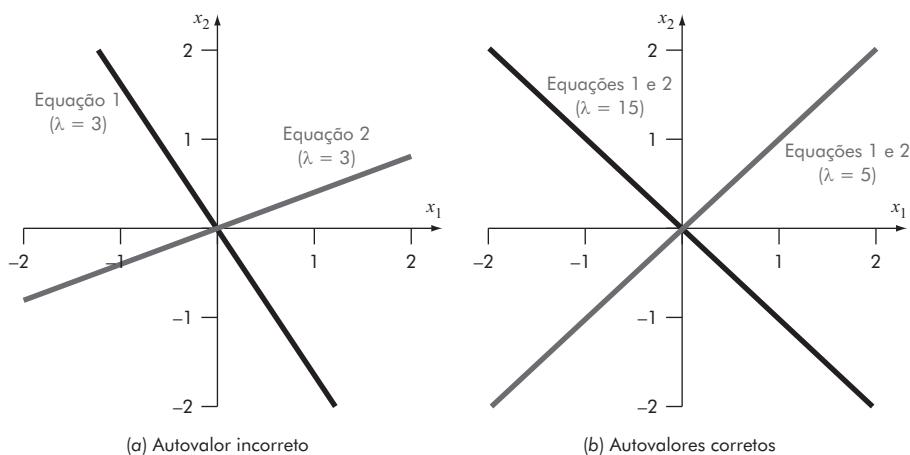


FIGURA 13.2 Gráfico de um sistema de duas equações lineares homogêneas do Exemplo 13.1. (a) Um autovalor incorreto ($\lambda = 3$) significa que as duas equações, que são identificadas como Equação 1 e 2 na figura, são traçadas como retas separadas e a única solução é o caso trivial ($x_1 = x_2 = 0$). (b) Em contraste, para os casos com os autovalores corretos ($\lambda = 5$ e 15), as equações caem uma em cima da outra.

Deve-se reconhecer que o MATLAB tem funções nativas para facilitar o método polinomial. Para o Exemplo 13.1, a função `poly` pode ser usada para gerar o polinômio característico como em

```
>> A = [10 -5; -5 10];
>> p = poly(A)

p =
    1     -20      75
```

Em seguida, a função `roots` pode ser empregada para calcular os autovalores:

```
>> d = roots(p)

d =
    15
     5
```

O exemplo anterior fornece o conhecimento matemático útil de que a solução de n equações homogêneas da forma da Equação (13.3) consiste em um conjunto de n autovalores e seus autovetores associados. Além disso, ele mostrou que os autovetores fornecem as razões das incógnitas representando a solução.

Na próxima seção mostraremos como essa informação tem utilidade na engenharia e na ciência voltando ao nosso problema físico de corpos oscilantes. No entanto, antes de fazer isso vamos analisar mais dois pontos matemáticos.

Primeiro, uma inspeção da Figura 13.2b indica que as retas representando cada solução do autovalor fazem um ângulo reto entre si, isto é, elas são *ortogonais*. Essa propriedade é verdadeira para matrizes simétricas com autovalores distintos.

Segundo, efetuando a multiplicação na Equação (13.3) e separando os termos, tem-se

$$[A]\{x\} = \lambda\{x\}$$

Quando vista dessa forma, pode-se notar que resolver para os autovalores e autovetores equivale a traduzir o conteúdo de informação de uma matriz $[A]$ em um escalar λ . Isso pode não parecer significativo para o sistema 2×2 que foi examinado, mas é bastante notável quando se considera que o tamanho de $[A]$ pode ser potencialmente muito maior.

13.2 FUNDAMENTOS FÍSICOS

O sistema massa-mola na Figura 13.3a é um contexto simples que ilustra como os autovalores ocorrem em um problema físico. Ele também ajudará a ilustrar alguns dos conceitos matemáticos introduzidos na seção anterior.

A fim de simplificar a análise, considere que cada massa não esteja sobre a ação de forças externas ou de amortecimento. Além disso, admita que cada mola tenha o mesmo comprimento natural l e a mesma constante da mola k . Por fim, considere que o deslocamento de cada mola seja medido em relação ao seu próprio sistema de coordenada local, com a origem na posição de equilíbrio da mola (Figura 13.3a). Sob essas hipóteses, a segunda lei de Newton pode ser usada para desenvolver um balanço de força para cada massa:

$$m_1 \frac{d^2x_1}{dt^2} = -kx_1 + k(x_2 - x_1) \quad (13.8a)$$

$$m_2 \frac{d^2x_2}{dt^2} = -k(x_2 - x_1) - kx_2 \quad (13.8b)$$

onde x_i é o deslocamento da massa i de sua posição de equilíbrio (Figura 13.3b).

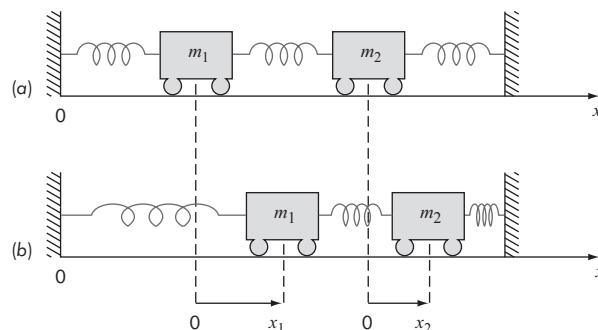


FIGURA 13.3 Sistema de duas massas e três molas, com rolamentos sem atrito, vibrando entre duas paredes fixas. A posição das massas pode ser referenciada em coordenadas locais, com as origens em suas respectivas posições de equilíbrio (a). Como em (b), a colocação das massas fora do equilíbrio cria forças nas molas que, depois de soltas, levam a oscilações das massas.

Da teoria da vibração, sabe-se que algumas das soluções da Equação (13.8) têm a forma

$$x_i = X_i \operatorname{sen}(\omega t) \quad (13.9)$$

onde X_i é a *amplitude* da oscilação da massa i (m) e ω é a *frequência angular* da oscilação (rad/s), que é igual a

$$\omega = \frac{2\pi}{T_p} \quad (13.10)$$

onde T_p é o *período*. Observe que o inverso do período é chamado de *frequência f* (ciclos/tempo). Se o tempo é medido em segundos, a unidade para f é o ciclos/s, que é chamada de *Hertz* (Hz).

A Equação (13.9) pode ser diferenciada duas vezes e substituída na Equação (13.8). Depois de agrupar os termos, o resultado é

$$\left(\frac{2k}{m_1} - \omega^2 \right) X_1 - \frac{k}{m_1} X_2 = 0 \quad (13.11a)$$

$$-\frac{k}{m_2} X_1 + \left(\frac{2k}{m_2} - \omega^2 \right) X_2 = 0 \quad (13.11b)$$

A comparação da Equação (13.11) com a Equação (13.3) indica que, nesse ponto, a solução foi reduzida a um problema de autovalor – onde, para esse caso, o autovalor é o quadrado da frequência angular. Para um sistema de dois graus de liberdade de tal como a Figura 13.3, existirão dois autovalores junto de seus autovetores. Como mostra o exemplo a seguir, estes últimos estabelecem uma relação única entre as variáveis.

EXEMPLO 13.2 Interpretação física dos autovalores e autovetores

Se $m_1 = m_2 = 40$ kg e $k = 200$ N/m, a Equação (13.11) é

$$(10 - \lambda)x_1 - 5x_2 = 0$$

$$-5x_1 + (10 - \lambda)x_2 = 0$$

Matematicamente, esse é o mesmo sistema que já foi resolvido com o método polinomial no Exemplo 13.1. Assim, os dois autovalores são $\omega^2 = 15$ e 5 s^{-2} , e os autovetores correspondentes são $X_1 = -X_2$ e $X_1 = X_2$. Interprete esses resultados com relação ao sistema massa-mola da Figura 13.3.

Solução. Esse exemplo fornece informação valiosa com relação ao comportamento do sistema na Figura 13.3. A princípio, ele mostra que o sistema tem dois modos de vibração principais com frequências angulares de $\omega = 3,873$ e $2,236$ radianos/s⁻¹, respectivamente. Esses valores também podem ser expressos como períodos (1,62 e 2,81 s, respectivamente) ou frequências (0,6164 e 0,3559 Hz, respectivamente).

Conforme apresentado na Seção 13.1, um único conjunto de valores não pode ser obtido para as amplitudes desconhecidas X . No entanto, suas razões são especificadas pelos autovetores. Assim, se o sistema está vibrando no primeiro modo, o primeiro autovetor nos

informa que a amplitude da segunda massa será igual, mas com sinal oposto à amplitude da primeira. Como na Figura 13.4a, as massas vibram separadas e, em seguida, juntas indefinidamente (como duas mãos batendo palmas a cada 1,62 s).

No segundo modo, o autovetor especifica que as duas massas têm amplitudes iguais em todos os instantes. Logo, como na Figura 13.4b, elas vibram para frente e para trás juntas a cada 2,81 s. Deve ser observado que a configuração das amplitudes fornece uma diretriz de como escolher os seus valores iniciais para alcançar movimentos puros em qualquer um dos dois modos. Qualquer outra configuração levará à superposição dos modos, a qual leva a comportamentos aparentemente caóticos de sistemas como os saltadores de *bungee jumping* na Figura 13.1. Mas, como esse exemplo deixa claro, há um comportamento sistemático básico que está incorporado pelos autovalores.

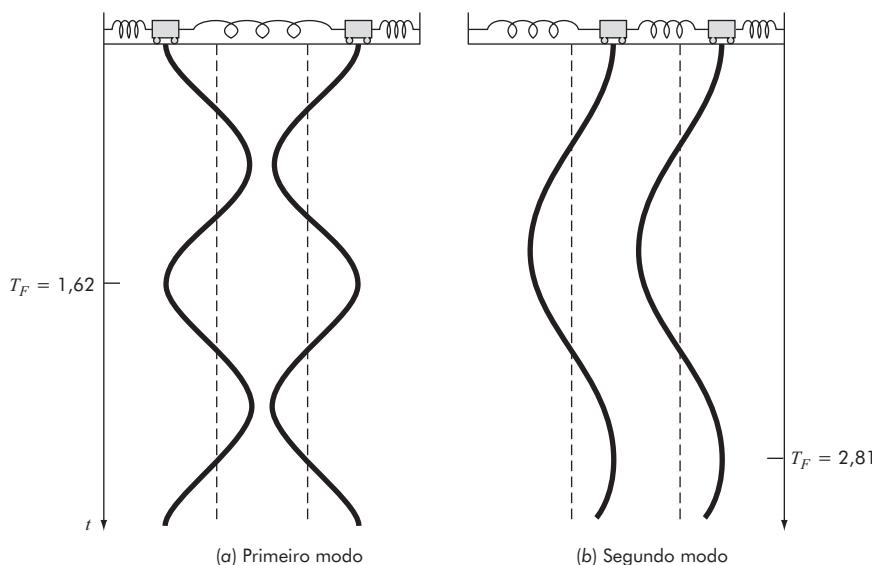


FIGURA 13.4 Os modos principais de vibração de duas massas iguais ligadas por três molas idênticas entre paredes fixas.

13.3 O MÉTODO DA POTÊNCIA

O método da potência é uma abordagem iterativa que pode ser usada para determinar o maior autovalor (ou autovalor dominante). Com pequenas modificações, também pode ser usado para determinar o menor autovalor. Ele tem a vantagem adicional de que o autovetor correspondente é obtido como um subproduto do método. Para implementar o método da potência, o sistema que está sendo analisado deve ser expresso na forma

$$[A]\{x\} = \lambda \{x\} \quad (13.12)$$

Como ilustra o exemplo seguinte, a Equação (13.12) forma a base para uma técnica de solução iterativa que, às vezes, fornece o maior autovalor e o seu autovetor associado.

EXEMPLO 13.3 Método da potência para o maior autovalor

Usando a mesma abordagem como na Seção 13.2, podemos derivar o seguinte conjunto de equações homogêneas para um sistema de três massas e quatro molas entre duas paredes fixas:

$$\begin{aligned} \left(\frac{2k}{m_1} - \omega^2\right)X_1 - \frac{k}{m_1}X_2 &= 0 \\ -\frac{k}{m_2}X_1 + \left(\frac{2k}{m_2} - \omega^2\right)X_2 - \frac{k}{m_2}X_3 &= 0 \\ -\frac{k}{m_3}X_2 + \left(\frac{k}{m_3} - \omega^2\right)X_3 &= 0 \end{aligned}$$

Se todas as massas são $m = 1 \text{ kg}$ e todas as constantes das molas são $k = 20 \text{ N/m}$, o sistema pode ser expresso na forma matricial da Equação (13.4) como

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} - \lambda[I] = 0$$

onde o autovalor λ é o quadrado da frequência angular ω^2 . Empregue o método da potência para determinar o maior autovalor e seu autovetor associado.

Solução. O sistema é primeiro escrito na forma da Equação (13.12):

$$\begin{aligned} 40X_1 - 20X_2 &= \lambda X_1 \\ -20X_1 + 40X_2 - 20X_3 &= \lambda X_2 \\ -20X_2 + 40X_3 &= \lambda X_3 \end{aligned}$$

Neste ponto, pode-se especificar valores iniciais dos X 's e usar o lado esquerdo para calcular um autovalor e autovetor. Uma boa primeira escolha é assumir que todos os X 's no lado esquerdo da equação são iguais a 1:

$$\begin{aligned} 40(1) - 20(1) &= 20 \\ -20(1) + 40(1) - 20(1) &= 0 \\ -20(1) + 40(1) &= 20 \end{aligned}$$

Em seguida, o lado direito é normalizado por 20 para tornar o maior elemento igual a 1:

$$\begin{Bmatrix} 20 \\ 0 \\ 20 \end{Bmatrix} = 20 \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}$$

Assim, o fator de normalização é nossa primeira estimativa do autovalor (20), e o autovetor correspondente é $\{1 \ 0 \ 1\}^T$. Essa iteração pode ser expressa de forma concisa na forma matricial como

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 20 \\ 0 \\ 20 \end{Bmatrix} = 20 \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}$$

A próxima iteração consiste em multiplicar a matriz pelo autovetor da última iteração, $\{1 \ 0 \ 1\}^T$, para fornecer

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 40 \\ -40 \\ 40 \end{Bmatrix} = 40 \begin{Bmatrix} 1 \\ -1 \\ 1 \end{Bmatrix}$$

Portanto, a estimativa do autovalor para a segunda iteração é 40, que pode ser empregada para determinar uma estimativa de erro:

$$|\varepsilon_a| = \left| \frac{40 - 20}{40} \right| \times 100\% = 50\%$$

O processo pode, então, ser repetido.

Terceira iteração:

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} 1 \\ -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 60 \\ -80 \\ 60 \end{Bmatrix} = -80 \begin{Bmatrix} -0,75 \\ 1 \\ -0,75 \end{Bmatrix}$$

onde $|\varepsilon_a| = 150\%$ (que é alto devido à mudança do sinal).

Quarta iteração:

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} -0,75 \\ 1 \\ -0,75 \end{Bmatrix} = \begin{Bmatrix} -50 \\ 70 \\ -50 \end{Bmatrix} = 70 \begin{Bmatrix} -0,71429 \\ 1 \\ -0,71429 \end{Bmatrix}$$

onde $|\varepsilon_a| = 214\%$ (outra mudança do sinal).

Quinta iteração:

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix} \begin{Bmatrix} -0,71429 \\ 1 \\ -0,71429 \end{Bmatrix} = \begin{Bmatrix} -48,51714 \\ 68,51714 \\ -48,51714 \end{Bmatrix} = 68,51714 \begin{Bmatrix} -0,70833 \\ 1 \\ -0,70833 \end{Bmatrix}$$

onde $|\varepsilon_a| = 2,08\%$.

Dessa forma, o autovalor está convergindo. Depois de várias iterações, ele se estabiliza em um valor de 68,28427 com um autovetor correspondente de $\{-0,707107 \ 1 \ -0,707107\}^T$.

Observe que existem alguns casos nos quais o método da potência convergirá para o segundo maior autovalor em vez do maior. James, Smith e Wolford (1985) forneceram uma ilustração de tal caso. Outros casos especiais são discutidos em Fadeev e Fadeeva (1963).

Além disso, existem casos em que estamos interessados na determinação do menor autovalor. Isso pode ser feito aplicando o método da potência à matriz inversa de $[A]$. Nesse caso, o método da potência convergirá para o maior valor de $1/\lambda$ – ou seja, para o menor valor de λ . Uma aplicação para encontrar o menor autovalor será deixada como problema.

Por fim, depois de encontrar o maior autovalor, é possível determinar o próximo mais alto substituindo a matriz original por uma que inclua apenas os autovalores restantes. O processo de remover o maior autovalor conhecido é chamado *deflação*.

É importante ressaltar que, embora o método da potência possa ser usado para localizar valores intermediários, métodos melhores estão disponíveis para casos em que é necessário determinar todos os autovalores, como descrito na próxima seção. Assim, o método da potência é usado principalmente quando se quer localizar o maior ou o menor autovalor.

13.4 FUNÇÃO DO MATLAB: eig

Como poderia ser esperado, o MATLAB tem recursos poderosos e robustos para cálculo de autovalores e autovetores. A função `eig`, que é usada para esse propósito, pode ser empregada para gerar um vetor dos autovalores como em

```
>> e = eig(A)
```

onde `e` é um vetor contendo os autovalores de uma matriz quadrada `A`. Alternativamente, ela pode ser chamada como

```
>> [V, D] = eig(A)
```

onde `D` é uma matriz diagonal dos autovalores e `V` é uma matriz cheia cujas colunas são os autovetores correspondentes.

Deve-se observar que o MATLAB escalona os autovetores dividindo-os por suas normas euclidianas. Assim, como mostra o exemplo a seguir, embora seus valores possam ser diferentes daqueles calculados com, por exemplo, o método polinomial, a razão de seus elementos será idêntica.

EXEMPLO 13.4 Autovalores e autovetores com o MATLAB

Use o MATLAB para determinar todos os autovalores e autovetores para o sistema descrito no Exemplo 13.3.

Solução. Lembre-se de que a matriz a ser analisada é

$$\begin{bmatrix} 40 & -20 & 0 \\ -20 & 40 & -20 \\ 0 & -20 & 40 \end{bmatrix}$$

Pode-se entrar com a matriz como

```
>> A = [40 -20 0; -20 40 -20; 0 -20 40];
```

Se quisermos apenas os autovalores, podemos entrar com o seguinte comando

```
>> e = eig(A)
e =
11.7157
40.0000
68.2843
```

Observe que o maior autovalor (68,2843) é consistente com o valor previamente determinado com o método da potência no Exemplo 13.3.

Se quisermos tanto os autovalores quanto os autovetores, podemos entrar com o seguinte comando

```
>> [v, d] = eig(A)

v =
    0.5000    -0.7071   -0.5000
    0.7071    -0.0000    0.7071
    0.5000    0.7071   -0.5000

d =
    11.7157         0         0
        0    40.0000         0
        0         0    68.2843
```

Embora os resultados estejam em escalas diferentes, o autovetor correspondente ao maior autovalor $\{-0.5 \ 0.7071 \ -0.5\}^T$ é consistente com o valor previamente determinado com o método da potência no Exemplo 13.3: $\{-0.707107 \ 1 \ -0.707107\}^T$. Isso pode ser demonstrado dividindo o autovetor do método da potência por sua norma euclidiana:

```
>> vpotencia=[-0.7071 1 -0.7071]';
>> vMATLAB=vpotencia/norm(vpotencia)

vMATLAB =
    -0.5000
    0.7071
    -0.5000
```

Assim, embora os valores dos elementos difiram, suas razões são idênticas.

► 13.5 ESTUDO DE CASO AUTOVALORES E TERREMOTOS

Contextualização. Engenheiros e cientistas usam modelos massa-mola para desenvolver uma percepção da dinâmica de estruturas sob a influência de distúrbios tais como terremotos. A Figura 13.5 mostra tal representação para um edifício de três andares. Cada massa do andar é representada por m_i , e cada rigidez do andar é representada por k_i , para $i = 1 \text{ a } 3$.

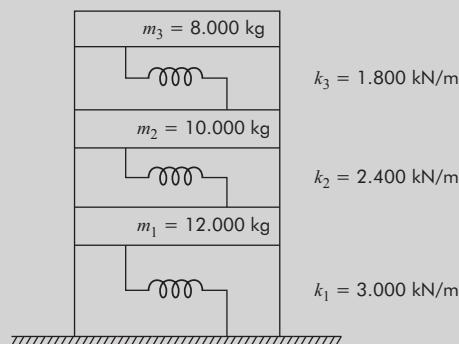


FIGURA 13.5 Edifício de três andares modelado como um sistema massa-mola.



Nesse caso, a análise é limitada ao movimento horizontal da estrutura à medida que ela está sujeita ao movimento horizontal da base decorrente de terremotos. Utilizando a mesma abordagem desenvolvida na Seção 13.2, balanços de forças dinâmicas podem ser desenvolvidos para esse sistema como

$$\begin{aligned} \left(\frac{k_1 + k_2}{m_1} - \omega_n^2 \right) X_1 - \frac{k_2}{m_1} X_2 &= 0 \\ - \frac{k_2}{m_2} X_1 + \left(\frac{k_2 + k_3}{m_2} - \omega_n^2 \right) X_2 - \frac{k_3}{m_2} X_3 &= 0 \\ - \frac{k_3}{m_3} X_2 + \left(\frac{k_3}{m_3} - \omega_n^2 \right) X_3 &= 0 \end{aligned}$$

onde X_i representa a translação horizontal do andar (m) e ω_n é a frequência natural ou frequência de ressonância (radianos/s). A frequência de ressonância pode ser expressa em Hertz (ciclos/s) dividindo-a por 2π radianos/ciclo.

Use o MATLAB para determinar os autovalores e autovetores para esse sistema. Represente graficamente os modos de vibração para a estrutura exibindo as amplitudes *versus* a altura para cada um dos autovetores. Normalize as amplitudes de modo que a translação do terceiro andar seja 1.

Solução. Os parâmetros podem ser substituídos nos balanços de força para fornecer

$$\begin{aligned} (450 - \omega_n^2)X_1 - 200X_2 &= 0 \\ - 240X_1 + (420 - \omega_n^2)X_2 - 180X_3 &= 0 \\ - 225X_2 + (225 - \omega_n^2)X_3 &= 0 \end{aligned}$$

Uma sessão do MATLAB pode ser conduzida para avaliar os autovalores e autovetores como

```
>> A=[450 -200 0;-240 420 -180;0 -225 225];
>> [v, d]=eig(A)

v =
-0.5879    -0.6344    0.2913
 0.7307    -0.3506    0.5725
-0.3471     0.6890    0.7664

d =
 698.5982         0         0
      0    339.4779         0
      0         0   56.9239
```

Portanto, os autovalores são 698,6, 339,5 e 56,92 e as frequências de ressonância em Hz são

```
>> wn=sqrt(diag(d))/2/pi
wn =
 4.2066    2.9324    1.2008
```



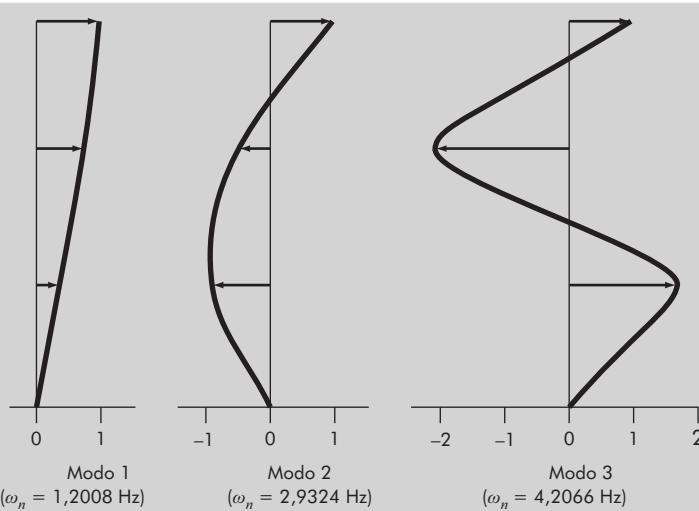


FIGURA 13.6 Os três modos principais de vibração do edifício de três andares.

Os autovetores correspondentes são (normalizados de modo que a amplitude para o terceiro andar seja 1)

$$\begin{Bmatrix} 1,6934 \\ -2,1049 \\ 1 \end{Bmatrix} \begin{Bmatrix} -0,9207 \\ -0,5088 \\ 1 \end{Bmatrix} \begin{Bmatrix} 0,3801 \\ 0,7470 \\ 1 \end{Bmatrix}$$

Um gráfico pode ser feito mostrando os três modos (Figura 13.6). Observe que os modos foram ordenados da menor para a maior frequência natural, como é de costume em engenharia estrutural.

As frequências naturais e as formas dos modos são características de estruturas em termos de suas tendências a entrar em ressonância nessas frequências. O conteúdo de frequência de um terremoto geralmente tem a maior parte da energia entre 0 e 20 Hz e é influenciado pela magnitude do terremoto, distância epicentral e outros fatores. Em vez de uma única frequência, eles contêm um espectro de frequências com amplitudes variadas. Os edifícios são mais receptivos à vibração em seus modos de vibração mais baixos devido às suas formas deformadas mais simples e porque a energia requerida para deformação é menor nos modos inferiores. Quando essas amplitudes coincidem com as frequências naturais de edifícios, grandes respostas dinâmicas são induzidas, criando grandes tensões e deformações nas vigas, colunas e fundações da estrutura. Com base em análises como a deste estudo de caso, engenheiros estruturais podem projetar edifícios de forma mais sensata para resistir a terremotos com um bom fator de segurança.

►►PROBLEMAS

13.1 Repita o Exemplo 13.1, porém para três massas com m 's = 40 kg e k 's = 240 N/m. Faça um gráfico como o da Figura 13.4 para identificar os modos principais de vibração.

13.2 Use o método da potência para determinar o maior autovalor e o autovetor correspondente para

$$\begin{bmatrix} 2 - \lambda & 8 & 10 \\ 8 & 4 - \lambda & 5 \\ 10 & 5 & 7 - \lambda \end{bmatrix}$$

13.3 Utilize o método da potência para determinar o menor autovalor e o autovetor correspondente para o sistema do Problema 13.2.

13.4 Deduza o conjunto de equações diferenciais para o sistema de três massas e quatro molas (Figura P13.4) que descreve seu movimento no tempo. Escreva as três equações diferenciais na forma matricial

$$\{\text{Vetor aceleração}\} + [\text{matriz } k/m]\{\text{vetor de deslocamento } x\} = 0$$

Observe que cada equação foi dividida pela massa. Determine os autovalores e as frequências naturais para os seguintes valores das massas e das constantes das molas: $k_1 = k_4 = 15 \text{ N/m}$, $k_2 = k_3 = 35 \text{ N/m}$ e $m_1 = m_2 = m_3 = 1,5 \text{ kg}$.

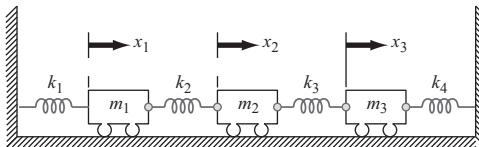


FIGURA P13.4

13.5 Considere o sistema massa-mola apresentado na Figura P13.5. As frequências para as vibrações das massas podem ser determinadas por meio dos autovalores e aplicando $M\ddot{x} + kx = 0$, o que produz

$$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{bmatrix} + \begin{bmatrix} 2k & -k & -k \\ -k & 2k & -k \\ -k & -k & 2k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Tentando $x = x_0 e^{i\omega t}$ como solução, obtemos a seguinte matriz

$$\begin{bmatrix} 2k - m_1\omega^2 & -k & -k \\ -k & 2k - m_2\omega^2 & -k \\ -k & -k & 2k - m_3\omega^2 \end{bmatrix} \begin{bmatrix} x_{01} \\ x_{02} \\ x_{03} \end{bmatrix} e^{i\omega t} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Use o comando `eig` do MATLAB para determinar os autovalores da matriz $k - m\omega^2$ anterior. Em

seguida, use esses autovalores para determinar as frequências (ω). Faça $m_1 = m_2 = m_3 = 1 \text{ kg}$ e $k = 2 \text{ N/m}$.

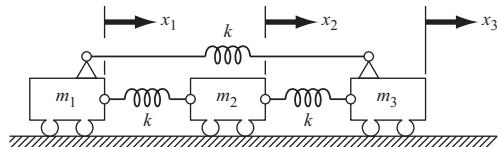


FIGURA P13.5

13.6 Como mostra a Figura P13.6, um circuito LC pode ser modelado pelo seguinte sistema de equações diferenciais:

$$\begin{aligned} L_1 \frac{d^2 i_1}{dt^2} + \frac{1}{C_1}(i_1 - i_2) &= 0 \\ L_2 \frac{d^2 i_2}{dt^2} + \frac{1}{C_2}(i_2 - i_3) - \frac{1}{C_1}(i_1 - i_2) &= 0 \\ L_3 \frac{d^2 i_3}{dt^2} + \frac{1}{C_3}i_3 - \frac{1}{C_2}(i_2 - i_3) &= 0 \end{aligned}$$

onde L é a indutância (H), t é o tempo (s), i é a corrente (A) e C é a capacitância (F). Considerando que a solução é da forma $i_j = I_j \sin(\omega t)$, determine os autovalores e autovetores para esse sistema com $L = 1 \text{ H}$ e $C = 0,25 \text{ F}$. Desenhe a rede ilustrando como as correntes oscilam em seus modos principais.

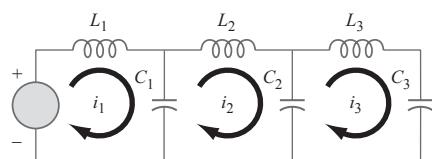


FIGURA P13.6

13.7 Repita o Problema 13.6, mas com apenas duas malhas. Isto é, omita a malha da corrente i_3 . Desenhe a rede ilustrando como as correntes oscilam em seus modos principais.

13.8 Repita o problema na Seção 13.5, mas não inclua o terceiro andar.

13.9 Repita o problema na Seção 13.5, mas adicione um quarto andar com $m_4 = 6.000 \text{ kg}$ e $k_4 = 1.200 \text{ kN/m}$.

13.10 A curvatura de uma coluna delgada submetida a uma carga axial P (Figura P13.10) pode ser modelada por

$$\frac{d^2 y}{dx^2} + p^2 y = 0$$

onde

$$P^2 = \frac{P}{EI}$$

sendo E o módulo de elasticidade e I o momento de inércia da seção transversal em relação a seu eixo neutro.

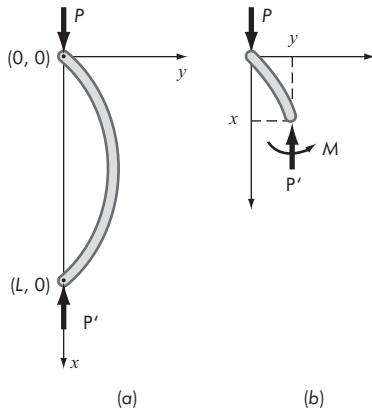


FIGURA P13.10 (a) Uma coluna delgada.
(b) Um diagrama de corpo livre de uma coluna.

Esse modelo pode ser convertido em um problema de autovalores substituindo a segunda derivada por uma aproximação por diferenças finitas centradas para fornecer

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{\Delta x^2} + p^2 y_i = 0$$

onde i é um nó localizado em uma posição ao longo do interior da coluna e Δx é o espaçamento entre os nós. Essa equação pode ser expressa como

$$y_{i-1} - (2 - \Delta x^2 p^2)y_i + y_{i+1} = 0$$

Escrevendo essa equação para uma série de nós interiores ao longo do eixo da coluna, obtém-se um sistema de equações homogêneas. Por exemplo, se a coluna é dividida em cinco segmentos (isto é, quatro nós interiores), o resultado é

Uma coluna de madeira carregada axialmente tem as seguintes características: $E = 10 \times 10^9$ Pa, $I = 1,25 \times 10^{-5}$ m⁴ e $L = 3$ m. Para a representação com cinco segmentos e quatro nós interiores:

- (a) Implemente o método polinomial com o MATLAB para determinar os autovalores para esse sistema.
- (b) Use a função `eig` do MATLAB para determinar os autovalores e autovetores.
- (c) Use o método da potência para determinar o maior autovalor e seu autovetor correspondente.

13.11 Um sistema de duas equações diferenciais ordinárias lineares homogêneas com coeficientes constantes pode ser escrito como

$$\begin{aligned} \frac{dy_1}{dt} &= -5y_1 + 3y_2, & y_1(0) &= 50 \\ \frac{dy_2}{dt} &= 100y_1 - 301y_2, & y_2(0) &= 100 \end{aligned}$$

Um estudante de equações diferenciais saberá que a solução para tais equações tem a forma

$$y_i = c_i e^{\lambda_i t}$$

onde c e λ são constantes a serem determinadas. A substituição dessa solução e de sua derivada nas equações originais converte o sistema em um problema de autovalores. Os autovalores e autovetores correspondentes podem ser usados para deduzir a solução geral para as equações diferenciais. Por exemplo, para o caso de duas equações, a solução geral pode ser escrita em termos de vetores como

$$\{y\} = c_1 \{v_1\} e^{\lambda_1 t} + c_2 \{v_2\} e^{\lambda_2 t}$$

onde $\{v_i\}$ é o autovetor associado ao i -ésimo autovalor (λ_i) e os c 's são coeficientes desconhecidos que podem ser determinados com as condições iniciais.

- (a) Converta o sistema em um problema de autovalores.

$$\left[\begin{array}{cccc} (2 - \Delta x^2 p^2) & -1 & 0 & 0 \\ -1 & (2 - \Delta x^2 p^2) & -1 & 0 \\ 0 & -1 & (2 - \Delta x^2 p^2) & -1 \\ 0 & 0 & -1 & (2 - \Delta x^2 p^2) \end{array} \right] \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{Bmatrix} = 0$$

- (b) Use o MATLAB para determinar os autovalores e os autovetores.
- (c) Empregue os resultados de (b) e as condições iniciais para determinar a solução geral.
- (d) Desenvolva um gráfico no MATLAB da solução para $t = 0$ a 1.

13.12 A água flui entre os Grandes Lagos da América do Norte como detalhado na Figura P13.12. Com base em balanços de massas, as seguintes equações diferenciais podem ser escritas para as concentrações em cada um dos lagos para um poluente que decai com cinética de primeira ordem:

$$\frac{dc_1}{dt} = -(0,0056 + k)c_1$$

$$\frac{dc_2}{dt} = -(0,01 + k)c_2$$

$$\frac{dc_3}{dt} = 0,01902c_1 + 0,01387c_2 - (0,047 + k)c_3$$

$$\frac{dc_4}{dt} = 0,33597c_3 - (0,376 + k)c_4$$

$$\frac{dc_5}{dt} = 0,11364c_4 - (0,133 + k)c_5$$

onde k é a taxa de decaimento de primeira ordem (/ano), que é igual $0,69315/(meia\text{-vida})$. Observe que as constantes em cada uma das equações representam o fluxo entre os lagos. Devido a testes de armas nucleares na atmosfera, as concentrações de estrôncio-90 (^{90}Sr) nos cinco lagos em 1963 eram aproximadamente $\{c\} = \{17,7 \quad 30,5 \quad 43,9 \quad 136,3 \quad 30,1\}^T$ em unidades de Bq/m^3 . Considerando que nenhum ^{90}Sr adicional entrou no sistema depois disso, use o MATLAB e a abordagem esboçada no Problema 13.11 para calcular as concentrações em cada um dos lagos de 1963 até 2010. Observe que o ^{90}Sr tem uma meia-vida de 28,8 anos.

13.13 Desenvolva uma função no MATLAB para determinar o maior autovalor e seu autovetor associado com o método da potência. Teste seu programa duplicando o Exemplo 13.3 e, em seguida, use-o para resolver o Problema 13.2.

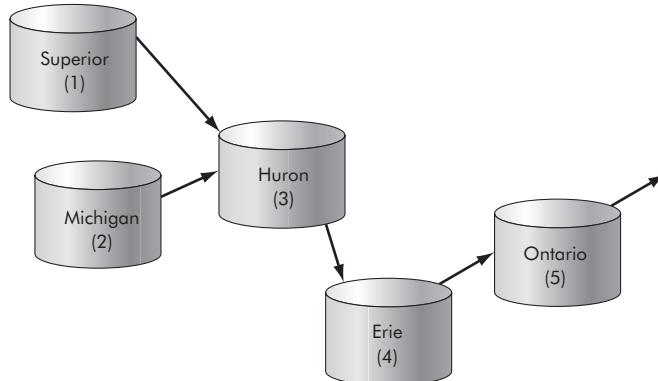
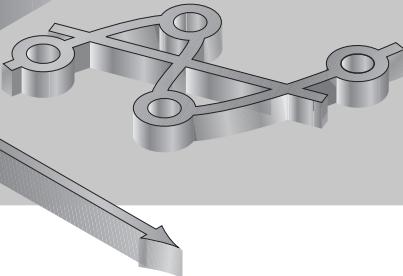


FIGURA P13.12 Os Grandes Lagos da América do Norte. As setas indicam como a água flui entre os lagos.



Parte 4

Ajuste de curvas

4.1 VISÃO GERAL

O que é ajuste de curvas?

Em geral, os dados são fornecidos em um conjunto discreto de valores entre um contínuo de possibilidades. Entretanto, pode ser necessário fazer estimativas em pontos que estão entre os valores discretos. Os Capítulos 14 a 18 descrevem técnicas para ajustar curvas a esses dados, com o objetivo de obter estimativas intermediárias. Além disso, também pode ser necessária uma versão simplificada de uma função complicada. Uma forma de fazer isso é calcular valores da função em alguns valores discretos no intervalo de interesse. Então, uma função mais simples pode ser desenvolvida para ajustar esses dados. Ambas as aplicações são conhecidas como *ajuste de curvas*.

Há duas abordagens gerais para o ajuste de curvas, as quais são distinguidas entre si com base na quantidade de erros associada com os dados. Primeiro, quando os dados exibirem grau significativo de erro ou “ruído”, a estratégia será encontrar uma única curva que represente a tendência geral dos dados. Como cada ponto individual pode estar incorreto, não será feito nenhum esforço para passar a curva por todos os pontos. Em vez disso, a curva é escolhida para seguir o padrão dos pontos considerados como um grupo. Uma abordagem dessa natureza é chamada de *regressão por mínimos quadrados* (Figura PT4.1a).

Segundo, quando os dados forem muito precisos, a abordagem básica é ajustar uma curva ou uma série de curvas que passam diretamente por cada um dos pontos. Esses dados frequentemente se originam de tabelas. Os valores da densidade da água ou da capacidade térmica dos gases como uma função da temperatura são exemplos disso. A estimativa de valores entre pontos discretos bem conhecidos é chamada *interpolação* (Figura PT4.1b e c).

Ajuste de curvas no contexto da engenharia e da ciência. Sua primeira exposição ao ajuste de curvas pode ter sido determinar valores intermediários a partir de dados tabulados – por exemplo, a partir de tabelas de juros em engenharia econômica, ou de tabelas de vapor em termodinâmica. No decorrer de sua carreira, você terá oportunidades frequentes de fazer estimativas de valores intermediários a partir de tais tabelas.

Embora muitas das propriedades da engenharia amplamente usadas sejam tabuladas, há muitas outras que não estão disponíveis nessa forma conveniente. Casos especiais e novos contextos de problemas frequentemente exigirão a medição de seus próprios dados e o desenvolvimento de sua relação de previsão. Em geral, são encontrados dois tipos de aplicações nas quais dados experimentais são ajustados: análise de tendência e teste de hipótese.

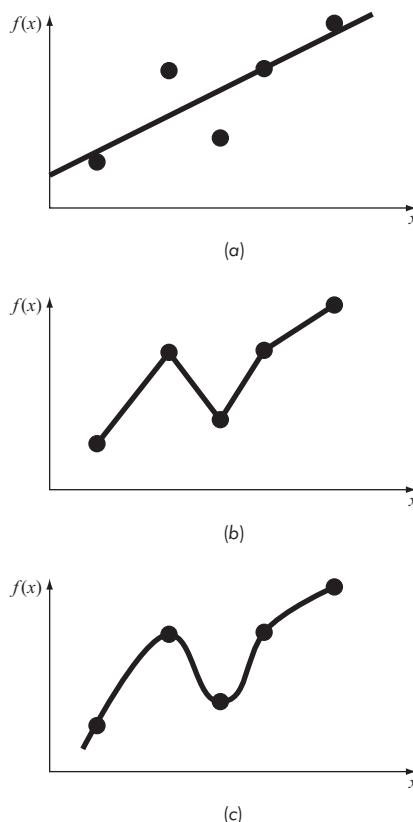


FIGURA PT4.1 Três tentativas de ajustar a “melhor” curva aos cinco pontos dados:
 (a) Regressão por mínimos quadrados, (b) interpolação linear e
 (c) interpolação curvilínea.

A *análise de tendência* é o processo que usa o padrão dos dados para fazer previsões. Nos casos em que os dados foram medidos com alta precisão, pode-se usar interpolação polinomial. Os dados imprecisos são, em geral, analisados com regressão por mínimos quadrados.

Esta análise pode ser usada também para estimar valores da variável dependente. Isso pode envolver extrapolação além dos limites dos dados observados ou interpolação dentro do intervalo dos dados. Todos os campos da engenharia e da ciência geralmente envolvem problemas desse tipo.

Uma segunda aplicação do ajuste de curvas a dados experimentais é no *teste de hipótese*. Aqui, um modelo matemático já existente é comparado com os dados medidos. Se os coeficientes do modelo forem desconhecidos, pode ser necessária a determinação de valores que melhor ajustem os dados observados. Entretanto, se estimativas para os coeficientes dos modelos já estiverem disponíveis, pode ser apropriado comparar os valores previstos pelo modelo com os valores observados para testar a adequação do modelo. Em geral, modelos alternativos são comparados e o “melhor” é escolhido com base nas observações empíricas.

Além dessas aplicações em engenharia e ciência, o ajuste de curvas é importante em outros métodos numéricos, como integração e solução aproximada de equações diferenciais. Por fim, as técnicas de ajuste de curvas podem ser usadas para encontrar funções simples que approximem funções complicadas.

4.2 ORGANIZAÇÃO DA PARTE 4

Após uma breve revisão de estatística, o Capítulo 14 é dedicado à *regressão linear*; ou seja, como determinar a “melhor” reta por meio de um conjunto de pontos dados que apresenta incertezas. Além de discutir como calcular a inclinação e a interseção dessa reta com o eixo y , serão apresentados métodos quantitativos e visuais para avaliar a validade dos resultados; também será descrita a *geração de números aleatórios* assim como diversas abordagens para a linearização de equações não lineares.

O Capítulo 15 inicia com discussões breves de regressão linear múltipla e polinomial. A *regressão polinomial* trata do desenvolvimento do melhor ajuste de parábolas, cúbicas ou polinômios de ordem mais alta. Isso é seguido por uma descrição da *regressão linear múltipla*, que se aplica aos casos em que a variável independente y é uma função linear de duas ou mais variáveis independentes x_1, x_2, \dots, x_n . Essa abordagem tem utilidade especial no tratamento de dados experimentais quando a variável de interesse depende de diversos fatores diferentes.

Depois da regressão múltipla, será ilustrado como a regressão polinomial e múltipla são ambas subconjuntos de um *modelo linear geral de mínimos quadrados*. Entre outras coisas, isso vai permitir introduzir uma representação matricial concisa da regressão e discutir suas propriedades estatísticas gerais. Por fim, as últimas seções do Capítulo 15 são dedicadas à *regressão não linear*. Essa abordagem se destina ao cálculo de ajuste por mínimos quadrados de uma equação não linear aos dados.

O Capítulo 16 trata da *análise de Fourier*, que envolve o ajuste de funções periódicas aos dados. Nossa ênfase será na *transformada rápida de Fourier* ou *FFT*. Esse método, que é imediatamente implementado com o MATLAB, tem várias aplicações de engenharia, que vão desde análise de vibração de estruturas até processamento de sinais.

No Capítulo 17, será descrita uma técnica alternativa de ajuste de curvas, chamada *interpolação*. Como discutido anteriormente, a interpolação é usada para fazer estimativas de valores intermediários entre pontos dados precisamente e polinômios serão derivados com esse propósito; serão introduzidos os conceitos básicos de interpolação polinomial usando retas e parábolas para ligar pontos. A seguir, será introduzido um procedimento geral para ajustar um polinômio de grau n e serão apresentadas duas formas para expressar esses polinômios na forma de uma equação. A primeira, chamada *polinômio interpolador de Newton*, é preferível quando o grau apropriado do polinômio for desconhecido. A segunda, chamada de *polinômio interpolador de Lagrange*, é preferível quando o grau adequado do polinômio for conhecido antecipadamente.

Por fim, o Capítulo 18 apresenta uma técnica alternativa para o ajuste de dados precisos, chamada de *interpolação por splines*, que ajusta polinômios aos dados, por partes, e é particularmente adequada para o ajuste de dados que sejam suaves no geral, mas apresentem variações locais abruptas. O capítulo finaliza com uma visão geral de como a interpolação por partes é implementada no MATLAB.

14

Regressão linear

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é introduzir como a regressão por mínimos quadrados pode ser usada para ajustar uma reta a dados de medição. Após esta leitura completa, você será capaz de:

- ✓ Identificar algumas estatísticas descritivas básicas e a distribuição normal.
- ✓ Aprender como calcular a inclinação e a interseção (com o eixo y) de uma reta de melhor ajuste aplicando a regressão linear.
- ✓ Aprender como gerar números aleatórios com o MATLAB e como eles podem ser empregados para simulações de Monte Carlo.
- ✓ Aprender como calcular e compreender o significado do coeficiente de determinação e do erro-padrão da estimativa.
- ✓ Entender como utilizar transformações para linearizar equações não lineares de modo que elas possam ser ajustadas com regressão linear.
- ✓ Aprender como implementar regressão linear com o MATLAB.

VOCÊ TEM UM PROBLEMA

No Capítulo 1, observamos que um corpo em queda livre, tal como um saltador de *bungee jumping*, está sujeito a uma força para cima em decorrência da resistência do ar. Como uma primeira aproximação, consideramos que essa força era proporcional ao quadrado da velocidade como em

$$F_U = c_d v^2 \quad (14.1)$$

onde F_U é a força para cima de resistência do ar ($N = \text{kg m/s}^2$), c_d é um coeficiente de arraste (kg/m) e v é a velocidade (m/s).

Expressões como a Equação (14.1) são provenientes do campo da mecânica dos fluidos. Embora essas relações sejam, em parte, derivadas da teoria, os experimentos desempenham um papel fundamental em sua formulação. Um deles está detalhado na Figura 14.1. Um indivíduo está suspenso em um túnel de vento e a força é medida para vários níveis de velocidade do vento. Os resultados que poderiam ser obtidos em um experimento deste tipo estão listados na Tabela 14.1.

A relação pode ser visualizada traçando-se a força *versus* a velocidade. Como na Figura 14.2, várias características dessa relação merecem ser mencionadas. Primeiro, os pontos indicam que a força aumenta à medida que a velocidade aumenta. Segundo, os pontos não aumentam suavemente, mas apresentam dispersão bastante significativa, particularmente em velocidades mais elevadas. Por fim, embora possa não ser óbvio, a relação entre força e velocidade pode não ser linear. Essa conclusão torna-se mais evidente se considerarmos que a força é zero para velocidade nula.

TABELA 14.1 Dados experimentais para força (N) e velocidade (m/s) de um experimento de túnel de vento.

$v, \text{ m/s}$	10	20	30	40	50	60	70	80
$F, \text{ N}$	25	70	380	550	610	1.220	830	1.450

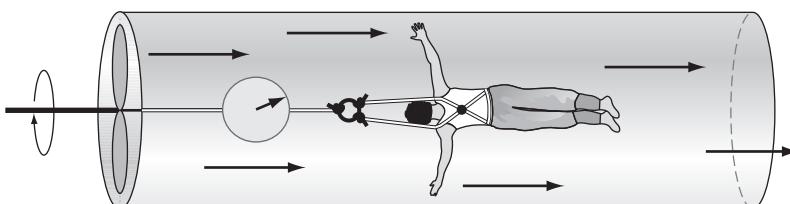


FIGURA 14.1 Experimento do túnel de vento para medir como a força de resistência do ar depende da velocidade.

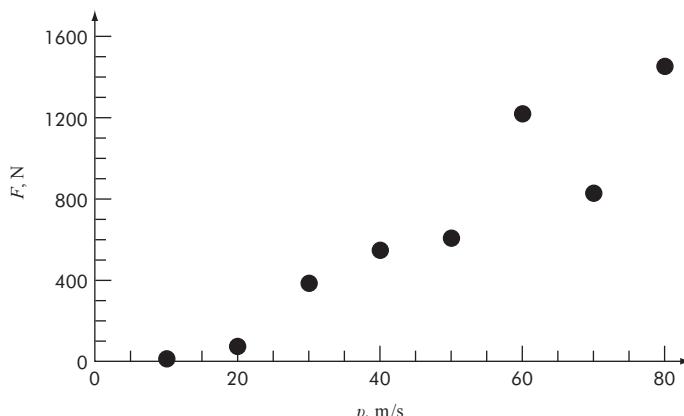


FIGURA 14.2 Gráfico de força versus velocidade do vento para um corpo suspenso em um túnel de vento.

Nos Capítulos 14 e 15, exploraremos como ajustar a “melhor” reta ou curva a esses dados. Ao fazer isso, ilustraremos como relações como a Equação (14.1) surgem de dados experimentais.

14.1 REVISÃO DE ESTATÍSTICA

Antes de descrever a regressão por mínimos quadrados, revisaremos alguns conceitos básicos do campo da estatística. A revisão inclui conceitos de média, desvio-padrão, soma dos quadrados residual e distribuição normal. Além disso, descreve-se como estatísticas descritivas simples e distribuições podem ser geradas no MATLAB. Se você estiver familiarizado com esses assuntos, sinta-se à vontade para desconsiderar as páginas seguintes e seguir diretamente para a Seção 14.2. Se você não estiver familiarizado com esses conceitos ou necessita de uma revisão, o material a seguir é elaborado como uma breve introdução.

14.1.1 Estatística descritiva

Considere que, no decorrer de um estudo de engenharia, tenham sido feitas diversas medidas de uma quantidade particular. Por exemplo, a Tabela 14.2 contém 24 leituras do coeficiente de expansão térmica de um aço estrutural. À primeira vista, os dados fornecem uma quantidade limitada de informações – ou seja, os valores variam de um mínimo de 6,395 a um máximo de 6,775. Uma informação adicional pode ser obtida resumindo-se os dados em uma ou mais estatísticas bem escolhidas que transportem tanta informação quanto possível sobre as características específicas desse conjunto de dados. Tais estatísticas descritivas são, em geral, escolhidas para representar (1) a localização do centro da distribuição de dados e (2) o grau de dispersão do conjunto de dados.

Medida da Localização. A medida mais comum da tendência central é a *média aritmética* (\bar{y}) de uma amostra, que é definida como a soma dos pontos individuais (y_i) dividida pelo número de pontos (n), ou

$$\bar{y} = \frac{\sum y_i}{n} \quad (14.2)$$

onde a somatória (e todas as somatórias posteriores nesta seção) vai de $i = 1$ até n .

Há várias alternativas para a média aritmética. A *mediana* é o ponto médio de um grupo de dados. Ela é calculada colocando-se primeiro os dados em ordem crescente. Se o número de medições é ímpar, a mediana é o valor do meio (central); se o número é par, a mediana é a média aritmética dos dois valores centrais. A mediana é às vezes chamada de *percentil 50*.

A *moda* é o valor que ocorre com maior frequência. O conceito frequentemente tem utilidade direta apenas quando se lida com dados discretos ou grosseiramente arredondados. Para variáveis contínuas como os dados na Tabela 14.2, o conceito não é muito prático. Por exemplo, existem quatro modas para esses dados: 6,555; 6,625; 6,655 e 6,715, sendo que todos esses valores ocorrem duas vezes. Se os números não tivessem sido arredondados para 3 dígitos decimais, seria improvável que qualquer um dos valores teria sequer se repetido duas vezes. No entanto, se dados contínuos são agrupados em intervalos equidistantes, isso pode ser uma estatística informativa. Retornaremos ao conceito de moda quando descrevermos histogramas mais à frente nesta seção.

Medidas de Dispersão. A medida de dispersão mais simples é a *amplitude* que corresponde à diferença entre o maior e o menor valor. Embora a amplitude da amostra seja certamente fácil de determinar, ela não é considerada uma medida muito confiável, pois é altamente sensível ao tamanho da amostra e muito sensível aos valores extremos.

TABELA 14.2 Medições do coeficiente de expansão térmica de um aço estrutural.

6,495	6,595	6,615	6,635	6,485	6,555
6,665	6,505	6,435	6,625	6,715	6,655
6,755	6,625	6,715	6,575	6,655	6,605
6,565	6,515	6,555	6,395	6,775	6,685

A medida de dispersão mais comum para uma amostra é o *desvio-padrão* (s_y) em torno da média,

$$s_y = \sqrt{\frac{S_t}{n - 1}} \quad (14.3)$$

onde S_t é a soma total dos quadrados dos resíduos entre os pontos dados e a média, ou

$$S_t = \sum(y_i - \bar{y})^2 \quad (14.4)$$

Portanto, se as medidas individuais estiverem amplamente dispersas em torno da média, S_t (e, consequentemente, s_y) será grande. Se elas estiverem bem agrupadas, o desvio-padrão será pequeno. A dispersão também pode ser representada pelo quadrado do desvio-padrão, o qual é chamado de *variância*:

$$s_y^2 = \frac{\sum(y_i - \bar{y})^2}{n - 1} \quad (14.5)$$

Observe que o denominador em ambas as Equações (14.3) e (14.5) é $n - 1$. A quantidade $n - 1$ é chamada de *graus de liberdade*. Então, dizemos que S_t e s_y são baseados em $n - 1$ graus de liberdade. Essa nomenclatura deve-se ao fato de a soma das quantidades nas quais S_t é baseada (isto é, $\bar{y} - y_1, \bar{y} - y_2, \dots, \bar{y} - y_n$) ser zero. Consequentemente, se \bar{y} for conhecido e $n - 1$ valores forem especificados, o valor remanescente está fixado. Portanto, diz-se que apenas $n - 1$ valores são livremente determinados. Outra justificativa para dividir por $n - 1$ é o fato de que não existe tal coisa como a dispersão de um único ponto dado. Nos casos em que $n = 1$, as Equações (14.3) e (14.5) fornecem resultados infinitos, sem significado.

Deve ser observado que uma fórmula alternativa, mais conveniente, está disponível para calcular a variância:

$$s_y^2 = \frac{\sum y_i^2 - (\sum y_i)^2 / n}{n - 1} \quad (14.6)$$

Essa versão não exige o cálculo prévio de \bar{y} e fornece um resultado idêntico ao da Equação (14.5).

Uma estatística final que tem utilidade na quantificação da dispersão dos dados é o *coeficiente de variação* (c.v.) e é a razão do desvio-padrão pela média. Como tal, o c.v. fornece uma medida normalizada da dispersão. Ele é, em geral, multiplicado por 100, de modo que possa ser expresso como uma porcentagem:

$$\text{c.v.} = \frac{s_y}{\bar{y}} \times 100\% \quad (14.7)$$

EXEMPLO 14.1 Estatísticas simples de uma amostra

Calcule a mediana, a média, a variância, o desvio-padrão e o coeficiente de variação para os dados da Tabela 14.2.

Solução. Os dados podem ser reunidos na forma de tabela, e as somas necessárias calculadas como na Tabela 14.3.

A média pode ser calculada como [Equação (14.2)],

$$\bar{y} = \frac{158,4}{24} = 6,6$$

Como há um número par de valores, a mediana é calculada como a média aritmética dos dois valores centrais: $(6,605 + 6,615)/2 = 6,61$.

TABELA 14.3 Dados e somas para cálculo de estatísticas descritivas simples para os coeficientes de expansão térmica da Tabela 14.2.

<i>i</i>	<i>y_i</i>	$(y_i - \bar{y})^2$	<i>y_i²</i>
1	6,395	0,04203	40,896
2	6,435	0,02723	41,409
3	6,485	0,01323	42,055
4	6,495	0,01103	42,185
5	6,505	0,00903	42,315
6	6,515	0,00723	42,445
7	6,555	0,00203	42,968
8	6,555	0,00203	42,968
9	6,565	0,00123	43,099
10	6,575	0,00063	43,231
11	6,595	0,00003	43,494
12	6,605	0,00002	43,626
13	6,615	0,00022	43,758
14	6,625	0,00062	43,891
15	6,625	0,00062	43,891
16	6,635	0,00122	44,023
17	6,655	0,00302	44,289
18	6,655	0,00302	44,289
19	6,665	0,00422	44,422
20	6,685	0,00722	44,689
21	6,715	0,01322	45,091
22	6,715	0,01322	45,091
23	6,755	0,02402	45,630
24	6,775	0,03062	45,901
Σ	158,400	0,21700	1045,657

Como na Tabela 14.3, a soma dos quadrados dos resíduos é 0,217000, que pode ser usada para calcular o desvio-padrão [Equação (14.3)]:

$$s_y = \sqrt{\frac{0,217000}{24 - 1}} = 0,097133$$

a variância [Equação (14.15)]:

$$s_y^2 = (0,097133)^2 = 0,009435$$

e o coeficiente de variação [Equação (14.17)]:

$$\text{c.v.} = \frac{0,097133}{6,6} \times 100\% = 1,47\%$$

A validade da Equação (14.6) também pode ser verificada calculando

$$s_y^2 = \frac{1045,657 - (158,400)^2 / 24}{24 - 1} = 0,009435$$

14.1.2 A distribuição normal

Outra característica relacionada com esta discussão é a distribuição de dados – ou seja, a forma como os dados estão dispersos em torno da média. Um histograma fornece uma representação visual simples da distribuição. Um *histograma* é construído classificando as medidas em intervalos (também chamados de intervalos de classes ou células, ou, ainda, *bins*). As unidades de medida são marcadas na abscissa e as frequências de ocorrência em cada intervalo são marcadas na ordenada.

Como um exemplo, um histograma pode ser criado para os dados da Tabela 14.2. O resultado (Figura 14.3) sugere que em sua maioria os dados estão agrupados próximo do valor médio de 6,6. Observe também que, agora que os dados foram agrupados, pode-se ver que o intervalo com a maioria dos valores está entre 6,6 a 6,64. Embora possa ser verificado que a moda é o ponto médio desse intervalo, 6,62, é mais comum reportar o intervalo mais frequente como o *intervalo de classe modal*.

Se tivéssemos um conjunto muito grande de dados, o histograma em geral poderia ser aproximado por uma curva suave. A curva simétrica, em forma de sino, superposta na Figura 14.3 é uma dessas formas características – a *distribuição normal*. Se fossem dadas medidas adicionais suficientes, o histograma para esse caso particular poderia algumas vezes se aproximar da distribuição normal.

Os conceitos de média, desvio-padrão, soma dos quadrados dos resíduos e distribuição normal têm todos grande relevância para a engenharia e a ciência. Um exemplo muito simples é sua utilização para determinar a confiança que pode ser associada a uma medida particular. Se uma quantidade for normalmente distribuída, o intervalo definido por $\bar{y} - s_y$ a $\bar{y} + s_y$ irá conter aproximadamente 68% do total das medidas. Analogamente, o intervalo definido por $\bar{y} - 2s_y$ a $\bar{y} + 2s_y$ irá conter aproximadamente 95%.

Por exemplo, para os dados da Tabela 14.2, foi calculado no Exemplo 14.1 que $\bar{y} = 6,6$ e $s_y = 0,097133$. Com base em nossa análise, pode-se, a princípio, afirmar-se que aproximadamente 95% das leituras deveriam estar entre 6,405734 e 6,794266. Se for apresentada uma medição no valor de 7,35, pode-se suspeitar que a medida esteja errada, já que está tão fora dos limites citados anteriormente.

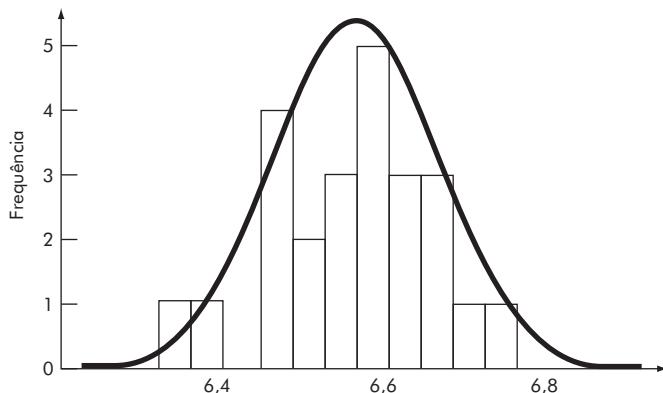


FIGURA 14.3 Histograma usado para descrever uma distribuição de dados. Conforme o número de pontos dados aumenta, o histograma poderá aproximar-se de uma curva suave, em forma de sino, chamada de distribuição normal.

14.1.3 Estatísticas descritivas no MATLAB

O MATLAB padrão tem várias funções para calcular estatísticas descritivas.¹ Por exemplo, a média aritmética é calculada como `mean(x)`. Se `x` é um vetor, a função retorna a média dos valores do vetor. Se for uma matriz, a função retorna um vetor linha contendo a média aritmética de cada coluna de `x`. A sessão do MATLAB a seguir é o resultado da utilização da função `mean` e outras funções estatísticas para analisar um vetor coluna `s` que contém os dados da Tabela 14.2.

```
>> format short g
>> mean(s), median(s), mode(s)

ans =
6.6

ans =
6.61

ans =
6.555

>> min(s), max(s)

ans =
6.395

ans =
6.775

>> range=max(s)-min(s)
```

¹ O MATLAB também oferece um Toolbox de Estatística que fornece uma ampla gama de tarefas estatísticas comuns, desde a geração de números aleatórios, até o ajuste de curvas para projeto de experimentos e controle estatístico de processos.

```

range =
0.38
>> var(s), std(s)

ans =
0.0094348

ans =
0.097133

```

Esses resultados estão consistentes com aqueles obtidos anteriormente no Exemplo 14.1. Observe que, embora existam quatro valores que ocorrem duas vezes, a função mode (que calcula a moda) retorna apenas o primeiro dos valores: 6,555.

O MATLAB também pode ser usado para gerar um histograma baseado na função hist. A função hist tem a sintaxe:

```
[n, x] = hist(y, x)
```

onde n é o número de elementos em cada intervalo (bin), x é um vetor especificando o ponto médio de cada intervalo e y é o vetor sendo analisado. Para os dados da Tabela 14.2, o resultado é

```

>> [n, x] = hist(s)

n =
    1     1     3     1     4     3     5     2     2     2

x =
    6,414 6,452 6,49 6,528 6,566 6,604 6,642 6,68 6,718 6,756

```

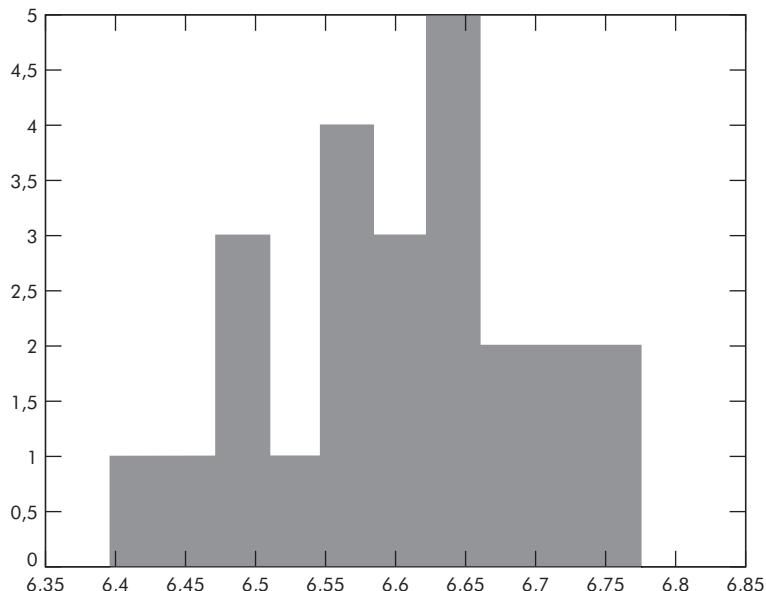


FIGURA 14.4 Histograma gerado com a função hist do MATLAB.

O histograma resultante apresentado na Figura 14.4 é similar àquele gerado manualmente na Figura 14.3. Observe que todos os argumentos e as saídas, com exceção de y , são opcionais. Por exemplo, `hist(y)` sem argumentos de saída simplesmente produz um histograma de barras com 10 intervalos determinados automaticamente baseados na faixa de valores em y .

14.2 NÚMEROS ALEATÓRIOS E SIMULAÇÃO

Nesta seção, descreveremos duas funções do MATLAB que podem ser usadas para produzir uma sequência de números aleatórios. A primeira (`rand`) gera números que estão uniformemente distribuídos, e a segunda (`randn`) gera números que têm uma distribuição normal.

14.2.1 Função do MATLAB: `rand`

Essa função gera uma sequência de números que estão uniformemente distribuídos entre 0 e 1. Uma representação simples de sua sintaxe é

$$r = \text{rand}(m, n)$$

onde r é uma matriz m por n de números aleatórios. A seguinte fórmula pode então ser usada para gerar uma distribuição uniforme em outro intervalo

$$\text{runiforme} = \text{inferior} + (\text{superior} - \text{inferior}) * \text{rand}(m, n)$$

onde `inferior` é o limite inferior e `superior` é o limite superior do intervalo.

EXEMPLO 14.2 Gerando valores aleatórios uniformes de arraste

Se a velocidade inicial é zero, a velocidade para baixo do saltador de *bungee jumping* em queda livre pode ser prevista com a seguinte solução analítica [Equação (1.9)]:

$$v = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right)$$

Considere que $g = 9,81 \text{ m/s}^2$ e $m = 68,1 \text{ kg}$, mas c_d não é conhecido de forma precisa. Por exemplo, você poderia saber que c_d varia uniformemente entre 0,225 e 0,275 (ou seja, $\pm 10\%$ em torno de um valor médio de 0,25 kg/m). Use a função `rand` para gerar 1.000 valores aleatórios de c_d uniformemente distribuídos e, em seguida, empregue esses valores juntamente com a solução analítica para calcular a distribuição resultante de velocidades em $t = 4 \text{ s}$.

Solução. Antes de gerar os números aleatórios, pode-se calcular a velocidade média:

$$v_{\text{med}} = \sqrt{\frac{9,81(68,1)}{0,25}} \tanh\left(\sqrt{\frac{9,81(0,25)}{68,1}} 4\right) = 33,1118 \frac{\text{m}}{\text{s}}$$

A amplitude (extremidades inferior e superior) também pode ser gerada:

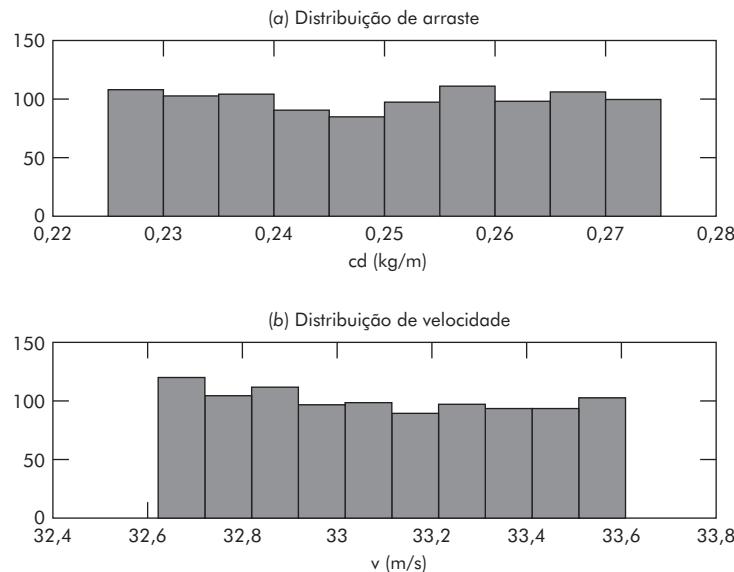


FIGURA 14.5 Histogramas de (a) coeficientes de arraste uniformemente distribuídos e (b) distribuição de velocidade resultante.

$$v_{\text{inferior}} = \sqrt{\frac{9,81(68,1)}{0,275}} \tanh\left(\sqrt{\frac{9,81(0,275)}{68,1}} 4\right) = 32,6223 \frac{\text{m}}{\text{s}}$$

$$v_{\text{superior}} = \sqrt{\frac{9,81(68,1)}{0,225}} \tanh\left(\sqrt{\frac{9,81(0,225)}{68,1}} 4\right) = 33,6198 \frac{\text{m}}{\text{s}}$$

Assim, pode-se ver que a velocidade varia por

$$\Delta v = \frac{33,6198 - 32,6223}{2(33,1118)} \times 100\% = 1,5063\%$$

O programa a seguir gera os valores aleatórios para c_d , com os cálculos de média, desvio-padrão, variação percentual e um histograma:

```
clc,format short g
n=1000;t=4;m=68.1;g=9.81;
cd=0.25;cdmin=cd-0.025;cdmax=cd+0.025
r=rand(n,1);
cdrand=cdmin+(cdmax-cdmin)*r;
media_cd=mean(cdrand),desvio_cd=std(cdrand)
Deltacd=(max(cdrand)-min(cdrand))/media_cd/2*100.
subplot(2,1,1)
hist(cdrand),title('(a) Distribuição de arraste')
xlabel('cd (kg/m)')
```

Os resultados são

```
media_cd =
    0.25018
desvio_cd =
    0.014528
Deltacd =
    9.9762
```

Esses resultados, assim como o histograma (Figura 14.5a), indicam que a função `rand` produziu 1.000 valores uniformemente distribuídos com o valor médio e intervalo desejados. Os valores podem ser empregados com a solução analítica para calcular a distribuição de velocidades resultante em $t = 4$ s.

```
vrand=sqrt(g*m./cdrvand).*tanh(sqrt(g*cdrvand/m)*t);
media_v=mean(vrand)
Deltav=(max(vrand)-min(vrand))/media_v/2*100.
subplot(2,1,2)
hist(vrand),title('(b) Distribuição de velocidade')
xlabel('v (m/s)')
```

Os resultados são:

```
media_v =
    33.1151
Deltav =
    1.5048
```

Esses resultados, assim como o histograma (Figura 14.5b), estão em excelente conformidade com os cálculos efetuados manualmente.

O exemplo anterior é formalmente conhecido como uma *simulação de Monte Carlo*. O termo, que é uma referência ao cassino Monte Carlo, em Mônaco, foi usado pela primeira vez por físicos que trabalhavam no projeto de armas nucleares nos anos 1940. Embora o método forneca resultados intuitivos para esse exemplo simples, há casos em que essas simulações de computador produzem resultados surpreendentes e fornecem informações que, de outra forma, seriam impossíveis de determinar. A abordagem só é viável devido à capacidade do computador de implementar cálculos repetitivos e tediosos de maneira eficiente.

14.2.2 Função do MATLAB: `randn`

Essa função gera uma sequência de números que estão normalmente distribuídos com uma média de 0 e um desvio-padrão de 1. Uma representação simples de sua sintaxe é

```
r = randn(m, n)
```

onde `r` é uma matriz `m` por `n` de números aleatórios. A seguinte fórmula pode então ser usada para gerar uma distribuição normal com uma média (`mn`) e desvio-padrão (`s`) diferentes,

```
rnornal = mn + s * rand(m, n)
```

EXEMPLO 14.3 Gerando valores aleatórios de arraste normalmente distribuídos

Analise o mesmo caso como no Exemplo 14.2, mas em vez de empregar uma distribuição uniforme, gere coeficientes de arraste normalmente distribuídos com uma média de 0,25 e um desvio-padrão de 0,01443.

Solução. O programa a seguir gera os valores aleatórios para c_d , com os cálculos de média, desvio-padrão, coeficiente de variação (expresso em %) e um histograma:

```
clc,format short g
n=1000;t=4;m=68.1;g=9.81;
cd=0.25;
desvio=0.01443;
r=randn(n,1);
cdrand=cd+desvio*r;
media_cd=mean(cdrand),desvio_cd=std(cdrand)
cvcd=desvio_cd/media_cd*100.
subplot(2,1,1)
hist(cdrand),title('(a) Distribuição de arraste')
xlabel('cd (kg/m)')
```

Os resultados são:

```
media_cd =
    0.24988
desvio_cd =
    0.014465
cvcd =
    5.7887
```

Esses resultados, assim como o histograma (Figura 14.6a), indicam que a função `randn` produziu 1.000 valores normalmente distribuídos com o valor médio, desvio-padrão e coeficiente de variação desejados. Os valores podem ser empregados junto à solução analítica para calcular a distribuição de velocidades resultante em $t = 4$ s.

```
vrand=sqrt(g*m./cdrand).*tanh(sqrt(g*cdrand/m)*t);
media_v=mean(vrand),desvio_v=std(vrand)
cvv=desvio_v/media_v*100.
subplot(2,1,2)
hist(vrand),title('(b) Distribuição de velocidade')
xlabel('v (m/s)')
```

Os resultados são:

```
media_v =
    33.117
desvio_v =
    0.28839
cvv =
    0.8708
```

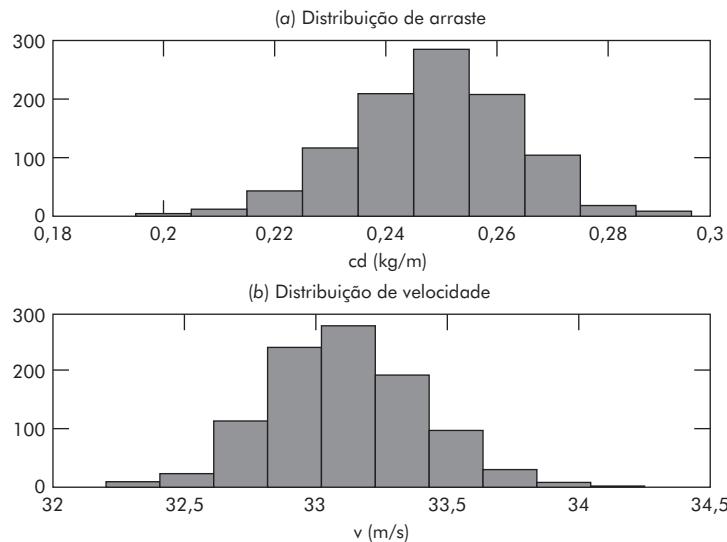


FIGURA 14.6 Histogramas de (a) coeficientes de arraste normalmente distribuídos e (b) distribuição de velocidade resultante.

Esses resultados, assim como o histograma (Figura 14.6b), indicam que as velocidades também estão normalmente distribuídas com uma média que está próxima do valor que seria calculado utilizando a média e a solução analítica. Além disso, calculamos o desvio-padrão associado, que corresponde a um coeficiente de variação de $\pm 0,8708\%$.

Embora simples, os exemplos anteriores ilustram como números aleatórios podem ser facilmente gerados dentro do MATLAB. Aplicações adicionais serão exploradas nos problemas no fim do capítulo.

14.3 REGRESSÃO LINEAR POR MÍNIMOS QUADRADOS

Quando um erro substancial estiver associado aos dados, a melhor estratégia de ajuste de curva é determinar uma função aproximada que ajuste a forma ou a tendência geral dos dados sem necessariamente passar pelos pontos individuais. Um meio de fazer isso é inspecionar visualmente os pontos marcados e então esboçar a “melhor” reta pelos pontos. Embora essa abordagem visual pareça atrativa do ponto de vista do bom-senso e seja válida para cálculos informais, é deficiente, porque é arbitrária; ou seja, a menos que os pontos definam perfeitamente uma reta (em tal caso, a interpolação seria apropriada), analistas diferentes desenhariam retas diferentes.

A fim de evitar essa subjetividade, deve ser desenvolvido algum critério para estabelecer uma base para o ajuste. Uma forma de fazê-lo é determinar a curva que minimize a discrepância entre os dados e os pontos da curva. Para isso, deve-se primeiro determinar a discrepância. O exemplo mais simples é o ajuste de uma reta a um conjunto de pares de observação: $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. A expressão matemática do ajuste por uma reta é

$$y = a_0 + a_1 x + e \quad (14.8)$$

onde a_0 e a_1 são coeficientes que representam a interseção com o eixo y e a inclinação, respectivamente, e e é o erro ou *resíduo* entre o modelo e a observação, o qual pode ser representado, depois de se reorganizar a Equação (14.8), por

$$e = y - a_0 - a_1x \quad (14.9)$$

Portanto, o erro ou resíduo é a discrepância entre o valor verdadeiro de y e o valor aproximado, $a_0 + a_1x$, previsto pela equação linear.

14.3.1 Critério para um “melhor” ajuste

Uma estratégia para ajustar uma “melhor” reta pelos dados é minimizar o valor absoluto da soma dos erros residuais para todos os dados disponíveis, como em

$$\sum_{i=1}^n e_i = \sum_{i=1}^n (y_i - a_0 - a_1x_i) \quad (14.10)$$

onde n é o número total de pontos. Entretanto, esse é um critério inadequado, como ilustra a Figura 14.7a, a qual descreve o ajuste de uma reta a dois pontos. É óbvio que o melhor ajuste é a reta ligando os pontos. Entretanto, qualquer reta passando pelo ponto médio do segmento que liga os pontos (exceto uma reta perfeitamente vertical) resulta em um valor mínimo da Equação (14.10) igual a zero, pois os erros positivo e negativo se cancelam.

O efeito dos sinais pode ser removido minimizando-se a soma dos valores absolutos das discrepâncias, como em

$$\sum_{i=1}^n |e_i| = \sum_{i=1}^n |y_i - a_0 - a_1x_i| \quad (14.11)$$

A Figura 14.7b ilustra por que esse critério também é inadequado. Nos quatro pontos mostrados, qualquer reta caindo dentro das retas tracejadas minimizaria a soma dos valores absolutos. Portanto, ele também não forneceria um melhor ajuste único.

Uma terceira estratégia para ajustar a melhor reta é o critério *minimax*. Nessa técnica, é escolhida a reta que minimiza a distância máxima que um ponto individual tenha da reta. Como descreve a Figura 14.7c, essa estratégia não é adequada para a regressão porque ela permite uma influência indevida a um ponto “discrepante” – ou seja, a um único ponto com um erro grande. Deve ser observado que o princípio minimax às vezes é adequado para ajustar uma função simples a função complicada (Carnahan, Luther e Wilkes, 1969).

Uma estratégia que supera as deficiências das abordagens anteriores é minimizar a soma dos quadrados dos resíduos:

$$S_r = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - a_0 - a_1x_i)^2 \quad (14.12)$$

Esse critério, que é chamado de *mínimos quadrados*, tem diversas vantagens, incluindo o fato de que ele fornece uma única reta para um dado conjunto de dados. Antes de se discutir essas propriedades, será apresentada uma técnica para determinar os valores de a_0 e a_1 que minimizam a Equação (14.12).

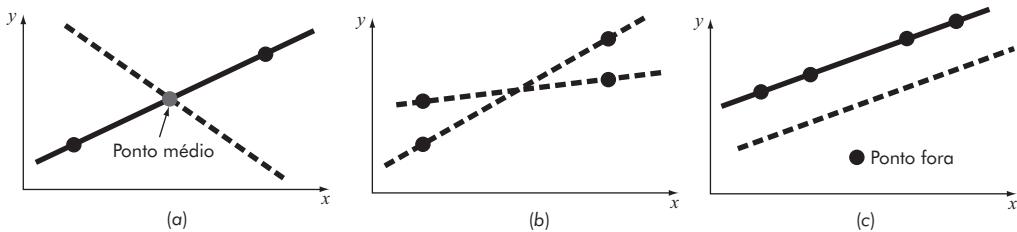


FIGURA 14.7 Exemplos de alguns critérios para “melhor ajuste” que são inadequados para a regressão: (a) minimizar a soma dos resíduos, (b) minimizar a soma dos valores absolutos dos resíduos, e (c) minimizar o erro máximo por qualquer ponto individual.

14.3.2 Ajuste por mínimos quadrados de uma reta

Para determinar os valores de a_0 e a_1 , a Equação (14.12) é derivada com relação a cada coeficiente desconhecido:

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum [(y_i - a_0 - a_1 x_i)x_i]$$

Observe que simplificamos os símbolos de somatória; a menos que haja menção contrária, todas as somatórias irão de $i = 1$ a n . Igualando essas derivadas a zero, será obtido um S_r mínimo. Se isso for feito, as equações podem ser expressas como

$$0 = \sum y_i - \sum a_0 - \sum a_1 x_i$$

$$0 = \sum x_i y_i - \sum a_0 x_i - \sum a_1 x_i^2$$

Agora, percebendo que $\sum a_0 = n a_0$, pode-se expressar essas equações como um conjunto de duas equações lineares simultâneas em duas variáveis (a_0 e a_1):

$$n a_0 + \left(\sum x_i \right) a_1 = \sum y_i \quad (14.13)$$

$$\left(\sum x_i \right) a_0 + \left(\sum x_i^2 \right) a_1 = \sum x_i y_i \quad (14.14)$$

Essas são as chamadas *equações normais*. Elas podem ser resolvidas simultaneamente para

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \quad (14.15)$$

Esse resultado pode, então, ser usado junto com a Equação (14.13) para determinar

$$a_0 = \bar{y} - a_1 \bar{x} \quad (14.16)$$

onde \bar{y} e \bar{x} são as médias de y e x , respectivamente.

EXEMPLO 14.4 Regressão linear

Ajuste uma reta aos valores na Tabela 14.1.

Solução. Nessa aplicação, a força é a variável dependente (y) e a velocidade é a variável independente (x). Os dados podem ser organizados na forma de uma tabela e as somas necessárias calculadas como na Tabela 14.4.

TABELA 14.4 Dados e somas necessárias para calcular a reta de melhor ajuste para os dados da Tabela 14.1.

i	x_i	y_i	x_i^2	$x_i y_i$
1	10	25	100	250
2	20	70	400	1.400
3	30	380	900	11.400
4	40	550	1.600	22.000
5	50	610	2.500	30.500
6	60	1.220	3.600	73.200
7	70	830	4.900	58.100
8	80	1.450	6.400	116.000
Σ	360	5.135	20.400	312.850

As médias podem ser calculadas como

$$\bar{x} = \frac{360}{8} = 45 \quad \bar{y} = \frac{5.135}{8} = 641,875$$

A inclinação e a interseção com o eixo y podem então ser calculadas com as Equações (14.15) e (14.16) como

$$a_1 = \frac{8(312.850) - 360(5.135)}{8(20.400) - (360)^2} = 19,47024$$

$$a_0 = 641,875 - 19,47024(45) = -234,2857$$

Utilizando força e velocidade em lugar de y e x , o ajuste por mínimos quadrados é

$$F = -234,2857 + 19,47024v$$

A reta com os dados é mostrada na Figura 14.8.

Observe que, embora a reta se ajuste bem aos dados, a interseção com o eixo das velocidades (eixo x) significa que a equação prevê forças negativas fisicamente inconsistentes no caso de velocidades mais baixas. Na Seção 14.4, mostraremos como transformações podem ser empregadas para derivar uma reta de melhor ajuste alternativa que é fisicamente mais consistente.

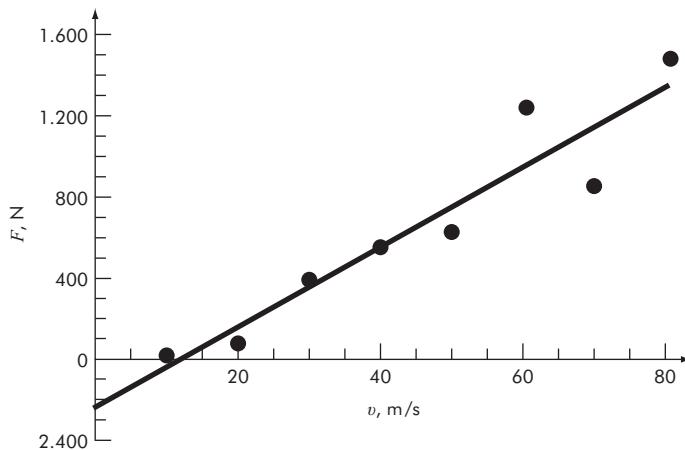


FIGURA 14.8 Ajuste por mínimos quadrados de uma reta aos dados da Tabela 14.1.

14.3.3 Quantificação do erro da regressão linear

Qualquer outra reta que não a calculada no Exemplo 14.4 resulta em uma soma maior dos quadrados dos resíduos; portanto, a reta é única e, em termos do critério escolhido por nós, é a “melhor” reta pelos pontos. Diversas propriedades adicionais desse ajuste podem ser elucidadas examinando-se com mais detalhes a forma como os resíduos foram calculados. Lembre-se de que a soma dos quadrados é definida como [Equação (14.12)]

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1 x_i)^2 \quad (14.17)$$

Observe a similaridade entre essa equação e a Equação (14.4)

$$S_t = \sum (y_i - \bar{y})^2 \quad (14.18)$$

Na Equação (14.18), o quadrado do resíduo representa o quadrado da discrepância entre os dados e uma única estimativa da medida da tendência central – a média. Na Equação (14.17), o quadrado do resíduo representa o quadrado da distância vertical entre os dados e uma outra medida da tendência central – a reta (Figura 14.9).

A analogia pode ser estendida para casos em que (1) a dispersão dos pontos em torno da reta tem valor absoluto parecido ao longo de todo o intervalo dos dados e (2) a distribuição desses pontos em torno da reta é normal. É possível demonstrar que, se esses critérios forem satisfeitos, a regressão por mínimos quadrados fornecerá as melhores estimativas (ou seja, as mais prováveis) de a_0 e a_1 (Draper e Smith, 1981). Isso é chamado de *princípio da probabilidade máxima*, em estatística. Além disso, se esses critérios forem satisfeitos, um “desvio-padrão” para a reta de regressão pode ser determinado por [compare com a Equação (14.3)]

$$s_{y/x} = \sqrt{\frac{S_r}{n-2}} \quad (14.19)$$

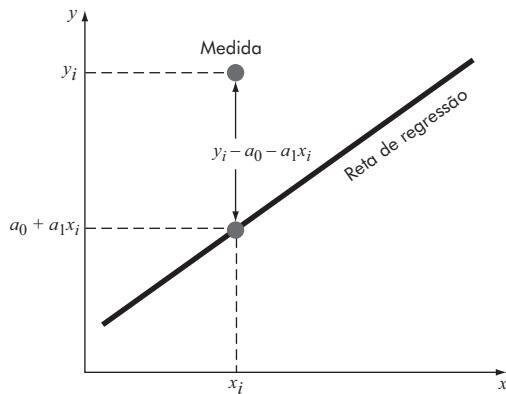


FIGURA 14.9 O resíduo na regressão linear representa a distância vertical entre os pontos dados e a reta.

onde $s_{y/x}$ é chamado *erro-padrão da estimativa*. O subscrito “ y/x ” indica que o erro é para um valor previsto de y correspondente a um valor particular de x . Observe também que agora estamos dividindo por $n - 2$, porque duas estimativas provenientes dos dados — a_0 e a_1 — foram usadas para calcular S_r ; portanto, perdemos dois graus de liberdade. De modo análogo à discussão do desvio-padrão, uma outra justificativa para dividir por $n - 2$ é que não existe nenhuma “dispersão de dados” em torno de uma reta ligando dois pontos. Então, nos casos em que $n = 2$, a Equação (14.19) fornece um resultado infinito, sem sentido.

Exatamente como no caso do desvio-padrão, o erro-padrão da estimativa quantifica a dispersão dos dados. Entretanto, $s_{y/x}$ quantifica a dispersão *em torno da reta de regressão*, como mostra a Figura 14.10b, ao contrário do desvio-padrão original s_y , que quantificava a dispersão *em torno da média* (Figura 14.10a).

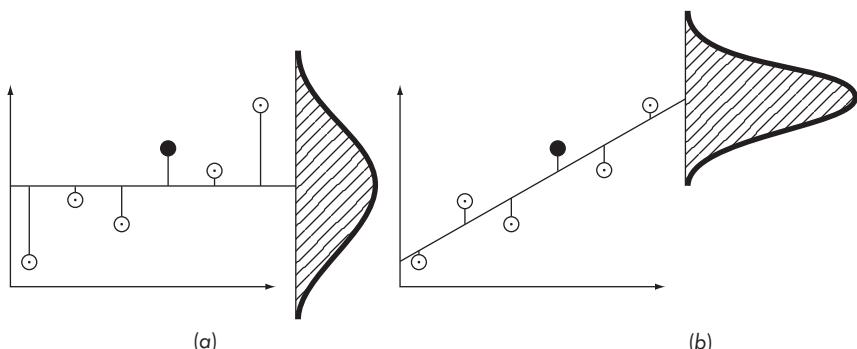


FIGURA 14.10 Dados de regressão que mostram (a) a dispersão dos dados em torno da média da variável e (b) a dispersão dos dados em torno da reta de melhor ajuste. A redução na dispersão ao ir de (a) para (b), como indicam as curvas em forma de sino à direita, representa a melhora decorrente da regressão linear.

Esses conceitos podem ser usados para quantificar a eficácia do ajuste. Isso é particularmente útil para comparar diversas regressões (Figura 14.11). Para isso, volta-se aos dados originais e determina-se a soma total dos quadrados em torno da média da variável dependente (no caso, y). Como no caso da Equação (14.18), essa quantidade é denotada por S_r , que é o módulo do erro residual associado com a variável dependente antes da regressão. Depois de fazer a regressão, pode-se calcular S_t , a soma dos quadrados dos resíduos em torno da reta de regressão, com a Equação (14.17). Isso caracteriza o erro residual que permanece depois da regressão. Portanto, às vezes ele é chamado de soma dos quadrados inexplicável. A diferença entre as duas quantidades, $S_t - S_r$, quantifica a melhora ou a redução de erro decorrente da descrição dos dados em termos de uma reta, em vez de um valor médio. Como o módulo dessa quantidade depende da escala, a diferença é normalizada por S_t para fornecer

$$r^2 = \frac{S_t - S_r}{S_t} \quad (14.20)$$

onde r^2 é chamado de *coeficiente de determinação* e r é o *coeficiente de correlação* ($= \sqrt{r^2}$). Para um ajuste perfeito, $S_r = 0$ e $r^2 = 1$, que significa que a reta explica 100% da variação dos dados. Para $r^2 = 0$, $S_r = S_t$, e o ajuste não representa nenhuma melhora. Uma formulação alternativa para r mais conveniente para implementação computacional é

$$r = \frac{n \sum(x_i y_i) - (\sum x_i)(\sum y_i)}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (14.21)$$

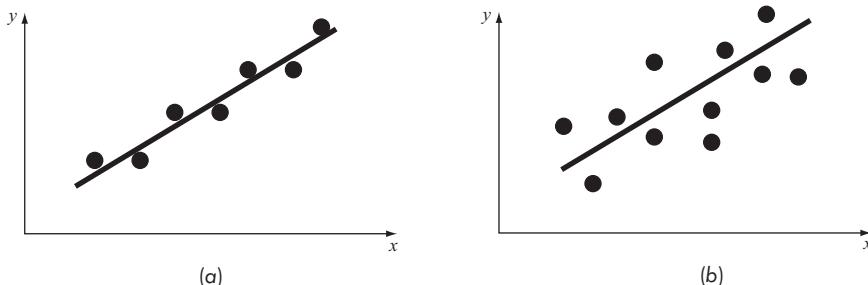


FIGURA 14.11 Exemplos de regressões lineares com erros residuais (a) pequenos e (b) grandes.

EXEMPLO 14.5 Estimativa de erros para um ajuste linear por mínimos quadrados

Calcule o desvio-padrão total, o erro-padrão da estimativa e o coeficiente de correlação para o ajuste no Exemplo 14.4.

Solução. Os dados podem ser organizados na forma de tabela e as somas necessárias calculadas como na Tabela 14.5.

TABELA 14.5 Dados e somas necessárias para calcular as estatísticas da eficácia do ajuste para os dados da Tabela 14.1.

<i>i</i>	<i>x_i</i>	<i>y_i</i>	<i>a₀ + a_rx_i</i>	(<i>y_i - ȳ</i>) ²	(<i>y_i - a₀ - a_rx_i</i>) ²
1	10	25	-39,58	380,535	4,171
2	20	70	155,12	327,041	7,245
3	30	380	349,82	68,579	911
4	40	550	544,52	8,441	30
5	50	610	739,23	1,016	16,699
6	60	1.220	933,93	334,229	81,837
7	70	830	1.128,63	35,391	89,180
8	80	1.450	1.323,33	653,066	16,044
Σ	360	5.135		1.808,297	216,118

O desvio-padrão é [Equação (14.3)]

$$s_y = \sqrt{\frac{1.808,297}{8 - 1}} = 508,26$$

e o erro-padrão da estimativa é [Equação (14.19)]

$$s_{y/x} = \sqrt{\frac{216,118}{8 - 2}} = 189,79$$

Portanto, como $s_{y/x} < s_y$, o modelo de regressão linear tem mérito. A extensão da melhora é quantificada por [Equação (14.20)]

$$r^2 = \frac{1.808,297 - 216,118}{1.808,297} = 0,8805$$

ou $r = \sqrt{0,8805} = 0,9383$. Esses resultados indicam que 88,05% da incerteza original foi explicada pelo modelo linear.

Antes de prosseguir, é preciso fazer um alerta. Embora o coeficiente de determinação forneça uma medida cômoda da eficácia do ajuste, é necessário não dar a ele mais importância do que a devida. Apenas o fato de r^2 estar “próximo” de 1 não significa que o ajuste seja necessariamente “bom”. Por exemplo, é possível obter um valor relativamente alto de r^2 quando a relação subjacente entre y e x não for nem mesmo linear. Draper e Smith (1981) forneceram diretrizes e material adicional relativos à avaliação dos resultados da regressão linear. Além disso, é necessário sempre inspecionar um gráfico dos dados junto com sua curva de regressão.

Um bom exemplo foi desenvolvido por Anscombe (1973). Como na Figura 14.12, ele apresentou quatro conjuntos de dados que consistem em 11 pontos cada. Embora seus gráficos sejam muito diferentes, todos têm a mesma melhor equação de ajuste, $y = 3 + 0,5x$, e o mesmo coeficiente de determinação $r^2 = 0,67$! Esse exemplo ilustra drasticamente por que o desenvolvimento de gráficos é tão importante.

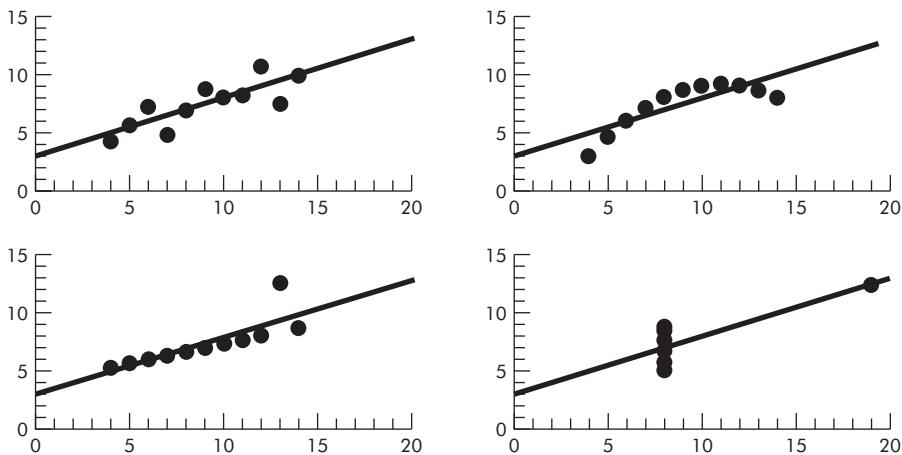


FIGURA 14.12 Quatro conjuntos de dados de Anscombe com a reta de melhor ajuste, $y = 3 + 0,5x$.

14.4 LINEARIZAÇÃO DE RELAÇÕES NÃO LINEARES

A regressão linear fornece uma técnica poderosa para ajustar a melhor reta aos dados. Entretanto, tem base no fato de que a relação entre as variáveis dependentes e independentes é linear. Isso não é regra, e o primeiro passo em qualquer análise de regressão deve ser traçar e inspecionar visualmente os dados, a fim de verificar se um modelo linear se aplica. Em alguns casos, técnicas como regressão polinomial, descrita no Capítulo 15, são apropriadas. Para outros, podem ser usadas transformações para expressar os dados em uma forma que seja compatível com a regressão linear.

Um exemplo é o *modelo exponencial*:

$$y = \alpha_1 e^{\beta_1 x} \quad (14.22)$$

onde α_1 e β_1 são constantes. Esse modelo é usado em muitos campos da engenharia e da ciência para caracterizar quantidades que aumentam (β_1 positivo) ou diminuem (β_1 negativo) a uma taxa que é diretamente proporcional ao seu próprio valor absoluto. Por exemplo, o crescimento populacional ou o decaimento radioativo podem exibir tal comportamento. Como descreve a Figura 14.13a, a equação representa uma relação não linear (para $\beta_1 \neq 0$) entre y e x .

Outro exemplo de um modelo não linear é a *equação de potência simples*:

$$y = \alpha_2 x^{\beta_2} \quad (14.23)$$

onde α_2 e β_2 são coeficientes constantes. Esse modelo tem larga aplicabilidade em todos os campos da engenharia e da ciência, sendo utilizado com frequência para ajustar dados experimentais quando o modelo subjacente não é conhecido. Como descreve a Figura 14.13b, a equação (para $\beta_2 \neq 0$) é não linear.

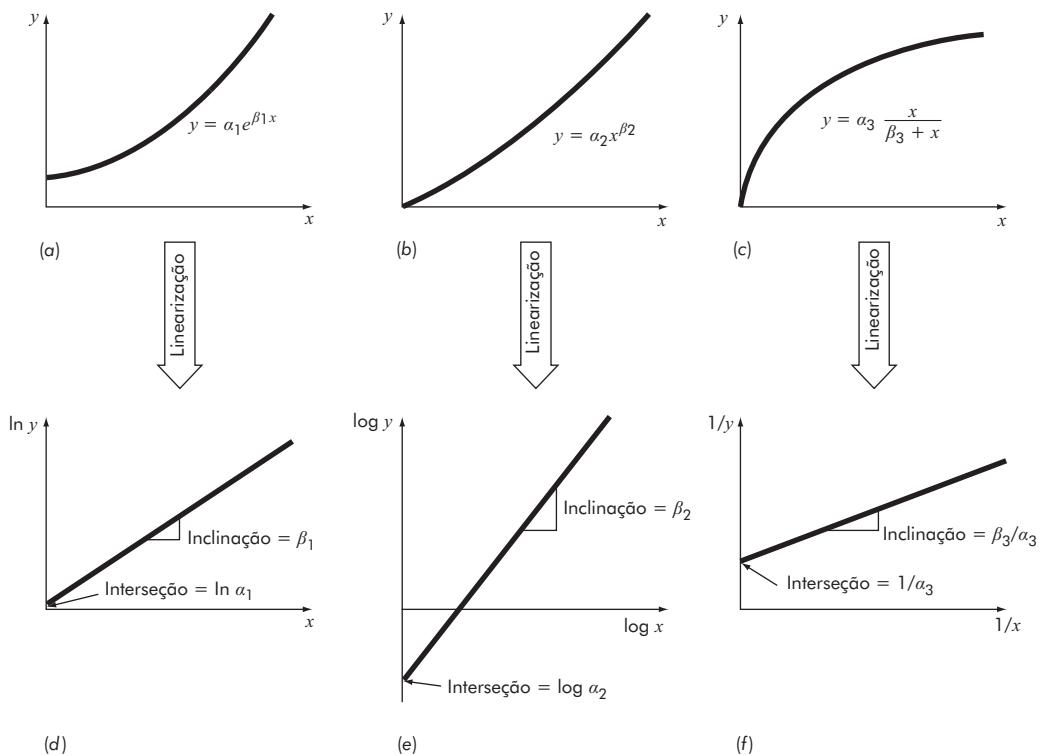


FIGURA 14.13 (a) A equação exponencial, (b) a equação de potência e (c) a equação da taxa de crescimento da saturação. As partes (d), (e) e (f) são versões linearizadas dessas equações que resultam de transformações simples.

Um terceiro exemplo de um modelo não linear é a *equação da taxa de crescimento da saturação*:

$$y = \alpha_3 \frac{x}{\beta_3 + x} \quad (14.24)$$

onde α_3 e β_3 são coeficientes constantes. Esse modelo, que é particularmente adequado para caracterizar a taxa de crescimento populacional sob condições limitantes, também representa uma relação não linear entre y e x (Figura 14.13c) que se nivela, ou “satura”, conforme x aumenta. Ele tem várias aplicações, principalmente em áreas tanto da engenharia quanto da ciência relacionadas à biologia.

As técnicas de regressão não linear estão disponíveis para ajustar tais equações aos dados experimentais diretamente. Entretanto, uma alternativa mais simples é usar manipulações matemáticas para transformar as equações para uma forma linear. Então, uma simples regressão linear pode ser usada para ajustar as equações aos dados.

Por exemplo, a Equação (14.22) pode ser linearizada tomando-se seu logaritmo natural para obter

$$\ln y = \ln \alpha_1 + \beta_1 x \quad (14.25)$$

Logo, um gráfico de $\ln y$ versus x irá fornecer uma reta com uma inclinação β_1 e uma interseção com o eixo y em $\ln \alpha_1$ (Figura 14.13d).

A Equação (14.23) é linearizada tomando-se seu logaritmo na base 10 para obter

$$\log y = \log \alpha_2 + \beta_2 \log x \quad (14.26)$$

Logo, um gráfico de $\log y$ versus $\log x$ irá fornecer uma reta com uma inclinação β_2 e uma interseção com o eixo y em $\log \alpha_2$ (Figura 14.13e). Observe que um logaritmo em uma base qualquer pode ser usado para linearizar esse modelo. No entanto, como feito aqui, o logaritmo na base 10 é o de modo geral empregado.

A Equação (14.24) é linearizada invertendo-a para obter

$$\frac{1}{y} = \frac{1}{\alpha_3} + \frac{\beta_3}{\alpha_3} \frac{1}{x} \quad (14.27)$$

Logo, um gráfico de $1/y$ em função de $1/x$ será linear, com uma inclinação β_3/α_3 e uma interseção com o eixo y em $1/\alpha_3$ (Figura 14.13f).

Nas suas formas transformadas, esses modelos podem usar a regressão linear para calcular os coeficientes constantes. Eles poderiam, então, ser convertidos de volta ao seu estado original e usados para propósitos de previsão. O exemplo a seguir ilustra esse procedimento para o modelo de potência simples.

EXEMPLO 14.6 Ajustando dados com a equação de potência simples

Ajuste a Equação (14.23) aos dados da Tabela 14.1 usando uma transformação logarítmica.

Solução. Os dados podem ser organizados na forma de tabela e as somas necessárias calculadas como na Tabela 14.6.

As médias podem ser calculadas como

$$\bar{x} = \frac{12,606}{8} = 1,5757 \quad \bar{y} = \frac{20,515}{8} = 2,5644$$

TABELA 14.6 Dados e somas necessárias para ajustar o modelo de potência simples aos dados da Tabela 14.1

<i>i</i>	x_i	y_i	$\log x_i$	$\log y_i$	$(\log x_i)^2$	$\log x_i \log y_i$
1	10	25	1,000	1,398	1,000	1,398
2	20	70	1,301	1,845	1,693	2,401
3	30	380	1,477	2,580	2,182	3,811
4	40	550	1,602	2,740	2,567	4,390
5	50	610	1,699	2,785	2,886	4,732
6	60	1220	1,778	3,086	3,162	5,488
7	70	830	1,845	2,919	3,404	5,386
8	80	1450	1,903	3,161	3,622	6,016
Σ		12,606	12,606	20,515	20,516	33,622

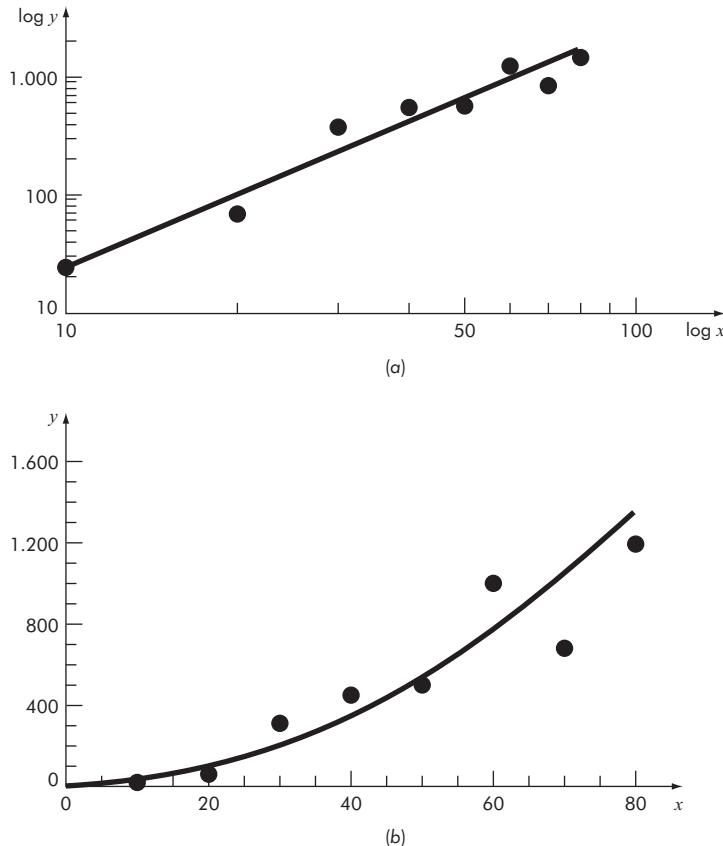


FIGURA 14.14 Ajuste por mínimos quadrados de um modelo de potência para os dados da Tabela 14.1. (a) Ajuste dos dados transformados. (b) Ajuste da equação de potência juntamente com os dados.

A inclinação e a interseção com o eixo y podem, então, ser calculadas com as Equações (14.15) e (14.16) como

$$a_1 = \frac{8(33,622) - 12,606(20,515)}{8(20,516) - (12,606)^2} = 1,9842$$

$$a_0 = 2,5644 - 1,9842(1,5757) = -0,5620$$

O ajuste por mínimos quadrados é

$$\log y = -0,5620 + 1,9842 \log x$$

O ajuste, juntamente com os dados, é mostrado na Figura 14.14a.

Também podemos exibir o ajuste utilizando as coordenadas não transformadas. Para isso, os coeficientes do modelo de potência simples são determinados como $\alpha_2 = 10^{-0,5620} = 0,2741$ e $\beta_2 = 1,9842$. Utilizando força e velocidade em lugar de y e x , o ajuste por mínimos quadrados é

$$F = 0,2741v^{1,9842}$$

Essa equação, juntamente com os dados, é mostrada na Figura 14.14b.

Os ajustes obtidos no Exemplo 14.6 (Figura 14.14) devem ser comparados com aquele obtido anteriormente no Exemplo 14.4 (Figura 14.8) utilizando regressão linear nos dados não transformados. Embora ambos os resultados pareçam aceitáveis, o resultado transformado tem a vantagem de não produzir previsões de força negativa em baixas velocidades. Além disso, sabe-se do estudo de mecânica dos fluidos que a força de arraste em um corpo que se move através de um fluido é geralmente descrita por um modelo com a velocidade ao quadrado. Assim, o conhecimento do campo que se estuda muitas vezes tem uma grande influência sobre a escolha da equação modelo apropriada a ser utilizada para o ajuste de curva.

14.4.1 Comentários gerais sobre regressão linear

Antes de prosseguir para regressão curvilinear e regressão linear múltipla, é necessário enfatizar a natureza introdutória do material anterior sobre regressão linear. Nós nos concentramos em deduções simples e usos práticos de equações para ajustar os dados. Você deve estar ciente de que existem aspectos teóricos da regressão que são de importância prática, mas que estão além do escopo deste livro. Por exemplo, algumas hipóteses estatísticas inerentes aos procedimentos por mínimos quadrados são:

1. Cada x tem um valor fixo; ele não é aleatório e é conhecido sem erros.
2. Os valores de y são variáveis aleatórias independentes e todos têm a mesma variância.
3. Os valores de y para um dado x devem estar normalmente distribuídos.

Essas hipóteses são relevantes para a dedução e o uso adequado da regressão. Por exemplo, a primeira hipótese significa que (1) os valores de x devem estar livres de erros e (2) a regressão de y em função de x não é a mesma que a de x em função de y . Dessa forma torna-se importante a consulta a outras referências, como Draper e Smith (1981), para apreciar aspectos e nuances da regressão que estão além do escopo deste livro.

14.5 APLICAÇÕES COMPUTACIONAIS

A regressão linear é tão comum que pode ser implementada na maioria das calculadoras de bolso. Nesta seção, mostraremos como um arquivo-M pode ser desenvolvido para determinar a inclinação e a interseção com o eixo y , assim como para criar um gráfico dos dados e da reta de melhor ajuste. Também mostraremos como a regressão linear pode ser implementada com a função nativa `polyfit`.

14.5.1 Função do MATLAB: `regrlin`

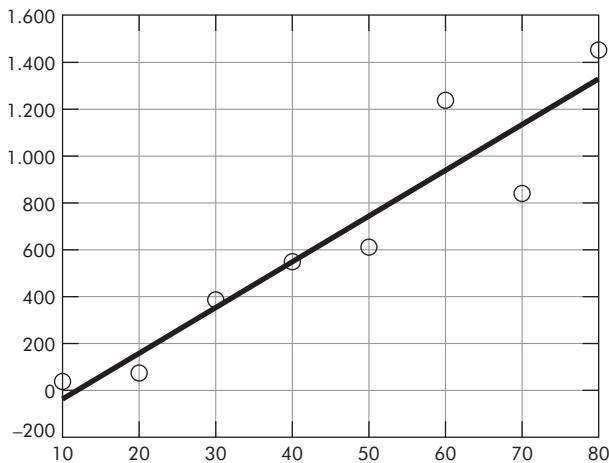
Um algoritmo para regressão linear pode ser facilmente desenvolvido (Figura 14.15). As somatórias necessárias são prontamente calculadas com a função `sum` do MATLAB. Em seguida, os seus resultados são usados para calcular a inclinação e a interseção com as Equações (14.15) e (14.16). A rotina exibe a interseção e a inclinação, o coeficiente de determinação e um gráfico da reta de melhor ajuste junto aos dados de medição.

Um exemplo simples do uso dessa função é ajustar os dados de força-velocidade analisados no Exemplo 14.4:

```
>> x = [10 20 30 40 50 60 70 80];
>> y = [25 70 380 550 610 1220 830 1450];
>> [a,r2] = regrlin(x,y)

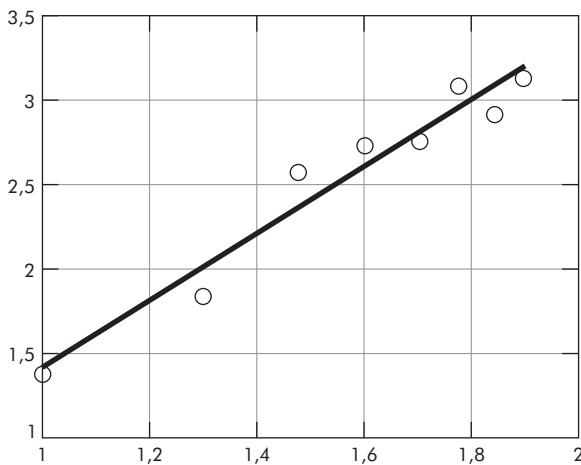
a =
    19.4702 -234.2857

r2 =
    0.8805
```



Elá pode também ser facilmente usada para ajustar o modelo de potênciā sim-
ples (Exemplo 14.6) pela aplicāção da função \log_{10} aos dados como em

```
>> [a,r2]=regrlin(log10(x),log10(y))
a =
    1.9842    -0.5620
r2 =
    0.9481
```



```

function [a, r2] = regrlin(x,y)
% regrlin: ajuste de curva por regressão linear
% [a, r2] = regrlin(x,y): Ajuste por mínimos
% quadrados de uma reta
%   aos dados pela solução das equações normais
% entrada:
%   x = variável independente
%   y = variável dependente
% saída:
%   a = vetor de inclinação, a(1), e interseção, a(2)
%   r2 = coeficiente de determinação

n = length(x);
if length(y) ~= n, error('x e y devem ter o mesmo
tamanho'); end
x = x(:); y = y(:); % converte para vetores coluna
sx = sum(x); sy = sum(y);
sx2 = sum(x.*x); sxy = sum(x.*y); sy2 = sum(y.*y);
a(1) = (n*sxy-sx*sy)/(n*sx2-sx^2);
a(2) = sy/n-a(1)*sx/n;
r2 = ((n*sxy-sx*sy)/sqrt(n*sx2-sx^2))/sqrt(n*sy2-sy^2))^2;
% cria um gráfico dos dados e da melhor reta de ajuste
xp = linspace(min(x),max(x),2);
yp = a(1)*xp+a(2);
plot(x,y,'o',xp,yp)
grid on

```

FIGURA 14.15 Função do MATLAB para implementar a regressão linear.

14.5.2 Funções do MATLAB: polyft e polyval

O MATLAB tem uma função nativa, `polyfit`, que ajusta por mínimos quadrados um polinômio de n -ésimo grau aos dados. Ela pode ser aplicada como em

```
>> p = polyft(x, y, n)
```

onde x e y são os vetores das variáveis independente e dependente, respectivamente, e n é o grau do polinômio. A função retorna um vetor p contendo os coeficientes do polinômio. Deve-se notar que o polinômio é representado por meio de potências decrescentes de x como na seguinte representação:

$$f(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1}$$

Como uma reta é um polinômio de primeiro grau, o comando `polyfit(x,y,1)` retornará a inclinação e a interseção da reta de melhor ajuste.

```
>> x = [10 20 30 40 50 60 70 80];
>> y = [25 70 380 550 610 1220 830 1450];
>> a = polyft(x,y,1)
a =
    19.4702 -234.2857
```

Assim, a inclinação é 19,4702 e a interseção é -234,2857.

Outra função, `polyval`, pode ser usada para calcular um valor utilizando os coeficientes. Ela tem o seguinte formato geral:

```
>> y = polyval(p, x)
```

onde p são os coeficientes do polinômio e y é o valor do melhor ajuste em x . Por exemplo,

```
>> y = polyval(a, 45)
y =
641.8750
```

► 14.6 ESTUDO DE CASO CINÉTICA ENZIMÁTICA

Contextualização. As *enzimas* agem como catalisadores para acelerar a taxa de reações químicas em células vivas. Na maioria dos casos, elas convertem uma substância química, o *substrato*, em outra, o *produto*. A equação de *Michaelis-Menten* é frequentemente usada para descrever tais reações:

$$v = \frac{v_m[S]}{k_s + [S]} \quad (14.28)$$

onde v é a velocidade inicial da reação, v_m é a velocidade inicial máxima da reação, $[S]$ é concentração de substrato e k_s é a constante de meia saturação. Como na Figura 14.16, a equação descreve uma relação de saturação que estabiliza com o crescimento de $[S]$. O gráfico também ilustra que a *constante de meia saturação* corresponde à concentração de substrato na qual a velocidade é metade da máxima.

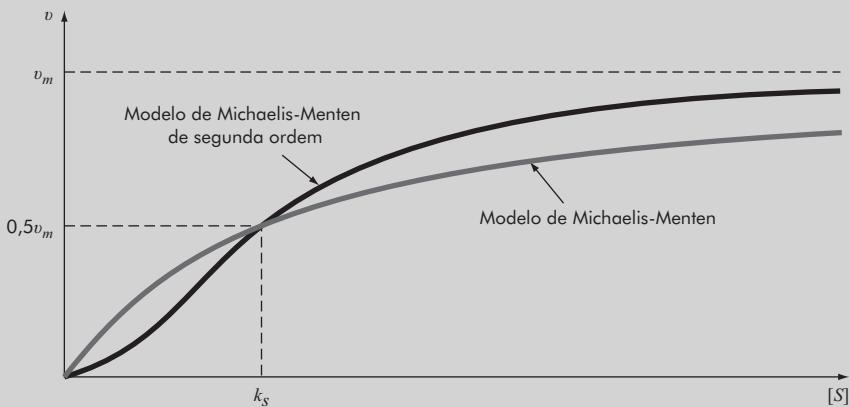


FIGURA 14.16 Duas versões do modelo de Michaelis-Menten de cinética enzimática.

Embora o modelo de Michaelis-Menten forneça um bom ponto de partida, ele foi refinado e estendido para incorporar características adicionais da cinética enzimática. Uma extensão simples envolve as chamadas *enzimas alostéricas*, em que a ligação de uma molécula de substrato em um sítio leva a uma ligação reforçada de moléculas subsequentes



em outros sítios. Para o caso com dois sítios de ligação interagindo, a seguinte versão de segunda ordem em geral resulta em um melhor ajuste:

$$v = \frac{v_m [S]^2}{k_s^2 + [S]^2} \quad (14.29)$$

Esse modelo também descreve uma curva de saturação; porém, como descreve a Figura 14.16, as concentrações ao quadrado tendem a tornar a forma mais *sigmoide*, ou em forma de S.

Considere que os seguintes dados sejam fornecidos:

[S]	1,3	1,8	3	4,5	6	8	9
v	0,07	0,13	0,22	0,275	0,335	0,35	0,36

Empregue regressão linear para ajustar esses dados com versões linearizadas das Equações (14.28) e (14.29). Além de estimar os parâmetros do modelo, avalie a validade dos ajustes com medidas estatísticas e gráficos.

Solução. A Equação (14.28), que está no formato do modelo da taxa de crescimento da saturação [Equação (14.24)], pode ser linearizada invertendo-a para fornecer [lembre-se da Equação (14.27)]:

$$\frac{1}{v} = \frac{1}{v_m} + \frac{k_s}{v_m} \frac{1}{[S]}$$

A função `regrlin` da Figura 14.15 pode então ser usada para determinar o ajuste por mínimos quadrados:

```
>> S=[1.3 1.8 3 4.5 6 8 9];
>> v=[0.07 0.13 0.22 0.275 0.335 0.35 0.36];
>> [a,r2]=regrlin(1./S,1./v)

a =
    16.4022    0.1902

r2 =
    0.9344
```

Os coeficientes do modelo podem ser calculados como

```
>> vm=1/a(2)

vm =
    5.2570

>> ks=vm*a(1)

ks =
    86.2260
```

Assim, o melhor ajuste do modelo é

$$v = \frac{5,2570[S]}{86,2260 + [S]}$$

Embora o valor elevado de r^2 possa levá-lo a acreditar que esse resultado é aceitável, uma inspeção dos coeficientes pode levantar dúvidas. Por exemplo, a velocidade máxima (5,2570) é muito superior à maior velocidade observada (0,36). Além disso, a taxa de meia saturação (86,2260) é muito maior que a concentração máxima de substrato (9).

Os problemas tornam-se evidentes quando o ajuste é traçado com os dados. A Figura 14.17a mostra a versão transformada. Embora a reta siga a tendência ascendente, os dados claramente parecem descrever uma curva. Quando a equação original é traçada com os dados na versão não transformada (Figura 14.17b), o ajuste é obviamente inaceitável. Os dados estão claramente se estabilizando em cerca de 0,36 ou 0,37. Se isso está correto, uma estimativa visual sugeriria que v_m deveria ser cerca de 0,36 e k_s deveria estar na faixa de 2 a 3.

Além dessa evidência visual, a pobreza do ajuste também é refletida por estatísticas como o coeficiente de determinação. Para o caso não transformado, um resultado muito menos aceitável de $r^2 = 0,6406$ é obtido.

A análise anterior pode ser repetida para o modelo de segunda ordem. A Equação (14.28) também pode ser linearizada invertendo-a para fornecer

$$\frac{1}{v} = \frac{1}{v_m} + \frac{k_s^2}{v_m [S]^2}$$

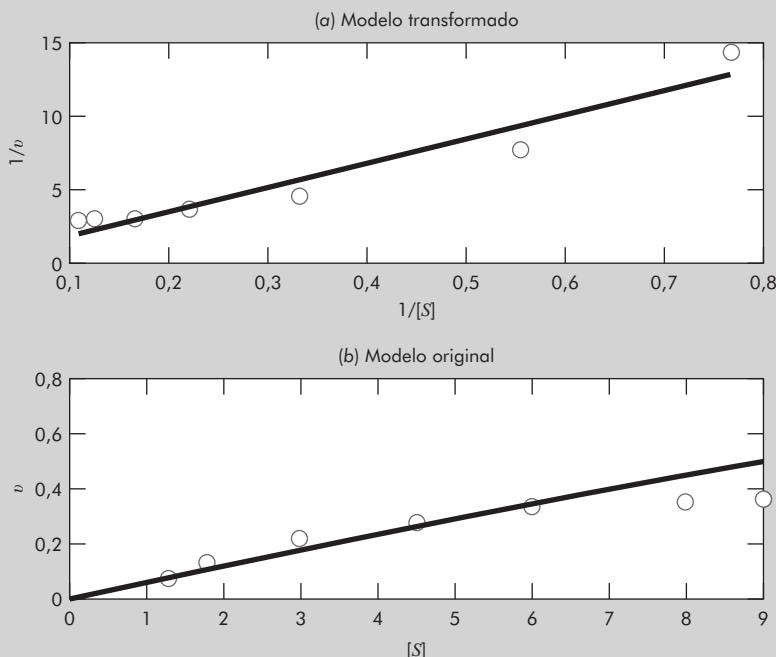


FIGURA 14.17 Gráficos do ajuste por mínimos quadrados (linha contínua) do modelo de Michaelis-Menten junto com os dados (pontos). O gráfico em (a) mostra o ajuste transformado, e (b) mostra como o ajuste fica quando visualizado na forma original, não transformada.

A função `regrlin` da Figura 14.15 pode novamente ser usada para determinar o ajuste por mínimos quadrados:

```

>> [a,r2]=regrlin(1./S.^2,1./v)
a =
    19.3760    2.4492
r2 =
    0.9929

```

Os coeficientes do modelo podem então ser calculados como

```
>> vm=1/a(2)
vm =
    0.4083
>> ks=sqrt(v)
ks =
    2.8127
```

Substituindo esses valores na Equação (14.29), obtém-se

$$v = \frac{0,4083[S]^2}{7,911 + [S]^2}$$

Embora saibamos que um valor elevado de r^2 não garante um bom ajuste, o fato de ele ser bastante alto (0,9929) é promissor. Além disso, os valores dos parâmetros também parecem consistentes com as tendências nos dados; ou seja, o v_m é ligeiramente superior à maior velocidade observada, e a taxa de meia saturação é menor que a máxima concentração de substrato (9).

A adequação do ajuste pode ser avaliada por meio de gráficos. Como na Figura 14.18a, o modelo transformado parece linear. Quando a equação original é traçada com os dados na versão não transformada (Figura 14.18b), o ajuste segue muito bem a tendência nas medições. Além dos gráficos, a qualidade do ajuste também é refletida pelo fato de que o coeficiente de determinação para o caso não transformado pode ser calculado como $r^2 = 0,9896$.

Com base em nossa análise, pode-se concluir que o modelo de segunda ordem fornece um bom ajuste para esse conjunto de dados. Isso poderia sugerir que estamos lidando com uma enzima alostérica.

Além desse resultado específico, existem algumas outras conclusões gerais que podem ser extraídas desse estudo de caso. Primeiro, nunca se deve apoiar apenas em estatísticas tais como r^2 como a única base de avaliação do ajuste. Segundo, as equações de regressão devem sempre ser avaliadas graficamente. E, para os casos em que transformações são empregadas, um gráfico do modelo não transformado junto aos dados deve sempre ser inspecionado.

Por fim, embora transformações possam produzir um ajuste aceitável dos dados transformados, isso nem sempre se traduz em um ajuste aceitável no formato original. A razão por que isso pode ocorrer é que minimizar a soma dos quadrados dos resíduos dos dados transformados não é o mesmo que fazer essa minimização para os dados não transformados. A regressão linear assume que a dispersão dos pontos em torno da reta de melhor ajuste segue uma distribuição Gaussiana e que o desvio-padrão é o mesmo em cada valor da variável dependente. Essas suposições raramente são verdadeiras depois de transformar os dados.

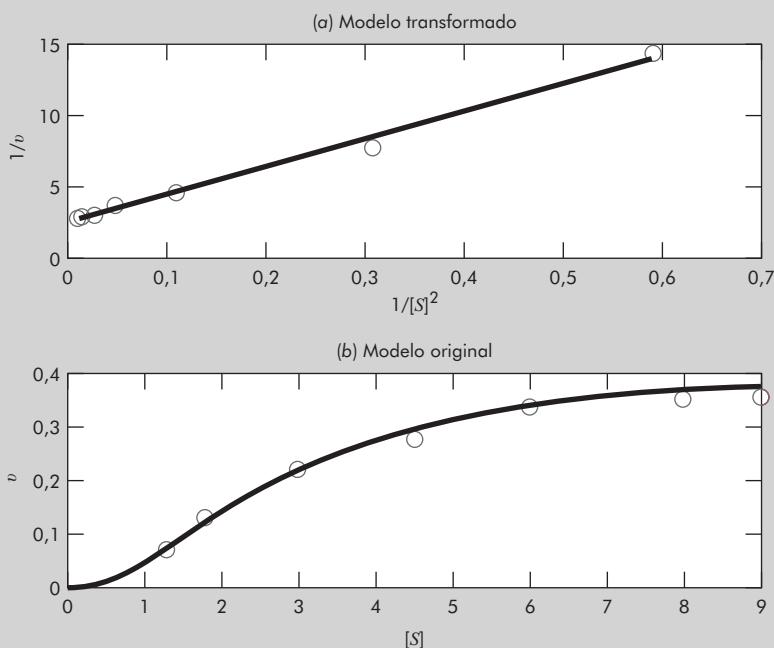


FIGURA 14.18 Gráficos do ajuste por mínimos quadrados (linha contínua) do modelo de Michaelis-Menten de segunda ordem junto com os dados (pontos). O gráfico em (a) mostra o ajuste transformado, e (b) mostra como o ajuste fica quando visualizado na forma original, não transformada.

Como uma consequência da última conclusão, algumas análises sugerem que em vez de utilizar transformações lineares, uma regressão não linear deveria ser empregada para ajustar os dados curvilineares. Nessa abordagem, uma curva de melhor ajuste é desenvolvida de modo a minimizar diretamente os resíduos não transformados. Descreveremos como isso é feito no Capítulo 15.

►► PROBLEMAS

14.1 Considere os dados

0,90	1,42	1,30	1,55	1,63
1,32	1,35	1,47	1,95	1,66
1,96	1,47	1,92	1,35	1,05
1,85	1,74	1,65	1,78	1,71
2,29	1,82	2,06	2,14	1,27

Determine (a) a média, (b) a mediana, (c) a moda, (d) a amplitude, (e) o desvio-padrão, (f) a variância e (g) o coeficiente de variação.

14.2 Construa um histograma a partir dos dados do Problema 14.1. Use uma faixa de 0,8 a 2,4, com intervalos de 0,2.

14.3 Considere os dados

29,65	28,55	28,65	30,15	29,35	29,75	29,25
30,65	28,15	29,85	29,05	30,25	30,85	28,75
29,65	30,45	29,15	30,45	33,65	29,35	29,75
31,25	29,45	30,15	29,65	30,55	29,65	29,25

Determine (a) a média, (b) a mediana, (c) a moda, (d) a amplitude, (e) o desvio-padrão, (f) a variância e (g) o coeficiente de variação.

- (h) Construa um histograma. Use um intervalo de 28 a 34, com incrementos de 0,4.
- (i) Considerando que a distribuição seja normal e que sua estimativa do desvio-padrão seja

válida, calcule o intervalo (ou seja, as extremidades inferior e superior) que englobe 68% das leituras. Determine se essa é uma estimativa válida para os dados desse problema.

14.4 Utilizando a mesma abordagem que foi empregada para deduzir as Equações (14.15) e (14.16), derive o ajuste por mínimos quadrados do seguinte modelo:

$$y = \alpha_1 x + e$$

ou seja, determine a inclinação que resulta no ajuste por mínimos quadrados para uma reta com interseção nula com o eixo y . Ajuste os seguintes dados com esse modelo e mostre o resultado por meio de um gráfico.

x	2	4	6	7	10	11	14	17	20
y	4	5	6	5	8	8	6	9	12

14.5 Use regressão por mínimos quadrados para ajustar uma reta a

x	0	2	4	6	9	11	12	15	17	19
y	5	6	7	6	9	8	8	10	12	12

Junto com a inclinação e a interseção com o eixo y , calcule o erro-padrão da estimativa e o coeficiente de correlação. Faça o gráfico dos dados e da reta de regressão. Em seguida, repita o problema, mas faça uma regressão de x versus y – ou seja, troque as variáveis. Interprete seus resultados.

14.6 Ajuste um modelo de potência aos dados da Tabela 14.1, mas use o logaritmo natural para efetuar as transformações.

14.7 Os seguintes dados foram reunidos para determinar a relação entre a pressão e a temperatura de um volume fixo de 1 kg de nitrogênio. O volume é 10 m³.

$T, ^\circ\text{C}$	-40	0	40	80	120	160
$p, \text{N/m}^2$	6.900	8.100	9.350	10.500	11.700	12.800

Use a lei dos gases ideais $pV = nRT$ para determinar R com base nesses dados. Observe que para a lei, a temperatura T deve ser expressa em kelvins.

14.8 Além dos exemplos na Figura 14.13, existem outros modelos que podem ser linearizados usando-se transformações. Por exemplo,

$$y = \alpha_4 x e^{\beta_4 x}$$

Linearize esse modelo e use-o para fazer uma estimativa de α_4 e β_4 com base nos dados a seguir. Faça um gráfico de seu ajuste juntamente com os dados.

x	0,1	0,2	0,4	0,6	0,9	1,3	1,5	1,7	1,8
y	0,75	1,25	1,45	1,25	0,85	0,55	0,35	0,28	0,18

14.9 A concentração de bactéria *E. coli* em uma área de prática de natação é monitorada após uma tempestade:

$t (\text{h})$	4	8	12	16	20	24
$c (\text{UFC/100 mL})$	1.600	1.320	1.000	890	650	560

O tempo é medido em horas seguindo o fim da tempestade, e a unidade UFC é uma “unidade formadora de colônia”. Use esses dados para estimar (a) a concentração no fim da tempestade ($t = 0$) e (b) o instante de tempo em que a concentração alcançará 200 UFC/100 mL. Observe que a escolha do modelo deve ser consistente com o fato de que concentrações negativas são impossíveis e que a concentração de bactéria sempre decresce com o tempo.

14.10 Em vez de usar um modelo exponencial na base e [Equação (14.22)], uma alternativa comum é empregar um modelo na base 10:

$$y = \alpha_5 10^{\beta_5 x}$$

Quando usada para ajuste de curvas, essa equação fornece resultados idênticos aos da base e , mas o valor do parâmetro no expoente (β_5) será diferente do valor estimado com a Equação (14.22) (β_1). Use a versão na base 10 para resolver o Problema 14.9. Além disso, deduza uma fórmula para relacionar β_1 a β_5 .

14.11 Determine uma equação para prever a taxa de metabolismo como uma função da massa com base nos dados a seguir. Use-a para prever a taxa de metabolismo de um tigre de 200 kg.

Animal	Massa (kg)	Metabolismo (watts)
Vaca	400	270
Humano	70	82
Ovelha	45	50
Galinha	2	4,8
Rato	0,3	1,45
Pombo	0,16	0,97

14.12 Em média, a área superficial A de seres humanos é relacionada ao peso W e à altura H . Medidas de vários indivíduos de altura 180 cm e diferentes pesos (kg) fornecem os valores de A (m^2) indicados na seguinte tabela:

$W(\text{kg})$	70	75	77	80	82	84	87	90
$A(\text{m}^2)$	2,10	2,12	2,15	2,20	2,22	2,23	2,26	2,30

Mostre que a lei da potência $A = aW^b$ fornece um ajuste razoavelmente bom para esses dados. Calcule as constantes a e b e faça uma previsão da área superficial de uma pessoa de 95 kg.

14.13 Ajuste um modelo exponencial a

x	0,4	0,8	1,2	1,6	2	2,3
y	800	985	1.490	1.950	2.850	3.600

Trace os dados e a equação em gráficos-padrão e semilogarítmico com a função `subplot` do MATLAB.

14.14 Um pesquisador relatou os dados tabulados a seguir para uma experiência a fim de determinar a taxa de crescimento da bactéria k (por dia) como uma função da concentração de oxigênio c (mg/L). Sabe-se que tais dados podem ser modelados pela seguinte equação:

$$k = \frac{k_{\max} c^2}{c_s + c^2}$$

onde c_s e k_{\max} são parâmetros. Use uma transformação para linearizar essa equação. A seguir, use regressão linear para estimar c_s e k_{\max} e prever a taxa de crescimento em $c = 2$ mg/L.

c	0,5	0,8	1,5	2,5	4
k	1,1	2,5	5,3	7,6	8,9

14.15 Desenvolva uma função no MATLAB para calcular estatísticas descritivas para um vetor de valores. A função deve determinar e exibir o número de valores, a média, a mediana, a moda, a amplitude, o desvio-padrão, a variância e o coeficiente de variação. Além disso, um histograma deve ser gerado. Teste a função com os dados do Problema 14.3.

14.16 Modifique a função `regrlin` da Figura 14.15 de modo que ela (a) calcule e retorne o erro-padrão da estimativa, e (b) use a função `subplot` para também exibir um gráfico dos resíduos (o previsto

menos o valor de y medido) versus x . Teste a função para os dados dos Exemplos 14.2 e 14.3.

14.17 Desenvolva uma função no MATLAB para ajustar um modelo de potência. A função deve retornar o coeficiente de melhor ajuste α_2 e a potência β_2 com o r^2 para o modelo não transformado. Além disso, use a função `subplot` para exibir gráficos tanto da equação transformada como da não transformada, juntamente com os dados. Teste a função com os dados do Problema 14.11.

14.18 Os seguintes dados mostram a relação entre a viscosidade do óleo SAE 70 e a temperatura. Depois de tomar o log dos dados, use uma regressão linear para encontrar a equação da reta que melhor ajusta os dados e o valor de r^2 .

Temperatura, °C	26,67	93,33	148,89	315,56
Viscosidade, μ, N · s/m²	1,35	0,085	0,012	0,00075

14.19 Você fez experiências e determinou os seguintes valores para a capacidade térmica c em várias temperaturas T para um gás:

T	-50	-30	0	60	90	110
c	1.250	1.280	1.350	1.480	1.580	1.700

Use regressão para determinar um modelo para prever c como função de T .

14.20 É conhecido que a força de tensão do plástico aumenta em função do tempo quando ele é tratado com calor. São coletados os seguintes dados:

Tempo	10	15	20	25	40	50	55	60	75
Força de tensão	5	20	18	40	33	54	70	60	78

- (a) Ajuste uma reta a esses dados e use a equação para determinar a força de tensão para o instante 32 minutos.
- (b) Repita a análise, mas para uma reta com interseção com o eixo y no zero.

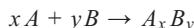
14.21 Os seguintes dados foram tomados de um reator de mistura para a reação $A \rightarrow B$. Use os dados para determinar a melhor estimativa possível para k_{01} e E_1 para o seguinte modelo cinético:

$$-\frac{dA}{dt} = k_{01}e^{-E_1/RT} A$$

onde R é a constante do gás e é igual a 0,00198 Kcal/mol/K.

$-\frac{dA}{dt}$ (mols/L/s)	460	960	2.485	1.600	1.245
A (mols/L)	200	150	50	20	10
T (K)	280	320	450	500	550

14.22 Dados de concentração foram tomados em 15 instantes de tempo para a reação de polimerização:



Considere que a reação ocorra via um mecanismo complexo que consiste em muitos passos. Diversos modelos foram considerados e as somas dos quadrados dos resíduos foram calculadas para os ajustes dos modelos aos dados. Os resultados estão mostrados a seguir. Qual modelo descreve melhor os dados (estatisticamente)? Explique sua escolha.

	Modelo A	Modelo B	Modelo C
S_r	135	105	100
Número de parâmetros do modelo no ajuste	2	3	5

14.23 A seguir estão os dados tomados do crescimento bacteriano em um reator em batelada (depois da fase de atraso terminar). Foi permitido que as bactérias crescessem tanto quanto possível nas primeiras 2,5 horas, e então elas foram induzidas a produzir uma proteína recombinante, sendo que essa produção diminui significativamente o crescimento bacteriano. O crescimento teórico das bactérias pode ser descrito por

$$\frac{dX}{dt} = \mu X$$

onde X é o número de bactérias e μ é a taxa de crescimento específico da bactéria durante o crescimento exponencial. Com base nos dados, estime a taxa de crescimento específico da bactéria durante as primeiras 2 horas de crescimento e durante as próximas 4 horas de crescimento.

Tempo, h	0	1	2	3	4	5	6
[Células], g/L	0,100	0,335	1,102	1,655	2,453	3,702	5,460

14.24 Um estudo de engenharia de transporte foi conduzido para determinar um projeto adequado

de faixas para bicicleta. Foram coletados dados sobre a largura das faixas para bicicletas e a distância média entre as bicicletas e os carros trafegando. Os dados de nove ruas são:

Distância, m	2,4	1,5	2,4	1,8	1,8	2,9	1,2	3	1,2
Largura da faixa, m	2,9	2,1	2,3	2,1	1,8	2,7	1,5	2,9	1,5

- (a) Trace os dados.
- (b) Ajuste uma reta aos dados, por regressão linear. Acrescente essa reta ao gráfico.
- (c) Se a distância média de segurança mínima entre as bicicletas e os carros trafegando é considerada 1,8 m, determine a largura de faixa mínima correspondente.

14.25 Na engenharia de recursos hídricos, a estimativa dos tamanhos dos reservatórios depende de estimativas acuradas do escoamento da água no rio que está sendo confinado. Em alguns rios, registros históricos de longa duração de tais dados de escoamento são difíceis de obter. Já os dados meteorológicos sobre precipitação estão disponíveis para muitos anos do passado. Portanto, é útil determinar a relação entre escoamento e precipitação. Essa relação pode ser usada para fazer uma estimativa do escoamento nos anos nos quais apenas medidas de precipitação foram feitas. Os seguintes dados estão disponíveis para um rio que deve ser estancado:

Precipitação, cm/ano	88,9	108,5	104,1	139,7	127	94	116,8	99,1
Escoamento, m³/s	14,6	16,7	15,3	23,2	19,5	16,1	18,1	16,6

- (a) Trace os dados.
- (b) Ajuste uma reta aos dados, por regressão linear. Superponha essa reta ao seu gráfico.
- (c) Use a reta de melhor ajuste para prever o escoamento anual de água se a precipitação for 120 cm.
- (d) Se a área de drenagem é 1.100 km², estime qual fração da precipitação é perdida por processos como evaporação, infiltração profunda da água no solo e uso destrutivo.

14.26 O mastro de um barco a vela tem área da seção transversal 10,65 cm² e é construído de uma liga de alumínio experimental. Foram feitos testes

para definir a relação entre a tensão e a deformação. Os resultados do teste são

Deformação,

cm/cm 0,0032 0,0045 0,0055 0,0016 0,0085 0,0005

Tensão,

N/cm² 4.970 5.170 5.500 3.590 6.900 1.240

A tensão causada pelo vento pode ser calculada como F/A_c , onde F é a força no mastro e A_c é a área da seção transversal do mastro. Esse valor pode então ser substituído na lei de Hooke para determinar a deflexão no mastro, $\Delta L = \text{deformação} \times L$, onde L é o comprimento do mastro. Se a força do vento for 25.000 N, use os dados para fazer uma estimativa da deflexão de um mastro de 9 m.

14.27 Os seguintes dados foram obtidos de um experimento que mediou a corrente em um fio para várias tensões impostas:

V, V	2	3	4	5	7	10
i, A	5,2	7,8	10,7	13	19,3	27,5

- (a) Com base em uma regressão linear desses dados, determine a corrente para a tensão de 3,5 V. Trace a reta e os dados e avalie o ajuste.
- (b) Refaça a regressão e force a interseção com o eixo y a ser zero.

14.28 É feita uma experiência para determinar a elongação percentual de um material condutor elétrico como uma função da temperatura. Os dados resultantes estão listados a seguir. Faça uma previsão para a elongação percentual para a temperatura de 400 °C.

Temperatura, °C	200	250	300	375	425	475	600
Elongação, %	7,5	8,6	8,7	10	11,3	12,7	15,3

14.29 A população p de uma pequena comunidade na periferia de uma cidade cresce rapidamente durante um período de 20 anos:

t	0	5	10	15	20
p	100	200	450	950	2.000

Como um engenheiro que trabalha para uma companhia de serviços públicos, você deve prever qual será a população daqui a cinco anos para antecipar a demanda por energia elétrica. Use um

modelo exponencial e regressão linear para fazer essa previsão.

14.30 A velocidade u do ar escoando por uma superfície plana é medida a diferentes distâncias y da superfície. Ajuste uma curva a esses dados considerando que a velocidade seja zero na superfície ($y = 0$). Use seu resultado para determinar a tensão de cisalhamento ($\mu du/dy$) na superfície onde $\mu = 1,8 \times 10^{-5}$ N·s/m².

y, m	0,002	0,006	0,012	0,018	0,024
u, m/s	0,287	0,899	1,915	3,048	4,299

14.31 A Equação de Andrade foi proposta como um modelo para o efeito da temperatura sobre a viscosidade:

$$\mu = De^{B/T_a}$$

onde μ é a viscosidade dinâmica da água (10^{-3} N·s/m²), T_a é a temperatura absoluta (K) e D e B são parâmetros. Ajuste esse modelo para os seguintes dados para água:

T	0	5	10	20	30	40
μ	1,787	1,519	1,307	1,002	0,7975	0,6529

14.32 Faça os mesmos cálculos como no Exemplo 14.2, mas além do coeficiente de arraste, varie também a massa uniformemente por $\pm 10\%$.

14.33 Efetue os mesmos cálculos como no Exemplo 14.3, mas além do coeficiente de arraste, varie também a massa normalmente em torno de seu valor médio com um coeficiente de variação de 5,7887%.

14.34 A fórmula de Manning para um canal rectangular pode ser escrita como

$$Q = \frac{1}{n_m} \frac{(BH)^{5/3}}{(B + 2H)^{2/3}} \sqrt{S}$$

onde Q é o fluxo (m^3/s), n_m é um coeficiente de rugosidade, B é a largura (m), H é a profundidade (m) e S é a inclinação. Você está aplicando essa fórmula a um fluxo em que se conhece a largura, 20 m, e a profundidade, 0,3 m. Infelizmente, você conhece a rugosidade e a inclinação com uma precisão de apenas $\pm 10\%$; ou seja, você sabe que a rugosidade é cerca de 0,03, com uma variação de 0,027 a 0,033 e a inclinação é 0,0003 com uma variação de 0,00027 a 0,00033. Admitindo distribuições

uniformes, use uma análise de Monte Carlo com $n = 10.000$ para estimar a distribuição de fluxo.

14.35 Uma análise de Monte Carlo pode ser usada para otimização. Por exemplo, a trajetória de uma bola pode ser calculada com

$$y = (\tan \theta_0)x - \frac{g}{2v_0^2 \cos^2 \theta_0}x^2 + y_0 \quad (\text{P14.35})$$

onde y é a altura (m), θ_0 é o ângulo inicial (radianos), v_0 é a velocidade inicial (m/s), g é a aceleração

da gravidade = 9,81 m/s² e y_0 é a altura inicial (m). Dados $y_0 = 1$ m, $v_0 = 25$ m/s e $\theta_0 = 50^\circ$, determine a altura máxima e a distância x correspondente (a) analiticamente empregando o cálculo e (b) numericamente com simulação de Monte Carlo. Para este último, desenvolva um programa que gere um vetor de 10.000 valores de x uniformemente distribuídos entre 0 e 60 m. Use esse vetor e a Equação (P14.35) para gerar um vetor de alturas. Em seguida, empregue a função `max` para determinar a altura máxima e a distância x associada.

15

Mínimos quadrados: modelo linear geral e regressão não linear

► OBJETIVOS DO CAPÍTULO

Este capítulo retoma o conceito de ajuste de uma reta e o estende para (a) ajustar um polinômio e (b) ajustar uma variável que é uma função linear de duas ou mais variáveis independentes. Em seguida, mostra-se como tais aplicações podem ser generalizadas e aplicadas a um amplo grupo de problemas. Por fim, ilustra como técnicas de otimização podem ser usadas para implementar regressão não linear. Após esta leitura completa, você será capaz de:

- ✓ Aprender como implementar regressão polinomial.
- ✓ Aprender como implementar regressão linear múltipla.
- ✓ Entender a formulação do modelo de mínimos quadrados linear geral.
- ✓ Entender como o modelo de mínimos quadrados linear geral pode ser resolvido com o MATLAB utilizando as equações normais ou a divisão à esquerda.
- ✓ Entender como implementar regressão não linear com técnicas de otimização.

15.1 REGRESSÃO POLINOMIAL

No Capítulo 14, foi desenvolvido um procedimento para deduzir a equação da reta usando o critério dos mínimos quadrados. Alguns dados, embora exibam um padrão definido tal como visto na Figura 15.1, são representados de forma insatisfatória por uma reta. Nesses casos, uma curva é mais adequada para ajustar os dados. Como discutido no Capítulo 14, um método utilizado para atingir esse objetivo é o de transformações. Uma alternativa é ajustar polinômios aos dados usando *regressão polinomial*.

O procedimento dos mínimos quadrados pode ser prontamente estendido para ajustar dados por um polinômio de grau mais alto. Por exemplo, considere que se queira ajustar um polinômio de segundo grau ou quadrático:

$$y = a_0 + a_1x + a_2x^2 + e \quad (15.1)$$

Nesse caso, a soma dos quadrados dos resíduos é

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2)^2 \quad (15.2)$$

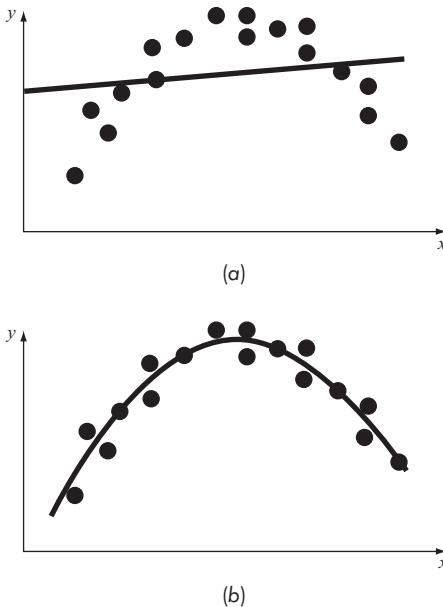


FIGURA 15.1 (a) Dados que são pouco adequados para regressão linear por mínimos quadrados. (b) Indicação de que uma parábola é preferível.

Para gerar o ajuste por mínimos quadrados, toma-se a derivada da Equação (15.2) com relação a cada um dos coeficientes desconhecidos do polinômio, como em

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2)$$

Essas equações podem ser igualadas a zero e reorganizadas para determinar o seguinte conjunto de equações normais:

$$(n) a_0 + (\sum x_i) a_1 + (\sum x_i^2) a_2 = \sum y_i$$

$$(\sum x_i) a_0 + (\sum x_i^2) a_1 + (\sum x_i^3) a_2 = \sum x_i y_i$$

$$(\sum x_i^2) a_0 + (\sum x_i^3) a_1 + (\sum x_i^4) a_2 = \sum x_i^2 y_i$$

onde todas as somatórias vão de $i = 1$ a n . Observe que essas três equações são lineares e têm três incógnitas a_0 , a_1 e a_2 . Os coeficientes das incógnitas podem ser calculados diretamente dos dados observados.

Nesse caso, vê-se que o problema de determinar o polinômio de segundo grau por mínimos quadrados é equivalente a resolver um sistema de três equações lineares

simultâneas. O caso bidimensional pode ser facilmente estendido para um polinômio de grau m , como em

$$y = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m + e$$

A análise anterior pode ser facilmente aplicada a esse caso mais geral. Portanto, pode-se perceber que determinar os coeficientes de um polinômio de grau m é equivalente a resolver um sistema de $m + 1$ equações lineares simultâneas. Nesse caso, o erro-padrão é formulado como

$$s_{y/x} = \sqrt{\frac{S_r}{n - (m + 1)}} \quad (15.3)$$

Essa quantidade é dividida por $n - (m + 1)$, porque $(m + 1)$ coeficientes provenientes dos dados — a_0, a_1, \dots, a_m — foram usados para calcular S_r ; logo, foram perdidos $m + 1$ graus de liberdade. Além do erro-padrão, um coeficiente de determinação pode ser calculado para a regressão polinomial com a Equação (14.20).

EXEMPLO 15.1 Regressão polinomial

Ajuste um polinômio de segundo grau aos dados nas duas primeiras colunas da Tabela 15.1.

TABELA 15.1 Cálculos para uma análise de erros do ajuste quadrático por mínimos quadrados.

x_i	y_i	$(y_i - \bar{y})^2$	$(y_i - a_0 - a_1x_i - a_2x_i^2)^2$
0	2,1	544,44	0,14332
1	7,7	314,47	1,00286
2	13,6	140,03	1,08160
3	27,2	3,12	0,80487
4	40,9	239,22	0,61959
5	61,1	1.272,11	0,09434
Σ	152,6	2.513,39	3,74657

Solução. A partir dos dados fornecidos, o seguinte pode ser calculado

$$m = 2 \quad \sum x_i = 15 \quad \sum x_i^4 = 979$$

$$n = 6 \quad \sum y_i = 152,6 \quad \sum x_i y_i = 585,6$$

$$\bar{x} = 2,5 \quad \sum x_i^2 = 55 \quad \sum x_i^2 y_i = 2.488,8$$

$$\bar{y} = 25,433 \quad \sum x_i^3 = 225$$

Portanto, as equações lineares simultâneas são

$$\begin{bmatrix} 6 & 15 & 55 \\ 15 & 55 & 225 \\ 55 & 225 & 979 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 152,6 \\ 585,6 \\ 2.488,8 \end{Bmatrix}$$

Essas equações podem ser resolvidas para calcular os coeficientes. Por exemplo, usando o MATLAB:

```
>> N = [6 15 55;15 55 225;55 225 979];
>> r = [152.6 585.6 2488.8];
>> a = N\r
a =
    2.4786
    2.3593
    1.8607
```

Portanto, a equação quadrática por mínimos quadrados para esse caso é

$$y = 2,4786 + 2,3593x + 1,8607x^2$$

O erro-padrão da estimativa com base na regressão polinomial é [Equação (15.3)]

$$s_{y/x} = \sqrt{\frac{3,74657}{6 - (2 + 1)}} = 1,1175$$

O coeficiente de determinação é

$$r^2 = \frac{2513,39 - 3,74657}{2513,39} = 0,99851$$

E o coeficiente de correlação é $r = 0,99925$.

Os resultados indicam que 99,851% da incerteza original foi explicada por esse modelo. Esse resultado confirma a conclusão de que a equação quadrática representa um excelente ajuste, como também é evidente a partir da Figura 15.2.

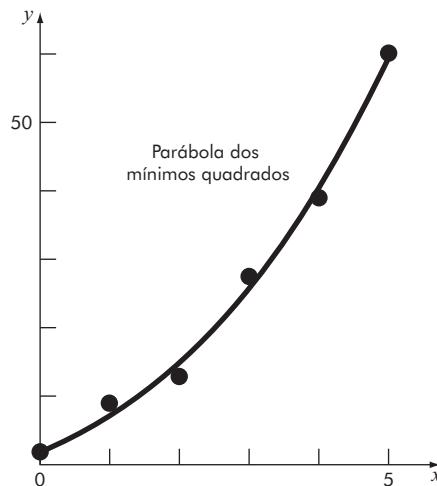


FIGURA 15.2 Ajuste por um polinômio de segundo grau.

15.2 REGRESSÃO LINEAR MÚLTIPLA

Uma extensão útil da regressão linear é o caso no qual y é uma função linear de duas ou mais variáveis independentes. Por exemplo, y poderia ser uma função linear de x_1 e x_2 , como em

$$y = a_0 + a_1x_1 + a_2x_2 + e$$

Essa equação é particularmente útil para ajustar dados experimentais, para os quais as variáveis estudadas são frequentemente uma função de duas outras variáveis. Nesse caso bidimensional, a “reta” de regressão se torna um “plano” de regressão (Figura 15.3).

Como nos casos anteriores, os “melhores” valores dos coeficientes são determinados escrevendo-se a soma dos quadrados dos resíduos:

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1x_{1,i} - a_2x_{2,i})^2 \quad (15.4)$$

e derivando-a com relação a cada um dos coeficientes desconhecidos:

$$\begin{aligned} \frac{\partial S_r}{\partial a_0} &= -2 \sum (y_i - a_0 - a_1x_{1,i} - a_2x_{2,i}) \\ \frac{\partial S_r}{\partial a_1} &= -2 \sum x_{1,i}(y_i - a_0 - a_1x_{1,i} - a_2x_{2,i}) \\ \frac{\partial S_r}{\partial a_2} &= -2 \sum x_{2,i}(y_i - a_0 - a_1x_{1,i} - a_2x_{2,i}) \end{aligned}$$

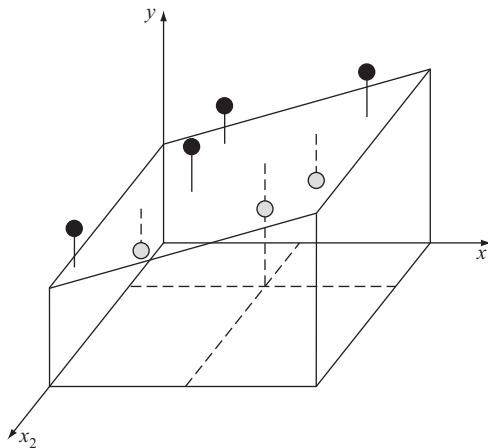


FIGURA 15.3 Descrição gráfica da regressão linear múltipla na qual y é uma função linear de x_1 e x_2 .

Os coeficientes que fornecem a soma mínima dos quadrados dos resíduos são obtidos igualando-se as derivadas parciais a zero e expressando o resultado na forma matricial como

$$\begin{bmatrix} n & \sum x_{1,i} & \sum x_{2,i} \\ \sum x_{1,i} & \sum x_{1,i}^2 & \sum x_{1,i}x_{2,i} \\ \sum x_{2,i} & \sum x_{1,i}x_{2,i} & \sum x_{2,i}^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_{1,i}y_i \\ \sum x_{2,i}y_i \end{Bmatrix} \quad (15.5)$$

EXEMPLO 15.2 Regressão linear múltipla

Os seguintes dados foram calculados a partir da equação $y = 5 + 4x_1 - 3x_2$:

x_1	x_2	y
0	0	5
2	1	10
2,5	2	9
1	3	0
4	6	3
7	2	27

Use regressão linear múltipla para ajustar esses dados.

Solução. As somatórias necessárias para desenvolver a Equação (15.5) estão calculadas na Tabela 15.2. Substituindo-as na Equação (15.5) tem-se o sistema:

$$\begin{bmatrix} 6 & 16,5 & 14 \\ 16,5 & 76,25 & 48 \\ 14 & 48 & 54 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} = \begin{Bmatrix} 54 \\ 243,5 \\ 100 \end{Bmatrix} \quad (15.6)$$

que pode ser resolvido, obtendo-se

$$a_0 = 5 \quad a_1 = 4 \quad a_2 = -3$$

que é consistente com a equação original a partir da qual os dados foram obtidos.

TABELA 15.2 Cálculos necessários para desenvolver as equações normais para o Exemplo 15.2.

y	x_1	x_2	x_1^2	x_2^2	x_1x_2	x_1y	x_2y
5	0	0	0	0	0	0	0
10	2	1	4	1	2	20	10
9	2,5	2	6,25	4	5	22,5	18
0	1	3	1	9	3	0	0
3	4	6	16	36	24	12	18
<u>27</u>	<u>7</u>	<u>2</u>	<u>49</u>	<u>4</u>	<u>14</u>	<u>189</u>	<u>54</u>
<u>54</u>	<u>16,5</u>	<u>14</u>	<u>76,25</u>	<u>54</u>	<u>48</u>	<u>243,5</u>	<u>100</u>

O caso bidimensional anterior pode ser facilmente aplicado a m dimensões como em

$$y = a_0 + a_1x_1 + a_2x_2 + \dots + a_mx_m + e$$

para o qual o erro-padrão é formulado como

$$s_{y/x} = \sqrt{\frac{S_r}{n - (m + 1)}}$$

e o coeficiente de determinação é calculado com a Equação (14.20).

Embora possam existir certos casos em que a variável seja linearmente relacionada a duas ou mais outras variáveis, a regressão linear múltipla tem utilidade adicional na dedução das equações de potência na forma geral

$$y = a_0x_1^{a_1}x_2^{a_2}\cdots x_m^{a_m}$$

Essas equações são extremamente úteis para ajustar dados experimentais. Para usar regressão linear múltipla, a equação é transformada tomando seu logaritmo para obter

$$\log y = \log a_0 + a_1 \log x_1 + a_2 \log x_2 + \dots + a_m \log x_m$$

15.3 MÍNIMOS QUADRADOS LINEAR GERAL

Nas páginas anteriores, foram introduzidos três tipos de regressão: linear simples, polinomial e linear múltipla. Na realidade, todas as três pertencem ao seguinte modelo geral de mínimos quadrados linear:

$$y = a_0z_0 + a_1z_1 + a_2z_2 + \dots + a_mz_m + e \quad (15.7)$$

onde z_0, z_1, \dots, z_m são $m + 1$ funções-base. Pode ser visto facilmente como a regressão linear simples e a linear múltipla se encaixam nesse modelo – isto é, $z_0 = 1, z_1 = x_1, z_2 = x_2, \dots, z_m = x_m$. Além disso, a regressão polinomial também é incluída se as funções-base forem monômios simples como em $z_0 = 1, z_1 = x, z_2 = x^2, \dots, z_m = x^m$.

Observe que a terminologia “linear” se refere apenas à dependência do modelo de seus parâmetros – ou seja, os a ’s. De modo semelhante à regressão polinomial, as próprias funções podem ser altamente não lineares. Por exemplo, os z ’s podem ser funções senoidais, como em

$$y = a_0 + a_1 \cos(\omega x) + a_2 \sin(\omega x)$$

Essa forma é a base da *análise de Fourier*.

Entretanto, um modelo aparentemente simples como

$$y = a_0(1 - e^{-a_1x})$$

é verdadeiramente não linear, porque não pode ser manipulado na forma da Equação (15.7).

A Equação (15.7) pode ser expressa na notação matricial como

$$\{y\} = [Z]\{a\} + \{e\} \quad (15.8)$$

onde $[Z]$ é a matriz dos valores calculados das funções-base nos valores medidos das variáveis independentes:

$$[Z] = \begin{bmatrix} z_{01} & z_{11} & \cdots & z_{m1} \\ z_{02} & z_{12} & \cdots & z_{m2} \\ \vdots & \vdots & & \vdots \\ z_{0n} & z_{1n} & \cdots & z_{mn} \end{bmatrix}$$

onde m é o número de variáveis no modelo e n é o número de pontos dados. Como $n \geq m + 1$, é importante notar que, na maioria das vezes, $[Z]$ não é uma matriz quadrada.

O vetor coluna $\{y\}$ contém os valores observados da variável dependente:

$$\{y\}^T = [y_1 \quad y_2 \quad \cdots \quad y_n]$$

O vetor coluna $\{a\}$ contém os coeficientes desconhecidos:

$$\{a\}^T = [a_0 \quad a_1 \quad \cdots \quad a_m]$$

e o vetor coluna $\{e\}$ contém os resíduos:

$$\{e\}^T = [e_1 \quad e_2 \quad \cdots \quad e_n]$$

A soma dos quadrados dos resíduos para esse modelo pode ser definida como

$$S_r = \sum_{i=1}^n \left(y_i - \sum_{j=0}^m a_j z_{ji} \right)^2 \quad (15.9)$$

Essa quantidade pode ser minimizada tomando-se suas derivadas parciais em relação a cada um dos coeficientes e igualando-se a equação resultante a zero. O resultado desse processo são as equações normais que podem ser expressas concisamente na forma matricial como

$$[[Z]^T [Z]] \{a\} = \{[Z]^T \{y\}\} \quad (15.10)$$

Pode-se mostrar que, de fato, a Equação (15.10) é equivalente às equações normais desenvolvidas anteriormente para a regressão linear simples, polinomial e linear múltipla.

O coeficiente de determinação e o erro-padrão também podem ser formulados em termos de álgebra matricial. Lembre-se de que r^2 é definido como

$$r^2 = \frac{S_t - S_r}{S_t} = 1 - \frac{S_r}{S_t}$$

Substituindo a definição de S_r e S_t , tem-se:

$$r^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2}$$

onde \hat{y} é a previsão do ajuste por mínimos quadrados. Os resíduos entre a curva de melhor ajuste e os dados, $y_i - \hat{y}_i$, podem ser expressos na forma vetorial como

$$\{y\} - [Z]\{a\}$$

A álgebra matricial também pode ser usada para manipular esse vetor para calcular tanto o coeficiente de determinação quanto o erro-padrão da estimativa, como ilustra o exemplo a seguir.

EXEMPLO 15.3 Regressão polinomial com o MATLAB

Repete o Exemplo 15.1, porém use operações matriciais, como descrito nesta seção.

Solução. Primeiro, entre com os dados a serem ajustados

```
>> x = [0 1 2 3 4 5]';
>> y = [2.1 7.7 13.6 27.2 40.9 61.1]';
```

Em seguida, crie a matriz $[Z]$:

```
>> Z = [ones(size(x)) x x.^2]
```

```
Z =
1     0     0
1     1     1
1     2     4
1     3     9
1     4    16
1     5    25
```

Pode-se verificar que $[Z]^T[Z]$ resulta na matriz de coeficientes para as equações normais:

```
>> Z'*Z
ans =
6     15     55
15     55    225
55    225    979
```

Esse é o mesmo resultado obtido com somatórias no Exemplo 15.1. Pode-se também calcular os coeficientes do ajuste quadrático por mínimos quadrados implementando a Equação (15.10):

```
>> a = (Z'*Z)\(Z'*y)
ans =
2.4786
2.3593
1.8607
```

Para calcular r^2 e s_{yx} , primeiro calcula-se a soma dos quadrados dos resíduos:

```
>> Sr = sum((y-Z*a).^2)
Sr =
3.7466
```

Então, r^2 pode ser calculado como

```
>> r2 = 1-Sr/sum((y-mean(y)).^2)
r2 =
0.9985
```

e s_{yx} pode ser calculado como

```
>> syx = sqrt(Sr/(length(x)-length(a)))
syx =
1.1175
```

Nosso objetivo principal na discussão anterior foi ilustrar a unidade entre as três abordagens e mostrar como elas podem ser todas expressas simplesmente na mesma notação matricial; além de preparar o terreno para a próxima seção, na qual será exposta alguma percepção sobre as estratégias preferidas para resolver a Equação (15.10). A notação matricial também terá relevância quando se abordar a regressão não linear na Seção 15.5.

15.4 DECOMPOSIÇÃO QR E OPERADOR BARRA INVERTIDA

A geração de um melhor ajuste pela solução das equações normais é amplamente utilizada e certamente adequada para várias aplicações de ajuste de curva na engenharia e na ciência. Deve ser ressaltado, no entanto, que as equações normais podem ser mal condicionadas e, portanto, sensíveis a erros de arredondamento.

Dois métodos mais avançados, a *decomposição QR* e a *decomposição por valor singular*, são mais eficientes a esse respeito. Embora a descrição desses métodos esteja além do escopo deste texto, eles são mencionados aqui porque podem ser implementados com o MATLAB.

Além disso, a decomposição QR é automaticamente usada em duas maneiras simples dentro do MATLAB. Primeiro, para os casos em que se quer ajustar um polinômio, a função nativa `polyfit` automaticamente usa a decomposição QR para obter os resultados.

Segundo, o problema de mínimos quadrados linear geral pode ser resolvido diretamente com o operador barra invertida. Lembre-se de que o modelo geral é formulado como a Equação (15.8)

$$\{y\} = [Z]\{a\} \quad (15.11)$$

Na Seção 10.4, a divisão à esquerda foi utilizada com o operador barra invertida para resolver sistemas de equações algébricas lineares em que o número de equações era igual ao número de incógnitas ($n = m$). Para a Equação (15.8) como derivada do modelo de mínimos quadrados geral, o número de equações é maior que o número de incógnitas ($n > m$). Esses sistemas são chamados *sobre determinados*. Quando o MATLAB detecta que você quer resolver tais sistemas com divisão à esquerda, ele automaticamente usa a decomposição QR para obter a solução. O exemplo a seguir ilustra como isso é feito.

EXEMPLO 15.4 Implementando a regressão polinomial com a função `polyft` e divisão à esquerda

Repita o Exemplo 15.3, porém use a função nativa `polyfit` e a divisão à esquerda para calcular os coeficientes.

Solução. Como no Exemplo 15.3, pode-se entrar com os dados e usá-los para criar a matriz $[Z]$ como em

```
>> x = [0 1 2 3 4 5]';
>> y = [2.1 7.7 13.6 27.2 40.9 61.1]';
>> Z = [ones(size(x)) x x.^2];
```

A função `polyfit` pode ser usada para calcular os coeficientes:

```
>> a = polyfit(x,y,2)
a =
    1.8607    2.3593    2.4786
```

O mesmo resultado também pode ser calculado por meio do operador barra invertida:

```
>> a = Z\y
a =
    2.4786
    2.3593
    1.8607
```

Como afirmado anteriormente, ambos os resultados são obtidos automaticamente com a decomposição QR.

15.5 REGRESSÃO NÃO LINEAR

Há muitos casos na engenharia e na ciência nos quais modelos não lineares devem ser ajustados aos dados. No presente contexto, esses modelos são definidos como aqueles que têm uma dependência não linear com seus parâmetros. Por exemplo,

$$y = a_0(1 - e^{-a_1x}) + e \quad (15.12)$$

Essa equação não pode ser manipulada de modo que fique de acordo com a forma geral da Equação (15.7).

Como nos casos dos mínimos quadrados lineares, a regressão não linear é baseada na determinação dos valores dos parâmetros que minimizem a soma dos quadrados dos resíduos. Entretanto, para o caso não linear, a solução deve prosseguir de uma forma iterativa.

Existem técnicas expressamente projetadas para regressão não linear. Por exemplo, o método de Gauss-Newton utiliza a expansão em série de Taylor para expressar a equação não linear original em uma forma aproximada linear. A seguir, a teoria dos mínimos quadrados pode ser usada para se obter novas estimativas dos parâmetros que se movam na direção de minimizar o resíduo. Detalhes sobre esta abordagem são fornecidos em outras referências (Chapra e Canale, 2010).

Uma alternativa é usar técnicas de otimização para determinar diretamente o ajuste por mínimos quadrados. Por exemplo, a Equação (15.12) pode ser expressa como uma função objetivo para calcular a soma dos quadrados:

$$f(a_0, a_1) = \sum_{i=1}^n [y_i - a_0(1 - e^{-a_1x_i})]^2 \quad (15.13)$$

Uma rotina de otimização pode então ser usada para determinar os valores de a_0 e a_1 que minimizam a função.

Como descrito na Seção 7.3.1, a função `fminsearch` do MATLAB pode ser usada para esse propósito. Ela tem a sintaxe geral

```
[x, fval] = fminsearch(fun, x0, opcoes, p1, p2, ...)
```

onde x é um vetor dos valores dos parâmetros que minimizam a função fun , $fval$ é o valor da função no mínimo, $x0$ é um vetor das aproximações iniciais para os parâmetros, $opcoes$ é uma estrutura que contém valores dos parâmetros de otimização como criados com a função `optimset` (lembre-se da Seção 6.5) e $p1$, $p2$, etc. são argumentos adicionais que são passados para a função objetivo. Observe que se o argumento $opcoes$ é omitido, o MATLAB utiliza valores-padrão (default) que são razoáveis para a maioria dos problemas. Caso se queira passar argumentos adicionais ($p1$, $p2$, ...), mas não se deseja definir $opcoes$, utilize colchetes vazios [] no lugar desse argumento.

EXEMPLO 15.5 Regressão não linear com o MATLAB

Lembre-se de que no Exemplo 14.6 ajustamos o modelo de potência aos dados da Tabela 14.1 por linearização utilizando logaritmos. Isso produziu o modelo:

$$F = 0,2741v^{1,9842}$$

Repita esse exercício, mas use regressão não linear. Empregue aproximações iniciais de 1 para os coeficientes.

Solução. Uma função deve ser criada para calcular a soma dos quadrados. O seguinte arquivo, chamado `fSSR.m`, é criado para a equação da potência:

```
function f = fSSR(a, xm, ym)
yp = a(1)*xm.^a(2);
f = sum((ym-yp).^2);
```

No modo de comando, pode-se entrar com os dados como

```
>> x = [10 20 30 40 50 60 70 80];
>> y = [25 70 380 550 610 1220 830 1450];
```

A minimização da função é então implementada por

```
>> fminsearch(@fSSR, [1, 1], [], x, y)
ans =
2.5384 1.4359
```

O modelo de melhor ajuste é, portanto,

$$F = 2,5384v^{1,4359}$$

Tanto o ajuste transformado original quanto a versão presente estão exibidos na Figura 15.4. Observe que, embora os coeficientes do modelo sejam bem diferentes, é difícil de avaliar qual ajuste é superior com base na inspeção do gráfico.

Esse exemplo ilustra como equações de melhor ajuste diferentes são obtidas quando é feito o ajuste do mesmo modelo utilizando regressão não linear *versus* regressão linear empregando transformações. Isso ocorre porque no primeiro caso minimizam-se os resíduos dos dados originais, enquanto no segundo minimizam-se os resíduos dos dados transformados.

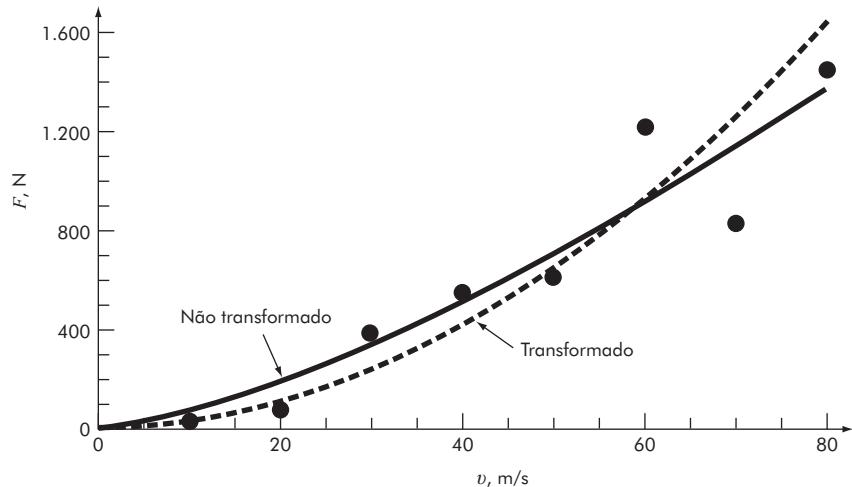


FIGURA 15.4 Comparação de ajustes de modelo transformado e não transformado para os dados de força versus velocidade da Tabela 14.1.

► 15.6 ESTUDO DE CASO AJUSTANDO DADOS EXPERIMENTAIS

Contextualização. Como mencionado no fim da Seção 15.2, embora existam vários casos em que uma variável é linearmente relacionada a duas ou mais outras variáveis, a regressão linear múltipla tem utilidade adicional na derivação de equações de potência de várias variáveis da forma geral

$$y = a_0 x_1^{a_1} x_2^{a_2} \cdots x_m^{a_m} \quad (15.14)$$

Essas equações são extremamente úteis no ajuste de dados experimentais. Para isso, a equação é transformada tomando-se seu logaritmo para produzir

$$\log y = \log a_0 + a_1 \log x_1 + a_2 \log x_2 + \cdots + a_m \log x_m \quad (15.15)$$

Assim, o logaritmo da variável dependente é linearmente dependente do logaritmo das variáveis independentes.

Um exemplo simples refere-se à transferência de gás em águas naturais como rios, lagos e estuários. Em particular, verificou-se que o coeficiente de transferência de massa de oxigênio dissolvido K_L (m/d) é relacionado à velocidade média da água do rio U (m/s) e à profundidade H (m) por

$$K_L = a_0 U^{a_1} H^{a_2} \quad (15.16)$$

Tomando o logaritmo comum, tem-se:

$$\log K_L = \log a_0 + a_1 \log U + a_2 \log H \quad (15.17)$$

Os seguintes dados foram coletados em um laboratório à temperatura constante de 20 °C:

U	0,5	2	10	0,5	2	10	0,5	2	10
H	0,15	0,15	0,15	0,3	0,3	0,3	0,5	0,5	0,5
K_L	0,48	3,9	57	0,85	5	77	0,8	9	92

Use esses dados e os mínimos quadrados linear geral para calcular as constantes na Equação (15.16).

Solução. De modo similar ao Exemplo 15.3, podemos desenvolver um programa para atribuir os dados, criar a matriz [Z] e calcular os coeficientes para o ajuste por mínimos quadrados:

```
% Cálculo do melhor ajuste dos valores transformados
clc; format short g
U=[0.5 2 10 0.5 2 10 0.5 2 10]';
H=[0.15 0.15 0.15 0.3 0.3 0.3 0.5 0.5 0.5]';
KL=[0.48 3.9 57 0.85 5 77 0.8 9 92]';
logU=log10(U); logH=log10(H); logKL=log10(KL);
Z=[ones(size(logKL)) logU logH];
a=(Z'*Z)\(Z'*logKL)
```

com o resultado:

```
a =
0.57627
1.562
0.50742
```

Portanto, o modelo de melhor ajuste é

$$\log K_L = 0,57627 + 1,562 \log U + 0,50742 \log H$$

ou, na forma não transformada (observe que, $a_0 = 10^{0,57627} = 3,7694$),

$$K_L = 3,7694 U^{1,5620} H^{0,5074}$$

As estatísticas também podem ser calculadas adicionando-se as seguintes linhas ao programa:

```
% Cálculo das estatísticas do ajuste
Sr=sum((logKL-Z*a).^2)
r2=1-Sr/sum((logKL-mean(logKL)).^2)
syx=sqrt(Sr/(length(logKL)-length(a)))
Sr =
0.024171
r2 =
0.99619
syx =
0.063471
```

Por fim, os gráficos do ajuste podem ser traçados. As linhas de código a seguir geram um gráfico das previsões do modelo *versus* os valores medidos para K_L . O comando subplot é empregado para fazer isso para as versões transformada e não transformada.

```
% Geração dos gráficos
clf
KLprev=10^a(1)*U.^a(2).*H.^a(3);
KLmin=min(KL); KLmax=max(KL);
dKL=(KLmax-KLmin)/100;
KLmod=[KLmin:dKL:KLmax];
subplot(1,2,1)
loglog(KLprev,KL,'ko',KLmod,KLmod,'k-')
axis square,title('(a) Gráfico log-log')
legend('previsão do modelo', 'Reta 1:1', 'Location','NorthWest')
xlabel('log(K_L) medido'),ylabel('log(K_L) previsto')
subplot(1,2,2)
plot(KLprev,KL,'ko',KLmod,KLmod,'k-')
axis square,title('(b) Gráfico não transformado')
legend('previsão do modelo', 'Reta 1:1', 'Location','NorthWest')
xlabel('log(K_L) medido'),ylabel('log(K_L) previsto')
```

O resultado é mostrado na Figura 15.5.

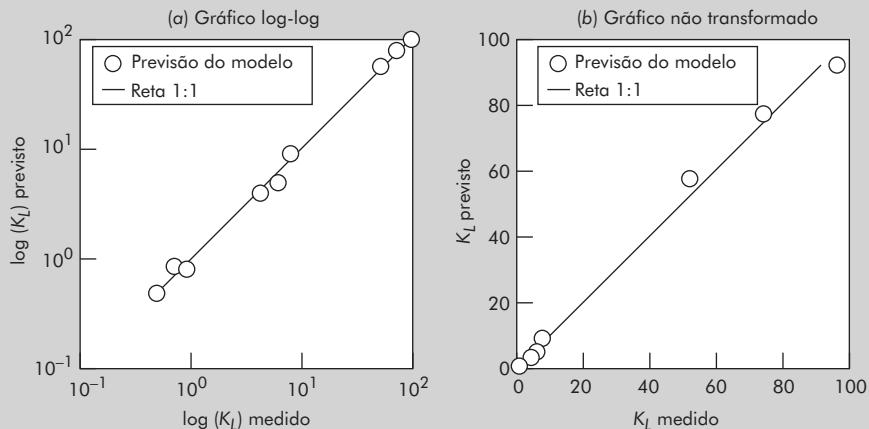


FIGURA 15.5 Gráficos dos valores previstos versus valores medidos do coeficiente de transferência de massa do oxigênio, empregando regressão múltipla. Os resultados são apresentados para (a) transformação logarítmica e (b) não transformado. A reta 1:1, que indica uma correlação perfeita, é sobreposta em ambos os gráficos.

►►PROBLEMAS

15.1 Ajuste uma parábola aos dados da Tabela 14.1. Determine r^2 para o ajuste e comente a eficácia do resultado.

15.2 Usando a mesma abordagem que foi empregada para deduzir as Equações (14.15) e (14.16), deduza um ajuste por mínimos quadrados do seguinte modelo:

$$y = a_1x + a_2x^2 + e$$

Ou seja, determine os coeficientes que resultam no ajuste por mínimos quadrados para um polinômio de 2º grau com interseção com o eixo y igual a zero. Teste a abordagem usando-a para ajustar os dados da Tabela 14.1.

15.3 Ajuste um polinômio cúbico aos seguintes dados:

x	3	4	5	7	8	9	11	12
y	1,6	3,6	4,4	3,4	2,2	2,8	3,8	4,6

Juntamente com os coeficientes, determine r^2 e s_{yx} .

15.4 Desenvolva uma função no MATLAB para implementar a regressão polinomial. Passe para a função dois vetores contendo os valores de x e y , com a ordem desejada m . Teste-a resolvendo o Problema 15.3.

15.5 Para os dados da Tabela P15.5, use regressão polinomial para obter uma equação de previsão para a concentração de oxigênio dissolvido como uma função da temperatura para o caso em que a concentração de cloreto é igual a zero. Empregue um polinômio de ordem suficientemente elevada de modo que as previsões igualem o número de dígitos significativos exibidos na tabela.

15.6 Use regressão linear múltipla para deduzir uma equação de previsão da concentração de oxigênio dissolvido como uma função da temperatura e do cloreto com base nos dados da Tabela P15.5. Utilize a equação para fazer uma estimativa da concentração de oxigênio dissolvido para uma concentração de cloreto de 15 g/L a $T = 12^\circ\text{C}$. Observe que o valor verdadeiro é 9,09 mg/L. Calcule o erro relativo percentual para sua previsão. Explique as possíveis causas para a discrepância.

TABELA P15.5 Concentração de oxigênio dissolvido em água como uma função da temperatura ($^\circ\text{C}$) e da concentração de cloreto (g/L).

$T, {}^\circ\text{C}$	Oxigênio dissolvido (mg/L) para temperatura ($^\circ\text{C}$) e concentração de cloreto (g/L)		
	$c = 0 \text{ g/L}$	$c = 10 \text{ g/L}$	$c = 20 \text{ g/L}$
0	14,6	12,9	11,4
5	12,8	11,3	10,3
10	11,3	10,1	8,96
15	10,1	9,03	8,08
20	9,09	8,17	7,35
25	8,26	7,46	6,73
30	7,56	6,85	6,20

15.7 Quando comparado aos modelos dos Problemas 15.5 e 15.6, um modelo um pouco mais sofisticado que leva em conta os efeitos tanto da temperatura quanto do cloreto na saturação do oxigênio dissolvido pode ser considerado como

$$o = f_3(T) + f_1(c)$$

Ou seja, supõe-se que uma constante mais um polinômio de 3º grau na temperatura e uma relação linear no cloreto forneçam resultados superiores. Use a abordagem por mínimos quadrados linear geral para ajustar esse modelo aos dados da Tabela P15.5. Utilize a equação resultante para fazer uma estimativa da concentração de oxigênio dissolvido para uma concentração de cloreto de 15 g/L a $T = 12^\circ\text{C}$. Observe que o valor verdadeiro é 9,09 mg/L. Calcule o erro relativo percentual para sua previsão.

15.8 Use regressão linear múltipla para ajustar

x_1	0	1	1	2	2	3	3	4	4
x_2	0	1	2	1	2	1	2	1	2
y	15,1	17,9	12,7	25,6	20,5	35,1	29,7	45,4	40,2

Calcule os coeficientes, o erro-padrão da estimativa e o coeficiente de correlação.

15.9 Os seguintes dados foram coletados para o escoamento estacionário de água em um tubo circular de concreto:

Experimento	Diâmetro, m	Inclinação, m/m	Escoamento, m ³ /s
1	0,3	0,001	0,04
2	0,6	0,001	0,24
3	0,9	0,001	0,69
4	0,3	0,01	0,13
5	0,6	0,01	0,82
6	0,9	0,01	2,38
7	0,3	0,05	0,31
8	0,6	0,05	1,95
9	0,9	0,05	5,66

Use regressão linear múltipla para ajustar o seguinte modelo a esses dados:

$$Q = \alpha_0 D^{\alpha_1} S^{\alpha_2}$$

onde Q é o escoamento (fluxo), D é o diâmetro e S é a inclinação.

15.10 Três organismos transportadores de doenças decaem exponencialmente em água do mar de acordo com o seguinte modelo:

$$p(t) = Ae^{-1.5t} + Be^{-0.3t} + Ce^{-0.05t}$$

Utilize o método dos mínimos quadrados linear geral para estimar a população inicial de cada organismo (A , B e C) dadas as seguintes medidas:

t	0,5	1	2	3	4	5	6	7	9
p(t)	6	4,4	3,2	2,7	2	1,9	1,7	1,4	1,1

15.11 O seguinte modelo é usado para representar o efeito da radiação solar na taxa de fotossíntese de plantas aquáticas:

$$P = P_m \frac{I}{I_{sat}} e^{-\frac{I}{I_{sat}} + 1}$$

onde P é a taxa de fotossíntese (mg m⁻³d⁻¹), P_m é a taxa de fotossíntese máxima (mg m⁻³d⁻¹), I é a radiação solar (μE m⁻²s⁻¹) e I_{sat} é a radiação solar óptima (μE m⁻²s⁻¹). Use regressão não linear para calcular P_m e I_{sat} com base nos seguintes dados:

I	50	80	130	200	250	350	450	550	700
P	99	177	202	248	229	219	173	142	72

15.12 Considere os seguintes dados:

x	1	2	3	4	5
y	2,2	2,8	3,6	4,5	5,5

Ajuste o modelo a seguir a esses dados usando o MATLAB e o modelo de mínimos quadrados linear geral

$$y = a + bx + \frac{c}{x}$$

15.13 No Problema 14.8 utilizamos transformações para linearizar e ajustar o seguinte modelo:

$$y = \alpha_4 x e^{\beta_4 x}$$

Use regressão não linear para fazer uma estimativa de α_4 e β_4 com base nos dados a seguir. Faça um gráfico de seu ajuste junto com os dados.

x	0,1	0,2	0,4	0,6	0,9	1,3	1,5	1,7	1,8
y	0,75	1,25	1,45	1,25	0,85	0,55	0,35	0,28	0,18

15.14 As reações enzimáticas são usadas extensivamente para caracterizar reações mediadas biologicamente. O exemplo a seguir é um modelo usado para ajustar essas reações:

$$v_0 = \frac{k_m [S]^3}{K + [S]^3}$$

onde v_0 é a taxa inicial da reação (M/s), $[S]$ é a concentração de substrato (M) e k_m e K são parâmetros. Os seguintes dados podem ser ajustados com esse modelo:

[S], M	v_0 , M/s
0,01	$6,078 \times 10^{-11}$
0,05	$7,595 \times 10^{-9}$
0,1	$6,063 \times 10^{-8}$
0,5	$5,788 \times 10^{-6}$
1	$1,737 \times 10^{-5}$
5	$2,423 \times 10^{-5}$
10	$2,430 \times 10^{-5}$
50	$2,431 \times 10^{-5}$
100	$2,431 \times 10^{-5}$

- (a) Use uma transformação para linearizar o modelo e calcular os parâmetros. Exiba os dados e o ajuste do modelo em um gráfico.
- (b) Efetue os mesmos cálculos como em (a), mas use regressão não linear.

15.15 Considere os dados:

x	5	10	15	20	25	30	35	40	45	50
y	17	24	31	33	37	37	40	40	42	41

Use regressão por mínimos quadrados para ajustar (a) uma reta, (b) uma equação de potência, (c) uma equação da taxa de crescimento da saturação, e (d) uma parábola. Para (b) e (c) empregue transformações pra linearizar os dados. Trace os dados e todas as curvas. Alguma dessas curvas é superior? Se sim, justifique.

15.16 Os dados a seguir representam o crescimento bacteriológico em uma cultura líquida durante certo número de dias:

Day	0	4	8	12	16	20
Quantidade	67,38	74,67	82,74	91,69	101,60	112,58
$\times 10^6$						

Encontre a equação do melhor ajuste para a tendência dos dados. Tente várias possibilidades – linear, quadrática e exponencial. Determine a melhor equação para prever a quantidade de bactérias após 30 dias.

15.17 A viscosidade dinâmica da água μ (10^{-3} N·s/m 2) está relacionada com a temperatura T (°C) da seguinte maneira:

T	0	5	10	20	30	40
μ	1,787	1,519	1,307	1,002	0,7975	0,6529

(a) Trace esses dados.

(b) Use regressão polinomial para ajustar uma parábola aos dados e prever μ em $T = 7,5$ °C.

15.18 Use o método dos mínimos quadrados linear geral e o seguinte conjunto de dados de pressão-volume para encontrar as melhores constantes possíveis (A_1 e A_2) para a equação de estado mostrada a seguir. $R = 82,05$ mL atm/gmol K e $T = 303$ K.

$$\frac{PV}{RT} = 1 + \frac{A_1}{V} + \frac{A_2}{V^2}$$

P(atm)	0,985	1,108	1,363	1,631
V(mL)	25.000	22.200	18.000	15.000

15.19 Engenheiros e cientistas ambientais que lidam com o impacto da chuva ácida devem determinar o valor do produto iônico da água K_w com uma função da temperatura. Os cientistas sugeriram a seguinte equação para modelar essa relação:

$$-\log_{10} K_w = \frac{a}{T_a} + b \log_{10} T_a + c T_a + d$$

onde T_a é a temperatura absoluta (K) e a , b , c e d são parâmetros. Empregue os seguintes dados e regressão para fazer uma estimativa dos parâmetros com o MATLAB. Também, gere um gráfico de K_w previsto versus os dados.

T (°C)	K_w
0	$1,164 \times 10^{-15}$
10	$2,950 \times 10^{-15}$
20	$6,846 \times 10^{-15}$
30	$1,467 \times 10^{-14}$
40	$2,929 \times 10^{-14}$

15.20 A distância necessária para parar um automóvel consiste nas componentes tanto de pensar como de frear, cada uma delas sendo uma função de sua velocidade. Os seguintes dados experimentais foram coletados para quantificar essa relação. Determine uma equação de melhor ajuste para as componentes de pensar e frear. Utilize essas equações para fazer uma estimativa da distância de parada total para um carro que está se movendo a 110 km/h.

Velocidade, km/h	30	45	60	75	90	120
Pensar, m	5,6	8,5	11,1	14,5	16,7	22,4
Frear, m	5,0	12,3	21,0	32,9	47,6	84,7

15.21 Um investigador relatou os dados tabulados a seguir. Sabe-se que esses dados podem ser modelados pela seguinte equação

$$x = e^{(y-b)/a}$$

onde a e b são parâmetros. Use regressão não linear para determinar a e b . Com base em sua análise, faça uma previsão de y em $x = 2,6$.

x	1	2	3	4	5
y	0,5	2	2,9	3,5	4

15.22 Sabe-se que os dados tabulados a seguir podem ser modelados pela seguinte equação

$$y = \left(\frac{a + \sqrt{x}}{b \sqrt{x}} \right)^2$$

Use regressão não linear para determinar os parâmetros a e b . Com base em sua análise, faça uma previsão de y em $x = 1,6$.

x	0,5	1	2	3	4
y	10,4	5,8	3,3	2,4	2

15.23 Um pesquisador relatou os dados tabulados a seguir para uma experiência a fim de determinar a taxa de crescimento da bactéria k (por dia) como uma função da concentração de oxigênio c (mg/L). Sabe-se que esses dados podem ser modelados pela seguinte equação:

$$k = \frac{k_{\max} c^2}{c_s + c^2}$$

Use regressão não linear para estimar c_s e k_{\max} e prever a taxa de crescimento em $c = 2$ mg/L.

c	0,5	0,8	1,5	2,5	4
k	1,1	2,4	5,3	7,6	8,9

15.24 Um material é testado para falha por fadiga cíclica enquanto uma tensão, em MPa, é aplicada ao material, e o número de ciclos necessários para causar falha é medido. Os resultados estão na tabela a seguir. Use regressão não linear para ajustar um modelo de potência para esses dados.

N , ciclos	1	10	100	1.000	10.000	100.000	1.000.000
Tensão, MPa	1.100	1.000	925	800	625	550	420

15.25 Os seguintes dados mostram a relação entre a viscosidade do óleo SAE 70 e a temperatura. Use regressão não linear para ajustar uma equação de potência para esses dados.

Temperatura, T , °C	26,67	93,33	148,89	315,56
Viscosidade, μ , N · s/m²	1,35	0,085	0,012	0,00075

15.26 A concentração de bactéria *E. coli* em uma área de prática de natação é monitorada após uma tempestade

t (h)	4	8	12	16	20	24
c (UFC/100 mL)	1.590	1.320	1.000	900	650	560

O tempo é medido em horas, seguindo o fim da tempestade, e a unidade UFC é uma “unidade formadora de colônia”. Empregue regressão não linear para ajustar um modelo exponencial [Equação (14.22)] para esses dados. Use o modelo para estimar (a) a concentração no fim da tempestade ($t = 0$) e (b) o instante de tempo em que a concentração alcançará 200 UFC/100 mL.

15.27 Use regressão não linear e o seguinte conjunto de dados de pressão-volume para encontrar as melhores constantes possíveis (A_1 e A_2) para a equação de estado mostrada a seguir. $R = 82,05 \text{ mL atm/gmol K}$ e $T = 303 \text{ K}$.

$$\frac{PV}{RT} = 1 + \frac{A_1}{V} + \frac{A_2}{V^2}$$

P (atm)	0,985	1,108	1,363	1,631
V (mL)	25.000	22.200	18.000	15.000

15.28 Três organismos transportadores de doenças com mesma população inicial A decaem exponencialmente em água de lago de acordo com o seguinte modelo:

$$p(t) = Ae^{-1.5t} + Ae^{-0.3t} + Ae^{-0.05t}$$

Faça uma estimativa da população inicial A dadas as seguintes medidas:

t , hr	0,5	1	2	3	4	5	6	7	9
$p(t)$	6,0	4,4	3,2	2,7	2,2	1,9	1,7	1,4	1,1

Análise de Fourier

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é introduzir a análise de Fourier. O tema, que recebeu o nome de Joseph Fourier, envolve a identificação de ciclos e padrões dentro de uma série temporal de dados. Após esta leitura completa, você será capaz de:

- ✓ Compreender as senoides e como elas podem ser usadas para ajuste de curvas.
- ✓ Aprender como usar a regressão por mínimos quadrados para ajustar uma senoide aos dados.
- ✓ Aprender como ajustar uma série de Fourier a uma função periódica.
- ✓ Compreender a relação entre senoides e exponenciais complexas com base na fórmula de Euler.
- ✓ Reconhecer os benefícios de analisar funções matemáticas e sinais no domínio da frequência (ou seja, como uma função da frequência).
- ✓ Entender como a integral e a transformada de Fourier estendem a análise de Fourier para funções não periódicas.
- ✓ Entender como a transformada de Fourier discreta (TFD) estende a análise de Fourier para sinais discretos.
- ✓ Reconhecer como a amostragem discreta afeta a capacidade da TFD de distinguir frequências. Em particular, aprender como calcular e interpretar a frequência de Nyquist.
- ✓ Reconhecer como a transformada rápida de Fourier (FFT) fornece uma maneira altamente eficiente de calcular a TFD para casos em que o tamanho do registro de dados é uma potência de 2.
- ✓ Aprender como usar a função `fft` do MATLAB para calcular uma TFD e entender como interpretar os resultados.
- ✓ Aprender como calcular e interpretar um espectro de potência.

VOCÊ TEM UM PROBLEMA

No início do Capítulo 8, utilizamos a segunda lei de Newton e balanços de força para prever as posições de equilíbrio de três saltadores conectados por cordas. Depois, no Capítulo 13, determinamos os autovalores e os autovetores do mesmo sistema para identificar suas frequências de ressonância e modos de vibração principais. Embora essa análise certamente tenha fornecido resultados úteis, ela requereu informações detalhadas do sistema, que incluem conhecimentos do modelo subjacente e dos parâmetros (isto é, as massas dos saltadores e as constantes da mola das cordas).

Considere então que você tenha medidas das posições ou das velocidades dos saltadores em instantes de tempo discretos e igualmente espaçados (lembre-se da Figura 13.1). Essa informação é conhecida como *série temporal*. No entanto, admita ainda que você não conheça o modelo adjacente ou os parâmetros necessários para calcular os autovalores. Para esses casos, existe alguma maneira de utilizar a série temporal para aprender algo fundamental sobre a dinâmica do sistema?

Neste capítulo, descrevemos uma abordagem, a *análise de Fourier*, que fornece um meio de alcançar esse objetivo. Essa abordagem é baseada na premissa de que funções mais complicadas (por exemplo, séries temporais) podem ser representadas pela soma de funções trigonométricas mais simples. Como um prelúdio para indicar a forma como isso é feito, é útil explorar como dados podem ser ajustados com funções senoidais.

16.1 AJUSTE DE CURVAS COM FUNÇÕES SENOIDAIS

Uma função periódica $f(t)$ é uma para a qual

$$f(t) = f(t + T) \quad (16.1)$$

onde T é uma constante chamada *período*, que é o menor intervalo de tempo para o qual a Equação (16.1) é válida. Exemplos comuns incluem tanto sinais artificiais como naturais (Figura 16.1).

As mais fundamentais são as funções senoidais. Na presente discussão, será usado o termo *senoide* para representar qualquer forma ondulatória que possa ser descrita por um seno ou um cosseno. Não existe nenhuma convenção bem estabelecida para escolher uma das duas funções e, de todo modo, os resultados seriam idênticos, porque as duas funções estão simplesmente deslocadas no tempo por $\pi/2$ radianos. Neste capítulo, será usado o cosseno, que pode ser representado, em geral, por

$$f(t) = A_0 + C_1 \cos(\omega_0 t + \theta) \quad (16.2)$$

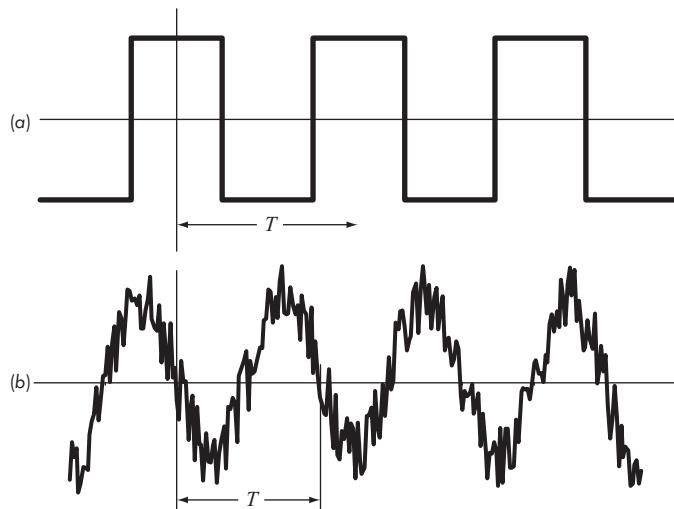


FIGURA 16.1 Além das funções trigonométricas como o seno e o cosseno, as funções periódicas incluem formas de onda idealizadas, como a onda quadrada apresentada em (a). Além dessas formas artificiais, sinais periódicos por natureza podem ser contaminados por ruídos, como as temperaturas do ar mostradas em (b).

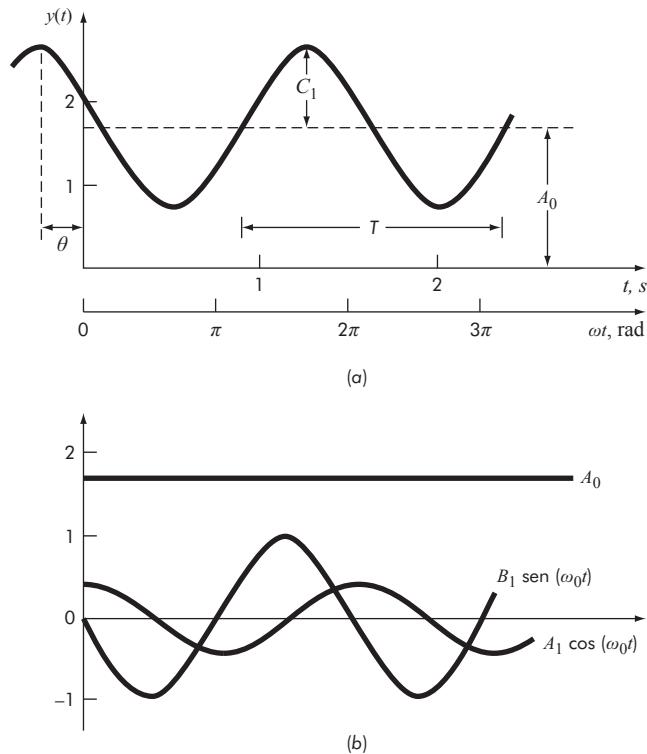


FIGURA 16.2 (a) Gráfico da função senoidal $y(t) = A_0 + C_1 \cos(\omega_0 t + \theta)$. Nesse caso, $A_0 = 1,7$; $C_1 = 1$; $\omega_0 = 2\pi/T = 2\pi/(1,5 \text{ s})$ e $\theta = \pi/3$ radianos = 1,0472 (= 0,25 s). Os outros parâmetros usados para descrever a curva são a frequência $f = \omega_0/(2\pi)$, que nesse caso é 1 ciclo/(1,5 s) = 0,6667 Hz e o período é $T = 1,5 \text{ s}$. (b) Uma expressão alternativa da mesma curva é $y(t) = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t)$. As três componentes dessa função são apresentadas em (b), onde $A_1 = 0,5$ e $B_1 = -0,866$. A soma das três curvas em (b) fornece a curva única em (a).

Uma inspeção da Equação (16.2) indica que quatro parâmetros servem para caracterizar unicamente a senoide (Figura 16.2a):

- ✓ O *valor médio* A_0 determina a altura média acima da abscissa.
- ✓ A *amplitude* C_1 especifica a altura da oscilação.
- ✓ A *frequência angular* ω_0 caracteriza com que frequência os ciclos ocorrem.
- ✓ O *ângulo de fase* (ou *deslocamento de fase*) θ parametriza a extensão pela qual a curva senoidal está deslocada horizontalmente.

Observe que a *frequência angular* (em radianos/tempo) está relacionada com a *frequência f* (em ciclos/tempo)¹ por

$$\omega_0 = 2\pi f \quad (16.3)$$

¹ Quando o tempo está em segundos, a unidade para a frequência é um ciclo/s ou *Hertz* (Hz).

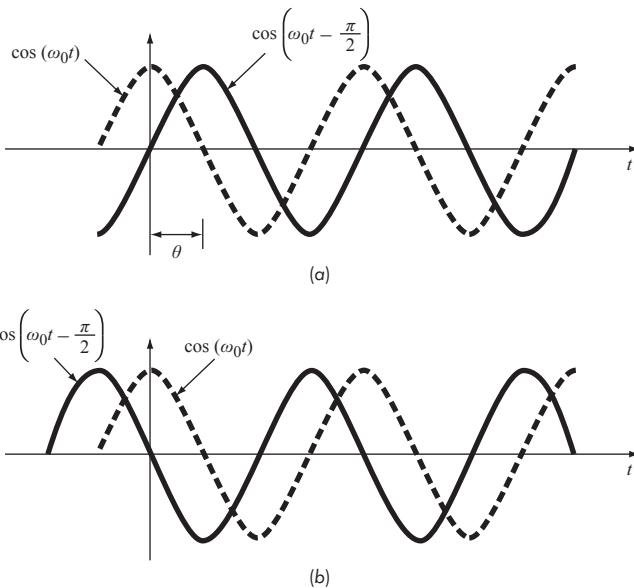


FIGURA 16.3 Descrição gráfica de (a) um ângulo de fase de atraso e de (b) um ângulo de fase de avanço. Observe que a curva atrasada em (a) pode ser descrita alternativamente como $\cos(\omega_0 t + 3\pi/2)$; ou seja, se uma curva tem um atraso por um ângulo α , ela também pode ser representada como tendo um avanço $2\pi - \alpha$.

e a frequência, por sua vez, está relacionada ao período T por

$$f = \frac{1}{T} \quad (16.4)$$

Além disso, o *ângulo de fase* representa a distância em radianos de $t = 0$ ao ponto no qual a função cosseno começa um novo ciclo. Como descreve a Figura 16.3a, um valor negativo é chamado de *ângulo de fase de atraso*, porque a curva $\cos(\omega_0 t - \theta)$ começa um novo ciclo θ radianos depois de $\cos(\omega_0 t)$. Logo, diz-se que $\cos(\omega_0 t - \theta)$ está atrasado em relação à $\cos(\omega_0 t)$. Reciprocamente, como na Figura 16.3b, um valor positivo é chamado de *ângulo de fase de avanço*.

Embora a Equação (16.2) seja uma caracterização matematicamente adequada de uma função senoidal, é complicado trabalhar com ela em termos de ajuste de curvas, porque o deslocamento de fase está incluído no argumento da função cosseno. Essa deficiência pode ser superada com o uso da identidade trigonométrica:

$$C_1 \cos(\omega_0 t + \theta) = C_1 [\cos(\omega_0 t + \theta) \cos(\theta) - \sin(\omega_0 t + \theta) \sin(\theta)] \quad (16.5)$$

Substituindo a Equação (16.5) na Equação (16.2) e agrupando os termos, obtém-se (Figura 16.2b)

$$f(t) = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t) \quad (16.6)$$

onde

$$A_1 = C_1 \cos(\theta) \quad B_1 = -C_1 \sin(\theta) \quad (16.7)$$

Dividindo-se as duas partes da Equação (16.7) obtém-se

$$\theta = \arctan\left(-\frac{B_1}{A_1}\right) \quad (16.8)$$

onde, se $A_1 < 0$, soma-se π a 0. Elevando ao quadrado e somando, a Equação (16.7) leva a

$$C_1 = \sqrt{A_1^2 + B_1^2} \quad (16.9)$$

Logo, a Equação (16.6) representa uma formulação alternativa da Equação (16.2), que ainda usa quatro parâmetros, mas que está na forma de um modelo linear geral [lembre-se da Equação (15.7)]. Como será discutido na próxima seção, ela pode ser aplicada de modo simples como a base para um ajuste por mínimos quadrados.

Antes de prosseguir para a próxima seção, entretanto, deve-se enfatizar que poderíamos ter usado uma função seno em vez do cosseno no modelo fundamental da Equação (16.2); por exemplo

$$f(t) = A_0 + C_1 \sin(\omega_0 t + \delta)$$

Relações simples podem ser aplicadas para converter entre as duas formas:

$$\sin(\omega_0 t + \delta) = \cos\left(\omega_0 t + \delta - \frac{\pi}{2}\right)$$

e

$$\cos(\omega_0 t + \delta) = \sin\left(\omega_0 t + \delta + \frac{\pi}{2}\right) \quad (16.10)$$

Ou seja, $\theta = \delta - \pi/2$. A única consideração importante é que uma ou outra forma deve ser usada de modo consistente. Assim, será usada a versão cosseno em toda a discussão.

16.1.1 Ajuste por mínimos quadrados de uma curva senoidal

A Equação (16.6) pode ser compreendida como um modelo linear de mínimos quadrados:

$$y = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t) + e \quad (16.11)$$

o qual é simplesmente um outro exemplo do modelo geral [lembre-se da Equação (15.7)]

$$y = a_0 z_0 + a_1 z_1 + a_2 z_2 + \cdots + a_m z_m + e$$

onde $z_0 = 1$, $z_1 = \cos(\omega_0 t)$, $z_2 = \sin(\omega_0 t)$ e todos os outros z 's = 0. Logo, o objetivo é determinar valores dos coeficientes que minimizem

$$S_r = \sum_{i=1}^N \{y_i - [A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t)]\}^2$$

As equações normais para conseguir essa minimização podem ser expressas na forma matricial como [lembre-se da Equação (15.10)]

$$\begin{bmatrix} N & \Sigma \cos(\omega_0 t) & \Sigma \sin(\omega_0 t) \\ \Sigma \cos(\omega_0 t) & \Sigma \cos^2(\omega_0 t) & \Sigma \cos(\omega_0 t) \sin(\omega_0 t) \\ \Sigma \sin(\omega_0 t) & \Sigma \cos(\omega_0 t) \sin(\omega_0 t) & \Sigma \sin^2(\omega_0 t) \end{bmatrix} \begin{Bmatrix} A_0 \\ B_1 \\ B_2 \end{Bmatrix} = \begin{Bmatrix} \Sigma y \\ \Sigma y \cos(\omega_0 t) \\ \Sigma y \sin(\omega_0 t) \end{Bmatrix} \quad (16.12)$$

Essas equações podem ser usadas para determinar os coeficientes desconhecidos. Entretanto, em vez de fazer isso, pode-se examinar o caso especial em que existem N observações igualmente espaçadas em intervalos Δt e com um intervalo total de $T = (N - 1)\Delta t$. Nessa situação, podem ser determinados os seguintes valores médios (ver Problema 16.3):

$$\begin{aligned} \frac{\Sigma \sin(\omega_0 t)}{N} &= 0 & \frac{\Sigma \cos(\omega_0 t)}{N} &= 0 \\ \frac{\Sigma \sin^2(\omega_0 t)}{N} &= \frac{1}{2} & \frac{\Sigma \cos^2(\omega_0 t)}{N} &= \frac{1}{2} \\ \frac{\Sigma \cos(\omega_0 t) \sin(\omega_0 t)}{N} &= 0 \end{aligned} \quad (16.13)$$

Então, para pontos uniformemente espaçados, as equações normais se tornam

$$\begin{bmatrix} N & 0 & 0 \\ 0 & N/2 & 0 \\ 0 & 0 & N/2 \end{bmatrix} \begin{Bmatrix} A_0 \\ B_1 \\ B_2 \end{Bmatrix} = \begin{Bmatrix} \Sigma y \\ \Sigma y \cos(\omega_0 t) \\ \Sigma y \sin(\omega_0 t) \end{Bmatrix}$$

A inversa de uma matriz diagonal é simplesmente uma outra matriz diagonal cujos elementos são os inversos da matriz original. Logo, os coeficientes podem ser determinados por

$$\begin{Bmatrix} A_0 \\ B_1 \\ B_2 \end{Bmatrix} = \begin{bmatrix} 1/N & 0 & 0 \\ 0 & 2/N & 0 \\ 0 & 0 & 2/N \end{bmatrix} \begin{Bmatrix} \Sigma y \\ \Sigma y \cos(\omega_0 t) \\ \Sigma y \sin(\omega_0 t) \end{Bmatrix}$$

ou

$$A_0 = \frac{\Sigma y}{N} \quad (16.14)$$

$$B_1 = \frac{2}{N} \Sigma y \cos(\omega_0 t) \quad (16.15)$$

$$B_2 = \frac{2}{N} \Sigma y \sin(\omega_0 t) \quad (16.16)$$

Observe que o primeiro coeficiente representa o valor médio da função.

EXEMPLO 16.1 Ajuste por mínimos quadrados de uma curva senoidal

A curva na Figura 16.2a é descrita por $y = 1,7 + \cos(4,189t + 1,0472)$. Gere 10 valores discretos para essa curva em intervalos de $\Delta t = 0,15$ para variação de $t = 0$ a $1,35$. Use essa informação para calcular os coeficientes da Equação (16.11) por um ajuste por mínimos quadrados.

Solução. Os dados necessários para o cálculo dos coeficientes com $\omega = 4,189$ são

t	y	$y \cos(\omega_0 t)$	$y \sin(\omega_0 t)$
0	2,200	2,200	0,000
0,15	1,595	1,291	0,938
0,30	1,031	0,319	0,980
0,45	0,722	-0,223	0,687
0,60	0,786	-0,636	0,462
0,75	1,200	-1,200	0,000
0,90	1,805	-1,460	-1,061
1,05	2,369	-0,732	-2,253
1,20	2,678	0,829	-2,547
1,35	2,614	2,114	-1,536
$\Sigma =$	17,000	2,502	-4,330

Esses resultados podem ser usados para determinar [Equações (16.14) a (16.16)]

$$A_0 = \frac{17,000}{10} = 1,7 \quad A_1 = \frac{2}{10} 2,502 = 0,500 \quad B_1 = \frac{2}{10} (-4,330) = -0,866$$

Logo, o ajuste por mínimos quadrados é

$$y = 1,7 + 0,500 \cos(\omega_0 t) - 0,866 \sin(\omega_0 t)$$

O modelo também pode ser expresso na forma da Equação (16.2) calculando-se [Equação (16.8)]

$$\theta = \arctan\left(\frac{-0,866}{0,500}\right) = 1,0472$$

e [Equação (16.9)]

$$C_1 = \sqrt{0,5^2 + (-0,866)^2} = 1,00$$

para obter

$$y = 1,7 + \cos(\omega_0 t + 1,0472)$$

ou, alternativamente, como uma função seno, usando-se a Equação (16.10)

$$y = 1,7 + \sin(\omega_0 t + 2,618)$$

A análise anterior pode ser estendida ao modelo geral

$$f(t) = A_0 + A_1 \cos(\omega_0 t) + B_1 \sin(\omega_0 t) + A_2 \cos(2\omega_0 t) + B_2 \sin(2\omega_0 t) + \dots + A_m \cos(m\omega_0 t) + B_m \sin(m\omega_0 t)$$

onde, para dados igualmente espaçados, os coeficientes podem ser calculados por

$$\left. \begin{aligned} A_0 &= \frac{\sum y}{N} \\ A_j &= \frac{2}{N} \sum y \cos(j\omega_0 t) \\ B_j &= \frac{2}{N} \sum y \sin(j\omega_0 t) \end{aligned} \right\} \quad j = 1, 2, \dots, m$$

Embora essas relações possam ser usadas para ajustar dados no sentido de regressão (isto é, $N > 2m + 1$), uma aplicação alternativa é usá-las para interpolação ou colocação – ou seja, usá-las para o caso em que o número de incógnitas, $2m + 1$, é igual ao número de pontos dados, N . Essa é a abordagem utilizada na série de Fourier contínua, como descrito a seguir.

16.2 SÉRIE DE FOURIER CONTÍNUA

No decorrer do estudo de problemas de fluxo de calor, Fourier mostrou que uma função periódica arbitrária pode ser representada por uma série infinita de funções senoidais com frequências harmonicamente relacionadas. Para uma função de período T , uma série de Fourier contínua pode ser escrita

$$f(t) = a_0 + a_1 \cos(\omega_0 t) + b_1 \sin(\omega_0 t) + a_2 \cos(2\omega_0 t) + b_2 \sin(2\omega_0 t) + \dots$$

ou, mais concisamente,

$$f(t) = a_0 + \sum_{k=1}^{\infty} [a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)] \quad (16.17)$$

onde a frequência angular do primeiro modo ($\omega_0 = 2\pi/T$) é chamada *frequência fundamental*, e os seus múltiplos constantes $2\omega_0$, $3\omega_0$, etc. são chamados *harmônicos*. Logo, a Equação (16.17) expressa $f(t)$ como uma combinação linear das funções base: 1, $\cos(\omega_0 t)$, $\sin(\omega_0 t)$, $\cos(2\omega_0 t)$, $\sin(2\omega_0 t)$, ...

Os coeficientes da Equação (16.17) podem ser calculados por

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega_0 t) dt \quad (16.18)$$

e

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega_0 t) dt \quad (16.19)$$

para $k = 1, 2, \dots$ e

$$a_0 = \frac{1}{T} \int_0^T f(t) dt \quad (16.20)$$

EXEMPLO 16.2 Aproximação por série de Fourier contínua

Use a série de Fourier contínua para aproximar a função de onda quadrada ou retangular (Figura 16.1a) com uma altura de 2 e um período $T = 2\pi/\omega_0$:

$$f(t) = \begin{cases} -1 & -T/2 < t < -T/4 \\ 1 & -T/4 < t < T/4 \\ -1 & T/4 < t < T/2 \end{cases}$$

Solução. Como a altura média da onda é zero, o valor de $a_0 = 0$ pode ser obtido de modo direto. Os coeficientes restantes podem ser calculados por [Equação (16.18)]

$$\begin{aligned} a_k &= \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(k\omega_0 t) dt \\ &= \frac{2}{T} \left[- \int_{-T/2}^{-T/4} \cos(k\omega_0 t) dt + \int_{-T/4}^{T/4} \cos(k\omega_0 t) dt - \int_{T/4}^{T/2} \cos(k\omega_0 t) dt \right] \end{aligned}$$

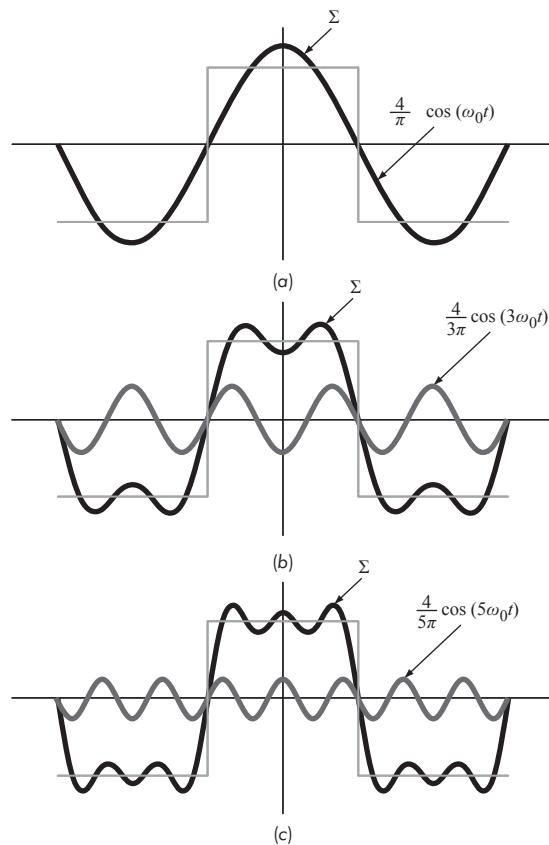


FIGURA 16.4 Aproximação por série de Fourier de uma onda quadrada. A série de gráficos mostra a somatória até, e incluindo, (a) o primeiro, (b) o segundo e (c) o terceiro termos. Os termos individuais que foram somados ou subtraídos em cada estágio também são mostrados.

As integrais podem ser calculadas, fornecendo

$$a_k = \begin{cases} 4/(k\pi) & \text{para } k = 1, 5, 9, \dots \\ -4/(k\pi) & \text{para } k = 3, 7, 11, \dots \\ 0 & \text{para } k = \text{inteiros pares} \end{cases}$$

Analogamente, pode ser determinado que todos os b 's são nulos; portanto, a aproximação por série de Fourier é

$$f(t) = \frac{4}{\pi} \cos(\omega_0 t) - \frac{4}{3\pi} \cos(3\omega_0 t) + \frac{4}{5\pi} \cos(5\omega_0 t) - \frac{4}{7\pi} \cos(7\omega_0 t) + \dots$$

Os resultados até os três primeiros termos são apresentados na Figura 16.4.

A série de Fourier também pode ser expressa em uma forma mais compacta, utilizando notação complexa. Isso é baseado na *fórmula de Euler* (Figura 16.5):

$$e^{\pm ix} = \cos x \pm i \sin x \quad (16.21)$$

onde $i = \sqrt{-1}$ e x está em radianos. A Equação (16.21) pode ser usada para expressar a série de Fourier concisamente como (Chapra e Canale, 2010)

$$f(t) = \sum_{k=-\infty}^{\infty} \tilde{c}_k e^{ik\omega_0 t} \quad (16.22)$$

onde os coeficientes são

$$\tilde{c}_k = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-ik\omega_0 t} dt \quad (16.23)$$

Observe que os tils ~ são incluídos para reforçar que os coeficientes são números complexos. Como ela é mais concisa, utilizaremos principalmente a forma complexa no restante do capítulo. Basta lembrar que ela é idêntica à representação senoidal.

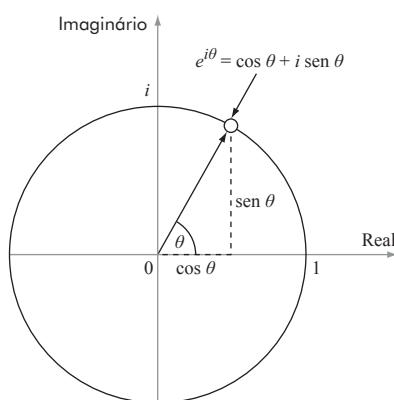


FIGURA 16.5 Descrição gráfica da fórmula de Euler. O vetor girante é chamado fasor.

16.3 DOMÍNIOS DO TEMPO E DA FREQUÊNCIA

Até este ponto, a discussão da análise de Fourier se limitou ao *domínio do tempo*. Isso foi feito porque a maioria de nós fica bastante confortável idealizando o comportamento de uma função nessa dimensão. Embora não seja tão familiar, o *domínio da frequência* fornece uma perspectiva alternativa para caracterizar o comportamento de funções oscilantes.

Assim como a amplitude pode ser traçada em função do tempo, ela também pode ser traçada em função da frequência. Ambos os tipos de expressão estão descritos na Figura 16.6a, na qual foi desenhado um gráfico tridimensional da função senoidal:

$$f(t) = C_1 \cos\left(t + \frac{\pi}{2}\right)$$

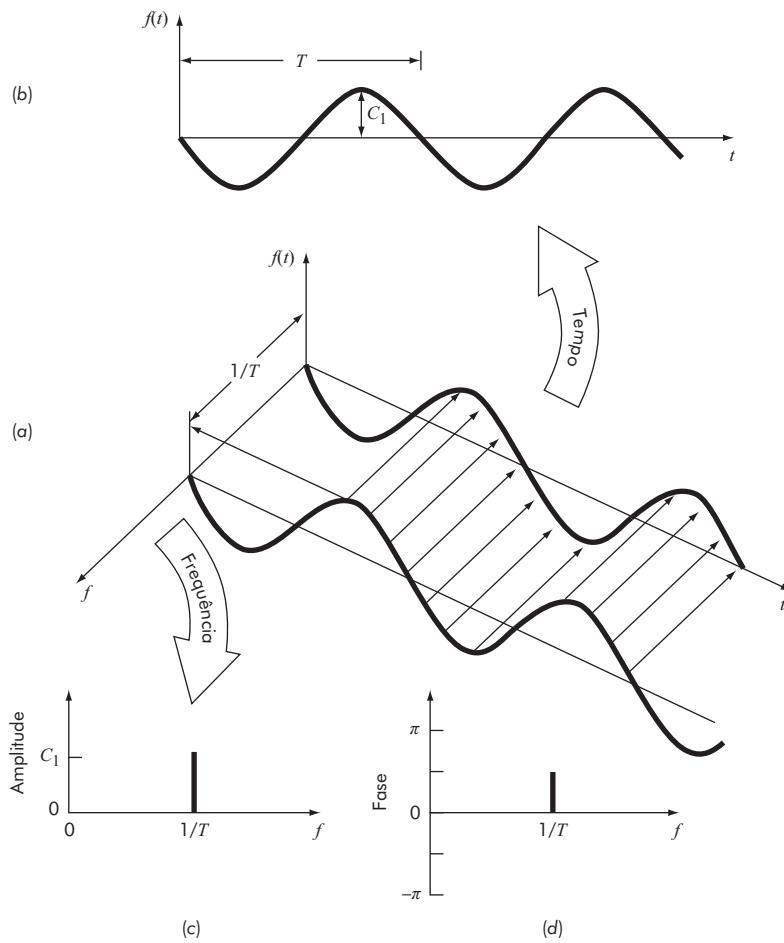


FIGURA 16.6 (a) Descrição de como uma função senoidal pode ser retratada nos domínios do tempo e da frequência. A projeção temporal é reproduzida em (b), enquanto a projeção amplitude-frequência é reproduzida em (c). A projeção fase-frequência é mostrada em (d).

Nesse gráfico, a amplitude da curva $f(t)$ é a variável dependente, e o tempo t e a frequência $f = \omega_0/2\pi$ são as variáveis independentes. Logo, os eixos da amplitude e do tempo formam um *plano temporal*, e os eixos da amplitude e da frequência formam um *plano de frequências*. A função senoidal pode, portanto, ser concebida como existindo a uma distância $1/T$ ao longo do eixo de frequência e correndo paralela ao eixo do tempo. Consequentemente, quando mencionamos o comportamento de uma função senoidal no domínio do tempo, nos referimos à projeção da curva sobre o plano temporal (Figura 16.6b). De modo similar, o comportamento no domínio da frequência é simplesmente a projeção no plano da frequência.

Como na Figura 16.6c, essa projeção é uma medida da amplitude positiva máxima da função senoidal C_1 . A variação total pico a pico é desnecessária devido à simetria. Juntamente com a posição $1/T$ ao longo do eixo de frequência, a Figura 16.6c agora define a amplitude e a frequência da função senoidal. Isso é informação suficiente para reproduzir a forma da curva no domínio do tempo. Contudo, um parâmetro a mais – isto é, o ângulo de fase – é necessário para posicionar a curva em relação a $t = 0$. Consequentemente, um diagrama de fase, como mostra a Figura 16.6d, também deve ser incluído. O ângulo de fase é determinado como a distância (em radianos) do zero ao ponto no qual um pico positivo ocorre. Se o pico ocorrer depois do zero, ele é chamado atrasado (lembre-se da discussão de atrasos ou de avanços na Seção 16.1), e, por convenção, o ângulo de fase é dado com um sinal negativo. Reciprocamente, um pico antes do zero é chamado avançado (ou adiantado) e o ângulo de fase é positivo. Logo, para a Figura 16.6, o pico está avançado (adiantado) com relação ao zero, e o ângulo de fase é marcado como $+\pi/2$. A Figura 16.7 descreve algumas outras possibilidades.

Pode-se ver agora que as Figuras 16.6c e d fornecem uma forma alternativa de apresentar ou resumir as características pertinentes da curva senoidal da Figura 16.6a. Elas são chamadas de *linhas espectrais*. É claro que, para uma única função senoidal, elas não são muito interessantes. Entretanto, quando aplicadas a situações mais complicadas, como uma série de Fourier, seus verdadeiros poder e valor são revelados. Por exemplo, a Figura 16.8 mostra as linhas espectrais de amplitude e de fase para a função onda quadrada do Exemplo 16.2.

Esses espectros fornecem informações que não seriam aparentes a partir do domínio do tempo. Isso pode ser visto comparando-se as Figuras 16.4 e 16.8. A Figura 16.4 apresenta duas perspectivas alternativas do domínio do tempo. A primeira, a onda quadrada original, não mostra nada sobre as curvas senoidais que a compõem. A alternativa é mostrar essas funções senoidais – ou seja, $(4/\pi) \cos(\omega_0 t)$, $-(4/3\pi) \cos(3\omega_0 t)$, $(4/5\pi) \cos(5\omega_0 t)$, etc. Tal alternativa não fornece uma visualização adequada da estrutura desses harmônicos. Já as Figuras 16.8a e b fornecem uma descrição gráfica dessa estrutura. Como tal, a linha espectral representa “impressões digitais” que ajudam a caracterizar e a entender uma forma de onda complicada. Elas são particularmente valiosas nos casos não idealizados, em que, algumas vezes, é permitido discernir a estrutura em sinais que de outra forma seriam obscuros. Na próxima seção, será descrita a transformada de Fourier que permitirá estender tais análises a formas de onda não periódicas.

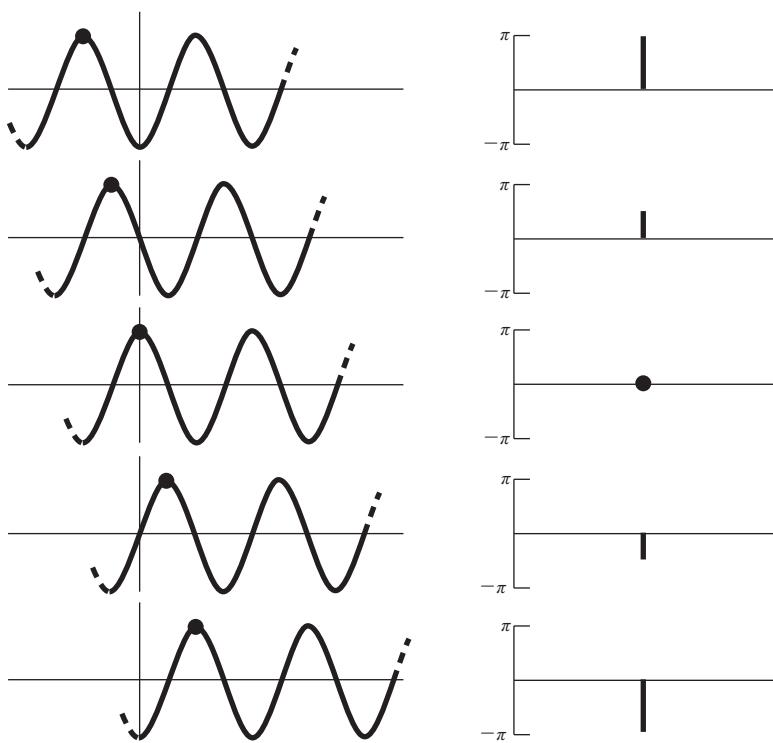
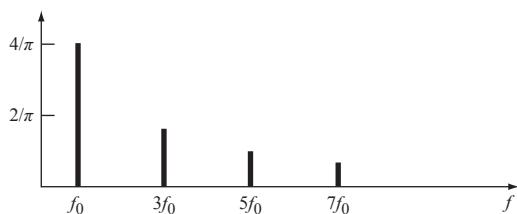
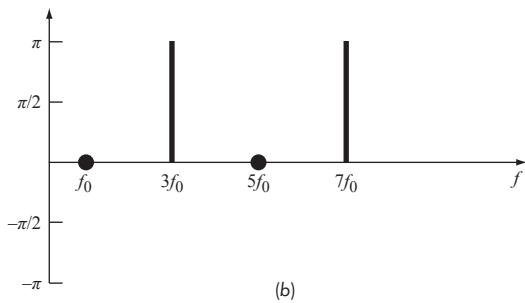


FIGURA 16.7 Várias fases de uma função senoidal que mostra as linhas espectrais de fase associadas.



(a)



(b)

FIGURA 16.8 Linhas espectrais de (a) amplitude e (b) fase para a onda quadrada da Figura 16.4.

16.4 INTEGRAL E TRANSFORMADA DE FOURIER

Embora a série de Fourier seja uma ferramenta útil na investigação de funções periódicas, há muitas formas de onda que não se repetem regularmente. Por exemplo, um raio (descarga atmosférica) ocorre apenas uma vez (ou, pelo menos, passará um longo tempo antes que ocorra novamente), mas causará interferência nos receptores operando em uma larga escala de frequências – como TVs, rádios e receptores de ondas curtas. Essas evidências sugerem que um sinal não recorrente como o produzido por um raio exibe um espectro de frequência contínuo. Como esses fenômenos são de grande interesse para os engenheiros, uma alternativa para a série de Fourier é valiosa para analisar essas ondas não periódicas.

A *integral de Fourier* é a ferramenta principal disponível para esses propósitos. Ela pode ser deduzida da forma exponencial da série de Fourier [Equações (16.22) e (16.23)]. A transição de uma função periódica para uma função não periódica pode ser feita permitindo-se que o período tenda a infinito; ou seja, quando T se torna infinito, a função nunca se repete e, assim, torna-se não periódica. Se for permitido que isso ocorra, pode ser demonstrado (por exemplo, Van Valkenburg, 1974; Hayt e Kemmerly, 1986) que a série de Fourier se reduz a

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \quad (16.24)$$

e os coeficientes se tornam uma função contínua da variável ω , como em

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (16.25)$$

A função $F(\omega)$, como definida pela Equação (16.25), é chamada *integral de Fourier* de $f(t)$. Além disso, as Equações (16.24) e (16.25) são conhecidas coletivamente como *par da transformada de Fourier*. Assim, além de ser chamada de integral de Fourier, $F(\omega)$ também é chamada de *transformada de Fourier* de $f(t)$. No mesmo propósito, $f(t)$, como foi definido pela Equação (16.24), é conhecida como a *transformada de Fourier inversa* de $F(\omega)$. Então, o par permite transformar para frente e para trás entre os domínios do tempo e da frequência para um sinal não periódico.

A distinção entre a série e a transformada de Fourier deveria estar bem clara. A principal diferença é que cada uma se aplica a uma classe diferente de função – a série, a formas de onda periódicas, e a transformada, a formas não periódicas. Além dessa diferença principal, as duas abordagens diferem na forma como se movem entre os domínios do tempo e da frequência. A série de Fourier converte uma função contínua periódica no domínio do tempo para amplitudes no domínio da frequência, em frequências discretas. Já a transformada de Fourier converte uma função contínua no domínio do tempo em uma função contínua no domínio da frequência. Assim, o espectro de frequência discreto gerado pela série de Fourier é análogo ao espectro de frequência contínuo gerado pela transformada de Fourier.

Depois de introduzida essa forma de analisar um sinal não periódico, concluiremos nossa dedução. Na próxima seção, será mostrado que um sinal é raramente caracterizado como uma função contínua do tipo necessário para implementar a Equação (16.25). Em vez disso, os dados estão invariavelmente na forma discreta.

Assim, mostraremos agora como calcular uma transformada de Fourier para tais medidas discretas.

16.5 TRANSFORMADA DE FOURIER DISCRETA (TFD)

Em engenharia, as funções são, em geral, representadas por conjuntos finitos de valores discretos. Além disso, geralmente, os dados são coletados ou convertidos para tal forma discreta. Como descreve a Figura 16.9, um intervalo de 0 a T pode ser dividido em n subintervalos com largura $\Delta t = T/n$. O subscrito j é usado para designar os instantes discretos nos quais as medidas foram tomadas. Assim, f_j designa o valor da função contínua $f(t)$ no instante t_j . Observe que os pontos dados estão especificados em $j = 0, 1, 2, \dots, n - 1$. Não está incluído um valor em $j = n$. (Ver Ramirez, 1985, para as razões para exclusão de f_n)

Para o sistema na Figura 16.9, uma transformada de Fourier discreta pode ser escrita como

$$F_k = \sum_{j=0}^{n-1} f_j e^{-ik\omega_0 j} \quad \text{para } k = 0 \text{ a } n - 1 \quad (16.26)$$

e a transformada inversa de Fourier como

$$f_j = \frac{1}{n} \sum_{k=0}^{n-1} F_k e^{ik\omega_0 j} \quad \text{para } j = 0 \text{ a } n - 1 \quad (16.27)$$

onde $\omega_0 = 2\pi/n$.

As Equações (16.26) e (16.27) representam o análogo discreto das Equações (16.25) e (16.24), respectivamente. Como tal, elas podem ser usadas para calcular tanto a transformada de Fourier direta quanto a inversa para dados discretos. Observe que o fator $1/n$ na Equação (16.27) é simplesmente um fator de escala que pode ser incluído ou na Equação (16.26) ou na Equação (16.27), mas não em ambas. Por exemplo, se ele é transferido para a Equação (16.26), o primeiro coeficiente F_0 (que é o análogo da constante a_0) é igual à média aritmética das amostras.

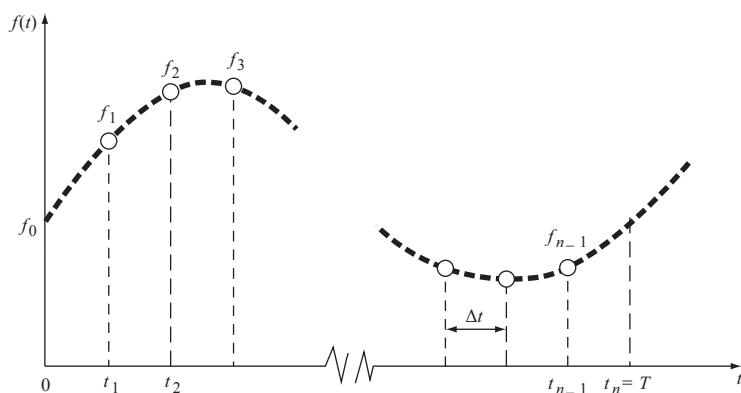


FIGURA 16.9 Pontos amostrais na série de Fourier discreta.

Antes de prosseguir, alguns aspectos da TFD devem ser mencionados. A maior frequência que pode ser medida em um sinal, chamada *frequência de Nyquist*, é metade da frequência de amostragem. Variações periódicas que ocorram mais rapidamente que o menor intervalo de tempo de amostragem não podem ser detectadas. A menor frequência que pode ser detectada é o inverso do tamanho total da amostra.

Como um exemplo, considere que você tome 100 amostras de dados ($n = 100$ amostras) em uma frequência (ou taxa) de amostragem $f_s = 1.000 \text{ Hz}$ (isto é, 1.000 amostras por segundo). Isso significa que o intervalo de amostragem é

$$\Delta t = \frac{1}{f_s} = \frac{1}{1.000 \text{ amostras/s}} = 0,001 \text{ s/amostra}$$

O tamanho total da amostra é

$$t_n = \frac{n}{f_s} = \frac{100 \text{ amostras}}{1.000 \text{ amostras/s}} = 0,1 \text{ s}$$

e o incremento de frequência é

$$\Delta f = \frac{f_s}{n} = \frac{1.000 \text{ amostras/s}}{100 \text{ amostras}} = 10 \text{ Hz}$$

A frequência de Nyquist é

$$f_{\max} = 0,5 f_s = 0,5(1.000 \text{ Hz}) = 500 \text{ Hz}$$

e a menor frequência detectável é

$$f_{\min} = \frac{1}{0,1 \text{ s}} = 10 \text{ Hz}$$

Assim, para esse exemplo, a TFD poderia detectar sinais com períodos de $1/500 = 0,002 \text{ s}$ até $1/10 = 0,1 \text{ s}$.

16.5.1 Transformada Rápida de Fourier (FFT)

Embora um algoritmo possa ser desenvolvido para calcular a TFD com base na Equação (16.26), ele é computacionalmente pesado, pois são necessárias n^2 operações. Consequentemente, para amostras de dados mesmo de tamanho moderado, a determinação direta da TFD pode demandar muito tempo.

A *transformada rápida de Fourier*, ou *FFT*,* é um algoritmo que foi desenvolvido para calcular a TFD de uma forma extremamente econômica. Sua velocidade é decorrente do fato de que ela utiliza os resultados dos cálculos anteriores para reduzir o número de operações. Especificamente, ela explora a periodicidade e a simetria das funções trigonométricas para calcular a transformada com aproximadamente $n \log_2 n$ operações (Figura 16.10). Logo, para $n = 50$ amostras, a FFT é da ordem de 10 vezes mais rápida do que a TFD padrão. Para $n = 1.000$, ela é cerca de 100 vezes mais rápida.

* N. de T.: FFT, em inglês *Fast Fourier Transform*. Embora a sigla se refira a um termo em inglês, seu uso é bastante comum no Brasil para indicar a Transformada Rápida de Fourier.

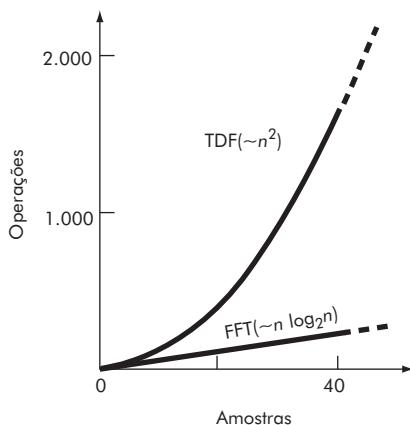


FIGURA 16.10 Gráfico do número de operações em função do tamanho da amostra para a TFD padrão e para a FFT.

O primeiro algoritmo do tipo FFT foi desenvolvido por Gauss no início do século XIX (Heideman et al., 1984). Outras importantes contribuições foram feitas por Runge, Danielson, Lanczos e outros no início do século XX. No entanto, como transformadas discretas muitas vezes levavam dias ou semanas para ser calculadas manualmente, elas não atraíram grande interesse antes do desenvolvimento do computador digital moderno.

Em 1965, J. W. Cooley e J. W. Tukey publicaram um artigo-chave no qual esboçaram um algoritmo para calcular a FFT. Esse esquema, que é parecido com aquele de Gauss e outros pesquisadores anteriores, é chamado de algoritmo de Cooley-Tukey. hoje, há muitas outras abordagens que são desdobramentos desse método. Como descrito a seguir, o MATLAB oferece uma função chamada `fft` que emprega esses algoritmos eficientes para calcular a TFD.

16.5.2 Função do MATLAB: `fft`

A função do MATLAB `fft` fornece uma maneira eficiente para calcular a TFD. Uma representação simples de sua sintaxe é

$$\mathbf{F} = \text{fft}(\mathbf{f}, \mathbf{n})$$

onde \mathbf{F} é um vetor que contém a TFD e \mathbf{f} é um vetor que contém o sinal. O parâmetro \mathbf{n} , que é opcional, indica que o usuário quer implementar uma FFT de n pontos. O vetor \mathbf{f} é preenchido com zeros, se tiver menos que n pontos, e é truncado, se tiver mais.

Observe que os elementos em \mathbf{F} são sequenciados no que é chamada de *ordem envolvente reversa*. A primeira metade dos valores são as frequências positivas (começando com a constante), e a segunda metade são as frequências negativas. Assim, se $n = 8$, a ordem é $0, 1, 2, 3, 4, -3, -2, -1$. O seguinte exemplo ilustra o uso da função para calcular a TFD de uma senoide simples.

EXEMPLO 16.3 Cálculo da TFD de uma senoide simples com o MATLAB

Aplique a função `fft` do MATLAB para determinar a transformada de Fourier discreta para uma senoide simples:

$$f(t) = 5 + \cos(2\pi(12,5)t) + \sin(2\pi(18,75)t)$$

Gere 8 pontos equidistantes com $\Delta t = 0,02$ s. Trace os resultados *versus* a frequência.

Solução. Antes de gerar a TFD, devemos calcular uma série de quantidades. A frequência de amostragem é

$$f_s = \frac{1}{\Delta t} = \frac{1}{0,02\text{ s}} = 50\text{ Hz}$$

O tamanho total da amostra é

$$t_n = \frac{n}{f_s} = \frac{8\text{ amostras}}{50\text{ amostras/s}} = 0,16\text{ s}$$

A frequência de Nyquist é

$$f_{\max} = 0,5f_s = 0,5(50\text{ Hz}) = 25\text{ Hz}$$

e a menor frequência detectável é

$$f_{\min} = \frac{1}{0,16\text{ s}} = 6,25\text{ Hz}$$

Assim, a análise pode detectar sinais com períodos de $1/25 = 0,04$ s até $1/6,25 = 0,16$ s. Portanto, devemos ser capazes de detectar ambos os sinais de 12,5 e 18,75 Hz.

Os comandos de MATLAB a seguir podem ser usados para gerar e traçar um gráfico da amostra (Figura 16.11a):

```
>> clc
>> n=8; dt=0.02; fs=1/dt; T=0.16;
>> t_intervalo=(0:n-1)/fs;
>> y=5+cos(2*pi*12.5*t_intervalo)+sin(2*pi*18.75*t_intervalo);
>> subplot(3,1,1);
>> plot(t_intervalo,y,'-ok','LineWidth',2,'MarkerFaceColor','black');
>> title('(a) f(t) versus tempo (s)');
```

Como mencionado no início da Seção 16.5, observe que o vetor `t_intervalo` omite o último ponto.

A função `fft` pode ser usada para calcular a TFD e exibir os resultados

```
>> Y=fft(y)/n;
>> Y'
```

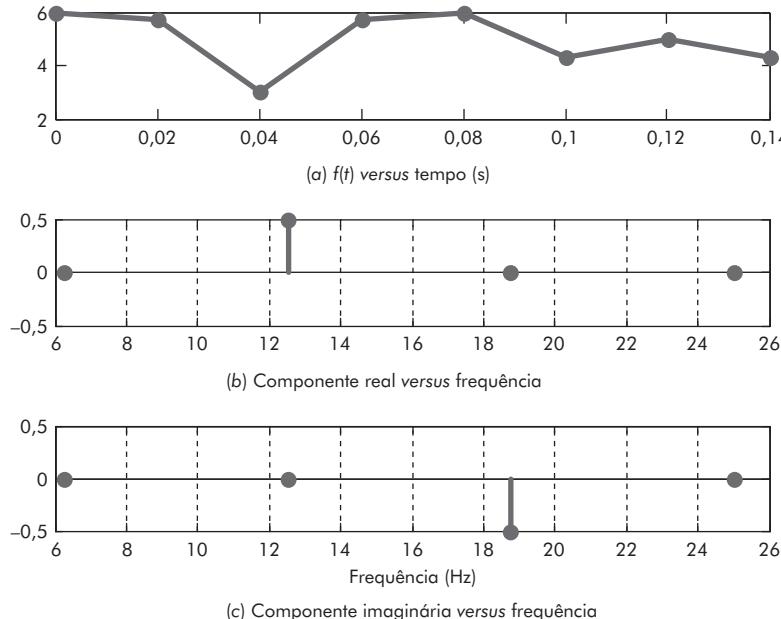


FIGURA 16.11 Resultados do cálculo da TFD com a função `fft` do MATLAB: (a) a amostra; e os gráficos das partes (b) real e (c) imaginária da TFD versus a frequência.

A transformada foi dividida por n , de modo que o primeiro coeficiente é igual à média aritmética das amostras. Quando esse código é executado, os resultados são exibidos como

```
ans =
5.0000
0.0000 - 0.0000i
0.5000
-0.0000 + 0.5000i
0
-0.0000 - 0.5000i
0.5000
0.0000 + 0.0000i
```

Observe que o primeiro coeficiente corresponde ao valor médio do sinal. Além disso, devido à *ordem envolvente reversa*, os resultados podem ser interpretados como na tabela a seguir:

Índice	k	Frequência	Período	Parte real	Parte imaginária
1	0	constante		5	0
2	1	6,25	0,16	0	0
3	2	12,5	0,08	0,5	0
4	3	18,75	0,053333	0	0,5
5	4	25	0,04	0	0
6	-3	31,25	0,032	0	-0,5
7	-2	37,5	0,026667	0,5	0
8	-1	43,75	0,022857	0	0

Observe que a `fft` detectou os sinais de 12,5 e 18,75 Hz, e a frequência de Nyquist foi destacada para indicar que os valores abaixo dela na tabela são redundantes. Isto é, eles são simplesmente reflexões dos resultados abaixo da frequência de Nyquist.

Se o valor constante for removido, pode-se traçar ambas as partes real e imaginária da TFD *versus* a frequência

```
>> nyquist=fs/2; fmin=1/T;
>> f=linspace(fmin,nyquist,n/2);
>> Y(1)=[];YP=Y(1:n/2);
>> subplot(3,1,2)
>> stem(f,real(YP),'linewidth',2,'MarkerFaceColor','blue')
>> grid,title('(b) Componente real versus frequência')
>> xlabel('frequência (Hz)')
>> subplot(3,1,3)
>> stem(f,imag(YP),'linewidth',2,'MarkerFaceColor','blue')
>> grid,title('(b) Componente imaginária versus frequência')
>> xlabel('frequência (Hz)')
```

Como esperado (lembre-se da Figura 16.7), um pico positivo ocorre para o cosseno em 12,5 Hz (Figura 16.11b) e um pico negativo ocorre para o seno em 18,75 Hz (Figura 16.11c)

16.6 O ESPECTRO DE POTÊNCIA

Além dos espectros de amplitude e fase, o espectro de potência fornece outra maneira útil de discernir os harmônicos fundamentais de sinais aparentemente aleatórios. Como o nome indica, ele deriva da análise da saída de potência de sistemas elétricos. Em termos da TFD, um *espectro de potência* consiste em um gráfico da potência associada a cada componente de frequência *versus* a frequência. A potência pode ser calculada somando-se os quadrados dos coeficientes de Fourier:

$$P_k = |\tilde{c}_k|^2$$

onde P_k é a potência associada à cada frequência $k\omega_0$.

EXEMPLO 16.4 Cálculo do espectro de potência com o MATLAB

Calcule o espectro de potência para o sinal senoidal para o qual a TFD foi calculada no Exemplo 16.3.

Solução. O seguinte programa pode ser desenvolvido para calcular o espectro de potência:

```
% Cálculo da TFD
clc;clf
n=8; dt=0.02;
fs=1/dt; t_intervalo=(0:n-1)/fs;
y=5+cos(2*pi*12.5*t_intervalo)+sin(2*pi*18.75*t_intervalo);
Y=fft(y)/n;
f = (0:n-1)*fs/n;
```

```

Y(1)=[];
f(1)=[];
% Cálculo e exibição do espectro de potência
nyquist=fs/2;
f = (1:n/2)/(n/2)*nyquist;
Pyy = abs(Y(1:n/2)).^2;
stem(f,Pyy,'LineWidth',2,'MarkerFaceColor','blue')
title('Espectro de Potência')
xlabel('Frequência (Hz)'); ylim([0 0.3])

```

Como indicado, a primeira parte do programa simplesmente calcula a TFD com os comandos pertinentes do Exemplo 16.3. A segunda parte, então, calcula e exibe o espectro de potência. Como mostra a Figura 16.12, o gráfico resultante indica que picos ocorrem em 12,5 e 18,75 Hz, conforme esperado.

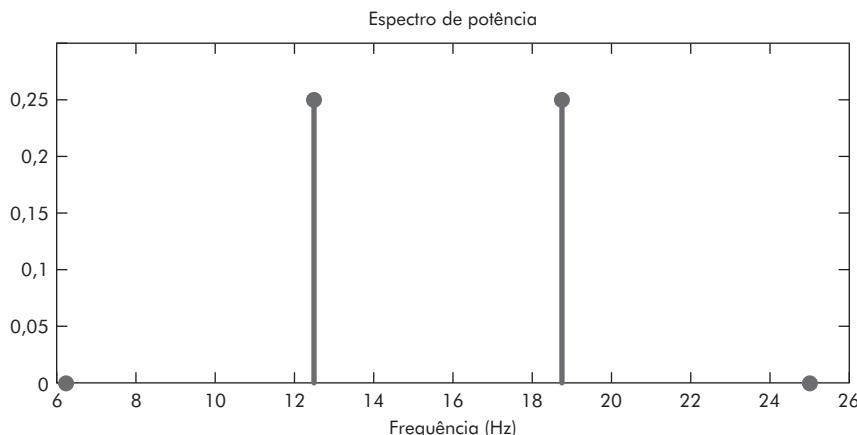


FIGURA 16.12 Espectro de potência para uma função senoidal simples com frequências de 12,5 e 18,75 Hz.

► 16.7 ESTUDO DE CASO MANCHAS SOLARES

Contextualização. Em 1848, Johann Rudolph Wolf concebeu um método para quantificar a atividade solar contando o número de manchas solares individuais ou grupos de manchas solares na superfície do sol. Ele calculou uma quantidade, agora chamada *número de manchas solares de Wolf*, adicionando 10 vezes o número de grupos mais a contagem total de manchas solares individuais. Como apresenta a Figura 16.13, o conjunto de dados para o número de manchas solares remonta a 1700. Com base no início dos registros históricos, Wolf determinou o comprimento do ciclo como sendo 11,1 anos. Use a análise de Fourier para confirmar esse resultado pela aplicação de uma FFT aos dados.



Solução. Os dados referentes ao ano e número de manchas solares estão contidos em um arquivo do MATLAB, `manchas_solares.dat`. Os comandos a seguir carregam o arquivo e atribuem as informações de ano e número de manchas solares a vetores:

```
>> load manchas_solares.dat
>> ano=manchas_solares(:,1); numero=manchas_solares(:,2);
```

Antes de aplicar a análise de Fourier, note-se que os dados parecem exibir uma tendência ascendente linear (Figura 16.13). O MATLAB pode ser usado para remover essa tendência:

```
>> n=length(numero);
>> a=polyfit(ano,numero,1);
>> tendencia_linear=polyval(a,ano);
>> ft=numero-tendencia_linear;
```

Em seguida, a função `fft` é empregada para gerar a TFD

```
>> F=fft(ft);
```

O espectro de potência pode então ser calculado e traçado

```
>> fs=1;
>> f=(0:n/2)*fs/n;
>> pow=abs(F(1:n/2+1)).^2;
>> plot(f,pow)
>> xlabel('Frequência (ciclos/ano)'); ylabel('Potência')
>> title('Potência versus frequência')
```

O resultado, como mostra a Figura 16.14, indica um pico na frequência de cerca de 0,0915 ciclos/ano. Isso corresponde a um período de $1/0,0915 = 10,93$ anos. Assim, a análise de Fourier está consistente com a estimativa de Wolf de 11 anos.

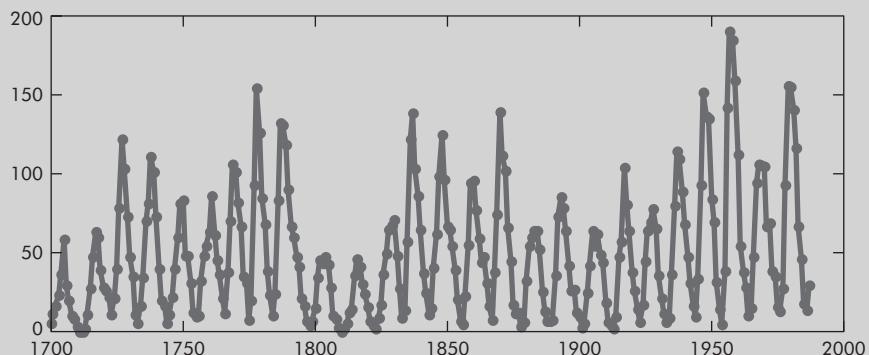


FIGURA 16.13 Gráfico do número de manchas solares de Wolf versus o ano. A linha pontilhada indica uma leve tendência de crescimento linear.

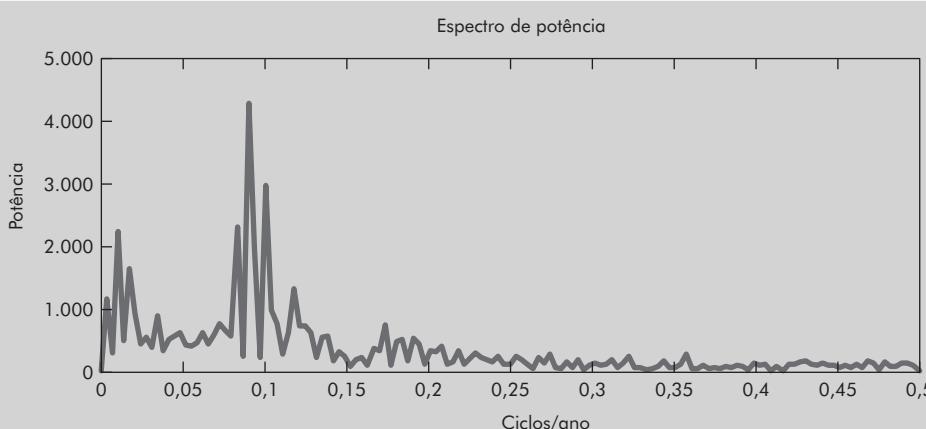


FIGURA 16.14 Espectro de potência para o número de manchas solares de Wolf versus o ano.

►►PROBLEMAS

16.1 O pH em um reator varia senoidalmente no decorrer de um dia. Empregue regressão por mínimos quadrados para ajustar a Equação (16.11) aos dados a seguir. Use seu ajuste para determinar a média, a amplitude e o instante de pH máximo. Observe que o período é 24 h.

Tempo, h	0	2	4	5	7	9
pH	7,6	7,2	7	6,5	7,5	7,2
Tempo, h	12	15	20	22	24	
pH	8,9	9,1	8,9	7,9	7	

16.2 A radiação solar em Tucson, Arizona, foi tabulada como

Tempo, mês	J	F	M	A	M	J
Radiação, W/m ²	144	188	245	311	351	359
Tempo, mês	J	A	S	O	N	D
Radiação, W/m ²	308	287	260	211	159	131

Considerando que cada mês dura 30 dias, ajuste uma curva senoidal a esses dados. Use a equação resultante para prever a radiação em meados de agosto.

16.3 Os valores médios de uma função podem ser determinados por

$$\bar{f} = \frac{\int_0^t f(t) dt}{t}$$

Use essa relação para verificar os resultados da Equação (16.13).

16.4 Em circuitos elétricos, é comum observar o comportamento de correntes na forma de uma onda quadrada como mostra a Figura P16.4 (observe que essa onda quadrada difere daquela descrita no Exemplo 16.2). Resolvendo a série de Fourier para

$$f(t) = \begin{cases} A_0 & 0 \leq t \leq T/2 \\ -A_0 & T/2 \leq t \leq T \end{cases}$$

a série de Fourier pode ser representada como

$$f(t) = \sum_{n=1}^{\infty} \left(\frac{4A_0}{(2n-1)\pi} \right) \sin\left(\frac{2\pi(2n-1)t}{T}\right)$$

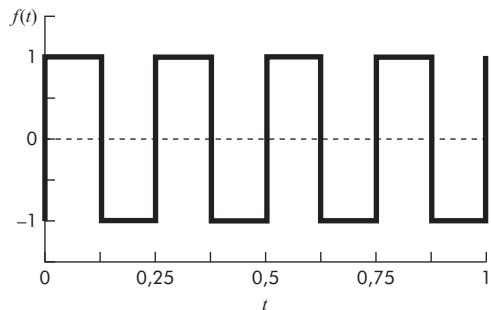


FIGURA P16.4

Desenvolva uma função no MATLAB para gerar um gráfico dos primeiros n termos da série de Fourier individualmente, bem como a soma desses n termos. Desenvolva sua função de modo que ela trace as curvas de $t = 0$ a $4T$. Use linhas finas tracejadas vermelhas para os termos individuais e linha grossa sólida preta para a soma (isto é, 'k', 'linewidth', 2). A primeira linha da função deve ser

```
function [t,f]=Fourier _ Quadrada(A0,T,n)
```

Execute a função para $A_0 = 1$, $T = 0,25$ s e $n = 6$.

16.5 Use a série de Fourier contínua para aproximar a onda dente de serra na Figura P16.5. Trace os primeiros quatro termos junto com a soma. Além disso, construa as linhas espectrais de amplitude e de fase para os quatro primeiros termos.

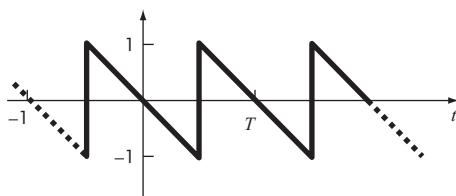


FIGURA P16.5 Onda dente de serra.

16.6 Use a série de Fourier contínua para aproximar a forma de onda triangular na Figura P16.6. Trace os primeiros quatro termos junto com a soma. Além disso, construa as linhas espectrais de amplitude e de fase para os quatro primeiros termos.

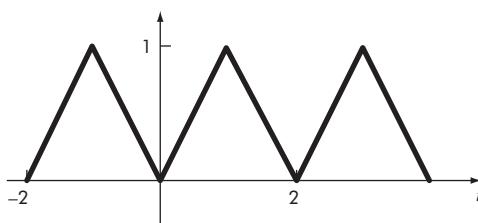


FIGURA P16.6 Onda triangular.

16.7 Use a expansão em série de Maclaurin para e^x , $\cos x$ e $\sin x$ para provar a fórmula de Euler [Equação (16.21)].

16.8 Um retificador de meia onda pode ser caracterizado por

$$C_1 = \left[\frac{1}{\pi} + \frac{1}{2} \operatorname{sen} t - \frac{2}{3\pi} \cos 2t - \frac{2}{15\pi} \cos 4t - \frac{2}{35\pi} \cos 6t - \dots \right]$$

onde C_1 é a amplitude da onda.

(a) Trace os primeiros quatro termos juntamente com a soma.

(b) Construa as linhas espectrais de amplitude e de fase para os quatro primeiros termos.

16.9 Refaça o Exemplo 16.3, porém para 64 pontos amostrados em uma taxa $\Delta t = 0,01$ s da função

$$f(t) = \cos[2\pi(12,5)t] + \cos[2\pi(25)t]$$

Use a função `fft` para gerar uma TFD desses valores e trace os resultados.

16.10 Use o MATLAB parar gerar 64 pontos da função

$$f(t) = \cos(10t) + \operatorname{sen}(3t)$$

de $t = 0$ a 2π . Adicione uma componente aleatória ao sinal com a função `randn`. Use a função `fft` para gerar uma TFD desses valores e trace os resultados.

16.11 Use o MATLAB para gerar 32 pontos para a senoide descrita na Figura 16.2 de $t = 0$ a 6 s. Calcule a TFD e crie subplots (a) do sinal original, e das partes (b) real e (c) imaginária da TFD versus a frequência.

16.12 Use a função `fft` para calcular a TFD para a onda triangular do Problema 16.6. Tome uma amostra da onda de $t = 0$ a $4T$ utilizando 128 pontos de amostragem.

16.13 Desenvolva uma função no MATLAB que utilize a função `fft` para gerar um gráfico de espectro de potência. Utilize-a para resolver o Problema 16.9.

Interpolação polinomial

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar a interpolação polinomial. Após esta leitura completa, você será capaz de:

- ✓ Reconhecer que o cálculo dos coeficientes do polinômio com equações simultâneas é um problema mal condicionado.
- ✓ Aprender como calcular os coeficientes do polinômio e interpolar com as funções nativas do MATLAB `polyfit` e `polyval`.
- ✓ Saber como efetuar uma interpolação utilizando polinômios de Newton.
- ✓ Saber como efetuar uma interpolação utilizando polinômios de Lagrange.
- ✓ Aprender como resolver um problema de interpolação inversa reformulando-o como um problema de raízes.
- ✓ Avaliar os perigos da extrapolação.
- ✓ Reconhecer que polinômios de grau mais alto podem manifestar grandes oscilações.

VOCÊ TEM UM PROBLEMA

Se quisermos melhorar a previsão de velocidade para o saltador de *bungee jumping* em queda livre, podemos expandir nosso modelo para ter em conta outros fatores além da massa e do coeficiente de arraste. Como mencionado na Seção 1.4, este último pode ser formulado como uma função de outros fatores, tais como a área do saltador e características como a densidade e viscosidade do ar.

A densidade e viscosidade do ar são frequentemente apresentadas na forma de tabelas como uma função da temperatura. Por exemplo, a Tabela 17.1 é reimpressa de um livro popular de mecânica dos fluidos (White, 1999).

Considere que você deseja obter a densidade em uma temperatura que não está incluída na tabela. Nesse caso, você teria que interpolar, ou seja, você teria que estimar o valor na temperatura desejada com base nas densidades que a delimitam. A abordagem mais simples é determinar a equação para a reta conectando os dois valores adjacentes e utilizar essa equação para estimar a densidade na temperatura intermediária desejada. Embora essa *interpolação linear* seja perfeitamente adequada em muitos casos, o erro pode ser introduzido quando os dados exibem curvatura significativa. Neste capítulo, exploraremos várias abordagens diferentes para obter estimativas adequadas para tais situações.

TABELA 17.1 Densidade (ρ), viscosidade dinâmica (μ) e viscosidade cinemática (v) como uma função da temperatura (T) em 1 atm como reportado por White (1999).

$T, ^\circ\text{C}$	$\rho, \text{kg/m}^3$	$\mu, \text{N} \cdot \text{s/m}^2$	$v, \text{m}^2/\text{s}$
-40	1,52	$1,51 \times 10^{-5}$	$0,99 \times 10^{-5}$
0	1,29	$1,71 \times 10^{-5}$	$1,33 \times 10^{-5}$
20	1,20	$1,80 \times 10^{-5}$	$1,50 \times 10^{-5}$
50	1,09	$1,95 \times 10^{-5}$	$1,79 \times 10^{-5}$
100	0,946	$2,17 \times 10^{-5}$	$2,30 \times 10^{-5}$
150	0,835	$2,38 \times 10^{-5}$	$2,85 \times 10^{-5}$
200	0,746	$2,57 \times 10^{-5}$	$3,45 \times 10^{-5}$
250	0,675	$2,75 \times 10^{-5}$	$4,08 \times 10^{-5}$
300	0,616	$2,93 \times 10^{-5}$	$4,75 \times 10^{-5}$
400	0,525	$3,25 \times 10^{-5}$	$6,20 \times 10^{-5}$
500	0,457	$3,55 \times 10^{-5}$	$7,77 \times 10^{-5}$

17.1 INTRODUÇÃO À INTERPOLAÇÃO

Você frequentemente terá a oportunidade de fazer estimativas de valores intermediários entre dados precisos. O método mais comum usado para esse propósito é a interpolação polinomial. A fórmula geral para um polinômio de grau ($n - 1$) pode ser escrita como

$$f(x) = a_1 + a_2x + a_3x^2 + \cdots + a_nx^{n-1} \quad (17.1)$$

Para n pontos dados, existe um, e somente um, polinômio ($n - 1$) que passa por todos os pontos. Por exemplo, existe uma única reta (isto é, um polinômio de primeiro grau) que liga dois pontos (Figura 17.1a). Analogamente, existe uma única parábola ligando um conjunto de três pontos (Figura 17.1b). A *interpolação polinomial* consiste em determinar o único polinômio de grau ($n - 1$) que passar pelos n pontos dados. Esse polinômio, então, fornece uma fórmula para calcular valores intermediários.

Antes de prosseguir, deve-se notar que o MATLAB representa os coeficientes do polinômio de uma maneira diferente da Equação (17.1). Em vez de utilizar potências crescentes de x , ele usa potências decrescentes, como em

$$f(x) = p_1x^{n-1} + p_2x^{n-2} + \cdots + p_{n-1}x + p_n \quad (17.2)$$

Para ser consistente com o MATLAB, adotaremos esse esquema na seção a seguir.

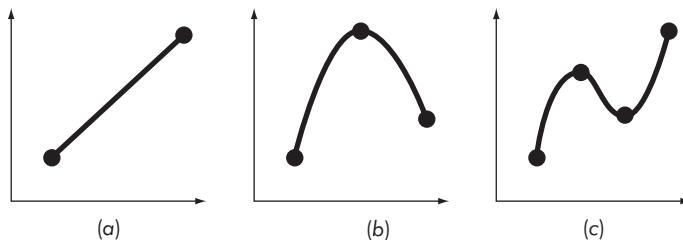


FIGURA 17.1 Exemplos de polinômios interpoladores: (a) de primeiro grau (linear) ligando dois pontos, (b) de segundo grau (quadrático ou parabólico) ligando três pontos, e (c) de terceiro grau (cúbico) ligando quatro pontos.

17.1.1 Determinação dos coeficientes do polinômio

Uma maneira simples de calcular os coeficientes da Equação (17.2) é baseada no fato de que n pontos dados são necessários para determinar os n coeficientes. Como no exemplo a seguir, isso permite gerar n equações algébricas lineares que podem ser resolvidas simultaneamente para calcular os coeficientes.

EXEMPLO 17.1 Determinação dos coeficientes do polinômio com equações simultâneas

Considere que se queira determinar os coeficientes da parábola, $f(x) = p_1x^2 + p_2x + p_3$, que passa através dos três últimos valores de densidade da Tabela 17.1:

$$x_1 = 300 \quad f(x_1) = 0,616$$

$$x_2 = 400 \quad f(x_2) = 0,525$$

$$x_3 = 500 \quad f(x_3) = 0,457$$

Solução. Cada um desses pares pode ser substituído na Equação (17.2) para produzir um sistema de três equações:

$$0,616 = p_1(300)^2 + p_2(300) + p_3$$

$$0,525 = p_1(400)^2 + p_2(400) + p_3$$

$$0,457 = p_1(500)^2 + p_2(500) + p_3$$

ou, na forma matricial:

$$\begin{bmatrix} 90.000 & 300 & 1 \\ 160.000 & 400 & 1 \\ 250.000 & 500 & 1 \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix} = \begin{Bmatrix} 0,616 \\ 0,525 \\ 0,457 \end{Bmatrix}$$

Assim, o problema é reduzido à resolução de três equações algébricas lineares simultâneas para os três coeficientes desconhecidos. Uma sessão simples do MATLAB pode ser usada para obter a solução:

```

>> format long
>> A = [ 90000 300 1; 160000 400 1; 250000 500 1];
>> b = [ 0.616 0.525 0.457 ]';
>> p = A\b
p =
    0.0000011500000
   -0.00171500000000
    1.02700000000000

```

Portanto, a parábola que passa exatamente através dos três pontos é

$$f(x) = 0,00000115x^2 - 0,001715x + 1,027$$

Esse polinômio fornece um meio para determinar pontos intermediários. Por exemplo, o valor da densidade em uma temperatura de 350 °C pode ser calculada como

$$f(350) = 0,00000115(350)^2 - 0,001715(350) + 1,027 = 0,567625$$

Embora a abordagem no Exemplo 17.1 forneça uma maneira fácil de efetuar interpolação, ela tem uma deficiência séria. Para compreender essa falha, observe que a matriz de coeficientes no Exemplo 17.1 tem uma estrutura determinada. Isso pode ser visto claramente expressando-a em termos gerais, como em:

$$\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix} \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix} = \begin{Bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \end{Bmatrix} \quad (17.3)$$

Matrizes de coeficientes dessa forma são chamadas de *matrizes de Vandermonde*. Essas matrizes são muito mal condicionadas; ou seja, suas soluções são muito sensíveis a erros de arredondamento. Isso pode ser ilustrado utilizando o MATLAB para calcular o número de condicionamento para a matriz de coeficientes do Exemplo 17.1 como

```

>> cond(A)
ans =
      5.8932e+006

```

Esse número de condicionamento, que é bastante grande para uma matriz 3×3 , implica que cerca de seis dígitos da solução seriam questionáveis. O mau condicionamento torna-se ainda pior à medida que o número de equações simultâneas aumenta.

Como consequência, há abordagens alternativas que não apresentam essa deficiência. Neste capítulo, descreveremos também duas alternativas que são adequadas para implementação computacional: os polinômios de Newton e de Lagrange. Antes disso, porém, reexaminaremos brevemente como os coeficientes do polinômio interpolador podem ser estimados diretamente com funções nativas do MATLAB.

17.1.2 Função nativas do MATLAB: polyfit e polyval

Como visto na Seção 14.5.2, a função `polyfit` pode ser usada para realizar uma regressão polinomial. Nessas aplicações, o número de pontos dados é maior que o número de coeficientes sendo estimados. Consequentemente, a reta de ajuste por mínimos quadrados não passa necessariamente através de qualquer dos pontos, mas, em vez disso, segue a tendência geral dos dados.

No caso em que o número de pontos dados é igual ao número de coeficientes, a função `polyfit` efetua a interpolação; ou seja, retorna os coeficientes do polinômio que passa diretamente através dos pontos dados. Por exemplo, ela pode ser usada para determinar os coeficientes da parábola que passa através dos três últimos valores da Tabela 17.1:

```
>> format long
>> T = [300 400 500];
>> densidade = [0.616 0.525 0.457];
>> p = polyfit(T,density,2)
p =
    0.0000011500000 -0.00171500000000 1.027000000000000
```

Pode-se então utilizar a função `polyval` para realizar a interpolação, como em

```
>> d = polyval(p,350)
d =
    0.567625000000000
```

Esses resultados concordam com aqueles obtidos anteriormente no Exemplo 17.1 com equações simultâneas.

17.2 POLINÔMIOS INTERPOLADORES DE NEWTON

Há uma grande variedade de formas alternativas para expressar um polinômio interpolador, além do formato familiar da Equação (17.2). Este polinômio por diferenças divididas de Newton está entre as fórmulas mais populares e úteis. Antes de apresentar a equação geral, serão introduzidas versões de primeiro e de segundo graus devido à sua interpretação visual simples.

17.2.1 Interpolação linear

A forma mais simples de interpolar é ligar dois pontos dados com uma reta. Essa técnica, chamada de *interpolação linear*, é descrita graficamente na Figura 17.2. Usando semelhança de triângulos,

$$\frac{f_1(x) - f(x_1)}{x - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (17.4)$$

que pode ser reorganizada para fornecer

$$f_1(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) \quad (17.5)$$

que é a fórmula de *interpolação linear de Newton*. A notação $f_1(x)$ indica que esse é um polinômio interpolador de primeiro grau. Observe que, além de representar a

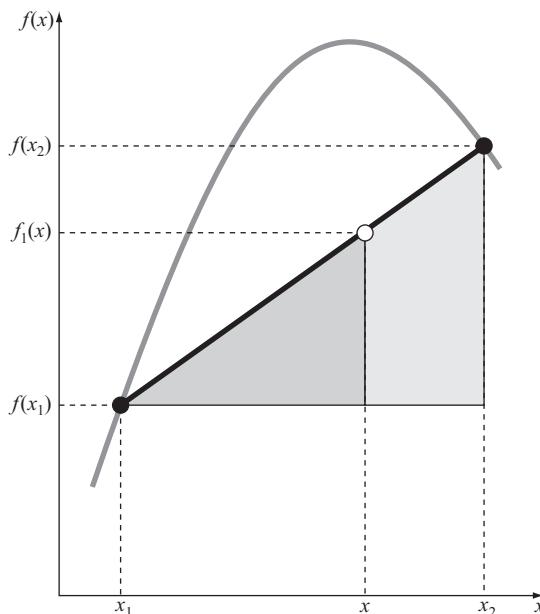


FIGURA 17.2 Descrição gráfica de uma interpolação linear. As áreas hachuradas indicam os triângulos semelhantes usados para deduzir a fórmula de interpolação linear [Equação (17.5)].

inclinação da reta ligando os pontos, o termo $[f(x_2) - f(x_1)]/(x_2 - x_1)$ é uma aproximação por diferenças divididas finitas da primeira derivada [lembre-se da Equação (4.20)]. Em geral, quanto menor o intervalo entre os pontos dados, melhor a aproximação, o que se deve ao fato de que, à medida que o intervalo diminui, uma função contínua será mais bem aproximada por uma reta. Essa característica é ilustrada no exemplo a seguir.

EXEMPLO 17.2 Interpolação linear

Faça uma estimativa do logaritmo natural de 2 usando uma interpolação linear. Primeiro, faça o cálculo interpolando entre $\ln 1 = 0$ e $\ln 6 = 1,791759$. Então, repita o procedimento, mas use o intervalo menor de $\ln 1$ a $\ln 4$ (1,386294). Observe que o valor verdadeiro de $\ln 2$ é 0,6931472.

Solução. Usamos a Equação (17.5) para $x_1 = 1$ a $x_2 = 6$ para fornecer

$$f_1(2) = 0 + \frac{1,791759 - 0}{6 - 1}(2 - 1) = 0,3583519$$

que representa um erro de $\varepsilon_t = 48,3\%$. O uso do intervalo menor de $x_1 = 1$ a $x_2 = 4$ fornece

$$f_1(2) = 0 + \frac{1,386294 - 0}{4 - 1}(2 - 1) = 0,4620981$$

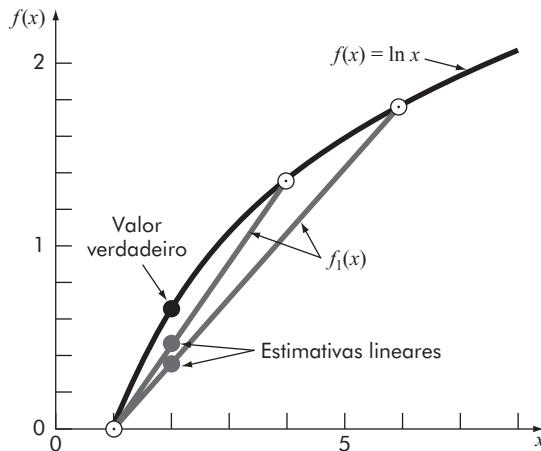


FIGURA 17.3 Duas interpolações lineares para estimar $\ln 2$. Observe como o intervalo menor fornece uma estimativa melhor.

Logo, o uso do intervalo menor reduz o erro relativo percentual para $\varepsilon_t = 33,3\%$. Ambas as interpolações são apresentadas na Figura 17.3, com a função verdadeira.

17.2.2 Interpolação quadrática

O erro no Exemplo 17.2 resultou da aproximação de uma curva por uma reta. Consequentemente, uma estratégia que torna a estimativa melhor é introduzir alguma curvatura na curva ligando os pontos. Se estiverem disponíveis três pontos, isso pode ser conseguido com um polinômio de segundo grau (também chamado de polinômio quadrático ou uma parábola). Uma fórmula particularmente conveniente para esse propósito é

$$f_2(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2) \quad (17.6)$$

Um procedimento simples pode ser usado para determinar os valores dos coeficientes. Para b_1 , a Equação (17.6) com $x = x_1$ pode ser usada para calcular

$$b_1 = f(x_1) \quad (17.7)$$

A Equação (17.7) pode ser substituída na Equação (17.6), a qual pode ser calculada em $x = x_2$ para

$$b_2 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (17.8)$$

Por fim, as Equações (17.7) e (17.8) podem ser substituídas na Equação (17.6), a qual pode ser calculada em $x = x_3$ e resolvida (depois de algumas manipulações algébricas) por

$$b_3 = \frac{\frac{f(x_3) - f(x_2)}{x_3 - x_2} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}}{x_3 - x_1} \quad (17.9)$$

Observe que, de modo análogo à interpolação linear, b_2 ainda representa a inclinação da reta ligando os pontos x_1 e x_2 . Logo, os dois primeiros termos da Equação (17.6) são equivalentes à interpolação linear entre x_1 e x_2 , como especificado na Equação (17.5). O último termo, $b_3(x - x_1)(x - x_2)$, introduz a curvatura de segundo grau na fórmula.

Antes de ilustrar como usar a Equação (17.6), é preciso examinar a forma do coeficiente b_3 . Ela é muito parecida com a aproximação por diferença dividida finita da segunda derivada introduzida anteriormente na Equação (4.27). Assim, a Equação (17.6) começa a manifestar uma estrutura que é muito parecida com a expansão em série de Taylor; ou seja, termos são adicionados sequencialmente para capturar o comportamento de uma curvatura de ordem cada vez mais elevada.

EXEMPLO 17.3 Interpolação quadrática

Empregue um polinômio de segundo grau de Newton para estimar $\ln 2$ com os mesmos três pontos usados no Exemplo 17.2:

$$\begin{aligned} x_1 &= 1 & f(x_1) &= 0 \\ x_2 &= 4 & f(x_2) &= 1,386294 \\ x_3 &= 6 & f(x_3) &= 1,791759 \end{aligned}$$

Solução. Aplicando a Equação (17.7), obtém-se

$$b_1 = 0$$

A Equação (17.8) fornece

$$b_2 = \frac{1,386294 - 0}{4 - 1} = 0,4620981$$

e a Equação (17.9) produz

$$b_3 = \frac{\frac{1,791759 - 1,386294}{6 - 4} - 0,4620981}{6 - 1} = -0,0518731$$

Substituindo esses valores na Equação (17.6), obtém-se a fórmula quadrática

$$f_2(x) = 0 + 0,4620981(x - 1) - 0,0518731(x - 1)(x - 4)$$

que pode ser calculada em $x = 2$ para fornecer $f_2(2) = 0,5658444$, que representa um erro relativo de $\varepsilon_r = 18,4\%$. Logo, a curvatura introduzida pela fórmula quadrática (Figura 17.4) melhora a interpolação, quando comparada com o resultado obtido usando-se retas no Exemplo 17.2 e Figura 17.3.

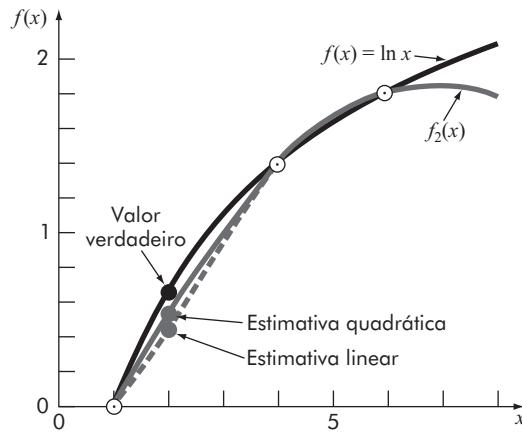


FIGURA 17.4 O uso da interpolação quadrática para fazer uma estimativa de $\ln 2$. A interpolação linear de $x = 1$ a $x = 4$ também está incluída para comparação.

17.2.3 Forma geral dos polinômios interpoladores de Newton

A análise anterior pode ser generalizada para ajustar um polinômio de grau $(n - 1)$ a n pontos dados. O polinômio de grau $(n - 1)$ é

$$f_{n-1}(x) = b_1 + b_2(x - x_1) + \dots + b_n(x - x_1)(x - x_2) \dots (x - x_{n-1}) \quad (17.10)$$

Como feito anteriormente com as interpolações linear e quadrática, os pontos dados podem ser usados para calcular os coeficientes b_1, b_2, \dots, b_n . Para um polinômio de grau $(n - 1)$, n pontos dados são necessários: $[x_1, f(x_1)], [x_2, f(x_2)], \dots, [x_n, f(x_n)]$. Usamos esses pontos e as seguintes equações para calcular os coeficientes:

$$b_1 = f(x_1) \quad (17.11)$$

$$b_2 = f[x_2, x_1] \quad (17.12)$$

$$b_3 = f[x_3, x_2, x_1] \quad (17.13)$$

⋮

$$b_n = f[x_n, x_{n-1}, \dots, x_2, x_1] \quad (17.14)$$

onde a função com colchetes corresponde a diferenças divididas finitas. Por exemplo, a primeira diferença dividida finita é representada em geral por

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j} \quad (17.15)$$

A segunda diferença dividida finita, que representa a diferença das duas primeiras diferenças divididas, é expressa em geral por

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k} \quad (17.16)$$

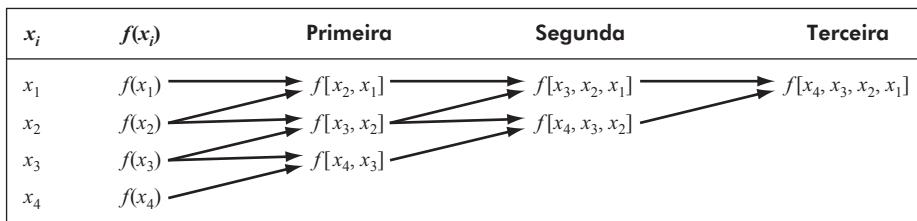


FIGURA 17.5 Descrição gráfica da natureza recursiva das diferenças divididas finitas. Essa representação é conhecida como uma tabela de diferenças divididas.

Analogamente, a n -ésima diferença dividida finita é

$$f[x_n, x_{n-1}, \dots, x_2, x_1] = \frac{f[x_n, x_{n-1}, \dots, x_2] - f[x_{n-1}, x_{n-2}, \dots, x_1]}{x_n - x_1} \quad (17.17)$$

Essas diferenças podem ser usadas para calcular os coeficientes nas Equações (17.11) a (17.14), os quais podem então ser substituídos na Equação (17.10) para fornecer o polinômio interpolador de Newton:

$$\begin{aligned} f_{n-1}(x) = & f(x_1) + (x - x_1)f[x_2, x_1] + (x - x_1)(x - x_2)f[x_3, x_2, x_1] \\ & + \cdots + (x - x_1)(x - x_2) \cdots (x - x_{n-1})f[x_n, x_{n-1}, \dots, x_2, x_1] \end{aligned} \quad (17.18)$$

Deve ser observado que não é necessário que os dados usados na Equação (17.18) sejam igualmente espaçados, ou que os valores das abscissas estejam necessariamente em ordem crescente, como ilustra o exemplo a seguir. No entanto, os pontos devem ser ordenados de modo que estejam centrados em torno e tão próximo quanto possível do valor desconhecido. Observe também como as Equações (17.15) a (17.17) são recursivas – isto é, diferenças de ordem mais alta são calculadas tomando-se diferenças das diferenças de ordem mais baixa (Figura 17.5). Essa propriedade será explorada quando for desenvolvida uma função eficiente no MATLAB para implementar o método.

EXEMPLO 17.4 Polinômios interpoladores de Newton

No Exemplo 17.3, dados em $x_1 = 1$, $x_2 = 4$ e $x_3 = 6$ foram usados para fazer uma estimativa de $\ln 2$ com uma parábola. Agora, adicionando um quarto ponto $[x_4 = 5; f(x_4) = 1,609438]$, faça uma estimativa de $\ln 2$ com um polinômio interpolador de Newton de terceiro grau.

Solução. O polinômio de terceiro grau, a Equação (17.10) com $n = 4$, é

$$f_3(x) = b_1 + b_2(x - x_1) + b_3(x - x_1)(x - x_2) + b_4(x - x_1)(x - x_2)(x - x_3)$$

As primeiras diferenças divididas para o problema são [Equação (17.15)]

$$f[x_2, x_1] = \frac{1,386294 - 0}{4 - 1} = 0,4620981$$

$$f[x_3, x_2] = \frac{1,791759 - 1,386294}{6 - 4} = 0,2027326$$

$$f[x_4, x_3] = \frac{1,609438 - 1,791759}{5 - 6} = 0,1823216$$

As segundas diferenças divididas são [Equação (17.16)]

$$f[x_3, x_2, x_1] = \frac{0,2027326 - 0,4620981}{6 - 1} = -0,05187311$$

$$f[x_4, x_3, x_2] = \frac{0,1823216 - 0,2027326}{5 - 4} = -0,02041100$$

A terceira diferença dividida é [Equação (17.17) com $n = 4$]

$$f[x_4, x_3, x_2, x_1] = \frac{-0,02041100 - (-0,05187311)}{5 - 1} = 0,007865529$$

Assim, a tabela de diferenças divididas é

x_i	$f(x_i)$	Primeira	Segunda	Terceira
1	0	0,4620981	-0,05187311	0,007865529
4	1,386294	0,2027326	-0,02041100	
6	1,791759	0,1823216		
5	1,609438			

Os resultados para $f(x_1), f[x_2, x_1], f[x_3, x_2, x_1]$ e $f[x_4, x_3, x_2, x_1]$ representam os coeficientes b_1, b_2, b_3 e b_4 , respectivamente, da Equação (17.10). Assim, a interpolação cúbica é

$$\begin{aligned} f_3(x) &= 0 + 0,4620981(x - 1) - 0,05187311(x - 1)(x - 4) \\ &\quad + 0,007865529(x - 1)(x - 4)(x - 6) \end{aligned}$$

a qual pode ser usada para calcular $f_3(2) = 0,6287686$, que representa um erro relativo de $\varepsilon_t = 9,3\%$. O polinômio cúbico completo é apresentado na Figura 17.6.

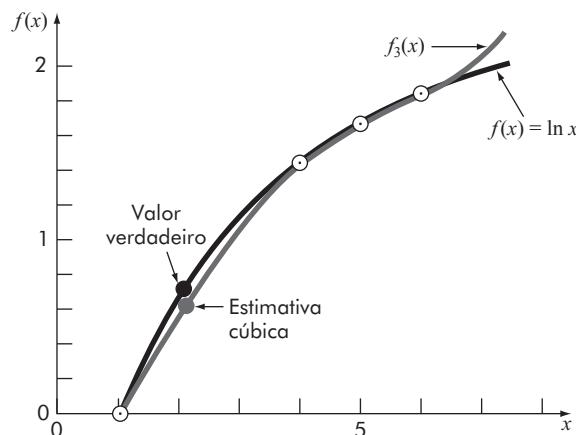


FIGURA 17.6 O uso da interpolação cúbica para estimar $\ln 2$.

17.2.4 Função do MATLAB: Newtint

É simples desenvolver uma função no MATLAB para implementar a interpolação de Newton. Como na Figura 17.7, o primeiro passo é calcular as diferenças divididas finitas e armazená-las em um arranjo. As diferenças são, então, usadas em conjunto com a Equação (17.18) para efetuar a interpolação.

Como um exemplo de utilização da função, podemos refazer os cálculos efetuados no Exemplo 17.3:

```
>> format long
>> x=[1 4 6 5]';
>> y=log(x);
>> Newtint(x,y,2)

ans =
0.62876857890841

function yint = Newtint(x,y,xx)
% Newtint: Polinômio interpolador de Newton
% yint = Newtint(x,y,xx): Utiliza um polinômio interpolador
% de Newton de grau (n - 1) com base em n pontos dados (x,y)
% para determinar um valor da variável dependente (yint)
% em um dado valor da variável independente, xx.
% entrada:
% x = variável independente
% y = variável dependente
% xx = valor da variável independente em que
%       a interpolação é calculada
% saída:
% yint = valor interpolado da variável dependente

% calcula as diferenças divididas finitas na forma de uma
% tabela de diferenças
n = length(x);
if length(y)~=n, error('x e y devem ter o mesmo tamanho'); end
b = zeros(n,n);
% atribui as variáveis dependentes à primeira coluna de b.
b(:,1) = y(:); % o (:) garante que y é um vetor coluna.
for j = 2:n
    for i = 1:n-j+1
        b(i,j) = (b(i+1,j-1)-b(i,j-1))/(x(i+j-1)-x(i));
    end
end
% usa as diferenças divididas finitas para interpolar
xt = 1;
yint = b(1,1);
for j = 1:n-1
    xt = xt*(xx-x(j));
    yint = yint+b(1,j+1)*xt;
end
```

FIGURA 17.7 Função do MATLAB para implementar a interpolação de Newton.

17.3 POLINÔMIOS INTERPOLADORES DE LAGRANGE

Considere que seja formulado um polinômio interpolador linear como a média ponderada dos dois valores que estão sendo ligados por uma reta:

$$f(x) = L_1 f(x_1) + L_2 f(x_2) \quad (17.19)$$

onde os L 's são coeficientes de ponderação. É lógico que o primeiro coeficiente de ponderação é uma reta que é igual a 1 em x_1 e 0 em x_2 :

$$L_1 = \frac{x - x_2}{x_1 - x_2}$$

De modo similar, o segundo coeficiente é uma reta que é igual a 1 em x_2 e 0 em x_1 :

$$L_2 = \frac{x - x_1}{x_2 - x_1}$$

Substituindo esses coeficientes na Equação (17.19), obtém-se a reta que liga os pontos (Figura 17.8):

$$f_1(x) = \frac{x - x_2}{x_1 - x_2} f(x_1) + \frac{x - x_1}{x_2 - x_1} f(x_2) \quad (17.20)$$

onde a nomenclatura $f_1(x)$ indica que este é um polinômio de primeiro grau. A Equação (17.20) é conhecida como *polinômio interpolador de Lagrange linear*.

A mesma estratégia pode ser empregada para ajustar uma parábola através de três pontos. Nesse caso, três parábolas seriam usadas, com cada uma passando através de um dos pontos e igualando a zero nos outros dois. A soma delas representaria, então, a única parábola que liga os três pontos. Esse polinômio interpolador de Lagrange de segundo grau pode ser escrito como

$$\begin{aligned} f_2(x) &= \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} f(x_1) + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} f(x_2) \\ &\quad + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} f(x_3) \end{aligned} \quad (17.21)$$

Observe como o primeiro termo é igual a $f(x_1)$ em x_1 e igual a zero em x_2 e x_3 . Os outros termos funcionam de maneira similar.

Ambas as versões de primeira e segunda ordem bem como os polinômios de Lagrange de ordem superior podem ser representados concisamente por

$$f_{n-1}(x) = \sum_{i=1}^n L_i(x) f(x_i) \quad (17.22)$$

sendo

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad (17.23)$$

onde n é o numero de pontos dados e \prod indica o “produto de”.

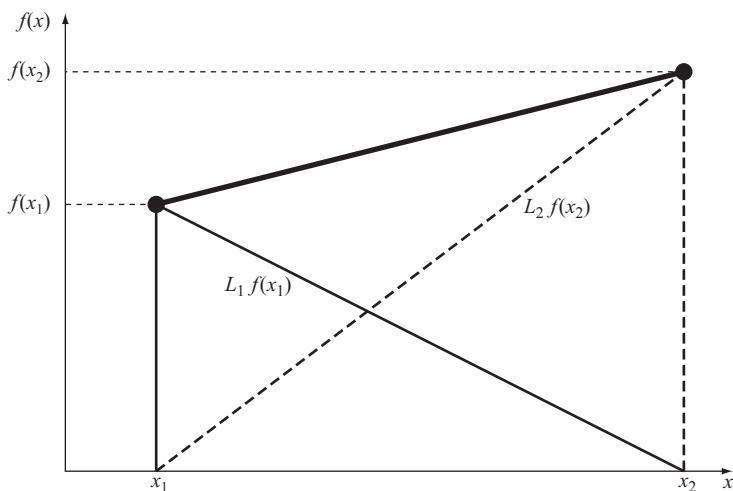


FIGURA 17.8 Descrição visual da lógica por trás dos polinômios interpoladores de Lagrange. A figura mostra o caso de primeira ordem. Cada um dos dois termos da Equação (17.20) passa através de um dos pontos e é zero no outro. A soma dos dois termos deve, portanto, ser a única reta que liga os dois termos.

EXEMPLO 17.5 Polinômios interpoladores de Lagrange

Use um polinômio interpolador de Lagrange de primeiro e de segundo graus para calcular a densidade de um óleo de motor não utilizado em $T = 15^\circ\text{C}$ com base nos seguintes dados:

$$x_1 = 0 \quad f(x_1) = 3,85$$

$$x_2 = 20 \quad f(x_2) = 0,800$$

$$x_3 = 40 \quad f(x_3) = 0,212$$

Solução. O polinômio de primeiro grau [Equação (17.20)] pode ser usado para se obter a estimativa em $x = 15$:

$$f_1(x) = \frac{15 - 20}{0 - 20} 3,85 + \frac{15 - 0}{20 - 0} 0,800 = 1,5625$$

De modo semelhante, o polinômio de segundo grau é desenvolvido como [Equação (17.21)]

$$\begin{aligned} f_2(x) &= \frac{(15 - 20)(15 - 40)}{(0 - 20)(0 - 40)} 3,85 + \frac{(15 - 0)(15 - 40)}{(20 - 0)(20 - 40)} 0,800 \\ &\quad + \frac{(15 - 0)(15 - 20)}{(40 - 0)(40 - 20)} 0,212 = 1,3316875 \end{aligned}$$

17.3.1 Função do MATLAB: Lagrange

É simples desenvolver uma função no MATLAB com base nas Equações (17.22) e (17.23). Como detalha a Figura 17.9, são passados como argumentos de entrada para a função dois vetores contendo as variáveis independente (x) e dependente (y). Passa-se também o valor da variável independente em que se quer interpolar (xx). O grau do polinômio é baseado no tamanho do vetor x que é passado para a função. Se n valores são passados, um polinômio de grau ($n - 1$) é ajustado.

Como um exemplo de utilização da função, podemos prever a densidade do ar em uma pressão de 1 atm e em uma temperatura de 15 °C, com base nos primeiros quatro valores da Tabela 17.1. Considerando que quatro valores são passados para a função, um polinômio de terceiro grau é implementado pela função `Lagrange` para fornecer:

```
>> format long
>> T = [-40 0 20 50];
>> d = [1.52 1.29 1.2 1.09];
>> densidade = Lagrange(T,d,15)

densidade =
1.22112847222222

function yint = Lagrange(x,y,xx)
% Lagrange: Polinômio Interpolador de Lagrange
% yint = Lagrange(x,y,xx): Utiliza um polinômio interpolador
% de Lagrange de grau (n - 1) com base em n pontos dados
% para determinar um valor da variável dependente (yint)
% em um dado valor da variável independente, xx.
% entrada:
% x = variável independente
% y = variável dependente
% xx = valor da variável independente em que
%       a interpolação é calculada
% saída:
% yint = valor interpolado da variável dependente
n = length(x);
if length(y)~=n, error('x e y devem ter o mesmo tamanho'); end
s = 0;
for i = 1:n
    produto = y(i);
    for j = 1:n
        if i ~= j
            produto = produto*(xx-x(j))/(x(i)-x(j));
        end
    end
    s = s+produto;
end
yint = s;
```

FIGURA 17.9 Função do MATLAB para implementar a interpolação de Lagrange.

17.4 INTERPOLAÇÃO INVERSA

Como a nomenclatura indica, os valores de $f(x)$ e de x são, na maioria dos contextos de interpolação, as variáveis dependentes e independentes, respectivamente. Como consequência, os valores dos x 's são, de modo típico, uniformemente espaçados. Um exemplo simples é uma tabela de valores deduzidos para a função $f(x) = 1/x$:

x	1	2	3	4	5	6	7
$f(x)$	1	0,5	0,3333	0,25	0,2	0,1667	0,1429

Agora, considere que seja necessário usar os mesmos dados, mas foi fornecido um valor para $f(x)$ para que se determine o valor correspondente de x . Por exemplo, para os dados da tabela anterior, admita que foi solicitada a determinação do valor de x que corresponde a $f(x) = 0,3$. Nesse caso, como a função está disponível e é fácil de manipular, a resposta correta pode ser determinada diretamente como $x = 1/0,3 = 3,3333$.

Esse problema é chamado de *interpolação inversa*. Para um caso mais complicado, podem ser atrativos a troca dos valores de $f(x)$ e x [isto é, simplesmente considerar x como função de $f(x)$] e o uso de uma abordagem como a interpolação de Newton ou de Lagrange para determinar o resultado. Infelizmente, quando as variáveis são invertidas, não há nenhuma garantia de que os valores ao longo da nova abscissa [os $f(x)$'s] serão regularmente espaçados. De fato, em muitos casos, os valores sofrerão um efeito “telescópio”, ou seja, terão a aparência de uma escala logarítmica, com alguns pontos adjacentes agrupados e outros muito espalhados. Por exemplo, para $f(x) = 1/x$ o resultado é

$f(x)$	0,1429	0,1667	0,2	0,25	0,3333	0,5	1
x	7	6	5	4	3	2	1

Esse espaçamento não uniforme na abscissa em geral leva a oscilações no polinômio interpolador resultante. Isso pode ocorrer mesmo para polinômios de grau mais baixo. Uma estratégia alternativa é ajustar um polinômio interpolador de grau n , $f_n(x)$, aos dados originais [isto é, com $f(x)$ em função de x]. Na maioria dos casos, como os x 's são igualmente espaçados, esse polinômio não será mal condicionado. A resposta ao problema então se reduz a determinar o valor de x que torna o polinômio igual ao $f(x)$ dado. Logo, o problema de interpolação se reduz a um problema de raízes!

Por exemplo, para o problema descrito, uma abordagem simples consiste em ajustar um polinômio quadrático aos três pontos: $(2; 0,5)$, $(3; 0,3333)$, $(4; 0,25)$. O resultado seria

$$f_2(x) = 0,041667x^2 - 0,375x + 1,08333$$

A resposta do problema de interpolação inversa de encontrar o x correspondente a $f(x) = 0,3$ envolveria, portanto, a determinação da raiz de

$$0,3 = 0,041667x^2 - 0,375x + 1,08333$$

Nesse caso simples, a fórmula quadrática pode ser usada para calcular

$$x = \frac{0,375 \pm \sqrt{(-0,375)^2 - 4(0,041667)0,78333}}{2(0,041667)} = \frac{5,704158}{3,295842}$$

Logo, a segunda raiz, 3,296, é uma boa aproximação do valor verdadeiro de 3,333. Se quiséssemos acurácia adicional, poderia ser usado um polinômio de terceiro ou de quarto grau, junto com um dos métodos de localização de raízes dos Capítulos 5 ou 6.

17.5 EXTRAPOLAÇÃO E OSCILAÇÕES

Antes de finalizar este capítulo, há duas questões relacionadas à interpolação polinomial que devem ser abordadas: a extrapolação e as oscilações.

17.5.1 Extrapolação

A *extrapolação* é o processo de fazer uma estimativa de um valor de $f(x)$ que está fora do intervalo dos pontos-base conhecidos, x_1, x_2, \dots, x_n . Como descreve a Figura 17.10, a natureza de extremidade aberta da extrapolação representa um passo no desconhecido, pois o processo estende a curva além da região conhecida. Dessa forma, a curva verdadeira pode facilmente divergir da previsão. Portanto, deve-se tomar extremo cuidado quando ocorrer um caso no qual seja preciso extrapolar.

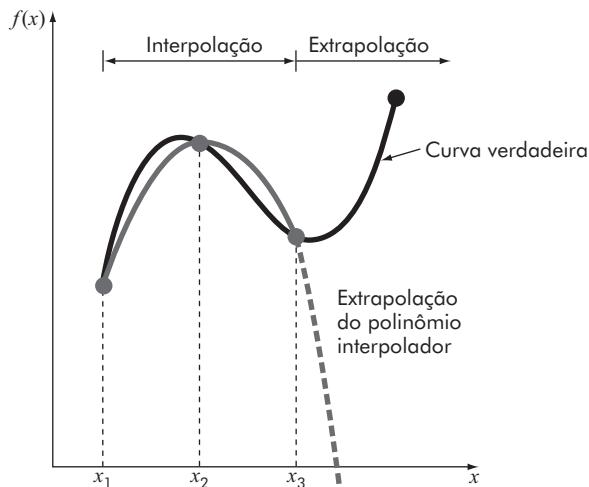


FIGURA 17.10 Ilustração da possível divergência de uma previsão extrapolada. A extrapolação é baseada no ajuste de uma parábola pelos primeiros três pontos conhecidos.

EXEMPLO 17.6 Perigos da extrapolação

Esse exemplo é baseado em outro originalmente desenvolvido por Forsythe, Malcolm e Moler¹. A população em milhões dos Estados Unidos de 1920 a 2000 pode ser tabulada como

Data	1920	1930	1940	1950	1960	1970	1980	1990	2000
População	106,46	123,08	132,12	152,27	180,67	205,05	227,23	249,46	281,42

Ajuste um polinômio sétimo grau aos 8 primeiros pontos (1920 a 1990). Use-o para calcular a população em 2000 por extrapolação e compare sua previsão com o resultado real.

Solução. Pode-se entrar com os dados como em

```
>> t = [1920:10:1990];
>> pop = [106.46 123.08 132.12 152.27 180.67 205.05 227.23
249.46];
```

A função `polyfit` pode ser usada para calcular os coeficientes

```
>> p = polyft(t,pop,7)
```

No entanto, quando esse comando é implementado, a seguinte mensagem é exibida:

```
Warning: Polynomial is badly conditioned. Remove repeated data
points or try centering and scaling as described in HELP
POLYFIT.
```

Podemos seguir a sugestão do MATLAB escalonando e centralizando os dados como em

```
>> ts = (t - 1955)/35;
```

Agora, a função `polyfit` funciona sem uma mensagem de erro:

```
>> p = polyft(ts,pop,7);
```

Podemos agora utilizar os coeficientes do polinômio com a função `polyval` para prever a população em 2000 como

```
>> polyval(p,(2000-1955)/35)
ans =
175.0800
```

que é muito menor do que o valor verdadeiro de 281,42. Uma melhor compreensão do problema pode ser obtida gerando um gráfico dos dados e do polinômio,

```
>> tt = linspace(1920,2000);
>> pp = polyval(p,(tt-1955)/35);
>> plot(t,pop,'o',tt,pp)
```

Como mostra a Figura 17.11, o resultado indica que o polinômio parece ajustar os dados satisfatoriamente de 1920 a 1990. No entanto, uma vez que nós nos movemos para além do alcance dos dados para o domínio da extrapolação, o polinômio de sétimo grau cai em direção à previsão errônea em 2000.

¹ Cleve Moler é um dos fundadores da The Math Works Inc., os desenvolvedores do MATLAB.

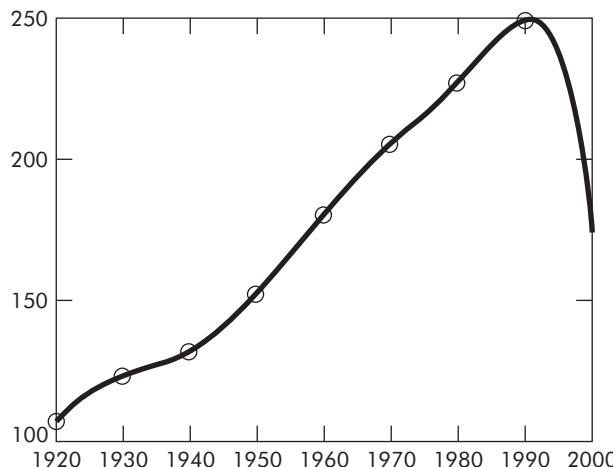


FIGURA 17.11 Utilização de um polinômio de sétima ordem para fazer uma previsão da população dos Estados Unidos em 2000 com base nos dados de 1920 a 1990.

17.5.2 Oscilações

Embora a expressão “quanto mais, melhor” seja verdade em muitos contextos, ela não é absolutamente verdadeira para interpolação polinomial. Polinômios de grau elevado tendem a ser mal condicionados – isto é, eles tendem a ser altamente sensíveis a erros de arredondamento. O exemplo a seguir ilustra bem esse ponto.

EXEMPLO 17.7 Perigos da interpolação por polinômios de grau elevado

Em 1901, Carl Runge publicou um estudo sobre os perigos do uso de polinômios de grau elevado para interpolação. Ele analisou a seguinte função de aparência simples:

$$f(x) = \frac{1}{1 + 25x^2} \quad (17.24)$$

que é hoje conhecida como *função de Runge*, e tomou pontos espaçados de forma equidistante dessa função ao longo do intervalo $[-1, 1]$. Em seguida, utilizou polinômios interpoladores de grau crescente e verificou que, à medida que mais pontos eram tomados, os polinômios e a curva original diferiam consideravelmente. Além disso, a situação deteriorava muito quando o grau era aumentado. Reproduza o resultado de Runge utilizando as funções `polyfit` e `polyval` para ajustar polinômios de quarto e décimo graus a 5 e 11 pontos igualmente espaçados gerados com a Equação (17.24). Crie gráficos de seus resultados juntamente com os valores amostrados e a função de Runge completa.

Solução. Os cinco pontos igualmente espaçados podem ser gerados como em

```
>> x = linspace(-1,1,5);
>> y = 1./(1+25*x.^2);
```

Em seguida, um vetor mais finamente espaçado de valores xx pode ser definido de modo que possamos criar um gráfico suave dos resultados:

```
>> xx = linspace(-1,1);
```

Lembre-se de que o comando `linspace` cria automaticamente 100 pontos se o número de pontos desejado não é especificado. A função `polyfit` pode ser usada para determinar os coeficientes do polinômio de quarto grau, e a função `polyval` pode ser usada para calcular a interpolação polinomial nos valores finamente espaçados de xx :

```
>> p = polyfit(x,y,4);
```

```
>> y4 = polyval(p,xx);
```

Por fim, podemos gerar valores para a própria função de Runge e traçá-los com o ajuste do polinômio e os dados amostrados:

```
>> yr = 1./(1+25*x.^2);
```

```
>> plot(x,y,'o',xx,y4,xx,yr,'--')
```

Como mostra a Figura 17.12, o polinômio não segue de forma satisfatória a função de Runge. Dando continuidade à análise, o polinômio de décima ordem pode ser gerado e traçado com

```
>> x = linspace(-1,1,11);
```

```
>> y = 1./(1+25*x.^2);
```

```
>> p = polyfit(x,y,10);
```

```
>> y10 = polyval(p,xx);
```

```
>> plot(x,y,'o',xx,y10,xx,yr,'--')
```

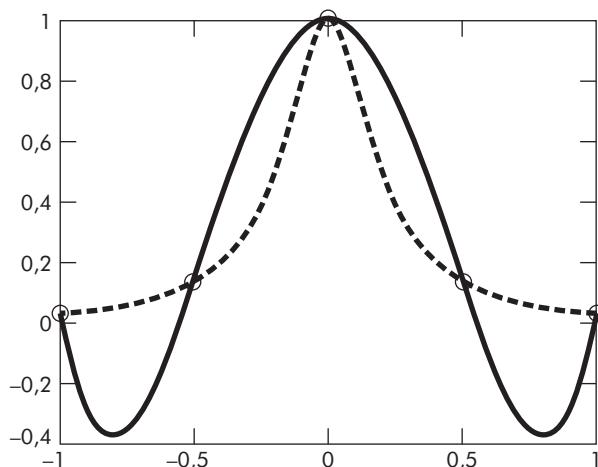


FIGURA 17.12 Comparação da função de Runge (linha tracejada) com um polinômio de quarto grau ajustado a 5 pontos amostrados da função.

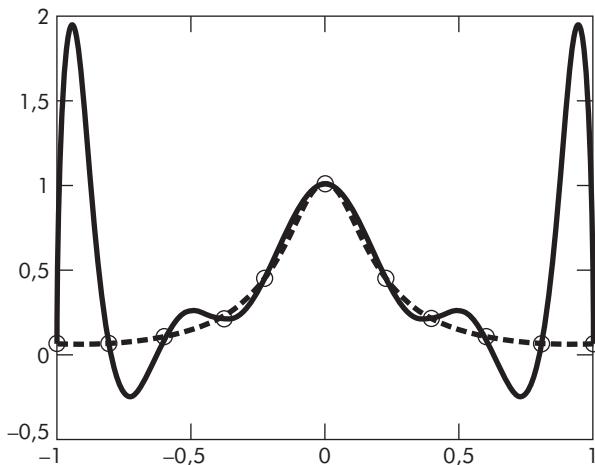


FIGURA 17.13 Comparação da função de Runge (linha tracejada) com um polinômio de décimo grau ajustado a 11 pontos amostrados da função.

Como mostra a Figura 17.13, o ajuste ficou ainda pior, particularmente nas extremidades do intervalo!

Embora possam existir certos contextos em que polinômios de grau elevado sejam necessários, eles são geralmente evitados. Em muitos contextos de ciência e de engenharia, polinômios de grau menor, do tipo descrito neste capítulo, podem ser efetivamente utilizados para capturar as tendências curvas dos dados, sem sofrer com as oscilações.

►► PROBLEMAS

- 17.1** Os dados a seguir são provenientes de uma tabela que foi medida com alta precisão. Use o melhor método numérico (para este tipo de problema) para determinar y em $x = 3,5$. Observe que um polinômio produzirá um valor exato. Sua solução deve provar que seu resultado é exato.

x	0	1,8	5	6	8,2	9,2	12
y	26	16,415	5,375	3,5	2,015	2,54	8

- 17.2** Use polinômios interpoladores de Newton para determinar y em $x = 3,5$ para a melhor precisão possível. Calcule as diferenças divididas finitas como na Figura 17.5 e ordene seus pontos para obter acurácia e convergência ótimas; ou seja, os pontos devem ser centrados em torno e tão próximo quanto possível do valor desconhecido.

x	0	1	2,5	3	4,5	5	6
y	2	5,4375	7,3516	7,5625	8,4453	9,1875	12

- 17.3** Utilize polinômios interpoladores de Newton para determinar y em $x = 8$ para a melhor precisão possível. Calcule as diferenças divididas finitas como na Figura 17.5 e ordene seus pontos para obter acurácia e convergência ótimas; ou seja, os pontos devem ser centrados em torno e tão próximo quanto possível do valor desconhecido.

x	0	1	2	5,5	11	13	16	18
y	0,5	3,134	5,3	9,9	10,2	9,35	7,2	6,2

- 17.4** Considere os dados

x	1	2	2,5	3	4	5
$f(x)$	0	5	6,5	7	3	1

- (a) Calcule $f(3,4)$ usando polinômios interpoladores de Newton de primeiro a terceiro graus. Escolha a sequência de pontos para fazer sua estimativa de modo a atingir a melhor acurácia possível.
 (b) Repita (a), mas utilize polinômios de Lagrange.

17.5 Considere os dados

x	1	2	3	5	6
$f(x)$	7	4	5,5	40	82

Calcule $f(4)$ usando polinômios interpoladores de Newton de primeiro a quarto graus. Escolha seus pontos-base para obter uma boa acurácia; ou seja, os pontos devem ser centrados em torno e tão próximo quanto possível do valor desconhecido. O que seus resultados indicam em relação ao grau do polinômio usado para gerar os dados da tabela?

17.6 Repita o Problema 17.5, porém utilizando polinômios de Lagrange de primeiro a terceiro graus.

17.7 A Tabela P15.5 lista valores para a concentração de oxigênio dissolvido em água como uma função da temperatura e da concentração de cloreto.

(a) Use interpolação quadrática e cúbica para determinar a concentração de oxigênio para $T = 12^\circ\text{C}$ e $c = 10 \text{ g/L}$.

(b) Use interpolação linear para determinar a concentração de oxigênio para $T = 12^\circ\text{C}$ e $c = 15 \text{ g/L}$.

(c) Repita (b) usando interpolação quadrática.

17.8 Empregue interpolação inversa utilizando um polinômio interpolador cúbico e bissecção para determinar o valor de x que corresponde a $f(x) = 1,7$ para os seguintes dados tabulados:

x	1	2	3	4	5	6	7
$f(x)$	3,6	1,8	1,2	0,9	0,72	1,5	0,51429

17.9 Empregue interpolação inversa para determinar o valor de x que corresponde a $f(x) = 0,85$ para os seguintes dados tabulados:

x	0	1	2	3	4	5
$f(x)$	0	0,5	0,8	0,9	0,941176	0,961538

Observe que os valores na tabela foram gerados pela função $f(x) = x^2/(1 + x^2)$.

(a) Determine o valor correto analiticamente.

(b) Use interpolação e a fórmula quadráticas para determinar o valor numericamente.

(c) Utilize interpolação cúbica e bissecção para determinar o valor numericamente.

17.10 Use a porção dada a seguir de uma tabela de vapor-d'água superaquecido em 200 MPa para encontrar (a) a entropia correspondente s para um volume específico v de 0,118 com interpolação

linear, (b) a mesma entropia correspondente utilizando interpolação quadrática, e (c) o volume correspondente a uma entropia de 6,45 utilizando interpolação inversa.

$v, \text{m}^3/\text{kg}$	0,10377	0,11144	0,12547
$s, \text{kJ}/(\text{kg K})$	6,4147	6,5453	6,7664

17.11 Os dados a seguir para a densidade do gás nitrogênio *versus* a temperatura são provenientes de uma tabela que foi medida com alta precisão. Use polinômios de primeiro a quinto graus para fazer uma estimativa da densidade em uma temperatura de 330 K. Qual é sua melhor estimativa? Empregue essa melhor estimativa e interpolação inversa para determinar a temperatura correspondente.

T, K	200	250	300	350	400	450
Densidade, kg/m^3	1,708	1,367	1,139	0,967	0,854	0,759

17.12 A lei de Ohm estabelece que a queda de tensão V através de um resistor ideal é linearmente proporcional à corrente i passando pelo resistor como em $V = iR$, onde R é a resistência. Entretanto, resistores reais nem sempre podem obedecer à lei de Ohm. Considere que você fez algumas experiências muito precisas para medir a queda de tensão e a corrente resultante para um resistor. Os resultados apresentados na tabela a seguir sugerem uma relação curvilínea em vez da reta representada pela lei de Ohm.

i	-1	-0,5	-0,25	0,25	0,5	1
V	-637	-96,5	-20,5	20,5	96,5	637

Para quantificar essa relação, uma curva deve ser ajustada aos dados. Devido aos erros nas medidas, a regressão seria o método preferido de ajuste de curva para analisar esses dados experimentais. Entretanto, o fato de essa relação ser suave e a precisão dos métodos experimentais sugerem que a interpolação poderia ser apropriada. Use um polinômio interpolador de quinto grau para ajustar os dados e calcule V para $i = 0,10$.

17.13 As funções de Bessel aparecem com frequência em análises avançadas de engenharia tais como o estudo de campos elétricos. Aqui estão alguns valores selecionados da função de Bessel de primeira tipo e de primeira ordem

<i>x</i>	1,8	2,0	2,2	2,4	2,6
$J_1(x)$	0,5815	0,5767	0,5560	0,5202	0,4708

Estime $J_1(2,1)$ utilizando polinômios interpoladores de terceiro e quarto graus. Determine o erro relativo percentual para cada caso com base no valor verdadeiro, que pode ser calculado com a função nativa `besselj` do MATLAB.

17.14 Repita o Exemplo 17.6, porém utilizando polinômios interpoladores de primeiro, segundo, terceiro e quarto graus para prever a população em 2000, com base nos dados mais recentes; ou seja, para a previsão linear, use os dados de 1980 a 1990, para a previsão quadrática, use os dados de 1970, 1980 e 1990, e assim por diante. Qual abordagem produz o melhor resultado?

17.15 O volume específico de um vapor superaquecido está listado em tabelas de vapor para diversas temperaturas. Por exemplo, na pressão absoluta de 3.000 lb/pol²:

<i>T, °C</i>	370	382	394	406	418
<i>v, Litro/kg</i>	5,9313	7,5838	8,8428	9,796	10,5311

Determine *v* em *T* = 400 °C.

17.16 A tensão vertical σ_z sob o canto de uma área retangular sujeita a uma carga uniforme de intensidade *q* é dada pela solução da equação de Boussinesq:

$$\sigma_z = \frac{q}{4\pi} \left[\frac{2mn\sqrt{m^2 + n^2 + 1}}{m^2 + n^2 + 1 + m^2n^2} \frac{m^2 + n^2 + 2}{m^2 + n^2 + 1} + \operatorname{sen}^{-1} \left(\frac{2mn\sqrt{m^2 + n^2 + 1}}{m^2 + n^2 + 1 + m^2n^2} \right) \right]$$

Como é difícil resolver manualmente essa equação, ela foi reformulada como

$$\sigma_z = q f_z(m, n)$$

onde $f_z(m, n)$ é chamado valor de influência e *m* e *n* são razões adimensionais, com $m = a/z$ e $n = b/z$ e *a* e *b* são definidos na Figura P17.16. O valor da influência é então tabulado, uma parte dos quais está dada na Tabela P17.16. Se *a* = 4,6 e *b* = 14, use um polinômio interpolador de terceiro grau para calcular σ_z a uma profundidade de 10 m abaixo do canto de um piso retangular que está sujeito a uma carga total de 100 t (toneladas métricas). Expressse sua resposta em toneladas por metros quadrados. Observe que *q* é igual à carga por área.

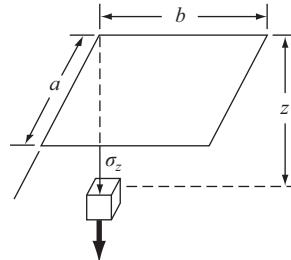


FIGURA P17.16

TABELA P17.16

<i>m</i>	<i>n</i> = 1,2	<i>n</i> = 1,4	<i>n</i> = 1,6
0,1	0,02926	0,03007	0,03058
0,2	0,05733	0,05894	0,05994
0,3	0,08323	0,08561	0,08709
0,4	0,10631	0,10941	0,11135
0,5	0,12626	0,13003	0,13241
0,6	0,14309	0,14749	0,15027
0,7	0,15703	0,16199	0,16515
0,8	0,16843	0,17389	0,17739

17.17 Você mediou a queda de tensão *V* através de um resistor para diversos valores diferentes de corrente *i*. Os resultados são

<i>i</i>	0,25	0,75	1,25	1,5	2,0
<i>V</i>	-0,45	-0,6	0,70	1,88	6,0

Use interpolação polinomial de primeiro a quarto graus para fazer uma estimativa da queda de tensão para *i* = 1,15. Interprete seus resultados.

17.18 A corrente em um fio é medida com grande precisão como uma função do tempo:

<i>t</i>	0	0,1250	0,2500	0,3750	0,5000
<i>i</i>	0	6,24	7,75	4,85	0,0000

Determine *i* em *t* = 0,23.

17.19 A aceleração da gravidade a uma altitude *y* acima da superfície da Terra é dada por

<i>y, m</i>	0	30.000	60.000	90.000	120.000
<i>g, m/s²</i>	9,8100	9,7487	9,6879	9,6278	9,5682

Calcule *g* em *y* = 55.000 m.

17.20 As temperaturas são medidas em diversos pontos de uma placa aquecida (Tabela P17.20). Faça uma estimativa da temperatura em (a) *x* = 4, *y* = 3,2, e (b) *x* = 4,3, *y* = 2,7.

TABELA P17.20 Temperatura ($^{\circ}\text{C}$) em diversos pontos de uma placa aquecida quadrada.

	$x = 0$	$x = 2$	$x = 4$	$x = 6$	$x = 8$
$y = 0$	100,00	90,00	80,00	70,00	60,00
$y = 2$	85,00	64,49	53,50	48,15	50,00
$y = 4$	70,00	48,90	38,43	35,03	40,00
$y = 6$	55,00	38,78	30,39	27,07	30,00
$y = 8$	40,00	35,00	30,00	25,00	20,00

17.21 Use a parte da tabela de vapor dada para H_2O superaquecida a 200 MPa para (a) encontrar a entropia correspondente s para um volume específico v de $0,108 \text{ m}^3/\text{kg}$ com interpolação linear, (b) encontrar a mesma entropia correspondente usando interpolação quadrática, e (c) encontrar o volume correspondente a uma entropia de 6,6 usando interpolação inversa.

$v (\text{m}^3/\text{kg})$	0,10377	0,11144	0,12540
$s (\text{kJ/kg} \cdot \text{K})$	6,4147	6,5453	6,7664

Interpolação por splines e por partes

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar a interpolação por splines. Após esta leitura completa, você será capaz de:

- ✓ Entender que os splines minimizam as oscilações ajustando, por partes, polinômios de ordem mais baixa aos dados.
- ✓ Aprender como desenvolver um código para implementar uma tabela de consulta.
- ✓ Reconhecer por que polinômios cúbicos são preferíveis aos quadráticos e splines de ordem mais alta.
- ✓ Entender as condições que fundamentam um ajuste por spline cúbico.
- ✓ Compreender as diferenças entre as condições de extremidade natural, amarrada e sem um nó.
- ✓ Aprender como ajustar um spline aos dados com funções nativas do MATLAB.
- ✓ Entender como uma interpolação multidimensional é implementada com o MATLAB.

18.1 INTRODUÇÃO AOS SPLINES

No Capítulo 17, polinômios de grau $(n - 1)$ foram usados para interpolar entre n pontos dados. Por exemplo, para oito pontos, pode-se determinar exatamente um polinômio de grau sete. Essa curva capturaria todas as sinuosidades (pelo menos até a sétima derivada, inclusive) sugeridas por esses pontos. Entretanto, há casos em que essas funções podem levar a resultados errôneos devido a erros de arredondamento e oscilações. Uma abordagem alternativa é aplicar polinômios de grau mais baixo a subconjuntos de pontos dados, de uma maneira por partes. Esses polinômios conectores são chamados de *funções splines*.

Curvas de terceiro grau, por exemplo, usadas para conectar cada par de pontos dados são chamadas de *splines cúbicos*. Essas funções podem ser construídas de modo que as conexões entre equações cúbicas adjacentes sejam visualmente suaves. Superficialmente, parece que a aproximação de terceiro grau por splines é inferior à expressão de grau sete. Você pode perguntar em quais situações um spline é preferível.

A Figura 18.1 ilustra uma situação na qual um spline funciona melhor do que um polinômio de grau mais alto. Esse é o caso quando uma função é geralmente suave, mas sofre uma mudança abrupta em algum ponto da região de interesse. O aumento em degrau apresentado na Figura 18.1 é um exemplo extremo de tal mudança e serve para ilustrar esse ponto.

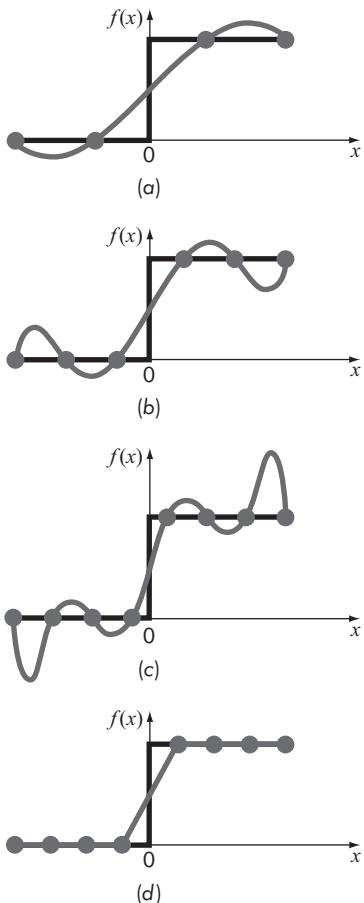


FIGURA 18.1 Representação visual de uma situação na qual os splines são superiores aos polinômios interpoladores de grau mais alto. A função a ser ajustada sofre uma mudança brusca em $x = 0$. As partes (a) a (c) indicam que a variação abrupta induz oscilações nos polinômios interpoladores. Já um spline linear, como é limitado a segmentos de reta, (d) fornece uma aproximação muito mais aceitável.

As Figuras 18.1a a 18.1c ilustram como polinômios de grau mais alto tendem a passar por grandes oscilações na vizinhança de uma mudança abrupta. Em contraste, o spline também liga os pontos, mas como é limitado a variações de grau mais baixo, as oscilações são mantidas em um mínimo. Dessa forma, os splines, em geral, fornecem uma aproximação superior do comportamento de funções que têm variações locais abruptas.

O conceito de splines originou-se de uma técnica de desenho na qual era usada uma faixa flexível (chamada *spline*) para desenhar uma curva suave passando por um conjunto de pontos. O processo é descrito na Figura 18.2 para uma série de cinco pinos (pontos dados). Nessa técnica, o desenhista coloca papel sobre uma tábua de madeira e prega tachinhas ou pinos no papel (e na tábua) nas posições dos pontos dados. Uma curva cúbica suave resulta do entrelaçamento da faixa entre os pinos. Assim, o nome “spline cúbico” foi adotado para polinômios desse tipo.

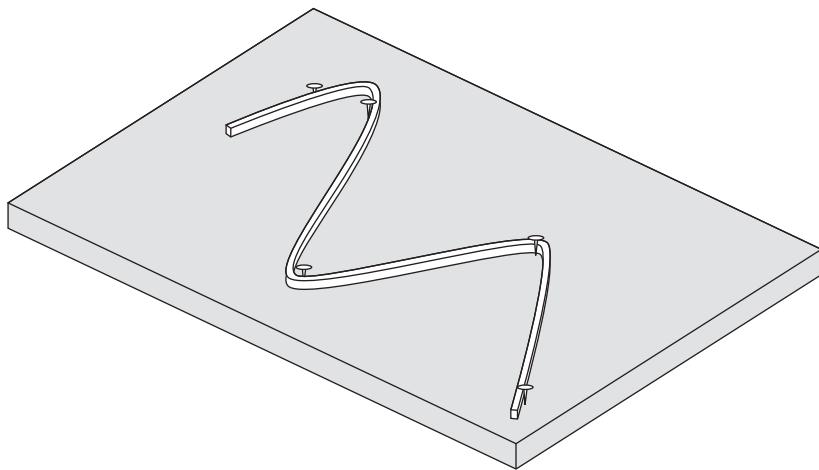


FIGURA 18.2 A técnica de desenho que usa um spline para desenhar curvas suaves por uma série de pontos. Observe como, nas extremidades, o spline se torna menos curvo. Isso é chamado de spline “natural”.

Neste capítulo, a princípio serão usadas funções lineares simples para introduzir alguns conceitos básicos e problemas associados à interpolação por splines. Então, desenvolveremos um algoritmo para ajustar splines quadráticos aos dados. Em seguida, serão apresentados os splines cúbicos, que são a versão mais comum e útil na engenharia e na ciência. Por fim, descrevemos as capacidades do MATLAB para interpolação por partes, incluindo sua habilidade para gerar splines.

18.2 SPLINES LINEARES

A notação utilizada para splines é apresentada na Figura 18.3. Para n pontos dados ($i = 1, 2, \dots, n$), existem $n - 1$ intervalos. Cada intervalo i tem sua própria função spline, $s_i(x)$. Para splines lineares, cada função é simplesmente uma reta que liga dois pontos em cada extremidade do intervalo, e é formulada como

$$s_i(x) = a_i + b_i(x - x_i) \quad (18.1)$$

onde a_i é a interseção com o eixo y , que é definida como

$$a_i = f_i \quad (18.2)$$

e b_i é a inclinação da reta ligando os pontos:

$$b_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} \quad (18.3)$$

onde f_i é uma abreviação para $f(x_i)$. Substituindo as Equações (18.1) e (18.2) na Equação (18.3), tem-se:

$$s_i(x) = f_i + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}(x - x_i) \quad (18.4)$$

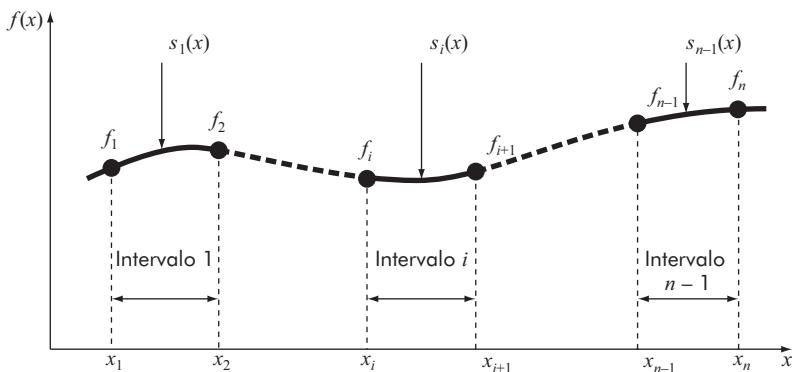


FIGURA 18.3 Notação usada para deduzir splines. Observe que há $n - 1$ intervalos e n pontos dados.

Essas equações podem ser usadas para calcular a função em qualquer ponto entre x_1 e x_n , a princípio localizando o intervalo no qual o ponto se encontra. A seguir, a equação apropriada é usada para determinar o valor da função dentro do intervalo. Uma inspeção da Equação (18.4) indica que o spline linear equivale à utilização de um polinômio de Newton de primeiro grau [Equação (17.5)] para interpolar dentro de cada intervalo.

EXEMPLO 18.1 Splines de primeiro grau

Ajuste os dados da Tabela 18.1 com splines de primeiro grau. Calcule a função em $x = 5$.

TABELA 18.1 Dados a serem ajustados com funções splines.

i	x_i	f_i
1	3,0	2,5
2	4,5	1,0
3	7,0	2,5
4	9,0	0,5

Solução. Os dados podem ser substituídos na Equação (18.4) para gerar as funções do spline linear. Por exemplo, para o segundo intervalo de $x = 4,5$ a $x = 7$, a função é

$$s_2(x) = 1,0 + \frac{2,5 - 1,0}{7,0 - 4,5}(x - 4,5)$$

As equações para os outros intervalos podem ser calculadas, e o spline de primeiro grau resultante está traçado na Figura 18.4a. O valor em $x = 5$ é 1,3.

$$s_2(x) = 1,0 + \frac{2,5 - 1,0}{7,0 - 4,5}(5 - 4,5) = 1,3$$

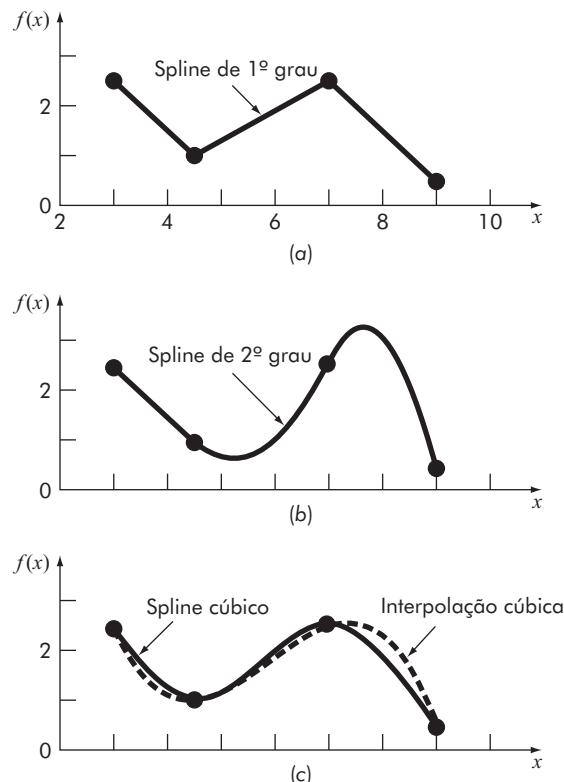


FIGURA 18.4 Ajuste por splines de um conjunto de quatro pontos. (a) Spline linear, (b) spline quadrático e (c) spline cúbico, com um polinômio interpolador cúbico também traçado.

A inspeção visual da Figura 18.4a indica que a principal desvantagem dos splines de primeiro grau é que eles não são suaves. Essencialmente, nos pontos dados nos quais dois splines se encontram (chamados *nós*), a inclinação varia abruptamente. Em termos formais, a primeira derivada da função é descontínua nesses pontos. Essa deficiência é superada usando-se splines polinomiais de grau mais alto, que garantem que eles sejam suaves nos nós igualando as derivadas em tais pontos, conforme será discutido posteriormente. Antes disso, a seção a seguir fornece uma aplicação em que splines lineares são úteis.

18.2.1 Tabela de consulta

Uma tabela de consulta é um procedimento comum frequentemente encontrado em aplicações de engenharia e de ciência. Ele é útil para efetuar interpolações repetidas de uma tabela de variáveis independentes e dependentes. Por exemplo, considere que se queira desenvolver um arquivo-M que empregue interpolação linear para determinar a densidade do ar em uma temperatura particular com base nos dados da Tabela 17.1. Uma maneira de fazer isso é passar para o arquivo-M a temperatura na qual se queira que a interpolação seja realizada juntamente com os dois valores

adjacentes. Uma abordagem mais geral é passar vetores que contêm todos os dados da tabela e deixar o arquivo-M determinar o intervalo. Isso é chamado *tabela de consulta*.

Assim, o arquivo-M realiza duas tarefas. Primeiro, procura o vetor da variável independente para encontrar o intervalo que contém o valor desconhecido. Em seguida, realiza a interpolação linear utilizando uma das técnicas descritas neste capítulo ou no Capítulo 17.

Para dados ordenados, existem duas maneiras simples de encontrar o intervalo. A primeira é chamada *busca sequencial*. Como o nome indica, esse método envolve a comparação do valor desejado com cada elemento do vetor em sequência até o intervalo ser localizado. Para dados em ordem crescente, isso pode ser feito testando se o valor desconhecido (ponto a ser interpolado) é menor que o valor sendo avaliado. Se essa hipótese for confirmada, sabemos que o valor desconhecido cai entre esse valor e o anterior que foi examinado. Se não, movemos para o próximo valor e repetimos a comparação. A seguinte função simples do MATLAB cumpre esse objetivo:

```
function yi = TabelaConsulta(x, y, xx)
n = length(x);
if xx < x(1) | xx > x(n)
    error('Interpolação fora do intervalo')
end
% busca sequencial
i = 1;
while(1)
    if xx <= x(i + 1), break, end
    i = i + 1;
end
% interpolação linear
yi = y(i) + (y(i+1)-y(i))/(x(i+1)-x(i))*(xx-x(i));
```

As variáveis independentes da tabela são armazenadas em ordem crescente no arranjo *x*, e as variáveis dependentes são armazenadas no arranjo *y*. Antes de iniciar a busca, deve-se incluir uma lógica para detectar erros, de modo a garantir que o valor desejado *xx* esteja dentro do intervalo dos *x*'s. Um laço *while ... break* compara o valor no qual a interpolação é desejada, *xx*, para determinar se ele é menor que o valor na parte superior do intervalo, *x(i+1)*. Para casos em que *xx* está no segundo intervalo ou superior, o resultado desse teste não será verdadeiro de início. Nesse caso, o contador *i* é incrementado por 1, de modo que na próxima iteração, *xx* é comparado com o valor na parte superior do segundo intervalo. O laço é repetido até que *xx* seja menor ou igual ao limite superior do intervalo, caso em que o laço é encerrado. Neste ponto, a interpolação pode ser efetuada simplesmente como mostrado.

Nas situações em que existe uma grande quantidade de dados, a ordenação sequencial é ineficiente, porque se deve procurar por todos os pontos anteriores para encontrar valores. Nesses casos, uma alternativa simples é a *busca binária*. A função do MATLAB a seguir realiza uma busca binária seguida pela interpolação linear:

```
function yi = TabelaConsultaBin(x, y, xx)
n = length(x);
```

```

if xx < x(1) | xx > x(n)
    error('Interpolação fora do intervalo')
end
% busca binária
iL = 1; iU = n;
while (1)
    if iU - iL <= 1, break, end
    iM = fx((iL + iU) / 2);
    if x(iM) < xx
        iL = iM;
    else
        iU = iM;
    end
end
% interpolação linear
yi = y(iL) + (y(iL+1)-y(iL))/(x(iL+1)-x(iL))*(xx - x(iL));

```

A abordagem é semelhante ao método da bissecção para localização de raízes. Como na bissecção, o índice no ponto central i_M é calculado como a média do primeiro índice, ou índice “inferior”, $i_L = 1$ e do último índice, ou índice “superior”, $i_U = n$. O valor desconhecido xx é então comparado com o valor de x no ponto central $x(i_M)$ para avaliar se ele está na metade inferior do arranjo ou na metade superior. Dependendo de sua localização, o índice inferior ou o superior é redefinido como igual ao índice central. O processo é repetido até que a diferença entre os índices superior e inferior seja menor ou igual a zero. Nesse ponto, o índice inferior encontra-se no limite inferior do intervalo contendo xx , o laço é encerrado e a interpolação linear é realizada.

A sessão do MATLAB a seguir ilustra como a função de busca binária pode ser aplicada para calcular a densidade do ar em 350 °C com base nos dados da Tabela 17.1. A busca sequencial seria similar.

```

>> T = [-40 0 20 50 100 150 200 250 300 400 500];
>> densidade = [1.52 1.29 1.2 1.09 .946 .935 .746 .675 .616...
    .525 .457];
>> TabelaConsultaBin(T,densidade,350)

ans =
    0.5705

```

O resultado pode ser verificado por cálculos efetuados à mão:

$$f(350) = 0,616 + \frac{0,525 - 0,616}{400 - 300}(350 - 300) = 0,5705$$

18.3 SPLINES QUADRÁTICOS

Para garantir que as n -ésimas derivadas sejam contínuas nos nós, um spline de grau pelo menos $n + 1$ deve ser usado. Polinômios de terceiro grau ou splines cúbicos que garantem continuidade das primeira e segunda derivadas são usados mais

frequentemente na prática. Embora derivadas de terceira ordem ou de ordem mais alta possam ser descontínuas quando se utiliza splines cúbicos, geralmente não podem ser detectadas visualmente e, consequentemente, são ignoradas.

Como a dedução dos splines cúbicos é um pouco complicada, optamos por ilustrar primeiro o conceito de interpolação por splines utilizando polinômios de segundo grau. Esses “splines quadráticos” têm primeira derivada contínua nos nós, e embora não tenham grande importância prática, servem bem para demonstrar o procedimento geral no desenvolvimento de splines de grau mais alto. Nestes, o objetivo é determinar um polinômio de segundo grau para cada intervalo entre os pontos dados. O polinômio para cada intervalo pode ser representado de forma geral como

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 \quad (18.5)$$

onde a notação é como mostra a Figura 18.3. Para n pontos dados ($i = 1, 2, \dots, n$), existem $n - 1$ intervalos e, consequentemente, $3(n - 1)$ constantes indeterminadas (os a ’s, b ’s e c ’s) para calcular. Portanto, $3(n - 1)$ equações ou condições são necessárias para calcular as incógnitas. Essas podem ser desenvolvidas do seguinte modo:

1. A função deve passar por todos os pontos. Isso é chamado de *condição de continuidade*. Ela pode ser expressa matematicamente como

$$f_i = a_i + b_i(x_i - x_i) + c_i(x_i - x_i)^2$$

o que simplifica para

$$a_i = f_i \quad (18.6)$$

Portanto, a constante em cada equação quadrática deve ser igual ao valor da variável dependente no início do intervalo, resultado que pode ser incorporado na Equação (18.5):

$$s_i(x) = f_i + b_i(x - x_i) + c_i(x - x_i)^2$$

Observe que, como um dos coeficientes foi determinado, o número de condições a ser avaliado foi agora reduzido para $2(n - 1)$.

2. Os valores da função de polinômios adjacentes devem ser iguais nos nós, condição que pode ser escrita para o nó $i + 1$ como

$$f_i + b_i(x_{i+1} - x_i) + c_i(x_{i+1} - x_i)^2 = f_{i+1} + b_{i+1}(x_{i+1} - x_{i+1}) + c_{i+1}(x_{i+1} - x_{i+1})^2 \quad (18.7)$$

Essa equação pode ser simplificada matematicamente pela definição da largura do i -ésimo intervalo como

$$h_i = x_{i+1} - x_i$$

Assim, a Equação (18.7) simplifica para

$$f_i + b_i h_i + c_i h_i^2 = f_{i+1} \quad (18.8)$$

E esta equação pode ser escrita para os nós $i = 1, \dots, n - 1$. Uma vez que isso equivale a $n - 1$ condições, significa que existem $2(n - 1) - (n - 1) = n - 1$ condições restantes.

3. As primeiras derivadas nos nós internos devem ser iguais. Essa é uma condição importante, pois indica que splines quadráticos adjacentes vão se unir suavemente, em vez de na forma “dentada” que vimos para os splines lineares (ver Figura 18.4a). A Equação (18.5) pode ser derivada para produzir

$$s'_i(x) = b_i + 2c_i(x - x_i)$$

A equivalência das derivadas em um nó interior, $i + 1$, pode, portanto, ser escrita como

$$b_i + 2c_i h_i = b_{i+1} \quad (18.9)$$

Escrever essa equação para todos os nós interiores equivale a $n - 2$ condições. Isso significa que há $n - 1 - (n - 2) = 1$ condição restante. A menos que se tenha alguma informação adicional relativa às funções ou suas derivadas, é preciso fazer uma escolha arbitrária para ter sucesso no cálculo das constantes. Embora existam diversas escolhas diferentes possíveis, optou-se pela condição descrita a seguir.

4. Considere que a derivada segunda seja nula no primeiro ponto. Como a derivada segunda da Equação (18.5) é $2c_1$, essa condição pode ser expressa matematicamente como

$$c_1 = 0$$

A interpretação visual dessa condição é que os primeiros dois pontos serão ligados por uma reta.

EXEMPLO 18.2 Splines quadráticos

Ajuste um spline quadrático aos mesmos dados empregados no Exemplo 18.1 (Tabela 18.1). Use os resultados para fazer uma estimativa do valor em $x = 5$.

Solução. Nesse problema, há quatro pontos dados e $n = 3$ intervalos. Portanto, depois de aplicar a condição de continuidade e a condição de derivada segunda nula, significa que $2(4 - 1) - 1 = 5$ condições são necessárias. A Equação (18.8) é escrita para $i = 1$ a 3 (com $c_1 = 0$) para fornecer

$$\begin{aligned} f_1 + b_1 h_1 &= f_2 \\ f_2 + b_2 h_2 + c_2 h_2^2 &= f_3 \\ f_3 + b_3 h_3 + c_3 h_3^2 &= f_4 \end{aligned}$$

A continuidade das derivadas, Equação (18.9), produz mais $3 - 1 = 2$ condições adicionais (novamente, lembre-se que $c_1 = 0$):

$$\begin{aligned} b_1 &= b_2 \\ b_2 + 2c_2 h_2 &= b_3 \end{aligned}$$

Os valores necessários da função e as larguras dos intervalos são:

$$\begin{array}{ll} f_1 = 2,5 & h_1 = 4,5 - 3,0 = 1,5 \\ f_2 = 1,0 & h_2 = 7,0 - 4,5 = 2,5 \\ f_3 = 2,5 & h_3 = 9,0 - 7,0 = 2,0 \\ f_4 = 0,5 & \end{array}$$

Esses valores podem ser substituídos nas condições, as quais podem ser expressas na forma matricial como

$$\left[\begin{array}{ccccc} 1,5 & 0 & 0 & 0 & 0 \\ 0 & 2,5 & 6,25 & 0 & 0 \\ 0 & 0 & 0 & 2 & 4 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 5 & -1 & 0 \end{array} \right] \left\{ \begin{array}{l} b_1 \\ b_2 \\ c_2 \\ b_3 \\ c_3 \end{array} \right\} = \left\{ \begin{array}{l} -1,5 \\ 1,5 \\ -2 \\ 0 \\ 0 \end{array} \right\}$$

Essas equações podem ser resolvidas utilizando o MATLAB com os resultados:

$$\begin{array}{ll} b_1 = -1 & \\ b_2 = -1 & c_2 = 0,64 \\ b_3 = 2,2 & c_3 = -1,6 \end{array}$$

Esses resultados, juntamente com os valores para os a 's [Equação (18.6)], podem ser substituídos nas equações quadráticas originais para desenvolver os seguintes splines quadráticos para cada intervalo:

$$\begin{aligned} s_1(x) &= 2,5 - (x - 3) \\ s_2(x) &= 1,0 - (x - 4,5) + 0,64(x - 4,5)^2 \\ s_3(x) &= 2,5 + 2,2(x - 7,0) - 1,6(x - 7,0)^2 \end{aligned}$$

Como $x = 5$ está no segundo intervalo, utiliza-se s_2 para fazer a previsão,

$$s_2(5) = 1,0 - (5 - 4,5) + 0,64(5 - 4,5)^2 = 0,66$$

O ajuste completo do spline quadrático é mostrado na Figura 18.4b. Observe que há duas desvantagens que prejudicam o ajuste: (1) a reta ligando os dois primeiros pontos e (2) o spline para o último intervalo parece ir até muito alto. Os splines cúbicos da próxima seção não exibem essas desvantagens e, por isso, são métodos melhores para interpolação por splines.

18.4 SPLINES CÚBICOS

Conforme mencionado no início da seção anterior, os splines cúbicos são mais frequentemente utilizados na prática. As deficiências dos splines lineares e quadráticos já foram discutidas. Splines de quarto grau ou de grau mais alto não são usados porque tendem a exibir as instabilidades inerentes a polinômios de ordem elevada. Os splines cúbicos são preferidos porque fornecem a representação mais simples que exibe a aparência desejada de suavidade, e tem por objetivo determinar

um polinômio de terceiro grau para cada intervalo entre nós, como representado de forma geral por

$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (18.10)$$

Logo, para n pontos dados ($i = 1, 2, \dots, n$), existem $n - 1$ intervalos e $4(n - 1)$ coeficientes desconhecidos para calcular. Consequentemente, $4(n - 1)$ condições são necessárias para calcular esses coeficientes.

As primeiras condições são idênticas àquelas usadas para o caso quadrático; ou seja, elas são definidas de modo que as funções passem pelos pontos e que as primeiras derivadas nos nós sejam iguais. Além dessas, outras condições são desenvolvidas para garantir que as segundas derivadas nos nós também sejam iguais, o que melhora muito a suavidade do ajuste.

Depois de desenvolver essas condições, duas adicionais são necessárias para obter a solução. Esse é um resultado bem melhor que o ocorrido para os splines quadráticos, em que foi necessário especificar uma única condição. Naquele caso, tivemos que especificar arbitrariamente uma segunda derivada igual a zero para o primeiro intervalo, tornando, assim, o resultado assimétrico. Para splines cúbicos, são necessárias duas condições adicionais, o que representa uma vantagem, já que podem ser aplicadas igualmente em ambas as extremidades e formuladas de diversas maneiras diferentes.

Uma abordagem bastante comum é assumir que as segundas derivadas no primeiro e último nós são iguais a zero. A interpretação visual para essas condições é que a função se torna uma reta nos nós extremos. A especificação de tais condições nas extremidades leva ao que é chamado de spline “natural”, que tem esse nome porque o spline desenhado naturalmente se comporta dessa forma (Figura 18.2).

Há uma série de outras condições que podem ser especificadas na extremidade. Duas das mais populares são a condição “amarrada” (*clamped condition*) e a condição “sem um nó” (*not-a-knot condition*), que serão descritas na Seção 18.4.2. Para a dedução a seguir, nos limitaremos aos splines naturais.

Uma vez especificadas as condições adicionais para as extremidades, teremos as $4(n - 1)$ condições necessárias para calcular os $4(n - 1)$ coeficientes desconhecidos. Embora certamente seja possível desenvolver splines cúbicos dessa forma, apresentaremos uma abordagem alternativa que exige a solução de apenas $n - 1$ equações. Além disso, as equações simultâneas serão tridiagonais e, portanto, podem ser resolvidas de forma muito eficiente. Embora a dedução dessa abordagem seja um pouco menos simples que a dos splines quadráticos, o ganho em eficiência vale o esforço.

18.4.1 Dedução dos splines cúbicos

De modo semelhante aos splines quadráticos, a primeira condição é que o spline deve passar por todos os pontos dados.

$$f_i = a_i + b_i(x_i - x_i) + c_i(x_i - x_i)^2 + d_i(x_i - x_i)^3$$

o que simplifica para

$$a_i = f_i \quad (18.11)$$

Portanto, a constante em cada spline cúbico deve ser igual ao valor da variável dependente no início do intervalo. Esse resultado pode ser incorporado na Equação (18.10):

$$s_i(x) = f_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (18.12)$$

Em seguida, aplica-se a condição que cada um dos splines cúbicos devem se unir nos nós. Para o nó $i + 1$, isso pode ser representado como

$$f_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = f_{i+1} \quad (18.13)$$

onde

$$h_i = x_{i+1} - x_i$$

As primeiras derivadas nos nós interiores devem ser iguais. A Equação (18.12) é derivada para produzir

$$s'_i(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \quad (18.14)$$

A equivalência das derivadas em um nó interior, $i + 1$, pode, portanto, ser escrita como

$$b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1} \quad (18.15)$$

As segundas derivadas nos nós interiores também devem ser iguais. A Equação (18.14) pode ser derivada para produzir

$$s''_i(x) = 2c_i + 6d_i(x - x_i) \quad (18.16)$$

A equivalência das segundas derivadas em um nó interior, $i + 1$, pode, portanto, ser escrita como

$$c_i + 3d_i h_i = c_{i+1} \quad (18.17)$$

Em seguida, pode-se resolver a Equação (18.17), para determinar d_i :

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \quad (18.18)$$

Isso pode ser substituído na Equação (18.13) para fornecer

$$f_i + b_i h_i + \frac{h_i^2}{3}(2c_i + c_{i+1}) = f_{i+1} \quad (18.19)$$

A Equação (18.18) também pode ser substituída na Equação (18.15), para fornecer

$$b_{i+1} = b_i + h_i(c_i + c_{i+1}) \quad (18.20)$$

A Equação (18.19) pode ser resolvida para

$$b_i = \frac{f_{i+1} - f_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1}) \quad (18.21)$$

O índice dessa equação pode ser reduzido por 1:

$$b_{i-1} = \frac{f_i - f_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{3}(2c_{i-1} + c_i) \quad (18.22)$$

O índice da Equação (18.20) também pode ser reduzido por 1:

$$b_i = b_{i-1} + h_{i-1}(c_{i-1} + c_i) \quad (18.23)$$

As Equações (18.21) e (18.22) podem ser substituídas na Equação (18.23) e o resultado simplificado, para produzir

$$h_{i-1}c_{i-1} + 2(h_{i-1} - h_i)c_i + h_ic_{i+1} = 3\frac{f_{i+1} - f_i}{h_i} - 3\frac{f_i - f_{i-1}}{h_{i-1}} \quad (18.24)$$

Essa equação pode ser escrita de forma mais concisa, reconhecendo que os termos no lado direito são diferenças divididas finitas [lembre-se da Equação (17.15)]:

$$f[x_i, x_j] = \frac{f_i - f_j}{x_i - x_j}$$

Portanto, a Equação (18.24) pode ser reescrita como

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_ic_{i+1} = 3(f[x_{i+1}, x_i] - f[x_i, x_{i-1}]) \quad (18.25)$$

A Equação (18.25) pode ser escrita para os nós interiores, $i = 2, 3, \dots, n-2$, que resulta em $n-3$ equações tridiagonais simultâneas com $n-1$ coeficientes desconhecidos, c_1, c_2, \dots, c_{n-1} . Portanto, se tivermos duas condições adicionais, podemos calcular os c 's. Uma vez feito isso, as Equações (18.21) e (18.18) podem ser usadas para determinar os coeficientes restantes, b e d .

Como mencionado anteriormente, as duas condições adicionais nas extremidades podem ser formuladas de várias maneiras. Uma abordagem comum, o spline natural, assume que as segundas derivadas nos nós das extremidades são iguais a zero. Para ver como essas condições podem ser integradas no esquema de solução, a segunda derivada no primeiro nó [Equação (18.16)] pode ser definida como igual a zero como em

$$s''_1(x_1) = 0 = 2c_1 + 6d_1(x_1 - x_1)$$

Assim, essa condição equivale a definir c_1 igual a zero.

A mesma avaliação pode ser feita no último nó:

$$s''_{n-1}(x_n) = 0 = 2c_{n-1} + 6d_{n-1}h_{n-1} \quad (18.26)$$

Lembrando a Equação (18.17), podemos convenientemente definir um parâmetro externo c_n , caso em que a Equação (18.26) torna-se

$$c_{n-1} + 3d_{n-1}h_{n-1} = c_n = 0$$

Dessa forma, para impor uma segunda derivada nula no último nó, define-se $c_n = 0$.

A equação final pode agora ser escrita na forma matricial como

$$\begin{bmatrix} 1 & & & \\ h_1 & 2(h_1 + h_2) & h_2 & \\ & \ddots & \ddots & \ddots \\ & & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ & & & & 1 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{Bmatrix} = \begin{Bmatrix} 0 \\ 3(f[x_3, x_2] - f[x_2, x_1]) \\ \vdots \\ 3(f[x_n, x_{n-1}] - f[x_{n-1}, x_{n-2}]) \\ 0 \end{Bmatrix} \quad (18.27)$$

Como mostrado, o sistema é tridiagonal e, portanto, eficiente para ser resolvido.

EXEMPLO 18.3 Splines cúbicos naturais

Ajuste splines cúbicos aos mesmos dados usados nos Exemplos 18.1 e 18.2 (Tabela 18.1). Utilize os resultados para fazer uma estimativa do valor em $x = 5$.

Solução. O primeiro passo é empregar a Equação (18.27) para gerar um conjunto de equações simultâneas que serão utilizadas para determinar os coeficientes c :

$$\begin{bmatrix} 1 & & & \\ h_1 & 2(h_1 + h_2) & h_2 & \\ & h_2 & 2(h_2 + h_3) & h_3 \\ & & & 1 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 3(f[x_3, x_2] - f[x_2, x_1]) \\ 3(f[x_4, x_3] - f[x_3, x_2]) \\ 0 \end{Bmatrix}$$

Os valores necessários da função e as larguras dos intervalos são:

$$\begin{array}{ll} f_1 = 2,5 & h_1 = 4,5 - 3,0 = 1,5 \\ f_2 = 1,0 & h_2 = 7,0 - 4,5 = 2,5 \\ f_3 = 2,5 & h_3 = 9,0 - 7,0 = 2,0 \\ f_4 = 0,5 & \end{array}$$

Esses valores podem ser substituídos para produzir

$$\begin{bmatrix} 1 & & & \\ 1,5 & 8 & 2,5 & 2 \\ & 2,5 & 9 & 1 \\ & & & 1 \end{bmatrix} \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 4,8 \\ -4,8 \\ 0 \end{Bmatrix}$$

Essas equações podem ser resolvidas utilizando o MATLAB com os resultados:

$$\begin{array}{ll} c_1 = 0 & c_2 = 0,839543726 \\ c_3 = -0,766539924 & c_4 = 0 \end{array}$$

As Equações (18.21) e (18.18) podem ser usadas para calcular os b 's e d 's

$$\begin{array}{ll} b_1 = -1,419771863 & d_1 = 0,186565272 \\ b_2 = -0,160456274 & d_2 = -0,214144487 \\ b_3 = 0,022053232 & d_3 = 0,127756654 \end{array}$$

Esses resultados, juntamente com os valores para os a 's [Equação (18.11)], podem ser substituídos na Equação (18.10) para desenvolver os seguintes splines cúbicos para cada intervalo:

$$\begin{aligned} s_1(x) &= 2,5 - 1,419771863(x - 3) + 0,186565272(x - 3)^3 \\ s_2(x) &= 1,0 - 0,160456274(x - 4,5) + 0,839543726(x - 4,5)^2 \\ &\quad - 0,214144487(x - 4,5)^3 \\ s_3(x) &= 2,5 + 0,022053232(x - 7,0) - 0,766539924(x - 7,0)^2 \\ &\quad + 0,127756654(x - 7,0)^3 \end{aligned}$$

Essas três equações podem ser empregadas para calcular valores dentro de cada intervalo. Por exemplo, o valor em $x = 5$, que está contido no segundo intervalo, é calculado como

$$\begin{aligned} s_2(5) &= 1,0 - 0,160456274(5 - 4,5) + 0,839543726(5 - 4,5)^2 - 0,214144487(5 - 4,5)^3 \\ &= 1,102889734. \end{aligned}$$

O ajuste completo do spline cúbico é apresentado na Figura 18.4c.

Os resultados dos Exemplos 18.1 a 18.3 estão resumidos na Figura 18.4. Observe a melhora progressiva no ajuste à medida que nos movemos dos splines lineares para os quadráticos, e destes para os cúbicos. Foi superposto também um polinômio interpolador cúbico na Figura 18.4c. Embora o spline cúbico consista em uma série de curvas de terceiro grau, o ajuste resultante é diferente do obtido usando-se um polinômio de terceiro grau. Isso se deve ao fato de que o spline natural exige segundas derivadas nulas nos nós extremos, enquanto o polinômio cúbico não tem tal restrição.

18.4.2 Condições nas extremidades

Embora sua base gráfica seja atraente, o spline natural é apenas um dos vários tipos, dependendo da condição especificada nas extremidades do intervalo. Outros dois tipos de condições nas extremidades populares são:

- ✓ *Condição de Extremidade Amarrada (Clamped).* Essa opção envolve a especificação das primeiras derivadas no primeiro e último nós, o que às vezes é chamado de spline “amarrado” (ou *clamped*), porque é o que ocorre quando se amarra a extremidade de um spline utilizado para desenhar (lembre-se da técnica de desenho ilustrada na Figura 18.2) de modo que ele tenha uma inclinação desejada. Por exemplo, se são especificadas primeiras derivadas nulas, o spline aplinará (achatará) ou torna-se horizontal nas extremidades.
- ✓ *Condição de Extremidade Sem um Nô (Not-a-Knot).* Uma terceira alternativa é forçar a continuidade da derivada terceira no segundo e penúltimo nós. Uma vez que o spline já especifica que o valor da função e suas derivadas primeira e segunda são iguais nesses nós, a especificação de derivadas terceiras contínuas significa que as mesmas funções cúbicas se aplicarão a cada um dos primeiros e

últimos dois segmentos adjacentes. E considerando que os primeiros nós internos não representam mais a junção de duas funções cúbicas diferentes, eles não são mais nós verdadeiros. Portanto, esse caso é chamado de *condição de extremidade sem um nó*. Ela tem a propriedade adicional de, para quatro pontos, produzir o mesmo resultado que é obtido por meio de um polinômio interpolador cúbico comum do tipo descrito no Capítulo 17.

Essas condições podem ser imediatamente aplicadas usando a Equação (18.25) para os nós interiores, $i = 2, 3, \dots, n - 2$, e usando a primeira (1) e última ($n - 1$) equações, como apresenta a Tabela 18.2.

A Figura 18.5 mostra uma comparação das três condições na extremidade como aplicadas para ajustar os dados da Tabela 18.1. O caso amarrado é definido de modo que as derivadas nas extremidades sejam iguais a zero.

Como esperado, o ajuste por spline para o caso amarrado se aplana nas extremidades. Já os casos natural e sem um nó seguem a tendência dos pontos dados mais rigorosamente. Observe como o spline natural tende a endireitar, como previsto, porque as segundas derivadas vão para zero nas extremidades. Como a condição sem um nó tem segundas derivadas não nulas nas extremidades, exibe maior curvatura.

TABELA 18.2 A primeira e última equações necessárias para especificar algumas condições de extremidade frequentemente utilizadas para splines cúbicos.

Condição	Primeira e Última equações
Natural	$c_1 = 0, c_n = 0$
Amarraada (onde f'_1 e f'_{n-1} são as primeiras derivadas especificadas no primeiro e último nós, respectivamente).	$2h_1c_1 + h_1c_2 = 3f[x_2, x_1] - 3f'_1$ $h_{n-1}c_{n-1} + 2h_{n-1}c_n = 3f'_{n-1} - 3f[x_n, x_{n-1}]$
Sem um nó	$h_2c_1 - (h_1 + h_2)c_2 + h_1c_3 = 0$ $h_{n-2}c_{n-2} - (h_{n-2} + h_{n-1})c_{n-1} + h_{n-2}c_n = 0$

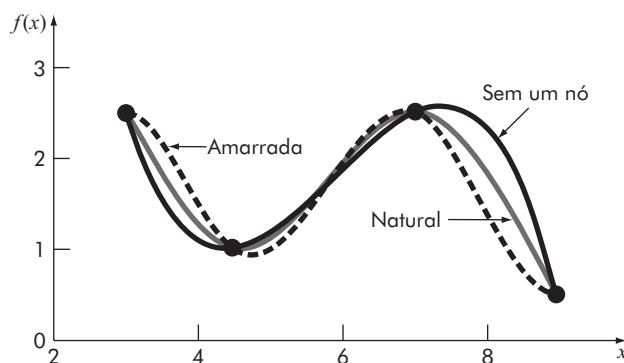


FIGURA 18.5 Comparação dos splines amarrado (com primeiras derivadas nulas), sem um nó e natural para os dados da Tabela 18.1.

18.5 INTERPOLAÇÃO POR PARTES NO MATLAB

O MATLAB possui varias funções nativas para implementar interpolação por partes. A função `spline` realiza a interpolação por splines cúbicos, conforme descrito neste capítulo. A função `pchip` implementa a interpolação cúbica hermitiana por partes. A função `interp1` também pode implementar as interpolações hermitiana e por splines, porém também pode executar uma série de outros tipos de interpolação por partes.

18.5.1 Função nativa do MATLAB: `spline`

Splines cúbicos podem ser facilmente calculados com a função nativa `spline` do MATLAB. Ela tem a seguinte sintaxe geral,

$$yy = \text{spline}(x, y, xx) \quad (18.28)$$

onde `x` e `y` são vetores contendo os valores dos pontos dados a serem utilizados na interpolação e `yy` é um vetor contendo os resultados da interpolação por splines avaliados nos pontos do vetor `xx`.

A função `spline`, por padrão (default), utiliza a condição sem um nó. Porém, se `y` contém dois valores a mais que o vetor `x`, então o primeiro e último valores em `y` são usados como derivadas nos pontos extremos. Consequentemente, essa opção fornece os meios para implementar a condição de extremidade amarrada.

EXEMPLO 18.4 Splines no MATLAB

A função de Runge é um exemplo notório de uma função que não pode ser bem ajustada com polinômios (lembre-se do Exemplo 17.7):

$$f(x) = \frac{1}{1 + 25x^2}$$

Use o MATLAB para ajustar nove pontos dados igualmente espaçados e amostrados dessa função no intervalo $[-1, 1]$. Empregue (a) um spline sem um nó e (b) um spline amarrado com inclinações nas extremidades de $f'_1 = 1$ e $f'_{n-1} = -4$.

Solução. (a) Nove pontos igualmente espaçados podem ser gerados, como em

```
>> x = linspace(-1,1,9);
>> y = 1./(1+25*x.^2);
```

Em seguida, um vetor de valores mais finamente espaçados pode ser gerado de modo que possamos criar um gráfico suave dos resultados como gerados com a função `spline`:

```
>> xx = linspace(-1,1);
>> yy = spline(x,y,xx);
```

Lembre-se de que a função `linspace` cria automaticamente 100 pontos, se o número desejado de pontos não é especificado. Por fim, podemos gerar valores para a própria função de Runge e exibi-los com o ajuste por splines e os dados originais:

```
>> yr = 1./(1+25*xx.^2);
>> plot(x,y,'o',xx,yy,xx,yr,'--')
```

Como ilustra a Figura 18.6, o spline sem um nó segue bem a função de Runge sem exibir oscilações pronunciadas entre os pontos.

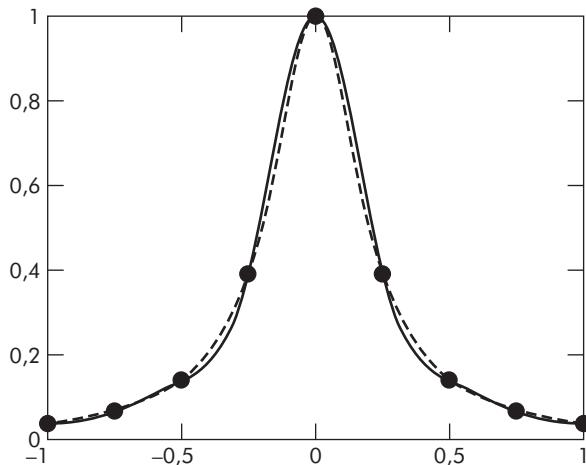


FIGURA 18.6 Comparação da função de Runge (linha tracejada) com um ajuste de 9 pontos por um spline sem um nó gerado com o MATLAB (linha sólida).

(b) A condição amarrada pode ser implementada com a criação de um novo vetor yc , que tem as primeiras derivadas desejadas como seus primeiro e último elementos. Ele pode então ser usado para gerar e traçar o ajuste por spline:

```
>> yc=[1 y -4];
>> yyc=spline(x,yc,xx);
>> plot(x,y,'o',xx,yyc,xx,yr,'--')
```

Como mostra a Figura 18.7, o spline amarrado exibe agora algumas oscilações decorrentes das inclinações artificiais que foram impostas nas fronteiras. Nos casos em que tivermos conhecimento das primeiras derivadas verdadeiras, o spline amarrado tende a melhorar o ajuste.

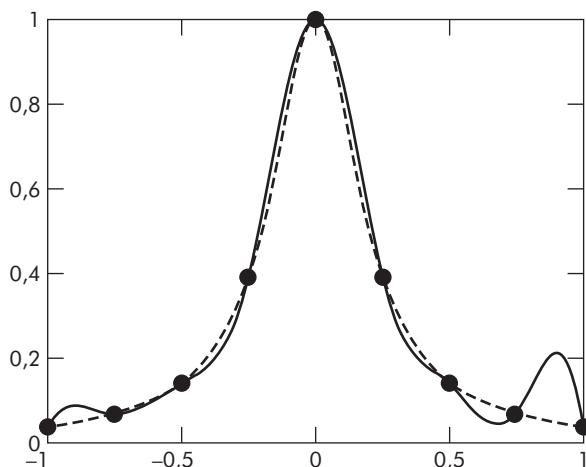


FIGURA 18.7 Comparação da função de Runge (linha tracejada) com um ajuste de 9 pontos por um spline amarrado gerado com o MATLAB (linha sólida). Observe que primeiras derivadas de 1 e -4 são especificadas nas fronteiras à esquerda e direita do intervalo, respectivamente.

18.5.2 Função Nativa do MATLAB: `interp1`

A função nativa `interp1` fornece um meio prático para implementar uma série de tipos diferentes de interpolação unidimensional por partes. Ela tem a sintaxe geral

```
yi = interp1(x, y, xi, 'metodo')
```

onde `x` e `y` são vetores que contêm os valores dos pontos dados a serem utilizados na interpolação, `yi` é um vetor que contém os resultados da interpolação avaliados nos pontos do vetor `xi` e '`metodo`' é o método desejado. Os vários métodos são:

- ✓ 'nearest' – interpolação por vizinho mais próximo, que define o valor de um ponto interpolado igual ao valor do ponto dado existente mais próximo. Assim, a interpolação se parece com uma série de platôs, que podem ser pensados como polinômios de grau zero.
- ✓ 'linear' – interpolação linear, que utiliza retas para ligar os pontos.
- ✓ 'spline' – interpolação por splines cúbicos por partes, que fornece o mesmo resultado que a função `spline`.
- ✓ 'pchip' e 'cubic' – interpolação cúbica hermitiana por partes.

Se o argumento '`metodo`' é omitido, o padrão é a interpolação linear.

A opção `pchip` (abreviação de “*piecewise cubic Hermite interpolation*” ou “*interpolação cúbica hermitiana por partes*”) merece mais discussão. Tal como com os splines cúbicos, a função `pchip` usa polinômios cúbicos para ligar os pontos dados com primeiras derivadas contínuas. Entretanto, ela difere dos splines cúbicos uma vez que as segundas derivadas não são necessariamente contínuas. Além disso, as primeiras derivadas nos nós não serão as mesmas que para os splines cúbicos. Em vez disso, elas são expressamente escolhidas de modo que se tenha uma interpolação com “*preservação de forma*” (*shape preserving*); ou seja, os valores interpolados não tendem a ultrapassar os pontos dados, como às vezes acontece com os splines cúbicos.

Portanto, existem contrapartidas entre as opções `spline` e `pchip`. Os resultados advindos da utilização da opção `spline` geralmente parecerão mais suaves, porque o olho humano pode detectar descontinuidades na segunda derivada. Além disso, essa opção será mais acurada se os dados são valores de uma função suave. No entanto, a opção `pchip` tem menos oscilações e não apresenta ultrapassagens se os dados não são suaves. Essas contrapartidas, bem como aquelas envolvendo as outras opções, são exploradas no exemplo a seguir.

EXEMPLO 18.5 Contrapartidas da utilização da função `interp1`

Você realiza um *test drive* em um automóvel em que você alternadamente acelera o automóvel e depois o mantém em uma velocidade constante. Observe que você nunca desacelera durante o experimento. A série temporal de medições pontuais de velocidade pode ser tabulada como

<i>t</i>	0	20	40	56	68	80	84	96	104	110
<i>v</i>	0	20	20	38	80	80	100	100	125	125

Use a função `interp1` do MATLAB para ajustar esses dados com (a) interpolação linear, (b) interpolação por vizinho mais próximo, (c) spline cúbico com condições de extremidade sem um nó, e (d) interpolação cúbica hermitiana por partes.

Solução. (a) Pode-se entrar com os dados, ajustá-los com interpolação linear e traçá-los com os seguintes comandos:

```
>> t = [0 20 40 56 68 80 84 96 104 110];
>> v = [0 20 20 38 80 80 100 100 125 125];
>> tt = linspace(0,110);
>> vl = interp1(t,v,tt);
>> plot(t,v,'o',tt,vl)
```

Os resultados (Figura 18.8a) não são suaves, porém não exibem nenhuma ultrapassagem.

(b) Os comandos para implementar e traçar a interpolação por vizinho mais próximo são

```
>> vn = interp1(t,v,tt,'nearest');
>> plot(t,v,'o',tt,vn)
```

Como na Figura 18.8b, os resultados parecem uma série de platôs. Essa opção não é nem uma descrição suave nem acurada do processo subjacente.

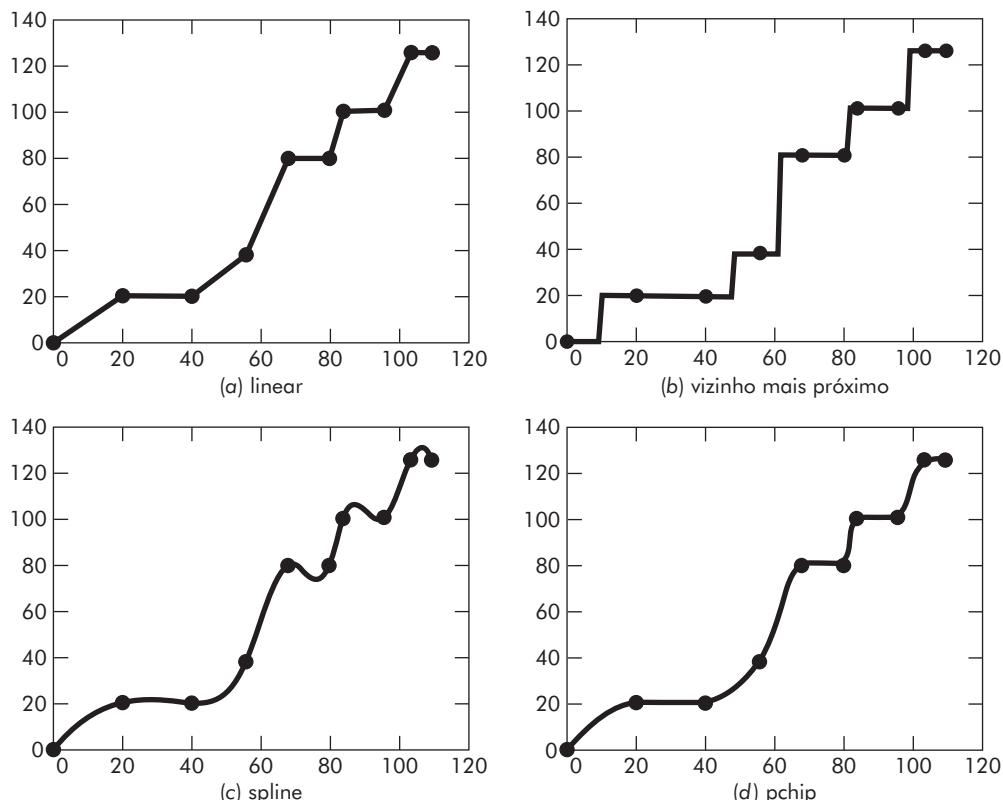


FIGURA 18.8 Uso de várias opções da função `interp1` para realizar interpolação polinomial por partes em uma série temporal de velocidades para um automóvel.

(c) Os comandos para implementar os splines cúbicos são

```
>> vs = interp1(t,v,tt,'spline');
>> plot(t,v,'o',tt,vs)
```

O resultado obtido (Figura 18.8c) é bastante suave; no entanto, ultrapassagens importantes ocorrem em vários locais, o que demonstra que o automóvel desacelerou várias vezes durante o experimento.

(d) Os comandos para implementar a interpolação cúbica hermitiana são

```
>> vh = interp1(t,v,tt,'pchip');
>> plot(t,v,'o',tt,vh)
```

Nesse caso, os resultados (Figura 18.8d) são fisicamente consistentes. Devido à sua natureza de preservação da forma, a velocidade aumenta monotonicamente e nunca exibe desaceleração; e embora o resultado não seja tão suave como para os splines cúbicos, a continuidade das primeiras derivadas nos nós torna as transições entre os pontos mais graduais e, portanto, mais realistas.

18.6 INTERPOLAÇÃO MULTIDIMENSIONAL

Os métodos de interpolação para problemas unidimensionais podem ser estendidos para interpolação multidimensional. Nesta seção, descreveremos o caso mais simples de interpolação bidimensional em coordenadas cartesianas. Além disso, descreveremos as capacidades do MATLAB para interpolação multidimensional.

18.6.1 Interpolação bilinear

A *interpolação bidimensional* trata da determinação de valores intermediários para funções de duas variáveis $z = f(x_i, y_i)$. Como mostra a Figura 18.9, têm-se valores

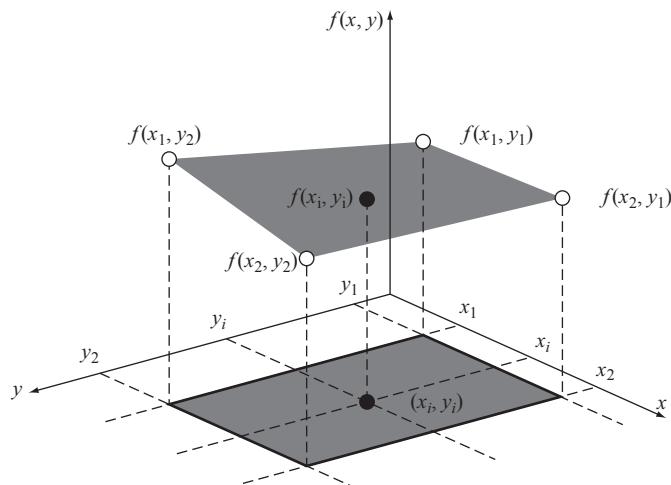


FIGURA 18.9 Descrição gráfica da interpolação bilinear em que um valor intermediário (círculo preenchido) é estimado com base em quatro valores dados (círculos abertos).

em quatro pontos: $f(x_1, y_1)$, $f(x_2, y_1)$, $f(x_1, y_2)$ e $f(x_2, y_2)$. Deseja-se interpolar entre esses pontos para estimar o valor em um ponto intermediário $f(x_i, y_i)$. Se usarmos uma função linear, o resultado será um plano conectando os pontos, como na Figura 18.9. Essas funções são chamadas *bilineares*.

Uma abordagem simples para desenvolver a função bilinear é descrita na Figura 18.10. Pode-se manter o valor de y fixo e aplicar uma interpolação linear unidimensional na direção x . Utilizando a forma de Lagrange, o resultado em (x_i, y_i) é

$$f(x_i, y_1) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_1) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_1) \quad (18.29)$$

e em (x_i, y_2) é

$$f(x_i, y_2) = \frac{x_i - x_2}{x_1 - x_2} f(x_1, y_2) + \frac{x_i - x_1}{x_2 - x_1} f(x_2, y_2) \quad (18.30)$$

Esses pontos podem então ser usados para interpolar linearmente ao longo da dimensão y , para produzir o resultado final:

$$f(x_i, y_i) = \frac{y_i - y_2}{y_1 - y_2} f(x_i, y_1) + \frac{y_i - y_1}{y_2 - y_1} f(x_i, y_2) \quad (18.31)$$

Uma única equação pode ser obtida substituindo as Equações (18.29) e (18.30) na Equação (18.31), para fornecer

$$\begin{aligned} f(x_i, y_i) &= \frac{x_i - x_2}{x_1 - x_2} \frac{y_i - y_2}{y_1 - y_2} f(x_1, y_1) + \frac{x_i - x_1}{x_2 - x_1} \frac{y_i - y_2}{y_1 - y_2} f(x_2, y_1) \\ &\quad + \frac{x_i - x_2}{x_1 - x_2} \frac{y_i - y_1}{y_2 - y_1} f(x_1, y_2) + \frac{x_i - x_1}{x_2 - x_1} \frac{y_i - y_1}{y_2 - y_1} f(x_2, y_2) \end{aligned} \quad (18.32)$$

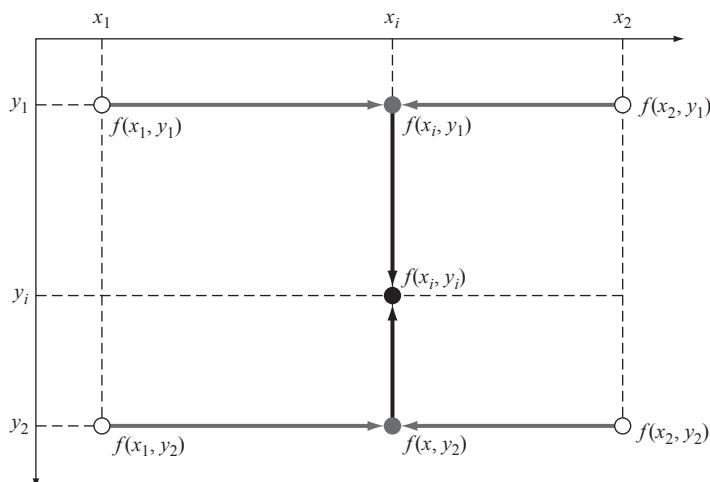


FIGURA 18.10 A interpolação bilinear bidimensional pode ser implementada aplicando a interpolação linear unidimensional ao longo da dimensão x para determinar valores de x_i . Esses valores podem então ser usados para interpolar linearmente ao longo da dimensão y , para produzir o resultado final em x_i, y_i .

EXEMPLO 18.6 Interpolação bilinear

Considere que você mediu as temperaturas em uma série de coordenadas na superfície de uma placa retangular aquecida:

$$T(2, 1) = 60 \quad T(9, 1) = 57,5$$

$$T(2, 6) = 55 \quad T(9, 6) = 70$$

Use interpolação bilinear para estimar a temperatura em $x_i = 5,25$ e $y_i = 4,8$.

Solução. Substituindo esses valores na Equação (18.32), tem-se:

$$\begin{aligned} f(5,25, 4,8) &= \frac{5,25 - 9}{2 - 9} \frac{4,8 - 6}{1 - 6} 60 + \frac{5,25 - 2}{9 - 2} \frac{4,8 - 6}{1 - 6} 57,5 \\ &+ \frac{5,25 - 9}{2 - 9} \frac{4,8 - 1}{6 - 1} 55 + \frac{5,25 - 2}{9 - 2} \frac{4,8 - 1}{6 - 1} 70 = 61,2143 \end{aligned}$$

18.6.2 Interpolação multidimensional no MATLAB

O MATLAB tem duas funções nativas para interpolação por partes bi e tridimensional: `interp2` e `interp3`. Como pode ser observado a partir de seus nomes, essas funções operam de maneira similar à `interp1` (Seção 18.5.2). Por exemplo, uma representação simples da sintaxe de `interp2` é

```
zi = interp2(x,y,z,xi,yi,'metodo')
```

onde x e y são matrizes que contêm as coordenadas dos pontos em que os valores na matriz z são dados, zi é uma matriz que contém os resultados da interpolação avaliada nos pontos nas matrizes xi e yi e ' $metodo$ ' é o método desejado. Observe que os métodos são idênticos àqueles usados por `interp1`; isto é, `linear`, `nearest`, `spline` e `cubic`.

Tal como com a função `interp1`, se o argumento ' $metodo$ ' é omitido, o padrão (default) é interpolação linear. Por exemplo, a função `interp2` pode ser usada para fazer a mesma avaliação, como no Exemplo 18.6:

```
>> x=[2 9];
>> y=[1 6];
>> z=[60 57.5;55 70];
>> interp2(x,y,z,5.25,4.8)
ans =
    61.2143
```

► 18.7 ESTUDO DE CASO TRANSFERÊNCIA DE CALOR

Contextualização. Os lagos na zona temperada podem tornar-se termicamente estratificados durante o verão. Como mostra a Figura 18.11, a água quente flutuante perto da superfície está sobre a água do fundo, mais fria e mais densa. Essa estratificação divide verticalmente, de forma efetiva, o lago em duas camadas: o *epilímnio* e o *hipolímnio*, separadas por um plano chamado *termoclinia*.

A estratificação térmica tem grande importância para os engenheiros ambientais e para os cientistas que estudam esses sistemas. Em particular, as termoclinas diminuem muito a mistura entre as duas camadas. Como resultado, a decomposição de matéria orgânica pode levar a uma diminuição grave de oxigênio na camada isolada do fundo.

A posição da termoclinia pode ser definida como o ponto de inflexão da curva temperatura-profundidade – isto é, o ponto no qual $d^2T/dz^2 = 0$. Ele também é o ponto no qual o valor absoluto da primeira derivada ou do gradiente é um máximo.

O gradiente de temperatura é importante por si só porque ele pode ser usado em conjunto com a lei de Fourier para determinar o fluxo de calor através da termoclinia:

$$J = -D\rho C \frac{dT}{dz} \quad (18.33)$$

onde J é o fluxo de calor [$\text{cal}/(\text{cm}^2 \cdot \text{s})$], α é o coeficiente de difusão turbulenta (cm^2/s), ρ é a densidade ($\approx 1 \text{ g/cm}^3$) e C é o calor específico [$\approx 1 \text{ cal}/(\text{g} \cdot \text{C})$].

Neste estudo de caso, splines cúbicos naturais são empregados para determinar a profundidade da termoclinia e o gradiente de temperatura para o lago Platte, em Michigan (Tabela 18.3). Este último também é usado para determinar o fluxo de calor para o caso em que $\alpha = 0,01 \text{ cm}^2/\text{s}$.

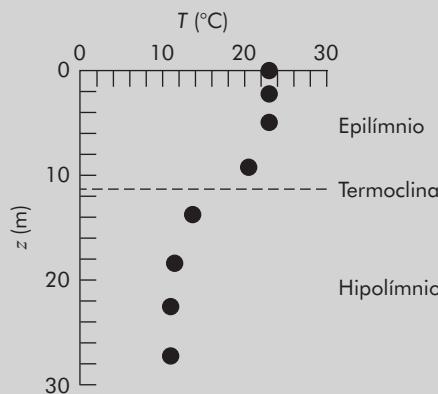


FIGURA 18.11 Temperatura versus profundidade durante o verão para o lago Platte, em Michigan.

TABELA 18.3 Temperatura versus profundidade durante o verão para o lago Platte, em Michigan.

z, m	0	2,3	4,9	9,1	13,7	18,3	22,9	27,2
$T, ^\circ\text{C}$	22,8	22,8	22,8	20,6	13,9	11,7	11,1	11,1

Solução. Como descrito anteriormente, desejamos utilizar splines naturais para realizar essa análise. Infelizmente, como a função nativa `spline` do MATLAB utiliza condições de extremidade sem um nó (not-a-knot), ela não atende às nossas necessidades. Além disso, a função `spline` não retorna as primeiras e segundas derivadas necessárias para nossas análises.

No entanto, não é difícil desenvolver nossa própria função do MATLAB para implementar um spline natural e retornar as derivadas. Esse código é apresentado na Figura 18.12. Após algumas linhas destinadas à detecção de erros na entrada de dados, a Equação (18.27) é montada e resolvida para determinar os coeficientes de segunda ordem (c). Observe que são utilizadas duas subfunções, h e fd , para calcular as diferenças finitas necessárias. Uma vez montada a Equação (18.27), ela é resolvida empregando divisão à esquerda. Um laço é então empregado para gerar os outros coeficientes (a , b e d).

```
function [yy,dy,d2] = natspline(x,y,xx)
% natspline: spline natural com diferenciação
% [yy,dy,d2] = natspline(x,y,xx): usa uma interpolação por
spline
% cúbico natural para encontrar yy, os valores da função
subjacente
% y nos pontos contidos no vetor xx. O vetor x especifica os
Pontos
% nos quais os valores de y são dados.
% Entrada:
% x = vetor de variáveis independentes
% y = vetor de variáveis dependentes
% xx = vetor de valores desejados da variável dependente
% Saída:
% yy = valores interpolados em xx
% dy = primeiras derivadas em xx
% d2 = segundas derivadas em xx
n = length(x);
if length(y) ~= n, error('x e y devem ter o mesmo tamanho');
end
if any(diff(x) <= 0), error('x não estritamente ascendente'), end
m = length(xx);
b = zeros(n,n);
aa(1,1) = 1; aa(n,n) = 1; %monta a Equação (18.27)
bb(1)=0; bb(n)=0;
for i = 2:n-1
    aa(i,i-1) = h(x, i - 1);
    aa(i,i) = 2 * (h(x, i - 1) + h(x, i));
    aa(i,i+1) = h(x, i);
    bb(i) = fd(x, i);
end
yy = b\bb;
dy = fd(x, 1:m-1);
d2 = fd(x, 1:m-2);
```

FIGURA 18.12 (continua)



```

bb(i) = 3 * (fd(i + 1, i, x, y) - fd(i, i - 1, x, y));
end
c=aa\bb'; %calcula os coeficientes c
for i = 1:n - 1 %calcula os coeficientes a, b e d
    a(i) = y(i);
    b(i) = fd(i + 1, i, x, y) - h(x, i) / 3 * (2 * c(i) + c(i + 1));
    d(i) = (c(i + 1) - c(i)) / 3 / h(x, i);
end
for i = 1:m %realiza interpolações nos valores desejados
    [yy(i),dy(i),d2(i)] = SplineInterp(x, n, a, b, c, d, xx(i));
end
end
function hh = h(x, i)
hh = x(i + 1) - x(i);
end
function fdd = fd(i, j, x, y)
fdd = (y(i) - y(j)) / (x(i) - x(j));
end
function [yyy,dyy,d2y] = SplineInterp(x, n, a, b, c, d, xi)
for ii = 1:n - 1
    if xi >= x(ii) - 0.000001 & xi <= x(ii + 1) + 0.000001
        yyy=a(ii)+b(ii)*(xi-x(ii))+c(ii)*(xi-x(ii))^2+d(ii)...
            *(xi-x(ii))^3;
        dyy=b(ii)+2*c(ii)*(xi-x(ii))+3*d(ii)*(xi-x(ii))^2;
        d2y=2*c(ii)+6*d(ii)*(xi-x(ii));
        break
    end
end
end

```

FIGURA 18.12 Função do MATLAB para determinar valores intermediários e derivadas com um spline natural. Observe que a função diff, empregada para detecção de erro, é descrita na Seção 21.7.1. (continuação)

Neste ponto, temos tudo que precisamos para gerar valores intermediários com a equação cúbica:

$$f(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Também podemos determinar as primeiras e segundas derivadas diferenciando essa equação duas vezes, para fornecer

$$\begin{aligned} f'(x) &= b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2 \\ f''(x) &= 2c_i + 6d_i(x - x_i) \end{aligned}$$

Como mostra a Figura 18.12, essas equações podem então ser implementadas em outra subfunção, SplineInterp, para determinar os valores e as derivadas nos valores intermediários desejados.

A seguir tem-se um script que usa a função natspline para gerar o spline e criar gráficos dos resultados:



```

z=[0 2.3 4.9 9.1 13.7 18.3 22.9 27.2];
T=[22.8 22.8 22.8 20.6 13.9 11.7 11.1 11.1];
zz=linspace(z(1),z(length(z)));
[TT,dT,dT2]=natspline(z,T,zz);
subplot(1,3,1),plot(T,z,'o',TT,zz)
title('(a) T'),legend('dados','T')
set(gca,'YDir','reverse'),grid
subplot(1,3,2),plot(dT,zz)
title('(b) dT/dz')
set(gca,'YDir','reverse'),grid
subplot(1,3,3),plot(dT2,zz)
title('(c) d2T/dz2')
set(gca,'YDir','reverse'),grid

```

Como mostra a Figura 18.13, a termoclinia parece estar localizada em uma profundidade de cerca de 11,5 m. Podemos utilizar métodos de localização de raízes (segunda derivada igual a zero) ou de otimização (primeira derivada mínima) para refinar essa estimativa. O resultado é que a termoclinia está localizada em 11,35 m, onde o gradiente é $-1,61^{\circ}\text{C}/\text{m}$.

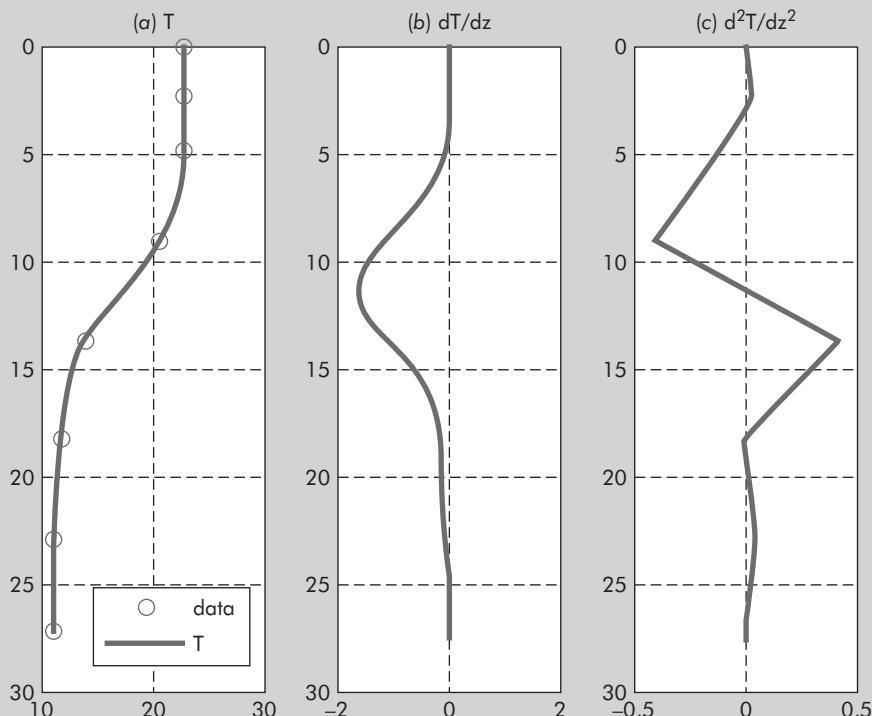


FIGURA 18.13 Gráficos de (a) temperatura, (b) gradiente e (c) segunda derivada versus profundidade (m) gerados com o programa de splines cúbicos naturais. A termoclinia está localizada no ponto de inflexão da curva temperatura-profundidade.

O gradiente pode ser usado para calcular o fluxo de calor através da termoclinia com a Equação (18.33):

$$J = -0,01 \frac{\text{cm}^2}{\text{s}} \times 1 \frac{\text{g}}{\text{cm}^3} \times 1 \frac{\text{cal}}{\text{g} \cdot ^\circ\text{C}} \times \left(-1,61 \frac{{}^\circ\text{C}}{\text{m}} \right) \times \frac{1 \text{ m}}{100 \text{ cm}} \times \frac{86.400 \text{ s}}{\text{d}} = 13,9 \frac{\text{cal}}{\text{cm}^2 \cdot \text{d}}$$

A análise anterior demonstra como a interpolação por splines pode ser usada para solução de problemas de engenharia e de ciência; porém, é também um exemplo de derivação numérica. Como tal, ela ilustra como abordagens numéricas de diferentes áreas podem ser usadas em conjunto para resolução de problemas. Descreveremos o tópico derivação numérica em detalhes no Capítulo 21.

►► PROBLEMAS

18.1 Considere os dados

x	1	2	2,5	3	4	5
$f(x)$	1	5	7	8	2	1

Ajuste-os com (a) um spline cúbico com condições de extremidade naturais, (b) um spline cúbico com condições de extremidade sem um nó, e (c) interpolação cúbica hermitiana por partes.

18.2 Um reator está termicamente estratificado como na seguinte tabela:

Profundidade, m	0	0,5	1	1,5	2	2,5	3
Temperatura, °C	70	70	55	22	13	10	10

Com base nessas temperaturas, o tanque pode ser idealizado como duas zonas separadas por um forte gradiente de temperatura ou uma *termoclinia*. A profundidade da termoclinia pode ser definida como o ponto de inflexão da curva temperatura-profundidade – isto é, o ponto no qual $d^2T/dz^2 = 0$. Nessa profundidade, o fluxo de calor da superfície para a camada do fundo pode ser calculado com a lei de Fourier:

$$J = -k \frac{dT}{dz}$$

Use um ajuste por spline cúbico amarrado com derivadas nulas nas extremidades para determinar a profundidade da termoclinia. Se $k = 0,01 \text{ cal}/(\text{s} \cdot \text{cm} \cdot {}^\circ\text{C})$, calcule o fluxo através dessa interface.

18.3 A equação a seguir está implementada na função nativa `humps`, que é utilizada pelo MATLAB para demonstrar algumas de suas capacidades numéricas:

$$f(x) = \frac{1}{(x - 0,3)^2 + 0,01} + \frac{1}{(x - 0,9)^2 + 0,04} - 6$$

A função `humps` exibe tanto regiões planas quanto íngremes em um intervalo relativamente curto de x . A tabela a seguir apresenta alguns valores que foram gerados em intervalos de 0,1 ao longo da faixa de $x = 0$ a 1:

x	0	0,1	0,2	0,3	0,4	0,5
$f(x)$	5,176	15,471	45,887	96,500	47,448	19,000
x	0,6	0,7	0,8	0,9	1	
$f(x)$	11,692	12,382	17,846	21,703	16,000	

Ajuste esses dados com (a) splines cúbicos com condições de extremidade sem um nó e (b) interpolação cúbica hermitiana por partes. Em ambos os casos, crie um gráfico comparando o ajuste com a função `humps` exata.

18.4 Desenvolva um gráfico de um ajuste por splines cúbicos dos dados a seguir com condições de extremidade (a) naturais e (b) sem um nó desenvolva também um gráfico utilizando (c) interpolação cúbica hermitiana por partes.

x	0	100	200	400
$f(x)$	0	0,82436	1,00000	0,73576
x	600	800	1000	
$f(x)$	0,40601	0,19915	0,09158	

Em cada caso, compare seu gráfico com a seguinte equação que foi usada para gerar os dados:

$$f(x) = \frac{x}{200} e^{-x/200 + 1}$$

18.5 Os dados a seguir foram amostrados da função degrau descrita na Figura 18.1

x	-1	-0,6	-0,2	0,2	0,6	1
$f(x)$	0	0	0	1	1	1

Ajuste-os com (a) splines cúbicos com condições de extremidade sem um nó, (b) splines cúbicos com condições de extremidade amarrada e derivadas nulas, e (c) interpolação cúbica hermitiana por partes. Em cada caso, crie um gráfico comparando o ajuste com a função degrau.

18.6 Desenvolva uma função no MATLAB para calcular um ajuste por splines cúbicos com condições de extremidade naturais. Teste seu código utilizando-o para refazer o Exemplo 18.3.

18.7 Os dados a seguir foram gerados com o polinômio de quinto grau: $f(x) = 0,0185x^5 - 0,444x^4 + 3,9125x^3 - 15,456x^2 + 27,069x - 14,1$:

x	1	3	5	6	7	9
$f(x)$	1,000	2,172	4,220	5,430	4,912	9,120

- (a) Ajuste-os com splines cúbicos com condições de extremidade sem um nó. Faça um gráfico comparando o ajuste com a função.
- (b) Repita (a), mas use condições de extremidade amarradas definindo as derivadas nas extremidades com os valores exatos, tal como determinado através da diferenciação da função.

18.8 As funções de Bessel aparecem com freqüência em análises avançadas de engenharia tais como o estudo de campos elétricos. Essas funções geralmente não são passíveis de avaliação simples e, portanto, são frequentemente compiladas em tabelas matemáticas padrões. Por exemplo,

x	1,8	2	2,2	2,4	2,6
$J_1(x)$	0,5815	0,5767	0,556	0,5202	0,4708

Estime $J_1(2,1)$, (a) utilizando um polinômio interpolador e (b) utilizando splines cúbicos. Observe que o valor verdadeiro é 0,5683.

18.9 Os dados a seguir definem a concentração de oxigênio dissolvido na água fresca em função da temperatura no nível do mar:

$T, {}^\circ\text{C}$	0	8	16	24	32	40
$\sigma, \text{mg/L}$	14,621	11,843	9,870	8,418	7,305	6,413

Use o MATLAB para ajustar os dados com (a) interpolação linear por partes, (b) um polinômio de quinto grau, e (c) splines. Mostre os resultados graficamente e use cada abordagem para estimar $\sigma(27)$. Observe que o resultado exato é 7,986 mg/L.

18.10 (a) Use o MATLAB para ajustar os dados a seguir por splines cúbicos e determinar y em $x = 1,5$:

x	0	2	4	7	10	12
y	20	20	12	7	6	6

(b) Repita (a), mas com primeiras derivadas nulas nos nós extremos.

18.11 A função de Runge é escrita como

$$f(x) = \frac{1}{1 + 25x^2}$$

Gere cinco valores igualmente espaçados dessa função ao longo do intervalo $[-1, 1]$. Ajuste esses dados com (a) um polinômio de quarto grau, (b) splines lineares, e (c) splines cúbicos. Apresente seus resultados graficamente.

18.12 Use o MATLAB para gerar oito pontos da função

$$f(t) = \sin^2 t$$

de $t = 0$ a 2π . Ajuste esses dados utilizando (a) splines cúbicos com condições de extremidade sem um nó, (b) splines cúbicos com derivadas nas extremidades iguais aos valores exatos calculados com diferenciação, e (c) interpolação cúbica hermitiana por partes. Desenvolva gráficos de cada ajuste bem como gráficos do erro absoluto (E_t = valor verdadeiro – aproximação) para cada abordagem.

18.13 Sabe-se que o coeficiente de arraste para esferas, tais como bolas esportivas, varia como uma função do número de Reynolds Re , um número adimensional que fornece uma medida da razão entre forças iniciais e forças viscosas:

$$Re = \frac{\rho V D}{\mu}$$

onde ρ é a densidade do fluido (kg/m^3), V é sua velocidade (m/s), D é o diâmetro (m) e μ é a viscosidade dinâmica ($\text{N} \cdot \text{s/m}^2$). Embora a relação entre o arraste e o número de Reynolds esteja às vezes disponível na forma de equação, ele é geralmente tabulado. Por exemplo, a tabela a seguir fornece valores para uma bola esférica lisa:

$Re \times 10^{-4}$	2	5,8	16,8	27,2	29,9	33,9
C_d	0,52	0,52	0,52	0,5	0,49	0,44
$Re \times 10^{-4}$	36,3	40	46	60	100	200
C_d	0,18	0,074	0,067	0,08	0,12	0,16
						0,19

(a) Desenvolva uma função no MATLAB que empregue a função `spline` para retornar um valor de C_d como uma função do número de Reynolds. A primeira linha da função deve ser

```
function CDout = Arraste(ReCD, ReIn)
```

onde `ReCD` é uma matriz de duas linhas que contém a tabela, `ReIn` é o número de Reynolds em que se deseja fazer uma estimativa do arraste e `CDout` é o coeficiente de arraste correspondente.

(b) Escreva um programa que utilize a função desenvolvida na parte (a) para gerar um gráfico (com os eixos devidamente identificados) da força de arraste *versus* a velocidade (lembre-se da Seção 1.4). Use os seguintes valores de parâmetros

para o programa: $D = 22$ cm, $\rho = 1,3$ kg/m³ e $\mu = 1,78 \times 10^{-5}$ Pa · s. Empregue um intervalo de velocidades de 4 a 40 m/s para seu gráfico.

18.14 A função a seguir descreve a distribuição de temperaturas em uma placa retangular para o intervalo $-2 \leq x \leq 0$ e $0 \leq y \leq 3$

$$T = 2 + x - y + 2x^2 + 2xy + y^2$$

Desenvolva um programa para: (a) Gerar um gráfico de malhas dessa função utilizando a função `surf` do MATLAB. Empregue a função `linspace` com espaçamento-padrão (isto é, 100 pontos interiores) para gerar os valores de x e y . (b) Use a função `interp2` do MATLAB com a opção de interpolação-padrão ('linear') para calcular a temperatura em $x = -1,63$ e $y = 1,627$. Determine o erro relativo percentual de seus resultados. (c) Repita (b), mas com 'spline'. Observação: para as partes (b) e (c), empregue a função `linspace` com 9 pontos interiores.

Parte 5

Integração e derivação

5.1 VISÃO GERAL

Nos primeiros anos de faculdade, você foi apresentado ao cálculo integral e diferencial, e aprendeu técnicas para obter derivadas e integrais exatas ou analíticas.

Matematicamente, a *derivada* representa a taxa de variação de uma variável dependente com relação a uma variável independente. Por exemplo, se nos é dada uma função $y(t)$ que especifica a posição de um corpo como uma função do tempo, a derivação fornece os meios para determinar sua velocidade, como em:

$$v(t) = \frac{d}{dt}y(t)$$

Como mostra a Figura PT5.1a, a derivada pode ser visualizada como a inclinação de uma função.

A integração é o processo inverso da derivação. Assim como a derivação usa diferenças para quantificar um processo instantâneo, a integração envolve a soma de informações instantâneas para fornecer um resultado total ao longo de um intervalo. Dessa forma, se nos é fornecida a velocidade como uma função do tempo, a integração pode ser usada para determinar a distância percorrida:

$$y(t) = \int_0^t v(t) dt$$

Como mostra a Figura PT5.1b, para funções situadas acima do eixo das abscissas, a integral pode ser visualizada como a área sob a curva de $v(t)$ de 0 a t . Consequentemente, assim como uma derivada pode ser pensada como uma inclinação, a integral pode ser vista como uma somatória.

Por causa da relação íntima entre derivação e integração, optamos por dedicar esta parte do livro a ambos os processos. Entre outras coisas, isso fornecerá a oportunidade de destacar suas semelhanças e diferenças de uma perspectiva numérica. Além disso, a discussão será relevante para a próxima parte do livro, na qual abordaremos as equações diferenciais.

Embora a derivação seja ensinada antes da integração nos cursos de cálculo, essa ordem foi invertida nos próximos capítulos, por várias razões. Primeiro, as bases da derivação numérica já foram introduzidas no Capítulo 4. Segundo, em parte por ser muito menos sensível a erros de arredondamento, a integração representa uma área mais desenvolvida dos métodos numéricos. Por fim, embora a derivação numérica não seja tão amplamente empregada, ela tem grande importância para a

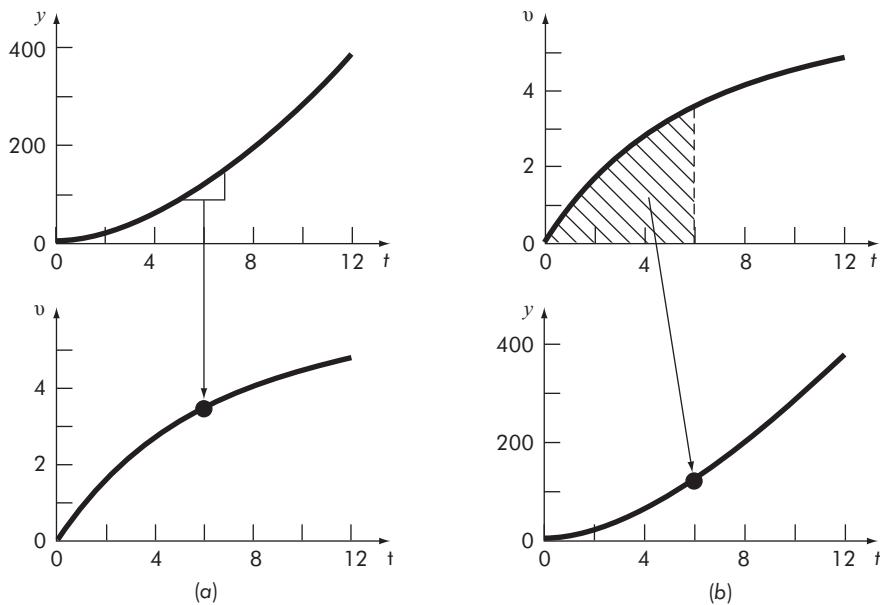


FIGURA PT5.1 O contraste entre (a) a derivação e (b) a integração.

solução de equações diferenciais. Assim, faz sentido abordá-la como o último tópico antes de descrever as equações diferenciais na Parte 6.

5.2 ORGANIZAÇÃO DA PARTE 5

O Capítulo 19 é dedicado às abordagens mais comuns de integração numérica – as *fórmulas de Newton-Cotes*, que são baseadas na substituição de uma função complicada ou de dados tabulados por um polinômio que é fácil de integrar. Três das mais amplamente usadas fórmulas de Newton-Cotes serão discutidas em detalhes: a *regra do trapézio*, a *regra 1/3 de Simpson* e a *regra 3/8 de Simpson*. Elas se destinam aos casos nos quais os dados a serem integrados estão igualmente espaçados. Além disso, também incluímos a discussão da integração numérica de dados desigualmente espaçados. Esse é um tópico muito importante, porque muitas aplicações do mundo real tratam de dados que têm essa forma.

Todo o material citado diz respeito à *integração fechada*, na qual os valores da função nos extremos de integração são conhecidos. No final do Capítulo 19, serão apresentadas *fórmulas de integração abertas*, nas quais os extremos de integração se estendem para além do intervalo de dados conhecidos; e embora não sejam geralmente usadas para integração definida, estão apresentadas aqui porque são extensivamente usadas na solução de equações diferenciais ordinárias na Parte 6.

As formulações estudadas no Capítulo 19 podem ser usadas para analisar tanto dados tabulados como equações. O Capítulo 20 trata de duas técnicas projetadas especialmente para integrar equações e funções: a *integração de Romberg* e a *quadratura*

de Gauss. São fornecidos algoritmos computacionais para ambos os métodos. Além disso, a *quadratura adaptativa* é discutida.

No Capítulo 21, são apresentadas informações adicionais sobre *derivação numérica* para suplementar o material introdutório do Capítulo 4. Os tópicos incluem *fórmulas de diferenças finitas de alta acurácia*, *extrapolação de Richardson* e derivação de dados desigualmente espaçados. Os efeitos dos erros na derivação e na integração numéricas também são discutidos.

Fórmulas de integração numérica

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar a integração numérica. Após esta leitura completa, você será capaz de:

- ✓ Reconhecer que as fórmulas de integração de Newton-Cotes são baseadas na estratégia de substituir uma função complicada ou dados tabulados por um polinômio que é fácil de integrar.
- ✓ Aprender como implementar as seguintes aplicações únicas das fórmulas de Newton-Cotes:
 - Regra do trapézio;
 - Regra 1/3 de Simpson;
 - Regra 3/8 de Simpson.
- ✓ Aprender como implementar as seguintes aplicações múltiplas das fórmulas de Newton-Cotes:
 - Regra do trapézio;
 - Regra 1/3 de Simpson.
- ✓ Reconhecer que fórmulas para número de segmentos par e número de pontos ímpar, como a regra 1/3 de Simpson, alcançam acurácia maior do que o esperado.
- ✓ Aprender como usar a regra do trapézio para integrar dados desigualmente espaçados.
- ✓ Compreender a diferença entre fórmulas de integração abertas e fechadas.

VOCÊ TEM UM PROBLEMA

Lembre-se de que a velocidade de um saltador de *bungee jumping* em queda livre como uma função do tempo pode ser calculada como

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) \quad (19.1)$$

Considere que gostaríamos de saber a distância vertical z que o saltador caiu após um certo tempo t . Essa distância pode ser calculada por integração:

$$z(t) = \int_0^t v(t) dt \quad (19.2)$$

Substituindo a Equação (19.1) na Equação (19.2), tem-se

$$z(t) = \int_0^t \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right) dt \quad (19.3)$$

Portanto, a integração fornece os meios para determinar a distância a partir da velocidade. O cálculo pode ser usado para resolver a Equação (19.3) para fornecer

$$z(t) = \frac{m}{c_d} \ln \left[\cosh\left(\sqrt{\frac{gc_d}{m}} t\right) \right] \quad (19.4)$$

Embora uma solução na forma fechada possa ser desenvolvida para esse caso, existem outras funções que não podem ser integradas analiticamente. Além disso, considere que haja alguma maneira de medir a velocidade do saltador em vários instantes de tempo durante a queda. Essas velocidades e os instantes de tempo correspondentes poderiam ser organizados como uma tabela de valores discretos. Nessa situação, também seria possível integrar os dados discretos para determinar a distância. Em ambos os casos, métodos de integração numérica estão disponíveis para obter soluções. Os Capítulos 19 e 20 introduzirão alguns desses métodos.

19.1 INTRODUÇÃO E CONTEXTUALIZAÇÃO

19.1.1 O que é integração?

De acordo com a definição do dicionário, integrar significa “juntar as partes em um todo; unir; indicar a quantidade total...”. Matematicamente, a integração (definida) é representada por

$$I = \int_a^b f(x) dx \quad (19.5)$$

o que indica a integral da função $f(x)$ em relação à variável independente x , avaliada entre os limites $x = a$ e $x = b$.

Como sugere a definição do dicionário, o “significado” da Equação (19.5) é o valor total ou a soma de $f(x) dx$ no intervalo de $x = a$ a b . De fato, o símbolo \int é, na realidade, um S maiúsculo estilizado usado para indicar a conexão íntima entre integração e soma.

A Figura 19.1 representa uma descrição gráfica do conceito. Para funções que estão acima do eixo x , a integral expressa pela Equação (19.5) corresponde à área sob a curva de $f(x)$ entre $x = a$ e b .

A integração numérica é, às vezes, chamada de quadratura, um termo arcaico que originalmente significava a construção de um quadrado tendo a mesma área que alguma figura curvilínea. Atualmente, o termo *quadratura* é geralmente tomado como sinônimo de integração definida numérica.

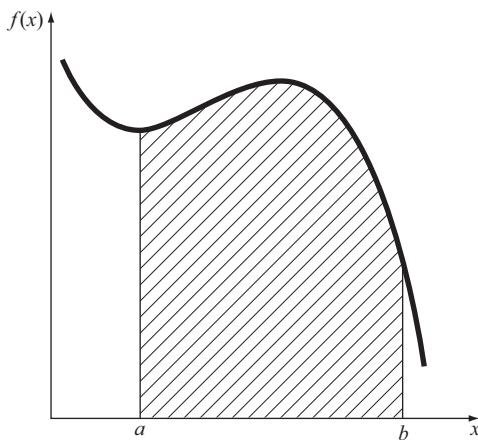


FIGURA 19.1 Representação gráfica da integral de $f(x)$ entre os limites $x = a$ e b . A integral é equivalente à área abaixo da curva.

19.1.2 Integração na engenharia e na ciência

A integração tem tantas aplicações científicas e de engenharia que são necessários cursos de cálculo integral já no primeiro ano da faculdade. Muitos exemplos específicos dessas aplicações podem ser dados em todos os campos da engenharia e da ciência, e diversos deles relacionam diretamente a ideia de integral com a área sob a curva. A Figura 19.2 descreve alguns casos nos quais a integração é usada para esse propósito.

Outra aplicação comum da integral se relaciona com a analogia entre integração e soma. Por exemplo, uma aplicação comum é determinar a média de uma função contínua. Lembre-se de que a média de n pontos discretos pode ser calculada por [Equação (14.2)]:

$$\text{Média} = \frac{\sum_{i=1}^n y_i}{n} \quad (19.6)$$

onde y_i são medidas individuais. A determinação da média de pontos discretos é ilustrada na Figura 19.3a.

Considere que y seja uma função contínua de uma variável independente x , como descreve a Figura 19.3b. Nesse caso, existe um número infinito de valores entre a e b . Do mesmo modo que a Equação (19.6) pode ser aplicada para determinar a média de leituras discretas, pode também haver interesse em se calcular a média de uma função contínua $y = f(x)$ no intervalo de a até b . A integração é usada para esse propósito, como especificado pela fórmula

$$\text{Média} = \frac{\int_a^b f(x) dx}{b - a} \quad (19.7)$$

Essa fórmula tem centenas de aplicações em engenharia. Por exemplo, é usada para calcular o centro de gravidade de objetos irregulares na mecânica e na engenharia civil e para determinar a corrente eficaz (valor médio quadrático) na engenharia elétrica.

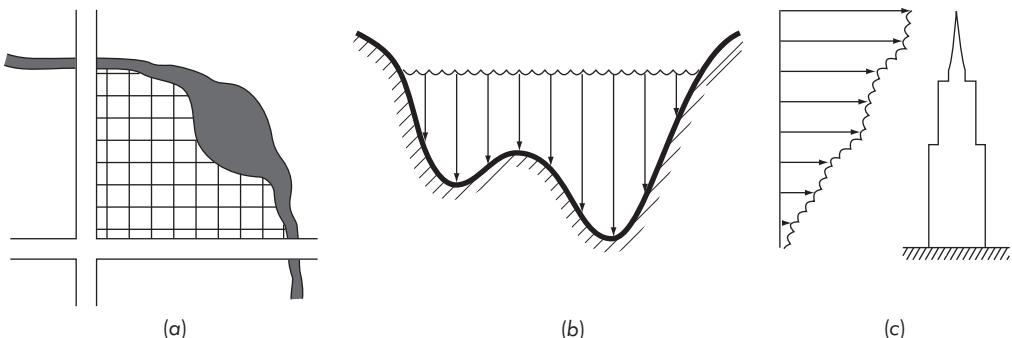


FIGURA 19.2 Exemplos de como a integração é usada para calcular áreas em aplicações de engenharia. (a) Um inspetor precisa saber a área de um campo limitado por um riacho sinuoso e duas estradas. (b) Um engenheiro de recursos hidráulicos precisa saber a área da seção transversal de um rio. (c) Um engenheiro de estruturas precisa determinar a força média decorrente de um vento não uniforme soprando contra o lado de um arranha-céu.

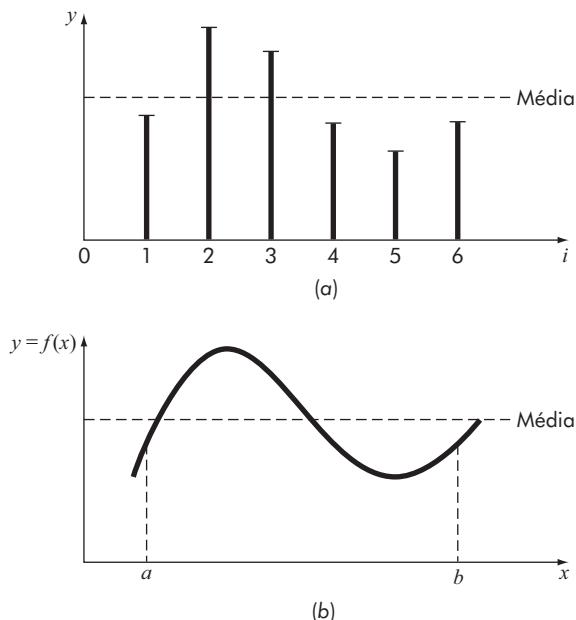


FIGURA 19.3 Ilustração da média para dados (a) discretos e (b) contínuos.

As integrais também são usadas pelos engenheiros e cientistas para calcular a quantidade total de uma dada variável física. A integral pode ser calculada sobre uma reta, uma área ou em um volume. Por exemplo, a massa total de um composto químico em um reator é dada pelo produto da concentração do composto químico pelo volume do reator, ou

$$\text{Massa} = \text{concentração} \times \text{volume}$$

onde a concentração tem unidades de massa por volume. Entretanto, considere que a concentração varie de uma posição a outra dentro do reator. Nesse caso, é necessário somar os produtos das concentrações locais c_i pelos elementos de volume correspondentes ΔV_i :

$$\text{Massa} = \sum_{i=1}^n c_i \Delta V_i$$

onde n é o número de volumes discretos. Para o caso contínuo, em que $c(x, y, z)$ é uma função conhecida e x, y e z são as variáveis independentes que indicam a posição em coordenadas cartesianas, a integração pode ser usada para o mesmo propósito:

$$\text{Massa} = \iiint c(x, y, z) dx dy dz$$

ou

$$\text{Massa} = \iiint_V c(V) dV$$

que é conhecida como uma *integral de volume*. Observe a forte analogia entre a soma e a integração.

Exemplos parecidos poderiam ser dados em outros campos da engenharia e da ciência. Por exemplo, a taxa total de transferência de energia através de um plano no qual o fluxo (em calorias por centímetros quadrados por segundo) é uma função da posição é dada por

$$\text{Taxa de transferência} = \iint_A \text{fluxo } dA$$

que é conhecida como uma *integral de área* (ou *de superfície*), onde A é a área.

Essas são apenas algumas poucas aplicações da integração com as quais o profissional pode se deparar regularmente no exercício de sua profissão. Quando as funções a serem analisadas forem simples, haverá normalmente a preferência pelo cálculo analítico. Entretanto, isso é difícil ou impossível quando a função é complicada, como é geralmente o caso em exemplos mais realistas. Além disso, a função subjacente, em geral, é desconhecida e é definida apenas por medidas em pontos discretos. Para ambos os casos, é necessário ter a habilidade de obter valores aproximados para integrais utilizando técnicas numéricas conforme descrito a seguir.

19.2 FÓRMULAS DE NEWTON-COTES

As *fórmulas de Newton-Cotes* são os esquemas mais comuns de integração numérica, e são baseadas na estratégia de substituir uma função complicada ou dados tabulados por um polinômio que seja fácil de integrar:

$$I = \int_a^b f(x) dx \cong \int_a^b f_n(x) dx \quad (19.8)$$

onde $f_n(x)$ é um polinômio da forma

$$f_n(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} + a_nx^n \quad (19.9)$$

e n é o grau do polinômio. Por exemplo, na Figura 19.4a, um polinômio de primeiro grau (uma reta) é usado como uma aproximação; já na Figura 19.4b, é usada uma parábola para o mesmo propósito.

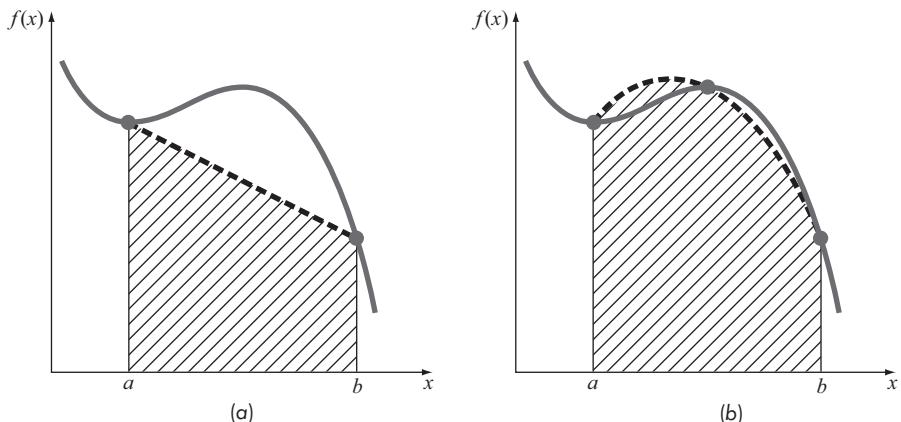


FIGURA 19.4 Aproximação de uma integral pela área sob (a) uma única reta e (b) uma única parábola.

A integral também pode ser aproximada utilizando uma série de polinômios aplicados por partes à função ou aos dados em segmentos de comprimento constante. Por exemplo, na Figura 19.5, três segmentos de reta são usados para aproximar a integral. Polinômios de grau mais alto podem ser usados para o mesmo propósito.

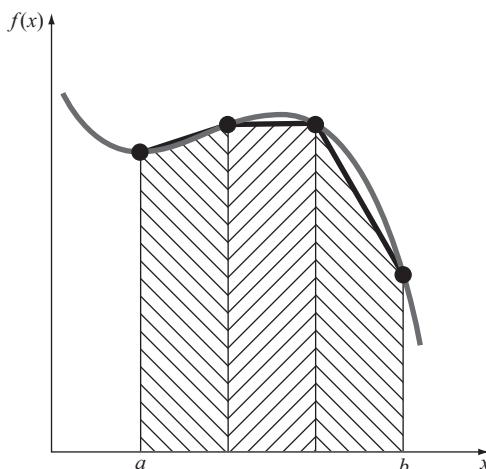


FIGURA 19.5 Aproximação de uma integral pela área sob três segmentos de reta.

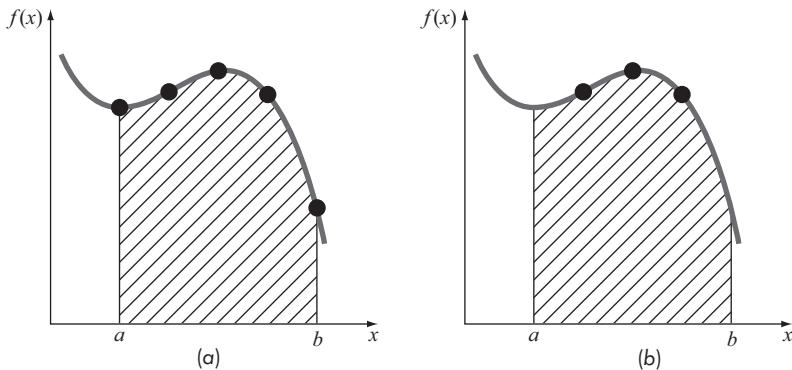


FIGURA 19.6 Diferença entre as fórmulas de integração (a) fechadas e (b) abertas.

Estão disponíveis formas abertas e fechadas das fórmulas de Newton-Cotes. As *formas fechadas* são aquelas nas quais os dados nos extremos inicial e final de integração são conhecidos (Figura 19.6a). As *formas abertas* têm extremos de integração que se estendem além do intervalo de dados (Figura 19.6b). Este capítulo enfatiza as formas fechadas. No entanto, material sobre as fórmulas de Newton-Cotes abertas é introduzido brevemente na Seção 19.7.

19.3 A REGRA DO TRAPÉZIO

A *regra do trapézio* é a primeira fórmula de integração fechada de Newton-Cotes. Ela corresponde ao caso no qual o polinômio na Equação (19.8) é de primeiro grau:

$$I = \int_a^b \left[f(a) + \frac{f(b) - f(a)}{b - a}(x - a) \right] dx \quad (19.10)$$

O resultado da integração é

$$I = (b - a) \frac{f(a) + f(b)}{2} \quad (19.11)$$

que é chamada de *regra do trapézio*.

Geometricamente, a regra dos trapézios é equivalente a aproximar a integral pela área do trapézio sob a reta que liga $f(a)$ e $f(b)$ na Figura 19.7. Lembre-se que, em geometria, a fórmula para calcular a área de um trapézio é a altura vezes a média das bases. Em nosso caso, o conceito é o mesmo, mas o trapézio está sobre o seu lado. Portanto, a estimativa da integral pode ser representada por

$$I = \text{largura} \times \text{altura média} \quad (19.12)$$

ou

$$I = (b - a) \times \text{altura média} \quad (19.13)$$

onde, para a regra dos trapézios, a altura média é a média dos valores da função nas extremidades, ou $[f(a) + f(b)]/2$.

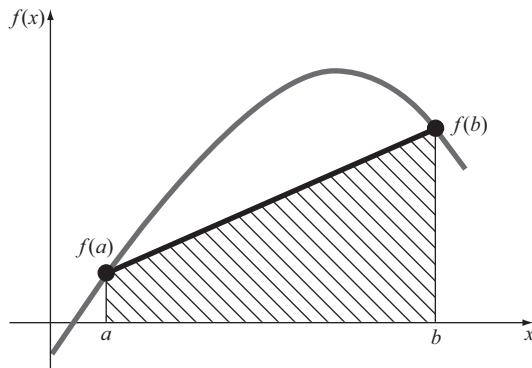


FIGURA 19.7 Descrição gráfica da regra do trapézio.

Todas as fórmulas de Newton-Cotes fechadas podem ser expressas na forma geral da Equação (19.13), e, de fato, elas diferem somente com relação à formulação da altura média.

19.3.1 Erro na regra do trapézio

Quando empregamos a integral sob um segmento de reta para aproximar a integral sob a curva, obviamente incorremos em um erro que pode ser substancial (Figura 19.8). Uma estimativa para o erro de truncamento local de uma única aplicação da regra do trapézio é

$$E_t = -\frac{1}{12} f''(\xi)(b-a)^3 \quad (19.14)$$

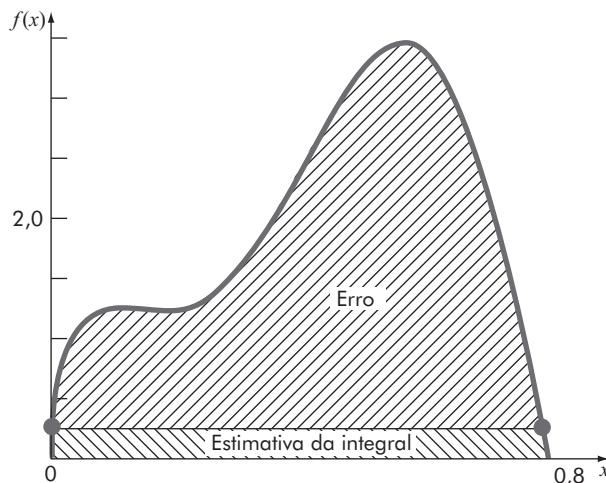


FIGURA 19.8 Descrição gráfica de uma única aplicação da regra do trapézio para aproximar a integral de $f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$ de $x = 0$ a $0,8$.

onde ξ está em algum ponto do intervalo entre a e b . A Equação (19.14) indica que, se a função que está sendo integrada for linear, a regra dos trapézios será exata. Caso contrário, para funções com derivada de segunda ordem e de ordem superior não nulas (isto é, com curvatura), pode ocorrer algum erro.

EXEMPLO 19.1 Uma única aplicação da regra dos trapézios

Use a Equação (19.11) para integrar numericamente

$$f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

de $a = 0$ a $b = 0,8$. Observe que o valor exato da integral pode ser determinado analiticamente como 1,640533.

Solução. Os valores da função $f(0) = 0,2$ e $f(0,8) = 0,232$ podem ser substituídos na Equação (19.11) para fornecer

$$I = (0,8 - 0) \frac{0,2 + 0,232}{2} = 0,1728$$

o que representa um erro de $E_t = 1,640533 - 0,1728 = 1,467733$, que corresponde a um erro relativo percentual de $\varepsilon_r = 89,5\%$. A razão para esse erro considerável é evidente na descrição gráfica na Figura 19.8. Observe que a área sob a reta despreza uma parte significativa da integral, que fica acima da reta.

Em situações reais, não teríamos conhecimento prévio do valor verdadeiro. Portanto, é necessária uma estimativa de erro aproximada. Para obter essa estimativa, a segunda derivada da função pode ser obtida derivando-se a função original duas vezes para obter

$$f''(x) = -400 + 4.050x - 10.800x^2 + 8.000x^3$$

O valor médio da segunda derivada pode ser calculado usando-se a Equação (19.7):

$$\bar{f}''(x) = \frac{\int_0^{0,8} (-400 + 4.050x - 10.800x^2 + 8.000x^3) dx}{0,8 - 0} = -60$$

o qual pode ser substituído na Equação (19.14) para fornecer

$$E_a = -\frac{1}{12}(-60)(0,8)^3 = 2,56$$

que é da mesma ordem de grandeza e do mesmo sinal que o erro verdadeiro. Entretanto, existe uma discrepância porque, para um intervalo desse tamanho, a média da segunda derivada não é necessariamente uma aproximação acurada de $f''(\xi)$. Assim, indicamos que o erro é aproximado usando a notação E_a em vez de E_t , que seria o erro exato.

19.3.2 A aplicação múltipla da regra do trapézio

Uma maneira de melhorar a acurácia da regra do trapézio é dividir o intervalo de integração de a a b em diversos segmentos e aplicar o método a cada segmento (Figura 19.9). As áreas correspondentes aos segmentos individuais podem então ser somadas para fornecer a integral para o intervalo inteiro. As equações resultantes são chamadas *fórmulas de integração por aplicações múltiplas ou compostas*.

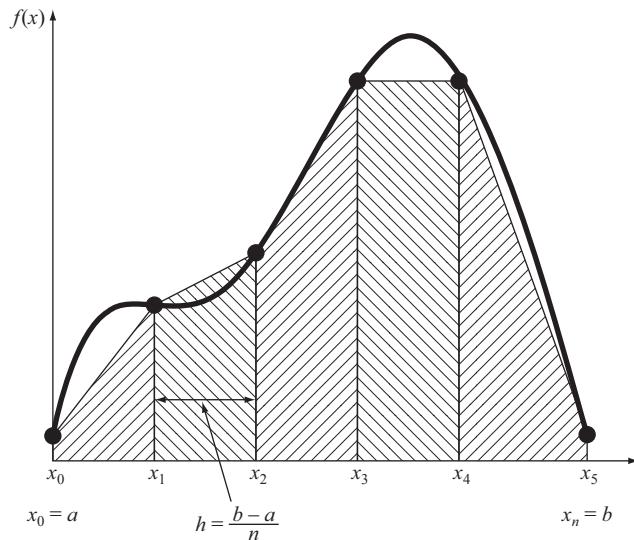


FIGURA 19.9 Aplicação múltipla da regra do trapézio.

A Figura 19.9 mostra a forma geral e a nomenclatura que será usada para caracterizar as integrais por aplicações múltiplas. Existem $n + 1$ pontos-base igualmente espaçados ($x_0, x_1, x_2, \dots, x_n$). Consequentemente, existem n segmentos de largura igual:

$$h = \frac{b - a}{n} \quad (19.15)$$

Se a e b forem designados por x_0 e x_n , respectivamente, a integral total pode ser representada como

$$I = \int_{x_0}^{x_1} f(x) dx + \int_{x_1}^{x_2} f(x) dx + \dots + \int_{x_{n-1}}^{x_n} f(x) dx$$

Substituindo cada integral pela regra do trapézio, obtém-se

$$I = h \frac{f(x_0) + f(x_1)}{2} + h \frac{f(x_1) + f(x_2)}{2} + \dots + h \frac{f(x_{n-1}) + f(x_n)}{2} \quad (19.16)$$

ou, agrupando os termos:

$$I = \frac{h}{2} \left[f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right] \quad (19.17)$$

ou, usando a Equação (19.15) para expressar a Equação (19.7) na forma geral da Equação (19.13):

$$I = \underbrace{(b - a)}_{\text{Largura}} \underbrace{\frac{f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n)}{2n}}_{\text{Altura média}} \quad (19.18)$$

Como a soma dos coeficientes de $f(x)$ no numerador dividido por $2n$ é igual a 1, a altura média representa uma média ponderada dos valores da função. De acordo com a Equação (19.18), os pontos interiores têm um peso duas vezes maior do que as extremidades $f(x_0)$ e $f(x_n)$.

Um erro para a aplicação múltipla da regra do trapézio pode ser obtido pela soma dos erros individuais em cada segmento, o que fornece

$$E_t = -\frac{(b-a)^3}{12n^3} \sum_{i=1}^n f''(\xi_i) \quad (19.19)$$

onde $f''(\xi_i)$ é a segunda derivada em um ponto ξ_i localizado no segmento i . Esse resultado pode ser simplificado por uma estimativa do valor médio da segunda derivada no intervalo todo como

$$\bar{f}'' \equiv \frac{\sum_{i=1}^n f''(\xi_i)}{n} \quad (19.20)$$

Portanto, $\sum f''(\xi_i) \cong n \bar{f}''$, e a Equação (19.19) pode ser reescrita como

$$E_a = -\frac{(b-a)^3}{12n^2} \bar{f}'' \quad (19.21)$$

Logo, se o número de segmentos for dobrado, o erro de truncamento será dividido por quatro. Observe que a Equação (19.21) é um erro aproximado devido à natureza aproximada da Equação (19.20).

EXEMPLO 19.2 Aplicação múltipla da regra dos trapézios

Use a regra do trapézio com dois segmentos para obter uma estimativa da integral de

$$f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

de $a = 0$ a $b = 0,8$. Empregue a Equação (19.21) para fazer uma estimativa do erro. Lembre-se de que o valor correto da integral é 1,640533.

Solução. Para $n = 2$ ($h = 0,4$):

$$f(0) = 0,2 \quad f(0,4) = 2,456 \quad f(0,8) = 0,232$$

$$I = 0,8 \frac{0,2 + 2(2,456) + 0,232}{4} = 1,0688$$

$$E_t = 1,640533 - 1,0688 = 0,57173 \quad \varepsilon_t = 34,9\%$$

$$E_a = -\frac{0,8^3}{12(2)^2}(-60) = 0,64$$

onde -60 é a média da segunda derivada determinada anteriormente no Exemplo 19.1.

Os resultados do exemplo anterior, junto com os da aplicação da regra do trapézio para 3 a 10 segmentos, estão resumidos na Tabela 19.1. Observe como o erro diminui à medida que o número de segmentos aumenta. Entretanto, observe também que a taxa de diminuição é gradual. Isso acontece porque o erro está inversamente relacionado com o quadrado de n [Equação (19.21)]. Portanto, dobrar o número de segmentos faz o erro tornar-se quatro vezes menor. Nas seções subsequentes, serão desenvolvidas fórmulas de ordem superior que são mais acuradas e que convergem mais rapidamente para a integral verdadeira à medida que o número de segmentos aumenta. Mas, antes de investigar essas fórmulas, será primeiro discutido como o MATLAB pode ser usado para implementar a regra do trapézio.

TABELA 19.1 Resultados da aplicação múltipla da regra do trapézio para obter uma estimativa da integral de $f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$ de $x = 0$ a $0,8$. O valor exato é 1,640533.

<i>n</i>	<i>h</i>	<i>I</i>	ϵ_i (%)
2	0,4	1,0688	34,9
3	0,2667	1,3695	16,5
4	0,2	1,4848	9,5
5	0,16	1,5399	6,1
6	0,1333	1,5703	4,3
7	0,1143	1,5887	3,2
8	0,1	1,6008	2,4
9	0,0889	1,6091	1,9
10	0,08	1,6150	1,6

19.3.3 Função do MATLAB: `trap`

Um algoritmo simples para implementar a aplicação múltipla da regra do trapézio pode ser escrito como apresenta a Figura 19.10. A função a ser integrada com os limites de integração e o número de segmentos são argumentos de entrada da função. Um laço é empregado para determinar a integral seguindo a Equação (19.18).

Uma aplicação da função pode ser desenvolvida para determinar a distância percorrida pelo saltador de *bungee jumping* em queda livre nos primeiros 3 s pelo cálculo da integral da Equação (19.3). Para esse exemplo, considere os seguintes valores de parâmetros: $g = 9,81 \text{ m/s}^2$, $m = 68,1 \text{ kg}$ e $c_d = 0,25 \text{ kg/m}$. Observe que o valor exato da integral pode ser calculado com a Equação (19.4) como 41,94805.

A função a ser integrada pode ser definida em um arquivo-M ou como uma função anônima,

```
>> v=@(t) sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t)
v =
```

```
@(t)sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t)
```

Calcularemos a integral com uma aproximação grosseira de cinco segmentos:

```

function I = trap(func,a,b,n,varargin)
% trap: aplicação múltipla da regra do trapézio
% I = trap(func,a,b,n,pl,p2,...):
%     regra do trapézio composta
% entrada:
% func = nome da função a ser integrada
% a, b = limites de integração
% n = número de segmentos (padrão = 100)
% pl,p2,... = parâmetros adicionais usados por func
% saída:
% I = Valor estimado da integral
if nargin<3,error('são necessários pelo menos 3 argumentos de entrada'),end
if ~(b>a),error('o limite superior deve ser maior do que o inferior'),end
if nargin<4|isempty(n),n=100;end
x = a; h = (b - a)/n;
s=func(a,varargin{:});
for i = 1 : n-1
    x = x + h;
    s = s + 2*func(x,varargin{:});
end
s = s + func(b,varargin{:});
I = (b - a) * s/(2*n);

```

FIGURA 19.10 Função do MATLAB para implementar a aplicação múltipla da regra do trapézio.

```

>> format long
>> trap(v,0,3,5)
ans =
    41.869929590727352

```

Observe que esse resultado já fornece um erro bastante pequeno de 0,186%. Para obter um resultado ainda mais acurado, podemos usar uma aproximação bastante fina baseada em 10.000 segmentos:

```

>> trap(v,0,3,10000)
ans =
    41.948049999175282

```

19.4 REGRAS DE SIMPSON

Além de aplicar a regra do trapézio com segmentos menores, outra forma de obter uma estimativa mais acurada de uma integral é usar polinômios de grau mais alto para ligar os pontos. Por exemplo, se existir um ponto extra no ponto médio entre $f(a)$ e $f(b)$, os três pontos podem ser ligados por uma parábola (Figura 19.11a). Se existirem dois pontos igualmente espaçados entre $f(a)$ e $f(b)$, os quatro pontos podem ser ligados por polinômios de 3º grau (Figura 19.11b). As fórmulas que resultam de tomar as integrais desses polinômios são chamadas de *regras de Simpson*.

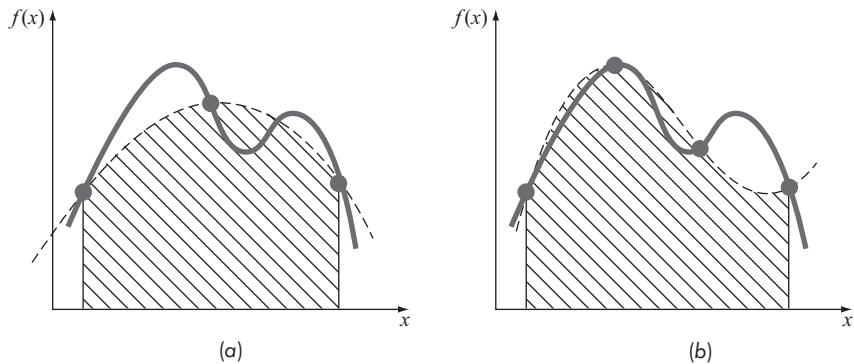


FIGURA 19.11 (a) Descrição gráfica da regra 1/3 de Simpson: ela consiste em tomar a área sob uma parábola ligando três pontos. (b) Descrição gráfica da regra 3/8 de Simpson: ela consiste em tomar a área sob uma equação cúbica ligando quatro pontos.

19.4.1 A regra 1/3 de Simpson

A regra 1/3 de Simpson corresponde ao caso em que o polinômio na Equação (19.8) é de segundo grau:

$$\begin{aligned} I = & \int_{x_0}^{x_2} \left[\frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \right. \\ & \left. + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \right] dx \end{aligned}$$

onde a e b são designados como x_0 e x_2 , respectivamente. O resultado da integração é

$$I = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] \quad (19.22)$$

em que, para esse caso, $h = (b - a)/2$. Essa equação é conhecida como *regra 1/3 de Simpson*. A designação “1/3” vem do fato de que h está dividido por 3 na Equação (19.22). A regra 1/3 de Simpson também pode ser expressa usando o formato da Equação (19.13):

$$I = (b - a) \frac{f(x_0) + 4f(x_1) + f(x_2)}{6} \quad (19.23)$$

onde $a = x_0$, $b = x_2$ e x_1 é o ponto médio entre a e b , o qual é dado por $(a + b)/2$. Observe que, de acordo com a Equação (19.23), o ponto médio tem peso de dois terços, e os dois pontos extremos, de um sexto.

É possível mostrar que a aplicação da regra 1/3 de Simpson para um único segmento tem um erro de truncamento de

$$E_t = -\frac{1}{90} h^5 f^{(4)}(\xi)$$

ou, como $h = (b - a)/2$:

$$E_t = -\frac{(b-a)^5}{2.880} f^{(4)}(\xi) \quad (19.24)$$

em que ξ é algum ponto no intervalo entre a e b . Logo, a regra 1/3 de Simpson é mais acurada que a regra do trapézio. No entanto, a comparação com a Equação (19.14) indica que ela é mais acurada que o esperado. Em vez de ser proporcional à terceira derivada, o erro é proporcional à quarta derivada. Consequentemente, a regra 1/3 de Simpson é acurada até a terceira ordem, mesmo quando baseada em apenas três pontos; ou seja, ela fornece resultados exatos para polinômios cúbicos, mesmo tendo sido deduzida a partir de uma parábola.

EXEMPLO 19.3 Uma única aplicação da regra 1/3 de Simpson

Use a Equação (19.23) para integrar

$$f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

de $a = 0$ a $b = 0,8$. Lembre-se de que a integral exata é 1,640533.

Solução. Para $n = 2$ ($h = 0,4$):

$$f(0) = 0,2 \quad f(0,4) = 2,456 \quad f(0,8) = 0,232$$

$$I = 0,8 \frac{0,2 + 4(2,456) + 0,232}{6} = 1,367467$$

$$E_t = 1,640533 - 1,367467 = 0,2730667 \quad \varepsilon_t = 16,6\%$$

o que é aproximadamente 5 vezes mais acurado do que uma única aplicação da regra do trapézio (Exemplo 19.1). O erro aproximado pode ser estimado como

$$E_a = -\frac{0,8^5}{2.880} (-2.400) = 0,2730667$$

onde -2.400 é o valor médio da quarta derivada para o intervalo. De modo semelhante ao Exemplo 19.1, o erro é aproximado (E_a) porque o valor médio da quarta derivada não é uma estimativa exata de $f^{(4)}(\xi)$. Entretanto, como esse caso trata de um polinômio de quinto grau, o resultado coincide.

19.4.2 A aplicação múltipla da regra 1/3 de Simpson

Da mesma forma que a regra do trapézio, a regra de Simpson pode ser melhorada dividindo-se o intervalo de integração em diversos segmentos do mesmo comprimento (Figura 19.12). A integral total pode ser representada como

$$I = \int_{x_0}^{x_2} f(x) dx + \int_{x_2}^{x_4} f(x) dx + \cdots + \int_{x_{n-2}}^{x_n} f(x) dx \quad (19.25)$$

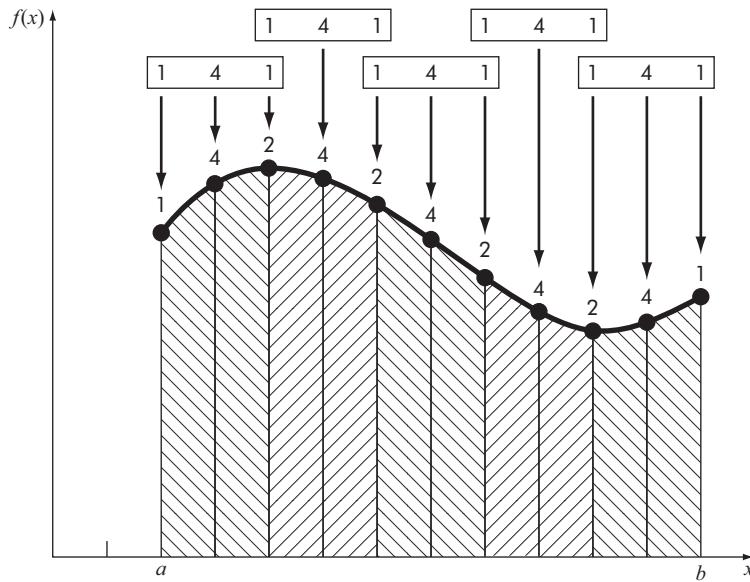


FIGURA 19.12 Aplicação múltipla da regra 1/3 de Simpson. Os pesos relativos estão representados acima dos valores da função. Observe que o método pode ser empregado apenas se o número de intervalos for par.

Substituindo cada integral individual pela regra 1/3 de Simpson, obtemos

$$I = 2h \frac{f(x_0) + 4f(x_1) + f(x_2)}{6} + 2h \frac{f(x_2) + 4f(x_3) + f(x_4)}{6} + \dots + 2h \frac{f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)}{6}$$

ou, combinando os termos e usando a Equação (19.15),

$$I = (b - a) \frac{f(x_0) + 4 \sum_{i=1,3,5}^{n-1} f(x_i) + 2 \sum_{j=2,4,6}^{n-2} f(x_j) + f(x_n)}{3n} \quad (19.26)$$

Observe que, como mostra a Figura 19.12, deve ser utilizado um número par de segmentos para implementar o método. Além disso, os coeficientes “4” e “2” na Equação (19.26) podem, a princípio, parecer peculiares; entretanto, seguem naturalmente da regra 1/3 de Simpson. Como mostra a Figura 19.12, os pontos ímpares representam o termo médio para cada aplicação e, portanto, levam o peso 4 da Equação (19.23). Os pontos pares são comuns a aplicações adjacentes e, então, são contados duas vezes.

Uma estimativa do erro para aplicação múltipla da regra de Simpson é obtida da mesma maneira que para a regra trapezoidal, somando-se os erros individuais para os segmentos e fazendo a média da derivada, resultando em

$$E_a = -\frac{(b - a)^5}{180n^4} \bar{f}^{(4)} \quad (19.27)$$

onde $\bar{f}^{(4)}$ é o valor médio da quarta derivada no intervalo.

EXEMPLO 19.4 Aplicação múltipla da regra 1/3 de Simpson

Use a Equação (19.26), com $n = 4$, para obter uma estimativa da integral de

$$f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

de $a = 0$ a $b = 0,8$. Empregue a Equação (19.27) para estimar o erro. Lembre-se de que a integral exata é 1,640533.

Solução. $n = 4$ ($h = 0,2$):

$$\begin{aligned} f(0) &= 0,2 & f(0,2) &= 1,288 \\ f(0,4) &= 2,456 & f(0,6) &= 3,464 \\ f(0,8) &= 0,232 \end{aligned}$$

Da Equação (19.26):

$$I = 0,8 \frac{0,2 + 4(1,288 + 3,464) + 2(2,456) + 0,232}{12} = 1,623467$$

$$E_t = 1,640533 - 1,623467 = 0,017067 \quad \varepsilon_t = 1,04\%$$

O erro estimado [Equação (19.27)] é

$$E_a = -\frac{(0,8)^5}{180(4)^4}(-2400) = 0,017067$$

que é exato (semelhante ao caso do Exemplo 19.3).

Como no Exemplo 19.4, a versão com aplicações múltiplas da regra 1/3 de Simpson fornece resultados muito acurados e, por essa razão, é considerada superior à regra do trapézio na maioria das aplicações. Entretanto, como mencionado anteriormente, ela é limitada aos casos nos quais os valores estão igualmente espaçados. Além disso, é limitada a situações nas quais haja um número par de segmentos e um número ímpar de pontos. Consequentemente, como será discutido na próxima seção, uma fórmula para um número ímpar de segmentos e par de pontos, conhecida como regra 3/8 de Simpson, é usada em conjunto com a regra 1/3, para permitir o cálculo tanto para um número par como para um número ímpar de segmentos.

19.4.3 Regra 3/8 de Simpson

De maneira análoga à dedução da regra do trapézio e da regra 1/3 de Simpson, um polinômio de Lagrange de terceiro grau pode ser ajustado a quatro pontos e integrado, para fornecer:

$$I = \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$$

onde $h = (b - a)/3$. Essa equação é chamada *regra 3/8 de Simpson*, porque h é multiplicado por 3/8. Ela é a terceira fórmula de integração de Newton-Cotes. A regra 3/8 também pode ser expressa na forma da Equação (19.13):

$$I = (b - a) \frac{f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)}{8} \quad (19.28)$$

Assim, os dois pontos interiores têm pesos de três oitavos, enquanto as extremidades têm peso de um oitavo. A regra de 3/8 de Simpson tem um erro de

$$E_t = -\frac{3}{80}h^5 f^{(4)}(\xi)$$

ou, como $h = (b - a)/3$,

$$E_t = -\frac{(b - a)^5}{6480} f^{(4)}(\xi) \quad (19.29)$$

Como o denominador da Equação (19.29) é maior do que o da Equação (19.24), a regra 3/8 é um pouco mais acurada do que a regra 1/3.

A regra 1/3 de Simpson é frequentemente o método preferido, pois alcança uma acurácia de terceira ordem com três pontos em vez dos quatro pontos necessários para a versão 3/8. Entretanto, a regra 3/8 tem utilidade quando o número de segmentos é ímpar. No caso do Exemplo 19.4, usamos a regra de Simpson para integrar a função com quatro segmentos. Considere que se deseja uma estimativa usando cinco segmentos. Uma opção é usar a versão com aplicação múltipla da regra do trapézio, como foi feito no Exemplo 19.2. No entanto, isso pode não ser aconselhável devido ao grande erro de truncamento associado com esse método. Uma alternativa é aplicar a regra 1/3 de Simpson aos primeiros dois segmentos e a regra 3/8 de Simpson aos últimos três (Figura 19.13). Dessa maneira, é possível obter-se uma estimativa com acurácia de terceira ordem em todo o intervalo.

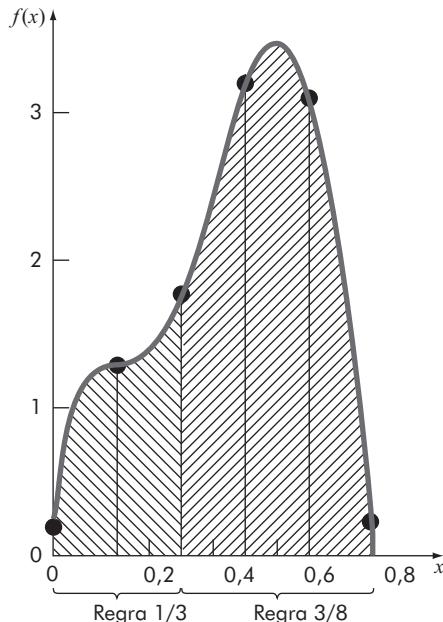


FIGURA 19.13 Ilustração de como as regras 1/3 e 3/8 de Simpson podem ser usadas em conjunto para tratar aplicações múltiplas com um número ímpar de intervalos.

EXEMPLO 19.5 A regra 3/3 de Simpson

(a) Use a regra 3/8 de Simpson para integrar

$$f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

de $a = 0$ a $b = 0,8$. (b) Use-a em conjunto com a regra 1/3 de Simpson para integrar a mesma função usando cinco segmentos.

Solução. (a) Uma única aplicação da regra 3/8 de Simpson exige quatro pontos igualmente espaçados:

$$\begin{array}{ll} f(0) = 0,2 & f(0,2667) = 1,432724 \\ f(0,5333) = 3,487177 & f(0,8) = 0,232 \end{array}$$

Usando a Equação (19.28):

$$I = 0,8 \frac{0,2 + 3(1,432724 + 3,487177) + 0,232}{8} = 1,51970$$

(b) Os dados necessários para uma aplicação com cinco segmentos ($h = 0,16$) são

$$\begin{array}{ll} f(0) = 0,2 & f(0,16) = 1,296919 \\ f(0,32) = 1,743393 & f(0,48) = 3,186015 \\ f(0,64) = 3,181929 & f(0,80) = 0,232 \end{array}$$

A integral para os primeiros dois segmentos é obtida usando-se a regra 1/3 de Simpson:

$$I = 0,32 \frac{0,2 + 4(1,296919) + 1,743393}{6} = 0,3803237$$

Para os três últimos segmentos, a regra 3/8 pode ser usada para obter

$$I = 0,48 \frac{1,743393 + 3(3,186015 + 3,181929) + 0,232}{8} = 1,264754$$

A integral total é calculada pela soma desses dois resultados:

$$I = 0,3803237 + 1,264754 = 1,645077$$

19.5 FÓRMULAS DE NEWTON-COTES DE ORDEM SUPERIOR

Como observado previamente, a regra do trapézio e ambas as regras de Simpson são membros de uma família de equações de integração conhecidas como fórmulas de integração fechadas de Newton-Cotes. Algumas das fórmulas estão resumidas na Tabela 19.2 junto com suas estimativas de erro.

Observe que, de modo semelhante às regras 1/3 e 3/8 de Simpson, as fórmulas de cinco e seis pontos têm erros da mesma ordem. Essa característica geral vale para fórmulas de mais pontos e leva ao resultado que fórmulas para um número par de segmentos (por exemplo, a regra 1/3 e a regra de Boole) em geral são os métodos preferidos.

Entretanto, é importante enfatizar também que, na prática da engenharia, as fórmulas de ordem superior (isto é, maior do que quatro pontos) raramente são usadas. As regras de Simpson são suficientes para a maioria das aplicações. A acurácia pode ser melhorada usando-se a versão com aplicações múltiplas. Além disso, quando a função for conhecida e for necessária alta acurácia, métodos como a integração de Romberg ou a quadratura de Gauss, descritos no Capítulo 20, oferecem alternativas viáveis e atrativas.

TABELA 19.2 Fórmulas de integração fechadas de Newton-Cotes. As fórmulas são apresentadas no formato da Equação (19.13) de modo que os pesos dos pontos dados na estimativa da altura média sejam aparentes. O tamanho do passo é dado por $h = (b - a)/n$.

Segmentos (n)	Pontos	Nome	Fórmula	Erro de truncamento
1	2	Regra do trapézio	$(b - a) \frac{f(x_0) + f(x_1)}{2}$	$-(1/12)h^3 f''(\xi)$
2	3	Regra 1/3 de Simpson	$(b - a) \frac{f(x_0) + 4f(x_1) + f(x_2)}{6}$	$-(1/90)h^5 f^{(4)}(\xi)$
3	4	Regra 3/8 de Simpson	$(b - a) \frac{f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)}{8}$	$-(3/80)h^5 f^{(4)}(\xi)$
4	5	Regra de Boole	$(b - a) \frac{7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)}{90}$	$-(8/945)h^7 f^{(6)}(\xi)$
5	6		$(b - a) \frac{19f(x_0) + 75f(x_1) + 50f(x_2) + 50f(x_3) + 75f(x_4) + 19f(x_5)}{288}$	$-(275/12,096)h^7 f^{(6)}(\xi)$

19.6 INTEGRAÇÃO COM SEGMENTOS DESIGUAIS

Até esse ponto, todas as fórmulas para integração numérica foram baseadas em dados igualmente espaçados. Na prática, existem muitas situações nas quais essa hipótese não é válida e precisamos lidar com segmentos de tamanhos distintos. Por exemplo, dados obtidos experimentalmente, muitas vezes, são desse tipo. Nesses casos, um método consiste em aplicar a regra do trapézio para cada segmento e somar os resultados:

$$I = h_1 \frac{f(x_0) + f(x_1)}{2} + h_2 \frac{f(x_1) + f(x_2)}{2} + \dots + h_n \frac{f(x_{n-1}) + f(x_n)}{2} \quad (19.30)$$

onde h_i é a largura do segmento i . Observe que essa foi a mesma abordagem usada na aplicação múltipla da regra do trapézio. A única diferença entre as Equações (19.16) e (19.30) é que os h 's na primeira são constantes.

EXEMPLO 19.6 Regra do trapézio com segmentos desiguais

A informação na Tabela 19.3 foi gerada usando o mesmo polinômio utilizado no Exemplo 19.1. Use a Equação (19.30) para determinar a integral a partir desses dados. Lembre-se de que a resposta correta é 1,640533.

TABELA 19.3 Dados para $f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$, com valores de x desigualmente espaçados.

x	$f(x)$	x	$f(x)$
0,00	0,200000	0,44	2,842985
0,12	1,309729	0,54	3,507297
0,22	1,305241	0,64	3,181929
0,32	1,743393	0,70	2,363000
0,36	2,074903	0,80	0,232000
0,40	2,456000		

Solução. Aplicando a Equação (19.30) aos dados da Tabela 19.3, obtemos

$$I = 0,12 \frac{0,2 + 1,309729}{2} + 0,10 \frac{1,309729 + 1,305241}{2} \\ + \dots + 0,10 \frac{2,363 + 0,232}{2} = 1,594801$$

o que representa um erro relativo percentual absoluto de $\varepsilon_r = 2,8\%$.

19.6.1 Função do MATLAB: `trapuneq`

Um algoritmo simples para implementar a regra do trapézio para dados desigualmente espaçados pode ser escrito como na Figura 19.14. Dois vetores, x e y , que contêm as variáveis independente e dependente são passados como argumentos de entrada para a função. São incluídas duas linhas específicas para verificação de erros, de modo a garantir que (a) os dois vetores sejam do mesmo tamanho e (b) os x 's estejam em ordem crescente.¹ Um laço é empregado para determinar a integral. Observe que os subscritos foram modificados daqueles da Equação (19.30), considerando que o MATLAB não permite índices iguais a zero em arranjos.

Uma aplicação da função pode ser desenvolvida para o mesmo problema que foi resolvido no Exemplo 19.6:

```
>> x=[0 .12 .22 .32 .36 .4 .44 .54 .64 .7 .8];
>> y=0.2+25*x-200*x.^2+675*x.^3-900*x.^4+400*x.^5;
>> trapuneq(x,y)
ans =
    1.5948
```

que é idêntico ao resultado obtido no Exemplo 19.6.

¹ A função `diff` é descrita na Seção 21.7.1.

```

function I = trapuneq(x,y)
% trapuneq: regra do trapézio para pontos desigualmente
espaçados
% I = trapuneq(x,y):
% Aplica a regra do trapézio para determinar a integral
% para n pontos dados (x, y) onde x e y devem ser do
% mesmo tamanho e x deve ser monotonicamente crescente
% entrada:
% x = vetor da variável independente
% y = vetor da variável dependente
% saída:
% I = Estimativa da integral
if nargin<2,error('são necessários pelo menos 2 argumentos de entrada'),end
if any(diff(x)<0),error('x não é monotonicamente crescente'),end
n = length(x);
if length(y)~=n,error('x e y devem ser do mesmo tamanho'),end
s = 0;
for k = 1:n-1
    s = s + (x(k+1)-x(k))*(y(k)+y(k+1))/2;
end
I = s;

```

FIGURA 19.14 Função do MATLAB para implementar a regra do trapézio para dados desigualmente espaçados.

19.6.2 Funções nativas do MATLAB: trapz e cumtrapz

O MATLAB tem uma função nativa que avalia integrais para dados da mesma maneira como a função que acabamos de apresentar na Figura 19.14. Ela tem a seguinte sintaxe

```
z = trapz(x,y)
```

onde os dois vetores, x e y , contêm as variáveis independente e dependente, respectivamente. A sessão simples do MATLAB a seguir mostra o uso dessa função para integrar os dados da Tabela 19.3:

```

>> x=[0 .12 .22 .32 .36 .4 .44 .54 .64 .7 .8];
>> y=0.2+25*x-200*x.^2+675*x.^3-900*x.^4+400*x.^5;
>> trapz(x,y)
ans =
    1.5948

```

Além disso, o MATLAB tem outra função, `cumtrapz`, que calcula a integral cumulativa. Uma representação simples de sua sintaxe é

```
z = cumtrapz(x, y)
```

onde os dois vetores, x e y , contêm as variáveis independente e dependente, respectivamente, e z é um vetor cujos elementos $z(k)$ armazenam a integral de $x(1)$ até $x(k)$.

EXEMPLO 19.7 Utilizando integração numérica para calcular a distância a partir da velocidade

Como descrito no início deste capítulo, uma boa aplicação da integral é calcular a distância $z(t)$ de um objeto com base em sua velocidade $v(t)$ como em [lembre-se da Equação (19.2)]:

$$z(t) = \int_0^t v(t) dt$$

Considere que se tenham medições de velocidade em uma série de instantes discretos de tempo desigualmente espaçados durante a queda livre. Use a Equação (19.2) para sinteticamente gerar tal informação para um saltador de 70 kg, com um coeficiente de arraste de 0,275 kg/m. Incorpore algum erro aleatório aos dados arredondando as velocidades para o inteiro mais próximo. Então, use a função `cumtrapz` para determinar a distância percorrida durante a queda e compare os resultados com a solução analítica [Equação (19.4)]. Além disso, faça um gráfico mostrando a distância em função do tempo determinada analiticamente e numericamente.

Solução. Alguns instantes de tempo desigualmente espaçados e velocidades arredondadas podem ser gerados como

```
>> format short g
>> t=[0 1 1.4 2 3 4.3 6 6.7 8];
>> g=9.81;m=70;cd=0.275;
>> v=round(sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t));
```

As distâncias podem, então, ser calculadas como

```
>> z=cumtrapz(t,v)
z =
    0   5   9.6   19.2   41.7   80.7   144.45   173.85   231.7
```

Portanto, após 8 segundos, o saltador caiu 231,7 m. Esse resultado é razoavelmente próximo à solução analítica [Equação (19.4)]:

$$z(t) = \frac{70}{0,275} \ln \left[\cosh \left(\sqrt{\frac{9,81(0,275)}{70}} t \right) \right] = 234,1$$

Um gráfico das soluções analítica e numérica, considerando tanto as velocidades exatas como as arredondadas, pode ser gerado com os seguintes comandos:

```
>> ta=linspace(t(1),t(length(t)));
>> za=m/cd*log(cosh(sqrt(g*cd/m)*ta));
>> plot(ta,za,t,z,'o')
>> title('Distância versus tempo')
>> xlabel('t (s)'), ylabel('z (m)')
>> legend('analítica','numérica')
```

Como mostra a Figura 19.15, os resultados numérico e analítico concordam muito bem.

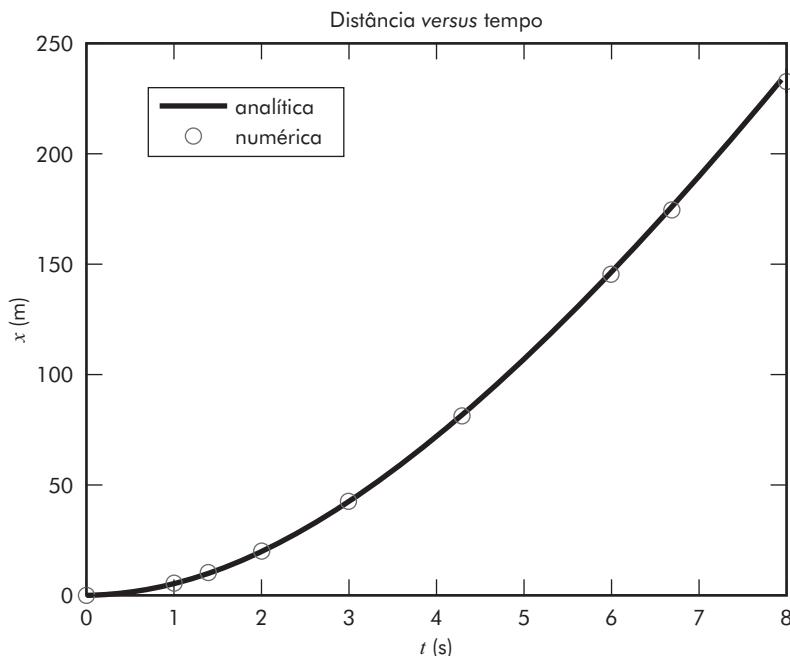


FIGURA 19.15 Gráfico de distância versus tempo. A linha contínua representa a solução analítica, enquanto os pontos foram determinados numericamente com a função `cumtrapz`.

19.7 FÓRMULAS DE INTEGRAÇÃO ABERTAS

Lembre-se, da Figura 19.6b, que as fórmulas de integração abertas têm extremos que se estendem além do intervalo dos dados. A Tabela 19.4 resume as *fórmulas de integração abertas de Newton-Cotes*. As fórmulas estão expressas no formato da Equação (19.13), de modo que os fatores de peso são evidentes. Como com as versões fechadas, pares sucessivos de fórmulas têm erros da mesma ordem. As fórmulas para números pares de segmentos e ímpares de pontos são, em geral, os métodos preferidos, pois exigem menos pontos para se obter a mesma acurácia do que as fórmulas para números ímpares de segmentos e pares de pontos.

As fórmulas abertas, em geral, não são usadas para o cálculo de integrais definidas; entretanto, terão utilidade na análise de integrais impróprias. Além disso, serão relevantes na discussão de métodos de passos múltiplos para a resolução de equações diferenciais ordinárias nos Capítulos 22 e 23.

TABELA 19.4 Fórmulas de integração abertas de Newton-Cotes. As fórmulas são apresentadas no formato da Equação (19.13), de modo que os pesos dos pontos dados na estimativa da altura média são aparentes. O tamanho do passo é dado por $h = (b - a)/n$.

Segmentos (n)	Pontos	Nome	Fórmula	Erro de truncamento
2	1	Método do ponto médio	$(b - a)f(x_1)$	$(1/3)h^3 f''(\xi)$
3	2		$(b - a)\frac{f(x_1) + f(x_2)}{2}$	$(3/4)h^3 f''(\xi)$
4	3		$(b - a)\frac{2f(x_1) - f(x_2) + 2f(x_3)}{3}$	$(14/45)h^5 f^{(4)}(\xi)$
5	4		$(b - a)\frac{11f(x_1) + f(x_2) + f(x_3) + 11f(x_4)}{24}$	$(95/144)h^5 f^{(4)}(\xi)$
6	5		$(b - a)\frac{11f(x_1) - 14f(x_2) + 26f(x_3) - 14f(x_4) + 11f(x_5)}{20}$	$(41/140)h^7 f^{(6)}(\xi)$

19.8 INTEGRAIS MÚLTIPAS

As integrais múltiplas são amplamente usadas na engenharia e na ciência. Por exemplo, uma equação geral para calcular a média de uma função bidimensional pode ser escrita como [lembre-se da Equação (19.7)]

$$\bar{f} = \frac{\int_c^d \left(\int_a^b f(x, y) dx \right) dy}{(d - c)(b - a)} \quad (19.31)$$

O numerador é chamado de *integral dupla*.

As técnicas discutidas neste capítulo (e no Capítulo 20) podem ser facilmente usadas para calcular integrais múltiplas. Um exemplo simples é tomar a integral dupla de uma função sobre uma área retangular (Figura 19.16).

Lembre-se, do cálculo, que essas integrais podem ser calculadas como integrais iteradas

$$\int_c^d \left(\int_a^b f(x, y) dx \right) dy = \int_a^b \left(\int_c^d f(x, y) dy \right) dx \quad (19.32)$$

Logo, a integral em uma das dimensões é calculada primeiro, e o seu resultado é integrado na segunda dimensão. A Equação (19.32) afirma que a ordem da integração não é importante.

Uma integral numérica dupla é baseada na mesma ideia. Primeiro, métodos como a regra do trapézio ou de Simpson para múltiplos segmentos são aplicados na primeira dimensão para cada valor da segunda dimensão mantido constante. Depois, o método é aplicado para integrar na segunda dimensão. A abordagem é ilustrada no exemplo a seguir.

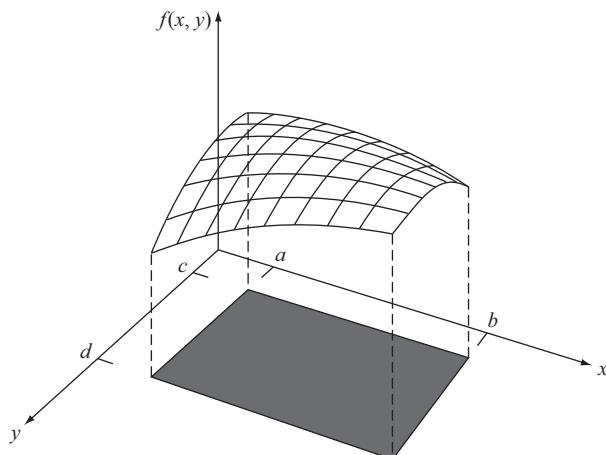


FIGURA 19.16 Integral dupla de uma função sobre uma área retangular.

EXEMPLO 19.8 Utilizando uma integral dupla para determinar a temperatura média

Considere que a temperatura de uma placa retangular aquecida seja descrita pela seguinte função:

$$T(x, y) = 2xy + 2x - x^2 - 2y^2 + 72$$

Se a placa tiver 8 m de comprimento (dimensão x) e 6 m de largura (dimensão y), calcule a temperatura média.

Solução. Usaremos simplesmente aplicações da regra do trapézio com dois segmentos em cada dimensão. As temperaturas nos valores necessários de x e y estão descritas na Figura 19.17. Observe que uma média simples desses valores é 47,33. A média da função também pode ser calculada analiticamente, fornecendo um resultado de 58,66667.

Para fazer o mesmo cálculo numericamente, a regra do trapézio é implementada ao longo da dimensão x para cada valor de y . Esses valores, então, são integrados ao longo da direção y para fornecer o resultado final de 2.544. Dividindo isso pela área, obtemos a temperatura média como sendo $2.544/(6 \times 8) = 53$.

Agora, podemos aplicar uma regra 1/3 de Simpson para um único segmento da mesma maneira. Isso resulta em uma integral de 2.816 e uma média de 58,66667, que é exata. Por que isso ocorre? Lembre-se de que a regra 1/3 de Simpson fornece resultados perfeitos para polinômios cúbicos. Como o termo de grau mais alto na função é de segundo grau, o mesmo resultado exato ocorre no caso presente.

Para funções algébricas de grau mais alto, bem como para funções transcendentais, é necessário usar aplicações múltiplas para alcançar uma estimativa mais acurada da integral. E o Capítulo 20 introduz técnicas que são mais eficientes do que a fórmula de Newton-Cotes para calcular integrais de funções dadas. Elas, em geral, fornecem uma forma superior para implementar integrações numéricas de integrais múltiplas.

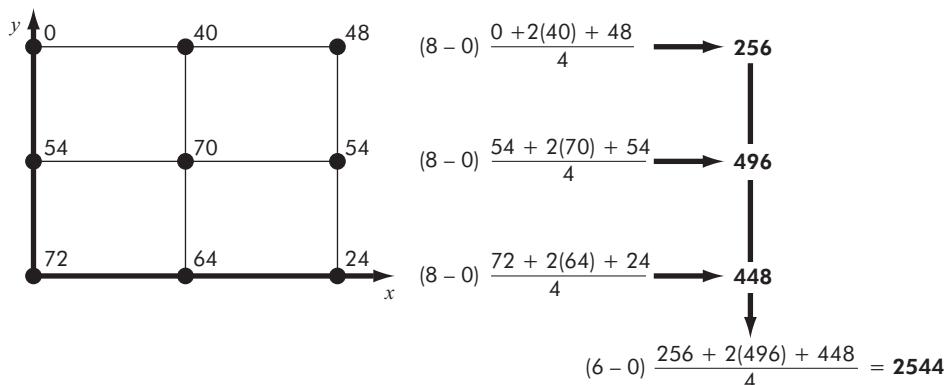


FIGURA 19.17 Cálculo numérico de uma integral dupla por meio da regra do trapézio para dois segmentos.

19.8.1 Funções nativas do MATLAB: dblquad e triplequad

O MATLAB tem funções nativas para implementar a integração numérica tanto de integrais duplas (dblquad) quanto de integrais triplas (triplequad). Uma representação simples da sintaxe para dblquad é

```
q = dblquad(fun, xmin, xmax, ymin, ymax, tol)
```

onde q é a integral dupla da função fun nos intervalos de x_{min} a x_{max} e de y_{min} a y_{max} . Se o parâmetro tol não é especificado, uma tolerância-padrão de 1×10^{-6} é usada.

O exemplo a seguir mostra como essa função pode ser usada para calcular a integral dupla avaliada no Exemplo 19.7:

```
>> q=dblquad(@(x,y) 2*x*y+2*x-x.^2-2*y.^2+72,0,8,0,6)
q =
    2816
```

► 19.9 ESTUDO DE CASO INTEGRAÇÃO NUMÉRICA PARA CALCULAR O TRABALHO

Contextualização. O cálculo do trabalho é um componente importante de várias áreas da engenharia e da ciência. A fórmula geral é

$$\text{Trabalho} = \text{força} \times \text{distância}$$

Quando esse conceito é estudado em física no ensino médio, são apresentadas aplicações simples que usam forças que permanecem constantes ao longo do deslocamento. Por exemplo, se uma força de 10 N é usada para puxar um bloco por uma distância de 5 m, o trabalho é calculado como 50 J (1 joule = 1 N · m).

Embora um cálculo tão simples seja útil para introduzir o conceito, problemas em situações reais geralmente são muito mais complexos. Por exemplo, considere que a força varie no decorrer do movimento. Nesses casos, a equação do trabalho é reescrita como

$$W = \int_{x_0}^{x_n} F(x) dx \quad (19.33)$$

onde W é o trabalho (J), x_0 e x_n são as posições inicial e final, respectivamente, e $F(x)$ é uma força que varia como função da posição. Se $F(x)$ for fácil de integrar, a Equação (19.33) pode ser calculada analiticamente. Entretanto, em um problema em situação real, a força pode não ser expressa de tal forma. De fato, quando forem analisados dados medidos, a força poderá estar disponível apenas de forma tabular. Nesses casos, a integração numérica é a única opção viável para os cálculos.

Mais complexidade é introduzida se o ângulo entre a força e a direção do movimento também variar como uma função da posição (Figura 19.18). A equação do trabalho pode ser ainda mais modificada para levar em conta esse efeito, como em

$$W = \int_{x_0}^{x_n} F(x) \cos[\theta(x)] dx \quad (19.34)$$

De novo, se $F(x)$ e $\theta(x)$ forem funções simples, a Equação (19.34) poderá ser resolvida analiticamente. No entanto, como na Figura 19.18, é mais provável que a relação funcional seja complicada. Para essas situações, os métodos numéricos fornecem a única alternativa para determinar a integral.

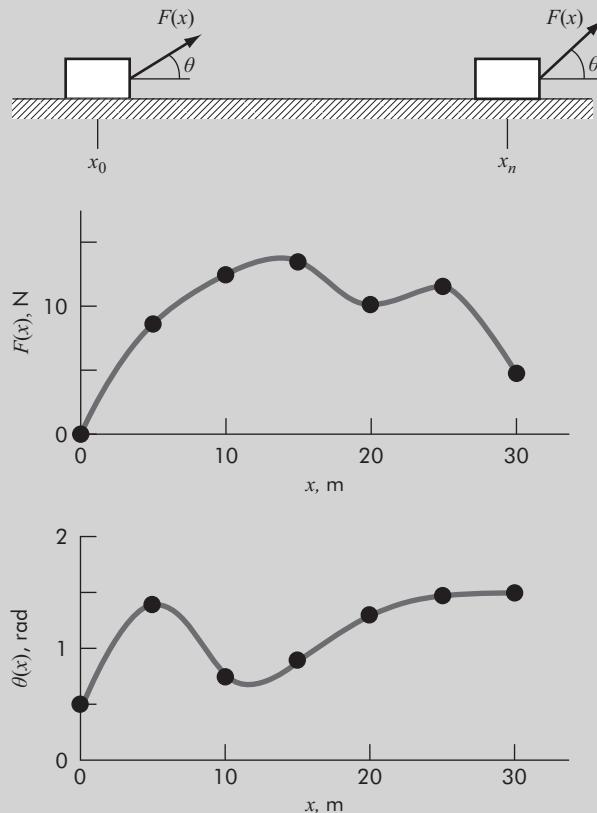


FIGURA 19.18 O caso de uma força variável agindo em um bloco. Nesse caso, o ângulo e o valor absoluto da força variam.



Considere que você devia fazer os cálculos para a situação descrita na Figura 19.18. Embora a figura mostre valores contínuos para $F(x)$ e $\theta(x)$, admita que, devido às restrições experimentais, foram fornecidas apenas medidas discretas em intervalos de $x = 5$ (Tabela 19.5). Use versões com uma única aplicação e com aplicações múltiplas da regra do trapézio e das regras 1/3 e 3/8 de Simpson para calcular o trabalho para esses dados.

TABELA 19.5 Dados para a força $F(x)$ e o ângulo $\theta(x)$ como uma função da posição x .

$x, \text{ m}$	$F(x), \text{ N}$	$\theta, \text{ rad}$	$F(x) \cos \theta$
0	0,0	0,50	0,0000
5	9,0	1,40	1,5297
10	13,0	0,75	9,5120
15	14,0	0,90	8,7025
20	10,5	1,30	2,8087
25	12,0	1,48	1,0881
30	5,0	1,50	0,3537

Solução. Os resultados da análise estão resumidos na Tabela 19.6. Foi calculado um erro relativo percentual ε_r com relação ao valor verdadeiro da integral de 129,52 que foi estimado com base nos valores tirados da Figura 19.18 em intervalos de 1 m.

TABELA 19.6 Estimativa do trabalho calculado por meio da regra do trapézio e as regras de Simpson. O erro relativo percentual ε_r é calculado com relação a um valor verdadeiro da integral (129,52 J) que foi estimado com base em valores em intervalos de 1 m.

Técnica	Segmentos	Trabalho	$\varepsilon_r, \%$
Regra do trapézio	1	5,31	95,9
	2	133,19	2,84
	3	124,98	3,51
	6	119,09	8,05
Regra 1/3 de Simpson	2	175,82	35,75
	6	117,13	9,57
Regra 3/8 de Simpson	3	139,93	8,04

Os resultados são interessantes porque a saída mais acurada ocorre para a simples regra do trapézio com dois segmentos. Estimativas mais refinadas usando mais segmentos, bem como as regras de Simpson, fornecem resultados menos acurados.

A razão para esse resultado aparentemente não intuitivo é que o espaçamento grosso dos pontos não é adequado para capturar as variações da força e dos ângulos. Isso é particularmente evidente na Figura 19.19, em que traçamos uma curva contínua para o produto de $F(x)$ e $\cos[\theta(x)]$. Observe como o uso de sete pontos para caracterizar uma função que varia no contínuo perde os dois picos em $x = 2,5$ e $12,5$ m. A omissão desses dois pontos limita a acurácia das estimativas por integração numérica na Tabela 19.6. O fato de a regra do trapézio com dois segmentos fornecer resultados mais acurados decorre de um acaso nas posições dos pontos para esse problema particular (Figura 19.20).

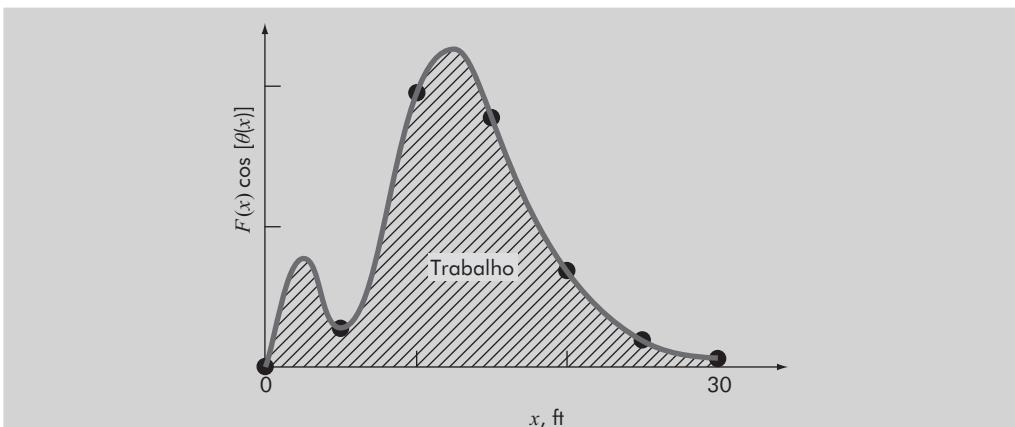


FIGURA 19.19 Gráfico no contínuo de $F(x) \cos[\theta(x)]$ em função da posição com os sete pontos discretos usados para desenvolver estimativas por integração numérica da Tabela 19.6. Observe como o uso desses sete pontos para caracterizar essa função que varia no contínuo perde os dois picos em $x = 2,5$ e $x = 12,5$ m.

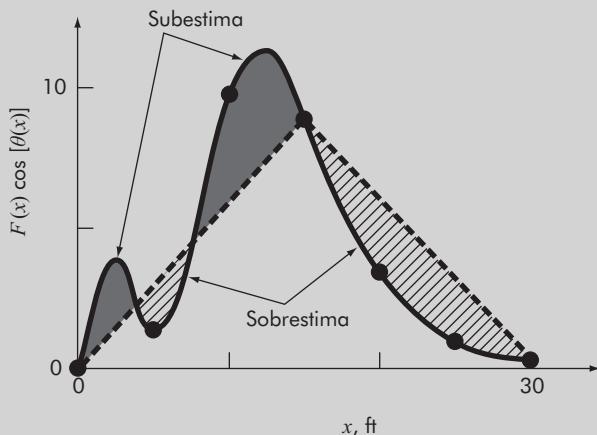


FIGURA 19.20 Descrição gráfica de por que a regra do trapézio com dois segmentos fornece uma boa estimativa da integral para esse caso particular. Por acaso, o uso de dois trapézios leva a um balanceamento dos erros positivos e negativos.

Pode-se concluir, da Figura 19.20, que um número adequado de medidas deve ser feito para calcular integrais de forma acurada. Neste caso, se os dados estivessem disponíveis em $F(2,5) \cos[\theta(2,5)] = 3,9007$ e $F(12,5) \cos[\theta(12,5)] = 11,3940$, poderíamos determinar uma estimativa melhorada da integral. Por exemplo, usando a função `trapz` do MATLAB, poderíamos calcular

```
>> x=[0 2.5 5 10 12.5 15 20 25 30];
>> y=[0 3.9007 1.5297 9.5120 11.3940 8.7025 2.8087 ...
1.0881 0.3537];
```

```
>> trapz(x,y)
```

```
ans =
132.6458
```

Incluindo esses dois pontos adicionais, obtemos uma estimativa melhorada da integral de 132,6458 ($\varepsilon_r = 2,16\%$). Logo, a inclusão de dados adicionais incorpora os picos que estavam faltando anteriormente e, como consequência, leva a melhores resultados.

►► PROBLEMAS

19.1 Derive a Equação (19.4) integrando a Equação (19.3).

19.2 Calcule a seguinte integral:

$$\int_0^4 (1 - e^{-x}) dx$$

(a) analiticamente, (b) por uma única aplicação da regra do trapézio, (c) por aplicações múltiplas da regra do trapézio, com $n = 2$ e 4 , (d) uma única aplicação da regra 1/3 de Simpson, (e) aplicação múltipla da regra 1/3 de Simpson, com $n = 4$, (f) uma única aplicação da regra 3/8 de Simpson, e (g) aplicação múltipla da regra de Simpson, com $n = 5$. Para cada estimativa numérica de (b) a (g), determine o erro relativo percentual com base em (a).

19.3 Calcule a seguinte integral:

$$\int_0^{\pi/2} (8 + 4 \cos x) dx$$

(a) analiticamente, (b) por uma única aplicação da regra do trapézio, (c) por aplicações múltiplas da regra do trapézio, com $n = 2$ e 4 , (d) uma única aplicação da regra 1/3 de Simpson, (e) aplicação múltipla da regra 1/3 de Simpson, com $n = 4$, (f) uma única aplicação da regra 3/8 de Simpson, e (g) aplicação múltipla da regra de Simpson, com $n = 5$. Para cada estimativa numérica de (b) a (g), determine o erro relativo percentual com base em (a).

19.4 Calcule a seguinte integral:

$$\int_{-2}^4 (1 - x - 4x^3 + 2x^5) dx$$

(a) analiticamente, (b) por uma única aplicação da regra do trapézio, (c) por aplicações múltiplas da regra do trapézio, com $n = 2$ e 4 , (d) uma única aplicação da regra 1/3 de Simpson, (e) regra 3/8 de Simpson, (f) regra de Boole. Para cada estimativa numérica de (b) a (f), determine o erro relativo percentual com base em (a).

19.5 A função

$$f(x) = e^{-x}$$

pode ser usada para gerar a seguinte tabela de dados desigualmente espaçados:

x	0	0,1	0,3	0,5	0,7	0,95	1,2
f(x)	1	0,9048	0,7408	0,6065	0,4966	0,3867	0,3012

Calcule a integral de $a = 0$ a $b = 0,6$, usando (a) métodos analíticos, (b) a regra do trapézio, e (c) uma combinação das regras do trapézio e de Simpson sempre que possível para obter a máxima acurácia. Para (b) e (c), calcule o erro relativo percentual.

19.6 Calcule a seguinte integral dupla:

$$\int_{-2}^2 \int_0^4 (x^2 - 3y^2 + xy^3) dx dy$$

(a) analiticamente, (b) usando a aplicação múltipla da regra do trapézio, com $n = 2$, e (c) usando aplicações únicas da regra 1/3 de Simpson. Para (b) e (c), calcule o erro relativo percentual.

19.7 Calcule a seguinte integral tripla:

$$\int_{-4}^4 \int_0^6 \int_{-1}^3 (x^3 - 2yz) dx dy dz$$

(a) analiticamente, e (b) usando aplicações únicas da regra 1/3 de Simpson. Para (b), calcule o erro relativo percentual.

19.8 Determine a distância percorrida para os seguintes dados de velocidade:

t	1	2	3,25	4,5	6	7	8	8,5	9	10
v	5	6	5,5	7	8,5	8	6	7	7	5

(a) Use a regra do trapézio; e também determine a velocidade média.

(b) Ajuste os dados com uma equação cúbica utilizando regressão polinomial. Integre o polinômio cúbico para determinar a distância.

19.9 A água exerce uma pressão na face a montante de um dique, como mostra a Figura P19.9. A pressão pode ser caracterizada por

$$p(z) = \rho g(D - z) \quad (\text{P19.9})$$

onde $p(z)$ é a pressão em pascals (ou N/m^2) exercida a uma elevação de z metros acima do fundo da represa; ρ é a densidade da água, a qual nesse problema considera-se ser constante e igual a 10^3 kg/m^3 ; g é a aceleração da gravidade ($9,8 \text{ m/s}^2$); e D é a elevação (em m) da superfície da água acima do fundo da represa. De acordo com a Equação (P19.9), a pressão aumenta linearmente com a profundidade, como mostra a Figura P19.9a. Omitindo a pressão atmosférica (pois ela age contra os dois lados do dique e essencialmente se cancela), a força total f_t pode ser determinada multiplicando-se a pressão pela área do lado do dique (como mostra a Figura 19.9b). Como tanto a pressão quanto a área variam com o nível de profundidade, a força total é obtida calculando

$$f_t = \int_0^D \rho g w(z)(D - z) dz$$

onde $w(z)$ é igual à largura do dique (m) na elevação z (Figura P19.9b). A reta de ação também pode ser obtida pelo cálculo de

$$d = \frac{\int_0^D \rho g z w(z)(D - z) dz}{\int_0^D \rho g w(z)(D - z) dz}$$

Use a regra de Simpson para determinar f_t e d .

19.10 A força exercida no mastro de um veleiro pode ser representada pela seguinte função:

$$f(z) = 200 \left(\frac{z}{5+z} \right) e^{-2z/H}$$

onde z é a elevação acima do convés e H é a altura do mastro. A força total F exercida no mastro

pode ser determinada integrando essa função ao longo da altura do mastro:

$$F = \int_0^H f(z) dz$$

A linha de ação também pode ser determinada por integração:

$$d = \frac{\int_0^H z f(z) dz}{\int_0^H f(z) dz}$$

- (a) Use a aplicação múltipla da regra do trapezio para calcular F e d para o caso em que $H = 30$ ($n = 6$).
- (b) Repita (a), mas use a aplicação múltipla da regra 1/3 de Simpson.

19.11 A força do vento distribuída contra o lado de um edifício é medida como

Altura l, m	0	30	60	90	120
Força, $F(l)$, N/m	0	340	1.200	1.550	2.700
Altura l, m	150	180	210	240	
Força, $F(l)$, N/m	3.100	3.200	3.500	3.750	

Calcule a força líquida e a linha de ação decorrentes dessa distribuição do vento.

19.12 Uma viga de 11 m está sujeita a uma carga, e a força de cisalhamento segue a equação

$$V(x) = 5 + 0,25x^2$$

onde V é a força de cisalhamento e x é a distância ao longo da viga. Sabemos que $V = dM/dx$, e M é o momento de deformação. A integração fornece a relação

$$M = M_o + \int_0^x V dx$$

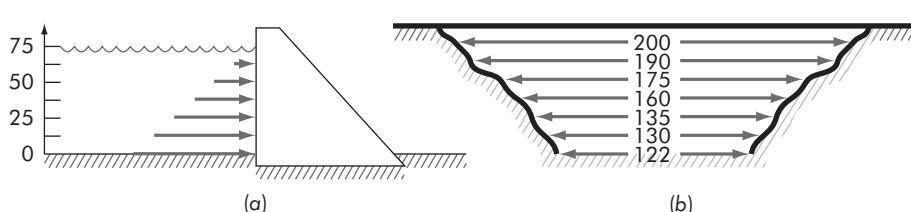


FIGURA P19.9 Água exercendo pressão na face a montante de um dique: (a) vista lateral que mostra a força linearmente crescente com a profundidade; (b) visão frontal que mostra a largura da represa em metros.

Se M_0 é 0 e $x = 11$, calcule M usando (a) integração analítica, (b) aplicação múltipla da regra do trapezio, e (c) aplicação múltipla das regras de Simpson. Para (b) e (c), use incrementos de 1 m.

19.13 A massa total de uma haste de densidade variável é dada por

$$m = \int_0^L \rho(x) A_c(x) dx$$

onde m é a massa, $\rho(x)$ é a densidade, $A_c(x)$ é a área da seção transversal, x é a distância ao longo da haste e L o comprimento total da haste. Os seguintes dados foram medidos para uma haste de 20 m. Determine a massa em gramas com a melhor precisão possível.

x, m	0	4	6	8	12	16	20
$\rho, \text{g/cm}^3$	4,00	3,95	3,89	3,80	3,60	3,41	3,30
A_c, cm^2	100	103	106	110	120	133	150

19.14 Um estudo em engenharia de transporte exige que você determine o número de carros que passa por uma interseção durante o horário de *rush hour* da manhã. Você fica em um lado da estrada e conta o número de carros que passa em 4 minutos, em diferentes instantes, conforme tabulado a seguir. Use o melhor método numérico para determinar (a) o número total de carros que passa entre 7:30 e 9:15 e (b) a taxa de carros passando pela interseção por minuto. (*Sugestão:* tenha cuidado com as unidades.)

Horário (h)	7:30	7:45	8:00	8:15	8:45	9:15
Taxa (carros em 4 min)	18	23	14	24	20	9

19.15 Determine o valor médio para os dados na Figura P19.15. Faça a integral necessária para a média na ordem mostrada na seguinte equação:

$$I = \int_{x_0}^{x_n} \left[\int_{y_0}^{y_m} f(x, y) dy \right] dx$$

19.16 A integração fornece um meio de calcular quanta massa entra em um reator, ou sai dele, em um período especificado de tempo, como em

$$M = \int_{t_1}^{t_2} Qc dt$$

onde t_1 e t_2 são os instantes inicial e final, respectivamente. Essa fórmula faz sentido do ponto de vista intuitivo se você lembrar a analogia entre integral e soma. Logo, a integral representa a somatória do produto entre a vazão e a concentração para fornecer a massa total entrando ou saindo entre t_1 e t_2 . Use integração numérica para avaliar essa equação para os dados listados a seguir.

t, min	0	10	20	30	35	40	45	50
$Q, \text{m}^3/\text{min}$	4	4,8	5,2	5,0	4,6	4,3	4,3	5,0
$c, \text{mg/m}^3$	10	35	55	52	40	37	32	34

19.17 A área da seção transversal de um canal pode ser calculada por

$$A_c = \int_0^B H(y) dy$$

onde B é a largura total do canal (m), H é a profundidade (m) e y é a distância da margem (m). De forma parecida, a vazão média Q (m^3/s) pode ser calculada como

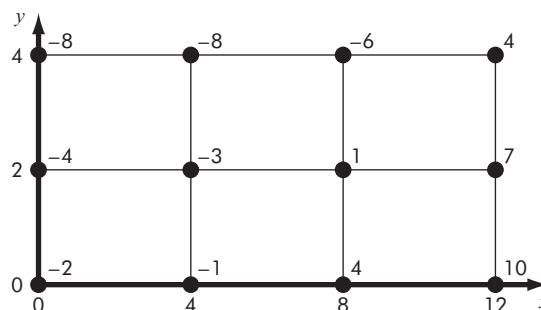


FIGURA P19.15

$$Q = \int_0^B U(y)H(y) dy$$

onde U é a velocidade da água (m/s). Use essas relações e um método numérico para determinar A_c e Q para os seguintes dados:

$y, \text{ m}$	0	2	4	5	6	9
$H, \text{ m}$	0,5	1,3	1,25	1,8	1	0,25
$U, \text{ m/s}$	0,03	0,06	0,05	0,13	0,11	0,02

19.18 A concentração média de uma substância \bar{c} (g/m^3) em um lago onde a área A_s (m^2) varia com a profundidade z (m) pode ser calculada por integração:

$$\bar{c} = \frac{\int_0^Z c(z)A_s(z) dz}{\int_0^Z A_s(z) dz}$$

onde Z é a profundidade total (m). Determine a concentração média com base nos seguintes dados:

$z, \text{ m}$	0	4	8	12	16
$A, 10^6 \text{ m}^2$	9,8175	5,1051	1,9635	0,3927	0,0000
$c, \text{ g/m}^3$	10,2	8,5	7,4	5,2	4,1

19.19 Como foi feito na Seção 19.9, determine o trabalho realizado se uma força constante de 1 N aplicada em um ângulo θ resulta nos deslocamentos a seguir. Use a função `cumtrapz` para determinar o trabalho cumulativo e trace o resultado *versus* θ .

$x, \text{ m}$	0	1	2,8	3,9	3,8	3,2	1,3
$\theta, \text{ graus}$	0	30	60	90	120	150	180

19.20 Calcule o trabalho como descrito na Seção 19.9, mas use as equações a seguir para $F(x)$ e $\theta(x)$:

$$F(x) = 1,6x - 0,045x^2$$

$$\theta(x) = -0,00055x^3 + 0,0123x^2 + 0,13x$$

A força está em Newtons e o ângulo está em radianos. Faça a integral de $x = 0$ a 30 m.

19.21 Como especificado na tabela a seguir, uma partícula esférica manufaturada tem uma densidade que varia como uma função da distância de seu centro ($r = 0$):

$r, \text{ mm}$	0	0,12	0,24	0,36	0,49
$\rho(\text{g/cm}^3)$	6	5,81	5,14	4,29	3,39
$r, \text{ mm}$	0,62	0,79	0,86	0,93	1
$\rho(\text{g/cm}^3)$	2,7	2,19	2,1	2,04	2

Use integração numérica para estimar a massa da partícula (em g) e a densidade média (em g/cm^3).

19.22 Como especificado na tabela a seguir, a densidade da Terra varia como uma função da distância de seu centro ($r = 0$):

$r, \text{ km}$	0	1100	1500	2450	3400	3630
$\rho(\text{g/cm}^3)$	13	12,4	12	11,2	9,7	5,7
$r, \text{ km}$	4500	5380	6060	6280	6380	
$\rho(\text{g/cm}^3)$	5,2	4,7	3,6	3,4	3	

Use integração numérica para estimar a massa da Terra (em toneladas métricas) e a densidade média (em g/cm^3). Desenvolva dois gráficos organizados verticalmente em uma mesma janela (utilize o comando `subplot`), um da densidade *versus* o raio (parte superior), e outro da massa *versus* o raio (parte inferior). Considere que a Terra é uma esfera perfeita.

20

Integração numérica de funções

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar métodos numéricos para integração de funções dadas. Após esta leitura completa, você será capaz de:

- ✓ Entender como a extrapolação de Richardson fornece meios para obter uma estimativa de integral mais acurada pela combinação de duas estimativas menos acuradas.
- ✓ Entender como a quadratura de Gauss fornece estimativas de integral superiores escolhendo abscissas ideais nas quais se deseja avaliar a função.
- ✓ Aprender como usar as funções nativas `quad` e `quadl` do MATLAB para integrar funções.

20.1 INTRODUÇÃO

No Capítulo 19, observamos que as funções a serem integradas numericamente terão geralmente duas formas: uma tabela de valores ou uma função (equação). A forma dos dados tem uma influência importante nas abordagens adotadas para calcular a integral. Para informações tabuladas, fica-se limitado pelo número de pontos dados. Já se uma equação estiver disponível, é possível gerar tantos valores de $f(x)$ quanto forem necessários para alcançar a precisão aceitável.

A princípio, a aplicação múltipla da regra 1/3 de Simpson pode parecer uma ferramenta razoável para esses problemas e embora seja certamente adequada para muitos deles, existem métodos mais eficientes disponíveis. Este capítulo é dedicado a três dessas técnicas, que tiram vantagem da possibilidade de gerar valores da função para desenvolver esquemas eficientes para integração numérica.

A primeira técnica é baseada na *extrapolação de Richardson*, que é um método para combinar duas estimativas numéricas para a integral a fim de obter um terceiro valor, mais acurado. O algoritmo computacional que implementa a extrapolação de Richardson de maneira altamente eficiente é chamado *integração de Romberg*. Essa técnica é recursiva e pode ser usada para gerar uma estimativa da integral dentro de uma tolerância de erro preestabelecida.

O segundo método é chamado *quadratura de Gauss*. Lembre que, no Capítulo 19, os valores de $f(x)$ para as fórmulas de Newton-Cotes foram determinados em valores especificados de x . Por exemplo, se usássemos a regra do trapézio para determinar uma integral, estaríamos restritos a tomar a média ponderada de $f(x)$ nas extremidades do intervalo. As fórmulas de quadratura de Gauss empregam valores de x que estão posicionados entre a e b , de maneira que é produzida uma estimativa da integral muito mais acurada.

A terceira abordagem é chamada de *quadratura adaptativa*. Essa técnica emprega a aplicação múltipla da regra 1/3 de Simpson a subintervalos do intervalo de integração de modo que permite o cálculo de estimativas de erro, que são, então, usadas para determinar se estimativas mais refinadas são necessárias para um subintervalo. Desse modo, uma segmentação mais refinada é usada apenas quando necessário. Duas funções nativas do MATLAB que usam quadradura adaptativa são apresentadas.

20.2 INTEGRAÇÃO DE ROMBERG

A integração de Romberg é uma técnica desenvolvida para produzir integrais numéricas eficientes de funções. É bastante parecida com as técnicas discutidas no Capítulo 19, no sentido de que é baseada em aplicações sucessivas da regra do trapézio; entretanto, mediante manipulações matemáticas, são obtidos resultados superiores com menos esforço.

20.2.1 Extrapolação de Richardson

Técnicas estão disponíveis para melhorar os resultados da integração numérica com base nas próprias estimativas da integral. Chamados em geral de *extrapolação de Richardson*, esses métodos usam duas estimativas de uma integral para calcular uma terceira aproximação mais acurada.

A estimativa e o erro associados com a aplicação múltipla da regra do trapézio podem ser representados de modo geral por

$$I = I(h) + E(h)$$

onde I é o valor exato da integral, $I(h)$ é a aproximação a partir da aplicação da regra do trapézio com n segmentos com tamanho do passo de $h = (b - a)/n$ e $E(h)$ é o erro de truncamento. Se fizermos duas estimativas separadas usando tamanhos de passo de h_1 e h_2 e tivermos os valores exatos para o erro:

$$I(h_1) + E(h_1) = I(h_2) + E(h_2) \quad (20.1)$$

Lembre-se agora de que o erro da aplicação múltipla da regra do trapézio pode ser representado aproximadamente pela Equação (19.21) [com $n = (b - a)/h$]:

$$E \cong -\frac{b-a}{12} h^2 \bar{f}'' \quad (20.2)$$

Considerando que \bar{f}'' é constante independentemente do tamanho do passo, a Equação (20.2) pode ser usada para determinar que a razão entre os dois erros será

$$\frac{E(h_1)}{E(h_2)} \cong \frac{h_1^2}{h_2^2} \quad (20.3)$$

Esse cálculo tem o efeito importante de remover o termo \bar{f}'' do cálculo. Feito isso, torna-se possível a utilização da informação embutida na Equação (20.2), sem o conhecimento prévio da segunda derivada da função. Para isso, reorganizamos a Equação (20.3), para obter

$$E(h_1) \cong E(h_2) \left(\frac{h_1}{h_2} \right)^2$$

a qual pode ser substituída na Equação (20.1):

$$I(h_1) + E(h_2) \left(\frac{h_1}{h_2} \right)^2 = I(h_2) + E(h_2)$$

que pode ser resolvida por

$$E(h_2) = \frac{I(h_1) - I(h_2)}{1 - (h_1/h_2)^2}$$

Dessa forma, desenvolvemos uma estimativa do erro de truncamento em termos das estimativas da integral e dos seus tamanhos de passo. Essa estimativa pode então ser substituída em

$$I = I(h_2) + E(h_2)$$

para fornecer uma estimativa melhorada da integral:

$$I = I(h_2) + \frac{1}{(h_1/h_2)^2 - 1} [I(h_2) - I(h_1)] \quad (20.4)$$

É possível mostrar (Ralston e Rabinowitz, 1978) que o erro nessa estimativa é $O(h^4)$. Desse modo, combinamos as estimativas de duas regras de trapézio $O(h^2)$ para obter uma nova estimativa de $O(h^4)$. Para o caso especial em que o intervalo é dividido na metade ($h_2 = h_1/2$), essa equação se torna

$$I = \frac{4}{3}I(h_2) - \frac{1}{3}I(h_1) \quad (20.5)$$

EXEMPLO 20.1 Extrapolação de Richardson

Use a extrapolação de Richardson para avaliar a integral de $f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$ de $a = 0$ a $b = 0,8$.

Solução. As aplicações únicas e múltiplas da regra do trapézio podem ser usadas para avaliar a integral:

Segmentos	h	Integral	ε_i
1	0,8	0,1728	89,5%
2	0,4	1,0688	34,9%
4	0,2	1,4848	9,5%

A extrapolação de Richardson pode ser usada para combinar esses resultados para obter estimativas melhoradas da integral. Por exemplo, as estimativas para um e dois segmentos podem ser combinadas, para produzir

$$I = \frac{4}{3}(1,0688) - \frac{1}{3}(0,1728) = 1,367467$$

O erro da integral melhorada é $E_t = 1,640533 - 1,367467 = 0,273067$ ($\varepsilon_t = 16,6\%$), que é superior às estimativas sobre as quais é baseado.

Da mesma maneira, as estimativas para dois e quatro segmentos podem ser combinadas, para fornecer

$$I = \frac{4}{3}(1,4848) - \frac{1}{3}(1,0688) = 1,623467$$

o que representa um erro de $E_t = 1,640533 - 1,623467 = 0,017067$ ($\varepsilon_t = 1,0\%$).

A Equação (20.4) fornece uma forma de combinar duas aplicações da regra do trapézio com $O(h^2)$, para calcular uma terceira estimativa com erro $O(h^4)$. Essa abordagem é um subconjunto de um método mais geral para combinar integrais e obter estimativas melhores. No caso do Exemplo 20.1, calculamos duas integrais melhoradas de $O(h^4)$, com base em três estimativas pela regra do trapézio, e que podem, por sua vez, ser combinadas para fornecer um valor ainda melhor com $O(h^6)$. Para o caso especial em que as estimativas originais pela regra do trapézio são baseadas em dividir o tamanho do passo sucessivamente por 2, a equação usada para uma acurácia $O(h^6)$ é

$$I = \frac{16}{15}I_m - \frac{1}{15}I_l \quad (20.6)$$

onde I_m e I_l são as estimativas mais e menos acuradas, respectivamente. Analogamente, dois resultados $O(h^6)$ podem ser combinados para calcular uma integral que é $O(h^8)$ usando

$$I = \frac{64}{63}I_m - \frac{1}{63}I_l \quad (20.7)$$

EXEMPLO 20.2 Correções de ordem superior

No Exemplo 20.1, usamos a extrapolação de Richardson para calcular duas estimativas da integral de $O(h^4)$. Utilize a Equação (20.6) para combinar essas estimativas para calcular uma integral com $O(h^6)$.

Solução. As duas estimativas da integral $O(h^4)$ obtidas no Exemplo 20.1 foram 1,367467 e 1,623467. Esses valores podem ser substituídos na Equação (20.6), para fornecer

$$I = \frac{16}{15}(1,623467) - \frac{1}{15}(1,367467) = 1,640533$$

que é a resposta correta até os sete algarismos significativos que são usados neste exemplo.

20.2.2 O algoritmo da integração de Romberg

Observe que os coeficientes em cada uma das equações de extrapolação [Equações (20.5), (20.6) e (20.7)] somam 1. Logo, eles representam fatores de peso que, à medida que a precisão aumenta, colocam peso relativamente maior na estimativa superior

da integral. Essas formulações podem ser expressas em uma forma geral que é adequada para a implementação computacional:

$$I_{j,k} = \frac{4^{k-1} I_{j+1,k-1} - I_{j,k-1}}{4^{k-1} - 1} \quad (20.8)$$

onde $I_{j+1,k-1}$ e $I_{j,k-1}$ são as integrais mais e menos acuradas, respectivamente, e $I_{j,k}$ é a integral melhorada. O índice k significa o nível de integração, em que $k = 1$ corresponde à estimativa original pela regra do trapézio, $k = 2$ corresponde a $O(h^4)$, $k = 3$ a $O(h^6)$, e assim por diante. O índice j é usado para distinguir entre a estimativa mais acurada ($j + 1$) e a menos acurada (j). Por exemplo, para $k = 2$ e $j = 1$, a Equação (20.8) se torna

$$I_{1,2} = \frac{4I_{2,1} - I_{1,1}}{3}$$

que é equivalente à Equação (20.5).

A forma geral representada pela Equação (20.8) é atribuída a Romberg, e sua aplicação sistemática para o cálculo de integrais é conhecida como *integração de Romberg*. A Figura 20.1 é uma descrição gráfica de uma sequência de estimativas da integral gerada usando essa abordagem. Cada matriz corresponde a uma única iteração. A primeira coluna contém os cálculos da regra do trapézio designados por $I_{j,1}$, em que $j = 1$ é para uma aplicação com um único segmento (o tamanho do passo é $b - a$), $j = 2$ é para uma aplicação com dois segmentos [o tamanho do passo é $(b - a)/2$], $j = 3$ é para uma aplicação com quatro segmentos [o tamanho do passo é $(b - a)/4$], e assim por diante. As outras colunas da matriz são geradas pela aplicação sistemática da Equação (20.8), para se obter estimativas sucessivamente melhores da integral.

Por exemplo, a primeira iteração (Figura 20.1a) envolve o cálculo de estimativas pela regra do trapézio com um e dois segmentos ($I_{1,1}$ e $I_{2,1}$). A Equação (20.8) é então usada para calcular o elemento $I_{1,2} = 1,367467$, o qual tem um erro de $O(h^4)$.

	$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
(a)	0,172800 1,068800	1,367467		
(b)	0,172800 1,068800 1,484800	1,367467 1,623467	1,640533	
(c)	0,172800 1,068800 1,484800 1,600800	1,367467 1,623467 1,639467	1,640533 1,640533	1,640533

FIGURA 20.1 Descrição gráfica da sequência de estimativas da integral gerada usando a integração de Romberg. (a) Primeira iteração. (b) Segunda iteração. (c) Terceira iteração.

Agora, devemos determinar se esse resultado é adequado para as nossas necessidades. Como em outros métodos aproximados deste livro, um critério de parada é necessário para avaliar a acurácia dos resultados. Um método que pode ser empregado para o propósito presente é

$$|\varepsilon_a| = \left| \frac{I_{1,k} - I_{2,k-1}}{I_{1,k}} \right| \times 100\% \quad (20.9)$$

onde ε_a é uma estimativa do erro percentual. Assim como foi feito previamente em outros processos iterativos, comparamos a nova estimativa com um valor anterior. Para a Equação (20.9), o valor anterior é a estimativa mais acurada do nível anterior de integração (isto é, o nível $k-1$ de integração com $j=2$). Quando a variação entre os valores antigo e novo, representados por ε_a , ficar abaixo de um critério de erro pré-especificado ε_s , os cálculos são parados. Para a Figura 20.1a, esse cálculo indica a seguinte variação percentual durante a primeira iteração:

$$|\varepsilon_a| = \left| \frac{1,367467 - 1,068800}{1,367467} \right| \times 100\% = 21,8\%$$

O objetivo da segunda iteração (Figura 20.1b) é obter a estimativa $O(h^6) - I_{1,3}$. Para isso, uma estimativa adicional da regra do trapézio, $I_{3,1} = 1,4848$, é determinada. Então, ela é combinada com $I_{2,1}$ usando a Equação (20.8), para gerar $I_{2,2} = 1,623467$. O resultado é, por sua vez, combinado com $I_{1,2}$ para fornecer $I_{1,3} = 1,640533$. A Equação (20.9) pode ser aplicada para determinar que esse resultado representa uma variação de 1,0% quando comparado com o resultado anterior, $I_{2,2}$.

A terceira iteração (Figura 20.1c) continua o processo da mesma forma. Nesse caso, é adicionada uma estimativa pela regra do trapézio à primeira coluna e, então, a Equação (20.8) é aplicada para calcular integrais sucessivamente mais acuradas ao longo da diagonal inferior. Depois de apenas três iterações, como estamos avaliando um polinômio de quinto grau, o resultado ($I_{1,4} = 1,640533$) é exato.

A integração é mais eficiente do que a regra do trapézio e as regras de Simpson discutidas no capítulo anterior. Por exemplo, para determinar a integral como mostra a Figura 20.1, a regra 1/3 de Simpson exige uma aplicação com 256 segmentos, para fornecer uma estimativa de 1,640533. Aproximações mais finas não são possíveis devido aos erros de arredondamento. Já a integração de Romberg fornece um resultado exato (até sete algarismos significativos), com base na combinação das regras do trapézio para um, dois, quatro e oito segmentos – isto é, com apenas 15 cálculos de função.

A Figura 20.2 apresenta uma função no MATLAB para integração de Romberg. Por meio de laços, esse algoritmo implementa o método de maneira eficiente. Observe que a função utiliza outra função (`trap`) para implementar a aplicação múltipla da regra do trapézio (lembre-se da Figura 19.10). A sessão do MATLAB a seguir mostra como ela pode ser usada para determinar a integral do polinômio do Exemplo 20.1:

```
>> f=@(x) 0.2+25*x-200*x.^2+675*x.^3-900*x.^4+400*x.^5;
>> romberg(f,0,0.8)
ans =
    1.6405
```

```

function [q,ea,iter]=romberg(func,a,b,es,maxit,varargin)
% romberg: Integração de Romberg
% q = romberg(func,a,b,es,maxit,p1,p2,...):
%     Integração de Romberg.
% entrada:
% func = nome da função a ser integrada
% a, b = limites de integração
% es = erro relativo desejado (padrão = 0.000001%)
% maxit = número máximo de iterações (padrão = 30)
% p1,p2,... = parâmetros adicionais usados por func
% saída:
% q = estimativa da integral
% ea = erro relativo aproximado (%)
% iter = número de iterações
if nargin<3,error('são necessários pelo menos 3 argumentos de
entrada'),end
if nargin<4|isempty(es), es=0.000001;end
if nargin<5|isempty(maxit), maxit=50;end
n = 1;
I(1,1) = trap(func,a,b,n,varargin{:});
iter = 0;
while iter<maxit
    iter = iter+1;
    n = 2^iter;
    I(iter+1,1) = trap(func,a,b,n,varargin{:});
    for k = 2:iter+1
        j = 2+iter-k;
        I(j,k) = (4^(k-1)*I(j+1,k-1)-I(j,k-1))/(4^(k-1)-1);
    end
    ea = abs((I(1,iter+1)-I(2,iter))/I(1,iter+1))*100;
    if ea<=es, break; end
end
q = I(1,iter+1);

```

FIGURA 20.2 Função do MATLAB para implementar a integração de Romberg.

20.3 QUADRATURA DE GAUSS

No Capítulo 19, estudamos o grupo de fórmulas de integração numérica ou quadratura conhecido como equações de Newton-Cotes. Uma característica dessas fórmulas (com exceção do caso especial de dados desigualmente espaçados) é que a estimativa da integral tem base em valores da função igualmente espaçados. Consequentemente, a posição dos pontos-base usados nessas equações era predeterminada ou fixa.

Por exemplo, como mostra a Figura 20.3a, a regra do trapézio é baseada em tomar a área sob uma reta que liga os valores da função nas extremidades do intervalo de integração. A fórmula que é usada para calcular essa área é

$$I \cong (b - a) \frac{f(a) + f(b)}{2} \quad (20.10)$$

onde a e b são os extremos de integração e $b - a$ é a largura do intervalo de integração. Como a regra do trapézio deve passar pelas extremidades, há casos, como a Figura 20.3a, nos quais a fórmula resulta em um erro grande.

Agora, considere que a restrição de pontos-base fixos seja removida e que haja disponibilidade para o cálculo da área sob uma reta que liga quaisquer dois pontos da curva. Posicionando sabiamente esses pontos, pode-se definir uma reta que balanceie os erros positivos e negativos. Assim, como na Figura 20.3b, chega-se a uma estimativa melhorada da integral.

Quadratura de Gauss é o nome de uma classe de técnicas que implementa essa estratégia. As fórmulas de quadratura de Gauss descritas nesta seção são chamadas de *fórmulas de Gauss-Legendre*. Antes de descrever essa abordagem, mostraremos como as fórmulas de integração numérica, tal como a regra do trapézio, podem ser deduzidas usando-se o método dos coeficientes indeterminados. Esse método será então empregado para deduzir as fórmulas de Gauss-Legendre.

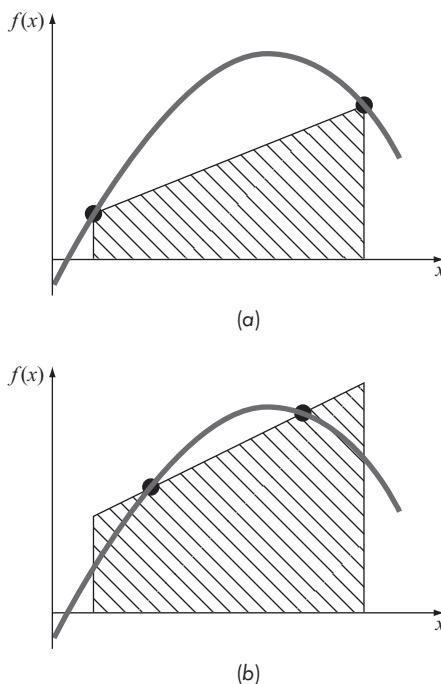


FIGURA 20.3 (a) Descrição gráfica da regra do trapézio com a área sob a reta que liga extremidades fixas. (b) Uma estimativa melhorada da integral obtida tomando-se a área sob a reta passando por dois pontos intermediários. Posicionando sabiamente esses pontos, os erros positivos e negativos são balanceados e uma estimativa melhorada da integral é obtida.

20.3.1 Método dos coeficientes indeterminados

No Capítulo 19, deduzimos a regra do trapézio integrando um polinômio interpolador linear e por raciocínio geométrico. O método dos coeficientes indeterminados oferece uma terceira abordagem, que também tem utilidade na dedução de outras técnicas de integração, como a quadratura de Gauss.

Para ilustrar essa abordagem, a Equação (20.10) é expressa como

$$I \cong c_0 f(a) + c_1 f(b) \quad (20.11)$$

onde os c 's são constantes. Perceba, agora, que a regra do trapézio deveria fornecer resultados exatos quando a função que está sendo integrada é uma constante ou uma reta. Duas equações simples que representam esses casos são $y = 1$ e $y = x$, e estão ilustradas na Figura 20.4. Logo, as seguintes igualdades devem ser válidas:

$$c_0 + c_1 = \int_{-(b-a)/2}^{(b-a)/2} 1 \, dx$$

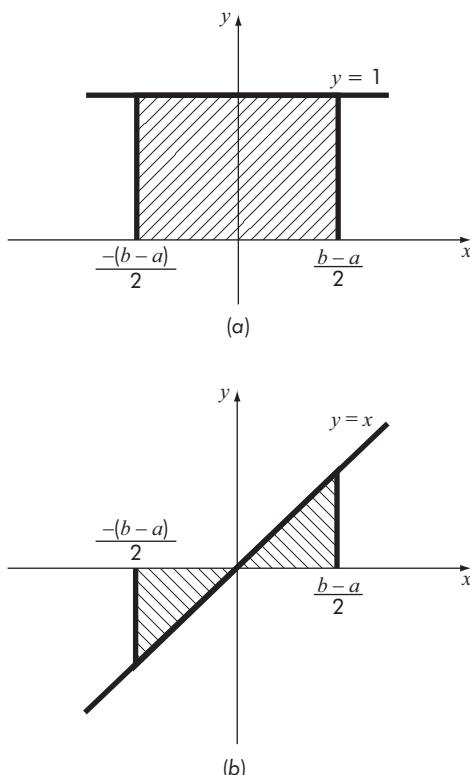


FIGURA 20.4 Duas integrais que devem ser calculadas exatamente pela regra do trapézio: (a) uma constante e (b) uma reta.

e

$$-c_0 \frac{b-a}{2} + c_1 \frac{b-a}{2} = \int_{-(b-a)/2}^{(b-a)/2} x \, dx$$

ou, calculando as integrais,

$$c_0 + c_1 = b - a$$

e

$$-c_0 \frac{b-a}{2} + c_1 \frac{b-a}{2} = 0$$

Essas são as duas equações com duas incógnitas que podem ser resolvidas por

$$c_0 = c_1 = \frac{b-a}{2}$$

o que, quando se substitui de volta na Equação (20.11), fornece

$$I = \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)$$

que é equivalente à regra do trapézio.

20.3.2 Dedução da fórmula de Gauss-Legendre de dois pontos

Do mesmo modo que a dedução da regra do trapézio, o objetivo da quadratura de Gauss é determinar os coeficientes de uma equação da forma

$$I \equiv c_0 f(x_0) + c_1 f(x_1) \quad (20.12)$$

onde os c 's são coeficientes desconhecidos. Entretanto, ao contrário da regra do trapézio, que usa extremidades fixas a e b , os argumentos da função x_0 e x_1 não são fixados nas extremidades, mas são incógnitas (Figura 20.5). Assim, agora temos um total de quatro incógnitas que devem ser calculadas, e necessitamos de quatro condições para determiná-las exatamente.

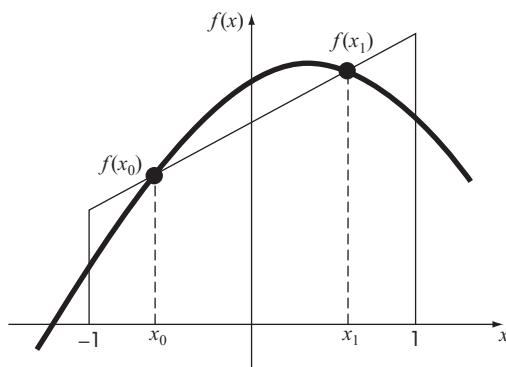


FIGURA 20.5 Descrição gráfica das variáveis desconhecidas x_0 e x_1 da integração por quadratura de Gauss.

De modo análogo à regra dos trapézios, podemos obter duas dessas condições, considerando que a Equação (20.12) calcula a integral de uma função constante e de uma função linear de forma exata. Então, para chegar a duas outras condições, simplesmente estendemos esse raciocínio admitindo que ele também calculará a integral de uma função parabólica ($y = x^2$) e de uma cúbica ($y = x^3$) de maneira exata. Fazendo isso, determinamos todas as quatro incógnitas e, ainda, deduzimos uma fórmula de integração linear de dois pontos que é exata para funções cúbicas. As quatro equações a serem resolvidas são

$$c_0 + c_1 = \int_{-1}^1 1 \, dx = 2 \quad (20.13)$$

$$c_0 x_0 + c_1 x_1 = \int_{-1}^1 x \, dx = 0 \quad (20.14)$$

$$c_0 x_0^2 + c_1 x_1^2 = \int_{-1}^1 x^2 \, dx = \frac{2}{3} \quad (20.15)$$

$$c_0 x_0^3 + c_1 x_1^3 = \int_{-1}^1 x^3 \, dx = 0 \quad (20.16)$$

As Equações (20.13) até (20.16) podem ser resolvidas simultaneamente para as quatro incógnitas. Resolva a Equação (20.14) para c_1 e substitua o resultado na Equação (20.16), que pode ser resolvida para

$$x_0^2 = x_1^2$$

Como x_0 e x_1 não podem ser iguais, isso significa que $x_0 = -x_1$. Substituindo esse resultado na Equação (20.14), obtém-se que $c_0 = c_1$. Consequentemente da Equação (20.13) segue que

$$c_0 = c_1 = 1$$

A substituição desses valores na Equação (20.15) fornece

$$\begin{aligned} x_0 &= -\frac{1}{\sqrt{3}} = -0,5773503\dots \\ x_1 &= \frac{1}{\sqrt{3}} = 0,5773503\dots \end{aligned}$$

Portanto, a fórmula de Gauss de dois pontos é

$$I = f\left(\frac{-1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \quad (20.17)$$

Logo, chega-se ao interessante resultado que uma simples adição de valores da função em $x = -1/\sqrt{3}$ e $1/\sqrt{3}$ fornece uma estimativa da integral que tem acurácia de terceira ordem.

Observe que os extremos de integração nas Equações (20.13) até (20.16) são de -1 a 1 . Isso foi feito para simplificar a matemática e tornar a formulação tão geral quanto possível. Uma simples mudança de variável pode ser usada para transformar

outros extremos de integração para essa forma, considerando que uma nova variável x_d está relacionada à variável original x de forma linear, como em

$$x = a_1 + a_2 x_d \quad (20.18)$$

Se o extremo inferior, $x = a$, corresponder a $x_d = -1$, esses valores podem ser substituídos na Equação (20.18), para fornecer

$$a = a_1 + a_2(-1) \quad (20.19)$$

Analogamente, o extremo superior, $x = b$, corresponde a $x_d = 1$, e fornece

$$b = a_1 + a_2(1) \quad (20.20)$$

As Equações (20.19) e (20.20) podem ser resolvidas simultaneamente por

$$a_1 = \frac{b + a}{2} \quad \text{e} \quad a_2 = \frac{b - a}{2} \quad (20.21)$$

o que pode ser substituído na Equação (20.22), para produzir

$$x = \frac{(b + a) + (b - a)x_d}{2} \quad (20.22)$$

Essa equação pode ser derivada para fornecer

$$dx = \frac{b - a}{2} dx_d \quad (20.23)$$

Os valores de x e dx , nas Equações (20.22) e (20.23), respectivamente, podem ser substituídos na equação a ser integrada. Essas substituições transformam efetivamente o intervalo de integração sem mudar o valor da integral. O exemplo a seguir ilustra como isso é feito na prática.

EXEMPLO 20.3 Fórmula de Gauss-Legendre de dois pontos

Use a Equação (20.17) para calcular a integral de

$$f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

entre os limites $x = 0$ a $0,8$. O valor exato da integral é 1,640533.

Solução. Antes de integrar a função, deve-se fazer uma mudança de variáveis de forma que os extremos sejam -1 a $+1$. Para isso, substitui-se $a = 0$ e $b = 0,8$ nas Equações (20.22) e (20.23), para obter

$$x = 0,4 + 0,4x_d \quad \text{e} \quad dx = 0,4dx_d$$

Ambas as equações podem ser substituídas na equação original para fornecer

$$\begin{aligned} & \int_0^{0,8} (0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5) dx \\ &= \int_{-1}^1 [0,2 + 25(0,4 + 0,4x_d) - 200(0,4 + 0,4x_d)^2 + 675(0,4 + 0,4x_d)^3 \\ &\quad - 900(0,4 + 0,4x_d)^4 + 400(0,4 + 0,4x_d)^5] 0,4dx_d \end{aligned}$$

Portanto, o lado direito está na forma adequada para o cálculo por meio da quadratura de Gauss. A função transformada pode ser calculada em $x_d = -1/\sqrt{3}$ como igual a 0,516741 e em $x_d = 1/\sqrt{3}$ como igual a 1,305837. Assim, de acordo com a Equação (20.17), a integral é $0,516741 + 1,305837 = 1,822578$, o que representa um erro relativo percentual de -11,1%. Esse resultado é comparável em módulo à aplicação da regra do trapézio com quatro segmentos ou a uma aplicação única das regras 1/3 e 3/8 de Simpson. Esse último resultado deve ser esperado, já que as regras de Simpson têm acurácia de terceira ordem. Contudo, devido à escolha inteligente dos pontos-base, a quadratura de Gauss alcança essa acurácia com base em apenas dois cálculos da função.

20.3.3 Fórmula com mais pontos

Além da fórmula de dois pontos descrita na seção anterior, versões com mais pontos podem ser desenvolvidas na forma geral

$$I \cong c_0 f(x_0) + c_1 f(x_1) + \dots + c_{n-1} f(x_{n-1}) \quad (20.24)$$

onde n é o número de pontos. Os valores para os c 's e os x 's para até (e incluindo) a fórmula de seis pontos estão resumidos na Tabela 20.1.

TABELA 20.1 Fatores de peso e argumentos da função usados nas fórmulas de Gauss-Legendre.

Pontos	Fatores de peso	Argumentos da função	Erro de truncamento
1	$c_0 = 2$	$x_0 = 0,0$	$\cong f^{(2)}(\xi)$
2	$c_0 = 1$ $c_1 = 1$	$x_0 = -1/\sqrt{3}$ $x_1 = 1/\sqrt{3}$	$\cong f^{(4)}(\xi)$
3	$c_0 = 5/9$ $c_1 = 8/9$ $c_2 = 5/9$	$x_0 = -\sqrt{3}/5$ $x_1 = 0,0$ $x_2 = \sqrt{3}/5$	$\cong f^{(6)}(\xi)$
4	$c_0 = (18 - \sqrt{30})/36$ $c_1 = (18 + \sqrt{30})/36$ $c_2 = (18 + \sqrt{30})/36$ $c_3 = (18 - \sqrt{30})/36$	$x_0 = -\sqrt{525 + 70\sqrt{30}}/35$ $x_1 = -\sqrt{525 - 70\sqrt{30}}/35$ $x_2 = \sqrt{525 - 70\sqrt{30}}/35$ $x_3 = \sqrt{525 + 70\sqrt{30}}/35$	$\cong f^{(8)}(\xi)$
5	$c_0 = (322 - 13\sqrt{70})/900$ $c_1 = (322 + 13\sqrt{70})/900$ $c_2 = 128/225$ $c_3 = (322 + 13\sqrt{70})/900$ $c_4 = (322 - 13\sqrt{70})/900$	$x_0 = -\sqrt{245 + 14\sqrt{70}}/21$ $x_1 = -\sqrt{245 - 14\sqrt{70}}/21$ $x_2 = 0,0$ $x_3 = \sqrt{245 - 14\sqrt{70}}/21$ $x_4 = \sqrt{245 + 14\sqrt{70}}/21$	$\cong f^{(10)}(\xi)$
6	$c_0 = 0,171324492379170$ $c_1 = 0,360761573048139$ $c_2 = 0,467913934572691$ $c_3 = 0,467913934572691$ $c_4 = 0,360761573048131$ $c_5 = 0,171324492379170$	$x_0 = -0,932469514203152$ $x_1 = -0,661209386466265$ $x_2 = -0,238619186083197$ $x_3 = 0,238619186083197$ $x_4 = 0,661209386466265$ $x_5 = 0,932469514203152$	$\cong f^{(12)}(\xi)$

EXEMPLO 20.4 Fórmula de Gauss-Legendre de três pontos

Use a fórmula de três pontos da Tabela 20.1 para fazer uma estimativa da integral da mesma função do Exemplo 20.3.

Solução. De acordo com a Tabela 20.1, a fórmula de três pontos é

$$I = 0,5555556f(-0,7745967) + 0,8888889f(0) + 0,5555556f(0,7745967)$$

o que é igual a

$$I = 0,2813013 + 0,8732444 + 0,4859876 = 1,640533$$

que é exato.

Como a quadratura de Gauss exige cálculos da função em pontos que não são uniformemente espaçados dentro do intervalo de integração, ela não é apropriada para os casos nos quais a função não é conhecida analiticamente. Logo, não é adequada para problemas de engenharia que lidam com dados tabulados. Entretanto, quando a função for conhecida, sua eficiência pode ser uma vantagem decisiva. Isso é particularmente verdade quando muitos cálculos de integrais devem ser feitos.

20.4 QUADRATURA ADAPTATIVA

Embora a integração de Romberg seja mais eficiente que a aplicação múltipla da regra 1/3 de Simpson, ambos os métodos usam pontos igualmente espaçados. Essa restrição não leva em consideração que algumas funções têm regiões de variações relativamente abruptas em que um espaçamento mais refinado poderia ser necessário. Assim, para alcançar uma precisão desejada, um espaçamento fino deve ser aplicado em toda parte, mesmo que ele seja necessário apenas para as regiões de variações bruscas. Os métodos de quadratura adaptativa reparam essa situação ajustando automaticamente o tamanho do passo de modo que passos pequenos sejam tomados em regiões de variações bruscas e passos maiores sejam considerados onde a função varia gradualmente.

20.4.1 Função do MATLAB: quadadapt

Os métodos de *quadratura adaptativa* levam em conta o fato de que muitas funções têm regiões de alta variabilidade juntamente com outras seções nas quais a variação é gradual. Eles fazem isso ajustando o tamanho do passo de modo que intervalos pequenos sejam usados em regiões de variações rápidas e intervalos maiores empregados onde a função varia gradualmente. Muitas dessas técnicas são baseadas na aplicação múltipla da regra 1/3 de Simpson, de maneira muito similar ao modo em que a aplicação múltipla da regra do trapézio foi usada na extrapolação de Richardson; ou seja, a regra 1/3 é aplicada em dois níveis de refinamento, e a diferença entre esses dois níveis é usada para estimar o erro de truncamento. Se o erro de truncamento é aceitável, nenhum refinamento adicional é necessário, e a estimativa da integral para o subintervalo é considerada adequada. Se a estimativa do erro é muito grande, o tamanho do passo é refinado (reduzido), e o processo é repetido até que o erro caia abaixo de níveis aceitáveis. A integral total é então calculada como a soma das estimativas das integrais para os subintervalos.

A base teórica da abordagem pode ser ilustrada para um intervalo $x = a$ até $x = b$ com um tamanho de passo de $h_1 = b - a$. Uma primeira estimativa da integral pode ser determinada com a regra 1/3 de Simpson:

$$I(h_1) = \frac{h_1}{6}[f(a) + 4f(c) + f(b)] \quad (20.25)$$

onde $c = (a + b)/2$.

Como na extrapolação de Richardson, uma estimativa mais refinada pode ser obtida dividindo pela metade o tamanho do passo, isto é, pela aplicação múltipla da regra 1/3 de Simpson com $n = 4$:

$$I(h_2) = \frac{h_2}{6}[f(a) + 4f(d) + 2f(c) + 4f(e) + f(b)] \quad (20.26)$$

onde $d = (a + c)/2$, $e = (c + b)/2$ e $h_2 = h_1/2$.

Como tanto $I(h_1)$ quanto $I(h_2)$ são estimativas da mesma integral, a diferença entre elas fornece uma medida do erro; isto é,

$$E \equiv I(h_2) - I(h_1) \quad (20.27)$$

Além disso, a estimativa e o erro associados com cada aplicação podem ser representados em geral como

$$I = I(h) + E(h) \quad (20.28)$$

onde I é o valor exato da integral, $I(h)$ é a aproximação de uma aplicação da regra 1/3 de Simpson para n segmentos com tamanho do passo de $h = (b - a)/n$ e $E(h)$ é o erro de truncamento correspondente.

Utilizando uma abordagem similar à extrapolação de Richardson, pode-se deduzir uma estimativa no erro das estimativas mais refinadas $I(h_2)$ como uma função da diferença entre as duas estimativas da integral:

$$E(h_2) = \frac{1}{15}[I(h_2) - I(h_1)] \quad (20.29)$$

O erro pode, então, ser somado a $I(h_2)$ para gerar uma estimativa ainda melhor:

$$I = I(h_2) + \frac{1}{15}[I(h_2) - I(h_1)] \quad (20.30)$$

Esse resultado é equivalente à *regra de Boole* (Tabela 19.2).

As equações desenvolvidas anteriormente podem agora ser combinadas em um algoritmo eficiente. A Figura 20.6 apresenta uma função do MATLAB baseada em um algoritmo originalmente desenvolvido por Cleve Moler (2004).

A função consiste em uma função principal `quadadapt` com uma função recursiva `quadstep` que efetivamente realiza a integração. A função principal `quadadapt` inclui como argumentos de entrada a função `f` e os limites de integração `a` e `b`. Depois de definir a tolerância, as avaliações da função necessárias para a apli-

cação inicial da regra 1/3 de Simpson [Equação (20.25)] são calculadas. Esses valores junto com os limites de integração são, então, passados para a função `quadstep`. Dentro da `quadstep`, os tamanhos do passo e os valores da função restantes são determinados e as duas estimativas da integral [Equações (20.25) e (20.26)] são calculadas.

Neste ponto, o erro é estimado como o valor absoluto da diferença entre as estimativas da integral. Dependendo do valor do erro, duas coisas podem acontecer:

1. Se o erro é menor ou igual à tolerância (`tol`), a regra de Boole é gerada; a função é encerrada e retorna o resultado.
2. Se o erro é maior que a tolerância, a função `quadstep` é chamada duas vezes para avaliar cada um dos dois subintervalos da chamada atual.

As duas chamadas recursivas no segundo passo representam a verdadeira beleza desse algoritmo. Elas continuam subdividindo até a tolerância ser atingida. Uma vez que isso ocorre, os resultados são retornados de volta ao caminho recursivo e combinados com as outras estimativas da integral ao longo do caminho. O processo termina quando a última chamada é satisfeita e a integral total é calculada e retornada à função principal.

```

function q = quadadapt(f,a,b,tol,varargin)
% Calcula a integral definida de f(x) de a a b
if nargin < 4 | isempty(tol),tol = 1.e-6;end
c = (a + b)/2;
fa = feval(f,a,varargin{:});
fc = feval(f,c,varargin{:});
fb = feval(f,b,varargin{:});
q = quadstep(f, a, b, tol, fa, fc, fb, varargin{:});
end

function q = quadstep(f,a,b,tol,fa,fc,fb,varargin)
% Subfunção recursiva usada pela quadadapt.
h = b - a; c = (a + b)/2;
fd = feval(f,(a+c)/2,varargin{:});
fe = feval(f,(c+b)/2,varargin{:});
q1 = h/6 * (fa + 4*fc + fb);
q2 = h/12 * (fa + 4*fd + 2*fc + 4*fe + fb);
if abs(q2 - q1) <= tol
    q = q2 + (q2 - q1)/15;
else
    qa = quadstep(f, a, c, tol, fa, fd, fc, varargin{:});
    qb = quadstep(f, c, b, tol, fc, fe, fb, varargin{:});
    q = qa + qb;
end
end

```

FIGURA 20.6 Função do MATLAB para implementar um algoritmo de quadratura adaptativa com base em um algoritmo originalmente desenvolvido por Cleve Moler (2004).

Deve-se ressaltar que o algoritmo na Figura 20.6 é uma versão reduzida ou simplificada da função `quad`, que é uma função nativa do MATLAB empregada para cálculo de integrais. Assim, o algoritmo da Figura 20.6 não está imune a falhas como em casos nos quais a integral não existe. No entanto, ele funciona bem para muitas aplicações e certamente serve para ilustrar como a quadratura adaptativa funciona. A seguir, tem-se uma sessão do MATLAB que mostra como a função `quadadapt` pode ser usada para determinar a integral do polinômio do Exemplo 20.1:

```
>> f=@(x) 0.2+25*x-200*x^2+675*x^3-900*x^4+400*x^5;
>> q=quadadapt(f,0,0.8)
q =
    1.6405333333333336
```

20.4.2 Funções nativas do MATLAB: `quad` e `quadl`

O MATLAB tem duas funções nativas, ambas baseadas no algoritmo desenvolvido por Gander e Gautschi (2000), para implementação da quadratura adaptativa:

- ✓ **`quad`.** Essa função usa a quadratura adaptativa de Simpson, e pode ser mais eficiente para baixa acurácia e função não suaves.
- ✓ **`quadl`.** Essa função usa a chamada *quadratura de Lobatto*, e pode ser mais eficiente para alta acurácia e funções suaves.

A sintaxe a seguir para a função `quad` é a mesma para a função `quadl`:

```
q = quad(fun, a, b, tol, trace, p1, p2, ...)
```

onde `fun` é a função a ser integrada, `a` e `b` são os limites de integração, `tol` é a tolerância de erro absoluto desejada (padrão = 10^{-6}), `trace` é uma variável que, quando definida com um valor diferente de zero, faz detalhes de cálculo adicionais serem exibidos, e `p1`, `p2`, ... são parâmetros que podem passar para `fun`. Deve-se observar que os operadores de arranjo `.*`, `./` e `.^` devem ser usados na definição de `fun`. Além disso, passa matrizes vazias para os parâmetros `tol` ou `trace` para usar os valores padrões.

EXEMPLO 20.5 Quadratura adaptativa

Use a função `quad` para integrar a seguinte função:

$$f(x) = \frac{1}{(x - q)^2 + 0,01} + \frac{1}{(x - r)^2 + 0,04} - s$$

entre os limites $x = 0$ a 1 . Observe que, para $q = 0,3$, $r = 0,9$ e $s = 6$, essa equação corresponde à função `humps` que o MATLAB usa para demonstrar algumas de suas capacidades numéricas. A função `humps` exibe tanto regiões planas (horizontais) quanto íngremes, em um intervalo relativamente curto de x ; portanto, é útil para demonstração e teste de rotinas como a `quad` e a `quadl`. Observe, ainda, que a função `humps` pode ser integrada analiticamente entre os limites dados para fornecer um valor exato de $29,85832539549867$.

Solução. Primeiro, avaliaremos a integral da maneira mais simples possível, utilizando a versão nativa de `humps` junto com a tolerância-padrão:

```
>> format long
>> quad(@humps,0,1)
ans =
    29.858326128427638
```

Assim, a solução é correta para sete algarismos significativos.

Em seguida, resolvemos o mesmo problema, mas utilizando uma tolerância menor e passando q , r e s como parâmetros. Podemos criar um arquivo-M para a função:

```
function y=minha_humps(x,q,r,s)
y = 1./((x-q).^2+0.01) + 1./((x-r).^2+0.04)-s;
```

Então, a integramos com uma tolerância de erro de 10^{-4} como em

```
>> quad(@minha_humps,0,1,1e-4,[],0.3,0.9,6)
ans =
29.858121332144918
```

Observe que, como foi utilizada uma tolerância maior, o resultado é agora acurado apenas para cinco algarismos significativos. No entanto, embora não seja aparente a partir de uma única aplicação, menos avaliações da função foram feitas e, portanto, o cálculo é executado mais rapidamente.

► 20.5 ESTUDO DE CASO VALOR MÉDIO QUADRÁTICO DA CORRENTE

Contextualização. Como isso resulta em uma transmissão eficiente de energia elétrica, a corrente em um circuito CA (circuito de corrente alternada) apresenta-se geralmente na forma de uma onda senoidal:

$$i = i_{\text{pico}} \operatorname{sen}(\omega t)$$

onde i é a corrente ($\text{A} = \text{C/s}$), i_{pico} é o valor de pico da corrente (A), ω é a frequência angular (radianos/s) e t é o tempo (s). A frequência angular é relacionada ao período T (s) por $\omega = 2\pi/T$.

A potência gerada está relacionada ao valor da corrente. Integração pode ser usada para determinar a corrente média em um ciclo:

$$\bar{i} = \frac{1}{T} \int_0^T i_{\text{pico}} \operatorname{sen}(\omega t) dt = \frac{i_{\text{pico}}}{T} (-\cos(2\pi) + \cos(0)) = 0$$

Apesar de a média ser igual a zero, essa corrente é capaz de gerar potência; portanto, uma alternativa à corrente média deve ser deduzida.

Para isso, os engenheiros eletricistas determinam o valor médio quadrático da corrente, i_{rms} (A)*, que é calculado como

$$i_{\text{rms}} = \sqrt{\frac{1}{T} \int_0^T i_{\text{pico}}^2 \operatorname{sen}^2(\omega t) dt} = \frac{i_{\text{pico}}}{\sqrt{2}} \quad (20.31)$$

Assim, como o nome sugere, o valor rms da corrente é a raiz quadrada do valor médio da corrente ao quadrado. Como $1/\sqrt{2} = 0,70707$, i_{rms} é igual a cerca de 70% do valor de pico da corrente, considerando a forma de onda senoidal assumida.

Essa grandeza tem significado porque ela está diretamente relacionada à potência média absorvida por um elemento em um circuito CA. Para entender isso, lembre-se de

* N. de T.: rms, do inglês *root mean square*.



que a *lei de Joule* estabelece que a potência instantânea absorvida por um elemento de circuito é igual ao produto da tensão através do elemento pela corrente que circula por ele:

$$P = iV \quad (20.32)$$

onde P é a potência ($\text{W} = \text{J/s}$) e V é a tensão ($\text{V} = \text{J/C}$). Para um resistor, a *lei de Ohm* estabelece que a tensão é diretamente proporcional à corrente:

$$V = iR \quad (20.33)$$

onde R é a resistência ($\Omega = \text{V/A} = \text{J} \cdot \text{s/C}^2$). Substituindo a Equação (20.33) na Equação (20.32), temos

$$P = i^2 R \quad (20.34)$$

A potência média pode ser determinada integrando a Equação (20.34) ao longo de um período com o resultado:

$$\bar{P} = i_{\text{rms}}^2 R$$

Assim, o circuito CA gera a potência equivalente a um circuito CC (circuito de corrente contínua) com uma corrente constante de i_{rms} e mesma resistência R .

Agora, embora a senoide simples seja amplamente empregada, não é de modo algum a única forma de onda que é usada. Para algumas dessas formas, tais como ondas triangulares ou quadradas, o valor de i_{rms} pode ser calculado analiticamente com integração na forma fechada. No entanto, algumas formas de onda devem ser analisadas com métodos de integração numérica.

Neste estudo de caso, calcularemos o valor médio quadrático de uma forma de onda de corrente não senoidal. Usaremos tanto as fórmulas de Newton-Cotes do Capítulo 19 como as abordagens descritas neste capítulo.

Solução. A integral que deve ser calculada é

$$i_{\text{rms}}^2 = \int_0^{1/2} (10e^{-t} \sin 2\pi t)^2 dt \quad (20.35)$$

Para propósitos comparativos, o valor exato dessa integral para 15 dígitos significativos é 15,41260804810169.

Estimativas da integral para várias aplicações da regra do trapézio e 1/3 de Simpson são apresentadas na Tabela 20.2. Observe que a regra de Simpson é mais acurada do que a regra do trapézio. O valor para a integral para sete dígitos significativos é obtido usando uma regra do trapézio com 128 segmentos ou uma regra de Simpson com 32 segmentos.

A função do MATLAB desenvolvida na Figura 20.2 pode ser usada para calcular a integral com a integração de Romberg:

```
>> format long
>> i2=@(t) (10*exp(-t).*sin(2*pi*t)).^2;
>> [q,ea,iter]=romberg(i2,0,.5)
q =
    15.41260804288977
```



```

ea =
1.480058787326946e-008
iter =
5

```

TABELA 20.2 Valores para a integral calculados com as fórmulas de Newton-Cotes.

Técnica	Segmentos	Integral	ϵ_t (%)
Regra do trapézio	1	0,0	100,0000
	2	15,163266493	1,6178
	4	15,401429095	0,0725
	8	15,411958360	$4,22 \times 10^{-3}$
	16	15,412568151	$2,59 \times 10^{-4}$
	32	15,412605565	$1,61 \times 10^{-5}$
	64	15,412607893	$1,01 \times 10^{-6}$
	128	15,412608038	$6,28 \times 10^{-8}$
Regra 1/3 de Simpson	2	20,217688657	31,1763
	4	15,480816629	0,4426
	8	15,415468115	0,0186
	16	15,412771415	$1,06 \times 10^{-3}$
	32	15,412618037	$6,48 \times 10^{-5}$

Assim, com o critério de parada padrão de $es = 1 \times 10^{-6}$, obtemos um resultado que é correto até nove algarismos significativos em cinco iterações. Podemos obter um resultado ainda melhor se utilizarmos um critério de para mais rigoroso:

```

>> [q,ea,iter]=romberg(i2,0,.5,1e-15)
q =
15.41260804810169
ea =
0
iter =
7

```

A quadratura de Gauss também pode ser usada para fazer a mesma estimativa. A princípio, uma mudança na variável é realizada pela aplicação das Equações (20.22) e (20.23), para fornecer

$$t = \frac{1}{4} + \frac{1}{4} t_d \quad dt = \frac{1}{4} dt_d$$

Essas relações podem então ser substituídas na Equação (20.35), para produzir

$$i_{rms}^2 = \int_{-1}^1 \left[10e^{-(0.25+0.25t_d)} \sin 2\pi(0.25 + 0.25t_d) \right]^2 0.25 dt \quad (20.36)$$

Para fórmula de Gauss-Legendre de dois pontos, essa função é calculada em $t_d = -1/\sqrt{3}$ e $1/\sqrt{3}$, com o resultado sendo 7,684096 e 4,313728, respectivamente. Esses valores podem ser substituídos na Equação (20.17), para fornecer uma estimativa da integral de 11,99782, que representa um erro de $\epsilon_t = 22,1\%$.



A fórmula de três pontos é (Tabela 20.1)

$$I = 0,5555556(1,237449) + 0,8888889(15,16327) + 0,5555556(2,684915) = 15,65755$$

que tem $\varepsilon_r = 1,6\%$. Os resultados do uso de fórmulas de mais pontos estão resumidos na Tabela 20.3.

Por fim, a integral pode ser calculada com as funções nativas do MATLAB `quad` e `quadl`:

```
>> irms2=quad(i2,0,.5)
irms2 =
15.41260804934509
```

TABELA 20.3 Resultados do uso de fórmulas da quadratura de Gauss de vários pontos para aproximar a integral.

Pontos	Estimativa	ε_r (%)
2	11,9978243	22,1
3	15,6575502	1,59
4	15,4058023	$4,42 \times 10^{-2}$
5	15,4126391	$2,01 \times 10^{-4}$
6	15,4126109	$1,82 \times 10^{-5}$

```
>> irms2=quadl(i2,0,.5)
irms2 =
15.41260804809967
```

Ambos os resultados são muito acurados, e o resultado obtido com a função `quadl` é um pouco melhor.

Podemos agora calcular o valor de i_{rms} simplesmente tomando a raiz quadrada da integral. Por exemplo, usando o resultado calculado com a função `quadl`, obtemos

```
>> irms=sqrt(irms2)
irms =
3.92588945948554
```

Esse resultado pode então ser empregado para orientar outros aspectos do projeto e da operação do circuito tal como cálculos de dissipação de potência.

Como foi feito para a senoide simples na Equação (20.31), um cálculo interessante envolve a comparação desse resultado com o valor de pico da corrente. Reconhecendo que isso é um problema de otimização, podemos imediatamente empregar a função `fminbnd` para determinar esse valor. Como estamos procurando um máximo, avaliamos o negativo da função:

```
>> [tmax,imax]=fminbnd(@(t) -10*exp(-t).*sin(2*pi*t),0,.5)
tmax =
0.22487940319321
imax =
-7.88685387393258
```

Uma corrente máxima de 7,88685 ocorre em $t = 0,2249$ s; portanto, para essa forma de onda particular, o valor médio quadrático é cerca de 49,8% do valor máximo.

►►PROBLEMAS

20.1 Use integração de Romberg para calcular

$$I = \int_1^2 \left(x + \frac{1}{x} \right)^2 dx$$

para uma acurácia de $\varepsilon_s = 0,5\%$. Seus resultados devem ser apresentados no formato da Figura 20.1. Utilize a solução analítica da integral para determinar o erro relativo percentual do resultado obtido com integração de Romberg. Verifique que ε_t é menor do que ε_s .

20.2 Calcule a integral a seguir (a) analiticamente, (b) usando integração de Romberg ($\varepsilon_s = 0,5\%$), (c) usando a fórmula da quadratura de Gauss de três pontos, e (d) usando a função quad do MATLAB:

$$I = \int_0^8 -0,055x^4 + 0,86x^3 - 4,2x^2 + 6,3x + 2 dx$$

20.3 Calcule a integral seguinte com (a) a integração de Romberg ($\varepsilon_s = 0,5\%$), (b) a fórmula da quadratura de Gauss de dois pontos, e (c) as funções quad e quadl do MATLAB:

$$I = \int_0^3 xe^{2x} dx$$

20.4 Não há solução na forma fechada para a função erro

$$\text{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a e^{-x^2} dx$$

Use as fórmulas de Gauss Legendre de (a) dois pontos e (b) três pontos para estimar $\text{erf}(1,5)$. Determine o erro relativo percentual para cada caso com base no valor verdadeiro, que pode ser determinado com a função nativa erf do MATLAB.

20.5 A força no mastro de um veleiro pode ser representada por

$$F = \int_0^H 200 \left(\frac{z}{5+z} \right) e^{-2z/H} dz$$

onde z é a elevação acima do convés e H é a altura do mastro. Calcule F para o caso em que $H=30$ usando (a) integração de Romberg para uma tolerância de $\varepsilon_s = 0,5\%$, (b) a fórmula de Gauss-Legendre de dois pontos, e (c) a função quad do MATLAB.

20.6 O valor médio quadrático de uma corrente pode ser calculado como

$$I_{\text{rms}} = \sqrt{\frac{1}{T} \int_0^T i^2(t) dt}$$

Para $T=1$, considere que $i(t)$ é definida como

$$i(t) = 8e^{-t/T} \sin\left(2\pi \frac{t}{T}\right) \quad \text{para } 0 \leq t \leq T/2$$

$$i(t) = 0 \quad \text{para } T/2 \leq t \leq T$$

Calcule o valor de I_{rms} usando (a) integração de Romberg para uma tolerância de 0,1%, (b) fórmulas de Gauss-Legendre de dois e três pontos, e (c) a função quad do MATLAB.

20.7 O calor necessário, ΔH (cal), para induzir uma variação de temperatura, ΔT (°C), de um material pode ser calculado como

$$\Delta H = mC_p(T)\Delta T$$

onde m é a massa (g) e $C_p(T)$ é o calor específico [cal/(g · °C)]. O calor específico aumenta com a temperatura, T (°C), de acordo com

$$C_p(T) = 0,132 + 1,56 \times 10^{-4}T + 2,64 \times 10^{-7}T^2$$

Escreva um programa que use a função quad para gerar um gráfico de ΔH versus ΔT para casos em que $m=1$ kg, a temperatura inicial é -100 °C e ΔT varia de 0 a 300 °C.

20.8 A quantidade de massa transportada por um tubo durante um período de tempo pode ser calculada por

$$M = \int_{t_1}^{t_2} Q(t)c(t) dt$$

onde M é a massa (mg), t_1 é o instante inicial (min), t_2 é o instante final (min), $Q(t)$ é a vazão (m^3/min) e $c(t)$ é a concentração (mg/m^3). As seguintes representações funcionais definem a vazão temporal da vazão e da concentração:

$$Q(t) = 9 + 5 \cos^2(0,4t)$$

$$c(t) = 5e^{-0,5t} + 2e^{0,15t}$$

Determine a massa transportada entre $t_1 = 2$ e $t_2 = 8$ min com (a) a integração de Romberg para uma tolerância de 0,1% e (b) a função quad do MATLAB.

20.9 Calcule a integral dupla

$$\int_{-2}^2 \int_0^4 (x^2 - 3y^2 + xy^3) dx dy$$

(a) analiticamente e (b) usando a função dblquad do MATLAB. Use o comando help para entender como implementar a função.

20.10 Calcule o trabalho como descrito na Seção 19.9, mas use as seguintes equações para $F(x)$ e $\theta(x)$:

$$F(x) = 1,6x - 0,045x^2$$

$$\theta(x) = -0,00055x^3 + 0,0123x^2 + 0,13x$$

A força está em newtons e o ângulo está em radianos. Efetue a integração de $x = 0$ a 30 m.

20.11 Efetue os mesmos cálculos como na Seção 20.5, mas para a corrente especificada por

$$i(t) = 6e^{-1,25t} \operatorname{sen} 2\pi t \quad \text{para } 0 \leq t \leq T/2$$

$$i(t) = 0 \quad \text{para } T/2 < t \leq T$$

onde $T = 1$ s.

20.12 Calcule a potência absorvida por um elemento em um circuito como descrito na Seção 20.5, mas para uma corrente senoidal simples $i = \operatorname{sen}(2\pi t/T)$ onde $T = 1$ s.

- (a) Considere que a lei de Ohm é válida e $R = 5 \Omega$.
- (b) Admita que a lei de Ohm não é válida e que a tensão e a corrente estão relacionadas pela seguinte relação não linear: $V = (5i - 1,25i^2)$.

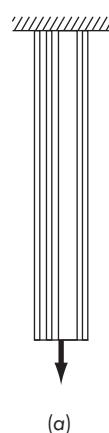
20.13 Considere que a corrente por um resistor é descrita pela função

$$i(t) = (60 - t)^2 + (60 - t) \operatorname{sen}(\sqrt{t})$$

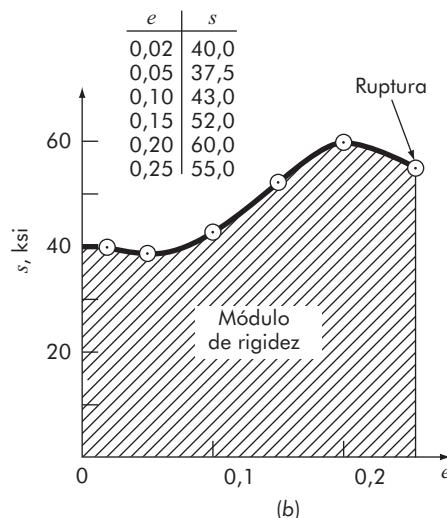
e a resistência é uma função da corrente:

$$R = 10i + 2i^{2/3}$$

Calcule a tensão média ao longo de $t = 0$ a 60 usando a regra 1/3 de Simpson composta.



(a)



$$V(t) = \frac{1}{C} \int_0^t i(t) dt$$

Use o MATLAB para ajustar esses dados com um polinômio de grau cinco. Então, use uma função de integração numérica com um valor de $C = 10^{-5}$ farad para gerar um gráfico de tensão *versus* tempo.

t, s	0	0,2	0,4	0,6
$i, 10^{-3} \text{ A}$	0,2	0,3683	0,3819	0,2282
t, s	0,8	1	1,2	
$i, 10^{-3} \text{ A}$	0,0486	0,0082	0,1441	

20.15 O trabalho realizado por um corpo é igual à força vezes a distância deslocada no sentido da força. A velocidade de um corpo no sentido de uma força é dada por

$$v = 4t \quad 0 \leq t \leq 5$$

$$v = 20 + (5 - t)^2 \quad 5 \leq t \leq 15$$

onde v está em m/s. Determine o trabalho se uma força constante de 200 N é aplicada para todo t .

20.16 Uma haste sujeita a uma carga axial (Figura P20.16a) será deformada, como mostra a curva de tensão-deformação na Figura P20.16b. A área

FIGURA P20.16 (a) Uma haste sob carga axial e (b) a curva de tensão-deformação, em que a tensão está em kips por polegada quadrada (10^3 lb/pol^2) e a deformação é adimensional.

sob a curva da tensão zero até o ponto de ruptura é chamada de *módulo de rigidez* do material, e fornece uma medida da energia por unidade de volume necessária para causar a ruptura do material; assim, é representativo da habilidade do material de suportar uma carga de impacto. Use integração numérica para calcular o módulo de rigidez para a curva de tensão-deformação vista na Figura P20.16b.

20.17 Se a distribuição de velocidade de um fluido escoando por tubo for conhecida (Figura P20.17), a vazão Q (isto é, o volume de água passando pelo tubo por unidade de tempo) pode ser calculada por $Q = \int v dA$, onde v é a velocidade e A é a área da seção transversal do tubo. (Para perceber o significado dessa relação fisicamente, lembre-se da conexão íntima entre soma e integração.) Para um tubo circular, $A = \pi r^2$ e $dA = 2\pi r dr$. Portanto,

$$Q = \int_0^r v(2\pi r) dr$$

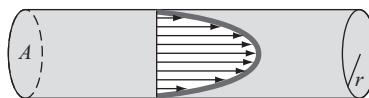


FIGURA P20.17

onde r é a distância radial medida a partir do centro do tubo. Se a distribuição de velocidade é dada por

$$v = 2 \left(1 - \frac{r}{r_0}\right)^{1/6}$$

onde r_0 é o raio total (nesse caso, 3 cm), calcule Q usando a regra do trapézio com aplicações múltiplas. Discuta os resultados.

20.18 Usando os dados seguintes, calcule o trabalho feito ao se esticar uma mola, que tem uma constante de mola $k = 300 \text{ N/m}$, até $x = 0,35 \text{ m}$. Para isso, ajuste os dados com um polinômio e, então, integre o polinômio numericamente para calcular o trabalho:

$F, 10^3 \cdot \text{N}$	0	0,01	0,028	0,046
x, m	0	0,05	0,10	0,15
$F, 10^3 \cdot \text{N}$	0,063	0,082	0,11	0,13
x, m	0,20	0,25	0,30	0,35

20.19 Calcule a distância vertical percorrida por um foguete se a velocidade vertical é dada por

$$v = 11t^2 - 5t \quad 0 \leq t \leq 10$$

$$v = 1100 - 5t \quad 10 \leq t \leq 20$$

$$v = 50t + 2(t - 20)^2 \quad 20 \leq t \leq 30$$

20.20 A velocidade para cima de um foguete pode ser calculada pela seguinte fórmula:

$$v = u \ln \left(\frac{m_0}{m_0 - qt} \right) - gt$$

onde v é a velocidade para cima, u é a velocidade relativa ao foguete na qual o fluido é expelido, m_0 é a massa inicial do foguete no instante $t = 0$, q é a taxa de consumo de combustível e g é a aceleração para baixo da gravidade (considerada constante e igual a $9,81 \text{ m/s}^2$). Se $u = 1.850 \text{ m/s}$, $m_0 = 160.000 \text{ kg}$ e $q = 2.500 \text{ kg/s}$, determine quanto alto o foguete irá voar em 30 s.

20.21 A distribuição normal é definida como

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

- (a) Use o MATLAB para integrar essa função de $x = -1$ a 1 e de -2 a 2.
- (b) Utilize o MATLAB para determinar os pontos de inflexão dessa função.

20.22 Use a integração de Romberg para calcular

$$\int_0^2 \frac{e^x \sin x}{1+x^2} dx$$

para uma acurácia de $\epsilon_s = 0,5\%$. Seus resultados devem ser apresentados na forma da Figura 20.1.

20.23 Lembre-se de que a velocidade do saltador de *bungee jumping* em queda livre pode ser calculada analiticamente como [Equação (1.9)]:

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh \left(\sqrt{\frac{gc_d}{m}} t \right)$$

onde $v(t)$ é a velocidade (m/s), t é o tempo (s), $g = 9,81 \text{ m/s}^2$, m é a massa (kg) e c_d é o coeficiente de arraste (kg/m).

- (a) Use a integração de Romberg para calcular o quanto longe o saltador viaja durante os 8 primeiros segundos de queda livre, dados $m = 80 \text{ kg}$ e $c_d = 0,2 \text{ kg/m}$. Calcule sua resposta para $\epsilon_s = 1\%$.

- (b) Efetue o mesmo cálculo com a função `quad` do MATLAB.

20.24 Prove que a Equação (20.30) é equivalente à lei de Boole.

20.25 Como especificado na tabela a seguir, a densidade da Terra varia como uma função da distância a partir de seu centro ($r = 0$):

$r, \text{ km}$	0	1.100	1.500	2.450	3.400	3.630	4.500
$\rho, \text{ g/cm}^3$	13	12,4	12	11,2	9,7	5,7	5,2
$r, \text{ km}$	5.380	6.060	6.280	6.380			
$\rho, \text{ g/cm}^3$	4,7	3,6	3,4	3			

Desenvolva um programa para ajustar esses dados com a função `interp1` usando a opção `pchip`. Gere um gráfico que mostre o ajuste resultante junto com os pontos dados. Então, use uma das

funções de integração do MATLAB para estimar a massa da terra (em toneladas métricas) pela integração da saída da função `interp1`.

20.26 Desenvolva uma função no MATLAB para implementar a integração de Romberg com base na Figura 20.2. Teste a função usando-a para determinar a integral do polinômio do Exemplo 20.1. Então, use-a para resolver o Problema 20.1.

20.27 Desenvolva uma função no MATLAB para implementar a quadratura adaptativa com base na Figura 20.6. Teste a função usando-a para determinar a integral do polinômio do Exemplo 20.1. Então, use-a para resolver o Problema 20.20.

21

Derivação numérica

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar a derivação numérica. Após esta leitura completa, você será capaz de:

- ✓ Compreender a aplicação de fórmulas de derivação numérica de alta acurácia para dados igualmente espaçados.
- ✓ Aprender como calcular derivadas para dados desigualmente espaçados.
- ✓ Entender como a extrapolação de Richardson é aplicada para derivação numérica.
- ✓ Reconhecer a sensibilidade da derivação numérica para dados com erros.
- ✓ Aprender como calcular derivadas no MATLAB com as funções `diff` e `gradient`.
- ✓ Saber como gerar gráficos de curvas de nível e campos vetoriais com o MATLAB.

VOCÊ TEM UM PROBLEMA

Lembre-se de que a velocidade de um saltador de *bungee jumping* em queda livre como uma função do tempo pode ser calculada como

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}}t\right) \quad (21.1)$$

No início do Capítulo 19, usamos o cálculo para integrar essa equação para determinar a distância vertical z que o saltador caiu após um tempo t .

$$z(t) = \frac{m}{c_d} \ln \left[\cosh \left(\sqrt{\frac{gc_d}{m}}t \right) \right] \quad (21.2)$$

Agora, considere que lhe foi proposto o problema reverso, ou seja, foi solicitada a determinação da velocidade com base na posição do saltador como uma função do tempo. Como a derivação é a operação inversa da integração, a primeira pode ser usada para efetuar o cálculo solicitado:

$$v(t) = \frac{dz(t)}{dt} \quad (21.3)$$

Substituindo a Equação (21.2) na Equação (21.3) e efetuando a derivação, obtemos de volta a Equação (21.1).

Além da velocidade, poderia ser também solicitado o cálculo da aceleração do saltador. Para isso, pode-se tomar a primeira derivada da velocidade, ou a segunda derivada do deslocamento:

$$a(t) = \frac{dv(t)}{dt} = \frac{d^2z(t)}{dt^2} \quad (21.4)$$

Em ambos os casos, o resultado seria

$$a(t) = g \operatorname{sech}^2 \left(\sqrt{\frac{gc_d}{m}} t \right) \quad (21.5)$$

Embora uma solução na forma fechada possa ser desenvolvida para esse caso, existem outras funções que podem ser difíceis, ou mesmo impossíveis, de derivar analiticamente. Além disso, considere que haja alguma maneira de medir a velocidade do saltador em vários instantes de tempo durante a queda. Essas velocidades e os instantes de tempo correspondentes podem ser organizados como uma tabela de valores discretos. Nessa situação, é útil derivar os dados discretos para determinar a velocidade e a aceleração. Nos dois casos, métodos de derivação numérica estão disponíveis para obter soluções. Este capítulo introduzirá alguns desses métodos.

21.1 INTRODUÇÃO E CONTEXTUALIZAÇÃO

21.1.1 O que é diferenciação?

O *cálculo* é a matemática da variação. Como os engenheiros e os cientistas devem lidar continuamente com sistemas e processos que variam, o cálculo é uma ferramenta essencial de sua profissão. No cerne do cálculo está o conceito matemático de derivação (ou diferenciação).

De acordo com o dicionário, *diferenciar* significa “marcar diferenças; distinguir; perceber a diferença entre”. Matematicamente, a *derivada*, que serve como veículo fundamental da derivação, representa a taxa de variação de uma variável dependente com relação a uma variável independente. Como mostra a Figura 21.1, a definição matemática da derivada começa com uma aproximação por diferenças:

$$\frac{\Delta y}{\Delta x} = \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} \quad (21.6)$$

onde y e $f(x)$ são representações alternativas para a variável dependente e x é a variável independente. Se for permitido que Δx se aproxime de zero, como ocorre ao nos movermos da Figura 21.1a para c, a diferença se torna uma derivada:

$$\frac{dy}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x} \quad (21.7)$$

onde dy/dx [que também pode ser designada por y' ou $f'(x_i)$ ¹] é a primeira derivada de y com relação a x , calculada em x_i . Como visto na descrição visual da Figura 21.1c, a derivada é a inclinação da reta tangente à curva em x_i .

¹ A forma dy/dx foi concebida por Leibnitz, enquanto a forma y' é atribuída a Lagrange. Observe que Newton utilizou a chamada notação de ponto: \dot{y} . Hoje em dia, a notação de ponto é geralmente empregada para derivadas temporais.

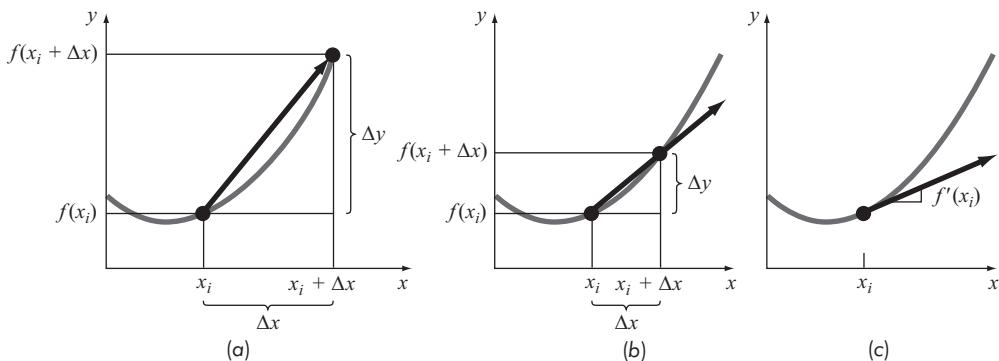


FIGURA 21.1 A definição gráfica de uma derivada: conforme Δx se aproxima de zero ao passar de (a) para (c), a aproximação por diferença se torna uma derivada.

A segunda derivada representa a derivada da primeira derivada,

$$\frac{d^2y}{dx^2} = \frac{d}{dx} \left(\frac{dy}{dx} \right) \quad (21.8)$$

Assim, a segunda derivada mostra o quanto rápido a inclinação está mudando. Ela é geralmente referida como a *curvatura*, pois um valor elevado para a segunda derivada significa uma curvatura elevada.

Por fim, derivadas parciais são usadas para funções que dependem de mais do que uma variável. Podemos visualizar as derivadas parciais como o cálculo da derivada da função em um ponto, mantendo todas as variáveis constantes, exceto aquela em relação à qual a derivada está sendo calculada. Por exemplo, dada uma função f que dependa tanto de x como de y , a derivada parcial de f com relação a x em um ponto arbitrário (x, y) é definida como

$$\frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \quad (21.9)$$

De modo similar, a derivada parcial de f com relação a y é definida como

$$\frac{\partial f}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \quad (21.10)$$

A fim de obter uma compreensão intuitiva das derivadas parciais, deve-se reconhecer que uma função que dependa de duas variáveis é uma superfície em vez de uma curva. Considere que se esteja escalando uma montanha e tem-se acesso a uma função f que fornece a elevação como uma função da longitude (o eixo x orientado no sentido leste-oeste) e da latitude (o eixo y orientado no sentido norte-sul). Se for feita uma parada em um ponto particular (x_0, y_0) , a inclinação para o leste é $\partial f(x_0, y_0)/\partial x$, e a inclinação para o norte é $\partial f(x_0, y_0)/\partial y$.

21.1.2 Derivação na engenharia e na ciência

A derivação de uma função tem tantas aplicações na engenharia e na ciência, que são necessários cursos de cálculo diferencial logo no primeiro ano da faculdade.

Muitos exemplos específicos dessas aplicações podem ser dados em todos os campos da engenharia e da ciência. A derivação é rotineira na engenharia e na ciência, porque muito do trabalho nessas áreas envolve a caracterização da mudança das variáveis no tempo e no espaço. De fato, muitas leis e outras generalizações que desempenham um papel importante nesse campo são baseadas na forma previsível pelas quais as variações se manifestam no mundo físico. Um exemplo relevante é a segunda lei de Newton, que não é enunciada em termos da posição de um objeto, mas, em vez disso, em termos da variação da posição com relação ao tempo.

Além desses exemplos temporais, diversas leis que governam o comportamento espacial das variáveis são expressas em termos de derivadas. Entre as mais comuns estão as *leis constitutivas* que definem como potenciais ou gradientes influenciam processos físicos. Por exemplo, a *lei de Fourier de condução de calor* quantifica a observação de que o calor flui de regiões de alta temperatura para regiões de baixa temperatura. No caso unidimensional, a lei pode ser expressa matematicamente como

$$q = -k \frac{dT}{dx} \quad (21.11)$$

onde $q(x)$ é o fluxo de calor (W/m^2), k é o coeficiente de condutividade térmica [$\text{W}/(\text{m} \cdot \text{K})$], T é a temperatura (K) e x é a distância (m). Assim, a derivada, ou o *gradiente*, fornece uma medida da intensidade da variação da temperatura no espaço, que governa a transferência de calor (Figura 21.2).

Leis similares fornecem modelos práticos em muitas outras áreas da engenharia e da ciência, incluindo a modelagem de mecânica dos fluidos, transferência de massa, cinética de reações químicas, eletricidade e mecânica dos sólidos (Tabela 21.1). A habilidade de obter estimativas acuradas para as derivadas é uma faceta que tem influência importante na capacidade de trabalhar de maneira eficiente nessas áreas.

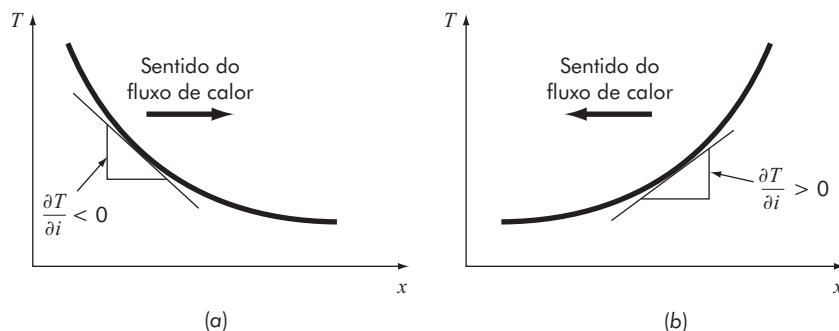


FIGURA 21.2 Descrição gráfica de um gradiente de temperatura. Como o calor se move “ladeira abaixo” da região de alta para região de baixa temperatura, o fluxo em (a) é da esquerda para direita. No entanto, devido à orientação das coordenadas cartesianas, a inclinação é negativa para esse caso. Assim, um gradiente negativo conduz a um fluxo positivo. Essa é a origem do sinal de menos na lei de Fourier de condução de calor. O caso inverso é mostrado em (b), no qual o gradiente positivo conduz a um fluxo negativo de calor da direita para a esquerda.

TABELA 21.1 As formas unidimensionais de algumas leis constitutivas frequentemente usadas na engenharia e na ciência

Lei	Equação	Área da física	Gradiente	Fluxo	Proporcionalidade
Lei de Fourier	$q = -k \frac{dT}{dx}$	Condução de calor	Temperatura	Fluxo de calor	Condutividade térmica
Lei de Fick	$J = -D \frac{dc}{dx}$	Difusão de massa	Concentração	Fluxo de massa	Difusividade
Lei de Darcy	$q = -k \frac{dh}{dx}$	Fluxo através de meios porosos	Carga hidráulica	Fluxo de água	Condutividade hidráulica
Lei de Ohm	$J = -\sigma \frac{dV}{dx}$	Eletricidade	Tensão	Fluxo de corrente	Condutividade elétrica
Lei de Viscosidade de Newton	$\tau = \mu \frac{du}{dx}$	Fluidos	Velocidade	Tensão de cisalhamento	Viscosidade dinâmica
Lei de Hooke	$\sigma = E \frac{\Delta L}{L}$	Elasticidade	Deformação	Tensão	Módulo de Young

Além das aplicações diretas na engenharia e na ciência, a derivação numérica é também importante em uma variedade de contextos matemáticos gerais, incluindo outras áreas de métodos numéricos. Por exemplo, lembre-se de que, no Capítulo 6, o método da secante era baseado em uma aproximação por diferenças finitas da derivada. Além disso, a solução de equações diferenciais é, provavelmente, a aplicação mais importante da derivação numérica. Um exemplo na forma do método de Euler já foi visto no Capítulo 1. No Capítulo 24, investigaremos como a derivação numérica fornece as bases para soluções de problemas de valor de contorno de equações diferenciais ordinárias.

Essas são apenas algumas poucas aplicações da derivação com as quais um engenheiro pode se deparar regularmente no exercício de sua profissão. Quando as funções que devem ser analisadas são simples, normalmente optar-se-á pela avaliação analítica. Entretanto, isso é difícil, ou impossível, quando a função é complicada. Além disso, a função subjacente, em geral, é desconhecida e definida apenas por medidas em pontos discretos. Para ambos os casos, é necessário ter a habilidade de obter valores aproximados para derivadas, usando as técnicas numéricas descritas a seguir.

21.2 FÓRMULAS DE DERIVAÇÃO DE ALTA ACURÁCIA

Já introduzimos a noção de derivação numérica no Capítulo 4. Lembre-se de que usamos a expansão em série de Taylor para deduzir as aproximações por diferenças divididas finitas de derivadas. No Capítulo 4, desenvolvemos aproximações por diferenças progressiva, regressiva e centrada para a primeira derivada e para derivadas de ordem superior. Foi visto ainda que, no melhor caso, essas estimativas tinham erros que eram $O(h^2)$ – isto é, seus erros eram proporcionais ao quadrado do tamanho do passo. Esse nível de precisão decorre do número de termos da série de Taylor que foram mantidos durante a dedução dessas fórmulas. Mostraremos agora como fórmulas de diferenças divididas finitas podem ser geradas incluindo termos adicionais na expansão em série de Taylor.

Por exemplo, a expansão em série de Taylor progressiva pode ser escrita como [lembre-se da Equação (4.13)]

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \dots \quad (21.12)$$

o que pode ser resolvido por

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(x_i)}{2!}h + O(h^2) \quad (21.13)$$

No Capítulo 4, truncamos esse resultado excluindo os termos da segunda derivada e de derivadas de ordem superior, e ficamos com a seguinte fórmula de diferença progressiva:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \quad (21.14)$$

Em contraste a essa abordagem, agora manteremos o termo da segunda derivada substituindo a seguinte aproximação por diferença progressiva da segunda derivada [lembre-se da Equação (4.27)]:

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} + O(h) \quad (21.15)$$

na Equação (21.13) para obter

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{2h^2}h + O(h^2) \quad (21.16)$$

ou, agrupando os termos:

$$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h} + O(h^2) \quad (21.17)$$

Observe que a inclusão do termo da segunda derivada melhorou a precisão para $O(h^2)$. Versões melhoradas análogas podem ser desenvolvidas para as fórmulas regressiva e centrada, bem como aproximações para as derivadas de ordem superior. As fórmulas estão resumidas na Figura 21.3 a 21.5 junto com os resultados do Capítulo 4. O exemplo seguinte ilustra a utilidade dessas fórmulas na estimativa de derivadas.

EXEMPLO 21.1 Fórmulas de derivação de alta acurácia

Lembre-se de que, no Exemplo 4.4, obtivemos estimativas da derivada de

$$f(x) = -0,1x^4 - 0,15x^3 - 0,5x^2 - 0,25x + 1,2$$

em $x = 0,5$ usando diferenças divididas finitas e tamanho de passo de $h = 0,25$. Os resultados estão resumidos na tabela a seguir. Observe que os erros foram calculados com base no valor verdadeiro de $f'(0,5) = -0,9125$.

	Regressiva $O(h)$	Centrada $O(h^2)$	Progressiva $O(h)$
Estimativa	-0,714	-0,934	-1,155
ε_f	21,7%	-2,4%	-26,5%

Primeira Derivada	Erro
$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h}$	$O(h^2)$
Segunda Derivada	
$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$	$O(h)$
$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Terceira Derivada	
$f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3}$	$O(h)$
$f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3}$	$O(h^2)$
Quarta Derivada	
$f''''(x_i) = \frac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4}$	$O(h)$
$f''''(x_i) = \frac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4}$	$O(h^2)$

FIGURA 21.3 Fórmulas de diferença dividida finita progressiva: são apresentadas duas versões para cada derivada. A segunda versão incorpora mais termos da expansão em série de Taylor e é, consequentemente, mais acurada.

Repita esses cálculos, mas use as fórmulas de alta acurácia das Figuras 21.3 a 21.5.

Solução. Os dados necessários para esse exemplo são

$$\begin{array}{ll} x_{i-2} = 0 & f(x_{i-2}) = 1,2 \\ x_{i-1} = 0,25 & f(x_{i-1}) = 1,1035156 \\ x_i = 0,5 & f(x_i) = 0,925 \\ x_{i+1} = 0,75 & f(x_{i+1}) = 0,6363281 \\ x_{i+2} = 1 & f(x_{i+2}) = 0,2 \end{array}$$

A diferença progressiva de acurácia $O(h^2)$ é calculada por (Figura 21.3)

$$f'(0,5) = \frac{-0,2 + 4(0,6363281) - 3(0,925)}{2(0,25)} = -0,859375 \quad \varepsilon_t = 5,82\%$$

A diferença regressiva de acurácia $O(h^2)$ é calculada por (Figura 21.4)

$$f'(0,5) = \frac{3(0,925) - 4(1,1035156) + 1,2}{2(0,25)} = -0,878125 \quad \varepsilon_t = 3,77\%$$

A diferença centrada de acurácia $O(h^4)$ é calculada por (Figura 21.5)

$$f'(0,5) = \frac{-0,2 + 8(0,6363281) - 8(1,1035156) + 1,2}{12(0,25)} = -0,9125 \quad \varepsilon_t = 0\%$$

Primeira Derivada	Erro
$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
$f'(x_i) = \frac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h}$	$O(h^2)$
Segunda Derivada	
$f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2}$	$O(h)$
$f''(x_i) = \frac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2}$	$O(h^2)$
Terceira Derivada	
$f'''(x_i) = \frac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3}$	$O(h)$
$f'''(x_i) = \frac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3}$	$O(h^2)$
Quarta Derivada	
$f''''(x_i) = \frac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4})}{h^4}$	$O(h)$
$f''''(x_i) = \frac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5})}{h^4}$	$O(h^2)$

FIGURA 21.4 Fórmulas de diferença dividida finita regressiva: são apresentadas duas versões para cada derivada. A segunda versão incorpora mais termos da expansão em série de Taylor e é, consequentemente, mais acurada.

Como esperado, os erros para as diferenças progressiva e regressiva são consideravelmente mais acurados do que os resultados do Exemplo 4.4. Entretanto, surpreendentemente, a diferença centrada fornece um resultado exato da derivada em $x = 0,5$. Isso ocorre porque a fórmula baseada na série de Taylor equivale a passar um polinômio de quarto grau pelos pontos dados.

21.3 EXTRAPOLAÇÃO DE RICHARDSON

Até este ponto, vimos que existem duas maneiras de melhorar as estimativas da derivada quando se utiliza diferença dividida finita: (1) diminuir o tamanho do passo ou (2) usar uma fórmula de ordem mais alta que utiliza mais pontos. Uma terceira abordagem, baseada na extrapolação de Richardson, usa duas estimativas da derivada para calcular uma terceira aproximação, mais acurada.

Lembre-se, da Seção 20.2.1, que a extrapolação de Richardson fornece uma maneira de obter uma estimativa melhorada da integral I pela fórmula [Equação (20.4)]

$$I = I(h_2) + \frac{1}{(h_1/h_2)^2 - 1} [I(h_2) - I(h_1)] \quad (21.18)$$

onde $I(h_1)$ e $I(h_2)$ são estimativas da integral usando dois tamanhos de passo, h_1 e h_2 . Devido à sua conveniência quando expressa como um algoritmo computacional, essa fórmula geralmente é escrita para o caso em que $h_2 = h_1/2$, como em

$$I = \frac{4}{3}I(h_2) - \frac{1}{3}I(h_1) \quad (21.19)$$

Primeira Derivada	Erro
$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
$f'(x_i) = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h}$	$O(h^4)$
Segunda Derivada	
$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$	$O(h^2)$
$f''(x_i) = \frac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})}{12h^2}$	$O(h^4)$
Terceira Derivada	
$f'''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2h^3}$	$O(h^2)$
$f'''(x_i) = \frac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{8h^3}$	$O(h^4)$
Quarta Derivada	
$f''''(x_i) = \frac{f(x_{i+2}) - 4f(x_{i+1}) + 6f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{h^4}$	$O(h^2)$
$f''''(x_i) = \frac{-f(x_{i+3}) + 12f(x_{i+2}) - 39f(x_{i+1}) + 56f(x_i) - 39f(x_{i-1}) + 12f(x_{i-2}) - f(x_{i-3})}{6h^4}$	$O(h^4)$

FIGURA 21.5 Fórmulas de diferença dividida centrada: são apresentadas duas versões para cada derivada. A última versão incorpora mais termos da expansão em série de Taylor e é, consequentemente, mais acurada.

De modo similar, a Equação (21.19) pode ser escrita para derivadas como

$$D = \frac{4}{3}D(h_2) - \frac{1}{3}D(h_1) \quad (21.20)$$

Para aproximações por diferenças centradas com $O(h^2)$, a aplicação dessa fórmula produzirá uma nova estimativa da derivada de $O(h^4)$.

EXEMPLO 21.2 Extrapolação de Richardson

Usando a mesma função do Exemplo 21.1, obtenha uma estimativa da primeira derivada em $x = 0,5$, empregando tamanhos de passo de $h_1 = 0,5$ e $h_2 = 0,25$. Então, utilize a Equação (21.20) para calcular uma estimativa melhorada com a extrapolação de Richardson. Lembre-se de que o valor verdadeiro é $-0,9125$.

Solução. A estimativa da primeira derivada pode ser calculada com diferença centrada por

$$D(0,5) = \frac{0,2 - 1,2}{1} = -1,0 \quad \varepsilon_t = -9,6\%$$

e

$$D(0,25) = \frac{0,6363281 - 1,103516}{0,5} = -0,934375 \quad \varepsilon_t = -2,4\%$$

A estimativa melhorada pode ser determinada aplicando a Equação (21.20), para obter

$$D = \frac{4}{3}(-0,934375) - \frac{1}{3}(-1) = -0,9125$$

o que, neste caso, é um resultado exato.

O exemplo anterior fornece um resultado perfeito, porque a função analisada era um polinômio de quarto grau. Esse resultado decorreu do fato de que a extrapolação de Richardson é, na realidade, equivalente a ajustar um polinômio de grau mais alto aos dados e então calcular a derivada por diferenças divididas centradas. Assim, o caso presente ajustou a derivada do polinômio de quarto grau de modo exato. É claro que, para a maioria das outras funções, isso não ocorre e nossa estimativa da derivada seria melhorada, mas não perfeita. Consequentemente, como foi o caso na aplicação da extrapolação de Richardson, a abordagem pode ser aplicada iterativamente usando um algoritmo de Romberg até que o resultado caia abaixo de um critério de erro aceitável.

21.4 DERIVADAS DE DADOS DESIGUALMENTE ESPAÇADOS

As abordagens discutidas até agora foram desenvolvidas principalmente para determinar a derivada de uma função dada. Para as aproximações por diferenças divididas finitas da Seção 21.2, os dados precisavam estar igualmente espaçados. Para a técnica de extrapolação de Richardson da Seção 21.3, os dados precisavam também estar igualmente espaçados e ser gerados por divisões sucessivas na metade dos intervalos. Esse controle do espaçamento dos dados, em geral, estará disponível apenas quando usarmos uma função para gerar uma tabela de valores.

Já as informações obtidas experimentalmente – isto é, dados ou estudos de campo – são, em geral, coletados em intervalos desiguais. Essas informações não podem ser analisadas com as técnicas discutidas até aqui.

Uma forma de tratar dados desigualmente espaçados é ajustar um polinômio interpolador de Lagrange de segundo grau [lembre-se da Equação (17.21)] para um conjunto de pontos adjacentes que delimitam o ponto local no qual se quer calcular a derivada. Lembre-se de que esse polinômio não exige que os pontos sejam igualmente espaçados. O polinômio de segundo grau pode então ser diferenciado analiticamente para fornecer uma fórmula que pode ser usada para estimar a derivada.

Por exemplo, pode-se ajustar um polinômio de Lagrange de segundo grau para três pontos adjacentes (x_0, y_0) , (x_1, y_1) e (x_2, y_2) . A diferenciação do polinômio produz:

$$\begin{aligned} f'(x) &= f(x_0) \frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} \\ &\quad + f(x_2) \frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)} \end{aligned} \tag{21.21}$$

onde x é o valor no qual se quer estimar a derivada. Embora essa equação seja certamente mais complicada que a aproximação da primeira derivada das Figuras 21.3 a 21.5, ela tem algumas vantagens importantes. Em primeiro lugar, pode ser usada

para se obter uma estimativa da derivada em qualquer lugar do intervalo prescrito por esses três pontos. Em segundo lugar, os pontos propriamente ditos não precisam ser igualmente espaçados. Em terceiro, a estimativa da derivada tem a mesma ordem de acurácia que a diferença centrada [Equação (4.25)]. Na realidade, para pontos igualmente espaçados, a Equação (21.21) calculada em $x = x_i$ se reduz à Equação (4.25).

EXEMPLO 21.3 Derivando dados desigualmente espaçados

Como na Figura 21.6, um gradiente de temperatura pode ser medido no solo. O fluxo de calor na interface solo-ar pode ser calculado pela lei de Fourier (Tabela 21.1):

$$q(z = 0) = -k \frac{dT}{dz} \Big|_{z=0}$$

onde $q(z)$ é o fluxo de calor (W/m^2), k é o coeficiente de difusão térmica do solo [= 0,5 $\text{W}/(\text{m} \cdot \text{K})$], T é a temperatura (K) e z é a distância medida para baixo da superfície do solo (profundidade). Observe que um valor positivo para o fluxo significa que calor é transferido do ar para o solo. Use derivação numérica para calcular o gradiente na interface solo-ar e empregue essa estimativa para determinar o fluxo de calor para dentro do solo.

Solução. A Equação (21.21) pode ser usada para calcular a derivada na interface ar-solo como

$$\begin{aligned} f'(0) &= 13,5 \frac{2(0) - 0,0125 - 0,0375}{(0 - 0,0125)(0 - 0,0375)} + 12 \frac{2(0) - 0 - 0,0375}{(0,0125 - 0)(0,0125 - 0,0375)} \\ &\quad + 10 \frac{2(0) - 0 - 0,0125}{(0,0375 - 0)(0,0375 - 0,0125)} \\ &= -1440 + 1440 - 133,333 = -133,333 \text{ K/m} \end{aligned}$$

o que pode ser usado para calcular

$$q(z = 0) = -0,5 \frac{\text{W}}{\text{m K}} \left(-133,333 \frac{\text{K}}{\text{m}} \right) = 66,667 \frac{\text{W}}{\text{m}^2}$$

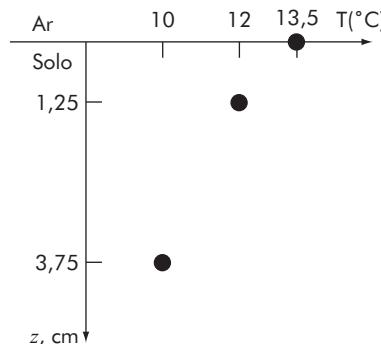


FIGURA 21.6 Temperatura em função da profundidade no solo.

21.5 DERIVADAS E INTEGRAIS PARA DADOS COM ERROS

Além de espaçamento desigual, um outro problema relacionado com a derivação de dados empíricos é que eles, em geral, incluem erros nas medidas. Uma deficiência da derivação numérica é que ela tende a amplificar os erros nos dados.

A Figura 21.7a mostra dados suaves, sem erros, que, quando derivados, fornecem um resultado suave (Figura 21.7b). Já a Figura 21.7c usa os mesmos dados, mas com alguns pontos ligeiramente levantados e outros ligeiramente abaixados. Essa pequena modificação quase não é percebida a partir da Figura 21.7c; entretanto, o efeito resultante na Figura 21.7d é significativo, porque o processo de derivação amplifica erros.

A amplificação de erros ocorre porque a derivação é subtrativa. Assim, erros positivos e negativos aleatórios tendem a se somar. Já o fato de a integração ser um processo de soma tende a torná-la muito tolerante em relação à incerteza nos dados. Em essência, como os pontos, em geral, são somados para formar uma integral, erros aleatórios positivos e negativos tendem a se cancelar.

Como esperado, a abordagem principal para determinar derivadas para dados imprecisos é o uso da regressão por mínimos quadrados para ajustar uma função suave e diferenciável aos dados. Na ausência de outras informações, uma regressão polinomial de grau baixo pode ser uma boa primeira escolha. Obviamente, se a relação funcional verdadeira entre as variáveis independente e dependente for conhecida, essa relação deve formar a base para o ajuste por mínimos quadrados.

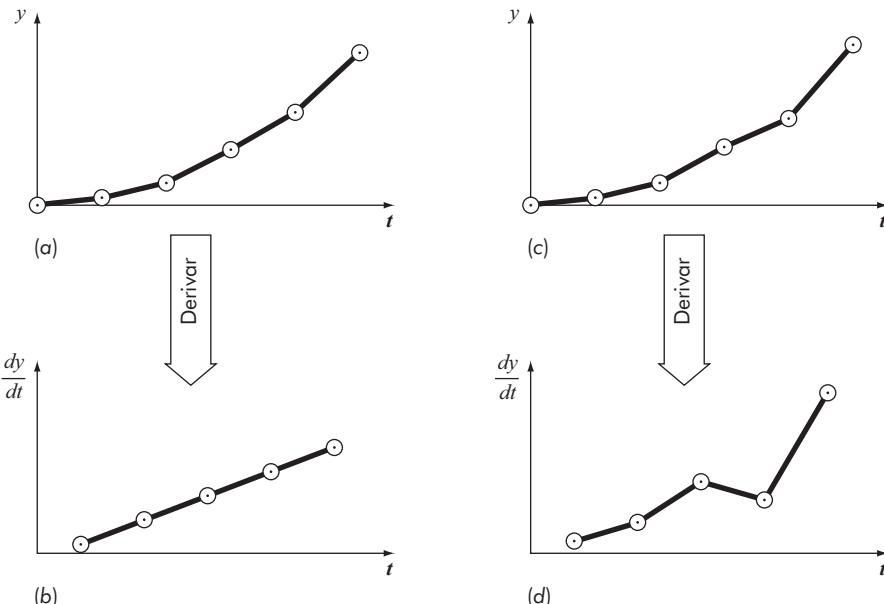


FIGURA 21.7 Ilustração de como pequenos erros nos dados são amplificados pela derivação numérica: (a) dados sem erros, (b) a derivação numérica resultante da curva (a), (c) dados ligeiramente modificados, e (d) a derivação resultante da curva (c) manifestando um aumento da variabilidade. Já a operação inversa de integração [movendo-se de (d) para (c) tomando a área sob (d)] tende a atenuar ou suavizar os erros nos dados.

21.6 DERIVADAS PARCIAIS

Derivadas parciais ao longo de uma dimensão são calculadas da mesma maneira que derivadas ordinárias. Por exemplo, considere que se queiram determinar as derivadas parciais para uma função bidimensional $f(x, y)$. Para dados igualmente espaçados, as derivadas parciais primeiras podem ser aproximadas com diferenças centradas:

$$\frac{\partial f}{\partial x} = \frac{f(x + \Delta x, y) - f(x - \Delta x, y)}{2\Delta x} \quad (21.22)$$

$$\frac{\partial f}{\partial y} = \frac{f(x, y + \Delta y) - f(x, y - \Delta y)}{2\Delta y} \quad (21.23)$$

Todas as outras fórmulas e abordagens discutidas até este ponto podem ser aplicadas para avaliar derivadas parciais de maneira similar.

Para derivadas de ordem superior, pode-se querer derivar uma função com relação a duas ou mais variáveis diferentes. O resultado é chamado de *derivada parcial mista*. Por exemplo, pode-se querer tomar a derivada parcial de $f(x, y)$ com relação a ambas as variáveis independentes

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial y} \right) \quad (21.24)$$

Para desenvolver uma aproximação por diferenças finitas, pode-se primeiro formar uma diferença em x das derivadas parciais em y :

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\frac{\partial f}{\partial y}(x + \Delta x, y) - \frac{\partial f}{\partial y}(x - \Delta x, y)}{2\Delta x} \quad (21.25)$$

Então, é possível usar diferenças finitas para avaliar cada uma das parciais em y :

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\frac{f(x + \Delta x, y + \Delta y) - f(x + \Delta x, y - \Delta y)}{2\Delta y} - \frac{f(x - \Delta x, y + \Delta y) - f(x - \Delta x, y - \Delta y)}{2\Delta y}}{2\Delta x} \quad (21.26)$$

Agrupando os termos, temos o resultado final

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{f(x + \Delta x, y + \Delta y) - f(x + \Delta x, y - \Delta y) - f(x - \Delta x, y + \Delta y) + f(x - \Delta x, y - \Delta y)}{4\Delta x \Delta y} \quad (21.27)$$

21.7 DERIVAÇÃO NUMÉRICA COM O MATLAB

O MATLAB tem a capacidade de determinar as derivadas de dados com base em duas funções nativas: `diff` e `gradient`.

21.7.1 Função nativa do MATLAB: `diff`

Quando um vetor unidimensional de comprimento n é passado como argumento de entrada, a função `diff` retorna um vetor de comprimento $n - 1$ que contém as

diferenças entre os elementos adjacentes. Como descreve o exemplo a seguir, essas diferenças podem ser empregadas para determinar aproximações por diferenças finitas das primeiras derivadas.

EXEMPLO 21.4 Usando a função `diff` para derivação

Explore como a função `diff` do MATLAB pode ser empregada para derivar a função

$$f(x) = 0,2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5$$

de $x = 0$ a $0,8$. Compare seus resultados com a solução exata:

$$f'(x) = 25 - 400x^2 + 2025x^2 - 3600x^3 + 2000x^4$$

Solução. Primeiro, podemos expressar $f(x)$ como uma função anônima:

```
>> f=@(x) 0.2+25*x-200*x.^2+675*x.^3-900*x.^4+400*x.^5;
```

Em seguida, podemos gerar uma série de valores igualmente espaçados das variáveis independente e dependente:

```
>> x=0:0.1:0.8;
>> y=f(x);
```

A função `diff` pode ser usada para determinar as diferenças entre elementos adjacentes de cada vetor. Por exemplo,

```
>> diff(x)
ans =
Columns 1 through 5
    0.1000    0.1000    0.1000    0.1000    0.1000
Columns 6 through 8
    0.1000    0.1000    0.1000
```

Como esperado, o resultado representa as diferenças entre cada par de elementos de x . Para calcular aproximações por diferenças finitas da derivada, simplesmente efetuamos uma divisão vetorial das diferenças de y pelas diferenças de x digitando:

```
>> d=diff(y)./diff(x)
d =
Columns 1 through 5
    10.8900   -0.0100    3.1900    8.4900    8.6900
Columns 6 through 8
    1.3900   -11.0100   -21.3100
```

Observe que, como estamos usando valores igualmente espaçados, depois de gerar os valores de x , podemos simplesmente efetuar o cálculo anterior de forma mais concisa como

```
>> d=diff(f(x))/0.1;
```

O vetor d contém agora estimativas das derivadas correspondentes ao ponto médio entre elementos adjacentes. Portanto, com o objetivo de desenvolver um gráfico de nossos resultados, devemos primeiro gerar um vetor contendo os valores de x para o ponto médio de cada intervalo:

```
>> n=length(x);
>> xm=(x(1:n-1)+x(2:n))./2;
```

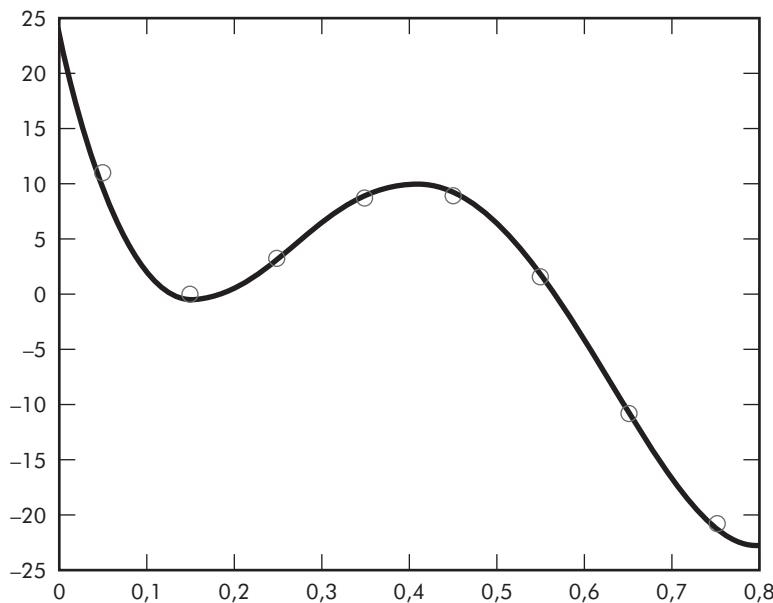


FIGURA 21.8 Comparação da derivada exata (linha) com a estimativa numérica (círculos) calculada com auxílio da função `diff` do MATLAB.

Como um último passo, podemos calcular valores para a derivada analítica em um nível mais fino de resolução para incluir no gráfico e permitir comparar os resultados numérico e analítico.

```
>> xa=0:.01:.8;
>> ya=25-400*xa+3*675*xa.^2-4*900*xa.^3+5*400*xa.^4;
```

Um gráfico das estimativas numérica e analítica pode ser gerado com

```
>> plot(xm,d,'o',xa,ya)
```

Como mostra a Figura 21.8, os resultados numérico e analítico concordam bem para este caso.

Observe que, além de avaliar derivadas, a função `diff` é oportuna como uma ferramenta de programação para testar certas características de vetores. Por exemplo, a declaração a seguir exibe uma mensagem de erro e finaliza um arquivo-M se ela determina que um vetor `x` possui elementos desigualmente espaçados:

```
if any(diff(diff(x))~=0), erro('desigualmente espaçado'),  
end
```

Outro uso comum é para detectar se um vetor está em ordem crescente ou decrescente. Por exemplo, o código a seguir rejeita um vetor que não está em ordem crescente (isto é, monotonicamente crescente):

```
if any(diff(x)<=0), erro('não está em ordem crescente'),  
end
```

21.7.2 Função nativa do MATLAB: gradient

A função `gradient` também retorna diferenças; entretanto, faz isso de uma maneira que é mais compatível com o cálculo de derivadas nos valores propriamente ditos, em vez de nos intervalos entre valores. Uma representação simples de sua sintaxe é

```
fx = gradient(f)
```

onde `f` é um vetor unidimensional de comprimento n e `fx` é um vetor de comprimento n que contém diferenças baseadas em `f`. Assim como com a função `diff`, o primeiro valor retornado é a diferença entre o primeiro e o segundo valor. Porém, para os valores intermediários, uma diferença centrada baseada nos valores adjacentes é retornada

$$\text{diff}_i = \frac{f_{i+1} - f_{i-1}}{2} \quad (21.28)$$

O último valor é então calculado como a diferença entre os dois últimos valores. Assim, os resultados se assemelham a usar diferenças centradas para todos os valores intermediários, com diferenças progressiva e regressiva nas extremidades.

Observe que o espaçamento entre os pontos é assumido igual a um. Se o vetor representa dados igualmente espaçados, a versão a seguir divide todos os resultados pelo intervalo `h`, portanto, retorna os valores reais das derivadas,

```
fx = gradient(f, h)
```

onde `h` é o espaçamento entre os pontos.

EXEMPLO 21.5 Usando a função `gradient` para derivação

Use a função `gradient` para derivar a mesma função que analisamos no Exemplo 21.4 com a função `diff`.

Solução. Da mesma forma como no Exemplo 21.4, podemos gerar uma série de valores igualmente espaçados das variáveis independente e dependente:

```
>> f=@(x) 0.2+25*x-200*x.^2+675*x.^3-900*x.^4+400*x.^5;
>> x=0:0.1:0.8;
>> y=f(x);
```

Podemos, então, usar a função `gradient` para determinar as derivadas como

```
>> dy=gradient(y,0.1)
dy =
Columns 1 through 5
    10.8900    5.4400    1.5900    5.8400    8.5900
Columns 6 through 9
    5.0400   -4.8100   -16.1600   -21.3100
```

Como no Exemplo 21.4, podemos gerar valores para a derivada analítica e exibir tanto a estimativa numérica como a estimativa analítica em um mesmo gráfico:

```
>> xa=0:.01:.8;
>> ya=25-400*xa+3*675*xa.^2-4*900*xa.^3+5*400*xa.^4;
>> plot(x,dy,'o', xa,ya)
```

Como mostra a Figura 21.9, os resultados não são tão acurados como aqueles obtidos com a função `diff` no Exemplo 21.4. Isso é devido ao fato de que a função `gradient` emprega intervalos que são duas vezes mais largos (0,2) do que aqueles usados pela função `diff` (0,1).

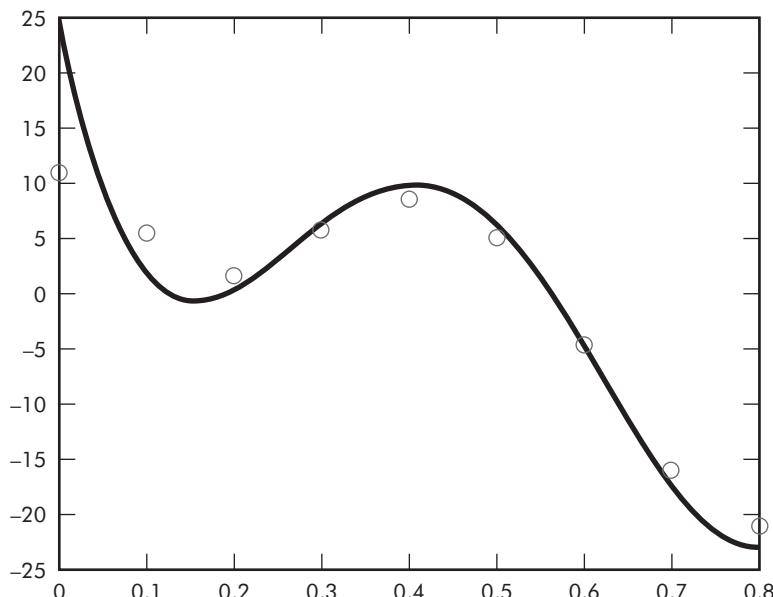


FIGURA 21.9 Comparação da derivada exata (linha) com a estimativa numérica (círculos) calculada com auxílio da função `gradient` do MATLAB.

Além de vetores unidimensionais, a função `gradient` é particularmente bem adequada para determinação de derivadas parciais de matrizes. Por exemplo, para uma matriz bidimensional, f , a função pode ser chamada como

$$[fx, fy] = \text{gradient}(f, h)$$

onde fx corresponde às diferenças na direção x (coluna), fy corresponde às diferenças na direção y (linha) e h é o espaçamento entre os pontos. Se h é omitido, o espaçamento entre pontos em ambas as dimensões é assumido igual a um. Na próxima seção, ilustraremos como a função `gradient` pode ser usada para visualizar campos vetoriais.

► 21.8 ESTUDO DE CASO VISUALIZAÇÃO DE CAMPOS

Contextualização. Além da determinação de derivadas em uma dimensão, a função gradient é também bastante útil para determinação de derivadas parciais em duas ou mais dimensões. Especificamente, ela pode ser usada em conjunto com outras funções do MATLAB para produzir visualizações de campo vetoriais.

Para compreender como isso é feito, podemos retornar à nossa discussão de derivadas parciais no fim da Seção 21.1.1. Lembre-se de que usamos uma elevação montanhosa como exemplo de uma função bidimensional, a qual podemos representar como

$$z = f(x, y)$$

onde z é elevação, x é a distância medida ao longo do eixo leste-oeste e y é a distância medida ao longo do eixo norte-sul.

Para esse exemplo, as derivadas parciais fornecem as inclinações nas direções dos eixos. Porém, se você estiver escalando a montanha, provavelmente estará mais interessado em determinar o sentido da máxima inclinação. Se entendermos as duas derivadas parciais como componentes vetoriais, a resposta é fornecida muito claramente por

$$\nabla f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j}$$

onde ∇f é conhecido como o *gradiente* de f . Esse vetor, que representa a inclinação mais acentuada, tem um módulo

$$\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

e um sentido

$$\theta = \tan^{-1} \left(\frac{\partial f / \partial y}{\partial f / \partial x} \right)$$

onde θ é o ângulo medido no sentido anti-horário a partir do eixo x .

Considere agora que geramos uma malha de pontos no plano xy e usamos as equações anteriores para desenhar o vetor gradiente em cada ponto. O resultado seria um campo de setas indicando o percurso mais íngreme para o pico a partir de qualquer ponto. Inversamente, se traçarmos o negativo do gradiente, ele indicaria como uma bola viajaria à medida que rolasse ladeira abaixo a partir de qualquer ponto.

Essas representações gráficas são tão úteis que o MATLAB tem uma função especial, chamada `quiver`, para criar tais gráficos. Uma representação simples de sua sintaxe é

```
quiver(x, y, u, v)
```

onde x e y são matrizes que contêm as coordenadas de posição e u e v são matrizes que contêm as derivadas parciais. O exemplo a seguir demonstra o uso da função `quiver` para visualizar um campo.

Empregue a função `gradient` para determinar as derivadas parciais para a seguinte função bidimensional:

$$f(x, y) = y - x - 2x^2 - 2xy - y^2$$



de $x = -2$ a 2 e $y = 1$ a 3 . Em seguida, use a função `quiver` para superpor um campo vetorial a um gráfico de curvas de nível da função.

Solução. Podemos expressar $f(x, y)$ como uma função anônima

```
>> f=@(x,y) y-x-2*x.^2-2.*x.*y-y.^2;
```

Uma série de valores igualmente espaçados das variáveis independente e dependente pode ser gerada como

```
>> [x,y]=meshgrid(-2:.25:0, 1:.25:3);
>> z=f(x,y);
```

A função `gradient` pode ser empregada para determinar as derivadas parciais:

```
>> [fx,fy]=gradient(z,0.25);
```

Podemos então desenvolver um gráfico de curvas de nível dos resultados:

```
>> cs=contour(x,y,z);clabel(cs);hold on
```

Por fim, a resultante das derivadas parciais pode ser superposta como vetores sobre o gráfico de curvas de nível:

```
>> quiver(x,y,-fx,-fy);hold off
```

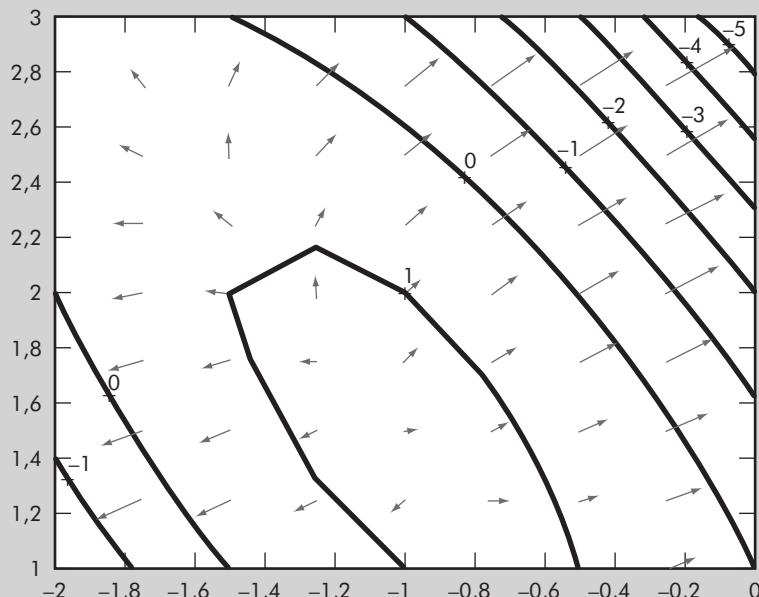


FIGURA 21.10 Gráfico de curvas de nível de uma função bidimensional com a resultante das derivadas parciais exibida como setas, gerado no MATLAB.

Observe que exibimos o negativo das resultantes, de modo que elas apontam “para baixo”.

O resultado está mostrado na Figura 21.10. O pico da função ocorre em $x = -1$ e $y = 1.5$ e então cai em todas as direções à medida que se afasta deste ponto. Como indicado pelo alongamento das setas, o gradiente cai mais acentuadamente para o nordeste e o sudoeste.

►► PROBLEMAS

21.1 Calcule aproximações por diferenças progressivas e regressivas de $O(h)$ e $O(h^2)$, e aproximações por diferenças centradas de $O(h^2)$ e $O(h^4)$ para a primeira derivada de $y = \cos x$ em $x = \pi/4$ usando um valor de $h = \pi/12$. Faça uma estimativa do erro relativo percentual verdadeiro ϵ_r para cada aproximação.

21.2 Use aproximações por diferenças centradas para obter estimativas para a primeira e a segunda derivadas de $y = e^x$ em $x = 2$ para $h = 0,1$. Empregue tanto fórmulas de ordem $O(h^2)$ como de ordem $O(h^4)$ para suas estimativas.

21.3 Utilize uma expansão em série de Taylor para deduzir uma aproximação por diferença finita centrada para a segunda derivada que seja acurada até segunda ordem. Para isso, você deve usar quatro expansões diferentes para os pontos x_{i-2} , x_{i-1} , x_{i+1} e x_{i+2} . Em cada caso, a expansão será em torno do ponto x_i . O intervalo Δx será usado no caso de $i - 1$ e $i + 1$ e $2\Delta x$ será usado no caso de $i - 2$ e $i + 2$. As quatro equações devem ser combinadas de forma a eliminar a primeira e a segunda derivadas. Mantenha termos suficientes em cada expansão para calcular o primeiro termo, que será truncado para determinar a ordem da aproximação.

21.4 Use a extrapolação de Richardson para obter uma estimativa da primeira derivada de $y = \cos x$ em $x = \pi/4$ utilizando tamanhos de passo de $h_1 = \pi/3$ e $h_2 = \pi/6$. Use diferenças centradas de $O(h^2)$ para a estimativa inicial.

21.5 Repita o Problema 21.4, mas para a primeira derivada de $\ln x$ em $x = 5$ usando $h_1 = 2$ e $h_2 = 1$.

21.6 Utilize a Equação (21.21) para determinar a primeira derivada de $y = 2x^4 - 6x^3 - 12x - 8$ em $x = 0$ com base nos valores em $x_0 = -0,5$, $x_1 = 1$ e $x_2 = 2$. Compare esse resultado com o valor verdadeiro e com a estimativa obtida usando uma aproximação por diferença centrada com base em $h = 1$.

21.7 Demonstre que, para pontos igualmente espaçados, a Equação (21.21) se reduz à Equação (4.25) em $x = x_1$.

21.8 Desenvolva um arquivo-M para aplicar o algoritmo de Romberg para estimar a derivada de uma função dada.

21.9 Desenvolva um arquivo-M para obter estimativas da primeira derivada para dados desigualmente espaçados. Teste-o com os seguintes dados:

x	0,6	1,5	1,6	2,5	3,5
$f(x)$	0,9036	0,3734	0,3261	0,08422	0,01596

onde $f(x) = 5e^{-2x}x$. Compare seus resultados com as derivadas verdadeiras.

21.10 Desenvolva uma função no MATLAB para calcular estimativas da primeira e segunda derivadas de ordem $O(h^2)$ com base nas fórmulas nas Figuras 21.3 a 21.5. A primeira linha da função deve ser definida como

```
function [dydx, d2ydx2] = diffeq(x,y)
```

onde x e y são vetores de entrada de comprimento n que contêm os valores das variáveis independente e dependente, respectivamente, e $dx dy$ e $d2ydx2$ são vetores de saída de comprimento n que contêm as estimativas da primeira e segunda derivadas em cada valor da variável independente. A função deve gerar um gráfico de $dydx$ e $d2ydx2$ versus x . Seu arquivo-M deve retornar uma mensagem de erro se (a) os vetores de entrada não têm o mesmo tamanho, ou (b) os valores para a variável independente não estão igualmente espaçados. Teste seu programa com os dados do Problema 21.11.

21.11 Foram coletados os seguintes dados para a distância percorrida em função do tempo para um foguete:

t, s	0	25	50	75	100	125
y, km	0	32	58	78	92	100

Use derivação numérica para obter estimativas da velocidade e da aceleração em cada instante.

21.12 A posição de um avião de caça em uma pista do porta-aviões foi cronometrada durante a aterrissagem:

t, s	0	0,52	1,04	1,75	2,37	3,25	3,83
x, m	153	185	208	249	261	271	273

onde x é a distância a partir da extremidade do porta-aviões. Estime (a) a velocidade (dx/dt) e (b) a aceleração (d^2x/dt^2) usando derivação numérica.

21.13 Use os seguintes dados para determinar a velocidade e a aceleração em $t = 10$ segundos:

Tempo, t, s	0	2	4	6	8	10	12	14	16
Posição, x, m	0	0,7	1,8	3,4	5,1	6,3	7,3	8,0	8,4

Use métodos de (a) diferença finita centrada, (b) diferença finita progressiva e (c) diferença finita regressiva corrigidos de segunda ordem.

21.14 Um avião está sendo monitorado por um radar, e os dados foram tomados a cada segundo em coordenadas polares θ e r .

t, s	200	202	204	206	208	210
$\theta, (\text{rad})$	0,75	0,72	0,70	0,68	0,67	0,66
r, m	5120	5370	5560	5800	6030	6240

Em 206 segundos, use diferença finita centrada (corrigida de segunda ordem) para determinar as expressões vetoriais da velocidade \vec{v} e da aceleração \vec{a} . A velocidade e a aceleração dadas em coordenadas polares são:

$$\vec{v} = \dot{r}\vec{e}_r + r\dot{\theta}\vec{e}_\theta \quad \text{e} \quad \vec{a} = (\ddot{r} - r\dot{\theta}^2)\vec{e}_r + (r\ddot{\theta} + 2\dot{r}\dot{\theta})\vec{e}_\theta$$

21.15 Use regressão para obter uma estimativa da aceleração para cada instante de tempo para os seguintes dados com polinômios de segundo, terceiro e quarto graus. Faça um gráfico de seus resultados:

t	1	2	3,25	4,5	6	7	8	8,5	9,3	10
v	10	12	11	14	17	16	12	14	14	10

21.16 A distribuição normal é definida por

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

Utilize o MATLAB para determinar os pontos de inflexão dessa função.

21.17 Os dados a seguir foram gerados a partir da distribuição normal:

x	-2	-1,5	-1	-0,5	0
$f(x)$	0,05399	0,12952	0,24197	0,35207	0,39894
x	0,5	1	1,5	2	
$f(x)$	0,35207	0,24197	0,12952	0,05399	

Use o MATLAB para determinar os pontos de inflexão desses dados.

21.18 Utilize o comando `diff(y)` para desenvolver uma função no MATLAB para calcular aproximações por diferenças finitas para a primeira e segunda derivadas em cada valor de x na tabela a seguir. Use aproximações por diferenças finitas que são corrigidas de segunda ordem, $O(x^2)$:

x	0	1	2	3	4	5	6	7	8	9	10
y	1,4	2,1	3,3	4,8	6,8	6,6	8,6	7,5	8,9	10,9	10

21.19 O objetivo deste problema é comparar aproximações por diferenças finitas progressiva,

regressiva e centrada acuradas até segunda ordem de uma função com o valor real da derivada. Isso será feito para

- $$f(x) = e^{-2x} - x$$
- (a) Use o cálculo para determinar o valor correto da derivada em $x = 2$.
- (b) Desenvolva uma função no MATLAB para calcular aproximações por diferenças centradas, começando com $x = 0,5$. Assim, para o primeiro cálculo, os valores de x para a aproximação por diferença centrada serão $x = 2 \pm 0,5$ ou $x = 1,5$ ou $2,5$. Então, diminua em incrementos de 0,01 até o valor mínimo de $\Delta x = 0,01$.
- (c) Repita a parte (b) para diferenças progressivas e regressivas de segunda ordem. (Observe que isso pode ser feito ao mesmo tempo em que a diferença centrada é calculada no laço.)
- (d) Faça gráficos dos resultados de (b) e (c) em função de x . Inclua o resultado exato no gráfico para comparação.

21.20 Você tem que medir a vazão de água através de um pequeno tubo. Para isso, coloca um balde na saída do tubo e mede o volume no balde como uma função do tempo, como tabulado a seguir. Faça uma estimativa da vazão em $t = 7$ s.

Tempo, s	0	1	5	8
Volume, cm³	0	1	8	16,4

21.21 A velocidade v (m/s) do ar que escoa por uma superfície plana é medida a diversas distâncias y (m) da superfície. Use a *lei de Newton da viscosidade* para determinar a tensão de cisalhamento τ (N/m²) na superfície ($y = 0$),

$$\tau = \mu \frac{du}{dy}$$

Considere o valor $\mu = 1,8 \times 10^{-5}$ N · s/m² para a viscosidade dinâmica.

y, m	0	0,002	0,006	0,012	0,018	0,024
$u, \text{m/s}$	0	0,287	0,899	1,915	3,048	4,299

21.22 A *primeira lei de difusão de Fick* afirma que

$$\text{Fluxo de massa} = -D \frac{dc}{dx} \quad (\text{P21.22})$$

onde o fluxo de massa é a quantidade de massa que passa por unidade de área por unidade de

tempo ($\text{g}/\text{cm}^2/\text{s}$), D é o coeficiente de difusão (cm^2/s), c é a concentração (g/cm^3) e x é a distância (cm). Um engenheiro ambiental mede as seguintes concentrações de um poluente no sedimento no fundo de um lago ($x = 0$ na interface sedimento-água e aumentando para baixo):

$x, \text{ cm}$	0	1	3
$c, 10^{-6} \text{ g}/\text{cm}^3$	0,06	0,32	0,6

Use a melhor técnica de derivação disponível para obter uma estimativa da derivada em $x = 0$. Empregue essa estimativa em conjunto com a Equação (P21.22) para calcular o fluxo de massa do poluente saindo do sedimento e indo para a água no fundo do lago ($D = 1,52 \times 10^{-6} \text{ cm}^2/\text{s}$). Para um lago com $3,6 \times 10^6 \text{ m}^2$ de sedimentos, quanto poluente seria transportado para ele em um período de um ano?

21.23 Os seguintes dados foram coletados quando um grande tanque de óleo estava sendo carregado

$t, \text{ min}$	0	10	20	30	45	60	75
$V, 10^6 \text{ barris}$	0,4	0,7	0,77	0,88	1,05	1,17	1,35

Calcule a vazão Q (isto é, dV/dt) para cada instante, com ordem de h^2 .

21.24 A lei de Fourier é usada rotineiramente pelos engenheiros arquitetônicos para determinar o fluxo de calor através de paredes. As seguintes temperaturas foram medidas a partir da superfície ($x = 0$) de uma parede de pedra:

$x, \text{ m}$	0	0,08	0,16
$T, ^\circ\text{C}$	20,2	17	15

Se o fluxo em $x = 0$ é $60 \text{ W}/\text{m}^2$, calcule k .

21.25 A área da superfície horizontal $A_s (\text{m}^2)$ de um lago a uma profundidade particular pode ser calculada a partir do volume por derivação:

$$A_s(z) = \frac{dV}{dz}(z)$$

onde V é o volume (m^3) e z é a profundidade (m) medida da superfície até o fundo. A concentração média de uma substância que varia com a profundidade, \bar{c} (g/m^3), pode ser calculada por integração:

$$\bar{c} = \frac{\int_0^Z c(z) A_s(z) dz}{\int_0^Z A_s(z) dz}$$

onde Z é a profundidade total (m). Calcule a concentração média com base nos seguintes dados:

$z, \text{ m}$	0	4	8	12	16
$V, 10^6 \text{ m}^3$	9,8175	5,1051	1,9635	0,3927	0,0000
$c, \text{ g}/\text{m}^3$	10,2	8,5	7,4	5,2	4,1

21.26 A lei de Faraday caracteriza a queda de tensão através de um indutor como

$$V_L = L \frac{di}{dt}$$

onde V_L é a queda de tensão (V), L é a indutância (em henrys; $1 \text{ H} = 1 \text{ V} \cdot \text{s}/\text{A}$), i é a corrente (A) e t é o tempo (s). Determine a queda de tensão como uma função do tempo a partir dos dados seguintes para uma indutância de 4 H .

t	0	0,1	0,2	0,3	0,5	0,7
i	0	0,16	0,32	0,56	0,84	2,0

21.27 Com base na lei de Faraday (Problema 21.26), use os seguintes dados de tensão para estimar a indutância se uma corrente de 2 A circula através de um indutor por um período de 400 milissegundos.

$t, \text{ ms}$	0	10	20	40	60	80	120	180	280	400
$V, \text{ volts}$	0	18	29	44	49	46	35	26	15	7

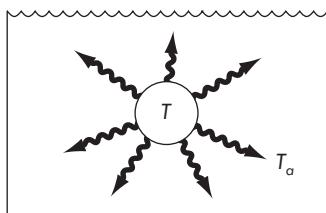
21.28 A taxa de resfriamento de um corpo (Figura P21.28) pode ser expressa por

$$\frac{dT}{dt} = -k(T - T_a)$$

onde T é a temperatura do corpo ($^\circ\text{C}$), T_a é a temperatura do meio ao redor ($^\circ\text{C}$) e k é uma constante de proporcionalidade (por minuto). Logo, essa equação (chamada de *lei de resfriamento de Newton*) especifica que a taxa de resfriamento é proporcional à diferença entre a temperatura do corpo e a do meio ao redor. Se uma bola de metal aquecida a $80 \text{ }^\circ\text{C}$ for jogada na água cuja temperatura é mantida constante em $T_a = 20 \text{ }^\circ\text{C}$, a temperatura da bola muda como em

Tempo, min	0	5	10	15	20	25
$T, ^\circ\text{C}$	80	44,5	30,0	24,1	21,7	20,7

Utilize derivação numérica para determinar dT/dt em cada instante de tempo. Trace dT/dt versus $T - T_a$ e empregue regressão linear para calcular k .

**FIGURA P21.28**

21.29 A entalpia de um gás real é uma função da pressão como descrito a seguir. Os dados foram tomados para um fluido real. Faça uma estimativa da entalpia do fluido a 400 K e a 50 atm (calcule a integral de 0 atm a 50 atm).

$$H = \int_0^P \left(V - T \left(\frac{\partial V}{\partial T} \right)_P \right) dP$$

<i>P</i> , atm	<i>V</i> , L		
	<i>T</i> = 350 K	<i>T</i> = 400 K	<i>T</i> = 450 K
0,1	220	250	282,5
5	4,1	4,7	5,23
10	2,2	2,5	2,7
20	1,35	1,49	1,55
25	1,1	1,2	1,24
30	0,90	0,99	1,03
40	0,68	0,75	0,78
45	0,61	0,675	0,7
50	0,54	0,6	0,62

21.30 Para o escoamento de fluido sobre uma superfície, o fluxo de calor para a superfície pode ser calculado com a lei de Fourier: y é a distância normal à superfície (m). As seguintes medidas foram feitas para o ar que escoa sobre uma placa plana em que y é a distância normal à superfície:

<i>y</i> , cm	0	1	3	5
<i>T</i> , K	900	480	270	210

Se as dimensões da placa são 200 cm de comprimento e 50 cm de largura e $k = 0,028 \text{ J/(s} \cdot \text{m} \cdot \text{K)}$, (a) determine o fluxo na superfície, e (b) o calor transferido em watts. Observe que $1 \text{ J} = 1 \text{ W} \cdot \text{s}$.

21.31 O gradiente de pressão para escoamento laminar através de um tubo de raio constante é dado por

$$\frac{dp}{dx} = -\frac{8\mu Q}{\pi r^4}$$

onde p é a pressão (N/m^2), x é a distância ao longo do eixo central do tubo (m), μ é a viscosidade dinâmica ($\text{N} \cdot \text{s/m}^2$), Q é escoamento (m^3/s) e r é o raio (m).

- (a) Determine a queda de pressão para um tubo de 10 cm de comprimento para um líquido viscoso ($\mu = 0,005 \text{ N} \cdot \text{s/m}^2$, densidade $\rho = 1 \times 10^3 \text{ kg/m}^3$), com um escoamento de $10 \times 10^{-6} \text{ m}^3/\text{s}$ e os seguintes raios variáveis ao longo de seu comprimento:

<i>x</i> , cm	0	2	4	5	6	7	10
<i>r</i> , mm	2	1,35	1,34	1,6	1,58	1,42	2

- (b) Compare seu resultado com a queda de pressão que teria ocorrido se o tubo tivesse um raio constante igual ao raio médio.
(c) Determine o número de Reynolds médio para o tubo para verificar que o escoamento é realmente laminar ($\text{Re} = \rho v D / \mu < 2.100$, onde v é a velocidade).

21.32 Os dados a seguir para o calor específico do benzeno foram gerados com um polinômio de grau n . Use derivação numérica para determinar n .

<i>T</i> , K	300	400	500	600
C_p , kJ/(kmol · K)	82,888	112,136	136,933	157,744
<i>T</i> , K	700	800	900	1000
C_p , kJ/(kmol · K)	175,036	189,273	200,923	210,450

21.33 O calor específico em uma pressão constante c_p [J/(kg · K)] de um gás ideal está relacionado à entalpia por

$$c_p = \frac{dh}{dT}$$

onde h é a entalpia (kJ/kg) e T é a temperatura absoluta (K). As seguintes entalpias foram fornecidas para o dióxido de carbono (CO_2) em várias temperaturas. Use esses valores para determinar o calor específico em J/(kg · K) para cada uma das temperaturas tabuladas. Observe que os pesos atômicos do carbono e oxigênio são 12,011 e 15,9994 g/mol, respectivamente.

<i>T</i> , K	750	800	900	1.000
<i>h</i> , kJ/kmol	29.629	32.179	37.405	42.769

21.34 Uma lei de taxa de ordem n é frequentemente utilizada para modelar reações químicas que dependem exclusivamente da concentração de um único reagente:

$$\frac{dc}{dt} = -kc^n$$

onde c é a concentração (mols), t é o tempo (min), n é a ordem da reação (adimensional) e k é a taxa de reação (min^{-1} mols $^{1-n}$). O *método diferencial* pode ser usado para calcular os parâmetros k e n . Isso envolve a aplicação de uma transformada logarítmica para a lei de taxa para produzir,

$$\log\left(-\frac{dc}{dt}\right) = \log k + n \log c$$

Portanto, se uma lei de taxa de ordem n é válida, um gráfico de $\log(-dc/dt)$ versus $\log c$ deveria produzir uma reta com uma inclinação de n e uma interseção de $\log k$. Use o método diferencial e regressão linear pra determinar k e n para os dados a seguir para a conversão de cianato de amônio para ureia:

$t, \text{ min}$	0	5	15	30	45
$c, \text{ mols}$	0,750	0,594	0,420	0,291	0,223

21.35 A demanda de oxigênio no sedimento [SOD – *Sediment Oxygen Demand* – em unidades de $\text{g}/(\text{m}^2 \cdot \text{d})$] é um parâmetro importante na determinação do conteúdo de oxigênio dissolvido em águas naturais. Ele é medido colocando-se um núcleo de sedimento em um recipiente cilíndrico (Figura P21.35). Depois de introduzida cuidadosamente uma camada de água destilada, oxigenada, acima dos sedimentos, o recipiente é coberto para evitar a transferência de gases. Um agitador é usado para misturar a água suavemente, e uma sonda de oxigênio controla como a concentração de oxigênio da água diminui ao longo do tempo. A SOD pode, então, ser calculada como

$$\text{SOD} = -H \frac{do}{dt}$$

onde H é a profundidade da água (m), o é a concentração de oxigênio (g/m^3) e t é o tempo (d).

Com base nos dados a seguir e $H=0,1$ m, use derivação numérica para gerar gráficos de (a) SOD versus tempo e (b) SOD versus concentração de oxigênio:

$t, \text{ d}$	0	0,125	0,25	0,375	0,5	0,625	0,75
$o, \text{ mg/L}$	10	7,11	4,59	2,57	1,15	0,33	0,03

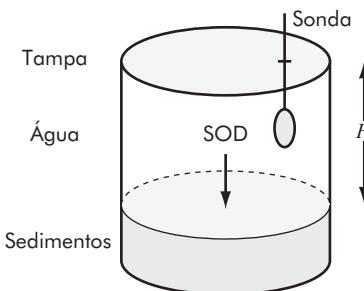


FIGURA P21.35

21.36 As seguintes relações podem ser usadas para analisar vigas uniformes sujeitas a cargas distribuídas:

$$\frac{dy}{dx} = \theta(x) \quad \frac{d\theta}{dx} = \frac{M(x)}{EI} \quad \frac{dM}{dx} = V(x) \quad \frac{dV}{dx} = -w(x)$$

onde x é a distância ao longo da viga (m), y é a deflexão (m), $\theta(x)$ é a inclinação (m/m), E é o módulo de elasticidade ($\text{Pa} = \text{N}/\text{m}^2$), I é o momento de inércia (m^4), $M(x)$ é o momento (N m), $V(x)$ é a força de cisalhamento (N) e $w(x)$ é a carga distribuída (N/m). Para o caso de uma carga linearmente crescente (lembre-se da Figura P5.13), a inclinação pode ser calculada analiticamente como

$$\theta(x) = \frac{w_0}{120EI L} (-5x^4 + 6L^2x^2 - L^4) \quad (\text{P21.36})$$

Empregue (a) integração numérica para calcular a deflexão (em m), e (b) derivação numérica para calcular o momento (em N m) e a força de cisalhamento (em N). Tome como base para seus cálculos numéricos os valores da inclinação calculados com a Equação (P21.36) em intervalos igualmente espaçados de $\Delta x = 0,125$ m ao longo de uma viga de 3 m. Use os seguintes parâmetros em seus cálculos: $E = 200 \text{ GPa}$, $I = 0,0003 \text{ m}^4$ e $w_0 = 2,5 \text{ kN}/\text{cm}$. Além disso, as deflexões nas extremidades da viga são definidas como $y(0) = y(L) = 0$. Seja cuidadoso com as unidades.

21.37 Você mediou as seguintes deflexões ao longo do comprimento de uma viga uniforme simplesmente apoiada (ver Problema 21.36)

$x, \text{ m}$	0	0,375	0,75	1,125	1,5
$y, \text{ cm}$	0	-0,2571	-0,9484	-1,9689	-3,2262
$x, \text{ m}$	1,875	2,25	2,625	3	
$y, \text{ cm}$	-4,6414	-6,1503	-7,7051	-9,275	

Empregue derivação numérica para calcular a inclinação, o momento (em N m), a força de cisalha-

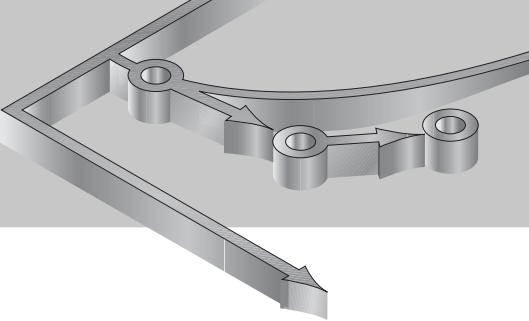
mento (em N) e a carga distribuída (em N/m). Use os seguintes valores de parâmetros em seus cálculos: $E = 200 \text{ GPa}$ e $I = 0,0003 \text{ m}^4$.

21.38 Calcule $\partial f/\partial x$, $\partial f/\partial y$ e $\partial^2 f/(\partial x \partial y)$ para a função a seguir em $x = y = 1$ (a) analiticamente e (b) numericamente com $\Delta x = \Delta y = 0,0001$:

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

21.39 Desenvolva um programa no MATLAB para gerar os mesmos cálculos e gráficos como na Seção 21.8, mas para as seguintes funções (para $x = -3$ a 3 e $y = -3$ a 3): (a) $f(x, y) = e^{-(x^2+y^2)}$ e (b) $f(x, y) = xe^{-(x^2+y^2)}$.

21.40 Desenvolva um programa no MATLAB para gerar os mesmos cálculos e gráficos como na Seção 21.8, mas para a função `peaks` do MATLAB, ao longo do intervalo de -3 a 3 para x e y .



Parte 6

Equações diferenciais ordinárias

6.1 VISÃO GERAL

O que são equações algébricas lineares?

As leis fundamentais da física, da mecânica, da eletricidade e da termodinâmica são, em geral, baseadas em observações empíricas que explicam a variação das propriedades físicas e dos estados dos sistemas. Em vez de descrever os estados de sistemas físicos diretamente, as leis geralmente são dadas em termos de variações espaciais e temporais, e definem mecanismos de variação. Quando combinadas com as leis de continuidade da energia, da massa e do momento, produzem equações diferenciais. A integração subsequente dessas equações diferenciais resulta em funções matemáticas que descrevem o estado espacial e temporal de um sistema em termos da variação de energia, massa ou velocidade. Como na Figura PT6.1, a integração pode ser implementada analiticamente, com o cálculo, ou numericamente, com o computador.

O problema do saltador de *bungee jumping* em queda livre introduzido no Capítulo 1 é um exemplo da dedução de uma equação diferencial a partir de uma lei fundamental. Lembre-se de que a segunda lei de Newton foi usada para desenvolver uma EDO que descreve a taxa de variação da velocidade de um saltador de *bungee jumping* em queda:

$$\frac{dv}{dt} = g - \frac{c_d}{m} v^2 \quad (\text{PT6.1})$$

onde g é a aceleração da gravidade, m é a massa e c_d é um coeficiente de arraste. Essas equações, que são compostas por uma função desconhecida e suas derivadas, são chamadas *equações diferenciais*. Elas são, algumas vezes, denominadas *equações de taxa*, porque expressam a taxa de variação de uma variável como uma função das variáveis e parâmetros.

Na Equação (PT6.1), a quantidade que está sendo derivada, v , é chamada *variável dependente*. A quantidade com relação à qual v é derivada, t , é denominada *variável independente*. Quando a função envolve uma única variável independente, a equação é conhecida como *equação diferencial ordinária* (ou *EDO*). Isso para diferenciá-la de uma *equação diferencial parcial* (ou *EDP*), que envolve duas ou mais variáveis independentes.

As equações diferenciais também são classificadas em relação à sua ordem. Por exemplo, a Equação (PT6.1) é denominada *equação de primeira ordem*, pois a sua derivada mais alta é a primeira derivada. Uma *equação de segunda ordem* inclui uma segunda derivada. Por exemplo, a equação que descreve a posição x de um sistema massa-mola com amortecimento é a equação de segunda ordem:

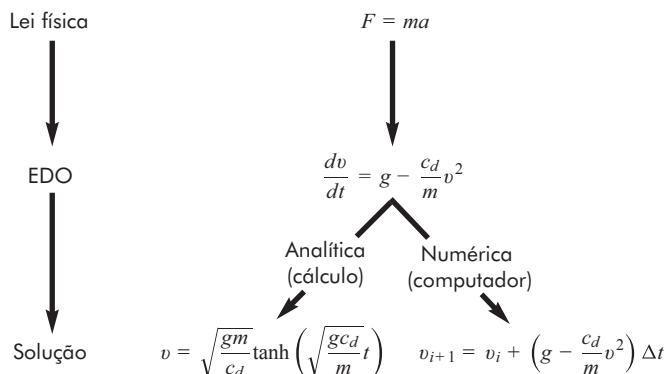


FIGURA PT6.1 A sequência de eventos no desenvolvimento e na resolução de EDOs para problemas de engenharia e de ciência. O exemplo mostrado é para a velocidade do saltador de *bungee jumping* em queda livre.

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0 \quad (\text{PT6.2})$$

onde m é a massa, c é um coeficiente de amortecimento e k é a constante da mola. De maneira similar, uma equação de ordem n inclui uma derivada n -ésima.

Equações de ordem superior podem ser reduzidas a um sistema de equações de primeira ordem. Para a Equação (PT6.2), isso é feito definindo-se uma nova variável v como a primeira derivada do deslocamento

$$v = \frac{dx}{dt} \quad (\text{PT6.3})$$

onde v é a velocidade. Essa equação pode ser derivada para fornecer

$$\frac{dv}{dt} = \frac{d^2x}{dt^2} \quad (\text{PT6.4})$$

As Equações (PT6.3) e (PT6.4) podem ser substituídas na Equação (PT6.2) para convertê-la em uma equação de primeira ordem:

$$m \frac{dv}{dt} + cv + kx = 0 \quad (\text{PT6.5})$$

Por fim, podemos expressar as Equações (PT6.3) e (PT6.5) como equações de taxa:

$$\frac{dx}{dt} = v \quad (\text{PT6.6})$$

$$\frac{dv}{dt} = -\frac{c}{m}v - \frac{k}{m}x \quad (\text{PT6.7})$$

Assim, as Equações (PT6.6) e (PT6.7) são um par de equações de primeira ordem que são equivalentes à equação original de segunda ordem [Equação (PT6.2)]. Como as outras equações diferenciais de ordem n podem ser reduzidas de forma análoga, esta parte do livro se concentra na solução de equações de primeira ordem.

Uma solução de uma equação diferencial ordinária é uma função específica da variável independente e dos parâmetros que satisfaz a equação diferencial original. Para ilustrar esse conceito, começaremos com um polinômio de quarto grau simples,

$$y = -0,5x^4 + 4x^3 - 10x^2 + 8,5x + 1 \quad (\text{PT6.8})$$

Agora, se derivarmos a Equação (PT6.8), obtemos uma EDO:

$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8,5 \quad (\text{PT6.9})$$

Essa equação também descreve o comportamento do polinômio, mas de modo diferente da Equação (PT6.8). Em vez de descrever explicitamente os valores de y para cada valor de x , a Equação (PT6.9) dá a taxa de variação de y com relação a x (isto é, a inclinação) em todos os valores de x . A Figura PT6.2 mostra o gráfico da função e da derivada em função de x . Observe como os valores nulos da derivada correspondem a pontos nos quais a função original é horizontal – ou seja, tem inclinação nula. Além disso, os valores absolutos máximos das derivadas estão nas extremidades do intervalo, em que as inclinações da função são maiores.

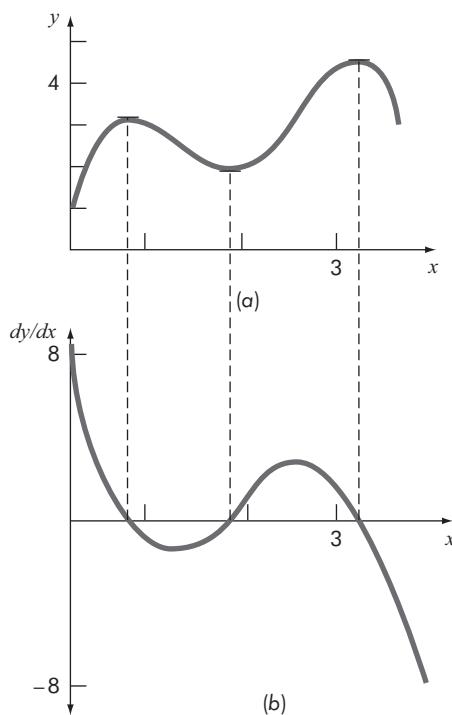


FIGURA PT6.2 Gráficos de (a) y versus x e (b) dy/dx versus x para a função $y = -0,5x^4 + 4x^3 - 10x^2 + 8,5x + 1$.

Embora, como visto, possamos determinar uma equação diferencial dada a função original, o objetivo aqui é determinar a função original dada a equação diferencial. A função original, então, representa a solução.

Sem computadores, as EDOS são normalmente resolvidas analiticamente com o cálculo. Por exemplo, a Equação (PT6.9) pode ser multiplicada por dx e integrada para produzir

$$y = \int (-2x^3 + 12x^2 - 20x + 8,5) dx \quad (\text{PT6.10})$$

O lado direito dessa equação é chamado *integral indefinida*, porque os limites de integração não são especificados. Chamamos dessa forma para diferenciar essa integral das *integrais definidas* discutidas anteriormente na Parte 5 [compare a Equação (PT6.10) com a Equação (19.5)].

Uma solução analítica para a Equação (PT6.10) é obtida se a integral indefinida puder ser avaliada exatamente na forma de equação. Para esse caso simples, isso pode ser feito com o resultado:

$$y = -0,5x^4 + 4x^3 - 10x^2 + 8,5x + C \quad (\text{PT6.11})$$

que é idêntico à função original, exceto por uma diferença importante. Durante a derivação e a posterior integração, perdemos o valor da constante 1 na equação original e ganhamos o valor C . Esse C é chamado *constante de integração*. O fato de tal constante arbitrária aparecer indica que a solução não é única. Na realidade, ela é apenas uma de um número infinito de possíveis funções (correspondendo a um número infinito de possíveis valores de C) que satisfazem a equação diferencial. Por exemplo, a Figura PT6.3 mostra seis funções possíveis que satisfazem a Equação (PT6.11).

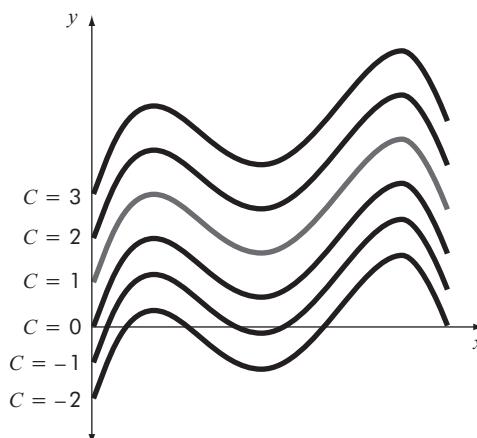


FIGURA PT6.3 Seis soluções possíveis para a integral de $-2x^3 + 12x^2 - 20x + 8,5$. Cada uma corresponde a um valor diferente da constante de integração C .

Portanto, para especificar completamente a solução, uma equação diferencial é normalmente acompanhada de condições auxiliares. Para EDOs de primeira ordem, é necessário um tipo de condição auxiliar denominada *valor inicial*, para determinar a constante e obter uma solução única. Por exemplo, a equação diferencial original pode ser acompanhada pela condição inicial que em $x = 0$, $y = 1$. Esses valores podem ser substituídos na Equação (PT6.11) para determinar $C = 1$. Desse modo, a única solução que satisfaz tanto a equação diferencial quanto a condição inicial especificada é

$$y = -0,5x^4 + 4x^3 - 10x^2 + 8,5x + 1$$

Assim, “prendemos” a Equação (PT6.11) forçando-a a passar pela condição inicial e, ao fazer isso, determinamos uma única solução para a EDO e completamos o ciclo para a função original [Equação (PT6.8)].

As condições iniciais, em geral, têm interpretação física tangível nas equações diferenciais provenientes de situações de problemas físicos reais. Por exemplo, no problema do saltador de *bungee jumping*, a condição inicial era um reflexo do fato físico de que, no instante zero, a velocidade vertical era nula. Se o saltador já estivesse em movimento vertical no instante zero, a solução teria sido modificada para levar em conta essa velocidade inicial.

Em se tratando de uma equação diferencial de ordem n , serão necessárias n condições para obter uma solução única. Se todas as condições forem especificadas em um mesmo valor da variável independente (por exemplo, em $x = 0$ ou $t = 0$), então o problema é chamado *problema de valor inicial*. Isso para diferenciá-lo dos *problemas de valor de contorno*, nos quais as especificações das condições ocorrem em valores diferentes da variável independente. Os Capítulos 22 e 23 se concentrarão nos problemas de valor inicial, e os problemas de valor de contorno serão estudados no Capítulo 24.

6.2 ORGANIZAÇÃO DA PARTE 6

O Capítulo 22 é dedicado aos métodos de passo único para resolução de EDOs de valor inicial. Como o nome indica, os *métodos de passo único* calculam uma predição futura y_{i+1} com base apenas na informação em um único ponto y_i e em nenhuma outra informação anterior. Isso está em contraste com as *abordagens de passo múltiplo*, que usam informação de vários pontos anteriores como base para a extrapolação para um novo valor.

Salvo uma pequena exceção, os métodos de passo único descritos no Capítulo 22 pertencem às *técnicas de Runge-Kutta*. Embora o capítulo pudesse ter sido organizado em torno dessa noção teórica, optamos por uma abordagem mais gráfica e intuitiva para introduzirmos os métodos. Assim, começamos o capítulo com o *método de Euler*, o qual tem uma interpretação gráfica muito simples. Além disso, como já introduzimos o método de Euler no Capítulo 1, nossa ênfase aqui é na quantificação de seu erro de truncamento e descrição de sua estabilidade.

Em seguida, usamos argumentos visualmente orientados para desenvolver duas versões melhoradas do método de Euler – as técnicas de *Heun* e do *ponto médio*.

Depois dessa introdução, deduziremos formalmente o conceito das abordagens de Runge-Kutta (ou RK) e mostraremos como as técnicas anteriores são de fato métodos RK de primeira e segunda ordens. Isso é seguido por uma discussão das formulações RK de ordem elevada que são frequentemente usadas na resolução de problemas de engenharia e de ciência. Além disso, discutimos a aplicação dos métodos de passo único aos *sistemas de EDOs*. Observe que todas as aplicações no Capítulo 22 são limitadas aos casos com um tamanho de passo fixo.

No Capítulo 23, discutimos abordagens mais avançadas para resolução de problemas de valor inicial. Primeiro, descrevemos os *métodos RK adaptativos*, que automaticamente ajustam o tamanho do passo em resposta ao erro de truncamento do cálculo. Esses métodos são especialmente pertinentes uma vez que são empregados pelo MATLAB para resolver EDOs.

A seguir, discutiremos *métodos de passo múltiplo*. Como mencionado anteriormente, esses algoritmos retêm informações dos passos anteriores para capturar de forma mais eficiente a trajetória da solução. Eles também fornecem estimativas do erro de truncamento que podem ser usadas para implementar o controle do tamanho do passo. Será descrito um método simples – o método de *Heun de início não automático* – para introduzir as características essenciais das abordagens de passo múltiplo.

Por fim, o capítulo termina com uma descrição de *EDOs rígidas*. Estas são tanto EDOs individuais quanto sistemas de EDOs que têm ambas componentes rápidas e lentas em suas soluções. Como consequência, elas requerem abordagens especiais de solução. Introduzimos a ideia de uma técnica de *solução implícita* como uma solução geralmente usada para remediar esse problema. Também, descrevemos funções nativas do MATLAB para resolução de EDOs rígidas.

No Capítulo 24, nos concentramos em duas abordagens para obtenção de soluções para *problemas de valor de contorno*: o *método shooting* (ou *método do tiro*) e o *método de diferenças finitas*. Além de demonstrar como essas técnicas são implementadas, ilustramos como elas lidam com *condições de contorno na derivada* e *EDOs não lineares*.

Problemas de valor inicial

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar a resolução de problemas de valor inicial para EDOs (Equações Diferenciais Ordinárias). Após esta leitura completa, você será capaz de:

- ✓ Entender o significado de erros de truncamento local e global e a relação deles com o tamanho do passo para métodos de passo único para resolução de EDOs.
- ✓ Aprender como implementar os seguintes métodos de Runge-Kutta (RK) para uma única EDO:
 - Euler;
 - Heun;
 - Ponto médio;
 - RK de quarta ordem.
- ✓ Aprender como iterar o corretor do método de Heun.
- ✓ Aprender como implementar os seguintes métodos de Runge-Kutta (RK) para sistemas de EDOs:
 - Euler;
 - RK de quarta ordem.

VOCÊ TEM UM PROBLEMA

Iniciamos este livro com o problema de simulação da velocidade de um saltador de *bungee jumping* em queda livre. Esse problema resultou na formulação e resolução de uma equação diferencial ordinária, o tópico deste capítulo. Retornaremos agora a esse problema e o tornaremos mais interessante, calculando o que acontece quando o saltador chega ao fim da corda elástica.

Para isso, devemos reconhecer que o saltador experimentará forças diferentes, dependendo da condição da corda: se está frouxa (ou não esticada – no início do salto) ou esticada. Se a corda não está esticada, a situação é a de queda livre, em que as únicas forças são a gravidade e o arraste. Porém, como o saltador pode agora se mover para cima bem como para baixo, o sinal da força de arraste deve ser modificado de modo que ela sempre tenda a retardar a velocidade.

$$\frac{dv}{dt} = g - \text{sign}(v) \frac{c_d}{m} v^2 \quad (22.1a)$$

onde v é a velocidade (m/s), t é o tempo (s), g é a aceleração devido à gravidade ($9,81 \text{ m/s}^2$), c_d é o coeficiente de arraste (kg/m) e m é a massa (kg). A função *signum*¹, sinal, retorna -1 ou 1 , dependendo de seu argumento ser negativo ou positivo, respectivamente. Assim, quando o saltador está caindo (velocidade positiva, sinal = 1), a força de arraste será negativa e, portanto, agirá de forma a reduzir a velocidade. Já quando o saltador está se movendo para cima (velocidade negativa, sinal = -1), a força de arraste será positiva, de modo que ela, de novo, reduz a velocidade.

Uma vez que a corda começa a esticar, obviamente exerce uma força para cima sobre o saltador. Como feito no Capítulo 8, a lei de Hooke pode ser usada como uma primeira aproximação dessa força. Além disso, uma força de amortecimento também deve ser incluída para se ter em conta o atrito à medida que a corda se estica e se contrai. Esses fatores podem ser incorporados juntamente com a gravidade e o arraste a um segundo balanço de forças que se aplica quando a corda está esticada. O resultado é a seguinte equação diferencial:

$$\frac{dv}{dt} = g - \text{sign}(v) \frac{c_d}{m} v^2 - \frac{k}{m}(x - L) - \frac{\gamma}{m}v \quad (22.1b)$$

onde k é a constante da mola da corda (N/m), x é a distância vertical medida para baixo a partir da plataforma de salto (m), L é o comprimento da corda não esticada (m) e γ é um coeficiente de amortecimento ($\text{N} \cdot \text{s/m}$).

Como a Equação (22.1b) é válida apenas quando a corda está esticada ($x > L$), a força da mola será sempre negativa. Isto é, ela sempre agirá de forma a puxar o saltador de volta para cima. A força de amortecimento aumenta em módulo à medida que a velocidade do saltador aumenta, e sempre age de modo a desacelerar o saltador para baixo.

Se quisermos simular a velocidade do saltador, podemos resolver primeiro a Equação (22.1a) até que a corda seja totalmente estendida. Depois, podemos mudar para a Equação (22.1b) para aqueles períodos em que a corda estiver esticada. Embora isso seja bastante simples, esse procedimento indica que o conhecimento da posição do saltador é necessário. Isso pode ser feito formulando-se outra equação diferencial para a distância:

$$\frac{dx}{dt} = v \quad (22.2)$$

Assim, a determinação da velocidade do saltador de *bungee jumping* equivale a resolver duas equações diferenciais ordinárias, em que uma das equações assume formas diferentes, dependendo do valor de uma das variáveis dependentes. Os Capítulos 22 e 23 exploram métodos para resolução deste e de outros problemas similares envolvendo equações diferenciais ordinárias (EDO).

¹ Algumas linguagens de computador representam a função signum, sinal, como `sgn(x)`. Como representado aqui, o MATLAB usa a nomenclatura `sign(x)`.

22.1 VISÃO GERAL

Este capítulo é dedicado à solução de equações diferenciais ordinárias da forma

$$\frac{dy}{dt} = f(t, y) \quad (22.3)$$

No Capítulo 1, usamos um método numérico para resolver uma equação desse tipo para a velocidade do saltador de *bungee jumping* em queda livre. Lembre-se de que o método tinha a forma geral

Novo valor = valor antigo + inclinação × tamanho do passo

ou, em termos matemáticos,

$$y_{i+1} = y_i + \phi h \quad (22.4)$$

onde a inclinação ϕ é chamada de *função incremento*. De acordo com essa equação, a estimativa da inclinação ϕ é usada para extrapolar de um valor antigo y_i para um valor novo y_{i+1} , em uma distância h . Essa fórmula pode ser aplicada passo a passo para traçar a trajetória da solução para o futuro. Esses métodos são chamados *métodos de passo único*, porque o valor da função incremento é baseado na informação em um único ponto i . Eles são também conhecidos como *métodos de Runge-Kutta*, após os dois matemáticos aplicados os terem discutido no início de 1900. Outra classe de métodos, chamada *métodos de passo múltiplo*, usa informações de vários pontos anteriores como base para extração de um novo valor. Os métodos de passo múltiplo serão brevemente descritos no Capítulo 23.

Todos os métodos de passo único podem ser expressos na forma geral da Equação (22.4), sendo que a única diferença é a maneira como é feita a estimativa da inclinação. A abordagem mais simples é usar a equação diferencial para obter uma estimativa da inclinação na forma da primeira derivada em t_i ; ou seja, a inclinação no início do intervalo é tomada como uma aproximação da inclinação média em todo o intervalo. Essa abordagem, chamada de método de Euler, é discutida a seguir. O assunto é seguido por outros métodos de passo único que usam estimativas alternativas da inclinação que resultam em previsões mais acuradas.

22.2 MÉTODO DE EULER

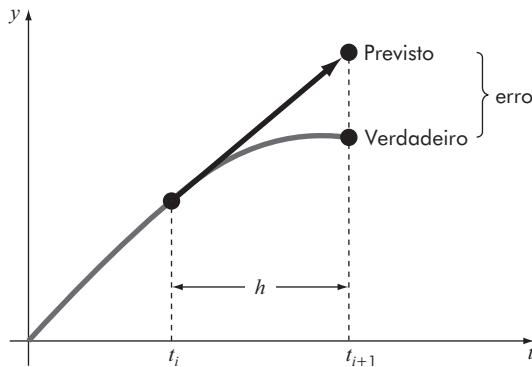
A primeira derivada fornece uma estimativa direta da inclinação em t_i (Figura 22.1)

$$\phi = f(t_i, y_i)$$

onde $f(t_i, y_i)$ é a equação diferencial calculada em t_i e y_i . Essa estimativa pode ser substituída na Equação (22.1):

$$y_{i+1} = y_i + f(t_i, y_i)h \quad (22.5)$$

Essa fórmula é conhecida como *método de Euler* (ou Euler-Cauchy ou ponto-inclinação). Um novo valor de y é previsto usando a inclinação (igual à primeira derivada no valor original de t) para extrapolar linearmente sobre um tamanho de passo h (Figura 22.1).

**FIGURA 22.1** Método de Euler.**EXEMPLO 22.1** Método de Euler

Use o método de Euler para integrar $y' = 4e^{0,8t} - 0,5y$ de $t = 0$ a 4 , com um tamanho de passo de 1 . A condição inicial em $t = 0$ é $y = 2$. Lembre-se de que a solução exata pode ser determinada analiticamente como:

$$y = \frac{4}{1,3}(e^{0,8t} - e^{-0,5t}) + 2e^{-0,5t}$$

Solução. A Equação (22.5) pode ser usada para implementar o método de Euler:

$$y(1) = y(0) + f(0, 2)(1)$$

onde $y(0) = 2$ e a estimativa da inclinação em $t = 0$ é

$$f(0, 2) = 4e^0 - 0,5(2) = 3$$

Portanto,

$$y(1) = 2 + 3(1) = 5$$

A solução verdadeira em $t = 1$ é

$$y = \frac{4}{1,3} (e^{0,8(1)} - e^{-0,5(1)}) + 2e^{-0,5(1)} = 6,19463$$

Assim, o erro relativo percentual é

$$\varepsilon_t = \left| \frac{6,19463 - 5}{6,19463} \right| \times 100\% = 19,28\%$$

Para o segundo passo:

$$\begin{aligned} y(2) &= y(1) + f(1, 5)(1) \\ &= 5 + [4e^{0,8(1)} - 0,5(5)](1) = 11,40216 \end{aligned}$$

TABELA 22.1 Comparação dos valores verdadeiro e aproximado da integral de $y' = 4e^{0.8t} - 0.5y$, com condição inicial de que $y = 2$ em $t = 0$. Os valores numéricos foram calculados usando o método de Euler com um tamanho de passo de 1.

t	$y_{\text{verdadeiro}}$	y_{Euler}	$ e_t (\%)$
0	2,00000	2,00000	
1	6,19463	5,00000	19,28
2	14,84392	11,40216	23,19
3	33,67717	25,51321	24,24
4	75,33896	56,84931	24,54

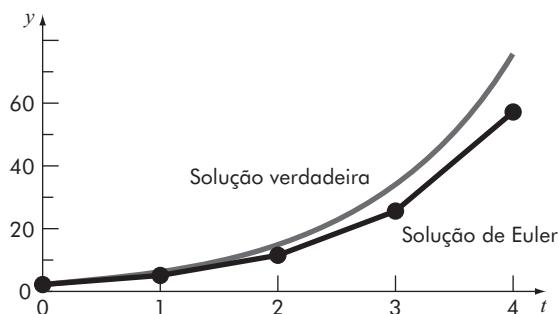


FIGURA 22.2 Comparação da solução verdadeira com a solução numérica usando o método de Euler para a integral de $y' = 4e^{0.8t} - 0.5y$ de $t = 0$ a 4 com um tamanho de passo de 1,0. A condição inicial em $t = 0$ é $y = 2$.

A solução verdadeira em $t = 2,0$ é 14,84392 e, portanto, o erro relativo percentual é 23,91%. O cálculo é repetido e os resultados estão resumidos na Tabela 22.1 e na Figura 22.2. Observe que, embora os cálculos capturem a tendência geral da solução verdadeira, o erro é considerável. Como discutirá a próxima seção, esse erro pode ser reduzido pelo uso de um tamanho de passo menor.

22.2.1 Análise de erro para o método de Euler

A solução numérica de EDOs envolve dois tipos de erro (lembre-se do Capítulo 4):

1. Erros de *truncamento* ou de discretização causados pela natureza das técnicas usadas para aproximar os valores de y .
2. Erros de *arredondamento* causados pelo número limitado de algarismos significativos que podem ser representados por um computador.

Os erros de truncamento são compostos de duas partes. A primeira é o *erro de truncamento local*, que resulta da aplicação do método em questão em um único passo. A segunda é o *erro de truncamento propagado*, que resulta das aproximações produzidas durante os passos anteriores. A soma dos dois é o *erro de truncamento global* ou total.

Uma visão do valor absoluto e das propriedades do erro de truncamento pode ser obtida deduzindo-se o método de Euler diretamente da expansão em série de

Taylor. Para fazê-lo, lembre-se de que a equação a ser integrada terá a forma geral da Equação (22.3), onde $dy/dt = y'$, e t e y são as variáveis independente e dependente, respectivamente. Se a solução – isto é, a função que descreve o comportamento de y – tiver derivadas contínuas, ela pode ser representada por uma expansão em série de Taylor em torno de um valor inicial (t_i, y_i) , como em [lembre-se da Equação (4.13)]:

$$y_{i+1} = y_i + y'_i h + \frac{y''_i}{2!} h^2 + \cdots + \frac{y^{(n)}_i}{n!} h^n + R_n \quad (22.6)$$

onde $h = t_{i+1} - t_i$ e R_n é o resto, dado por

$$R_n = \frac{y^{(n+1)}(\xi)}{(n+1)!} h^{n+1} \quad (22.7)$$

onde ξ está em algum ponto do intervalo entre t_i e t_{i+1} . Uma forma alternativa pode ser desenvolvida substituindo-se a Equação (22.3) nas Equações (22.6) e (22.7), para obter

$$y_{i+1} = y_i + f(t_i, y_i)h + \frac{f'(t_i, y_i)}{2!} h^2 + \cdots + \frac{f^{(n-1)}(t_i, y_i)}{n!} h^n + O(h^{n+1}) \quad (22.8)$$

onde $O(h^{n+1})$ especifica que o erro de truncamento é proporcional ao tamanho do passo elevado à $(n+1)$ -ésima potência.

Comparando-se as Equações (22.5) e (22.8), pode ser visto que o método de Euler corresponde à série de Taylor até o termo $f(t_i, y_i)h$ incluindo este. Além disso, a comparação indica que ocorre um erro de truncamento porque aproximamos a solução verdadeira usando um número finito de termos da série de Taylor. Então, truncamos ou deixamos de fora uma parte da solução. Por exemplo, o erro de truncamento no método de Euler é atribuído aos termos restantes na série de Taylor que não foram incluídos na Equação (22.5). Subtraindo a Equação (22.5) da Equação (22.8), obtemos

$$E_t = \frac{f'(t_i, y_i)}{2!} h^2 + \cdots + O(h^{n+1}) \quad (22.9)$$

onde E_t é o erro de truncamento local verdadeiro. Para h suficientemente pequeno, os termos de ordem superior na Equação (22.9) são geralmente desprezíveis, e o resultado é frequentemente representado por

$$E_a = \frac{f'(t_i, y_i)}{2!} h^2 \quad (22.10)$$

ou

$$E_a = O(h^2) \quad (22.11)$$

onde E_a é o erro de truncamento local aproximado.

De acordo com a Equação (22.11), vemos que o erro local é proporcional ao quadrado do tamanho do passo e à primeira derivada da equação diferencial. É possível observar também que o erro de truncamento global é $O(h)$ – isto é, proporcional ao tamanho do passo (Carnahan et al., 1969). Essas observações levam a algumas conclusões úteis:

1. O erro global pode ser reduzido diminuindo-se o tamanho do passo.
2. O método fornecerá previsões livres de erro se a função subjacente (isto é, a solução da equação diferencial) for linear, porque, para uma reta, a segunda derivada seria nula.

A última conclusão faz sentido do ponto de vista intuitivo, porque o método de Euler usa segmentos de reta para aproximar a solução; portanto, ele é conhecido como um *método de primeira ordem*.

Deve ser observado também que esse padrão geral é válido para os métodos de passo único de ordem superior descritos nas páginas seguintes. Isto é, um método de ordem n fornecerá um resultado perfeito se a solução subjacente for um polinômio de grau n . Além disso, o erro de truncamento local será $O(h^{n+1})$ e o erro global, $O(h^n)$.

22.2.2 Estabilidade do método de Euler

Na seção anterior, aprendemos que o erro de truncamento do método de Euler depende do tamanho do passo de uma maneira previsível com base na série de Taylor. Esta é uma questão de acurácia.

A estabilidade de um método de solução é outra importante análise que deve ser considerada na resolução de EDOs. Uma solução numérica é dita instável se os erros crescem exponencialmente para um problema para o qual há uma solução limitada. A estabilidade de uma aplicação particular pode depender de três fatores: a equação diferencial, o método numérico e o tamanho do passo.

Uma visão sobre o tamanho de passo necessário para estabilidade pode ser obtida estudando-se uma EDO muito simples:

$$\frac{dy}{dt} = -ay \quad (22.12)$$

Se $y(0) = y_0$, o cálculo pode ser usado para determinar a solução como

$$y = y_0 e^{-at}$$

Assim, a solução inicia em y_0 e aproxima-se de zero assintoticamente.

Agora, considere que o método de Euler seja usado para resolver o mesmo problema numericamente:

$$y_{i+1} = y_i + \frac{dy_i}{dt}h$$

Substituindo na Equação (22.12), tem-se

$$y_{i+1} = y_i - ay_i h$$

ou

$$y_{i+1} = y_i(1 - ah) \quad (22.13)$$

A quantidade entre parênteses, $1 - ah$, é chamada *fator amplificador*. Se seu valor absoluto é maior que a unidade, a solução crescerá de maneira ilimitada. Então, claramente, a estabilidade depende do tamanho do passo h ; ou seja, se $h > 2/a$, $|y_i| \rightarrow \infty$ quando $i \rightarrow \infty$. Com base nessa análise, o método de Euler é dito *condicionalmente estável*.

Observe que há certas EDOs em que os erros sempre crescem independentemente do método, e, por isso, são chamadas *mal condicionadas*.

Inacurácia e instabilidade são frequentemente confundidas. Isso ocorre provavelmente porque (a) ambas representam situações em que a solução numérica falha e (b) ambas são afetadas pelo tamanho do passo. Entretanto, elas são problemas distintos. Por exemplo, um método incorreto pode ser muito estável. Retornaremos ao tópico quando discutirmos sistemas rígidos, no Capítulo 23.

22.2.3 Função do MATLAB: `eulode`

Já desenvolvemos um arquivo-M simples para implementar o método de Euler para o problema do saltador de *bungee jumping* em queda. Lembre-se, da Seção 3.6, que essa função usou o método de Euler para calcular a velocidade após um dado intervalo de tempo de queda livre. Agora, desenvolveremos um algoritmo mais geral, para todos os fins.

A Figura 22.3 mostra um arquivo-M que usa o método de Euler para calcular valores da variável dependente y ao longo de um intervalo de valores da variável independente t . O nome da função contendo o lado direito da equação diferencial é passado para a função como a variável `dydt`. Os valores inicial e final do intervalo desejado da variável independente são passados como um vetor `tspan`. O valor inicial e o tamanho de passo desejado são passados como `y0` e `h`, respectivamente.

A princípio, a função gera um vetor t ao longo do intervalo desejado da variável dependente utilizando um incremento de h . Naqueles casos em que o tamanho do passo não é igualmente divisível no intervalo, o último valor ficará aquém do último valor do intervalo. Se isso ocorre, o valor final é adicionado a t , de modo que a série abrange o intervalo completo. O comprimento do vetor t é determinado como n . Além disso, um vetor da variável dependente y é pré-alocado com n valores da condição inicial para melhorar a eficiência.

Neste ponto, o método de Euler [Equação (22.5)] é implementado por um único laço:

```
for i = 1:n-1
    y(i+1) = y(i) + dydt(t(i),y(i),varargin{:})*(t(i+1) -
    t(i));
end
```

Observe como uma função é usada para gerar um valor para a derivada nos valores apropriados das variáveis independente e dependente, e como o passo de tempo é automaticamente calculado com base na diferença entre valores adjacentes no vetor t .

A EDO a ser resolvida pode ser definida de várias maneiras. A princípio, a equação diferencial pode ser definida como uma função anônima. Por exemplo, para a EDO do Exemplo 22.1:

```
>> dydt=@(t,y) 4*exp(0.8*t) - 0.5*y;
```

A solução pode ser gerada como

```
>> [t,y] = eulode(dydt,[0 4],2,1);
>> disp([t,y])
```

```

function [t,y] = eulode(dydt,tspan,y0,h,varargin)
% eulode: Solução de EDO pelo método de Euler
% [t,y] = eulode(dydt,tspan,y0,h,p1,p2,...):
%     usa o método de Euler para integrar uma EDO
% entrada:
% dydt = nome do arquivo-M que avalia a EDO
% tspan = [ti, tf] onde ti e tf são os valores
% inicial e final da variável independente
% y0 = valor inicial da variável dependente
% h = tamanho do passo
% p1,p2,... = parâmetros adicionais usados por dydt
% saída:
% t = vetor da variável independente
% y = vetor de solução para a variável dependente
if nargin<4,error('são necessários pelo menos 4 argumentos de
entrada'),end
ti = tspan(1);tf = tspan(2);
if ~(tf>ti),error('o limite superior deve ser maior que o
inferior'),end
t = (ti:h:tf)'; n = length(t);
% se necessário, inclui um valor adicional de t
% de modo que o intervalo vá de t = ti a tf
if t(n)<tf
    t(n+1) = tf;
    n = n+1;
end
y = y0*ones(n,1); %pré-aloca y para melhorar a eficiência
for i = 1:n-1 %implementa o método de Euler
    y(i+1) = y(i) + dydt(t(i),y(i),varargin{:})*(t(i+1)-t(i));
end

```

FIGURA 22.3 Função do MATLAB para implementar o método de Euler.

com o resultado (compare com a Tabela 22.1):

0	2.0000
1.0000	5.0000
2.0000	11.4022
3.0000	25.5132
4.0000	56.8493

Embora a utilização de uma função anônima seja praticável para este caso, haverá problemas mais complexos nos quais a definição da EDO requer várias linhas de código. Nesses casos, a criação de um arquivo-M separado é a opção mais viável (se não a única).

22.3 MELHORIAS NO MÉTODO DE EULER

Uma fonte fundamental de erro no método de Euler é que consideramos que a derivada no início do intervalo pode ser usada no intervalo todo. Estão disponíveis duas modificações simples para contornar essas deficiências. Como será demonstrado na

Seção 22.4, ambas as modificações na verdade pertencem a uma classe maior de técnicas de solução chamadas de *Runge-Kutta*. Entretanto, como elas têm uma interpretação gráfica muito simples, nós as apresentaremos antes de sua dedução formal como métodos de Runge-Kutta.

22.3.1 Método de Heun

Um método para melhorar a estimativa da inclinação envolve a determinação de duas derivadas para o intervalo – uma no ponto inicial e outra no ponto final. Então, é feita uma média das duas derivadas para obter uma estimativa melhorada da inclinação no intervalo todo. Essa abordagem, chamada de *método de Heun*, é descrita graficamente na Figura 22.4.

Lembre-se de que, no método de Euler, a inclinação no início de um intervalo

$$y'_i = f(t_i, y_i) \quad (22.14)$$

é usada para extrapolar linearmente para y_{i+1} :

$$y_{i+1}^0 = y_i + f(t_i, y_i)h \quad (22.15)$$

No método de Euler padrão, pararíamos nesse ponto. No entanto, no método de Heun, o y_{i+1}^0 calculado com a Equação (22.15) não é a resposta final, mas uma previsão intermediária. É por isso que o distinguimos com o sobrescrito 0. A Equação (22.15) é chamada *equação preditora*, e fornece uma estimativa que permite o cálculo da inclinação na extremidade do intervalo:

$$y'_{i+1} = f(t_{i+1}, y_{i+1}^0) \quad (22.16)$$

Assim, as duas inclinações [Equações (22.14) e (22.16)] podem ser combinadas para obter uma inclinação média para o intervalo:

$$\bar{y}' = \frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1}^0)}{2}$$

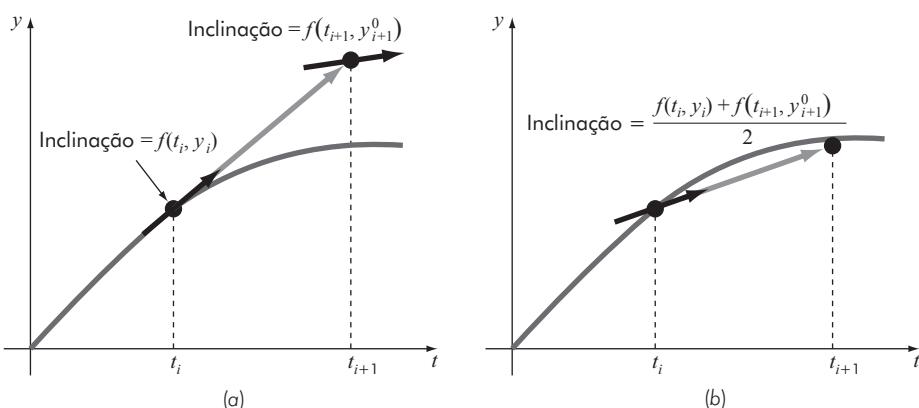


FIGURA 22.4 Descrição gráfica do método de Heun. (a) Preditor e (b) corretor.

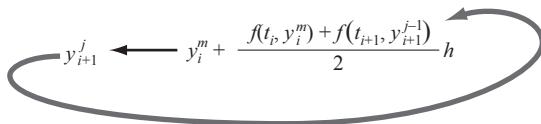


FIGURA 22.5 Representação gráfica da iteração do corretor do método de Heun para obter uma estimativa melhorada.

Essa inclinação média é então usada para extrapolar linearmente de y_i a y_{i+1} usando o método de Euler:

$$y_{i+1} = y_i + \frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1}^0)}{2} h \quad (22.17)$$

que é chamada de *equação corretora*.

O método de Heun é uma *abordagem do tipo predictor-corretor*. Como deduzimos anteriormente, ele pode ser representado sucintamente como

$$\text{Preditor (Figura 22.4a): } y_{i+1}^0 = y_i^m + f(t_i, y_i)h \quad (22.18)$$

$$\text{Corretor (Figura 22.4b): } y_{i+1}^j = y_i^m + \frac{f(t_i, y_i^m) + f(t_{i+1}, y_{i+1}^{j-1})}{2} h \quad (22.19)$$

(for $j = 1, 2, \dots, m$)

Observe que, como a Equação (22.19) tem y_{i+1} em ambos os lados do sinal de igualdade, pode ser aplicada de forma iterativa; ou seja, uma estimativa antiga pode ser usada repetidamente para fornecer uma estimativa melhorada de y_{i+1} . O processo é descrito na Figura 22.5.

Como no caso dos métodos iterativos similares discutidos nas seções anteriores do livro, um critério de parada para a convergência do corretor é fornecido por

$$|\varepsilon_a| = \left| \frac{y_{i+1}^j - y_{i+1}^{j-1}}{y_{i+1}^j} \right| \times 100\%$$

onde y_{i+1}^{j-1} e y_{i+1}^j são os resultados da interação anterior e da iteração presente do corretor, respectivamente. Deve ficar claro que esse processo iterativo não converge necessariamente para a resposta verdadeira, mas convergirá para uma estimativa com um erro de truncamento finito, como ilustra o exemplo a seguir.

EXEMPLO 22.2 Método de Heun

Use o método de Heun para integrar $y' = 4e^{0.8t} - 0.5y$ de $t = 0$ a 4 , com um tamanho de passo de 1 . A condição inicial em $t = 0$ é $y = 2$. Use um critério de parada de $0,00001\%$ para encerrar as iterações do corretor.

Solução. A inclinação em (t_0, y_0) é calculada como

$$y'_0 = 4e^0 - 0.5(2) = 3$$

TABELA 22.2 Comparação dos valores verdadeiro e aproximado da integral de $y' = 4e^{0.8t} - 0,5y$, com a condição inicial de $y = 2$ em $t = 0$. Os valores numéricos foram calculados usando os métodos de Euler e Heun com um tamanho de passo de 1. O método de Heun foi implementado com e sem iteração do corretor.

t	$y_{\text{verdadeiro}}$	y_{Euler}	$ \varepsilon_t $ (%)	Sem iteração		Com iteração	
				y_{Heun}	$ \varepsilon_t $ (%)	y_{Heun}	$ \varepsilon_t $ (%)
0	2,00000	2,00000		2,00000		2,00000	
1	6,19463	5,00000	19,28	6,70108	8,18	6,36087	2,68
2	14,84392	11,40216	23,19	16,31978	9,94	15,30224	3,09
3	33,67717	25,51321	24,24	37,19925	10,46	34,74328	3,17
4	75,33896	56,84931	24,54	83,33777	10,62	77,73510	3,18

Então, o preditor é usado para calcular um valor em 1,0:

$$y_1^0 = 2 + 3(1) = 5$$

Observe que esse é o resultado que seria obtido pelo método de Euler padrão. O valor verdadeiro na Tabela 22.2 mostra que ele corresponde a um erro relativo percentual de 19,28%. Agora, para melhorar a estimativa para y_{i+1} , usamos o valor y_1^0 para prever a inclinação no final do intervalo

$$y_1' = f(x_1, y_1^0) = 4e^{0.8(1)} - 0,5(5) = 6,402164$$

que pode ser combinada com a inclinação inicial para produzir uma inclinação média no intervalo de $t = 0$ a 1:

$$\bar{y}' = \frac{3 + 6,402164}{2} = 4,701082$$

Esse resultado pode então ser substituído no corretor [Equação (22.19)], para fornecer uma previsão em $t = 1$:

$$y_1^1 = 2 + 4,701082(1) = 6,701082$$

o que representa um erro relativo percentual de -8,18%. Logo, o método de Heun sem iteração do corretor reduz o valor absoluto do erro por um fator de cerca de 2,4, quando comparado com o método de Euler. Neste ponto, podemos também calcular um erro aproximado como

$$|\varepsilon_a| = \left| \frac{6,701082 - 5}{6,701082} \right| \times 100\% = 25,39\%$$

Agora, a estimativa de y_1 pode ser refinada substituindo-se o novo resultado de volta no lado direito da Equação (22.19), para fornecer:

$$y_1^2 = 2 + \frac{3 + 4e^{0.8(1)} - 0,5(6,701082)}{2} 1 = 6,275811$$

o que representa um erro relativo percentual de 1,31% e um erro aproximado de

$$|\varepsilon_a| = \left| \frac{6,275811 - 6,701082}{6,275811} \right| \times 100\% = 6,776\%$$

A próxima iteração fornece

$$y_1^2 = 2 + \frac{3 + 4e^{0,8(1)} - 0,5(6,275811)}{2} = 6,382129$$

o que representa um erro verdadeiro de 3,03% e um erro aproximado de 1,666%.

O erro aproximado continuará diminuindo à medida que o processo iterativo converge para um resultado final estável. Neste exemplo, após 12 iterações o erro aproximado cai abaixo do critério de parada. Neste ponto, o resultado em $t = 1$ é 6,36087, o que representa um erro relativo verdadeiro de 2,68%. A Tabela 22.2 mostra resultados para o restante dos cálculos juntamente com os resultados para o método de Euler e para o método de Heun sem iteração do corretor.

Uma visão sobre o erro local do método de Heun pode ser obtida reconhecendo-se que ele está relacionado à regra do trapézio. No exemplo anterior, a derivada é uma função tanto da variável dependente y quanto da variável independente t . Para o caso de polinômios, nos quais a EDO é apenas uma função da variável independente, o passo do preditor [Equação (22.18)] não é necessário e o corretor é aplicado apenas uma vez para cada iteração. Nesses casos, a técnica é expressa concisamente como

$$y_{i+1} = y_i + \frac{f(t_i) + f(t_{i+1})}{2}h \quad (22.20)$$

Observe a semelhança entre o segundo termo do lado direito da Equação (22.20) e a regra do trapézio [Equação (19.11)]. A ligação entre os dois métodos pode ser formalmente demonstrada começando com a equação diferencial ordinária

$$\frac{dy}{dt} = f(t) \quad (22.21)$$

Essa equação pode ser resolvida por integração:

$$\int_{y_i}^{y_{i+1}} dy = \int_{t_i}^{t_{i+1}} f(t) dt \quad (22.22)$$

o que fornece

$$y_{i+1} - y_i = \int_{t_i}^{t_{i+1}} f(t) dt \quad (22.23)$$

ou

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t) dt \quad (22.24)$$

Agora, lembre-se de que a regra do trapézio [Equação (19.11)] é definida por

$$\int_{t_i}^{t_{i+1}} f(t) dt = \frac{f(t_i) + f(t_{i+1})}{2} h \quad (22.25)$$

onde $h = t_{i+1} - t_i$. Substituindo a Equação (22.25) na Equação (22.24), obtemos

$$y_{i+1} = y_i + \frac{f(t_i) + f(t_{i+1})}{2} h \quad (22.26)$$

o que é equivalente à Equação (22.20). Por essa razão, o método de Heun é muitas vezes referido como a regra do trapézio.

Como a Equação (22.26) é uma expressão direta da regra do trapézio, o erro de truncamento local é dado por [lembre-se da Equação (19.14)]

$$E_t = -\frac{f''(\xi)}{12} h^3 \quad (22.27)$$

onde ξ está entre t_i e t_{i+1} . Logo, o método é de segunda ordem, porque a segunda derivada da EDO é zero quando a solução verdadeira é quadrática. Além disso, os erros local e global são $O(h^3)$ e $O(h^2)$, respectivamente. Portanto, a diminuição do tamanho do passo faz o erro decrescer a uma taxa mais rápida do que no caso do método de Euler.

22.3.2 O método do ponto médio

A Figura 22.6 ilustra uma outra modificação simples no método de Euler. Chamada de *método do ponto médio* (ou do *polígono melhorado* ou de *Euler modificado*), essa técnica usa o método de Euler para prever um valor y no ponto médio do intervalo (Figura 22.6a):

$$y_{i+1/2} = y_i + f(t_i, y_i) \frac{h}{2} \quad (22.28)$$

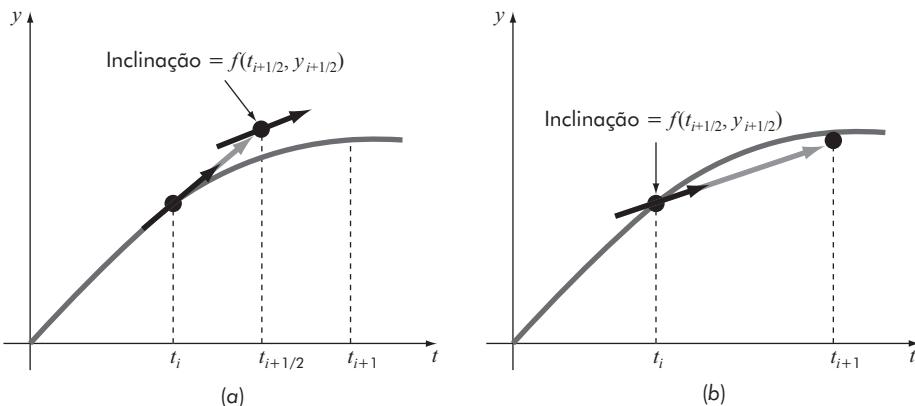


FIGURA 22.6 Descrição gráfica do método do ponto médio. (a) Preditor e (b) corretor.

Então, esse valor previsto é usado para calcular uma inclinação no ponto médio:

$$y'_{i+1/2} = f(t_{i+1/2}, y_{i+1/2}) \quad (22.29)$$

a qual supomos representar uma aproximação válida da inclinação média para o intervalo todo. Essa inclinação é então usada para extrapolar linearmente de t_i a t_{i+1} (Figura 22.6b):

$$y_{i+1} = y_i + f(t_{i+1/2}, y_{i+1/2})h \quad (22.30)$$

Observe que, como y_{i+1} não aparece em ambos os lados, o corretor [Equação (22.30)] não pode ser aplicado iterativamente para melhorar a solução, como foi feito com o método de Heun.

Como em nossa discussão sobre o método de Heun, o método do ponto médio também pode ser ligado às fórmulas de integração de Newton-Cotes. Lembre-se, da Tabela 19.4, que a fórmula de integração aberta de Newton-Cotes mais simples, a qual é chamada de *método do ponto médio*, pode ser representada por

$$\int_a^b f(x) dx \cong (b - a)f(x_1) \quad (22.31)$$

onde x_1 é o ponto médio do intervalo (a, b) . Usando a nomenclatura para o caso presente, ela pode ser expressa como

$$\int_{t_i}^{t_{i+1}} f(t) dt \cong hf(t_{i+1/2}) \quad (22.32)$$

A substituição dessa fórmula na Equação (22.24) fornece a Equação (22.30). Logo, do mesmo modo como o método de Heun pode ser chamado de regra do trapézio, o método do ponto médio leva seu nome da fórmula de integração subjacente na qual é baseado. Este é superior ao método de Euler, porque utiliza uma estimativa da inclinação no ponto médio do intervalo de previsão. Lembre-se, de nossa discussão de derivação numérica na Seção 4.3.4, de que diferenças divididas finitas centradas são aproximações melhores das derivadas do que as versões tanto progressiva quanto regressiva. No mesmo sentido, uma aproximação centrada tal como a Equação (22.29) tem um erro de truncamento local de $O(h^2)$ em comparação com a aproximação progressiva do método de Euler, o qual tem um erro de $O(h)$. Consequentemente, os erros local e global do método do ponto médio são $O(h^3)$ e $O(h^2)$, respectivamente.

22.4 MÉTODOS DE RUNGE-KUTTA

Os métodos de Runge-Kutta (RK) alcançam a acurácia de uma abordagem por série de Taylor sem exigir cálculos de derivadas de ordem superior. Há muitas variações, mas todas podem ser postas na forma geral da Equação (22.4):

$$y_{i+1} = y_i + \phi h \quad (22.33)$$

onde ϕ é chamada *função incremento*, a qual pode ser interpretada como representativa da inclinação em um intervalo. A função incremento pode ser escrita na forma geral

$$\phi = a_1 k_1 + a_2 k_2 + \dots + a_n k_n \quad (22.34)$$

onde os a 's são constantes e os k 's são

$$k_1 = f(t_i, y_i) \quad (22.34a)$$

$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h) \quad (22.34b)$$

$$k_3 = f(t_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h) \quad (22.34c)$$

⋮

$$k_n = f(t_i + p_{n-1} h, y_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + \dots + q_{n-1,n-1} k_{n-1} h) \quad (22.34d)$$

onde os p 's e os q 's são constantes. Observe que os k 's são relações de ocorrência. Isto é, k_1 aparece na equação para k_2 , o qual aparece na equação para k_3 , e assim por diante. Como cada k é um cálculo da função, essa recorrência torna os métodos RK eficientes para cálculos computacionais.

Vários tipos de métodos de Runge-Kutta podem ser deduzidos usando-se um número diferente de termos na função incremento, conforme especificado por n . Observe que o método RK de primeira ordem, com $n = 1$ é, na realidade, o método de Euler. Uma vez que n seja escolhido, os valores para os a 's, os p 's e os q 's são calculados igualando-se a Equação (22.33) a termos da expansão em série de Taylor. Logo, pelo menos para as versões de ordem mais baixa, o número de termos, n , geralmente representa a ordem da abordagem. Por exemplo, na Seção 22.4.1, os métodos RK de segunda ordem usam uma função incremento com dois termos ($n = 2$). Esses métodos de segunda ordem serão exatos se a solução da equação diferencial for quadrática. Além disso, como os termos com h^3 e com grau mais alto foram desprezados durante a dedução, o erro de truncamento local é $O(h^3)$ e o erro global é $O(h^2)$. Na Seção 22.4.2, o método RK de quarta ordem ($n = 4$) é apresentado para o qual o erro de truncamento global é $O(h^4)$.

22.4.1 Métodos de Runge-Kutta de segunda ordem

A versão de segunda ordem da Equação (22.33) é

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2) h \quad (22.35)$$

onde

$$k_1 = f(t_i, y_i) \quad (22.35a)$$

$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h) \quad (22.35b)$$

Os valores para a_1 , a_2 , p_1 e q_{11} são calculados igualando-se a Equação (22.35) à expansão em série de Taylor até os termos de 2º grau. Dessa forma, três equações podem ser deduzidas para calcular as quatro constantes desconhecidas (ver Chapra e Canale, 2010, para detalhes). As três equações são

$$a_1 + a_2 = 1 \quad (22.36)$$

$$a_2 p_1 = 1/2 \quad (22.37)$$

$$a_2 q_{11} = 1/2 \quad (22.38)$$

Como temos três equações com quatro incógnitas, essas equações são ditas indeterminadas. Portanto, devemos escolher um valor para uma das incógnitas para

determinar as outras três. Considere que especifiquemos um valor para a_2 . Então, as Equações (22.36) a (22.38) podem ser resolvidas simultaneamente para

$$a_1 = 1 - a_2 \quad (22.39)$$

$$p_1 = q_{11} = \frac{1}{2a_2} \quad (22.40)$$

Como podemos escolher um número infinito de valores para a_2 , existe um número infinito de métodos RK de segunda ordem. Cada versão fornece exatamente o mesmo resultado se a solução da EDO for quadrática, linear ou constante. Entretanto, elas fornecem resultados diferentes quando (como é o caso típico) a solução é mais complicada. Apresentaremos a seguir três das versões usadas mais frequentemente e preferidas.

Método de Heun sem iteração ($a_2 = 1/2$). Se for considerado que a_2 é $1/2$, as Equações (22.39) e (22.40) podem ser resolvidas por $a_1 = 1/2$ e $p_1 = q_{11} = 1$. Esses parâmetros, quando substituídos na Equação (22.35), fornecem

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2 \right) h \quad (22.41)$$

onde

$$k_1 = f(t_i, y_i) \quad (22.41a)$$

$$k_2 = f(t_i + h, y_i + k_1 h) \quad (22.41b)$$

Observe que k_1 é a inclinação no início do intervalo, e k_2 é a inclinação no final do intervalo. Consequentemente, esse método de segunda ordem de Runge-Kutta é, na realidade, a técnica de Heun sem iteração do corretor.

Método do ponto médio ($a_2 = 1$). Se for considerado que a_2 é 1, então $a_1 = 0$, $p_1 = q_{11} = 1/2$, e a Equação (22.35) se torna

$$y_{i+1} = y_i + k_2 h \quad (22.42)$$

onde

$$k_1 = f(t_i, y_i) \quad (22.42a)$$

$$k_2 = f(t_i + h/2, y_i + k_1 h/2) \quad (22.42b)$$

Esse é o método do ponto médio.

Método de Ralston ($a_2 = 2/3$). Ralston (1962) e Ralston e Rabinowitz (1978) determinaram que a escolha $a_2 = 2/3$ fornece um limitante mínimo para o erro de truncamento para os algoritmos RK de segunda ordem. Para essa versão, $a_1 = 1/3$ e $p_1 = q_{11} = 3/4$, e a Equação (22.35) torna-se

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2 \right) h \quad (22.43)$$

onde

$$k_1 = f(t_i, y_i) \quad (22.43a)$$

$$k_2 = f\left(t_i + \frac{3}{4}h, y_i + \frac{3}{4}k_1 h\right) \quad (22.43b)$$

22.4.2 Método de Runge-Kutta de quarta ordem clássico

Os métodos RK mais populares são os de quarta ordem. Como no caso das abordagens de segunda ordem, existe um número infinito de versões. A forma seguinte é a usada, de modo geral, e, assim, a chamamos de *método RK de quarta ordem clássico*:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h \quad (22.44)$$

onde

$$k_1 = f(t_i, y_i) \quad (22.44a)$$

$$k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right) \quad (22.44b)$$

$$k_3 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h\right) \quad (22.44c)$$

$$k_4 = f(t_i + h, y_i + k_3 h) \quad (22.44d)$$

Observe que para EDOs que são uma função apenas de t , o método RK de quarta ordem clássico é parecido com a regra 1/3 de Simpson; além disso, também é semelhante à abordagem de Heun em termos de desenvolvimento de múltiplas estimativas da inclinação para se chegar a uma inclinação média melhorada no intervalo. Como mostra a Figura 22.7, cada um dos k 's representa uma inclinação. Então, a Equação (22.44) representa uma média ponderada delas para chegar a uma inclinação melhorada.

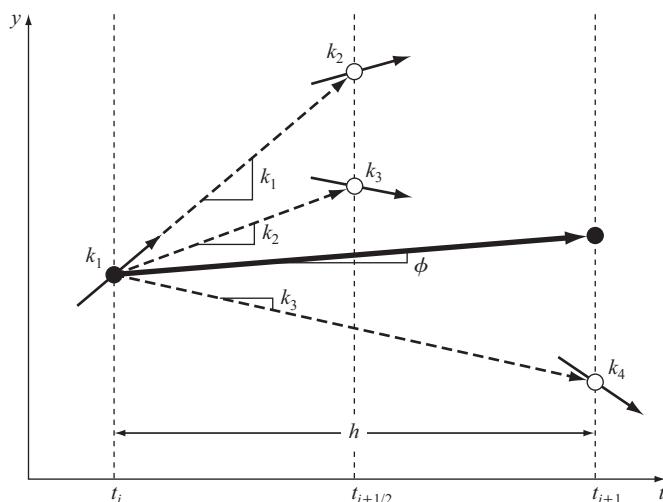


FIGURA 22.7 Descrição gráfica das estimativas de inclinação envolvidas no método RK de quarta ordem.

EXEMPLO 22.3 Método RK de quarta ordem clássico

Use o método RK de quarta ordem clássico para integrar $y' = 4e^{0.8t} - 0.5y$ de $t = 0$ a 1 , com um tamanho de passo de 1 com $y(0) = 2$.

Solução. Nesse caso, a inclinação no início do intervalo é calculada como

$$k_1 = f(0, 2) = 4e^{0.8(0)} - 0.5(2) = 3$$

Esse valor é usado para calcular um valor de y e uma inclinação no ponto médio:

$$y(0.5) = 2 + 3(0.5) = 3.5$$

$$k_2 = f(0.5, 3.5) = 4e^{0.8(0.5)} - 0.5(3.5) = 4.217299$$

Essa inclinação, por sua vez, é usada para calcular um outro valor de y e uma outra inclinação no ponto médio:

$$y(0.5) = 2 + 4.217299(0.5) = 4.108649$$

$$k_3 = f(0.5, 4.108649) = 4e^{0.8(0.5)} - 0.5(4.108649) = 3.912974$$

A seguir, essa inclinação é usada para calcular um valor de y e uma inclinação no final do intervalo:

$$y(1,0) = 2 + 3.912974(1,0) = 5.912974$$

$$k_4 = f(1,0, 5.912974) = 4e^{0.8(1,0)} - 0.5(5.912974) = 5.945677$$

Por fim, as quatro estimativas de inclinação são combinadas para fornecer uma inclinação média. Essa inclinação é então usada para fazer a previsão final no fim do intervalo:

$$\phi = \frac{1}{6} [3 + 2(4.217299) + 2(3.912974) + 5.945677] = 4.201037$$

$$y(1,0) = 2 + 4.201037(1,0) = 6.201037$$

que se compara favoravelmente com a solução verdadeira de 6.194631 ($\varepsilon_t = 0,103\%$).

Certamente é possível desenvolver métodos RK de quinta ordem e de ordem superior. Por exemplo, o método RK de quinta ordem de Butcher (1964) é escrito como

$$y_{i+1} = y_i + \frac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)h \quad (22.45)$$

onde

$$k_1 = f(t_i, y_i) \quad (22.45a)$$

$$k_2 = f\left(t_i + \frac{1}{4}h, y_i + \frac{1}{4}k_1 h\right) \quad (22.45b)$$

$$k_3 = f\left(t_i + \frac{1}{4}h, y_i + \frac{1}{8}k_1 h + \frac{1}{8}k_2 h\right) \quad (22.45c)$$

$$k_4 = f\left(t_i + \frac{1}{2}h, y_i - \frac{1}{2}k_2 h + k_3 h\right) \quad (22.45d)$$

$$k_5 = f\left(t_i + \frac{3}{4}h, y_i + \frac{3}{16}k_1 h + \frac{9}{16}k_4 h\right) \quad (22.45e)$$

$$k_6 = f\left(t_i + h, y_i - \frac{3}{7}k_1 h + \frac{2}{7}k_2 h + \frac{12}{7}k_3 h - \frac{12}{7}k_4 h + \frac{8}{7}k_5 h\right) \quad (22.45f)$$

Observe a similaridade entre o método de Butcher e a regra de Boole na Tabela 19.2. Como esperado, esse método tem um erro de truncamento global de $O(h^5)$.

Embora a versão de quinta ordem forneça mais acurácia, observe que seis cálculos de função são necessários. Lembre-se de que, até as versões de quarta ordem, n cálculos de função são necessários para um método RK de n -ésima ordem. Curiosamente, para ordens mais elevadas que quatro, um ou dois cálculos de função adicionais são necessários. Como os cálculos de função representam a maior parte do tempo computacional, métodos de quinta ordem e superior são geralmente considerados relativamente menos eficientes que as versões de quarta ordem. Essa é uma das principais razões para a popularidade do método RK de quarta ordem.

22.5 SISTEMAS DE EQUAÇÕES

Muitos problemas práticos em engenharia e em ciência exigem a solução de um sistema de equações diferenciais ordinárias simultâneas em vez de uma única equação. Esses sistemas podem ser representados de forma geral por

$$\begin{aligned} \frac{dy_1}{dt} &= f_1(t, y_1, y_2, \dots, y_n) \\ \frac{dy_2}{dt} &= f_2(t, y_1, y_2, \dots, y_n) \\ &\vdots \\ \frac{dy_n}{dt} &= f_n(t, y_1, y_2, \dots, y_n) \end{aligned} \quad (22.46)$$

A solução de tal sistema exige que sejam conhecidas n condições iniciais no valor inicial de t .

Um exemplo é o cálculo da posição e da velocidade do saltador de *bungee jumping* que definimos no início deste capítulo. Para a parte de queda livre do salto, esse problema equivale à resolução do seguinte sistema de EDOs:

$$\frac{dx}{dt} = v \quad (22.47)$$

$$\frac{dv}{dt} = g - \frac{c_d}{m}v^2 \quad (22.48)$$

Se a plataforma estacionária de onde o saltador pula é definida como $x = 0$, as condições iniciais são $x(0) = v(0) = 0$.

22.5.1 Método de Euler

Todos os métodos discutidos neste capítulo para uma única equação podem ser estendidos aos sistemas de EDOs. As aplicações de engenharia podem envolver milhares de equações simultâneas. Em cada caso, o procedimento para resolver um sistema de equações envolve simplesmente a aplicação da técnica de passo único em todas as equações para cada passo, antes de prosseguir para o próximo passo. Isso é bem ilustrado pelo próximo exemplo para o método de Euler.

EXEMPLO 22.4 Resolvendo sistemas de EDOs por meio do método de Euler

Determine a velocidade e a posição do saltador de *bungee jumping* em queda livre por meio do método de Euler. Considere que em $t = 0$, $x = v = 0$, e integre para $t = 10$ s com um tamanho de passo de 2 s. Como feito anteriormente, nos Exemplos 1.1 e 1.2, a aceleração da gravidade é $9,81 \text{ m/s}^2$, e o saltador tem uma massa de 68,1 kg, com um coeficiente de arraste de $0,25 \text{ kg/m}$.

Lembre-se de que a solução analítica para a velocidade é [Equação (1.9)]:

$$v(t) = \sqrt{\frac{gm}{c_d}} \tanh\left(\sqrt{\frac{gc_d}{m}} t\right)$$

Esse resultado pode ser substituído na Equação (22.47), que pode ser integrada para determinar uma solução analítica para a posição como

$$x(t) = \frac{m}{c_d} \ln\left[\cosh\left(\sqrt{\frac{gc_d}{m}} t\right)\right]$$

Use essas soluções analíticas para calcular os erros relativos verdadeiros dos resultados.

Solução. As EDOs podem ser usadas para calcular as inclinações em $t = 0$ como

$$\begin{aligned}\frac{dx}{dt} &= 0 \\ \frac{dv}{dt} &= 9,81 - \frac{0,25}{68,1}(0)^2 = 9,81\end{aligned}$$

O método de Euler é então usado para calcular os valores em $t = 2$ s,

$$\begin{aligned}x &= 0 + 0(2) = 0 \\ v &= 0 + 9,81(2) = 19,62\end{aligned}$$

As soluções analíticas podem ser calculadas como $x(2) = 19,16629$ e $v(2) = 18,72919$. Assim, os erros relativos percentuais são 100% e 4,756%, respectivamente.

O processo pode ser repetido para calcular os resultados em $t = 4$ s como

$$\begin{aligned}x &= 0 + 19,62(2) = 39,24 \\ v &= 19,62 + \left(9,81 - \frac{0,25}{68,1}(19,62)^2\right)2 = 36,41368\end{aligned}$$

Procedendo de maneira semelhante, temos os resultados apresentados na Tabela 22.3.

TABELA 22.3 Posição e velocidade de um saltador de *bungee jumping* em queda livre calculadas numericamente com o método de Euler.

t	$x_{\text{verdadeiro}}$	$v_{\text{verdadeiro}}$	x_{Euler}	v_{Euler}	$\varepsilon_t(x)$	$\varepsilon_t(v)$
0	0	0	0	0	100,00%	4,76%
2	19,1663	18,7292	0	19,6200	100,00%	4,76%
4	71,9304	33,1118	39,2400	36,4137	45,45%	9,97%
6	147,9462	42,0762	112,0674	46,2983	24,25%	10,03%
8	237,5104	46,9575	204,6640	50,1802	13,83%	6,86%
10	334,1782	49,4214	305,0244	51,3123	8,72%	3,83%

Embora o exemplo anterior ilustre como o método de Euler pode ser implementado para sistemas de EDOs, os resultados não são muito acurados por causa do tamanho de passo grande. Além disso, os resultados para a posição são um pouco insatisfatórios, pois x não varia até a segunda iteração. A utilização de um passo menor atenua bastante essas deficiências. Como descrito a seguir, a utilização de soluções de ordem mais elevada fornece bons resultados mesmo com um tamanho de passo relativamente grande.

22.5.2 Métodos de Runge-Kutta

Observe que qualquer um dos métodos RK de ordem superior apresentados neste capítulo pode ser aplicado para sistemas de equações. Porém, é preciso cautela na determinação das inclinações. A Figura 22.7 mostra a maneira adequada de fazer isso para o método de quarta ordem. A princípio, desenvolvem-se inclinações para todas as variáveis no valor inicial. Essas inclinações (um conjunto de k_1 's) são então usadas para fazer previsões da variável independente no ponto médio do intervalo. Os valores no ponto médio são, por sua vez, usados para calcular, neste, um conjunto de inclinações (os k_2 's). Essas novas inclinações são desse modo levadas de volta ao ponto inicial para fazer um outro conjunto de previsões de inclinações no ponto médio (os k_3 's). Estas então são usadas para fazer previsões no ponto final do intervalo, que são utilizadas para desenvolver inclinações no final do intervalo (os k_4 's). Por fim, os k 's são combinados em um conjunto de funções incremento [como na Equação (22.44)] e trazidos de volta ao ponto inicial para fazer a previsão final. O exemplo a seguir ilustra a abordagem.

EXEMPLO 22.5 Resolvendo sistemas de EDOs por meio do método RK de quarta ordem

Use o método RK de quarta ordem para resolver o mesmo problema que abordamos no Exemplo 22.4.

Solução. A princípio, é conveniente expressar as EDOs no formato funcional da Equação (22.46) como

$$\begin{aligned}\frac{dx}{dt} &= f_1(t, x, v) = v \\ \frac{dv}{dt} &= f_2(t, x, v) = g - \frac{c_d}{m} v^2\end{aligned}$$

O primeiro passo para obter a solução é determinar todas as inclinações no início do intervalo:

$$k_{1,1} = f_1(0, 0, 0) = 0$$

$$k_{1,2} = f_2(0, 0, 0) = 9,81 - \frac{0,25}{68,1}(0)^2 = 9,81$$

onde $k_{i,j}$ é o i -ésimo valor de k para a j -ésima variável dependente. A seguir, devemos calcular os primeiros valores de x e v no ponto médio do primeiro passo:

$$x(1) = x(0) + k_{1,1} \frac{h}{2} = 0 + 0 \frac{2}{2} = 0$$

$$v(1) = v(0) + k_{1,2} \frac{h}{2} = 0 + 9,81 \frac{2}{2} = 9,81$$

o que pode ser usado para calcular o primeiro conjunto de inclinações no ponto médio:

$$k_{2,1} = f_1(1, 0, 9,81) = 9,8100$$

$$k_{2,2} = f_2(1, 0, 9,81) = 9,4567$$

Essas são usadas para determinar o segundo conjunto de previsões para o ponto médio:

$$x(1) = x(0) + k_{2,1} \frac{h}{2} = 0 + 9,8100 \frac{2}{2} = 9,8100$$

$$v(1) = v(0) + k_{2,2} \frac{h}{2} = 0 + 9,4567 \frac{2}{2} = 9,4567$$

que podem ser utilizadas para calcular o segundo conjunto de inclinações no ponto médio:

$$k_{3,1} = f_1(1, 9,8100, 9,4567) = 9,4567$$

$$k_{3,2} = f_2(1, 9,8100, 9,4567) = 9,4817$$

Essas são usadas para determinar as previsões no final do intervalo:

$$x(2) = x(0) + k_{3,1}h = 0 + 9,4567(2) = 18,9134$$

$$v(2) = v(0) + k_{3,2}h = 0 + 9,4817(2) = 18,9634$$

que podem ser utilizadas para calcular as inclinações na extremidade final:

$$k_{4,1} = f_1(2, 18,9134, 18,9634) = 18,9634$$

$$k_{4,2} = f_2(2, 18,9134, 18,9634) = 8,4898$$

Os valores de k podem então ser usados para calcular [Equação (22.44)]:

$$x(2) = 0 + \frac{1}{6} [0 + 2(9,8100 + 9,4567) + 18,9634] 2 = 19,1656$$

$$v(2) = 0 + \frac{1}{6} [9,8100 + 2(9,4567 + 9,4817) + 8,4898] 2 = 18,7256$$

Procedendo de forma análoga nos próximos passos, obtemos os valores apresentados na Tabela 22.4. Ao contrário dos resultados obtidos com o método de Euler, as previsões do método RK de quarta ordem estão muito mais próximas dos valores verdadeiros. Além

disso, um valor altamente acurado e diferente de zero é calculado para a posição no primeiro passo.

TABELA 22.4 Posição e velocidade de um saltador de *bungee jumping* calculadas numericamente com o método RK de quarta ordem.

t	$x_{\text{verdadeiro}}$	$v_{\text{verdadeiro}}$	x_{RK4}	v_{RK4}	$\varepsilon_t(x)$	$\varepsilon_t(v)$
0	0	0	0	0	0,004%	0,019%
2	19,1663	18,7292	19,1656	18,7256	0,001%	0,037%
4	71,9304	33,1118	71,9311	33,0995	0,004%	0,051%
6	147,9462	42,0762	147,9521	42,0547	0,000%	0,049%
8	237,5104	46,9575	237,5104	46,9345	0,005%	0,038%
10	334,1782	49,4214	334,1626	49,4027		

22.5.3 Função do MATLAB: rk4sys

A Figura 22.8 mostra um arquivo-M chamado `rk4sys` que usa o método de Runge-Kutta de quarta ordem para resolver um sistema de EDOs. Esse código é semelhante em muitos aspectos à função desenvolvida anteriormente (Figura 22.3) para resolver uma única EDO com o método de Euler. Por exemplo, passa-se como argumento de entrada o nome da função definindo a EDO.

```
function [tp,yp] = rk4sys(dydt,tspan,y0,h,varargin)
% rk4sys: Runge-Kutta de quarta ordem para um sistema de EDOs
% [t,y] = rk4sys(dydt,tspan,y0,h,p1,p2,...): integra
% um sistema de EDOs com o método RK de quarta ordem
% entrada:
% dydt = nome do arquivo-M que avalia as EDOs
% tspan = [ti, tf]; instantes inicial e final com a saída
% gerada em intervalos de h, ou
% = [t0 t1 ... tf]; em instantes específicos
% y0 = valor inicial da variável dependente
% h = tamanho do passo
% p1,p2,... = parâmetros adicionais usados por dydt
% saída:
% t = vetor da variável independente
% y = vetor de solução para a variável dependente
if nargin<4,error('são necessários pelo menos 4 argumentos de
entrada'), end
if any(diff(tspan)<=0),error('tspan não está em ordem crescen-
te'), end
n = length(tspan);
ti = tspan(1);tf = tspan(n);
```

FIGURA 22.8 (continua)

```

if n == 2
    t = (ti:h:tf)'; n = length(t);
    if t(n)<tf
        t(n+1) = tf;
        n = n+1;
    end
else
    t = tspan;
end
tt = ti; y(1,:) = y0;
np = 1; tp(np) = tt; yp(np,:) = y(1,:);
i=1;
while(1)
    tend = t(np+1);
    hh = t(np+1) - t(np);
    if hh>h,hh = h;end
    while(1)
        if tt+hh>tend,hh = tend-tt;end
        k1 = dydt(tt,y(i,:),varargin{:})';
        ymid = y(i,:) + k1.*hh./2;
        k2 = dydt(tt+hh/2,ymid,varargin{:})';
        ymid = y(i,:) + k2.*hh/2;
        k3 = dydt(tt+hh/2,ymid,varargin{:})';
        yend = y(i,:) + k3.*hh;
        k4 = dydt(tt+hh,yend,varargin{:})';
        phi = (k1+2*(k2+k3)+k4)/6;
        y(i+1,:) = y(i,:)+ phi.*hh;
        tt = tt+hh;
        i=i+1;
        if tt>=tend,break,end
    end
    np = np+1; tp(np) = tt; yp(np,:) = y(i,:);
    if tt>=tf,break,end
end

```

FIGURA 22.8 Função do MATLAB para implementar o método RK de quarta ordem para um sistema de EDOs. (continuação)

No entanto, ela tem uma característica adicional que permite gerar uma saída de duas maneiras, dependendo de como a variável de entrada `tspan` é especificada. De modo semelhante ao da Figura 22.3, pode-se definir `tspan = [ti tf]`, onde `ti` e `tf` são os instantes de tempo inicial e final, respectivamente. Dessa maneira, a rotina automaticamente gera valores de saída entre esses limites em intervalos igualmente espaçados de `h`. Alternativamente, nos casos em que se deseja obter resultados em instantes específicos, pode-se definir `tspan= [t0, t1, ..., tf]`. Observe que, em ambos os casos, os valores de `tspan` devem estar em ordem crescente.

Podemos empregar a função `rk4sys` para resolver o mesmo problema como no Exemplo 22.5. A princípio, podemos desenvolver um arquivo-M para armazenar as EDOs:

```

function dy = dydtsys(t, y)
dy = [y(2);9.81-0.25/68.1*y(2)^2];

```

onde $y(1)$ é a distância (x) e $y(2)$ é a velocidade (v). A solução pode ser gerada como

```
>> [t y] = rk4sys(@dydtsys,[0 10],[0 0],2);
>> disp([t' y(:,1) y(:,2)])
    0          0          0
    2.0000    19.1656   18.7256
    4.0000    71.9311   33.0995
    6.0000   147.9521   42.0547
    8.0000   237.5104   46.9345
   10.0000   334.1626   49.4027
```

Também podemos usar `tspan` para gerar resultados em valores específicos da variável independente. Por exemplo,

```
>> tspan=[0 6 10];
>> [t y] = rk4sys(@dydtsys,tspan,[0 0],2);
>> disp([t' y(:,1) y(:,2)])
    0          0          0
    6.0000   147.9521   42.0547
   10.0000   334.1626   49.4027
```

► 22.6 ESTUDO DE CASO MODELOS PREDADOR-PRESA E CAOS

Contextualização. Os engenheiros e os cientistas tratam diversos problemas envolvendo sistemas de equações diferenciais ordinárias não lineares. Este estudo de caso se concentra em duas dessas aplicações. A primeira está relacionada com os modelos predador-presa, que são usados para estudar interações entre espécies. A segunda são equações que derivam da dinâmica dos fluidos, que são utilizadas para simular a atmosfera.

Os *modelos predador-presa* foram desenvolvidos independentemente no começo do século XX pelo matemático italiano Vito Volterra e pelo biólogo norte-americano Alfred J. Lotka. Essas equações são, em geral, denominadas *equações de Lotka-Volterra*. O exemplo mais simples é o seguinte par de EDOs:

$$\frac{dx}{dt} = ax - bxy \quad (22.49)$$

$$\frac{dy}{dt} = -cy + dxy \quad (22.50)$$

onde x e y são os números de presas e predadores, respectivamente, a é a taxa de crescimento das presas, c é a taxa de morte dos predadores e b e d são taxas que caracterizam o efeito da interação predador-presa na morte das presas e no crescimento dos predadores, respectivamente. Os termos multiplicativos (isto é, envolvendo xy) tornam essas equações não lineares.

Um exemplo de um modelo não linear simples baseado na dinâmica dos fluidos atmosféricos são as *equações de Lorenz*, desenvolvidas pelo meteorologista norte-americano Edward Lorenz:

$$\begin{aligned}\frac{dx}{dt} &= -\sigma x + \sigma y \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= -bz + xy\end{aligned}$$

Lorenz desenvolveu essas equações para relacionar a intensidade de movimento do fluido atmosférico x às variações de temperatura y e z nas direções horizontal e vertical, respectivamente. Como no modelo predador-presa, as não linearidades decorrem dos termos multiplicativos simples: xz e xy .

Use métodos numéricos para obter soluções para essas equações. Faça gráficos dos seus resultados para visualizar como as variáveis dependentes variam no tempo. Além disso, faça um gráfico das variáveis dependentes uma em função da outra, para ver se emergem quaisquer padrões interessantes.

Solução. Os seguintes valores de parâmetros podem ser usados para a simulação predador-presa: $a = 1,2$; $b = 0,6$; $c = 0,8$ e $d = 0,3$. Empregue condições iniciais de $x = 2$ e $y = 1$ e integre de $t = 0$ a 30, usando um tamanho de passo de $h = 0,0625$.

A princípio, podemos desenvolver uma função para armazenar as equações diferenciais:

```
function yp = predprey(t, y, a, b, c, d)
yp = [a*y(1)-b*y(1)*y(2);-c*y(2)+d*y(1)*y(2)];
```

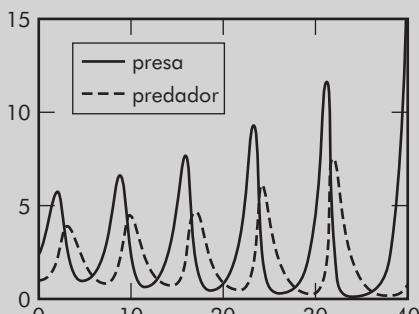
O programa a seguir emprega essa função para gerar soluções por meio do método de Euler e do método RK de quarta ordem. Observe que a função `eulersys` foi baseada na modificação da função `rk4sys` (Figura 22.8). Deixaremos o desenvolvimento dessa função como um exercício para o leitor. Além de exibir a solução como um gráfico em função do tempo (x e y versus t), o programa também gera um gráfico de y versus x . Esses gráficos de *plano de fase* (ou simplesmente *plano de fase*)* são frequentemente úteis na elucidação de características da estrutura subjacente do modelo que podem não ser evidentes a partir de um gráfico em função do tempo.

```
h=0.0625;tspan=[0 40];y0=[2 1];
a=1.2;b=0.6;c=0.8;d=0.3;
[t y] = eulersys(@predprey, tspan, y0, h, a, b, c, d);
subplot(2,2,1);plot(t,y(:,1),t,y(:,2),'--')
legend('presa','predador');title('(a) Gráfico em função do tempo - Euler')
subplot(2,2,2);plot(y(:,1),y(:,2))
title('(b) Plano de fases - Euler')
[t y] = rk4sys(@predprey,tspan,y0,h,a,b,c,d);
subplot(2,2,3);plot(t,y(:,1),t,y(:,2),'--')
title('(c) Gráfico em função do tempo - RK4')
subplot(2,2,4);plot(y(:,1),y(:,2))
title('(d) Plano de fases - RK4')
```

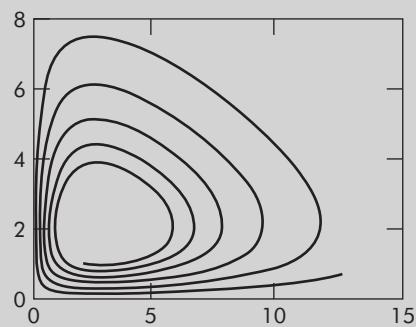
* N. de T.: O espaço n -dimensional cujos eixos coordenados consistem nos eixos formados pelas variáveis de estado é chamado de espaço de estados (ou espaço de fase). No caso particular de um sistema de 2^a ordem, o espaço de fase é bidimensional e conhecido como plano de fase.



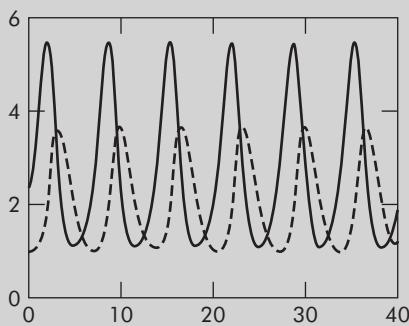
A solução obtida com o método de Euler está apresentada na parte superior da Figura 22.9. O comportamento em função do tempo (Figura 22.9a) indica que as amplitudes das oscilações estão se expandindo. Isso é reforçado pelo plano de fases (Figura 22.9b). Portanto, esses resultados indicam que o método de Euler bruto necessitaria de um passo de tempo muito menor para obter resultados acurados.



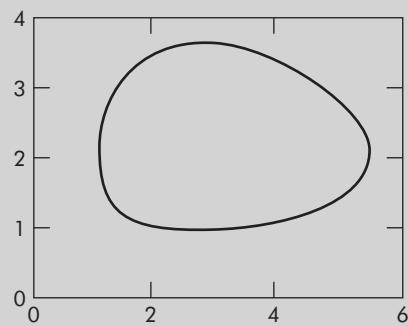
(a) Gráfico em função do tempo – Euler



(b) Plano de fases – Euler



(c) Gráfico em função do tempo – RK4



(d) Plano de fases – RK4

FIGURA 22.9 (a) Gráfico em função do tempo – Euler; (b) Plano de fases – Euler; (c) Gráfico em função do tempo – RK4; (d) Plano de fases – RK4; Solução para o modelo de Lotka-Volterra. Gráficos (a) em função do tempo e (b) no plano de fases obtidos com o método de Euler, e gráficos (c) em função do tempo e (d) no plano de fases obtidos com o método RK4.

Ao contrário, como o método RK4 tem um erro de truncamento muito menor, ele produz bons resultados com o mesmo passo de tempo. Como na Figura 22.9c, um padrão cíclico emerge no tempo. Como a população de predadores é a princípio pequena, o número de presas cresce exponencialmente. Em certo ponto, as presas se tornam tão numerosas que a população de predadores começa a crescer. Às vezes, o crescimento de predadores faz o número de presas diminuir. Essa diminuição, por sua vez, leva a uma diminuição dos predadores. Em certas ocasiões, o processo se repete. Observe que, como esperado, o pico de predadores ocorre um pouco depois do das presas. Além disso, note que o processo tem um período fixo – isto é, ele se repete em um mesmo intervalo de tempo.

Uma representação no plano de fases para a solução RK4 acurada (Figura 22.9d) indica que a interação entre o predador e a presa equivale a uma órbita fechada percorrida no

sentido anti-horário. Curiosamente, existe um *ponto crítico* ou de repouso no centro da órbita. A localização exata desse ponto pode ser determinada igualando-se as Equações (22.49) e (22.50) ao estado estacionário ($dy/dt = dx/dt = 0$) e encontrando $(x, y) = (0, 0)$ e $(c/d, a/b)$. O primeiro é o resultado trivial, em que, se começarmos sem nenhum predador, nada acontecerá. O último é a saída mais interessante, em que, se as condições iniciais forem escolhidas como $x = c/d$ e $y = a/b$, as derivadas serão nulas e as populações permanecerão constantes.

Agora, usaremos a mesma abordagem para investigar as trajetórias das equações de Lorenz com os seguintes valores de parâmetros: $\sigma = 10$, $b = 8/3$ e $r = 28$. Use condições iniciais $x = y = z = 5$ e integre de $t = 0$ a 20. Para esse caso, utilizaremos o método RK de quarta ordem para obter soluções, com um passo de tempo constante de $h = 0,03125$.

Os resultados são bastante diferentes das equações de Lotka-Volterra. Como na Figura 22.10, a variável x parece seguir um padrão quase aleatório de oscilações, variando de valores negativos para valores positivos. As outras variáveis exibem comportamento similar. Entretanto, mesmo que os padrões pareçam aleatórios, a frequência de oscilação e as amplitudes parecem bastante consistentes.

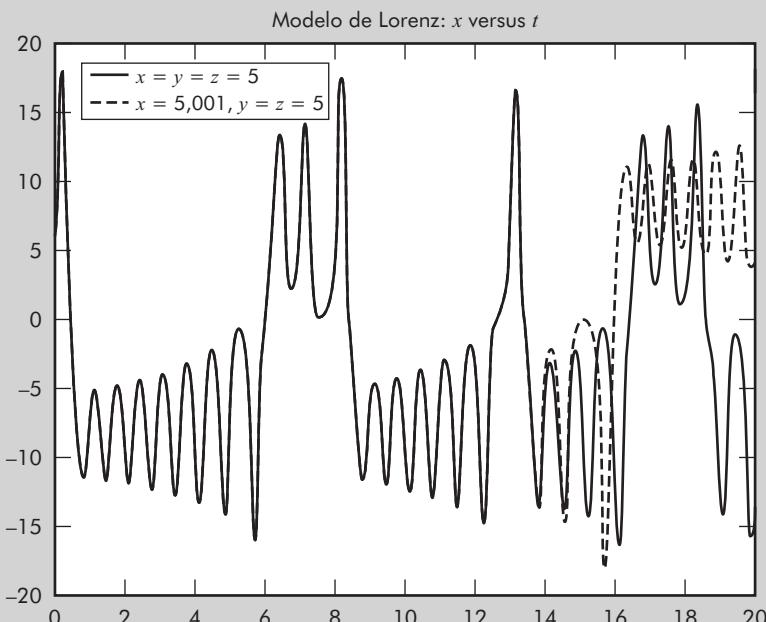


FIGURA 22.10 Representação no domínio do tempo de x versus t para as equações de Lorenz. A linha sólida é para as condições iniciais $(5, 5, 5)$. A linha tracejada é para o caso da condição inicial para x ser ligeiramente perturbada $(5,001, 5, 5)$.

Uma característica interessante dessas soluções pode ser ilustrada variando ligeiramente a condição inicial para x (de 5 para 5,001). Os resultados estão superpostos com a linha tracejada na Figura 22.10. Embora as soluções permaneçam juntas por certo tempo, depois de cerca de $t = 15$, elas divergem significativamente. Dessa forma, podemos ver que as equações de Lorenz são bastante sensíveis às condições iniciais. O termo *caótico* é usado

para descrever essas soluções. Em seu estudo original, isso levou Lorenz à conclusão de que a previsão de tempo de longo alcance pode ser impossível.

A sensibilidade de sistemas dinâmicos a perturbações pequenas de suas condições iniciais é às vezes chamada *efeito borboleta*. A ideia é que o bater de asas de uma borboleta pode provocar pequenas mudanças na atmosfera que, por fim, levam a um fenômeno climático de grande escala, como um tornado.

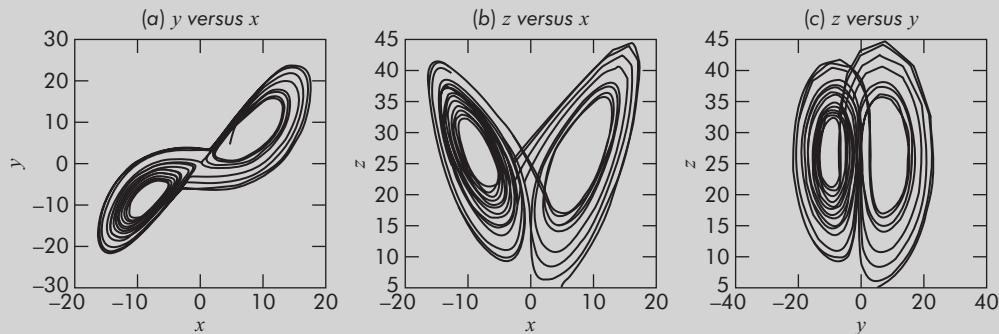


FIGURA 22.11 Representação no plano de fases para as equações de Lorenz. Projeções (a) xy , (b) xz e (c) yz .

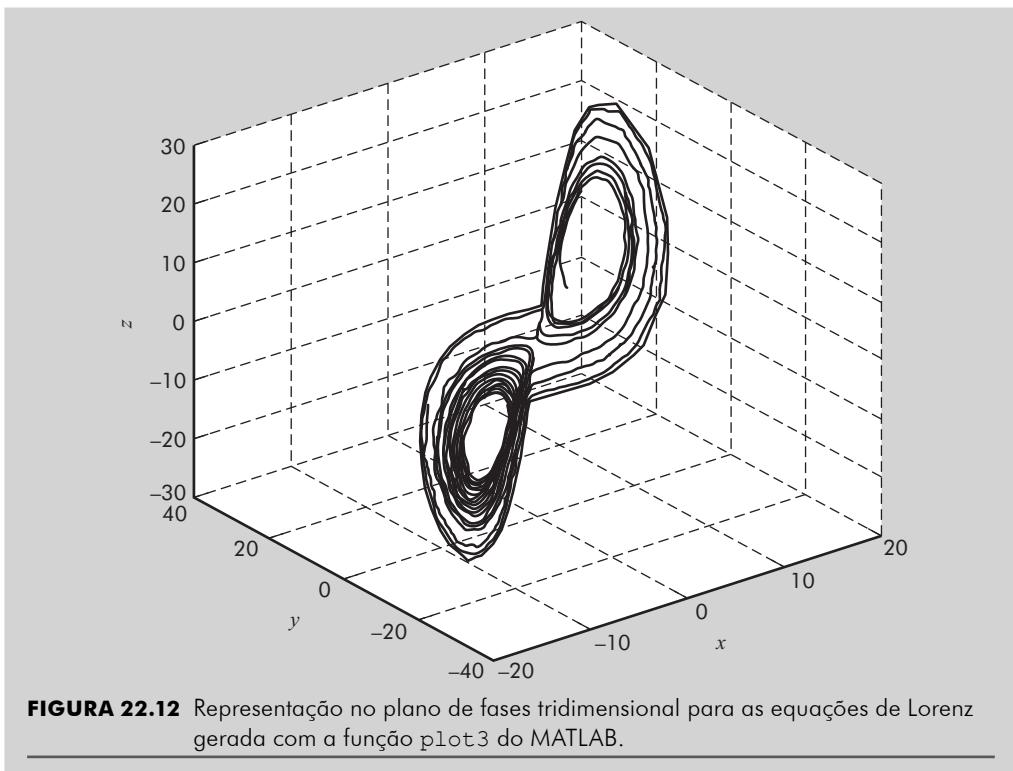
Embora os gráficos no domínio do tempo sejam caóticos, os planos de fase revelam uma estrutura subjacente. Como se trata de três variáveis independentes, podemos gerar projeções. A Figura 22.11 mostra projeções nos planos xy , xz e yz . Observe como é aparente uma estrutura quando observada da perspectiva do espaço de estados. A solução forma órbitas em torno de que parecem ser pontos críticos, que são chamados *atratores estranhos* no jargão dos matemáticos que estudam esses sistemas não lineares.

Além das projeções de duas variáveis, a função `plot3` do MATLAB fornece um meio de gerar diretamente uma representação no plano de fases tridimensional:

```
>> plot3(y(:,1),y(:,2),y(:,3))
>> xlabel('x'); ylabel('y'); zlabel('z'); grid
```

De modo análogo à Figura 22.11, o gráfico tridimensional (Figura 22.12) descreve trajetórias cíclicas em um padrão definido em torno de um par de pontos críticos.

Como uma observação final, a sensibilidade de sistemas caóticos às condições iniciais tem implicações para cálculos numéricicos. Além das condições iniciais propriamente ditas, tamanhos de passo diferentes ou algoritmos diferentes (e, em alguns casos, até mesmo computadores diferentes) podem introduzir diferenças pequenas na solução. De maneira similar à Figura 22.10, essas discrepâncias às vezes levarão a grandes desvios. Alguns dos problemas neste capítulo e no Capítulo 23 foram concebidos para demonstrar esta questão.



►► PROBLEMAS

22.1 Resolva o seguinte problema de valor inicial no intervalo de $t = 0$ a 2 em que $y(0) = 1$. Mostre todos seus resultados em um mesmo gráfico.

$$\frac{dy}{dt} = yt^3 - 1,5y$$

- (a) Analiticamente.
- (b) Usando o método de Euler, com $h = 0,5$ e $0,25$.
- (c) Usando o método do ponto médio, com $h = 0,5$.
- (d) Usando o método RK de quarta ordem, com $h = 0,5$.

22.2 Resolva o seguinte problema no intervalo de $x = 0$ a 1, em que $y(0) = 1$, usando um tamanho de passo de 0,25. Mostre todos seus resultados em um mesmo gráfico.

$$\frac{dy}{dx} = (1 + 4x)\sqrt{y}$$

- (a) Analiticamente.
- (b) Usando o método de Euler.
- (c) Usando o método de Heun sem iteração.

(d) Usando o método de Ralston.

(e) Usando o método RK de quarta ordem.

22.3 Resolva o seguinte problema no intervalo de $t = 0$ a 2, em que $y(0) = 1$, usando um tamanho de passo de 0,5. Mostre todos seus resultados em um mesmo gráfico.

$$\frac{dy}{dt} = -2y + t^2$$

Obtenha suas soluções com (a) o método de Heun, sem iteração do corretor, (b) o método de Heun, com iteração do corretor, até que $\varepsilon_s < 0,1\%$, (c) o método do ponto médio, e (d) o método de Ralston.

22.4 O crescimento de populações de organismos tem várias aplicações na engenharia e na ciência. Um dos modelos mais simples considera que a taxa de variação da população p é proporcional à população em qualquer tempo t :

$$\frac{dp}{dt} = k_g p \quad (\text{P22.4.1})$$

onde k_g é a taxa de crescimento. A população mundial em milhões de 1950 até 2000 era

<i>t</i>	1950	1955	1960	1965	1970	1975
<i>p</i>	2.560	2.780	3.040	3.350	3.710	4.090
<i>t</i>	1980	1985	1990	1995	2000	
<i>p</i>	4.450	4.850	5.280	5.690	6.080	

- (a) Considerando que a Equação (P22.4.1) é válida, use os dados de 1950 até 1970 para estimar k_g .
(b) Use o método RK de quarta ordem com os resultados de (a) para simular a população mundial de 1950 a 2050, com um tamanho de passo de 5 anos. Mostre os resultados de simulação com os dados em um gráfico.

22.5 Embora o modelo no Problema 22.4 funcione adequadamente quando o crescimento da população é ilimitado, ele falha quando fatores como escassez de alimentos, poluição e falta de espaço inibem o crescimento. Nesses casos, a taxa de crescimento não é uma constante, mas pode ser formulada como

$$k_g = k_{gm}(1 - p/p_{\max})$$

onde k_{gm} é a taxa de crescimento máxima sob condições ilimitadas, p é a população e p_{\max} é a população máxima. Observe que p_{\max} é algumas vezes chamada *capacidade de carga*. Assim, para uma baixa densidade populacional $p \ll p_{\max}$, $k_g \rightarrow k_{gm}$. À medida que p se aproxima de p_{\max} , a taxa de crescimento aproxima-se de zero. Usando essa formulação da taxa de crescimento, a taxa de variação de uma população pode ser modelada como

$$\frac{dp}{dt} = k_{gm}(1 - p/p_{\max})p$$

Essa equação é conhecida como *modelo logístico*. A solução analítica desse modelo é

$$p = p_0 \frac{p_{\max}}{p_0 + (p_{\max} - p_0)e^{-k_{gm}t}}$$

Simule a população mundial de 1950 a 2050 usando (a) a solução analítica, e (b) o método RK de quarta ordem com um tamanho de passo de 5 anos. Empregue as seguintes condições iniciais e valores de parâmetros para sua simulação: p_0 (em 1950) = 2.560 milhões de pessoas, $k_{gm} = 0,026/\text{ano}$ e $p_{\max} = 12.000$ milhões de pessoas. Mostre seus resultados em um gráfico com os dados do Problema 22.4.

22.6 Considere que um projétil seja lançado para cima da superfície da Terra e que a única força que está agindo no objeto seja a força gravitacio-

nal para baixo. Nessas condições, um balanço de forças pode ser usado para deduzir que

$$\frac{dv}{dt} = -g(0) \frac{R^2}{(R + x)^2}$$

onde v é a velocidade para cima (m/s), t é o tempo (s), x é a altitude (m) medida para cima a partir da superfície da Terra, $g(0)$ é a aceleração gravitacional na superfície da Terra ($\approx 9,81 \text{ m/s}^2$) e R é o raio da Terra ($\approx 6,37 \times 10^6 \text{ m}$). Percebendo que $dx/dt = v$, use o método de Euler para determinar a altura máxima que seria atingida se $v(t) = 1500 \text{ m/s}$.

22.7 Resolva o seguinte par de EDOs no intervalo de $t = 0$ a 0,4 usando um tamanho de passo de 0,1. As condições iniciais são $y(0) = 2$ e $z(0) = 4$. Obtenha sua solução com (a) o método de Euler e (b) o método RK de quarta ordem. Mostre seus resultados em um gráfico.

$$\begin{aligned} \frac{dy}{dt} &= -2y + 5e^{-t} \\ \frac{dz}{dt} &= -\frac{yz^2}{2} \end{aligned}$$

22.8 A *equação de van der Pol* é um modelo de um circuito eletrônico que surgiu nos tempos de tubos de vácuo:

$$\frac{d^2y}{dt^2} - (1 - y^2) \frac{dy}{dt} + y = 0$$

Dadas as condições iniciais, $y(0) = y'(0) = 0$, resolva essa equação de $t = 0$ a 10, por meio do método de Euler, com um tamanho de passo de (a) 0,25 e (b) 0,125. Trace ambas as soluções no mesmo gráfico.

22.9 Dadas as condições iniciais $y(0) = 1$ e $y'(0) = 0$, resolva o seguinte problema de valor inicial de $t = 0$ a 4:

$$\frac{d^2y}{dt^2} + 4y = 0$$

Obtenha sua solução com (a) o método de Euler e (b) o método RK de quarta ordem. Em ambos os casos, use um tamanho de passo de 0,1. Trace ambas as soluções no mesmo gráfico com a solução exata $y = \cos 2t$.

22.10 Desenvolva um arquivo-M para resolver uma única EDO com o método de Heun com iteração. Projete o arquivo-M de modo que ele gere um gráfico dos resultados. Teste seu programa utilizando-o para determinar a população como descrito no Problema 22.5. Empregue um tamanho de passo de 5 anos e itere o corretor até que $\varepsilon_s < 0,1\%$.

22.11 Desenvolva um arquivo-M para resolver uma única EDO com o método do ponto médio. Projete o arquivo-M de modo que ele gere um gráfico dos resultados. Teste seu programa utilizando-o para determinar a população como descrito no Problema 22.5. Empregue um tamanho de passo de 5 anos.

22.12 Desenvolva um arquivo-M para resolver uma única EDO com o método RK de quarta ordem. Projete o arquivo-M de modo que ele gere um gráfico dos resultados. Teste seu programa utilizando-o para resolver o Problema 22.2. Empregue um tamanho de passo de 0,1.

22.13 Desenvolva um arquivo-M para resolver um sistema de EDOs com o método de Euler. Projete seu programa de modo que ele gere um gráfico dos resultados. Teste seu programa utilizando-o para resolver o Problema 22.7 com um tamanho de passo de 0,25.

22.14 O Isle Royale National Park é um arquipélago de 210 milhas quadradas composto por uma única ilha grande e por muitas ilhas pequenas no Lago Superior. Os alces chegaram em torno de 1900 e, por volta de 1930, sua população se aproximou de 3.000, destruindo a vegetação. Em 1949, lobos cruzaram uma ponte de gelo a partir de Ontário. A partir do final da década de 1950, o número de alces e lobos foi monitorado.

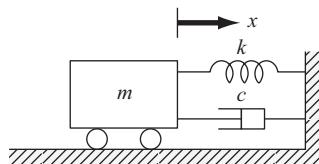


FIGURA P22.15

(a) Integre as equações de Lotka-Volterra (Seção 22.6) de 1960 até 2020 usando os seguintes valores de coeficientes: $a = 0,23$, $b = 0,0133$, $c = 0,4$ e $d = 0,0004$. Compare sua simulação com os dados usando um gráfico em função do tempo e determine a soma dos quadrados dos resíduos entre seu modelo e os dados, tanto para os alces quanto para os lobos.

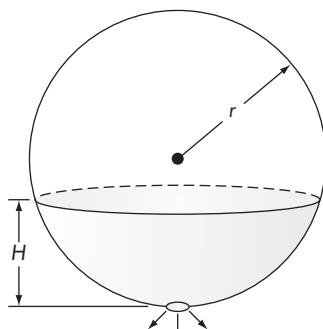
(b) Trace o plano de fases de sua solução.

22.15 O movimento de um sistema massa-mola amortecido (Figura P22.15) é descrito pela seguinte equação diferencial ordinária:

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

onde x é o deslocamento a partir da posição de equilíbrio (m), t é o tempo (s), $m = 20\text{ kg}$ é massa e c é o coeficiente de amortecimento ($\text{N} \cdot \text{s/m}$). O coeficiente de amortecimento c toma os três valores

Ano	Alces	Lobos	Ano	Alces	Lobos	Ano	Alces	Lobos
1959	563	20	1975	1.355	41	1991	1.313	12
1960	610	22	1976	1.282	44	1992	1.590	12
1961	628	22	1977	1.143	34	1993	1.879	13
1962	639	23	1978	1.001	40	1994	1.770	17
1963	663	20	1979	1.028	43	1995	2.422	16
1964	707	26	1980	910	50	1996	1.163	22
1965	733	28	1981	863	30	1997	500	24
1966	765	26	1982	872	14	1998	699	14
1967	912	22	1983	932	23	1999	750	25
1968	1.042	22	1984	1.038	24	2000	850	29
1969	1.268	17	1985	1.115	22	2001	900	19
1970	1.295	18	1986	1.192	20	2002	1.100	17
1971	1.439	20	1987	1.268	16	2003	900	19
1972	1.493	23	1988	1.335	12	2004	750	29
1973	1.435	24	1989	1.397	12	2005	540	30
1974	1.467	31	1990	1.216	15	2006	450	30

**FIGURA P22.16** Tanque esférico.

de 5 (subamortecido), 40 (criticamente amortecido) e 200 (superamortecido). A constante da mola $k = 20 \text{ N/m}$. A velocidade inicial é zero e o deslocamento inicial é $x = 1$. Resolva essa equação usando um método numérico no período de tempo $0 \leq t \leq 15 \text{ s}$. Faça um gráfico do deslocamento em função do tempo para cada um dos três valores de coeficiente de amortecimento.

22.16 Um tanque esférico tem um orifício circular no fundo pelo qual o líquido escoa para fora (Figura P22.16). A vazão pelo buraco pode ser estimada por

$$Q_{\text{fora}} = CA \sqrt{2gh}$$

onde Q_{fora} é o escoamento para fora (m^3/s), C é um coeficiente obtido empiricamente, A é a área do orifício (m^2), g é a constante gravitacional ($= 9,81 \text{ m/s}^2$) e h é a profundidade do líquido no tanque. Use um dos métodos numéricos descritos neste capítulo para determinar quanto tempo levará para que a água escoe para fora de um tanque de 3 m de diâmetro com uma altura inicial de 2,75 m. Observe que o orifício tem um diâmetro de 3 m e $C = 0,55$.

22.17 Na investigação de homicídio ou de morte accidental, em geral, é importante determinar o momento da morte. De observações experimentais, sabe-se que a temperatura da superfície de um objeto varia a uma taxa proporcional à diferença entre a temperatura do objeto e a do meio que o cerca ou a temperatura ambiente. Isso é conhecido como a lei de resfriamento de Newton. Assim, se $T(t)$ for a temperatura do objeto no instante t e T_a for a temperatura constante do ambiente:

$$\frac{dT}{dt} = -K(T - T_a)$$

onde $K > 0$ é uma constante de proporcionalidade.

Considere que, no instante $t = 0$, um corpo seja descoberto e que sua temperatura medida seja T_0 ; que no instante de sua morte, a temperatura do corpo, T_d , tinha o valor normal de 37°C ; que a temperatura do corpo, quando foi descoberto, era de $29,5^\circ\text{C}$ e que duas horas depois era de $23,5^\circ\text{C}$. A temperatura ambiente é 20°C .

- (a) Determine K e o momento da morte.
- (b) Resolva a EDO numericamente e trace seus resultados.

22.18 A reação $A \rightarrow B$ ocorre em dois reatores em série. Os reatores estão bem misturados, mas não estão no estado estacionário. O balanço de massa para cada reator de tanque agitado é mostrado a seguir:

$$\begin{aligned}\frac{dCA_1}{dt} &= \frac{1}{\tau}(CA_0 - CA_1) - kCA_1 \\ \frac{dCB_1}{dt} &= -\frac{1}{\tau}CB_1 + kCA_1 \\ \frac{dCA_2}{dt} &= \frac{1}{\tau}(CA_1 - CA_2) - kCA_2 \\ \frac{dCB_2}{dt} &= \frac{1}{\tau}(CB_1 - CB_2) + kCA_2\end{aligned}$$

onde CA_0 é a concentração de A na entrada do primeiro reator, CA_1 é a concentração de A na saída do primeiro reator (e na entrada do segundo), CA_2 é a concentração de A na saída do segundo reator, CB_1 é a concentração de B na saída do primeiro reator (e na entrada do segundo), CB_2 é concentração de B no segundo reator, τ é o tempo de residência para cada reator e k é a taxa constante para a reação de A produzir B . Se CA_0 for igual a 20, encontre as concentrações de A e B em ambos os reatores durante os primeiros 10 minutos de operação. Use $k = 0,12/\text{min}$ e $\tau = 5 \text{ min}$ e considere que as condições iniciais de todas as variáveis dependentes sejam zero.

22.19 Um reator de batelada pode ser descrito pelas seguintes equações:

$$\begin{aligned}\frac{dC}{dt} &= -e^{(-10/(T+273))} C \\ \frac{dT}{dt} &= 1.000e^{(-10/(T+273))} C - 10(T - 20)\end{aligned}$$

onde C é a concentração do reagente e T é a temperatura do reator. A princípio, o reator está a 16°C e tem uma concentração do reagente C de $1,0 \text{ g mol/L}$. Encontre a concentração e a temperatura do reator como uma função do tempo.

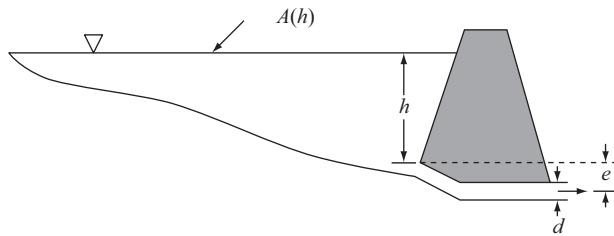


FIGURA P22.21

22.20 A seguinte equação pode ser usada para modelar a deflexão do mastro da vela de um barco sujeito à força do vento:

$$\frac{d^2y}{dz^2} = \frac{f(z)}{2EI}(L - z)^2$$

onde $f(z)$ é a força do vento, E é o módulo de elasticidade, L é o comprimento do mastro e I é o momento de inércia. Considere que a força varia com a altura de acordo com

$$f(z) = \frac{200z}{5+z} e^{-2z/30}$$

Calcule a deflexão se $y = 0$ e $dy/dz = 0$ em $z = 0$. Use valores de parâmetros de $L = 30$, $E = 1,3 \times 10^8$ e $I = 0,05$ para seus cálculos.

22.21 Um reservatório de água é drenado por um tubo, como mostra a Figura P22.21. Sob diversas hipóteses simplificadoras, a seguinte equação diferencial descreve como a profundidade varia com o tempo:

$$\frac{dh}{dt} = -\frac{\pi d^2}{4A(h)} \sqrt{2g(h+e)}$$

onde h é a profundidade (m), t é o tempo (s), d é o diâmetro do tubo (m), $A(h)$ é a área da superfície do reservatório como uma função da profundidade (m^2), g é a constante gravitacional ($= 9,81 \text{ m/s}^2$) e e é a profundidade da saída do tubo abaixo do fundo do reservatório (m). Com base na seguinte tabela da área-profundidade, resolva essa equação diferencial para determinar quanto tempo leva para o reservatório esvaziar-se, dado que $h(0) = 6 \text{ m}$, $d = 0,25 \text{ m}$ e $e = 1 \text{ m}$.

$h, (\text{m})$	6	5	4	3	2	1	0
$A(h), 10^4 \text{ m}^2$	1,17	0,97	0,67	0,45	0,32	0,18	0

22.22 Engenheiros e cientistas usam modelos massa-mola para desenvolver uma percepção da dinâmica de estruturas sob a influência de distúrbios, tais como terremotos. A Figura P22.22 mostra essa representação para um edifício de três andares.

Nesse caso, a análise é limitada ao movimento horizontal das estruturas. Utilizando a segunda lei de Newton, os balanços de força podem ser escritos para esse sistema como

$$\begin{aligned}\frac{d^2x_1}{dt^2} &= -\frac{k_1}{m_1}x_1 + \frac{k_2}{m_1}(x_2 - x_1) \\ \frac{d^2x_2}{dt^2} &= \frac{k_2}{m_2}(x_1 - x_2) + \frac{k_3}{m_2}(x_3 - x_2) \\ \frac{d^2x_3}{dt^2} &= \frac{k_3}{m_3}(x_2 - x_3)\end{aligned}$$

Simule a dinâmica dessa estrutura de $t = 0$ a 20 s , dada a condição inicial que a velocidade do andar térreo é $dx_1/dt = 1 \text{ m/s}$ e todos os outros valores iniciais de deslocamentos e velocidades são zero. Apresente seus resultados como dois gráficos em função do tempo (a) dos deslocamentos e (b) das velocidades. Além disso, desenvolva um gráfico de plano de fase tridimensional dos deslocamentos.

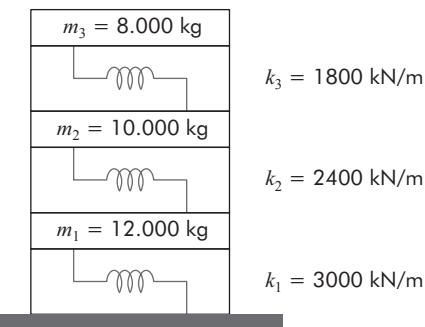


FIGURA P22.22

22.23 Repita as mesmas simulações como na Seção 22.6 para as equações de Lorenz, porém gere as soluções com o método do ponto médio.

22.24 Faça as mesmas simulações como na Seção 22.6 para as equações de Lorenz, porém use um valor de $r = 99,96$. Compare seus resultados com aqueles obtidos na Seção 22.6.

Métodos adaptativos e sistemas rígidos

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar métodos mais avançados para resolução de problemas de valor inicial para equações diferenciais ordinárias. Após esta leitura completa, você será capaz de:

- ✓ Entender como os métodos de Runge-Kutta Fehlberg usam métodos RK de ordens diferentes para fornecer estimativas de erro que são empregadas para ajustar o tamanho do passo.
- ✓ Habituar-se com funções nativas do MATLAB para resolução de EDOs.
- ✓ Aprender como ajustar as opções para os solucionadores de EDOs do MATLAB.
- ✓ Aprender como passar parâmetros para os solucionadores de EDOs do MATLAB.
- ✓ Entender a diferença entre métodos de passo único e de passo múltiplo para resolução de EDOs.
- ✓ Compreender o que é rigidez e suas implicações para resolução de EDOs.

23.1 MÉTODOS ADAPTATIVOS DE RUNGE-KUTTA

Até este ponto, apresentamos métodos para resolver EDOs que empregam um tamanho de passo constante. Para um número significativo de problemas, isso pode representar uma limitação séria. Por exemplo, considere que estejamos integrando uma EDO com uma solução do tipo descrito na Figura 23.1. Para a maior parte do intervalo, a solução varia gradualmente. Esse comportamento sugere que um tamanho de passo razoavelmente grande pode ser utilizado para obter resultados adequados. Entretanto, para uma região localizada entre $x = 1,75$ e $x = 2,25$, a solução passa por uma variação abrupta. A consequência prática de lidar com essas funções é que é necessário um tamanho de passo muito pequeno para capturar adequadamente o comportamento impulsivo. Se fosse utilizado um algoritmo com tamanho de passo constante, o pequeno tamanho do passo necessário para a região de variação abrupta teria de ser aplicado em todo o cálculo. Como consequência, um tamanho de passo muito menor do que o necessário – e, portanto, muito mais cálculos – seria desperdiçado na região de variação gradual.

Os algoritmos que ajustam automaticamente o tamanho do passo podem evitar esses exageros e, desse modo, apresentam grande vantagem. Como eles se “adaptam” à trajetória da solução, afirmamos que têm *controle adaptativo do tamanho do passo*. A implementação dessas abordagens exige que uma estimativa do erro de truncamento local seja obtida em cada passo, que pode servir de base para aumentar ou para diminuir o tamanho do passo.

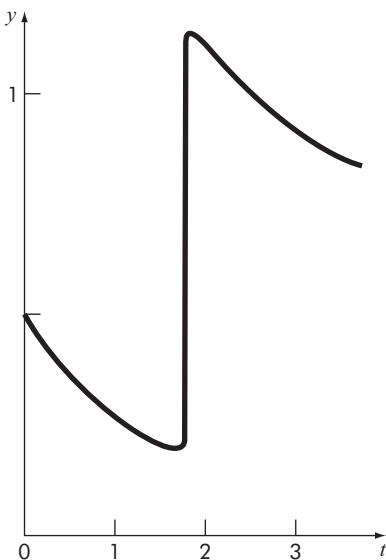


FIGURA 23.1 Exemplo de uma solução de uma EDO que exibe uma variação abrupta. Um ajuste automático no tamanho do passo tem grandes vantagens nesses casos.

Antes de continuar, ressaltamos que, além de resolver EDOs, os métodos descritos neste capítulo também podem ser usados para calcular integrais definidas. O cálculo da integral definida

$$I = \int_a^b f(x) dx$$

é equivalente à resolução da equação diferencial

$$\frac{dy}{dx} = f(x)$$

para determinar $y(b)$ dada a condição inicial $y(a) = 0$. Logo, as técnicas a seguir podem ser usadas para calcular de maneira eficiente integrais definidas que envolvem funções suaves em geral, mas que exibem regiões de variação abrupta.

Há duas abordagens principais para incorporar controle adaptativo do tamanho do passo em métodos de passo único. A *divisão do passo pela metade* implica fazer cada passo duas vezes, uma como um passo completo e depois como dois meios passos. A diferença nos dois resultados representa uma estimativa do erro de truncamento local. O tamanho do passo pode então ser ajustado com base nessa estimativa de erro.

Na segunda abordagem, chamada *métodos RK embutidos*, o erro de truncamento local é estimado como a diferença entre duas previsões que usam métodos RK de ordens diferentes. Esses são atualmente os métodos escolhidos, porque são mais eficientes do que a abordagem de divisão do passo na metade.

Os métodos embutidos foram desenvolvidos primeiro por Fehlberg; por isso, são chamados às vezes de *métodos RK-Fehlberg*. À primeira vista, a ideia de utilizar

duas predições de diferentes ordens pode parecer muito cara computacionalmente. Por exemplo, previsões de quarta e quinta ordens equivalem a um cálculo de dez valores de função no total por passo [lembre-se das Equações (22.24) e (22.45)]. Fehlberg contornou esse problema de forma inteligente deduzindo um método RK de quinta ordem que emprega a maior parte dos mesmos cálculos de função necessários para o método RK de quarta ordem que o acompanha. Assim, a abordagem fornece uma estimativa de erro com base em apenas seis cálculos de função!

23.1.1 Funções nativas do MATLAB para sistemas não rígidos

Desde que Fehlberg originalmente desenvolveu sua abordagem, outras abordagens ainda melhores foram desenvolvidas. Várias dessas estão disponíveis como funções nativas no MATLAB.

ode23. A função `ode23` emprega o algoritmo BS23 (Bogacki e Shampine, 1989; Shampine, 1994), que emprega simultaneamente fórmulas de segunda e terceira ordens para resolver a EDO e fazer estimativas dos erros para ajuste do tamanho do passo. As fórmulas para avançar a solução são

$$y_{i+1} = y_i + \frac{1}{9}(2k_1 + 3k_2 + 4k_3)h \quad (23.1)$$

onde

$$k_1 = f(t_i, y_i) \quad (23.1a)$$

$$k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right) \quad (23.1b)$$

$$k_3 = f\left(t_i + \frac{3}{4}h, y_i + \frac{3}{4}k_2 h\right) \quad (23.1c)$$

O erro é estimado como

$$E_{i+1} = \frac{1}{72}(-5k_1 + 6k_2 + 8k_3 - 9k_4)h \quad (23.2)$$

onde

$$k_4 = f(t_{i+1}, y_{i+1}) \quad (23.2a)$$

Observe que, embora pareça haver quatro cálculos de função, existem apenas três, porque depois do primeiro passo, a constante k_1 para o passo presente será a constante k_4 do passo anterior. Assim, a abordagem produz uma predição e uma estimativa de erro com base em três cálculos em vez de cinco, que normalmente resultaria da utilização em conjunto das fórmulas RK de segunda (dois cálculos) e terceira (três cálculos) ordens.

Após cada passo, o erro é verificado para determinar se ele está dentro de uma tolerância desejada. Se estiver, o valor de y_{i+1} é aceito e k_4 torna-se k_1 para o próximo passo. Se o erro é muito grande, o passo é repetido com tamanhos de passo reduzidos até o erro estimado satisfazer

$$E \leq \max(\text{RelTol} \times |y|, \text{AbsTol}) \quad (23.3)$$

onde RelTol é a tolerância relativa (padrão = 10^{-3}) e AbsTol é a tolerância absoluta (padrão = 10^{-6}). Observe que o critério para o erro relativo usa uma fração em vez de um erro relativo percentual, como fizemos em várias ocasiões antes deste ponto.

ode45. A função `ode45` usa um algoritmo desenvolvido por Dormand e Prince (1980), que usa simultaneamente fórmulas RK de quarta e quinta ordens para resolver a EDO e fazer estimativas do erro para ajuste do tamanho do passo. O MATLAB recomenda que a `ode45` é a melhor função para aplicar como uma “primeira tentativa” para a maioria dos problemas.

ode113. A função `ode113` usa um método Adams-Bashforth-Moulton de ordem variável. Ela é útil para tolerâncias de erro rigorosas ou funções de EDO computacionalmente intensas. Observe que esse é um método de passo múltiplo, como será descrito mais à frente na Seção 23.2.

Essas funções podem ser chamadas de várias maneiras diferentes. A abordagem mais simples é

```
[t, y] = ode45(@odefun, tspan, y0)
```

onde `y` é o arranjo da solução em que cada coluna é uma das variáveis dependentes e cada linha corresponde a um instante de tempo no vetor coluna `t`, `odefun` é o nome da função retornando um vetor coluna dos lados direitos das equações diferenciais, `tspan` especifica o intervalo de integração e `y0` é um vetor que contém os valores iniciais.

Observe que `tspan` pode ser formulado de duas maneiras. Primeiro, se ele é definido como um vetor de dois valores,

```
tspan = [ti tf];
```

a integração é realizada de `ti` a `tf`. Segundo, para obter soluções em instantes de tempo específicos `t0, t1, ..., tn` (todos aumentando ou todos diminuindo), use

```
tspan = [t0 t1 ... tn];
```

A seguir temos um exemplo de como a função `ode45` pode ser usada para resolver uma EDO única, $y' = 4e^{0.8t} - 0.5y$ de $t = 0$ a 4, com uma condição inicial de $y(0) = 2$. Lembre-se, do Exemplo 22.1, que a solução analítica em $t = 4$ é 75,33896. Representando a EDO como uma função anônima, a `ode45` pode ser usada para gerar o mesmo resultado numericamente como

```
>> dydt=@(t,y) 4*exp(0.8*t)-0.5*y;
>> [t,y]=ode45(dydt,[0 4],2);
>> y(length(t))
ans =
    75.3390
```

Como descrito no exemplo seguinte, a EDO geralmente é armazenada em seu próprio arquivo-M quando se trabalha com sistemas de equações.

EXEMPLO 23.1 Utilizando o MATLAB para resolver um sistema de EDOs

Use a função `ode45` para resolver o seguinte conjunto de EDOs não lineares de $t = 0$ a 20:

$$\frac{dy_1}{dt} = 1,2y_1 - 0,6y_1y_2 \quad \frac{dy_2}{dt} = -0,8y_2 + 0,3y_1y_2$$

onde $y_1 = 2$ e $y_2 = 1$ em $t = 0$. Essas equações são conhecidas como *equações predador-presa*.

Solução. Antes de obter uma solução com o MATLAB, devemos criar uma função para calcular o lado direito das EDOs. Uma maneira de fazer isso é criar um arquivo-M como em

```
function yp = predprey(t,y)
yp = [1.2*y(1)-0.6*y(1)*y(2);-0.8*y(2)+0.3*y(1)*y(2)];
```

Esse arquivo-M foi salvo com o nome `predprey.m`.

Depois, entre com os comandos a seguir para especificar o intervalo de integração e as condições iniciais:

```
>> tspan = [0 20];
>> y0 = [2, 1];
```

A função pode então ser chamada por

```
>> [t,y] = ode45(@predprey, tspan, y0);
```

Esse comando resolverá a equação diferencial em `predprey.m` no intervalo definido por `tspan` usando a condição inicial encontrada em `y0`. Os resultados podem ser exibidos simplesmente digitando:

```
>> plot(t,y)
```

que produz a Figura 23.2.

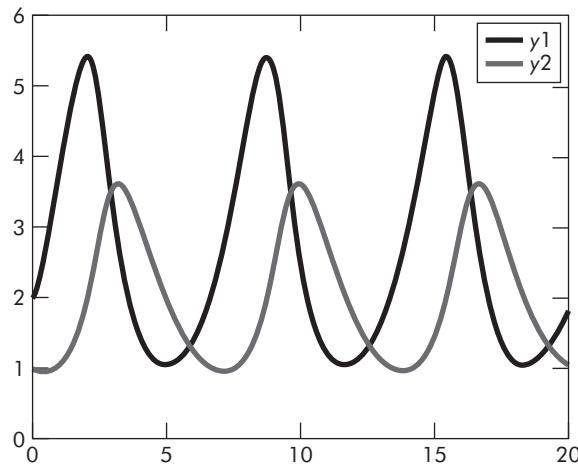


FIGURA 23.2 Solução do modelo predador-presa obtida com o MATLAB.

Além de um gráfico em função do tempo, também é instrutivo gerar um *gráfico de plano de fase* – ou seja, um gráfico das variáveis dependentes, uma *versus* a outra, obtido por

```
>> plot(y(:,1),y(:,2))
```

que produz a Figura 23.3.

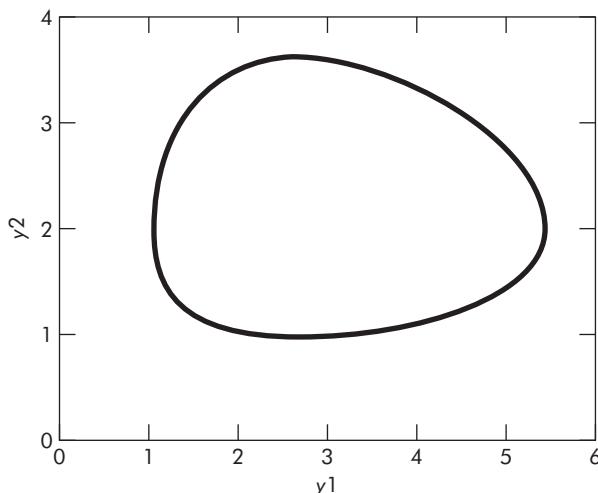


FIGURA 23.3 Plano de fase do modelo predador-presa obtido com o MATLAB.

Como no exemplo anterior, o solucionador do MATLAB usa parâmetros padrões (default) para controlar vários aspectos da integração. Além disso, também não há controle sobre os parâmetros das equações diferenciais. Para ter controle sobre essas características, argumentos adicionais são incluídos como em

```
[t, y] = ode45(@defun, tspan, y0, opcoes, p1, p2, ...)
```

onde `opcoes` é uma estrutura de dados que é criada com a função `odeset` para controlar características da solução, e `p1`, `p2`, ... são parâmetros que você quer passar para `defun`.

A função `odeset` tem a seguinte sintaxe geral

```
opcoes = odeset('par1', val1, 'par2', val2, ...)
```

onde o parâmetro `pari` tem o valor `vali`. Uma lista completa de todos os parâmetros possíveis pode ser obtida simplesmente entrando com o comando `odeset` no prompt de comando do MATLAB. Alguns dos parâmetros frequentemente usados são

'RelTol'	Permite ajustar a tolerância relativa.
'AbsTol'	Permite ajustar a tolerância absoluta.
'InitialStep'	O solucionador determina automaticamente o passo inicial. Essa opção permite que você defina o seu próprio.
'MaxStep'	O passo máximo padrão corresponde a um décimo do intervalo <code>tspan</code> . Essa opção permite que se ultrapasse esse padrão.

EXEMPLO 23.2 Utilizando a função `odeset` para controlar as opções de integração

Use a função `ode23` para resolver a seguinte EDO de $t = 0$ a 4:

$$\frac{dy}{dt} = 10e^{-(t-2)^2/[2(0,075)^2]} - 0,6y$$

onde $y(0) = 0,5$. Obtenha soluções para erro relativo com tolerância-padrão (10^{-3}) e tolerância mais rigorosa (10^{-4}).

Solução. A princípio, criaremos um arquivo-M para calcular o lado direito da EDO:

```
function yp = dydt(t, y)
yp = 10*exp(-(t-2)*(t-2)/(2*.075^2))-0.6*y;
```

Então, podemos implementar o solucionador sem definir as opções. Portanto, o valor-padrão para o erro relativo (10^{-3}) é automaticamente usado:

```
>> ode23(@dydt, [0 4], 0.5);
```

Observe que não definimos a função igual às variáveis de saída `[t, y]`. Quando implementamos um dos solucionadores de EDOs dessa maneira, o MATLAB automaticamente cria um gráfico dos resultados exibindo círculos nos valores calculados. Como na Figura 23.4a, observe como a função `ode23` toma passos relativamente grandes nas regiões suaves da solução, enquanto passos menores são tomados na região de variações rápidas, em torno de $t = 2$.

Podemos obter uma solução mais acurada utilizando a função `odeset` para definir a tolerância do erro relativo como 10^{-4} :

```
>> opcoes=odeset('RelTol',1e-4);
>> ode23(@dydt, [0, 4], 0.5, opcoes);
```

Como na Figura 23.4b, o solucionador toma mais passos menores para atingir a maior precisão.

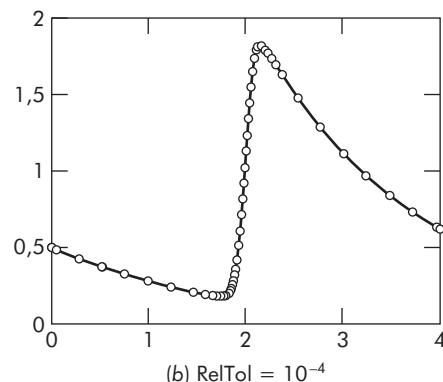
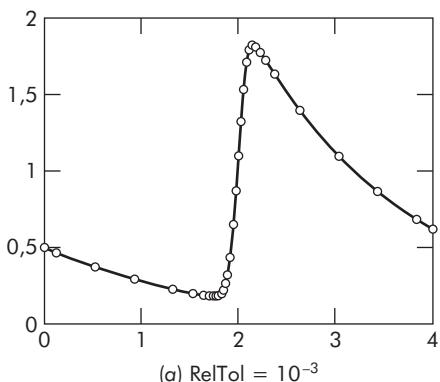


FIGURA 23.4 Solução de EDO com o MATLAB. Para (b), uma menor tolerância do erro relativo é usada e, portanto, muitos mais passos são tomados.

23.1.2 Eventos

Os solucionadores de EDOs do MATLAB são frequentemente implementados para um intervalo de integração pré-especificado; ou seja, eles são geralmente usados para obter uma solução de um valor inicial a um valor final da variável dependente. No entanto, há vários problemas em que não conhecemos o instante de tempo final.

Um bom exemplo refere-se ao saltador de *bungee jumping* em queda livre que temos utilizado ao longo deste livro. Considere que o instrutor inadvertidamente esqueceu-se de conectar a corda ao saltador. O instante de tempo final para esse caso, que corresponde ao saltador batendo no chão, não é um dado do problema. Na verdade, o objetivo de resolver as EDOs é determinar quando o saltador alcança o chão.

A opção `events` do MATLAB fornece meios para resolver esses problemas. Ela funciona por meio da resolução das equações diferenciais, até que uma das variáveis dependentes atinja zero. Obviamente, pode haver casos em que se pretenda finalizar os cálculos em um valor diferente de zero. Como descrito nos parágrafos seguintes, esses casos podem ser facilmente acomodados.

Utilizaremos nosso problema do saltador de *bungee jumping* para ilustrar a abordagem. O sistema de EDOs pode ser formulado como

$$\begin{aligned}\frac{dx}{dt} &= v \\ \frac{dv}{dt} &= g - \frac{c_d}{m}v|v|\end{aligned}$$

onde x é a distância (m), t é o tempo (s), v é a velocidade (m/s) em que velocidade positiva está no sentido para baixo, g é a aceleração da gravidade ($= 9,81 \text{ m/s}^2$), c_d é um coeficiente de arraste de segunda ordem (kg/m) e m é a massa (kg). Observe que, nessa formulação, tanto a distância quanto a velocidade são positivas no sentido para baixo e o nível do chão é definido como a distância (altura) zero. Para este exemplo, considere-se que o saltador está inicialmente localizado a 200 m acima do chão, e a velocidade inicial é 20 m/s no sentido para cima – isto é $x(0) = -200$ e $v(0) = -20$.

O primeiro passo consiste em expressar o sistema de EDOs como um arquivo-M:

```
function dydt=queda_livre(t,y,cd,m)
% y(1) = x e y(2) = v
grav=9.81;
dydt=[y(2);grav-cd/m*y(2)*abs(y(2))];
```

Para implementar o evento, dois outros arquivos-M devem ser desenvolvidos. São eles: (1) uma função que define o evento, e (2) um script que gera a solução.

Para nosso problema do saltador de *bungee jumping*, a função evento (que nomeamos `evento_fim`) pode ser escrita como

```
function [detectar, paraint, sentido] = fm_evento(t, y,
varargin)
% Localiza o instante de tempo quando a altura passa por zero
% e para a integração.
detectar=y(1); % Detecta altura = 0
paraint=1; % Para a integração
sentido=0; % O sentido não importa
```

Essa função tem como argumentos de entrada os valores das variáveis independente (t) e dependente (x) junto com os parâmetros do modelo (`varargin`). Ela então calcula e retorna três variáveis. A primeira, `detectar`, especifica que o MATLAB deve detectar o evento quando a variável dependente $y(1)$ for igual zero – isto é, quando a altura $x = 0$. A segunda, `paraint`, é definida igual a 1. Isso instrui o MATLAB a parar quando o evento ocorrer. A última variável, `sentido`, é definida igual a 0 se todos os zeros devem ser detectados (esse é o padrão), +1 se apenas os zeros onde a função evento cresce devem ser detectados e -1 se apenas os zeros onde a função evento diminui devem ser detectados. Em nosso caso, como o sentido da aproximação para o zero não é importante, definimos a variável `sentido` igual a zero¹.

Por fim, um script pode ser desenvolvido para gerar a solução:

```
opcoes=odeset('events',@fm _ evento)
y0=[-200 -20];
[t,y,te,ye]=ode45(@queda _ livre,[0 inf], y0, opcoes, 0.25,
68.1);
te,ye
plot(t,-y(:,1),'-',t,y(:,2), '--','LineWidth',2)
legend('Altura (m)', 'Velocidade (m/s)')
xlabel('tempo (s)');
ylabel('x (m) e v (m/s)')
```

Na primeira linha, a função `odeset` é usada para invocar a opção `events` e especificar que o evento que está sendo buscado está definido na função `fim _ evento`. Em seguida, definem-se as condições iniciais (y_0) e o intervalo de integração ($tspan$). Observe que, como não se sabe quando o saltador atingirá o chão, define-se o limite de integração superior como infinito. A terceira linha então utiliza a função `ode45` para gerar a solução real. Como em todos os solucionadores de EDOs do MATLAB, a função retorna as respostas nos vetores t e y . Além disso, quando a opção `events` é invocada, a função `ode45` também retorna o instante de tempo em que o evento ocorre (te) e os valores correspondentes das variáveis dependentes (ye). As linhas restantes do script simplesmente exibem e traçam os resultados. Quando o script é executado, a saída é exibida como

```
te =
9.5475
ye =
0.0000    46.2454
```

Assim, o saltador atinge o chão em 9,5475 s com uma velocidade de 46,2454 m/s. O gráfico está apresentado na Figura 23.5.

¹ Observe que, como mencionado anteriormente, pode-se intencionar a detecção de um evento diferente de zero; por exemplo, detectar quando o saltador alcança a posição $x = 5$. Para isso, simplesmente defne-se `detectar = y(1) - 5`.

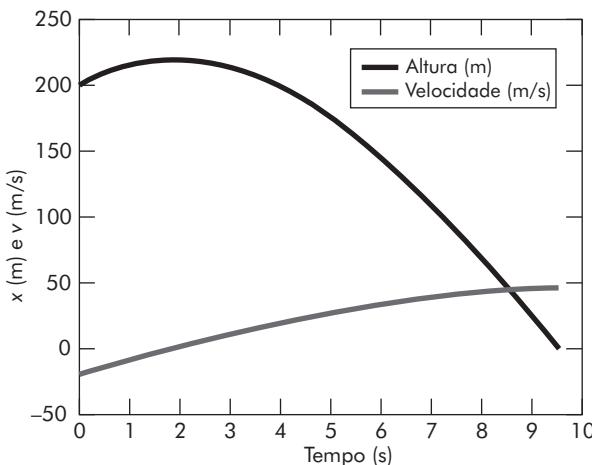


FIGURA 23.5 Gráfico gerado no MATLAB da altura acima do chão e velocidade do saltador de *bungee jumping* sem a corda.

23.2 MÉTODOS DE PASSO MÚLTIPLO

Os métodos de passo único descritos nas seções anteriores utilizam informações em um único ponto t_i para prever um valor da variável dependente y_{i+1} em um ponto futuro t_{i+1} (Figura 23.6a). Abordagens alternativas, chamadas *métodos de passo múltiplo* (Figura 23.6b), são baseadas na percepção de que, uma vez que o cálculo tenha começado, uma informação valiosa de pontos anteriores está à nossa disposição. A curvatura das linhas que liga esses valores anteriores fornece informação com relação à trajetória da solução. Os métodos de passo múltiplo exploram essa informação para resolver EDOs. Nesta seção, apresentaremos um método simples de segunda ordem, que serve para ilustrar as características gerais das abordagens de passo múltiplo.

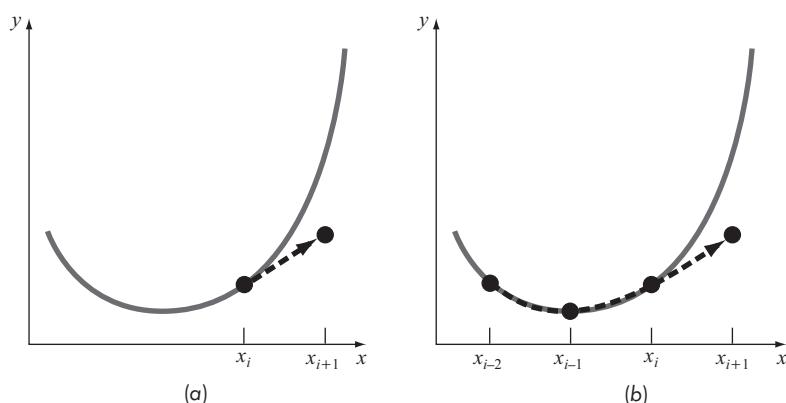


FIGURA 23.6 Descrição gráfica da diferença fundamental entre os métodos (a) de passo único e (b) de passo múltiplo para resolver EDOs.

23.2.1 O método de Heun de início não automático

Lembre-se de que a abordagem de Heun usa o método de Euler como um preditor [Equação (22.15)]:

$$y_{i+1}^0 = y_i + f(t_i, y_i)h \quad (23.4)$$

e a regra do trapézio como um corretor [Equação (22.17)]:

$$y_{i+1} = y_i + \frac{f(t_i, y_i) + f(t_{i+1}, y_{i+1}^0)}{2}h \quad (23.5)$$

Logo, o preditor e o corretor têm erros de truncamento locais de $O(h^2)$ e $O(h^3)$, respectivamente. Isso sugere que o preditor é o elo fraco no método, já que tem o erro maior. Essa fraqueza é significativa, pois a eficiência do passo iterativo de correção depende da acurácia da previsão inicial. Consequentemente, uma forma de melhorar o método de Heun é desenvolver um preditor que tenha um erro local de $O(h^3)$, o que é possível com o uso do método de Euler e da inclinação em y_i e informação extra de um ponto anterior y_{i-1} , como em

$$y_{i+1}^0 = y_{i-1} + f(t_i, y_i)2h \quad (23.6)$$

Observe que essa fórmula alcança $O(h^3)$ às custas da utilização de um tamanho de passo maior, $2h$. Além disso, observe que a equação não é de início automático, pois envolve um valor anterior da variável dependente y_{i-1} . Esse valor não estaria disponível em um problema de valor inicial típico. Por isso, as Equações (23.5) e (23.6) são chamadas de *método de Heun de início não automático*. Como mostra a Figura 23.7, a estimativa da derivada na Equação (23.6) está agora localizada no ponto médio, em vez de no início do intervalo no qual a previsão é feita. Essa centralização melhora o erro do preditor para $O(h^3)$.

O método de Heun de início não automático pode ser resumido como

$$\text{Preditor (Figura 23.7a): } y_{i+1}^0 = y_{i-1}^m + f(t_i, y_i^m)2h \quad (23.7)$$

$$\text{Corretor (Figura 23.7b): } y_{i+1}^j = y_i^m + \frac{f(t_i, y_i^m) + f(t_{i+1}, y_{i+1}^{j-1})}{2}h \quad (23.8)$$

(para $j = 1, 2, \dots, m$)

onde os sobrescritos foram adicionados para denotar que o corretor é aplicado iterativamente de $j = 1$ a m para obter soluções refinadas. Observe que y_i^m e y_{i-1}^m são o resultado final das iterações do corretor nos passos anteriores no tempo. As iterações são encerradas com base em uma estimativa do erro aproximado,

$$|\varepsilon_a| = \left| \frac{y_{i+1}^j - y_{i+1}^{j-1}}{y_{i+1}^j} \right| \times 100\% \quad (23.9)$$

Quando $|\varepsilon_a|$ for menor do que uma tolerância de erro pré-especificada ε_s , as iterações são encerradas. Nesse ponto, $j = m$. O uso das Equações (23.7) a (23.9) para resolver uma EDO é ilustrado no exemplo a seguir.

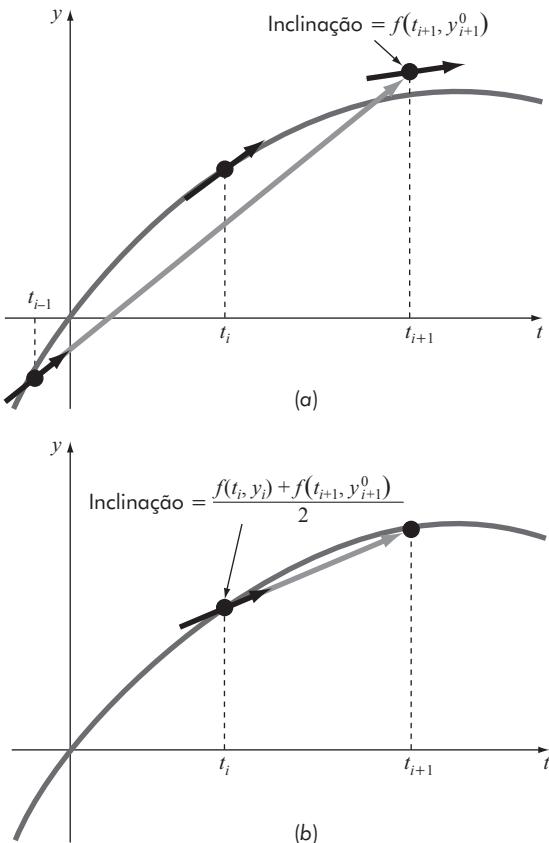


FIGURA 23.7 Descrição gráfica do método de Heun de início não automático.

- (a) O método do ponto médio, que é usado como um preditor.
 (b) A regra do trapézio, que é usada como um corretor.

EXEMPLO 23.3 Método de Heun de início não automático

Use o método de Heun de início não automático para fazer os mesmos cálculos feitos anteriormente no Exemplo 22.2 com o método de Heun; ou seja, integre $y' = 4e^{0.8t} - 0,5y$ de $t = 0$ a 4, com um tamanho de passo de 1. Como no Exemplo 22.2, a condição inicial em $t = 0$ é $y = 2$. Entretanto, como agora estamos lidando com um método de passo múltiplo, precisamos da informação adicional de que y é igual a $-0,3929953$ em $t = -1$.

Solução. O preditor [Equação (23.7)] é usado para extrapolar linearmente de $t = -1$ a 1:

$$y_1^0 = -0,3929953 + [4e^{0.8(0)} - 0,5(2)]2 = 5,607005$$

O corretor [Equação (23.8)] é então usado para calcular o valor:

$$y_1^1 = 2 + \frac{4e^{0.8(0)} - 0,5(2) + 4e^{0.8(1)} - 0,5(5,607005)}{2}1 = 6,549331$$

o qual representa um erro relativo percentual de $-5,73\%$ (valor verdadeiro = $6,194631$). Esse erro é um pouco menor que o valor de $-8,18\%$ que ocorreu no método de Heun de início automático.

Agora, a Equação (23.8) pode ser aplicada iterativamente para melhorar a solução:

$$y_1^2 = 2 + \frac{3 + 4e^{0,8(1)} - 0,5(6,549331)}{2} = 6,313749$$

o que representa um erro de $-1,92\%$. Uma estimativa aproximada do erro pode ser determinada por meio da Equação (23.9):

$$|\varepsilon_a| = \left| \frac{6,313749 - 6,549331}{6,313749} \right| \times 100\% = 3,7\%$$

A Equação (23.8) pode ser aplicada iterativamente até que ε_a caia abaixo de um valor pré-especificado ε_s . Como no caso do método de Heun (lembre-se do Exemplo 22.2), as iterações convergem para um valor de $6,36087$ ($\varepsilon_t = -2,68\%$). No entanto, como o valor inicial do preditor é mais acurado, o método de passo múltiplo converge a uma taxa um pouco mais rápida.

Para o segundo passo, o preditor é

$$y_2^0 = 2 + [4e^{0,8(1)} - 0,5(6,36087)]2 = 13,44346 \quad \varepsilon_t = 9,43\%$$

o qual é superior à previsão de $12,0826$ ($\varepsilon_t = 18\%$) que foi calculada com o método de Heun original. O primeiro corretor fornece $15,76693$ ($\varepsilon_t = 6,8\%$), e as iterações subsequentes convergem para o mesmo resultado obtido com o método de Heun de início automático: $15,30224$ ($\varepsilon_t = -3,09\%$). Como no passo anterior, a taxa de convergência do corretor é um pouco melhor por causa da previsão inicial melhor.

23.2.2 Estimativas de erro

Além de fornecer maior eficiência, o método de Heun de início não automático também pode ser usado para estimar o erro de truncamento local. Como com os métodos RK adaptativos na Seção 23.1, a estimativa de erro fornece então um critério para variação do tamanho do passo.

A estimativa de erro pode ser deduzida reconhecendo que o preditor é equivalente à regra do ponto médio. Portanto, seu erro de truncamento local é (Tabela 19.4)

$$E_p = \frac{1}{3}h^3 y^{(3)}(\xi_p) = \frac{1}{3}h^3 f''(\xi_p) \quad (23.10)$$

onde o subscrito p indica que esse é o erro do preditor. Essa estimativa pode ser combinada com a estimativa de y_{i+1} do passo do preditor, para fornecer

$$\text{Valor verdadeiro} = y_{i+1}^0 + \frac{1}{3}h^3 y^{(3)}(\xi_p) \quad (23.11)$$

Reconhecendo que o corretor é equivalente à regra do trapézio, uma estimativa similar do erro de truncamento local para o corretor é (Tabela 19.2)

$$E_c = -\frac{1}{12}h^3 y^{(3)}(\xi_c) = -\frac{1}{12}h^3 f''(\xi_c) \quad (23.12)$$

Essa estimativa de erro pode ser combinada com o resultado do corretor y_{i+1} , para fornecer

$$\text{Valor verdadeiro} = y_{i+1}^m - \frac{1}{12}h^3y^{(3)}(\xi_c) \quad (23.13)$$

A Equação (23.11) pode ser subtraída da Equação (23.13), para fornecer

$$0 = y_{i+1}^m - y_{i+1}^0 - \frac{5}{12}h^3y^{(3)}(\xi) \quad (23.14)$$

onde ξ agora está entre t_{i-1} e t_i . Agora, dividindo a Equação (23.14) por 5 e reorganizando o resultado, obtemos

$$\frac{y_{i+1}^0 - y_{i+1}^m}{5} = -\frac{1}{12}h^3y^{(3)}(\xi) \quad (23.15)$$

Observe que os lados direitos das Equações (23.12) e (23.15) são idênticos, com a exceção do argumento da terceira derivada. Se a terceira derivada não variar muito no intervalo em questão, podemos considerar que os lados direitos são iguais e, portanto, os lados esquerdos também deveriam ser iguais, como em

$$E_c = -\frac{y_{i+1}^0 - y_{i+1}^m}{5} \quad (23.16)$$

Assim, chegamos a uma relação que pode ser usada para obter uma estimativa do erro de truncamento em cada passo, com base em duas quantidades que são subprodutos rotineiros dos cálculos: o preditor (y_{i+1}^0) e o corretor y_{i+1}^m .

EXEMPLO 23.4 Estimativa do erro de truncamento em cada passo

Use a Equação (23.16) para obter uma estimativa do erro de truncamento em cada passo do Exemplo 23.3. Observe que os valores verdadeiros em $t = 1$ e 2 são 6,194631 e 14,84392, respectivamente.

Solução. Em $t_{i+1} = 1$, o preditor dá 5,607005 e o corretor produz 6,360865. Esses valores podem ser substituídos na Equação (23.16) para fornecer

$$E_c = -\frac{6,360865 - 5,607005}{5} = -0,150722$$

o qual se compara bem com o erro exato,

$$E_t = 6,194631 - 6,360865 = -0,1662341$$

Em $t_{i+1} = 2$, o preditor dá 13,44346, e o corretor produz 15,30224, que podem ser usados para calcular

$$E_c = -\frac{15,30224 - 13,44346}{5} = -0,37176$$

que também se compara favoravelmente com o erro exato, $E_t = 14,84392 - 15,30224 = -0,45831$.

O exposto anteriormente foi uma breve introdução dos métodos de passo múltiplo. Informação adicional pode ser encontrada em outras referências (por exemplo, Chapra e Canale, 2010). Embora eles ainda tenham seu lugar para resolução de certos tipos de problemas, os métodos de passo múltiplo geralmente não são o método escolhido para a maioria dos problemas rotineiramente encontrados na engenharia e na ciência, mas ainda são utilizados. Por exemplo, a função `ode113` do MATLAB é um método de passo múltiplo. Portanto, incluímos esta seção para apresentar ao estudante seus princípios básicos.

23.3 RIGIDEZ

A rigidez é um problema especial que pode aparecer na solução de equações diferenciais ordinárias. Um *sistema rígido* é aquele que envolve componentes que variam rapidamente junto com componentes que variam lentamente. Em alguns casos, as componentes que variam rapidamente são transientes efêmeros que desaparecem rapidamente, depois disso a solução passa a ser dominada pelas componentes que variam lentamente. Embora os fenômenos transitórios existam apenas para uma pequena parte do intervalo de integração, eles podem determinar o passo no tempo para toda a solução.

Tanto as EDOs individuais quanto os sistemas de EDOs podem ser rígidos. Um exemplo de uma EDO individual rígida é

$$\frac{dy}{dt} = -1.000y + 3.000 - 2.000e^{-t} \quad (23.17)$$

Se $y(0) = 0$, a solução analítica pode ser escrita como

$$y = 3 - 0,998e^{-1.000t} - 2,002e^{-t} \quad (23.18)$$

Como na Figura 23.8, a solução é o princípio dominado pelo termo exponencial rápido ($e^{-1.000t}$). Depois de um curto período ($t < 0,005$), esse transiente desaparece e a solução passa a ser dominada pela exponencial lenta (e^{-t}).

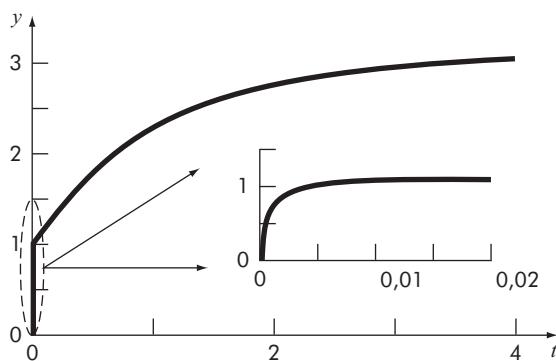


FIGURA 23.8 Gráfico de uma solução rígida de uma única EDO. Embora a solução pareça começar em 1, na realidade existe um transiente rápido de $y = 0$ a 1, que ocorre em menos do que 0,005 unidades de tempo. Esse transiente é perceptível apenas quando a resposta é vista na inserção em uma escala de tempo mais fina.

Pode-se ganhar uma percepção do tamanho do passo necessário para a estabilidade dessa solução examinando a parte homogênea da Equação (23.17):

$$\frac{dy}{dt} = -ay \quad (23.19)$$

Se $y(0) = y_0$, o cálculo pode ser usado para determinar a solução como

$$y = y_0 e^{-at}$$

Logo, a solução começa em y_0 e se aproxima assintoticamente de zero.

O método de Euler pode ser usado para resolver o mesmo problema numericamente:

$$y_{i+1} = y_i + \frac{dy_i}{dt}h$$

Substituindo a Equação (23.19), obtemos

$$y_{i+1} = y_i - ay_i h$$

ou

$$y_{i+1} = y_i(1 - ah) \quad (23.20)$$

A estabilidade dessa fórmula depende claramente do tamanho do passo h , ou seja, $|1 - ah|$ deve ser menor que 1. Logo, se $h > 2/a$, $|y_i| \rightarrow \infty$ quando $i \rightarrow \infty$.

Para a parte transiente rápida da Equação (23.18), esse critério pode ser usado para mostrar que o tamanho do passo para manter a estabilidade deve ser $2/1.000 = 0,002$. Além disso, deve ser observado que, embora esse critério mantenha a estabilidade (ou seja, uma solução limitada), é necessário um tamanho de passo ainda menor para obter uma solução acurada. Logo, embora o transiente ocorra apenas para uma pequena fração do intervalo de integração, ele controla o tamanho máximo permitível do passo.

Em vez de usar abordagens explícitas, os métodos implícitos oferecem um recurso alternativo. Essas representações são chamadas *implícitas* porque a incógnita aparece em ambos os lados da equação. Uma forma implícita do método de Euler pode ser desenvolvida pelo cálculo da derivada em um instante futuro,

$$y_{i+1} = y_i + \frac{dy_{i+1}}{dt}h$$

Isso é chamado de *método de Euler regressivo* ou *implícito*. A substituição da Equação (23.19) fornece

$$y_{i+1} = y_i - ay_{i+1}h$$

que pode ser resolvida por

$$y_{i+1} = \frac{y_i}{1 + ah} \quad (23.21)$$

Para esse caso, independentemente do tamanho do passo, $|y_i| \rightarrow 0$ quando $i \rightarrow \infty$. Assim, a abordagem é chamada *incondicionalmente estável*.

EXEMPLO 23.5 Euler explícito e implícito

Use os métodos de Euler implícito e explícito para resolver a Equação (23.17), em que $y(0) = 0$. (a) Utilize Euler explícito com tamanhos de passo de 0,0005 e 0,0015 para determinar y entre $t = 0$ e 0,006; e (b) Euler implícito com tamanho de passo de 0,05 para determinar y entre 0 e 4.

Solução. (a) Para esse problema, o método de Euler explícito é

$$y_{i+1} = y_i + (-1.000y_i + 3.000 - 2.000e^{-t_i})h$$

O resultado para $h = 0,0005$ está mostrado na Figura 23.9a com a solução analítica. Embora apresente algum erro de truncamento, o resultado captura a forma geral da solução analítica. Em contraste, quando o tamanho do passo é aumentado para um valor um pouco abaixo do limite de estabilidade ($h = 0,0015$), a solução apresenta oscilações. Usar $h < 0,002$ resultaria em uma solução totalmente instável, isto é, que iria para infinito à medida que a solução prosseguisse.

(b) O método de Euler implícito é

$$y_{i+1} = y_i + (-1.000y_{i+1} + 3.000 - 2.000e^{-t_{i+1}})h$$

Agora, como a EDO é linear, podemos reescrever essa equação de modo que y_{i+1} fique isolado no lado esquerdo

$$y_{i+1} = \frac{y_i + 3.000h - 2.000he^{-t_{i+1}}}{1 + 1.000h}$$

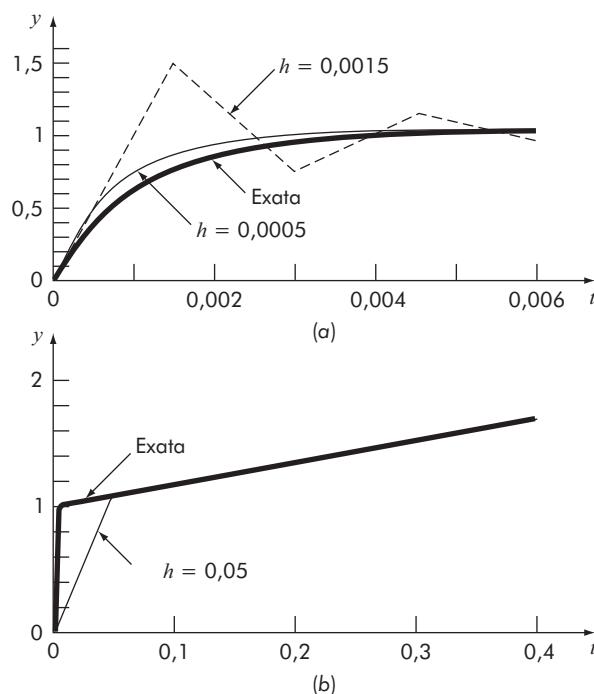


FIGURA 23.9 Solução de uma EDO rígida com os métodos de Euler (a) explícito e (b) implícito.

O resultado para $h = 0,05$ está mostrado na Figura 23.9b com a solução analítica. Observe que, embora tenhamos usado um tamanho de passo muito maior do que aquele que induziu estabilidade no Euler explícito, a solução numérica segue de perto o resultado analítico.

Os sistemas de EDOs também podem ser rígidos. Um exemplo é

$$\frac{dy_1}{dt} = -5y_1 + 3y_2 \quad (23.22a)$$

$$\frac{dy_2}{dt} = 100y_1 - 301y_2 \quad (23.22b)$$

Para as condições iniciais $y_1(0) = 52,29$ e $y_2(0) = 83,82$, a solução exata é

$$y_1 = 52,96e^{-3,9899t} - 0,67e^{-302,0101t} \quad (23.23a)$$

$$y_2 = 17,83e^{-3,9899t} + 65,99e^{-302,0101t} \quad (23.23b)$$

Observe que os expoentes são negativos e diferem de cerca de duas ordens de grandeza. Como no caso de uma única equação, são os expoentes grandes que respondem rapidamente e são o cerne da rigidez do problema.

Um método de Euler implícito para sistemas pode ser formulado para o caso presente como

$$y_{1,i+1} = y_{1,i} + (-5y_{1,i+1} + 3y_{2,i+1})h \quad (23.24a)$$

$$y_{2,i+1} = y_{2,i} + (100y_{1,i+1} - 301y_{2,i+1})h \quad (23.24b)$$

Agrupando os termos, obtemos

$$(1 + 5h)y_{1,i+1} - 3y_{2,i+1} = y_{1,i} \quad (23.25a)$$

$$-100y_{1,i+1} + (1 + 301h)y_{2,i+1} = y_{2,i} \quad (23.25b)$$

Logo, podemos ver que o problema consiste em resolver um conjunto de equações simultâneas para cada passo no tempo.

Para EDOs não lineares, a solução se torna ainda mais difícil, já que envolve a resolução de um sistema de equações simultâneas não lineares (lembre-se da Seção 12.2). Assim, embora se ganhe estabilidade com as abordagens implícitas, há um consequente aumento da complexidade da solução.

23.3.1 Funções nativas do MATLAB para sistemas rígidos

O MATLAB tem uma série de funções nativas para resolução de sistemas rígidos de EDOs. São elas:

ode15s. Essa função é um solucionador de ordem variável baseado em fórmulas de derivação numérica; é um solucionador de passo múltiplo que opcionalmente usa fórmulas de diferenciação regressiva de Gear; e é utilizada para problemas rígidos de baixa ou média acurácia.

ode23s. Essa função é baseada em uma fórmula de Rosenbrock modificada de ordem 2. Como ela é um solucionador de passo único, pode ser mais eficiente do que a função `ode15s` para tolerâncias grosseiras; e pode resolver alguns tipos de problemas rígidos melhor do que a `ode15s`.

ode23t. Essa função é uma implementação da regra do trapézio com um interpolador “livre”; e é usada para problemas moderadamente rígidos com baixa acurácia em que é necessária uma solução sem amortecimento numérico.

ode23tb. Essa é uma implementação de uma fórmula de Runge-Kutta implícita com um primeiro estágio, que é uma regra do trapézio, e um segundo estágio, que é uma fórmula de diferenciação regressiva de ordem 2. Esse solucionador também pode ser mais eficiente do que o `ode15s` para tolerâncias grosseiras.

EXEMPLO 23.6 MATLAB para EDOs Rígidas

A equação de van der Pol é um modelo de um circuito eletrônico que surgiu nos tempos de tubos de vácuo,

$$\frac{d^2y_1}{dt^2} - \mu(1 - y_1^2)\frac{dy_1}{dt} + y_1 = 0 \quad (\text{E23.6.1})$$

A solução para essa equação torna-se progressivamente mais rígida à medida que μ cresce. Dadas as condições iniciais, $y_1(0) = dy_1/dt = 1$, use o MATLAB para resolver os seguintes casos: (a) para $\mu = 1$, use `ode45` para resolver para $t = 0$ a 20; e (b) para $\mu = 1.000$, use `ode23s` para resolver para $t = 0$ a 6.000.

Solução. (a) O primeiro passo é converter a EDO de segunda ordem em um par de EDOs de primeira ordem, definindo

$$\frac{dy_1}{dt} = y_2$$

Utilizando essa equação, a Equação (E23.6.1) pode ser escrita como

$$\frac{dy_2}{dt} = \mu(1 - y_1^2)y_2 - y_1 = 0$$

Um arquivo-M pode agora ser criado para armazenar esse par de equações diferenciais ordinárias:

```
function yp = vanderpol(t,y,mu)
yp = [y(2);mu*(1-y(1)^2)*y(2)-y(1)];
```

Perceba como o valor de μ é passado como um argumento de entrada. Como no Exemplo 23.1, a função `ode45` pode ser chamada e os resultados traçados:

```
>> [t,y] = ode45(@vanderpol,[0 20],[1 1],[],1);
>> plot(t,y(:,1),'-',t,y(:,2),'--')
>> legend('y1','y2');
```

Observe que como não estamos especificando quaisquer opções, devemos utilizar colchetes abertos [] no lugar reservado. A natureza suave do gráfico (Figura 23.10a) sugere que a equação de van der Pol com $\mu = 1$ não é um sistema rígido.

(b) Se um solucionador padrão como o `ode45` é usado para o caso rígido ($\mu = 1.000$), ele falhará (experimente, se você quiser). No entanto, o solucionador `ode23s` faz um trabalho eficiente:

```
>> [t,y] = ode23s(@vanderpol,[0 6000],[1 1],[],1000);
>> plot(t,y(:,1))
```

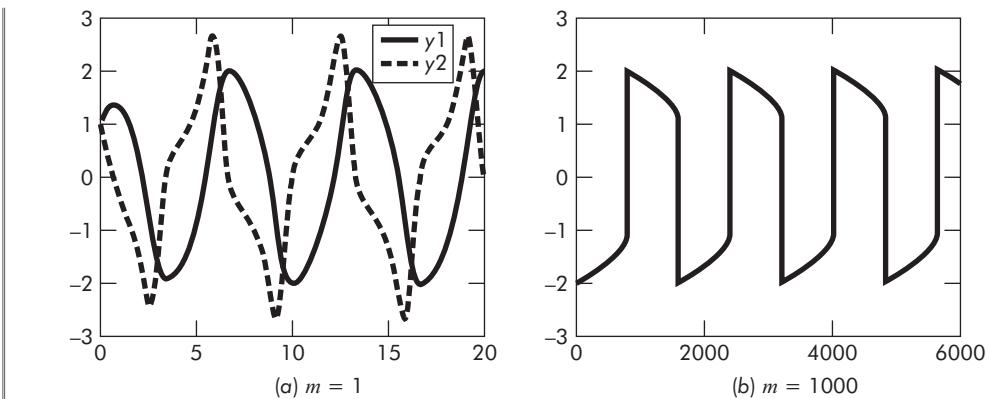


FIGURA 23.10 Soluções para a equação de van der Pol. (a) Forma não rígida resolvida com `ode45` e (b) forma rígida resolvida com `ode23s`.

Mostramos apenas a componente y_1 , porque o resultado para y_2 tem uma escala muito maior. Observe como essa solução (Figura 23.10b) tem extremidades bem mais acentuadas do que no caso da Figura 23.10a. Isso é uma manifestação visual da “rigidez” da solução.

23.4 APLICAÇÃO DO MATLAB: SALTADOR DE BUNGEE JUMPING COM CORDA

Nesta seção, usaremos o MATLAB para determinar a dinâmica vertical de um saltador conectado a uma plataforma estacionária com uma corda de *bungee jumping*. Como desenvolvido no início do Capítulo 22, o problema consiste em resolver duas EDOs acopladas para a posição e velocidade verticais. A equação diferencial para posição é

$$\frac{dx}{dt} = v \quad (23.26)$$

A equação diferencial para velocidade é diferente dependendo se o saltador tiver caído para uma distância em que a corda está totalmente estendida e começa a se esticar. Assim, se a distância da queda é menor do que o comprimento da corda, o saltador está sujeito apenas às forças gravitacional e de arraste.

$$\frac{dv}{dt} = g - \text{sign}(v) \frac{c_d}{m} v^2 \quad (23.27a)$$

Uma vez que a corda começa a esticar, as forças da mola (elástica) e de amortecimento da corda também devem ser incluídas:

$$\frac{dv}{dt} = g - \text{sign}(v) \frac{c_d}{m} v^2 - \frac{k}{m}(x - L) - \frac{\gamma}{m}v \quad (23.27b)$$

O exemplo a seguir mostra como o MATLAB pode ser usado para resolver esse problema.

EXEMPLO 23.7 Saltador de *bungee jumping* com corda

Determine a posição e a velocidade de um saltador de *bungee jumping* com os seguintes parâmetros: $L = 30$ m, $g = 9,81$ m/s², $m = 68,1$ kg, $c_d = 0,25$ kg/m, $k = 40$ N/m e $\gamma = 8$ N · s/m. Efetue os cálculos de $t = 0$ a 50 s e considere que as condições iniciais são $x(0) = v(0) = 0$.

Solução. O seguinte arquivo-M pode ser definido para calcular os lados direitos das EDOs:

```
function dydt = bungee(t,y,L,cd,m,k,gamma)
g = 9.81;
cord = 0;
if y(1) > L %determina se a corda exerce uma força
cord = k/m*(y(1)-L)+gamma/m*y(2);
end
dydt = [y(2); g - sign(y(2))*cd/m*y(2)^2 - cord];
```

Observe que as derivadas são retornadas como um vetor coluna porque esse é o formato exigido pelos solucionadores do MATLAB.

Como essas equações não são rígidas, podemos usar a função `ode45` para obter as soluções e mostrá-las em um gráfico:

```
>> [t,y] = ode45(@bungee,[0 50],[0 0],[],30,0.25,68.1,40,8);
>> plot(t,-y(:,1),'-',t,y(:,2),':')
>> legend('x (m)', 'v (m/s)')
```

Como na Figura 23.11, invertemos o sinal da distância para o gráfico de modo que a distância negativa está no sentido para baixo. Observe como a simulação captura o movimento de pulo do saltador.

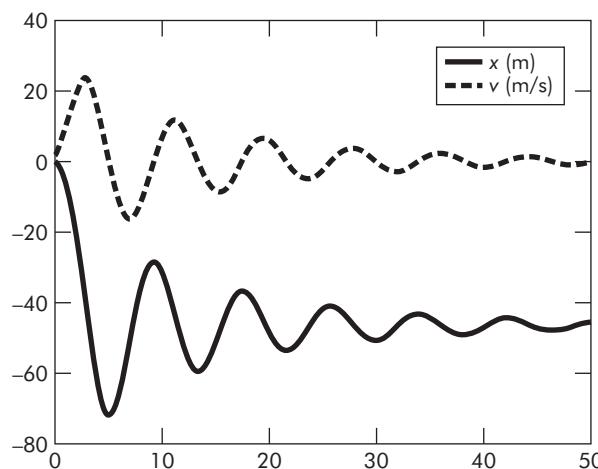


FIGURA 23.11 Gráfico da distância e velocidade de um saltador de *bungee jumping*.

► 23.5 ESTUDO DE CASO FONTE INTERMITENTE DE PLÍNIO

Contextualização. O filósofo romano naturalista Plínio, o Velho, supostamente teve uma fonte intermitente em seu jardim. Como na Figura 23.12, a água entra em um tanque cilíndrico, em uma taxa constante Q_{entrada} , e enche o tanque até atingir o nível y_{alto} . Neste ponto, a água escoa para fora do tanque através de um tubo de descarga circular (um sifão), produzindo uma fonte na saída do tubo. A fonte funciona até que o nível de água diminua para y_{baixo} ; depois disso o sifão enche-se com ar e a fonte para. O ciclo é então repetido à medida que o tanque enche-se de água até atingir y_{alto} e a fonte jorra de novo.

Quando o sifão está funcionando, o fluxo para fora $Q_{\text{saída}}$ pode ser calculado com a fórmula a seguir baseada na *lei de Torricelli*:

$$Q_{\text{saída}} = C \sqrt{2gy} \pi r^2 \quad (23.28)$$

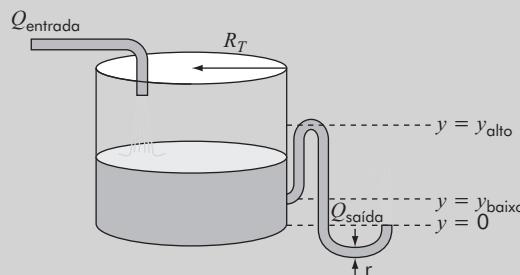


FIGURA 23.12 Fonte intermitente.

Desprezando o volume de água no tubo, calcule e trace o nível da água no tanque como uma função do tempo durante 100 s. Considere uma condição inicial de um tanque vazio $y(0) = 0$ e empregue os parâmetros seguintes para seus cálculos:

$$\begin{aligned} R_T &= 0,05 \text{ m} & r &= 0,007 \text{ m} & y_{\text{baixo}} &= 0,025 \text{ m} \\ y_{\text{alto}} &= 0,1 \text{ m} & C &= 0,6 & g &= 9,81 \text{ m/s}^2 \\ Q_{\text{entrada}} &= 50 \times 10^{-6} \text{ m}^3/\text{s} \end{aligned}$$

Solução. Quando a fonte está funcionando, a taxa de variação no volume do tanque V (m^3) é determinada por um balanço simples do fluxo para dentro menos o fluxo para fora:

$$\frac{dV}{dt} = Q_{\text{entrada}} - Q_{\text{saída}} \quad (23.29)$$

onde V é o volume (m^3). Como o tanque é cilíndrico, $V = \pi R_t^2 y$. Substituindo essa relação com a Equação (23.28) na Equação (23.29), obtemos

$$\frac{dy}{dt} = \frac{Q_{\text{entrada}} - C \sqrt{2gy} \pi r^2}{\pi R_t^2} \quad (23.30)$$

Quando a fonte não está funcionando, o segundo termo no numerador vai a zero. Podemos incorporar esse mecanismo no modelo introduzindo uma nova variável adimensional,

nomeada *sifão*, que é igual a zero, quando a fonte está parada, e igual a um, quando a fonte está funcionando:

$$\frac{dy}{dt} = \frac{Q_{\text{entrada}} - \text{sifão} \times C \sqrt{2\pi g} y r^2}{\pi R_t^2} \quad (23.31)$$

Neste contexto, a variável *sifão* pode ser pensada como um interruptor que desliga e liga a fonte. Essas variáveis de dois estados são chamadas de *variáveis lógicas* ou *booleanas*, nas quais zero é o equivalente a falso, e um é o equivalente a verdadeiro.

Em seguida, devemos relacionar *sifão* à variável dependente y . A princípio, *sifão* é definida como zero sempre que o nível cair abaixo de y_{baixo} . Inversamente, *sifão* é definida como um sempre que o nível subir acima de y_{alto} . A função a seguir do MATLAB segue essa lógica no cálculo da derivada:

```
function dy=plinio_edo(t,y)
global sifao
Rt=0.05; r=0.007; yhi=0.1; yalto=0.1; ybaixo=0.025;
C=0.6; g=9.81; Qentrada=0.00005;
if y(1) <= ybaixo
    sifao=0;
elseif y(1) >= yalto
    sifao=1;
end
Qsaída=sifao*C*sqrt(2*g*y(1))*pi*r^2;
dy=(Qentrada-Qsaída)/(pi*Rt^2);
```

Observe que, como seu valor deve ser mantido entre chamadas da função, a variável *sifao* é declarada como uma variável global. Embora o uso de variáveis globais não seja encorajado (particularmente em programas extensos), esse uso é útil no contexto presente.

O programa a seguir emprega a função nativa `ode45` para integrar `plinio_edo` e gerar um gráfico da solução:

```
global sifao
sifao=0;
tspan=[0 100]; y0=0;
[tp,yp]=ode45(@plinio_edo,tspan,y0);
plot(tp,yp)
xlabel('Tempo (s)')
ylabel('Nível de água no tanque (m)')
```

Como mostra a Figura 23.13, o resultado está claramente incorreto. Exceto para o processo de enchimento original, o nível parece começar a decrescer antes de atingir y_{alto} . De modo similar, quando o sifão está jorrando água, ele é fechado bem antes do nível cair para y_{baixo} .

Neste ponto, suspeitando que o problema demande uma função computacional mais poderosa que a rotina confiável `ode45`, você pode ser tentado a usar algum dos outros solucionadores de EDOs do MATLAB, tais como `ode23s` ou `ode23tb`. Porém, se você fizer isso, descobrirá que, embora essas rotinas produzam resultados um pouco diferentes, elas ainda geram soluções incorretas.

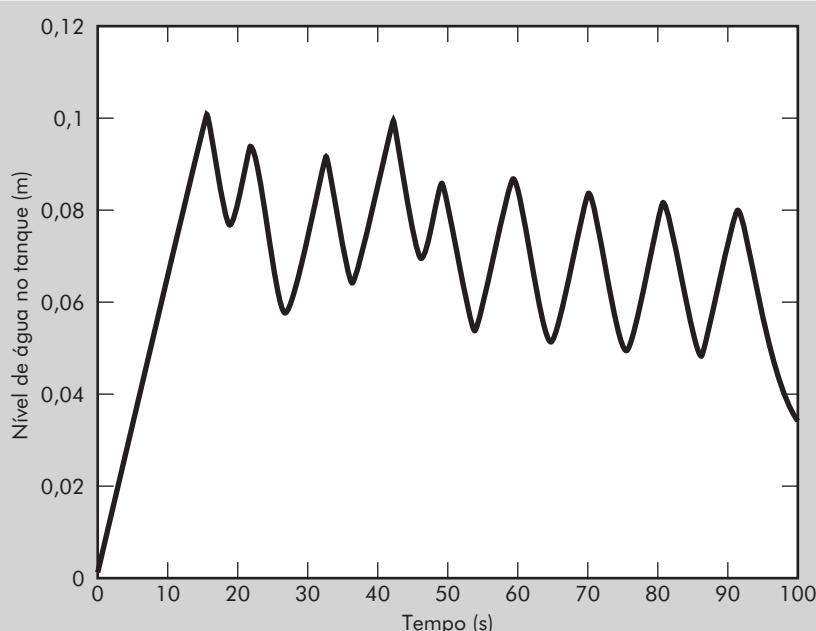


FIGURA 23.13 O nível na fonte de Plínio versus o tempo como simulado com a função `ode45`.

A dificuldade surge porque a EDO é descontínua no ponto em que o sifão é desligado ou ligado. Por exemplo, quando o tanque está enchendo, a derivada depende apenas do fluxo para dentro e, para os parâmetros presentes, tem um valor constante de $6,366 \times 10^{-3}$ m/s. Entretanto, assim que o nível atinge y_{alto} , o fluxo para fora entra em ação e a derivada cai de forma abrupta para $-1,013 \times 10^{-2}$ m/s. Embora rotinas com controle adaptativo do tamanho do passo usadas pelo MATLAB funcionem de modo eficaz para vários problemas, elas muitas vezes têm problemas quando lidam com essas descontinuidades. Como elas inferem o comportamento da solução comparando os resultados de diferentes passos, uma descontinuidade representa algo semelhante a pisar em um buraco profundo em uma rua escura.

Neste ponto, sua primeira inclinação pode ser simplesmente desistir. Afinal, se é tão difícil para o MATLAB, nenhuma pessoa razoável pode esperar que você obtenha uma solução. Como engenheiros e cientistas raramente fogem com essas desculpas, seu único recurso é desenvolver uma solução com base em seus conhecimentos de métodos numéricos.

Como o problema resulta da aplicação de um tamanho de passo adaptativo por meio de uma descontinuidade, pode-se reverter o método para uma abordagem mais simples e utilizar um tamanho de passo pequeno e constante. Raciocinando bem, essa é justamente a abordagem que se deve tomar no caso de estar-se atravessando uma rua escura cheia de buracos. Podemos implementar essa estratégia de solução simplesmente substituindo `ode45` pela função `rk4sys` de passo constante apresentada no Capítulo 22 (Figura 22.8). Para o programa descrito anteriormente, a quarta linha é formulada como

```
[tp,yp]=rk4sys(@plinio_edo,tspan,y0,0.0625);
```

Como na Figura 23.14, a solução agora evolui como esperado. O tanque enche até y_{alto} e depois esvazia até atingir y_{baixo} , quando o ciclo se repete.

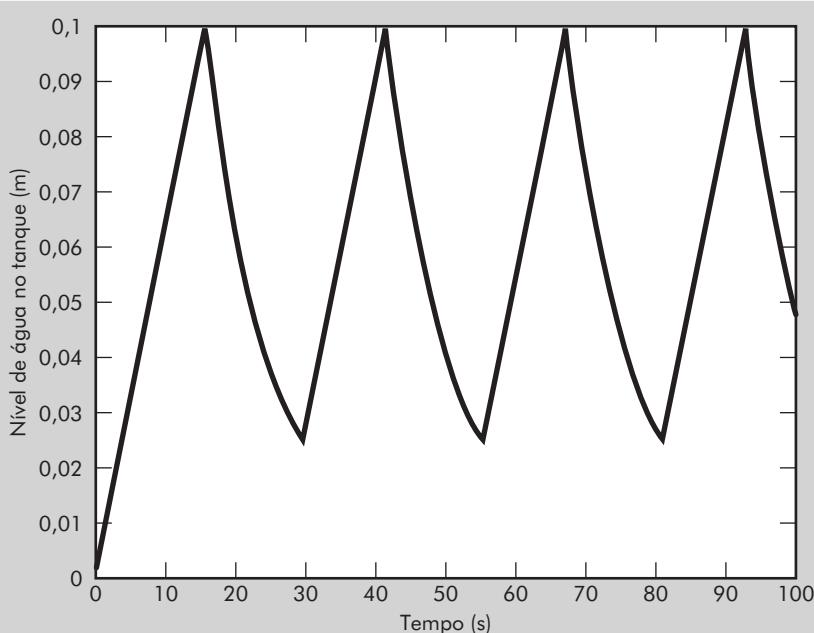


FIGURA 23.14 O nível na fonte de Plínio versus o tempo como simulado com um tamanho de passo pequeno e constante por meio da função `rk4sys` (Figura 22.8).

Duas mensagens podem ser obtidas deste estudo de caso. Primeiro, embora seja da natureza humana pensar o oposto, o mais simples algumas vezes é melhor. Afinal, parafraseando Einstein, “tudo deve ser o mais simples possível, mas não mais simples que isso.” Segundo, você nunca deve confiar cegamente em todo resultado gerado por um computador. Você provavelmente já ouviu a velha máxima, “se entra lixo, sai lixo”, em referência ao impacto da qualidade dos dados de entrada na validade da saída gerada pelo computador. Infelizmente, algumas pessoas pensam que, independentemente do que entrou (os dados) e do que está acontecendo no interior (o algoritmo), o computador gera um resultado religiosamente correto. Situações como aquela descrita na Figura 23.13 são especialmente perigosas – isto é, embora a saída seja incorreta, ela não é obviamente incorreta; ou seja, a simulação não se torna instável ou produz níveis negativos. Na verdade, a solução se move para cima e para baixo na forma de uma fonte intermitente, embora incorreta.

Esperamos que este estudo de caso ilustre que mesmo uma grande parte de pacotes computacionais como o MATLAB não é infalível. Portanto, engenheiros e cientistas mais sofisticados sempre examinam saídas numéricas com um ceticismo saudável com base em sua experiência considerável e conhecimento dos problemas que estão resolvendo.

►►PROBLEMAS

23.1 Repita as mesmas simulações como na Seção 23.5 para a fonte de Plínio, mas gere as soluções com `ode23`, `ode23s` e `ode113`. Use o comando `subplot` para desenvolver um gráfico com três painéis verticais da solução em função do tempo.

23.2 As EDOs seguintes foram propostas como um modelo de uma epidemia:

$$\frac{dS}{dt} = -aSI$$

$$\frac{dI}{dt} = aSI - rI$$

$$\frac{dR}{dt} = rI$$

onde S são os indivíduos suscetíveis, I os infectados, R os recuperados, a é a taxa de infecção e r é a taxa de recuperação. Uma cidade tem 10.000 pessoas, e todas elas são suscetíveis.

- (a) Se um único indivíduo infectado entra na cidade em $t = 0$, calcule a progressão da epidemia até que o número de indivíduos infectados caia abaixo de 10. Use os seguintes parâmetros: $a = 0,002/(\text{pessoa} \cdot \text{semana})$ e $r = 0,15/\text{d}$. Desenvolva gráficos da solução em função do tempo de todas as variáveis de estado. Também, gere um gráfico do plano de estados de S versus I versus R .
- (b) Considere que, depois da recuperação, há uma perda de imunidade que torna indivíduos suscetíveis. Esse mecanismo de reinfeção pode ser calculado como ρR , onde ρ é a taxa de reinfeção. Modifique o modelo para incluir esse mecanismo e repita os cálculos em (a) usando $\rho = 0,03/\text{d}$.

23.3 Resolva o seguinte problema de valor inicial no intervalo de $t = 2$ a 3:

$$\frac{dy}{dt} = -0,5y + e^{-t}$$

Use o método de Heun de início não automático com um tamanho de passo de 0,5 e condições iniciais de $y(1,5) = 5,222138$ e $y(2,0) = 4,143883$. Itere o corretor para $\varepsilon_s = 0,1\%$. Calcule os erros relativos percentuais para seus resultados com base nas soluções exatas obtidas analiticamente: $y(2,5) = 3,273888$ e $y(3) = 2,577988$.

23.4 Resolva o seguinte problema de valor inicial no intervalo de $t = 0$ a 0,5,

$$\frac{dy}{dt} = yt^2 - y$$

Use o método RK de quarta ordem para prever o primeiro valor em $t = 0,25$. Depois, utilize o método de Heun de início não automático para fazer a predição em $t = 0,5$. Observação: $y(0) = 1$.

23.5 Dado

$$\frac{dy}{dt} = -100.000y + 99.999e^{-t}$$

- (a) Estime o tamanho do passo necessário para manter estabilidade por meio do método de Euler explícito.
- (b) Se $y(0) = 0$, use o Euler implícito para obter uma solução de $t = 0$ a 2 usando um tamanho de passo de 0,1.

23.6 Dado

$$\frac{dy}{dt} = 30(\sin t - y) + 3 \cos t$$

Se $y(0) = 0$, use o Euler implícito para obter uma solução de $t = 0$ a 4 usando um tamanho de passo de 0,4.

23.7 Dado

$$\frac{dx_1}{dt} = 999x_1 + 1999x_2$$

$$\frac{dx_2}{dt} = -1000x_1 - 2000x_2$$

Se $x_1(0) = x_2(0) = 1$, obtenha uma solução de $t = 0$ a 0,2 usando um tamanho de passo de 0,05 com os métodos de Euler (a) explícito e (b) implícito.

23.8 A seguinte EDO parasitária, não linear, foi sugerida por Hornbeck (1975):

$$\frac{dy}{dt} = 5(y - t^2)$$

Se a condição inicial é $y(0) = 0,08$, obtenha uma solução de $t = 0$ a 5:

- (a) Analiticamente.
- (b) Usando o método RK de quarta ordem com um tamanho de passo constante de 0,03125.
- (c) Usando a função do MATLAB `ode45`.
- (d) Usando a função do MATLAB `ode23s`.
- (e) Usando a função do MATLAB `ode23tb`.

Apresente seus resultados na forma de gráficos.

23.9 Lembre-se, do Exemplo 20.5, que a função `humps` seguinte exibe tanto regiões planas quanto íngremes em um intervalo relativamente curto de x ,

$$f(x) = \frac{1}{(x - 0,3)^2 + 0,01} + \frac{1}{(x - 0,9)^2 + 0,04} - 6$$

Determine o valor da integral definida dessa função entre $x = 0$ e 1 usando as funções (a) `quad` e (b) `ode45`.

23.10 As oscilações de um pêndulo podem ser simuladas com o modelo não linear seguinte:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \operatorname{sen}\theta = 0$$

onde θ é o ângulo de deslocamento, g é a aceleração da gravidade e l é o comprimento do pêndulo. Para pequenos deslocamentos angulares, o $\operatorname{sen}\theta$ é aproximadamente igual a θ e o modelo pode ser linearizado como

$$\frac{d^2\theta}{dt^2} + \frac{g}{l}\theta = 0$$

Use a função `ode45` para determinar θ como uma função do tempo para ambos os modelos linear e não linear, sendo $l = 0,6$ m e $g = 9,81$ m/s². Primeiro, resolva para o caso em que a condição inicial é para um deslocamento pequeno ($\theta = \pi/8$ e $d\theta/dt = 0$). Em seguida, repita o cálculo para um deslocamento grande ($\theta = \pi/2$). Para cada caso, trace as simulações linear e não linear no mesmo gráfico.

23.11 Empregue a opção `events` descrita na Seção 23.1.2 para determinar o período de um pêndulo linear de 1 m de comprimento (ver descrição no Problema 23.10). Calcule o período para as seguintes condições iniciais: (a) $\theta = \pi/8$, (b) $\theta = \pi/4$, e (c) $\theta = \pi/2$. Para os três casos, definida a velocidade angular inicial igual a zero. (Sugestão: Uma boa maneira de calcular o período é determinar quanto tempo o pêndulo leva para chegar em $\theta = 0$ [isto é, a parte inferior de seu arco]). O período é igual a quatro vezes esse valor.

23.12 Repita o Problema 23.11, mas para o pêndulo não linear descrito no Problema 23.10.

23.13 O sistema seguinte é um exemplo clássico de EDOs rígidas, que podem ocorrer na solução problemas de cinética de reações químicas:

$$\frac{dc_1}{dt} = -0,013c_1 - 1000c_1c_3$$

$$\frac{dc_2}{dt} = -2500c_2c_3$$

$$\frac{dc_3}{dt} = -0,013c_1 - 1000c_1c_3 - 2500c_2c_3$$

Resolva essas equações de $t = 0$ a 50 , com condições iniciais $c_1(0) = c_2(0) = 1$ e $c_3(0) = 0$. Também, utilize as funções padrões (por exemplo, `ode45`) e rígidas (por exemplo, `ode23s`) do MATLAB para obter suas soluções.

23.14 A EDO de segunda ordem seguinte é considerada não linear:

$$\frac{d^2y}{dx^2} = -1001 \frac{dy}{dx} - 1000y$$

Resolva essa equação diferencial (a) analiticamente e (b) numericamente para $x = 0$ a 5 . Para (b) use uma abordagem implícita com $h = 0,5$. Observe que as condições iniciais são $y(0) = 1$ e $y'(0) = 0$. Apresente ambos os resultados em um gráfico.

23.15 Considere uma haste fina de comprimento l se movendo no plano xy , como mostra a Figura P23.15. A haste está fixada por um pino em uma extremidade e existe uma massa na outra. Observe que $g = 9,81$ m/s² e $l = 0,5$ m. Esse sistema pode ser resolvido por meio de

$$\ddot{\theta} - \frac{g}{l}\theta = 0$$

Seja $\theta = 0$ e $\dot{\theta} = 0,25$ rad/s. Resolva por meio de qualquer método estudado neste capítulo. Faça um gráfico do ângulo em função do tempo e da velocidade angular em função do tempo. (Sugestão: Decomponha a EDO de segunda ordem).

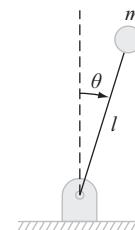


FIGURA P23.15

23.16 Dada a EDO de primeira ordem:

$$\frac{dx}{dt} = -700x - 1000e^{-t}$$

$$x(t = 0) = 4$$

Resolva essa equação diferencial usando um método numérico para o período de tempo $0 \leq t \leq 5$. Também, resolva analiticamente e trace as soluções analítica e numérica para o transiente rápido e a fase de transição lenta da escala de tempo.

23.17 Resolva a seguinte equação diferencial de $t = 0$ a 2

$$\frac{dy}{dt} = -10y$$

com a condição inicial $y(0) = 1$. Use as técnicas seguintes para obter suas soluções: (a) analiticamente, (b) o método de Euler explícito, e (c) o método de Euler implícito. Para (b) e (c) use $h = 0,1$, e 0,2. Trace seus resultados.

23.18 As equações de Lotka-Volterra descritas na Seção 22.6 foram refinadas para incluir fatores adicionais que impactam a dinâmica predador-presa. Por exemplo, além da predação, a população de presa pode ser limitada por outros fatores tais como espaço. A limitação de espaço pode ser incorporada no modelo como uma capacidade de carga (lembre-se do modelo logístico descrito no Problema 22.5) como em

$$\begin{aligned}\frac{dx}{dt} &= a \left(1 - \frac{x}{K}\right)x - bxy \\ \frac{dy}{dt} &= -cy + dxy\end{aligned}$$

onde K é a capacidade de carga. Use os mesmos valores de parâmetros e condições iniciais, como na Seção 22.6, para integrar essas equações de $t = 0$ a 100, usando a função `ode45` e desenvolva gráficos dos resultados em função do tempo e do plano de fases.

- (a) Empregue um valor muito grande de $K = 10^8$ para confirmar que você obtém os mesmos resultados como na Seção 22.6.
- (b) Compare (a) com a capacidade de carga mais realista de $K = 200$. Discuta seus resultados.

23.19 Duas massas são presas a uma parede por molas lineares (Figura P23.19). Balanços de forças com base na segunda lei de Newton podem ser escritos como

$$\frac{d^2x_1}{dt^2} = -\frac{k_1}{m_1}(x_1 - L_1) + \frac{k_2}{m_1}(x_2 - x_1 - w_1 - L_2)$$

$$\frac{d^2x_2}{dt^2} = -\frac{k_2}{m_2}(x_2 - x_1 - w_1 - L_2)$$

onde k é a constante das molas, m é a massa, L é o comprimento da mola não esticada e w é a largura da massa. Calcule as posições das massas como uma função do tempo usando os seguintes valores de parâmetros: $k_1 = k_2 = 5$, $m_1 = m_2 = 2$, $w_1 = w_2 = 5$ e $L_1 = L_2 = 2$. Defina as condições iniciais como $x_1 = L_1$ e $x_2 = L_1 + w_1 + L_2 + 6$. Efetue a simulação de $t = 0$ a 20. Faça gráficos em função do tempo para ambos os deslocamentos e ambas as velocidades. Faça também um gráfico de plano de fases de x_1 versus x_2 .

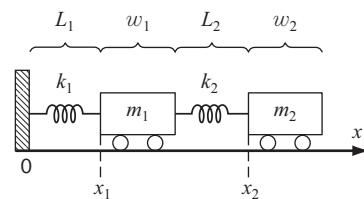


FIGURA P23.19

23.20 Use a função `ode45` para integrar as equações diferenciais para o sistema descrito no Problema 23.19. Gere gráficos (subplots) verticalmente dispostos dos deslocamentos (parte superior) e das velocidades (parte inferior). Empregue a função `fft` para calcular a transformada discreta de Fourier do deslocamento da primeira massa. Gere e trace um espectro de potência para identificar as frequências de ressonância do sistema.

23.21 Efetue os mesmos cálculos como no Problema 23.20, mas com base no primeiro andar da estrutura no Problema 22.22.

23.22 Use a abordagem e o exemplo descritos na Seção 23.1.2, mas determine o tempo, a altura e a velocidade quando o saltador de *bungee jumping* está o mais distante acima do chão, e gere um gráfico da solução.

Problemas de valor de contorno

► OBJETIVOS DO CAPÍTULO

O objetivo principal deste capítulo é apresentar métodos para resolução de problemas de valor de contorno para EDOs. Após esta leitura completa, você será capaz de:

- ✓ Entender a diferença entre problemas de valor inicial e problemas de valor de contorno.
- ✓ Aprender como expressar uma EDO de n -ésima ordem como um sistema de n EDOs de primeira ordem.
- ✓ Aprender como implementar o método *shooting* (método do tiro) para EDOs lineares por meio de interpolação linear, para gerar “tiros” acurados.
- ✓ Entender como condições de contorno na derivada são incorporadas no método do tiro.
- ✓ Aprender a resolver EDOs não lineares com o método do tiro por meio de localização de raízes, para gerar “tiros” acurados.
- ✓ Aprender como implementar o método de diferenças finitas.
- ✓ Entender como as condições de contorno na derivada são incorporadas no método de diferenças finitas.
- ✓ Compreender como resolver EDOs não lineares com o método de diferenças finitas por meio de técnicas de localização de raízes para sistemas de equações algébricas não lineares.

VOCÊ TEM UM PROBLEMA

Até este ponto, calculamos a velocidade de um saltador de *bungee jumping* em queda livre pela integração de uma única EDO:

$$\frac{dv}{dt} = g - \frac{c_d}{m}v^2 \quad (24.1)$$

Considere que, em vez da velocidade, seja solicitado que você determine a posição do saltador como uma função do tempo. Uma maneira de fazer isso é reconhecer que a velocidade é a primeira derivada da distância:

$$\frac{dx}{dt} = v \quad (24.2)$$

Assim, pela solução do sistema de duas EDOs representado pelas Equações (24.1) e (24.2), podemos determinar simultaneamente tanto a velocidade quanto a posição.

No entanto, como estamos agora integrando duas EDOs, precisamos de duas condições para obter a solução. Já estamos familiarizados com uma maneira de fazer isso para o caso em que temos valores tanto para a posição como para a velocidade, no instante de tempo inicial:

$$\begin{aligned}x(t = 0) &= x_i \\v(t = 0) &= v_i\end{aligned}$$

Dadas essas condições, podemos facilmente integrar as EDOs usando as técnicas numéricas descritas nos Capítulos 22 e 23. Isso é conhecido como um *problema de valor inicial*.

Mas, e se nós não conhecemos os valores para a posição e a velocidade em $t = 0$? Consideremos que saibamos a posição inicial; porém, em vez de termos a velocidade inicial, queremos que o saltador esteja em uma posição específica em um instante de tempo posterior; ou seja

$$\begin{aligned}x(t = 0) &= x_i \\x(t = t_f) &= x_f\end{aligned}$$

Como as duas condições são dadas em valores diferentes da variável independente, isso é conhecido como um *problema de valor de contorno*.

Esses problemas necessitam de técnicas de solução especiais. Algumas dessas estão relacionadas com os métodos para problemas de valor inicial que foram descritos nos dois capítulos anteriores. No entanto, outros empregam estratégias totalmente diferentes para obter soluções. Este capítulo destina-se a apresentar os mais comuns desses métodos.

24.1 INTRODUÇÃO E CONTEXTUALIZAÇÃO

24.1.1 O que são problemas de valor de contorno?

Uma equação diferencial ordinária é acompanhada de condições auxiliares, que são usadas para calcular as constantes de integração que resultam durante a solução da equação. Para uma equação de n -ésima ordem, n condições são necessárias. Se todas as condições são especificadas para o mesmo valor da variável independente, então estamos lidando com um *problema de valor inicial* (Figura 24.1a). Até este ponto, o material da Parte 6 (Capítulos 22 e 23) foi dedicado a esse tipo de problema.

Em contraste, frequentemente há casos em que as condições não são conhecidas em um único ponto, mas, em vez disso, são dadas em valores diferentes da variável independente. Como esses valores normalmente são especificados nos pontos extremos ou fronteiras de um sistema, eles são geralmente chamados de *problemas de valor de contorno ou de fronteira* (Figura 24.1b). Uma variedade de aplicações significativas de engenharia se enquadra dentro dessa classe. Neste capítulo, descrevemos algumas das abordagens básicas para solução desses problemas.

24.1.2 Problemas de valor de contorno em engenharia e em ciência

No início deste capítulo, mostramos como a determinação da posição e da velocidade de um corpo em queda pode ser formulado como problema de valor de contorno. Para aquele exemplo, um par de EDOs foi integrado no tempo. Embora outros exemplos que envolvam a variável tempo possam ser desenvolvidos, problemas de valor de contorno surgem mais naturalmente quando se integra no espaço. Isso acontece porque condições auxiliares são geralmente especificadas em diferentes pontos do espaço.

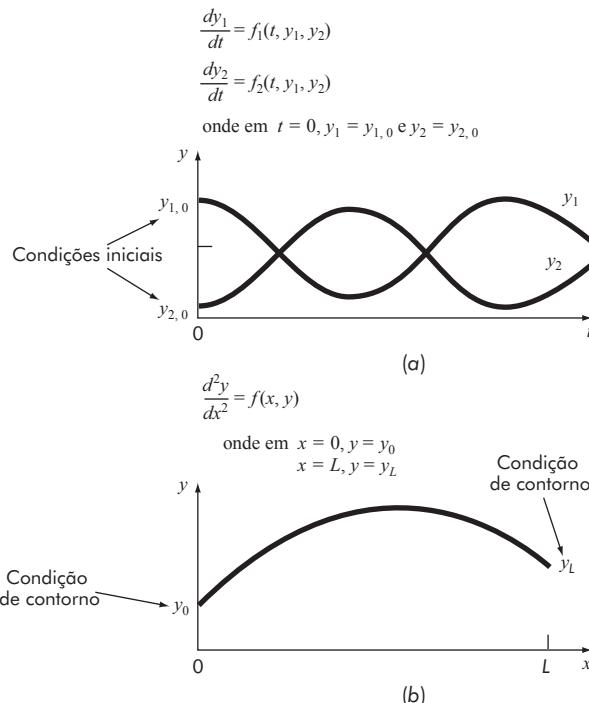


FIGURA 24.1 Problema de valor inicial versus problemas de valor de contorno.
 (a) Um problema de valor inicial no qual todas as condições são especificadas no mesmo valor da variável independente.
 (b) Um problema de contorno em que as condições são especificadas em valores diferentes da variável independente.

Um caso ilustrativo é a simulação da distribuição de temperatura em estado estacionário para uma haste fina e longa, posicionada entre duas paredes de temperatura constante (Figura 24.2). As dimensões da seção transversal da haste são suficientemente pequenas, de modo que os gradientes de temperatura radiais são mínimos e, consequentemente, a temperatura é uma função exclusiva da coordenada axial x . O calor é transferido ao longo do eixo longitudinal da haste por condução e entre a haste e o gás circundante por convecção. Para esse exemplo, a radiação é considerada desprezível¹.

Como mostra a Figura 24.2, um balanço de calor pode ser tomado em torno de um elemento de espessura Δx como

$$0 = q(x)A_c - q(x + \Delta x)A_c + hA_s(T_\infty - T) \quad (24.3)$$

onde $q(x)$ é o fluxo que entra no elemento devido à condução [$J/(m^2 \cdot s)$], $q(x + \Delta x)$ é o fluxo que sai do elemento devido à condução [$J/(m^2 \cdot s)$], A_c é a área da seção transversal (m^2) = πr^2 , r é o raio (m), h é o coeficiente de transferência de calor por convecção [$J/(m^2 \cdot K \cdot s)$], A_s é a área da superfície do elemento (m^2) = $2\pi r\Delta x$, T_∞ é a temperatura do gás circundante (K) e T é a temperatura da haste (K).

¹ A radiação foi incorporada neste problema, mais adiante neste capítulo, no Exemplo 24.4.

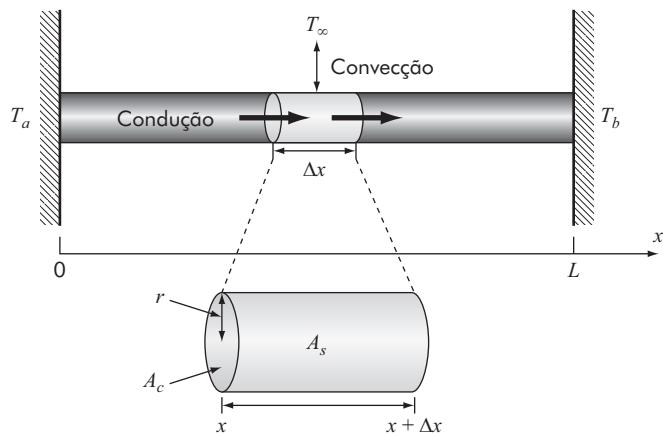


FIGURA 24.2 Balanço de calor para um elemento diferencial de uma haste aquecida sujeita a condução e a convecção.

A Equação (24.3) pode ser dividida pelo volume do elemento ($\pi r^2 \Delta x$), para fornecer

$$0 = \frac{q(x) - q(x + \Delta x)}{\Delta x} + \frac{2h}{r}(T_\infty - T)$$

Tomando o limite $\Delta x \rightarrow 0$, temos

$$0 = -\frac{dq}{dx} + \frac{2h}{r}(T_\infty - T) \quad (24.4)$$

O fluxo pode ser relacionado ao gradiente de temperatura pela *lei de Fourier*:

$$q = -k \frac{dT}{dx} \quad (24.5)$$

onde k é o coeficiente de condutividade térmica [$J/(s \cdot m \cdot K)$]. A Equação (24.5) pode ser derivada com relação a x , substituída na Equação (24.4), e o resultado dividido por k para fornecer

$$0 = \frac{d^2T}{dx^2} + h'(T_\infty - T) \quad (24.6)$$

onde h' é um parâmetro de transferência de calor mais amplo que reflete os impactos relativos de convecção e condução (m^{-2}) = $2h/(rk)$.

A Equação (24.6) representa um modelo matemático que pode ser usado para calcular a temperatura ao longo da dimensão axial da haste. Como ela é uma EDO de segunda ordem, duas condições são necessárias para obter uma solução. Como mostra a Figura 24.2, um caso comum é aquele em que as temperaturas nas extremidades da haste são mantidas em valores fixos. Essas condições podem ser expressas matematicamente como

$$T(0) = T_a$$

$$T(L) = T_b$$

Porque representam fisicamente as condições nas “fronteiras” da haste dão origem à terminologia: condições de contorno ou de fronteira.

Dadas essas condições, o modelo representado pela Equação (24.6) pode ser resolvido. Como essa EDO particular é linear, uma solução analítica é possível, como mostra o exemplo a seguir.

EXEMPLO 24.1 Solução analítica para uma haste aquecida

Use o cálculo para resolver a Equação (24.6) para uma haste de 10 m com $h' = 0,05 \text{ m}^{-2}$ [$h = 1 \text{ J}/(\text{m}^2 \cdot \text{K} \cdot \text{s})$, $r = 0,2 \text{ m}$, $k = 200 \text{ J}/(\text{s} \cdot \text{m} \cdot \text{K})$], $T_\infty = 200 \text{ K}$ e as condições de contorno:

$$T(0) = 300 \text{ K} \quad T(10) = 400 \text{ K}$$

Solução. Essa EDO pode ser resolvida de diversas formas. Uma abordagem simples é expressar a equação como

$$\frac{d^2T}{dx^2} - h'T = -h'T_\infty$$

Considerando que essa é uma EDO linear com coeficientes constantes, a solução geral pode ser imediatamente obtida igualando o lado direito a zero e admitindo uma solução da forma $T = e^{\lambda x}$. Substituindo essa solução junto com sua segunda derivada na forma homogênea da EDO, temos

$$\lambda^2 e^{\lambda x} - h' e^{\lambda x} = 0$$

que pode ser resolvida para $\lambda = \pm\sqrt{h'}$. Assim, a solução geral é

$$T = A e^{\lambda x} + B e^{-\lambda x}$$

onde A e B são constantes de integração. Usando o método de coeficientes indeterminados, podemos obter a solução particular $T = T_\infty$. Portanto, a solução total é

$$T = T_\infty + A e^{\lambda x} + B e^{-\lambda x}$$

As constantes podem ser calculadas pela aplicação das condições de contorno

$$T_a = T_\infty + A + B$$

$$T_b = T_\infty + A e^{\lambda L} + B e^{-\lambda L}$$

Essas duas equações podem ser resolvidas simultaneamente para

$$A = \frac{(T_a - T_\infty)e^{-\lambda L} - (T_b - T_\infty)}{e^{-\lambda L} - e^{\lambda L}}$$

$$B = \frac{(T_b - T_\infty) - (T_a - T_\infty)e^{\lambda L}}{e^{-\lambda L} - e^{\lambda L}}$$

Substituindo os valores dos parâmetros desse problema, obtemos $A = 20,4671$ e $B = 79,5329$. Assim, a solução final é

$$T = 200 + 20,4671 e^{\sqrt{0,05}x} + 79,5329 e^{-\sqrt{0,05}x} \quad (24.7)$$

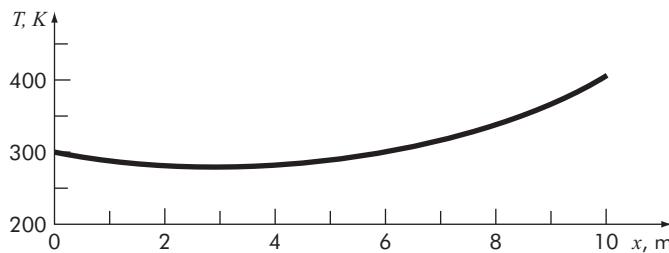


FIGURA 24.3 Solução analítica para a haste aquecida.

Como mostra a Figura 24.3, a solução é uma curva suave que liga as duas temperaturas das fronteiras. A temperatura no meio sofre uma queda devido à perda de calor por convecção para o gás circundante mais frio.

Nas seções seguintes, ilustraremos abordagens numéricas para resolução do mesmo problema que acabamos de resolver analiticamente no Exemplo 24.1. A solução analítica exata será útil para avaliar a acurácia das soluções obtidas com os métodos numéricos aproximados.

24.2 O MÉTODO SHOOTING

O método *shooting* (ou método do tiro) é baseado na conversão de um problema de valor de contorno em um problema de valor inicial equivalente. Uma abordagem de tentativa e erro é, então, implementada para desenvolver uma solução para a versão de valor inicial que satisfaça as condições de contorno dadas.

Embora o método possa ser empregado para equações não lineares e de ordem mais alta, ele é bem ilustrado para uma EDO linear de segunda ordem, tal como a equação da haste aquecida descrita na seção anterior:

$$0 = \frac{d^2T}{dx^2} + h'(T_\infty - T) \quad (24.8)$$

sujeita às condições de contorno

$$\begin{aligned} T(0) &= T_a \\ T(L) &= T_b \end{aligned}$$

Esse problema de contorno pode ser convertido em um problema de valor inicial definindo-se a taxa de variação da temperatura, ou *gradiente*, como

$$\frac{dT}{dx} = z \quad (24.9)$$

e reescrevendo a Equação (24.8) como

$$\frac{dz}{dx} = -h'(T_\infty - T) \quad (24.10)$$

Assim, convertemos a equação de segunda ordem única [Equação (24.8)] em um par de EDOs de primeira ordem [Equações (24.9) e (24.10)].

Se tivéssemos condições iniciais para T e z , poderíamos resolver essas equações como um problema de valor inicial com os métodos descritos nos Capítulos 22 e 23. No entanto, como temos apenas um valor inicial para uma das variáveis, $T(0) = T_a$, simplesmente fazemos um “palpite” para a outra, $z(0) = z_{a1}$, e, então, efetuamos a integração.

Depois disso, teremos gerado um valor de T no fim do intervalo, o qual chamaremos de T_{b1} . A menos que estejamos com muita sorte, esse resultado irá diferir do valor desejado T_b .

Agora, considerando que o valor de T_{b1} é muito alto ($T_{b1} > T_b$), faz sentido pensar que um valor menor da inclinação inicial, $z(0) = z_{a2}$, pode resultar em uma predição melhor. Usando essa nova estimativa, podemos integrar novamente para gerar um segundo resultado no fim do intervalo T_{b2} . Podemos então continuar esse processo na forma de tentativa e erro até atingirmos uma estimativa para $z(0)$ que resulte no valor correto de $T(L) = T_b$.

Neste ponto, a origem do nome *método do tiro* deve ser bastante clara. Assim como se ajusta o ângulo de um canhão para atingir um alvo, ajustamos a trajetória de nossa solução, considerando valores de $z(0)$, até atingirmos nosso alvo $T(L) = T_b$.

Embora possamos continuar tentando valores, uma estratégia mais eficiente é possível para EDOs lineares. Nesses casos, a trajetória do tiro perfeito z_a é linearmente relacionada aos resultados de nossos tiros errados (z_{a1}, T_{b1}) e (z_{a2}, T_{b2}). Consequentemente, interpolação linear pode ser empregada para chegar à trajetória desejada:

$$z_a = z_{a1} + \frac{z_{a2} - z_{a1}}{T_{b2} - T_{b1}}(T_b - T_{b1}) \quad (24.11)$$

A abordagem pode ser ilustrada com um exemplo.

EXEMPLO 24.2 O método do tiro para uma EDO linear

Use o método do tiro para resolver a Equação (24.6) para as mesmas condições do Exemplo 24.1: $L = 10$ m, $h' = 0,05$ m $^{-2}$, $T_\infty = 200$ K, $T(0) = 300$ K e $T(10) = 400$ K.

Solução. A Equação (24.6) é expressa como um par de EDOs de primeira ordem:

$$\begin{aligned} \frac{dT}{dx} &= z \\ \frac{dz}{dx} &= -0,05(200 - T) \end{aligned}$$

Junto com o valor inicial para temperatura, $T(0) = 300$ K, tentamos arbitrariamente um valor de $z_{a1} = -5$ K/m para o valor inicial de $z(0)$. A solução é então obtida integrando o par de EDOs de $x = 0$ a 10. Podemos fazer isso com a função `ode45` do MATLAB, definindo um arquivo-M para armazenar as equações diferenciais:

```
function dy=Ex2402(x,y)
dy=[y(2);-0.05*(200-y(1))];
```

Podemos então gerar a solução como

```
>> [t,y]=ode45(@Ex2402,[0 10],[300,-5]);
>> Tb1=y(length(y))
Tb1 =
569.7539
```

Assim, obtemos um valor no fim do intervalo de $T_{b1} = 569,7539$ (Figura 24.4a), que difere da condição de contorno desejada de $T_b = 400$. Portanto, fazemos outra estimativa $z_{a2} = -20$ e efetuamos os cálculos novamente. Desta vez, o resultado de $T_{b2} = 259,5131$ é obtido (Figura 24.4b).

Agora, como a EDO original é linear, podemos usar a Equação (24.1) para determinar a trajetória correta para produzir o tiro perfeito:

$$z_a = -5 + \frac{-20 - (-5)}{259,5131 - 569,7539}(400 - 569,7539) = -13,2075$$

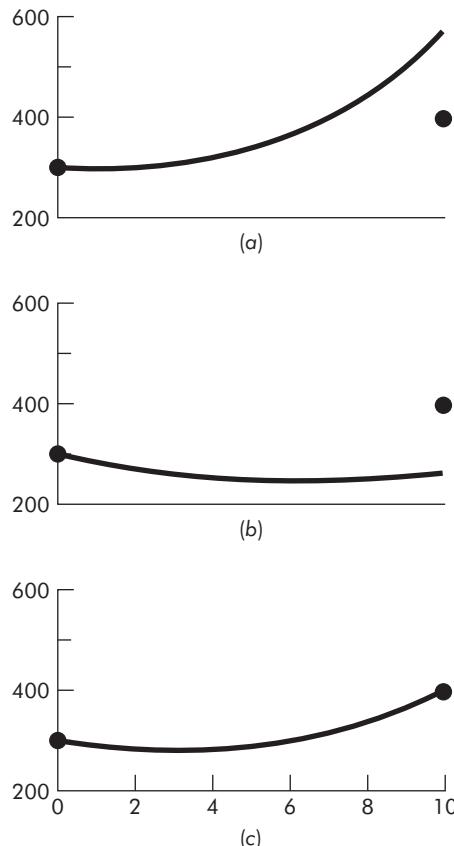


FIGURA 24.4 Temperatura (K) versus distância (m) calculados com o método do tiro: (a) o primeiro “tiro”, (b) o segundo “tiro” e (c) o “tiro” exato final.

Esse valor pode então ser usado com a função `ode45` para gerar a solução correta, como mostra a Figura 24.4c.

Embora não seja óbvio a partir do gráfico, a solução analítica também está traçada na Figura 24.4c. Assim, o método do tiro produz uma solução que é virtualmente indistinguível do resultado exato.

24.2.1 Condições de contorno na derivada

A *condição de contorno de Dirichlet* ou fixa, discutida até este ponto, é apenas uma entre os diversos tipos que são utilizados na engenharia e na ciência. Uma alternativa comum é o caso em que a derivada é dada, em geral conhecido como *condição de contorno de Neumann*.

Como ele já está configurado para calcular tanto a variável dependente quanto sua derivada, a incorporação das condições de contorno na derivada no método do tiro é relativamente simples.

De modo semelhante ao caso da condição de contorno fixa, a princípio expressamos a EDO de segunda ordem como um par de EDOs de primeira ordem. Neste ponto, uma das condições iniciais necessárias, seja a variável dependente ou a sua derivada, será desconhecida. Com base em tentativas para a condição inicial desconhecida, geramos soluções para calcular a condição da extremidade dada. Tal como com a condição inicial, pode ser esta também para a variável dependente ou para a sua derivada. Para EDOs lineares, a interpolação pode então ser usada para determinar o valor da condição inicial desconhecida necessária para gerar o tiro final “perfeito” que atinge a condição da extremidade.

EXEMPLO 24.3 O método do tiro com condições de contorno na derivada

Use o método do tiro para resolver a Equação (24.6) para a haste no Exemplo 24.1: $L = 10\text{ m}$, $h' = 0,05\text{ m}^{-2}$ [$h = 1\text{ J}/(\text{m}^2 \cdot \text{K} \cdot \text{s})$, $r = 0,2\text{ m}$, $k = 200\text{ J}/(\text{s} \cdot \text{m} \cdot \text{K})$], $T_\infty = 200\text{ K}$ e $T(10) = 400\text{ K}$. Porém, para esse caso, em vez de ter uma temperatura fixa de 300 K , a extremidade esquerda está sujeita à convecção, como mostra a Figura 24.5. Para simplificar, consideremos que o coeficiente de transferência de calor por convecção para a área da extremidade é o mesmo que para a superfície da haste.

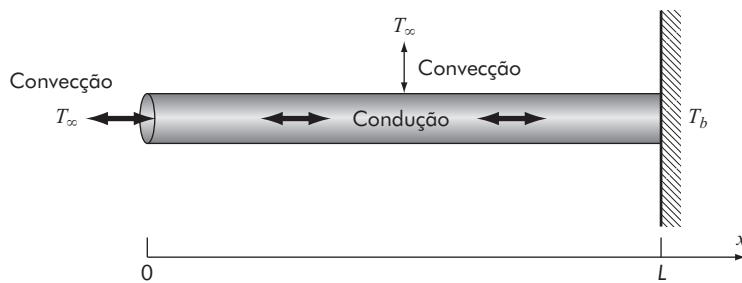


FIGURA 24.5 Haste com uma condição de contorno convectiva em uma extremidade e uma temperatura fixa na outra.

Solução. Como no Exemplo 24.2, a Equação (24.6) é a princípio expressa como

$$\frac{dT}{dx} = z$$

$$\frac{dz}{dx} = -0,05(200 - T)$$

Embora possa não ser óbvio, a convecção através da extremidade é equivalente à especificação de um gradiente como condição de contorno. Para observar isso, devemos reconhecer que, como o sistema está em estado estacionário, a convecção deve ser igual à condução na extremidade esquerda da haste ($x = 0$). Usando a lei de Fourier [Equação (24.5)] para representar a condução, o balanço de calor na extremidade pode ser formulado como

$$hA_c(T_\infty - T(0)) = -kA_c \frac{dT}{dx}(0) \quad (24.12)$$

Essa equação pode ser resolvida para o gradiente

$$\frac{dT}{dx}(0) = \frac{h}{k}(T(0) - T_\infty) \quad (24.13)$$

Se considerarmos um valor para a temperatura, pode-se perceber que essa equação especifica o gradiente.

O método do tiro é implementado por tentativas arbitrárias de um valor para $T(0)$. Se escolhermos um valor de $T(0) = T_{a1} = 300$ K, a Equação (24.13) então fornece o valor inicial para o gradiente

$$z_{a1} = \frac{dT}{dx}(0) = \frac{1}{200}(300 - 200) = 0,5$$

A solução é obtida integrando o par de EDOs de $x = 0$ a 10. Isso pode ser feito com a função `ode45` do MATLAB, definindo o princípio um arquivo-M para armazenar as equações diferenciais da mesma maneira como no Exemplo 24.2. Podemos então gerar a solução como

```
>> [t,y]=ode45(@Ex2402,[0 10],[300,0.5]);
>> Tb1=y(length(y))
Tb1 =
683.5088
```

Como esperado, o valor na extremidade do intervalo de $T_{b1} = 683,5088$ K difere da condição de contorno desejada de $T_b = 400$. Portanto, fazemos outra tentativa $T_{a2} = 150$ K, que corresponde a $z_{a2} = -0,25$, e efetuamos os cálculos novamente.

```
>> [t,y]=ode45(@Ex2402,[0 10],[150,-0.25]);
>> Tb2=y(length(y))
Tb2 =
-41.7544
```

Interpolação linear pode então ser empregada para calcular a temperatura inicial correta:

$$T_a = 300 + \frac{150 - 300}{-41,7544 - 683,5088}(400 - 683,5088) = 241,3643 \text{ K}$$

que corresponde a um gradiente de $z_a = 0,2068$. Por meio dessas condições iniciais, a função `ode45` pode ser empregada para gerar a solução correta, como mostra a Figura 24.6. Pode-se verificar que a condição de contorno foi satisfeita com a substituição das condições iniciais na Equação (24.12), para fornecer

$$1 \frac{J}{m^2 K s} \pi \times (0,2 \text{ m})^2 \times (200 \text{ K} - 241,3643 \text{ K}) = -200 \frac{J}{m K s} \pi \times (0,2 \text{ m})^2 \times 0,2068 \frac{K}{m}$$

que pode ser calculada para produzir $-5,1980 \text{ J/s} = -5,1980 \text{ J/s}$. Assim, condução e convecção são iguais, e a transferência de calor para fora da extremidade esquerda da haste ocorre a uma taxa de $5,1980 \text{ W}$.

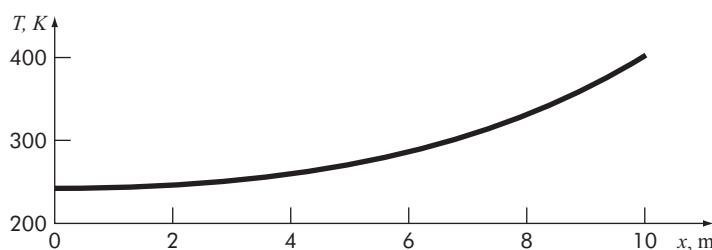


FIGURA 24.6 A solução de uma EDO de segunda ordem com uma condição de contorno convectiva em uma extremidade e uma temperatura fixa na outra.

24.2.2 O método shooting para EDOs não lineares

Para problemas de contorno não lineares, a interpolação ou extrapolação linear pelos dois pontos das soluções não resultará necessariamente em uma estimativa acurada da condição de contorno exigida para alcançar uma solução exata. Uma alternativa é fazer três aplicações do método do tiro e usar um polinômio interpolador quadrático, a fim de obter uma estimativa da condição de contorno adequada. Entretanto, é pouco provável que essa abordagem forneça uma resposta exata, e iterações adicionais seriam necessárias para obter a solução.

Outra abordagem para um problema não linear envolve reformulá-lo como um problema de raízes. Lembre-se de que a forma geral de um problema de raízes é encontrar o valor de x que torne a função $f(x) = 0$. Agora, usaremos o problema da haste aquecida para entender como método do tiro pode ser reformulado dessa forma.

Perceba que a solução do par de equações diferenciais também é uma “função”, em termos de tentativas de uma condição na extremidade esquerda da haste, z_a , e a integração fornece uma previsão da temperatura na extremidade direita, T_b . Logo, podemos pensar na integração como

$$T_b = f(z_a)$$

Isto é, ela representa um processo pelo qual uma tentativa de z_a produz uma previsão de T_b . Visto dessa forma, podemos desejar o valor de z_a que forneça um valor específico de T_b . Se, como no exemplo, quisermos $T_b = 400$, o problema pode ser enunciado como

$$400 = f(z_a)$$

Trazendo a meta de 400 para o lado direito da equação, geramos uma nova função $res(z_a)$ que representa a diferença, ou o *resíduo*, entre o que temos, $f(z_a)$, e o que queremos, 400.

$$res(z_a) = f(z_a) - 400$$

Se fizermos essa função anular-se, teremos obtido a solução. O próximo exemplo ilustra essa abordagem.

EXEMPLO 24.4 O método do tiro para EDOs não lineares

Embora tenha sido útil para ilustrar o método do tiro, a Equação (24.6) não foi um modelo completamente realista para uma haste aquecida. Em princípio, essa haste perderia calor por mecanismos tais como radiação, que são não lineares.

Considere que a seguinte EDO não linear seja usada para simular a temperatura da haste aquecida:

$$0 = \frac{d^2T}{dx^2} + h'(T_\infty - T) + \sigma' (T_\infty^4 - T^4)$$

onde σ' é um parâmetro de transferência de calor mais amplo que reflete os impactos relativos de radiação e condução $= 2,7 \times 10^{-9} \text{ K}^{-3}\text{m}^{-2}$. Essa equação pode servir para ilustrar como o método do tiro é usado para resolver um problema de contorno não linear de dois pontos. As condições remanescentes do problema são especificadas como no Exemplo 24.2: $L = 10 \text{ m}$, $h = 0,05 \text{ m}^{-2}$, $T_\infty = 200 \text{ K}$, $T(0) = 300 \text{ K}$ e $T(10) = 400 \text{ K}$.

Solução. Assim como com a EDO linear, a equação de segunda ordem não linear é a princípio expressa como duas EDOs de primeira ordem:

$$\begin{aligned} \frac{dT}{dx} &= z \\ \frac{dz}{dx} &= -0,05(200 - T) - 2,7 \times 10^{-9}(1,6 \times 10^9 - T^4) \end{aligned}$$

Um arquivo-M pode ser desenvolvido para calcular os lados direitos dessas equações:

```
function dy=dydxn(x,y)
dy=[y(2);-0.05*(200-y(1))-2.7e-9*(1.6e9-y(1)^4)];
```

Em seguida, podemos implementar uma função para armazenar o resíduo que tentaremos levar a zero como

```
function r=res(za)
[x,y]=ode45(@dydxn,[0 10],[300 za]);
r=y(length(x),1)-400;
```

Observe como usamos a função `ode45` para resolver as duas EDOs, a fim de gerar a temperatura na extremidade da haste: `y(length(x),1)`. Podemos então encontrar a raiz com a função `fzero`:

```
>> fzero(@res,-50)
ans =
-41.7434
```

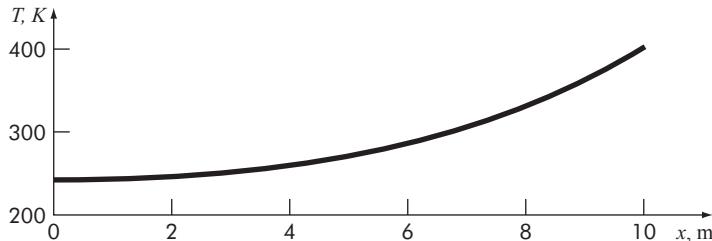


FIGURA 24.7 O resultado de usar o método do tiro para resolver um problema não linear.

Assim, percebemos que, se definirmos a trajetória inicial $z(0) = -41,7434$, a função resíduo será levada a zero, e a condição de contorno da temperatura $T(10) = 400$ na extremidade da haste deve ser satisfeita. Isso pode ser verificado gerando a solução completa e traçando as temperaturas *versus* x :

```
>> [x,y]=ode45(@dydxn,[0 10],[300 fzero(@res,-50)]);
>> plot(x,y(:,1))
```

O resultado é mostrado na Figura 24.7 com o caso linear original do Exemplo 24.2. Como esperado, o caso não linear é mais curvado para baixo do que o modelo linear. Isso é devido à perda de calor adicional para o gás circundante por radiação.

24.3 MÉTODOS DE DIFERENÇAS FINITAS

As alternativas mais comuns ao método do tiro são as abordagens por diferenças finitas, em que estas (Capítulo 21) substituem as derivadas na equação original.

Assim, uma equação diferencial linear é transformada em um conjunto de equações algébricas simultâneas que podem ser resolvidas usando os métodos apresentados na Parte 3.

Podemos ilustrar a abordagem para o modelo da haste aquecida [Equação (24.6)]:

$$0 = \frac{d^2T}{dx^2} + h'(T_\infty - T) \quad (24.14)$$

O domínio da solução é a princípio dividido em uma série de nós (Figura 24.8). Em cada nó, aproximações por diferenças finitas podem ser escritas para as derivadas na equação. Por exemplo, no nó i , a segunda derivada pode ser representada por (Figura 21.5):

$$\frac{d^2T}{dx^2} = \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} \quad (24.15)$$

Essa aproximação pode ser substituída na Equação (24.14), para fornecer

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} + h'(T_\infty - T_i) = 0$$

Assim, a equação diferencial foi convertida em uma equação algébrica. Agrupando os termos, obtemos

$$-T_{i-1} + (2 + h'\Delta x^2)T_i - T_{i+1} = h'\Delta x^2 T_\infty \quad (24.16)$$

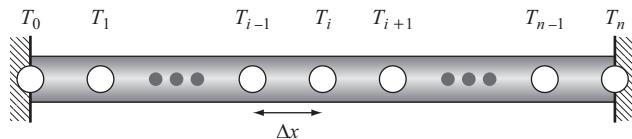


FIGURA 24.8 Para implementar a abordagem por diferenças finitas, a haste aquecida é dividida em uma série de nós.

Essa equação pode ser escrita para cada um dos $n - 1$ nós interiores da haste. O primeiro e último nós, T_0 e T_n , respectivamente, são especificados por condições de contorno. Portanto, o problema se reduz à resolução de $n - 1$ equações algébricas lineares simultâneas para as $n - 1$ incógnitas.

Antes de fornecer um exemplo, ressaltamos duas características interessantes da Equação (24.16). Observe que, como os nós são enumerados consecutivamente, e como cada equação consiste em um nó (i) e seus vizinhos adjacentes ($i - 1$ e $i + 1$), os sistemas resultantes de equações algébricas lineares serão tridiagonais, e, assim, podem ser resolvidos com algoritmos eficientes que estão disponíveis para esses sistemas (lembre-se da Seção 9.4).

Além disso, uma inspeção dos coeficientes no lado esquerdo da Equação (24.16) indica que o sistema de equações lineares também será diagonal dominante. Portanto, soluções convergentes também podem ser geradas com técnicas iterativas como o método de Gauss-Seidel (Seção 12.1).

EXEMPLO 24.5 Aproximação por diferenças finitas para problemas de valor de contorno

Use a abordagem por diferenças finitas para resolver o mesmo problema como nos Exemplos 24.1 e 24.2. Use quatro nós interiores com um tamanho de segmento de $\Delta x = 2$ m.

Solução. Empregando os parâmetros do Exemplo 24.1 e $x = 2$ m, podemos escrever a Equação (24.16) para cada um dos nós interiores da haste. Por exemplo, para o nó 1:

$$-T_0 + 2,2T_1 - T_2 = 40$$

Substituindo a condição de contorno $T_0 = 300$, obtemos

$$2,2T_1 - T_2 = 340$$

Depois de escrever a Equação (24.16) para os outros nós interiores, as equações podem ser reunidas na forma matricial como

$$\begin{bmatrix} 2,2 & -1 & 0 & 0 \\ -1 & 2,2 & -1 & 0 \\ 0 & -1 & 2,2 & -1 \\ 0 & 0 & -1 & 2,2 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 340 \\ 40 \\ 40 \\ 440 \end{Bmatrix}$$

Observe que a matriz é tanto tridiagonal quanto diagonal dominante.

O MATLAB pode ser usado para gerar a solução:

```
>> A=[2.2 -1 0 0;
-1 2.2 -1 0;
0 -1 2.2 -1;
0 0 -1 2.2];
>> b=[340 40 40 440]';
>> T=A\b
T =
283.2660
283.1853
299.7416
336.2462
```

A Tabela 24.1 fornece uma comparação entre a solução analítica [Equação (24.7)] e as soluções numéricas obtidas com o método do tiro (Exemplo 24.2) e o método de diferenças finitas (Exemplo 24.5). Observe que, embora existam algumas discrepâncias, as abordagens numéricas concordam razoavelmente bem com a solução analítica. Além disso, a maior discrepância ocorre para o método de diferenças finitas devido ao espaçamento grosso entre nós que foi usado no Exemplo 24.5. Uma melhor concordância ocorre se um espaçamento entre nós mais fino for utilizado.

TABELA 24.1 Comparação da solução analítica para a temperatura com os resultados obtidos com os métodos do tiro e de diferenças finitas.

<i>x</i>	Solução analítica	Método do tiro	Diferenças finitas
0	300	300	300
2	282,8634	282,8889	283,2660
4	282,5775	282,6158	283,1853
6	299,0843	299,1254	299,7416
8	335,7404	335,7718	336,2462
10	400	400	400

24.3.1 Condições de contorno na derivada

Como mencionado em nossa discussão sobre o método do tiro, a *condição de contorno de Dirichlet* ou fixa é apenas uma entre os diversos tipos que são utilizados na engenharia e na ciência. Uma alternativa comum, chamada *condição de contorno de Neumann*, é o caso em que a derivada é dada.

Podemos usar a haste aquecida introduzida anteriormente neste capítulo para demonstrar como uma condição de contorno na derivada pode ser incorporada na abordagem por diferenças finitas:

$$0 = \frac{d^2T}{dx^2} + h'(T_\infty - T)$$

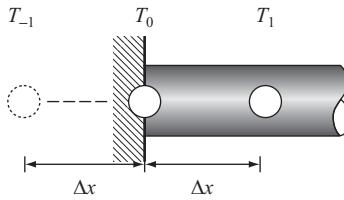


FIGURA 24.9 Um nó de fronteira na extremidade esquerda de uma haste aquecida.
Para aproximar a derivada na fronteira, um nó imaginário é posicionado a uma distância Δx à esquerda da extremidade da haste.

No entanto, em contraste com nossa discussão anterior, iremos prescrever uma condição de contorno da derivada em uma extremidade da haste:

$$\begin{aligned}\frac{dT}{dx}(0) &= T'_a \\ T(L) &= T_b\end{aligned}$$

Dessa forma, temos uma condição de contorno na derivada em uma extremidade do domínio da solução e uma condição de contorno fixa na outra.

Assim como na seção anterior, a haste é dividida em uma série de nós, e uma versão de diferenças finitas da equação diferencial [Equação (24.16)] é aplicada a cada nó interior. No entanto, como sua temperatura não é especificada, o nó na extremidade esquerda também deve ser incluído. A Figura 24.9 apresenta o nó (0) na extremidade esquerda de uma chapa aquecida para a qual a condição de contorno de derivada se aplica. Escrevendo a Equação (24.16) para esse nó, obtemos

$$-T_{-1} + (2 + h'\Delta x^2)T_0 - T_1 = h'\Delta x^2 T_\infty \quad (24.17)$$

Observe que um nó imaginário (-1) situado à esquerda da extremidade da haste é necessário para essa equação. Embora possa parecer que esse ponto exterior representa uma dificuldade, na realidade serve como o veículo para incorporar a condição de contorno da derivada no problema. Isso é feito representando a primeira derivada na dimensão x em (0) pela diferença centrada [Equação (4.25)]:

$$\frac{dT}{dx} = \frac{T_1 - T_{-1}}{2\Delta x}$$

que pode ser resolvida para

$$T_{-1} = T_1 - 2\Delta x \frac{dT}{dx}$$

Agora, temos uma fórmula para T_{-1} que, na realidade, reflete o impacto da derivada. Ela pode ser substituída na Equação (24.17), para fornecer

$$(2 + h'\Delta x^2)T_0 - 2T_1 = h'\Delta x^2 T_\infty - 2\Delta x \frac{dT}{dx} \quad (24.18)$$

Consequentemente, a derivada foi incorporada no balanço.

Um exemplo comum de uma condição de contorno na derivada é a situação em que a extremidade da haste está isolada. Neste caso, a derivada é definida como igual a zero. Essa conclusão segue diretamente da lei de Fourier [Equação (24.5)], porque isolar uma extremidade significa que o fluxo de calor (e, consequentemente, o gradiente de temperatura) deve ser zero. O exemplo seguinte ilustra como a solução é afetada por essas condições de contorno.

EXEMPLO 24.6 Incorporando condições de contorno na derivada

Gere a solução por diferenças finitas para uma haste de 10 m com $\Delta x = 2$ m, $h' = 0,05 \text{ m}^{-2}$, $T_{\infty} = 200 \text{ K}$ e as condições de contorno: $T'_a = 0$ e $T_b = 400 \text{ K}$. Observe que a primeira condição significa que a inclinação da solução deve se aproximar de zero na extremidade esquerda da haste; também, gere a solução para $dT/dx = -20$ em $x = 0$.

Solução. A Equação (24.18) pode ser usada para representar o nó 0 como

$$2,2T_0 - 2T_1 = 40$$

Podemos escrever a Equação (24.16) para os nós interiores. Por exemplo, para o nó 1,

$$-T_0 + 2,2T_1 - T_2 = 40$$

Uma abordagem similar pode ser usada para os nós interiores remanescentes. O sistema de equações final pode ser escrito na forma matricial como

$$\begin{bmatrix} 2,2 & -2 & & & \\ -1 & 2,2 & -1 & & \\ & -1 & 2,2 & -1 & \\ & & -1 & 2,2 & -1 \\ & & & -1 & 2,2 \end{bmatrix} \begin{Bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 40 \\ 40 \\ 40 \\ 40 \\ 440 \end{Bmatrix}$$

Essas equações podem ser resolvidas para

$$T_0 = 243,0278$$

$$T_1 = 247,3306$$

$$T_2 = 261,0994$$

$$T_3 = 287,0882$$

$$T_4 = 330,4946$$

Como mostra a Figura 24.10, a solução é plana em $x = 0$ devido à condição de derivada nula e, então, curva-se para cima para a condição fixa de $T = 400$ em $x = 10$.

Para o caso em que a derivada em $x = 0$ é definida como igual a -20 , as equações simultâneas são

$$\begin{bmatrix} 2,2 & -2 & & & \\ -1 & 2,2 & -1 & & \\ & -1 & 2,2 & -1 & \\ & & -1 & 2,2 & -1 \\ & & & -1 & 2,2 \end{bmatrix} \begin{Bmatrix} T_0 \\ T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 120 \\ 40 \\ 40 \\ 40 \\ 440 \end{Bmatrix}$$

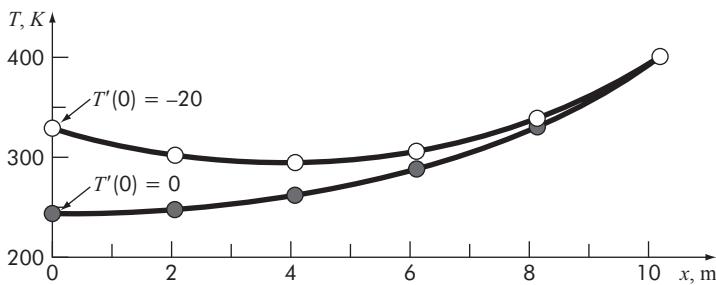


FIGURA 24.10 A solução de uma EDO de segunda ordem com uma condição de contorno na derivada em uma extremidade e uma condição de contorno fixa na outra. Dois casos são mostrados refletindo diferentes valores de derivada em $x = 0$.

que podem ser resolvidas para determinar

$$T_0 = 328,2710$$

$$T_1 = 301,0981$$

$$T_2 = 294,1448$$

$$T_3 = 306,0204$$

$$T_4 = 339,1002$$

Como na Figura 24.10, a solução em $x = 0$ agora curva-se para baixo devido à derivada negativa imposta na fronteira.

24.3.2 Abordagens por diferenças finitas para EDOs não lineares

Para EDOs não lineares, a substituição de diferenças finitas produz um sistema de equações simultâneas não lineares. Assim, a abordagem mais geral para resolução desses problemas é usar métodos de localização de raízes para sistemas de equação, como o método de Newton-Raphson, descrito na Seção 12.2.2. Embora essa abordagem seja certamente possível, uma adaptação de substituição sucessiva pode, às vezes, fornecer uma alternativa mais simples.

A haste aquecida com convecção e radiação introduzida no Exemplo 24.4 fornece um bom veículo para demonstrar essa abordagem,

$$0 = \frac{d^2 T}{dx^2} + h'(T_\infty - T) + \sigma' (T_\infty^4 - T^4)$$

Podemos converter essa equação diferencial para a forma algébrica escrevendo-a para um nó i e substituindo a Equação (24.15) para a segunda derivada:

$$0 = \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} + h'(T_\infty - T_i) + \sigma' (T_\infty^4 - T_i^4)$$

Agrupando os termos, temos

$$-T_{i-1} + (2 + h'\Delta x^2)T_i - T_{i+1} = h'\Delta x^2 T_\infty + \sigma' \Delta x^2 (T_\infty^4 - T_i^4)$$

Observe que, embora exista um termo não linear no lado direito, o lado esquerdo está expresso na forma de um sistema algébrico linear que é diagonalmente dominante. Se considerarmos que o termo não linear desconhecido na direita é igual ao seu valor da iteração anterior, a equação pode ser resolvida para

$$T_i = \frac{h' \Delta x^2 T_\infty + \sigma'^2 \Delta x^2 (T_\infty^4 - T_i^4) + T_{i-1} + T_{i+1}}{2 + h' \Delta x^2} \quad (24.19)$$

Como no método de Gauss-Seidel, podemos usar a Equação (24.19) para calcular sucessivamente a temperatura de cada nó e iterar até que o processo converja para uma tolerância aceitável. Embora essa abordagem não funcione para todos os casos, ela converge para várias EDOs deduzidas de sistemas físicos reais. Assim, por vezes ela pode ser útil para resolução de problemas rotineiramente encontrados na engenharia e na ciência.

EXEMPLO 24.7 O método das diferenças finitas para EDOs não lineares

Use a abordagem de diferenças finitas para simular a temperatura de uma haste aquecida sujeita tanto a convecção quanto a radiação:

$$0 = \frac{d^2 T}{dx^2} + h'(T_\infty - T) + \sigma'^2 (T_\infty^4 - T^4)$$

onde $\sigma' = 2,7 \times 10^{-9} \text{ K}^{-3} \text{ m}^{-2}$, $L = 10 \text{ m}$, $h' = 0,05 \text{ m}^{-2}$, $T_\infty = 200 \text{ K}$, $T(0) = 300 \text{ K}$ e $T(10) = 400 \text{ K}$. Use quatro nós interiores com um tamanho de segmento de $\Delta x = 2 \text{ m}$. Lembre-se de que resolvemos o mesmo problema com o método do tiro no Exemplo 24.4.

Solução. Por meio da Equação (24.19), podemos calcular sucessivamente as temperaturas dos nós interiores da haste. Como com a técnica Gauss-Seidel padrão, os valores iniciais dos nós interiores são zero, com os nós da fronteira definidos com as condições fixas de $T_0 = 300$ e $T_5 = 400$. Os resultados para a primeira iteração são

$$T_1 = \frac{0,05(2)^2 200 + 2,7 \times 10^{-9} (2)^2 (200^4 - 0^4) + 300 + 0}{2 + 0,05(2)^2} = 159,2432$$

$$T_2 = \frac{0,05(2)^2 200 + 2,7 \times 10^{-9} (2)^2 (200^4 - 0^4) + 159,2432 + 0}{2 + 0,05(2)^2} = 97,9674$$

$$T_3 = \frac{0,05(2)^2 200 + 2,7 \times 10^{-9} (2)^2 (200^4 - 0^4) + 97,9674 + 0}{2 + 0,05(2)^2} = 70,4461$$

$$T_4 = \frac{0,05(2)^2 200 + 2,7 \times 10^{-9} (2)^2 (200^4 - 0^4) + 70,4461 + 400}{2 + 0,05(2)^2} = 226,8704$$

O processo pode ser continuado até convergir para o resultado final:

$$T_0 = 300$$

$$T_1 = 250,4827$$

$$T_2 = 236,2962$$

$$T_3 = 245,7596$$

$$T_4 = 286,4921$$

$$T_5 = 400$$

Esses resultados estão apresentados na Figura 24.11, junto com o resultado obtido no Exemplo 24.4 com o método do tiro.

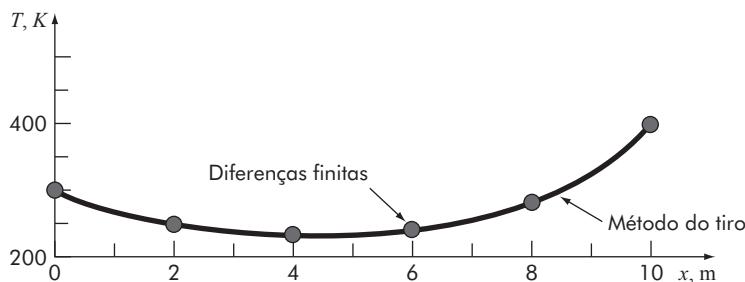


FIGURA 24.11 Os círculos cheios são o resultado da utilização do método de diferenças finitas para resolver um problema não linear. A linha gerada com o método do tiro no Exemplo 24.4 é mostrada para comparação.

►► PROBLEMAS

24.1 Um balanço de calor em estado estacionário para uma haste pode ser representado por

$$\frac{d^2T}{dx^2} - 0,15T = 0$$

Obtenha uma solução para uma haste de 10 m com $T(0) = 240$ e $T(10) = 150$ (a) analiticamente, (b) com o método do tiro, e (c) usando a abordagem por diferenças finitas com $\Delta x = 1$.

24.2 Repita o Problema 24.1, mas com a extremidade direita isolada e a temperatura da extremidade esquerda fixa em 240.

24.3 Use o método do tiro para resolver

$$7\frac{d^2y}{dx^2} - 2\frac{dy}{dx} - y + x = 0$$

com as condições de contorno $y(0) = 5$ e $y(20) = 8$.

24.4 Resolva o Problema 24.3 com a abordagem de diferenças finitas usando $\Delta x = 2$.

24.5 A equação diferencial não linear seguinte foi resolvida nos Exemplos 24.4 e 24.7.

$$0 = \frac{d^2T}{dx^2} + h'(T_\infty - T) + \sigma' (T_\infty^4 - T^4) \quad (\text{P24.5})$$

Essas equações são às vezes linearizadas para obter uma solução aproximada. Isso é feito empregando-se uma expansão em série de Taylor de primeira ordem para linearizar o termo elevado à quarta potência na equação como

$$\sigma' T^4 = \sigma' \bar{T}^4 + 4\sigma' \bar{T}^3(T - \bar{T})$$

onde \bar{T} é uma temperatura base em relação a qual o termo é linearizado. Substitua essa relação na Equação (P24.5) e então resolva a equação linear resultante com a abordagem de diferenças finitas. Empregue $\bar{T} = 300$, $\Delta x = 1$ m e os parâmetros do Exemplo 24.4 para obter sua solução. Trace os resultados junto àqueles obtidos para as versões não lineares nos Exemplos 24.4 e 24.7.

24.6 Desenvolva um arquivo-M para implementar o método do tiro para uma EDO de segunda ordem linear. Teste seu programa refazendo o Exemplo 24.2.

24.7 Desenvolva um arquivo-M para implementar a abordagem de diferenças finitas para resolução de uma EDO de segunda ordem linear com condições de contorno de Dirichlet. Teste-o refazendo o Exemplo 24.5.

24.8 Uma haste aquecida isolada com uma fonte de calor uniforme pode ser modelada com a *equação de Poisson*:

$$\frac{d^2T}{dx^2} = -f(x)$$

Dada uma fonte de calor $f(x) = 25 \text{ } ^\circ\text{C/m}^2$ e as condições de contorno $T(x=0) = 40 \text{ } ^\circ\text{C}$ e $T(x=10) = 200 \text{ } ^\circ\text{C}$, determine a distribuição de temperaturas com (a) o método do tiro e (b) o método de diferenças finitas ($\Delta x = 2$).

24.9 Repita o Problema 24.8, mas para a seguinte fonte de calor variável no espaço: $f(x) = 0,12x^3 - 2,4x^2 + 12x$.

24.10 A distribuição de temperaturas em uma aleta de refrigeração cônica (Figura P24.10) é descrita pela equação diferencial seguinte, que foi adimensionalizada:

$$\frac{d^2u}{dx^2} + \left(\frac{2}{x}\right)\left(\frac{du}{dx} - pu\right) = 0$$

onde u é a temperatura ($0 \leq u \leq 1$), x é a distância axial ($0 \leq x \leq 1$) e p é um parâmetro adimensional que descreve a transferência de calor e a geometria:

$$p = \frac{hL}{k} \sqrt{1 + \frac{4}{2m^2}}$$

onde h é o coeficiente de transferência de calor, k é a condutividade térmica, L é o comprimento ou a altura do cone e m é a inclinação da parede do cone. Essa equação tem as condições de contorno:

$$u(x=0) = 0 \quad u(x=1) = 1$$

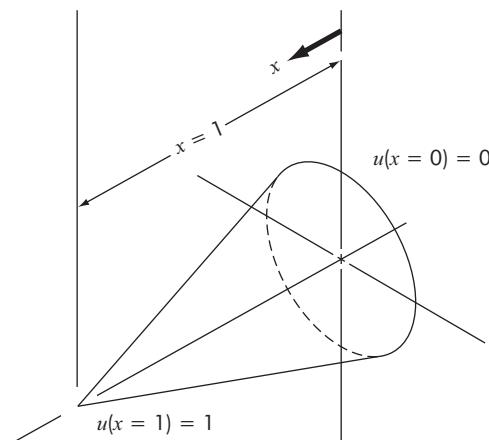


FIGURA P24.10

Resolva essa equação para a distribuição de temperaturas usando o método de diferenças finitas. Use para as derivadas diferenças finitas acuradas até segunda ordem. Escreva um programa para obter a solução e trace a temperatura versus a distância axial para os diversos valores de $p = 10, 20, 50$ e 100 .

24.11 O composto A difunde-se por meio de um tubo de 4 cm de comprimento e reage à medida que é difundido. A equação que rege a difusão com reação é

$$D \frac{d^2A}{dx^2} - kA = 0$$

Em uma extremidade do tubo ($x = 0$), existe uma grande fonte de A que resulta em uma concentração fixa de $0,1 \text{ M}$. Na outra extremidade do tubo, existe um material que absorve rapidamente qualquer A , tornando a concentração 0 M . Se $D = 1,5 \times 10^{-6} \text{ cm}^2/\text{s}$ e $k = 5 \times 10^{-6} \text{ s}^{-1}$, qual é a concentração de A como função da distância no tubo?

24.12 A equação diferencial seguinte descreve a concentração em estado estacionário de uma substância que reage com cinética de primeira ordem, em um reator de fluxo em pistão axialmente disperso (Figura P24.12):

$$D \frac{d^2c}{dx^2} - U \frac{dc}{dx} - kc = 0$$

onde D é o coeficiente de dispersão (m^2/h), c é a concentração (mol/L), x é a distância (m), U é a velocidade (m/h) e k é a taxa de reação ($/\text{h}$). As condições de contorno podem ser formuladas como

$$Uc_{\text{entrada}} = Uc(x=0) - D \frac{dc}{dx}(x=0)$$

$$\frac{dc}{dx}(x=L) = 0$$

onde c_{entrada} é a concentração na entrada do fluxo (mol/L) e L é o comprimento do reator. Essas condições são chamadas *condições de contorno de Danckwerts*.

Use a abordagem por diferenças finitas para determinar a concentração como uma função da distância dados os parâmetros seguintes: $D = 5000 \text{ m}^2/\text{h}$, $U = 100 \text{ m/h}$, $k = 2/\text{h}$, $L = 100 \text{ m}$ e $c_{\text{entrada}} = 100 \text{ mol/L}$. Utilize aproximações por diferenças finitas centradas, com $\Delta x = 10 \text{ m}$, para obter suas soluções. Compare seus resultados numéricos com a solução analítica:

$$c = \frac{Uc_{\text{entrada}}}{(U - D\lambda_1)\lambda_2 e^{\lambda_2 L} - (U - D\lambda_2)\lambda_1 e^{\lambda_1 L}} \times (\lambda_2 e^{\lambda_2 L} e^{\lambda_1 x} - \lambda_1 e^{\lambda_1 L} e^{\lambda_2 x})$$

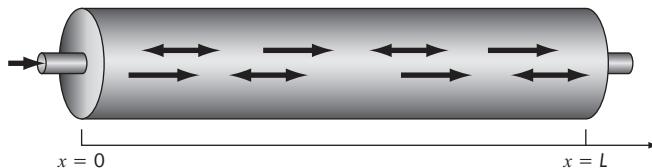
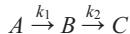


FIGURA P24.12 Reator de fluxo em pistão axialmente disperso.

onde

$$\frac{\lambda_1}{\lambda_2} = \frac{U}{2D} \left(1 \pm \sqrt{1 + \frac{4kD}{U^2}} \right)$$

24.13 Uma série reações em fase líquida, de primeira ordem, cria um produto desejável (B) e um subproduto indesejável (C):



Se as reações ocorrem em um reator de fluxo em pistão axialmente disperso (Figura P24.12), balanços de massa podem ser usados para desenvolver as seguintes EDOs de segunda ordem:

$$D \frac{d^2 c_a}{dx^2} - U \frac{dc_a}{dx} - k_1 c_a = 0$$

$$D \frac{d^2 c_b}{dx^2} - U \frac{dc_b}{dx} + k_1 c_a - k_2 c_b = 0$$

$$D \frac{d^2 c_c}{dx^2} - U \frac{dc_c}{dx} + k_2 c_b = 0$$

Use a abordagem de diferenças finitas para calcular a concentração de cada reagente como uma função

da distância, dados: $D = 0,1 \text{ m}^2/\text{min}$, $U = 1 \text{ m}/\text{min}$, $k_1 = 3/\text{min}$, $k_2 = 1/\text{min}$, $L = 0,5 \text{ m}$, $c_{a, \text{entrada}} = 10 \text{ mol/L}$. Use aproximações por diferenças finitas centradas com $\Delta x = 0,05$ para obter suas soluções e considere condições de contorno de Danckwerts, como descrito no Problema 24.12. Também, calcule a soma dos reagentes como uma função da distância. Seus resultados fazem sentido?

24.14 Um biofilme com uma espessura L_f (cm) forma-se na superfície de um sólido (Figura P24.14). Depois de atravessar uma camada de difusão de espessura L (cm), um composto químico A difunde-se no biofilme, onde é sujeito a uma reação de primeira ordem irreversível que o converte em um produto B .

Um balanço de massa em estado estacionário pode ser usado para deduzir as seguintes equações diferenciais ordinárias para o composto A :

$$D \frac{d^2 c_a}{dx^2} = 0 \quad 0 \leq x < L$$

$$D_f \frac{d^2 c_a}{dx^2} - kc_a = 0 \quad L \leq x < L + L_f$$

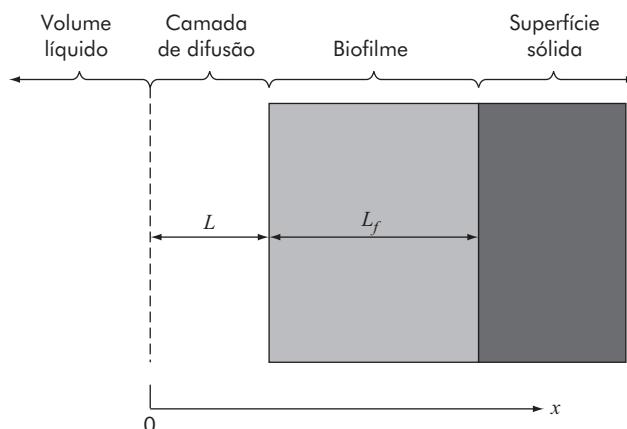


FIGURA P24.14 Biofilme crescendo em uma superfície sólida.

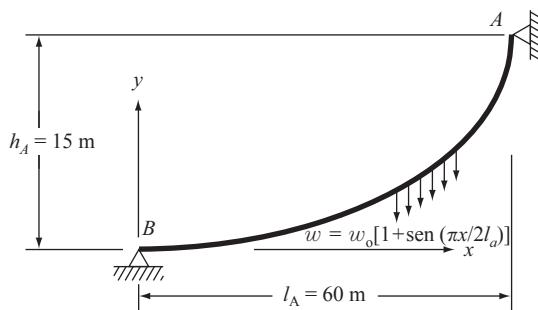


FIGURA P24.15

onde D é o coeficiente de difusão na camada de difusão $= 0,8 \text{ cm}^2/\text{d}$, D_f é o coeficiente de difusão no biofilme $= 0,64 \text{ cm}^2/\text{d}$ e k é a taxa de primeira ordem para a conversão de A para $B = 0,1/\text{d}$. As seguintes condições de contorno são válidas:

$$c_a = c_{a0} \quad \text{para } x = 0$$

$$\frac{dc_a}{dx} = 0 \quad \text{para } x = L + L_f$$

onde c_{a0} é a concentração de A no volume líquido $= 100 \text{ mol/L}$. Use o método de diferenças finitas para calcular a distribuição em estado estacionário de A de $x = 0$ a $L + L_f$, onde $L = 0,008 \text{ cm}$ e $L_f = 0,004 \text{ cm}$. Use diferenças finitas centradas com $x = 0,001 \text{ cm}$.

24.15 Um cabo está pendurado por dois suportes em A e B (Figura P24.15), e está carregado com uma distribuição de carga cujo módulo varia com x como

$$w = w_0 \left[1 + \operatorname{sen}\left(\frac{\pi x}{2l_A}\right) \right]$$

onde $w_0 = 450 \text{ N/m}$. A inclinação do cabo é $(dy/dx) = 0$ em $x = 0$, que é o seu ponto mais baixo. Ele também é o ponto no qual a tensão no cabo está em um mínimo de T_0 . A equação diferencial que governa o cabo é

$$\frac{d^2y}{dx^2} = \frac{w_0}{T_0} \left[1 + \operatorname{sen}\left(\frac{\pi x}{2l_A}\right) \right]$$

Resolva essa equação usando um método numérico e trace a forma do cabo (y em função de x). Para a solução numérica, o valor de T_0 é desconhecido, de modo que a solução deve usar uma técnica iterativa, análoga ao método do tiro, para convergir para um valor correto de h_A para diversos valores de T_0 .

24.16 A equação diferencial básica da curva elástica para uma viga simplesmente apoiada e uniformemente carregada (Figura P24.16) é dada por

$$EI \frac{d^2y}{dx^2} = \frac{wLx}{2} - \frac{wx^2}{2}$$

onde E é o módulo de elasticidade e I é o momento de inércia. As condições de contorno são $y(0) = y(L) = 0$. Determine a deflexão da viga usando (a) a abordagem por diferenças finitas ($\Delta x = 0,6 \text{ m}$) e (b) o método do tiro. Os seguintes valores dos parâmetros se aplicam: $E = 200 \text{ GPa}$, $I = 30.000 \text{ cm}^4$, $w = 15 \text{ kN/m}$ e $L = 3 \text{ m}$. Compare seus resultados numéricos com a solução analítica:

$$y = \frac{wLx^3}{12EI} - \frac{wx^4}{24EI} - \frac{wL^3x}{24EI}$$

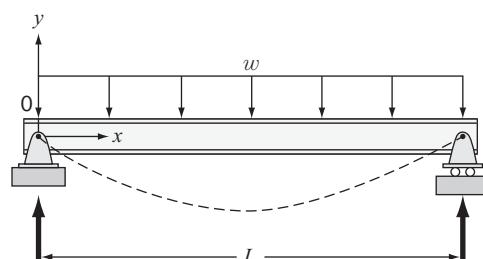


FIGURA P24.16

24.17 No Problema 24.16, a equação diferencial básica da curva elástica para uma viga uniformemente carregada foi formulada como

$$EI \frac{d^2y}{dx^2} = \frac{wLx}{2} - \frac{wx^2}{2}$$

Observe que o lado direito representa o momento como uma função de x . Uma abordagem equivalente pode ser formulada em termos da quarta derivada da deflexão como

$$EI \frac{d^4y}{dx^4} = -w$$

Para essa formulação, são necessárias quatro condições de contorno. Para os suportes mostrados na Figura P24.16, as condições são que os deslocamentos das extremidades são nulos, $y(0) = y(L) = 0$, e que os momentos nas extremidades são nulos, $y''(0) = y''(L) = 0$. Determine a deflexão da viga usando a abordagem por diferenças finitas ($\Delta x = 0,6$ m). Os seguintes valores dos parâmetros se aplicam: $E = 200$ GPa, $I = 30.000$ cm 4 , $w = 15$ kN/m e $L = 3$ m. Compare seus resultados numéricos com a solução analítica dada no Problema 24.16.

24.18 Sob uma série de hipóteses simplificadoras, a altura do lençol freático em um aquífero não confinado unidimensional (Figura P24.18) pode ser modelada com a seguinte EDO de segunda ordem:

$$Kh \frac{d^2h}{dx^2} + N = 0$$

onde x é a distância (m), K é a condutividade hidráulica (m/d), h é a altura do lençol freático (m), \bar{h} é a altura média do lençol freático (m) e N é a taxa de infiltração (m/d).

Determine a altura do lençol freático para $x = 0$ a 1000 m, onde $h(0) = 10$ m e $h(1000) = 5$ m. Use os seguintes parâmetros para o cálculo: $K = 1$ m/d e

$N = 0,0001$ m/d. Defina a altura média do lençol freático como a média das condições de contorno. Obtenha sua solução com (a) o método do tiro e (b) o método de diferenças finitas ($\Delta x = 100$ m).

24.19 No Problema 24.18, um modelo de águas subterrâneas linearizado foi usado para simular a altura do lençol freático para um aquífero não confinado. Um resultado mais realista pode ser obtido por meio da seguinte EDO não linear:

$$\frac{d}{dx} \left(Kh \frac{dh}{dx} \right) + N = 0$$

onde x é a distância (m), K é a condutividade hidráulica (m/d), h é a altura do lençol freático (m) e N é a taxa de infiltração (m/d). Calcule a altura do lençol freático para o mesmo caso como no Problema 24.18. Isto é, resolva de $x = 0$ a 1000 m com $h(0) = 10$ m, $h(1000) = 5$ m, $K = 1$ m/d e $N = 0,0001$ m/d. Obtenha sua solução com (a) o método do tiro e (b) o método de diferenças finitas ($\Delta x = 100$ m).

24.20 Assim como a lei de Fourier e o balanço de calor podem ser empregados para caracterizar distribuições de temperaturas, relações análogas estão disponíveis para modelar problemas de campo em outras áreas da engenharia. Por exemplo, a engenharia elétrica usa uma abordagem similar na modelagem de campos eletrostáticos. Sob uma série de hipóteses simplificadoras, um análogo da lei de Fourier pode ser representado na forma unidimensional como

$$D = -\varepsilon \frac{dV}{dx}$$

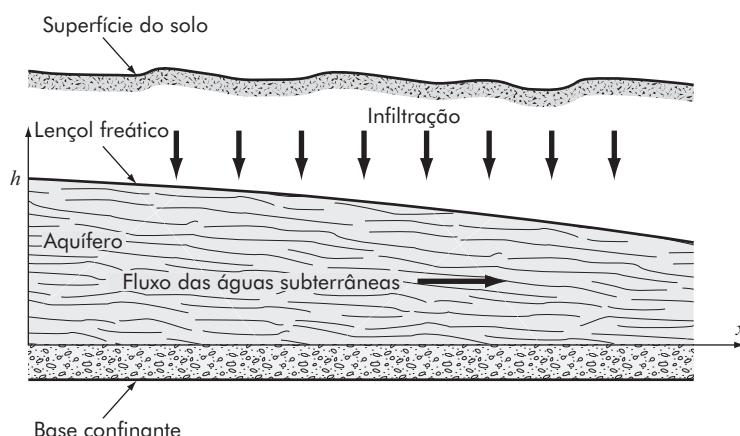


FIGURA P24.18 Aquífero não confinado ou "freático".

onde D é chamado de densidade de fluxo elétrico, ϵ é a permissividade do material e V é o potencial eletrostático. De modo similar, a equação de Poisson (ver Problema 24.8) para campos eletrostáticos pode ser representada em uma dimensão como

$$\frac{d^2 V}{dx^2} = -\frac{\rho_v}{\epsilon}$$

onde ρ_v é a densidade de cargas. Use a técnica de diferenças finitas, com $\Delta x = 2$, para determinar V para um fio em que $V(0) = 1.000$, $V(20) = 0$, $\epsilon = 2$, $L = 20$ e $\rho_v = 30$.

24.21 Considere que a posição de um corpo em queda é governada pela seguinte equação diferencial:

$$\frac{d^2 x}{dt^2} + \frac{c}{m} \frac{dx}{dt} - g = 0$$

onde c é um coeficiente de arraste de primeira ordem = 12,5 kg/s, m é a massa = 70 kg e g é a aceleração da gravidade = 9,81 m/s². Use o método do tiro para resolver essa equação para as condições de contorno:

$$\begin{aligned} x(0) &= 0 \\ x(12) &= 500 \end{aligned}$$

24.22 Como na Figura P24.22, uma haste metálica isolada tem uma condição de contorno de temperatura fixa (T_0) em sua extremidade esquerda. Em sua extremidade direita, ela está conectada a um tubo de paredes finas cheio de água através do qual calor é conduzido. O tudo é isolado em sua extremidade direita e troca calor por convecção

com o ar circundante de temperatura fixa T_∞ . O fluxo de calor convectivo em um ponto x ao longo do tubo (W/m²) é representado por

$$J_{\text{conv}} = h(T_\infty - T_2(x))$$

onde h é o coeficiente de transferência de calor por convecção [W/(m² · K)]. Use o método de diferenças finitas, com $\Delta x = 0,1$, para calcular a distribuição de temperaturas para o caso em que tanto a haste quanto o tubo são cilíndricos com o mesmo raio r (m). Use os seguintes parâmetros para sua análise: $L_{\text{haste}} = 0,6$ m, $L_{\text{tubo}} = 0,8$ m, $T_0 = 400$ K, $T_\infty = 300$ K, $r = 3$ cm, $\rho_1 = 7870$ kg/m³, $C_{p1} = 447$ J/(kg · K), $k_1 = 80,2$ W/(m · K), $\rho_2 = 1000$ kg/m³, $C_{p2} = 4,18$ kJ/(kg · K), $k_2 = 0,615$ W/(m · K) e $h = 3000$ W/(m² · K). Os subscritos indicam a haste (1) e o tubo (2).

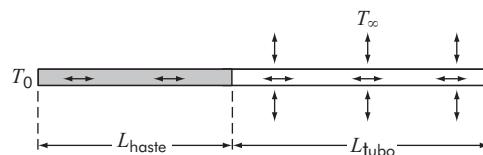


FIGURA P24.22

24.23 Faça os mesmos cálculos como no Problema 24.22, mas para o caso em que o tubo também está isolado (isto é, não há convecção) e a parede do lado direito é mantida em uma temperatura fixa de 200 K.

Referências

- Anscombe, F. J., "Graphs in Statistical Analysis," *Am. Stat.*, 27(1):17-21, 1973.
- Attaway, S., *MATLAB: A Practical Introduction to Programming and Problem Solving*, Elsevier Science, Burlington, MA, 2009.
- Bogacki, P. and L. F. Shampine, "A 3(2) Pair of Runge-Kutta Formulas," *Appl. Math. Letters*, 2(1989):1-9, 1989.
- Brent, R. P., *Algorithms for Minimization Without Derivatives*, Prentice Hall, Englewood Cliffs, NJ, 1973.
- Butcher, J. C., "On Runge-Kutta Processes of Higher Order," *J. Austral. Math. Soc.*, 4:179, 1964.
- Carnahan, B., H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods*, Wiley, New York, 1969.
- Chapra, S. C. and R. P. Canale, *Numerical Methods for Engineers*, 6th ed., McGraw-Hill, New York, 2010.
- Cooley, J. W. and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Math. Comput.*, 19:297-301, 1965.
- Dekker, T. J., "Finding a Zero by Means of Successive Linear Interpolation." In B. Dejon and P. Henrici (editors), *Constructive Aspects of the Fundamental Theorem of Algebra*, Wiley-Interscience, New York, 1969, pp. 37-48.
- Dormand, J. R. and P. J. Prince, "A Family of Embedded Runge-Kutta Formulae," *J. Comp. Appl. Math.*, 6:19-26, 1980.
- Draper, N. R. and H. Smith, *Applied Regression Analysis*, 2d ed., Wiley, New York, 1981.
- Faddeev, D. K. and V. N. Faddeeva, *Computational Methods of Linear Algebra*, Freeman, San Francisco, 1963.
- Forsythe, G. E., M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computation*, Prentice Hall, Englewood Cliffs, NJ, 1977.
- Gabel, R. A. and R. A. Roberts, *Signals and Linear Systems*, Wiley, New York, 1987.
- Gander, W. and W. Gautschi, Adaptive Quadrature-Revisited, *BIT Num. Math.*, 40:84-101, 2000.
- Gerald, C. F. and P. O. Wheatley, *Applied Numerical Analysis*, 3d ed., Addison-Wesley, Reading, MA, 1989.
- Hanselman, D. and B. Littlefield, *Mastering MATLAB 7*, Prentice Hall, Upper Saddle River, NJ, 2005.
- Hayt, W. H. and J. E. Kemmerly, *Engineering Circuit Analysis*, McGraw-Hill, New York, 1986.
- Heideman, M. T., D. H. Johnson, and C. S. Burrus, "Gauss and the History of the Fast Fourier Transform," *IEEE ASSP Mag.*, 1(4):14-21, 1984.
- Hornbeck, R. W., *Numerical Methods*, Quantum, New York, 1975.

- James, M. L., G. M. Smith, and J. C. Wolford, *Applied Numerical Methods for Digital Computations with FORTRAN and CSMP*, 3d ed., Harper & Row, New York, 1985.
- Moler, C. B., *Numerical Computing with MATLAB*, SIAM, Philadelphia, 2004.
- Moore, H., *MATLAB for Engineers*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2008.
- Ortega, J. M., *Numerical Analysis-A Second Course*, Academic Press, New York, 1972.
- Palm, W. J. III, *A Concise Introduction to MATLAB*, McGraw-Hill, New York, 2007.
- Ralston, A., "Runge-Kutta Methods with Minimum Error Bounds," *Match. Comp.*, 16:431, 1962.
- Ralston, A. and P. Rabinowitz, *A First Course in Numerical Analysis*, 2d ed., McGraw-Hill, New York, 1978.
- Ramirez, R. W., *The FFT, Fundamentals and Concepts*, Prentice Hall, Englewood Cliffs, NJ, 1985.
- Recktenwald, G., *Numerical Methods with MATLAB*, Prentice Hall, Englewood Cliffs, NJ, 2000.
- Scarborough, I. B., *Numerical Mathematical Analysis*, 6th ed., Johns Hopkins Press, Baltimore, MD, 1966.
- Shampine, L. F., *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, 1994.
- Van Valkenburg, M. E., *Network Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1974.
- White, F. M., *Fluid Mechanics*. McGraw-Hill, New York, 1999.

APÊNDICE A

Funções nativas do MATLAB

abs, 34-35
acos, 34-35
ascii, 56-57
axis, 45-46
axis square, 39-40
beep, 68-69
besselj, 427
ceil, 35-36
chol, 265, 267
clabel, 196, 539
clear, 55-56
cond, 276-278
contour, 196, 539
conv, 173
cumtrapz, 483-484
dblquad, 487-488
deconv, 172-173
det, 233-234
diag, 315
diff, 533-535
disp, 52-53
double, 56-57
eig, 312-313
elfun, 34-35
eps, 101
erf, 517-518
error, 58
event, 595-596
exp, 34-35
eye, 218
factorial, 45-46, 65n2
fft, 396
fix, 433-434
floor, 35-36, 196
fminbnd, 194
fminsearch, 197-199
format bank, 27
format compact, 25n1
format long, 27, 102
format long e, 27, 300-301
format long eng, 27
format long g, 27
format loose, 25n1
format short, 27
format short e, 27, 300-301
format short eng, 27
format short g, 278-279
fplot, 75
fprintf, 54
fzero, 168-170, 175-176
getframe, 69-70
gradient, 536-537
grid, 37-38
help, 50-51, 56-57
help elfun, 34-35
hist, 331-332
hold off, 38-39
hold on, 38-39
humps, 85-86, 512-513
inline, 75
input, 52-53
interp1, 445-446
interp2, 450
interp3, 450
inv, 218-220
isempty, 94-95
legend, 374-375, 484-485
length, 36-37
LineWidth, 38-39
linspace, 77
load, 55-57
log, 34-35
log10, 348-349
log2, 138n2
loglog, 45-46
logspace, 30-31
lookfor, 41, 49-50
lu, 262
MarkerEdgeColor, 38-39
MarkerFaceColor, 38-39
MarkerSize, 38-39
max, 35-36, 243-244
mean, 51-52, 330-331
median, 330-331
mesh, 68-69
meshgrid, 196, 539
min, 35-36, 330-331
mode, 330-331
movie, 69-71
nargin, 63-64
norm, 276-277
ode113, 591-592
ode15s, 605-606
ode23, 589-590
ode23s, 605-606
ode23t, 605-606
ode23tb, 605-606
ode45, 591-592, 606-607
odeset, 593-594
ones, 28-29
optimset, 168-170, 175-176
pause, 68-69
pchip, 443-444, 446-447
peaks, 544-545
pi, 27
plot, 37-38
plot3, 39-40, 582-583

poly, 171-173
polyfit, 350-351, 408, 420-
 421
polyval, 350-351, 408, 420-
 421
prod, 35-36
quad, 512-513
quadl, 512-513
quiver, 538
rand, 331-334
randn, 331-335
realmax, 101
realmin, 101
roots, 171-174
round, 35-36

save, 55-56
semilogy, 44-46
set, 453-454
sign, 61-62
sin, 34-35
size, 219
sort, 35-36
spline, 443-444
sqrt, 34-35
sqrtm, 35-36
std, 330-331
stem, 399-400
subplot, 39-40
sum, 35-36, 265
surf, 196

tanh, 6-7, 34-35
tic, 69-70
title, 37-38
toc, 69-70
trapz, 483-484, 491-492
triplequad, 487-488
var, 330-331
varargin, 78
who, 28-29
whos, 28-29
 xlabel, 37-38
 ylabel, 37-38
 ylim, 399-400
zeros, 28-29
 zlabel, 196

APÊNDICE B

Funções desenvolvidas no MATLAB

Nome do arquivo-M	Descrição	Página
bissec	Localização de raízes com bissecção	139-140
eulode	Integração de uma equação diferencial única com o método de Euler	560-561
fzerosimp	Método de Brent para localização de raízes	167
GaussIngenua	Resolução de sistemas lineares com eliminação de Gauss sem pivotamento	239
GaussPivot	Resolução de sistemas lineares com eliminação de Gauss com pivotamento parcial	244-245
GaussSeidel	Resolução de sistemas lineares com o método de Gauss-Seidel	288-289
aureamin	Mínimo de uma função unidimensional com busca da seção áurea	192
busca_inc	Localização de raízes com uma busca incremental	132
IterMeth	Algoritmo geral para cálculos iterativos	94-95
Lagrange	Interpolação com o polinômio de Lagrange	417-418
regrlin	Ajuste de uma reta com regressão linear	349-350
natspline	Spline cúbico com condições de extremidade naturais	452-453
Newtint	Interpolação com o polinômio de Newton	415
newtmult	Localização de raízes para sistemas de equações não lineares	297
newtraph	Localização de raízes com o método de Newton-Raphson	161
quadadapt	Quadratura adaptativa	511-512
rk4sys	Integração de sistemas de EDOs com o método RK de quarta ordem	576-577
romberg	Integração de uma função com integração de Romberg	502-503
TabelaConsulta	Tabela de busca com interpolação linear	432-433
TabelaConsultaBin	Tabela de busca com busca binária seguida de interpolação linear	433-434
trap	Integração de uma função com a aplicação múltipla da regra do trapézio	473-474
trapuneq	Integração de dados desigualmente espaçados com a regra do trapézio	482-483
Tridiag	Resolução de sistemas lineares tridiagonais	247

Índice

- ~, 60-61
&, 60-61
|, 60-61
', 27-28
* , 31-32
+ , 31-32
^ , 31-32
/ , 31-32
\, 31-32, 219-220. *Ver também* Operador barra invertida
\n, 54-55
\t, 54-55
<, 59
<=, 59
>, 59
>=, 59
-, 31-32
=, 59
==, 59
%d, 54-55
- A**
Abordagem preditor-corretor, 563-564
Acurácia, 88-90
Afundamento de oxigênio, 202-203
Água da chuva, 144-147
Ajustando dados experimentais, 373-376
Ajuste de curvas, 321-457
 análise de Fourier. *Ver* Análise de Fourier
 interpolação polinomial. *Ver* Interpolação polinomial
 mínimos quadrados linear geral, 367-370
 organização da parte, 322-323
- Ajuste de curvas aos dados. *Ver* Ajuste de curvas
Alcalinidade, 150
Algarismos significativos, 27
Algoritmo de Cooley-Tukey, 396
Amplitude, 326-327
Análise de circuito, 222-225
Análise de Fourier, 380-403
 domínios do tempo e da frequência, 390-393
 FFT, 394-396
 fft, 396-399
 funções senoidais, 380-388
 integral e transformada de Fourier, 391-394
 potência espectral, 398-400
 série de Fourier contínua, 387-389
 TDF, 394-399
Análise de sensibilidade, 78
Análise de tendência, 321-322
Análise exploratória de dados, 41-44
AND, 59
Ângulo de fase, 382-383
 de atraso, 382-383
 de avanço, 382-383
Animação, 70-71
- bissecc, 130-140
busca_inc, 132
eulode, 560-561
fzerosimp, 167
GaussIngenua, 239
GaussPivot, 243-245

- método da potência, 310-313
método polinomial, 306-307
Autovetor, 306-307
Avaliação do condicionamento de matrizes, 275-277
axis square, 39-40
- B**
- Baixo Rio Colorado, 281-283
Balanço de calor, 127-128
energia, 127-128
força, 127-128
massa, 127-128
bank, 27
beep, 68-69
besselj, 425-426
Bin, 329-330
bissec, 139-141
Bissecção, 134-143
Bit, 95-96
Borrão, 103-104
Brent, Richard, 163, 194
Busca binária, 432-434
da seção áurea, 187-192, 194-195
incremental, 131-134
sequencial, 432-433
busca_inc, 132
- C**
- Cabo catenário, 178-179
Cálculo, 521-522
dependente do tempo, 12
do determinante, 244-246
do trabalho, 488-492
em regime estacionário, 11-12
transiente, 11-12
Cálculos de respostas a estímulos, 269-272
Cálculos grandes, 102
Cancelamento na subtração, 102, 243-244
Caos, 581-582
Capacidade de carga, 584
Case sensitive, 25-26
ceil, 35-36
chol, 265
Cinética enzimática, 350-356
clear, 55-56
Coeficiente de arraste, 404
de arraste concentrado, 5-7, 16-17
de correlação, 342
- de determinação, 342-343
de distribuição, 252-253
de restituição, 87
de variação, 326-327
do polinômio, 406-407
Comandos de formatação, 27
Command window, 24-25
Complemento de 2, 97-98
Concatenação, 28-31
de íons de hidrogênio, 146-147
cond, 276-277
Condição de continuidade, 435-436
contorno de Dirichlet, 624, 629-630
contorno de Neumann, 624, 630-631
extremidade amarrada, 431-432
extremidade natural, 442-443
Condição sem um nó, 442-443
Condisionalmente estável, 559-560
Condições de contorno de Danckwerts, 636-637
contorno na derivada, 624-626, 629-633
extremidade, 442-443
Condições lógicas, 59-61
Conservação de carga, 13, 222
energia, 13, 223-224
massa, 13
momento, 13
Constante de Henry, 144-145
integração, 548-550
meia saturação, 350-351
Planck, 98-99
Contagem de operações, 239-242
Controle adaptativo do tamanho do passo, 588-589
Convergência, 154-157, 159, 175-176, 286-288
linear, 153-154
quadrática, 157
Cooley, J.W., 396
Critério de parada, 91-92
Ctrl+Break, 69-70
Ctrl+c, 69-70
cumtrapz, 483-484
Curva da borboleta, 47
Curvatura, 522-523
- D**
- Dados com erro, 531-533
Dados com incertezas, 120-121
dblquad, 487-488
- Decisões, 56-64
Decomposição de Cholesky, 263-266
LU. Ver Decomposição LU
por valor singular, 370-371
QR, 266n, 370-371
terminologia, 254n
Decomposição LU, 254n, 254-263
com pivotamento, 260-262
eliminação de Gauss, 256-263
MATLAB, 262-263
pivotamento parcial, 260-262
vantagens de, 254-255
visão geral, 254-256
Deflação, 312-313
Dekker, Theodorus, 163
Densidade do ar, 404
Derivação, 459, 461, 520-545
amplificação de erros, 532-533
dados com erro, 531-533
dados desigualmente espaçados, 530-532
diff, 533-535
extrapolação de Richardson, 528-531
fórmulas de derivação de alta acurácia, 524-528
gradiente, 536-537
numérica, 110-115
Derivada(s), 459, 521-522
parcial mista, 533-534
parciais, 532-534
Desvio-padrão, 326-327
det, 233-234
Determinante, 231-233
do Jacobiano, 294-296
DFT, 394-399
Diagonal principal (matriz), 211-212
Diferença centrada, 111-112, 529
finita, 110-111
progressiva, 110-111, 113, 527
regressiva, 111-112, 528
diff, 533-535
disp, 52-53
Distância *versus* tempo, 484-485
Distribuição normal, 329-330, 540-541
Divergência, 155-156
Divisão à esquerda, 30-31, 210-221, 229-230, 266, 370
à esquerda do MATLAB, 266
de matrizes, 215
do passo na metade, 588-589

- Dominância da diagonal, 286-287
 Domínio
 da frequência, 390
 do tempo, 390
- E**
- Edit window, 24-25
 EDO. *Ver* Equação diferencial ordinária (EDO)
 Efeito borboleta, 581-582
 Efeito telescópio, 418-419
 eig, 312-314
 Elemento pivô, 237
 Eletroneutralidade, 150
 Eliminação de Gauss. *Ver* Eliminação de Gauss ingênua
 Eliminação de Gauss ingênua, 234-242, 254-255
 arquivo-M, 239
 cálculo de determinante, 244-246
 contagem de operações, 239-242
 decomposição LU, 256-263
 eliminação progressiva, 236-237
 pivotamento parcial, 242-245
 substituição reversa, 237
 visão geral/fases, 236
 end, 52n
 Energia potencial, 197-198
 Enganos, 119-120
 Engenharia de recursos hídricos, 357-359
 Entrada-saída, 52-57
 Enzima, 351
 alostérica, 252-253
 Epilímnio, 451
 eps, 101
 Epsilon da Máquina, 99-100
 Equação
 da taxa de crescimento da saturação, 345-346
 de Arrhenius, 45-46
 de Boussinesq, 425-426
 de Colebrook, 173, 180-181
 de estado de Redlich-Kwong, 177-178
 de Laplace, 301-302
 de Manning, 84-85, 359-360
 de Michaelis-Menten, 350-351
 de pivô, 237
 de Poisson, 635-636, 639-640
 de potência simples, 344-346
 de primeira ordem, 547-548
 de segunda ordem, 547-548
 de Swamee-Jain, 173-174
 de taxa, 547
 de van der Pol, 584, 605-606
 de von Karman, 149
 diferencial, 6-7, 547
 diferencial parcial (EDP), 547-548
 do corretor, 562-563
 do predador-presa, 591-592
 do preditor, 562-563
 exponencial, 345-346
 normal, 338-339
 Equações algébricas lineares, 207-229
 eliminação de Gauss, 234-242
 forma geral, 205
 forma matricial, 219-220
 Gauss-Seidel, 284-291
 MATLAB, 219-222, 229-230
 problemas de variáveis concentradas, 206-207
 problemas de variáveis distribuídas, 207
 visão geral, 207-208
 Equações de Lotka-Volterra, 578-579, 614-615
 Equações diferenciais de ordem superior, 547-548
 Equações diferenciais ordinárias (EDO), 547-643
 definição, 547-548
 métodos adaptativos e sistemas rígidos. *Ver* Métodos adaptativos e sistemas rígidos
 organização da parte, 551-552
 problemas de valor de contorno. *Ver* Problemas de valor de contorno
 problemas de valor inicial. *Ver* Problemas de valor inicial
 rigidez, 601-608
 visão geral, 547-551
 Equações não lineares simultâneas, 291-298
 Equilíbrio e energia potencial mínima, 197-199
 erf, 517-518
 Erro, 88-96
 absoluto, 90
 balanço, 114-116
 de arredondamento, 95-104, 557-558
 de overflow, 98-99
 de truncamento, 103-115, 557-558
 global, 558
 local, 185-186
 propagado, 557-558
 derivação, 531-533
 enganos, 119-120
 formulação, 119-121
 função do MATLAB, 58
 incerteza nos dados, 120-121
 método de Euler, 557-560
 método de Heun com início não automático, 600-602
 numérico total, 114-120
 regra do trapézio, 468-469
 regressão linear, 340-344
 relativo fracionário verdadeiro, 90-91
 Erro-padrão da estimativa, 341-342
 error, 58
 Erros de modelo, 119-121
 Escalares, 25-27
 Especificadores (cores, símbolos, tipos de linha), 38-39
 Espectro de potência, 398-400
 Estatística, 325-332
 Estatística descritiva, 325-327, 330-331
 Estilos de marcadores, 38-39
 Estratificação térmica, 451
 Estudos de caso
 ajustando dados experimentais, 373-376
 análise de circuitos, 222-225
 análise exploratória de dados, 41-44
 atrito da tubulação, 173-177
 cinética enzimática, 350-356
 equilíbrio e energia potencial mínima, 197-199
 fonte intermitente de Plínio, 608-612
 força de arraste, 16-19
 gases de efeito estufa e água da chuva, 144-147
 manchas solares, 400-402
 modelo de uma haste aquecida, 247-250
 modelos predador-presa, 578-583
 poluição do ar de interiores, 277-280
 reações químicas, 298-301
 terremotos, 314-316
 trabalho, cálculo de, 488-492
 transferência de calor, 451-455
 valor médio quadrático da corrente, 513-517
 velocidade do saltador de *bungee jumping*, 78-82
 visualização de campos, 547-540
 Euclides, 187-188
 eulode, 560-561
 Eventos, 594-597
 events, 595-596
 Expansão em série de McLaurin, 45-46, 91-92, 402-403
 Explicita, 127-128

- Extrapolação, 419-422
 Extrapolação de Richardson, 496-499, 528-531
 eye, 218
- F**
 factorial, 65n
 Falsa posição, 140-143
 Fasor, 389
 Fator
 amplificador, 559-560
 de atrito, 173
 de atrito de Fanning, 149
 Fatorização de Cholesky, 263-266
 FFT, 394-396
 fft, 396-399
 floor, 35-36
 Flops, 239-241
 Fluxo de calor, 451
 fmimbind, 194-195
 fmimsearch, 197-198, 371-372
 Fonte intermitente, 609-610
 de Plínio, 608-612
 for...end, 65-68
 Força de arraste, 5-7, 324-325, 332-335
 Forma de Frobenius, 274-275
 format compact, 24-25n
 format long, 27, 102
 format loose, 24-25n
 format short, 27
 Formato de dupla precisão do IEEE, 100
 Fórmula da falsa posição, 140-141
 Fórmula(s) de
 derivação de alta acurácia, 524-528
 Euler, 389
 Gauss-Legendre, 506-509
 de dois pontos, 505-508
 de três pontos, 507-508
 integração abertas, 468, 485-486
 de Newton-Cotes, 468, 485-486
 integração compostas, 470-471
 integração fechadas, 468, 480-481
 de Newton-Cotes, 468, 480-481
 interpolação linear de Newton, 408
 Newton-Cotes, 465-468, 480-482, 485-486
 Newton-Raphson, 156
 Fórmulas de integração numérica, 462-495
 cálculo da distância a partir da velocidade, 483-485
 de Newton-Cotes, 465-468, 480-482, 485-486
 de ordem superior, 480-482
- integrais múltiplas, 485-488
 métodos abertos, 485-486
 métodos fechados, 468-481
 regra do trapézio, 468-475
 regras de Simpson, 474-481
 segmentos desiguais, 481-485
 temperatura média, 486-487
- Fourier, Joseph, 380
 fplot, 75
 fprintf, 54-55
- Frequência, 308-309, 381-383
 angular, 308-309, 382-383
 de amostragem, 396-397
 de Nyquist, 394-397
 de ressonância, 315
 fundamental, 387-388
 natural, 315
- Função, 49-52. *Ver também* Nomes das funções individuais
 anônima, 74-75
 de Bessel, 425-426, 455-456
 de Runge, 421-422, 443-445
 forçante, 5
 função, 75-78
 incremento, 554-555, 567-568
 integração numérica. *Ver* Integração numérica de funções
 passada, 75
 periódica, 380-381
 por partes, 85-86
 principal, 52-53
 sinal, 553-554
 spline, 428
- Funções
 nativas, 34-37, 643-645. *Ver também*
 Função
 senoidais, 380-388
- fzero, 168-170
 fzerosimp, 167
- G**
 Gases de efeito estufa e água da chuva, 144-147
 GaussIngenua, 239
 GaussPivot, 243-245
 GaussSeidel, 287-289
 Gerenciamento de arquivos, 55-57
 getframe, 69-70
 gradient, 536-537
 Gradiente, 523-524, 538, 620-621
 Gráfico de plano de fase, 579-580, 592-593
 de Euler, 580-581
 RK4, 580-581
- Gráfico no tempo de Euler, 580-581
 Gráfico no tempo RK4, 580-581
 Gráficos, 37-41
 Grandes Lagos, 317-319
 Graphics window, 24-25
 Graus de liberdade, 326-327
- H**
 Harmônicos, 387-388
 Haste aquecida, 247-250, 619-621
 Hélice, 39-41
 Hélice cônica, 45-46
 help, 34-35, 41
 Hertz (Hz), 308-309, 382-383
 Hipolímnio, 451
 hist, 330-331
 Histograma, 329-332
 hold off, 38-39
 hold on, 38-39
 Homogêneas, 304-305
 humps, 455, 513
- I**
 i, 27
 if, 56-58
 if de única linha, 58
 if...else, 60-61
 if...elseif, 60-62
 Implícito, 127-128, 603-604
 Imprecisão, 88-90
 Impressão, 25-26
 Inacurácia, 88-90
 Incerteza, 88-89
 nos dados, 120-121
- Incondicionalmente estável, 603-604
 Indentação, 73-74
 inf, 101
 Informação alfanumérica, 30-31
 inline, 75
 input, 52-53
 Integração de Romberg, 498-503
 Integração e derivação, 459-545
 definições, 459, 462-463
 derivação. *Ver* Derivação numérica
 fórmulas de integração numérica. *Ver*
 Fórmulas de integração numérica
 integração, 462-466
 integração numérica de funções. *Ver*
 Integração numérica de funções
 organização da parte, 459-461
 segmentos desiguais, 481-485
- Integração fechada, 461

- Integração numérica. *Ver* Integração e derivação
- Integração numérica de funções, 496-520
extrapolação de Richardson, 496-499
fórmulas de Gauss-Legendre, 505-509
de dois pontos, 505-508
de três pontos, 507-508
integração de Romberg, 498-503
métodos dos coeficientes indeterminados, 503-506
quadratura adaptativa, 509-513
quadratura de Gauss, 502-510
- Integrais múltiplas, 485-488
- Integral
de área, 465-466
de Fourier de $f(t)$, 393
de Fourier, 392-393
de volume, 465-466
definida, 548-550
dupla, 485-487
dupla numérica, 485-487
indefinida, 548-550
- interp1, 445-448
interp2, 450
interp3, 450
- Interpolação. *Ver* Interpolação polinomial
bidimensional, 448-450
bilinear, 448-450
cúbica, 414
curvilínea, 321-322
Hermitiana cúbica por partes, 446-448
inversa, 418-420
linear, 321-322, 405, 408-410
multidimensional, 448-450
por partes, 443-448
quadrática, 410-413
quadrática inversa, 164-166
spline cúbica por partes, 446-447
- Interpolação polinomial, 404-427
coeficientes do polinômio, 406-407
de ordem superior, 421-422
extrapolação, 419-422
de Lagrange, 415-419
de Newton, 408-416
Lagrange, 417-418
Newtint, 415-416
oscilações, 421-424
polyfit, 408
polyval, 408
- Interpolação por splines e por partes, 428-457
condições de extremidade, 442-443
cúbico, 437-443
- inter interpolação bilinear, 448-450
linear, 429-432
multidimensional, 448-45
quadrático, 433-438
tabela de consulta, 432-434
- Intervalo de classe modal, 329-330
- inv, 218
- Inversa, 215. *Ver também* Matriz inversa
- Isle Royale National Park, 584-585
- Iteração de
Jacobi, 286-287
ponto fixo, 152-156
ponto fixo simples, 152-156
um ponto, 152
- J**
- j, 27
- Jacobiano, 294-297
- L**
- Laço
com pós-teste, 68-69
com pré-teste, 68-69
com teste no meio, 68-69
infinito, 68-70
- Laços, 65-70
- Lago Platte, Michigan, 451
- Lagrange, 417-418
- Largura de linha, 38-39
- Lei(s) de
conservação, 11-14
constitutivas, 523-525
- Darcy, 524-525
- difusão de Fick, 541-542
- Fick, 524-525
- Fourier, 451, 524-525, 618-619
- Hooke, 197-198, 209, 358-359, 524-525, 553-554
- Joule, 513-514
- Kirchhoff, 127-128, 222-223
- Kirchhoff das correntes, 222-224
- Kirchhoff das tensões, 223-224
- movimento de Newton, 127-128
- Ohm, 223-224, 424-425, 513-514, 524-525
- resfriamento de Newton, 21-22, 541-542, 585-586
- taxa de ordem n , 543-544
- Torricelli, 108-109
- viscosidade de Newton, 524-525, 541-542
- M**
- Mal condicionado/mau condicionamento, 95-96, 230-231, 242-243, 272-273, 559-560
- Manchas solares, 400-402
- Manipulação aritmética dos números no computador, 101-103
- Manipulando matrizes no MATLAB, 213-219
- Mantissa, 97-100
- Matemática computacional, 1
- MATLAB
algarismos significativos, 27
animação, 70-71
animamento, 71-74
arquivos-M, 48-53
arredondamento, 35-36
case sensitive, 25-26
coeficientes de polinômios, 405
comandos de formatação, 27
funções nativas, 34-37. *Ver também* Função gráficos, 37-41
impressão após atribuição, 25-26
janelas, 24-25

- linhas vazias, 24-25
 modo calculadora, 24-25
 número imaginário unitário, 27
 operadores relacionais, 59
 pré-alocação de memória, 66-68
 prompt de comando, 24-25
 recursos adicionais, 39-41
 reticências, 30-31
 toolbox de estatística, 330n
- Matriz, 27-28
 aumento, 216
 companheira, 171-172
 de banda, 213
 de Hilbert, 275-276
 de permutação, 215, 218, 226-227, 260
 de transposição, 215
 de Vandermonde, 281-282, 407
 definição, 210-211
 diagonal, 211-212
 dimensão, 210-211
 equações algébricas lineares, e, 219-220
 identidade, 211-212, 215
 inversa, 215. *Ver também* Matriz inversa
 Jacobiana, 296-297
 linha/coluna, 210-211
 permutação, 215
 quadrada, 211-213
 regras de operação, 213-219
 transposta, 27-28, 215
 triangular inferior, 213
 triangular superior, 211-212
 tridiagonal, 213
- Matriz inversa, 215, 219-220, 229-230, 268-277
 avaliação do condicionamento de matrizes, 275-277
 calculando a inversa, 268-270
 cálculos de respostas a estímulos, 269-272
 inv, 218
 MATLAB, 276-278
 mau condicionamento, 272-273
 normas de vetores e de matrizes, 273-274
 número de condicionamento de matriz, 274-278
 problema do saltador de *bungee jumping*, 271-272
 max, 35-36, 243-244, 329-330
 mean, 35-36, 329-330
 median, 329-330
 Mediana, 325-326
- Medida da dispersão, 326-327
 Medida da localização, 325-327
 Menor frequência detectável, 396-397
 Menores, 231-232
 Método(s)
 abertos de localização de raízes, 151-181
 da inclinação do ponto, 554-555
 da potência, 310-313
 de interpolação linear, 140-141, 164
 de localização de raízes de Brent, 163-168
 de passo único, 554-555
 de primeira ordem, 559-560
 diferencial, 543-544
 do polinômio, 306-307
 do ponto médio, 485-486, 566-570
 do tiro, 620-628
- Método Babilônico, 85-86, 120-121, 177-178
 Método(s) da secante, 161-164
 modificada, 162-163
- Método de Butcher, 571-572
- Método de Euler, 9-10, 554-561, 572-575
 -Cauchy, 554-555
 explícito, 603-604
 implícito, 603-604
 regressivo, 603-604
- Método de Gauss-Seidel, 284-291
- Método de Heun, 562-567
 com início não automático, 597-601
 sem iteração, 569-570
- Método de Newton-Raphson, 156-161, 293-298
 para o Problema do Saltador de *Bungee Jumping*, 160-161
- Método de otimização de Brent, 194
- Método de Ralston, 569-570
- Método RK de quarta ordem, 569-573
- Método RK de quarta ordem clássico, 569-573
- Métodos
 de acente, 197-198
 de declive, 197-198
 de diferenças finitas, 627-635
 de passo múltiplo, 597-602
 diretos, 197-198
 dos coeficientes indeterminados, 503-506
 dos valores com sinal, 96-97
 gradientes, 197-198
 intervalares, 131-143
 não gradientes, 197-198
- Métodos adaptativos e sistemas rígidos, 588-615
bungee jumping com corda, 588-597
 de Heun de início não automático, 597-601
 de passo múltiplo, 597-602
 de Runge-Kutta, 588-615
 estimativas de erro, 600-602
 eventos, 594-597
 funções do MATLAB, 589-592, 605-606
 rigidez, 601-608
- Métodos de Fehlberg, 589-590
- Métodos de Runge-Kutta, 567-576
 adaptativos, 588-597
 Fehlberg, 589-590
- Métodos numéricos
 definição, 1
 o que é abordado neste livro, 15-16
 por que estudá-los, 1-2
 reformulação, 8-9
- Métodos RK, 567-576
 de quinta ordem, 571-572
 de segunda ordem, 568-570
 embutidos, 588-589
 -Fehlberg, 589-590
- Métodos/cálculos iterativos, 90-91, 284-302
 algoritmo computacional, 93-96
 estimativas de erro, 91-93
 Gauss-Seidel, 284-291
 método de Jacobi, 286-287
 relaxação, 287-291
 sistemas não lineares, 291-298
- min, 35-36, 329-330
- Minimax, 337-338
- Minimos quadrados, 338-339
 linear geral, 367-370
- Moda, 325-326
- mode, 329-330
- Modelagem matemática, 5
- Modelo de haste aquecida, 247-250
- Modelo de Michaelis-Menten, 147-149, 351-352
 de segunda ordem, 351-352
- Modelo de Streeter-Phelps, 202-203
- Modelo exponencial, 344
- Modelo logístico, 584
- Modelos massa-mola, 587
- Modelos predador-presa, 578-583
- Modo calculadora, 24-25
- Módulo de rigidez, 518-519
- Módulo de Young, 200-201

- Moler, Cleve, 165-167, 421n, 510-512
 movie, 69-70
 Multimodal, 185-186
 Multiplicação de matrizes, 213-214
 Multiplicação de matrizes no MATLAB, 33-34
 Multiplicação de vetores e matrizes, 32-33
- N**
- Não homogêneo, 304-305
 nargin, 63-64
 natspline, 452-453
 nearest (interpolação por vizinho mais próximo), 445-446
 n -ésima diferença dividida finita, 412-413
 Newtint, 415-416
 newtraph, 160-161
 Nó, 431-432
 norm, 276-277
 Norma, 273-274, 276-277
 da soma das colunas, 274-275
 da soma das linhas, 274-275
 espectral, 274-275
 Euclidiana, 273
 Normalização, 97-100, 237
 Normas de matrizes e de vetores, 273-274
 NOT, 59
 Notação
 de ponto, 522n
 posicional, 96-97
 Número
 de Avogadro, 98-99
 de condicionamento da matriz, 274-278
 de manchas solares de Wolf, 400-401
 de Reynolds, 16-17, 149, 173-174, 455-456
 imaginário unitário, 27
 Números aleatórios, 331-337
- O**
- ode113, 591-592, 601-602
 ode15s, 605-606
 ode23, 589-590
 ode23s, 605-606
 ode23t, 605-606
 ode23tb, 605-606
 ode45, 591-592
 odeset, 593-594
 Onda dente de serra, 402-403
 Onda triangular, 402-403
 ones, 28-29
 Operações
 com arranjos, 33-34
 com ponto flutuante (flops), 239-241
 elemento por elemento, 33-34
 matemáticas, 31-35
 barra invertida, 219-220, 266, 370-371
 dois pontos, 29-30
 Operadores relacionais, 59
 optimset, 168-170
 OR, 59
 Ordem envolvente reversa, 396-399
 Ortogonal, 307-308
 Oscilações, 421-424
 Otimização
 bidimensional, 184-185
 multidimensional, 195-198
 unidimensional, 184-195
 Ótimo global, 185-186
 Ótimo local, 185-186
 Overflow, 101
- P**
- Painel, 39-40
 Palavra, 95-96
 Par da transformada de Fourier, 393
 Parábola, 410
 Parábola lateral, 164-165
 Parâmetros, 5
 Passando parâmetros, 78
 pause, 68-69
 pchip, 476-477
 Pequeno número de equações. *Ver* Resolução de pequeno número de equações
 Percentil, 49-50, 325-326
 Período, 308-309, 380-381
 pi, 27
 Pivotamento, 242-246
 completo, 242-243
 parcial, 242-245
 Plano de frequência, 391-393
 Plano temporal, 391-393
 Plínio, o Grande, 608-609
 plot, 38-39
 plot3, 582-583
 polar, 47
 Polinômio, 169-173
 característico, 304-305
 cúbico, 414
 de segunda ordem, 410
 quadrático, 410
 Polinômio de Lagrange, 165
 de ordem superior, 415-416
 Polinômio interpolador de Lagrange, 415-419
 de segunda ordem, 415-416
 de Newton, 408-416
 linear de Lagrange, 416
 Poluição do ar, 277-280
 de interiores, 277-280
 poly, 171-172, 307
 polyfit, 350-351, 408
 polyval, 350-351, 408
 Ponto crítico, 580-581
 Pré-alocação de memória, 66-68
 Precisão, 88-90
 simples, 121-122
 Primeira diferença dividida, 412-413
 Princípio
 da probabilidade máxima, 341-342
 de Arquimedes, 150
 de conversão da massa, 206
 Princípios fundamentais (problemas de projeto), 127-128
 Problemas de dinâmica, 303
 Problemas de estática, 303
 Problemas de valor de contorno, 616-643
 condições de contorno na derivada, 624-626, 629-633
 introdução e contextualização, 616-621
 método do tiro, 620-628
 métodos de diferenças finitas, 627-635
 problemas de valor inicial, comparação, 616-618
 Problemas de valor inicial, 553-579
 método de Euler, 554-561, 572-575
 método de Heun, 562-567
 método do ponto médio, 566-568
 métodos RK, 567-573, 574-576
 problemas de valor de contorno, comparação, 616-618
 rk4sys, 576-579
 sistemas de equações, 572-579
 visão geral, 554-555
 Problemas de variáveis
 concentradas, 206-207
 distribuídas, 207
 Problemas do saltador de *bungee jumping*
 estudo de caso, 16-19
 método de Euler, 573-574
 função evento, 595
 método RK de quarta ordem, 575-576
 MATLAB, 221-222
 matriz inversa, 271-272
 método de Newton-Raphson, 160-161
 EDO, 607-609
 contextualização, 4-5
 solução analítica, 6-9
 solução numérica, 9-12

- Processo de extração em estágios, 252-253
 prod, 35-36
Produto
 externo, 32-33
 interno, 32-33, 103-104
Programação estruturada, 56-70
 decisões, 56-64
 for...end, 65-68
 if, 56-58
 if...else, 60-61
 if...elseif, 60-62
 laços, 65-70
 switch, 67-68
 while...break, 67-69
Projeto de cima para baixo, 71-72
Prompt de comando, 24-25
Proporcionalidade, 271-272
- Q**
quad, 511-513
quadadapt, 511-512
quadl, 511-513
Quadratura, 462-463
 adaptativa, 509-513
 de Gauss, 502-510
 de Lobatto, 511-512
Qualidade do ajuste, 336-344
quiver, 538-539
- R**
rand, 332-333
randn, 334-335
range, 329-330
Rayleigh, Lord, 16-17
Razão áurea, 187-188
Reações químicas, 298-301
realmax, 101
realmín, 101
Reator de batelada, 585-586
Reator de fluxo em pistão axialmente disperso, 636-637
Regra 1/3, 474-481
Regra 1/3 de Simpson, 474-481
Regra 3/8, 478-481
Regra 3/8 de Simpson, 478-481
Regra de Boole, 450, 510-511, 571-572
Regra de Cramer, 231, 233-234
Regra do Trapézio, 468-475, 480-481, 566-567
 com segmentos desiguais, 481-482
 de dois segmentos, 487-488
Regras de Simpson, 474-481
Regressão. Ver Ajuste de curvas
- Regressão linear, 347-350
 múltipla, 364-368
 por mínimos quadrados, 336-344, 348-349
 Regressão não linear, 371-373
 Regressão polinomial, 361-365, 368-370
 Regresso por mínimos quadrados, 321-322, 336-344, 348-349
 regrlin, 348-350
Representação
 de inteiro, 96-98
 dos números no computador, 95-101
 em ponto flutuante, 97-100
 na base-8, 96-97
Resíduo, 336-337, 340-341, 626-627
Resolução de um número pequeno de equações
 eliminação de variáveis, 233-235
 método gráfico, 230-232
 regra de Cramer, 233-234
Reta de melhor ajuste, 336-344
Reticências, 30-31
Retificador de meia onda, 402-403
Rigidiz, 601-608
rk4sys, 576-579
Rolamentos de rolos, 203-204
romberg, 501-503
roots, 171-172, 307
round, 35-36
Runge, Carl, 421-422
- S**
Saltador de bungee jumping com corda, 607-609
Saltador de bungee jumping em queda livre. Ver Problema do saltador de bungee jumping
save, 55-56
Script file, 48-50
script, 48-50
Segunda diferença dividida, 414
 finita, 412-413
 finita progressiva, 114-115
Segunda lei de movimento de Newton, 5, 523-524
semilogy, 44-45
Sequência de caracteres (strings), 30-31
Série de Fourier, 393
 contínua, 387-389
Série(s) de Taylor, 103-111
 aproximação de uma função, 107-108
 de ordem zero, 108-109
 de segunda ordem, 105-106
erro de truncamento, 109-111
expansão em série de Taylor de ordem n, 106-107
resto, 108-110
Série temporal, 380-381
short, 27
short e, 27
short eng, 27
short g, 27
sign, 61-62, 554n
Significando, 97-98
Simulação de Monte Carlo, 334-335
Singular, 230-231
Sistema
 binário, 96-97
 de duas massas e três molas, 307-308
 de duas molas, 198
 de equações não lineares, 291-298
 decimal, 95-97
 massa-mola, 307-308
 massa-mola amortecido, 584-585
 na base-2, 96-97
 na base-10, 96-97
 octogonal, 96-97
 pentadiagonal, 253
 rígido, 601-602
 tridiagonal, 245-247
Sistemas numéricos, 95-97
size, 219
Sobredeterminado, 219-220, 370-371
Sobrerelaxamento, 287-288
Sobrerelaxamento sucessivo (SOR), 287-288
Solução analítica, 8-9
Solução na forma fechada, 8-9
SOR, 287-288
sort, 35-36
spline, 443-444
 amarrado, 442-443
 cúbico, 431-432, 437-443
 cúbico natural, 441-442
 de primeira ordem, 430-432
 interpolação por splines cúbicos por partes, 446-447
 linear, 429-432
 quadrático, 431-438
std, 329-330
Subfunções, 51-52
subplot, 39-40
Sub-relaxamento, 287-288
Substituição
 regressiva, 237, 256
 sucessiva, 152, 292-293

- sum, 35-36
 Superposição, 271-272
 switch, 63-64
- T**
 Tabela de consulta, 432-434
 Tabela de diferenças divididas, 413-414
 TabelaConsultaBin, 433-434
 Tamanho total da amostra, 396-397
 \tanh , 7n
 Tanque esférico, 585-586
 Taxa de quadros, 69-70
 Técnicas de localização de raízes, 126-181
 bissecção, 134-143
 busca incremental, 131-134
 estimativas iniciais, 129-131
 falsa posição, 140-143
 interpolação quadrática inversa, 164-166
 iteração de ponto fixo simples, 152-156
 método de Brent, 163-168
 método de Newton-Raphson, 156-161
 métodos abertos, 151-181
 métodos da secante, 161-164
 métodos gráficos, 128-131
 métodos intervalares, 131-143
 Tentativa e erro, 123
 Teorema de Taylor, 103-104
 Teorema do valor médio para derivada, 108-110
 Terceira diferença dividida, 414
 Termocâlna, 451, 455
 Terremotos, 314-316
 Teste de hipótese, 321-322
 The MathWorks, Inc., 41
 tic, 69-70
- toc, 69-70
 Toolbox de estatística, 330n
 Transferência de calor, 451-455
 Transformada
 discreta de Fourier (DFT), 394-399
 inversa de Fourier, 393-394
 rápida de Fourier (FFT), 394-396
 Transformada de Fourier, 393
 de $f(i)$, 393
 Transposta, 27-28
 (matriz), 215
 trap, 472-474
 trapuneq, 482-483
 trapz, 483-484
 Tridiag, 247
 triplequad, 487-488
 Tronco, 150
 Tukey, J. W., 396
- U**
 Underflow, 101
 Unimodal, 188-189
- V**
 Valor de influência, 425-426
 Valor médio quadrático da corrente, 513-517
 Valor-padrão, 63-64
 Valores característicos. Ver também Autovaleores
 var, 329-330
 varargin, 78
 Variância, 326-327
 Variável,
 Booleana, 609-610
 dependente, 5, 547
- fictícia, 81
 independente, 5, 547
 local, 51-52
 lógica, 609-610
 Velocidade do saltador de *bungee jumping*, 78-82
 Velocidade terminal, 8-9
 Vetor
 cartesiano, 85-86
 coluna, 27-28, 211-212
 linha, 27-28, 210-211
 Vetorização, 66-67
 Viés, 88-89
 Viga em balanço, 202-203
 Visão geral do livro, 15-16
 métodos numéricos abordados, 15
 Parte 1, 2-3
 Parte 2, 124-125
 Parte 3, 207-208
 Parte 4, 322-323
 Parte 5, 459-461
 Parte 6, 551-552
 Viscosidade, 404
 Visualização de campos, 538-540
 Volterra, 578-579
- W**
 while, 67-68
 while...break, 67-69
 who, 28-29
 whos, 28-29
 Wolf, Johann Rudolph, 400-401
 www.mathworks.com, 41
- Z**
 Zeros, 28-29, 123