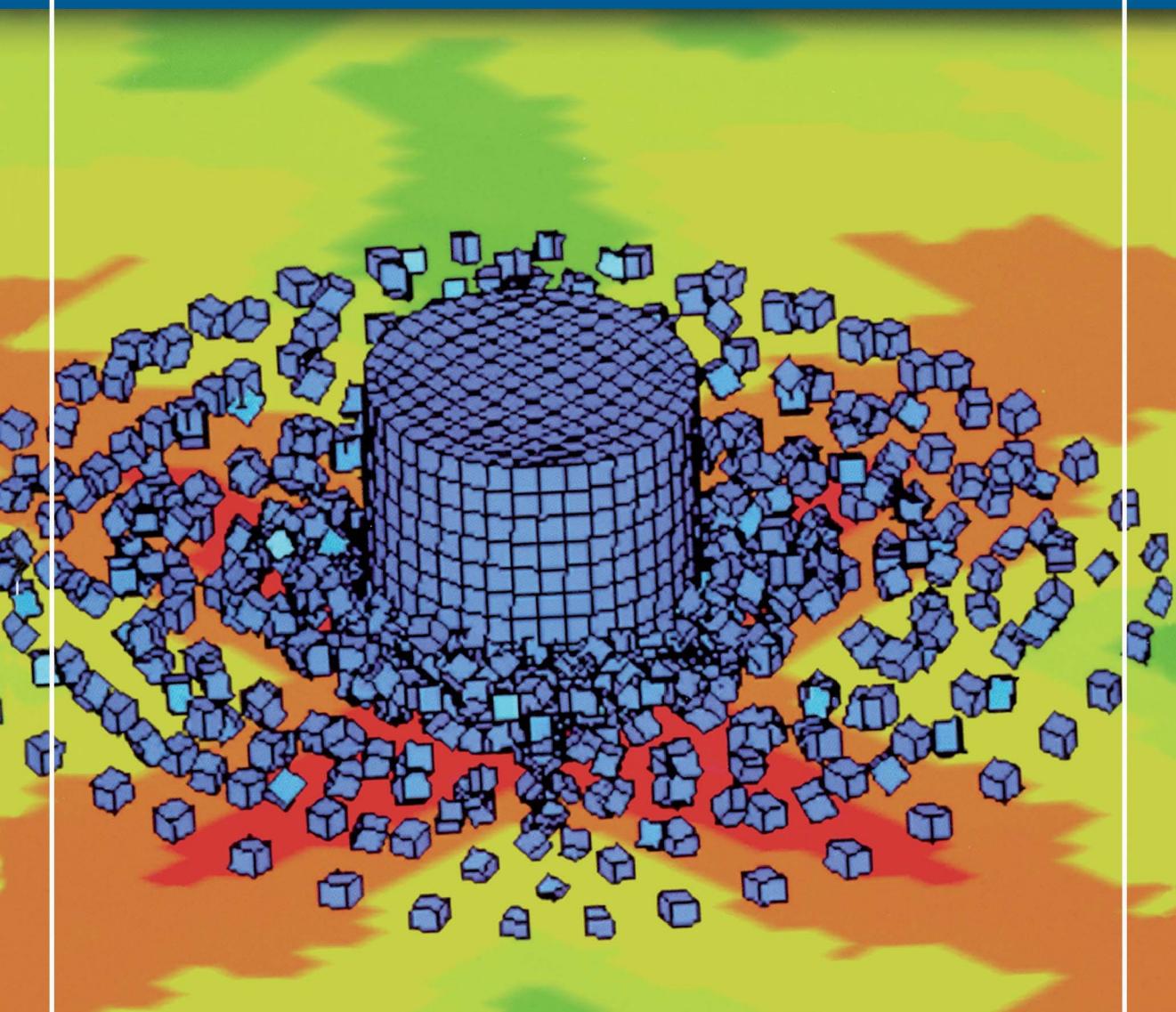


# Métodos Numéricos para Engenheiros e Cientistas

**Uma introdução com aplicações usando o MATLAB**



**AMOS GILAT  
VISH SUBRAMANIAM**



# Métodos Numéricos para Engenheiros e Cientistas

**Amos Gilat**, Ph.D., é Professor em Engenharia Mecânica na The Ohio State University. Seus principais interesses de pesquisa estão voltados à plasticidade, ao desenvolvimento de técnicas experimentais para testar materiais em uma ampla faixa de taxas de deformação e temperaturas, e à investigação das relações constitutivas relacionadas à viscoplasticidade. A pesquisa do Dr. Gilat tem sido apoiada pela National Science Foundation, pela NASA, pelos Departamentos de Energia e Defesa dos EUA, e por várias indústrias.

**Vish Subramaniam**, Ph.D., é professor de Engenharia Mecânica e Físico-Química na The Ohio State University. Seus principais interesses de pesquisa estão voltados ao plasma e aos processos físicos do laser, particularmente aqueles que envolvem fenômenos em condições de desequilíbrio. A pesquisa do Dr. Subramaniam é experimental e teórica, e conta com o apoio do Departamento de Defesa dos EUA, da National Science Foundation e de numerosas indústrias.



G463m Gilat, AmosPozo.

Métodos numéricos para engenheiros e cientistas [recurso eletrônico] : uma introdução com aplicações usando o MATLAB / Amos Gilat, Vish Subramaniam ; tradução Alberto Resende de Conti. – Dados eletrônicos. – Porto Alegre : Bookman, 2008.

Editedo também como livro impresso em 2008.  
ISBN 978-85-7780-297-5

1. Computação – Programas – MATLAB. 2. MATLAB-Engenharia. I. Subramaniam, Vish. II. Título.

CDU 004.4:62MATLAB

AMOS GILAT  
VISH SUBRAMANIAM

*Departamento de Engenharia Mecânica  
The Ohio State University*

# Métodos Numéricos para Engenheiros e Cientistas

---

Uma introdução com aplicações usando o MATLAB

---

**Tradução:**

Alberto Resende de Conti  
Doutor em Engenharia Elétrica pela UFMG

**Consultoria, supervisão e revisão técnica desta edição:**

Antonio Pertence Júnior  
Professor titular de Matemática da Faculdade de Sabará/MG  
Membro efetivo da SBM

Versão impressa  
desta obra: 2008



2008

Obra originalmente publicada sob o título  
*Numerical Methods for Engineers and Scientists: An Introduction with Applications Using MATLAB*, 1st. Edition  
ISBN 978-0-471-73440-6

Copyright © 2008. John Wiley & Sons, Inc. All rights reserved. This translation published under license.

Capa: *Henrique Chaves Caravantes, arte sobre capa original*

Preparação de original: *Renato A. Merker*

Supervisão editorial: *Denise Weber Nowaczyk*

Editoração eletrônica: *Techbooks*

A imagem da capa mostra uma simulação numérica do impacto de pedras de gelo na superfície de uma nave espacial. Cortesia do Dr. Kelly Carney, NASA Glenn Research Center, Cleveland, Ohio.

Reservados todos os direitos de publicação, em língua portuguesa, à  
ARTMED® EDITORA S.A.  
(BOOKMAN® COMPANHIA EDITORA é uma divisão da ARTMED® EDITORA S.A.)  
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.

SÃO PAULO  
Av. Angélica, 1.091 - Higienópolis  
01227-100 São Paulo SP  
Fone (11) 3665-1100 Fax (11) 3667-1333

SAC 0800 703-3444

IMPRESSO NO BRASIL  
PRINTED IN BRAZIL

*Para Yaela, Taly e Edan*

*Para Radha, Priya e Sonya*

# Prefácio

---

Este livro-texto é recomendado para um curso inicial sobre métodos numéricos dedicado a estudantes de engenharia e outras disciplinas da área de ciências exatas. O livro engloba os fundamentos dos métodos numéricos de um ponto de vista aplicado. Ele explica as idéias básicas de vários métodos e mostra a sua utilização na solução de problemas na engenharia e nas ciências exatas em geral.

No passado, cursos sobre métodos numéricos eram essencialmente matemáticos, enfatizando a análise numérica e a teoria. Mais recentemente, graças à disponibilidade de computadores pessoais e de programas tão poderosos quanto acessíveis, o conteúdo e a natureza de um curso inicial sobre métodos numéricos voltado para estudantes de engenharia e ciência têm mudado. A ênfase está cada vez mais na aplicação e na implementação de métodos numéricos em ferramentas de fácil utilização. Em um curso típico, os estudantes aprenderão os fundamentos dos métodos numéricos, a programação de computadores (ou melhorarão suas habilidades de programação se já souberem programar) e usarão programas avançados como uma ferramenta de solução de problemas. O MATLAB é um bom exemplo de um desses programas. Ele pode ser usado por estudantes para escrever os seus próprios programas e também como uma ferramenta para resolver problemas a partir do uso de suas funções residentes. Um dos objetivos de um curso sobre métodos numéricos é preparar estudantes de ciência e engenharia para cursos futuros em suas áreas de especialização (e suas futuras carreiras), nos quais eles terão que utilizar computadores para resolver problemas.

## Principais objetivos do livro

Ensinar os fundamentos dos métodos numéricos, com ênfase nos métodos mais essenciais.

Fornecer aos estudantes a oportunidade de melhorar suas habilidades de programação usando o ambiente MATLAB para implementar algoritmos.

Ensinar o emprego do MATLAB como uma ferramenta (usando suas funções residentes) para resolver problemas de ciência e engenharia, e para verificar os resultados de outros programas que os estudantes possam escrever.

### Características/pedagogia do livro

- Este livro foi escrito em uma linguagem simples, clara e direta. Freqüentemente são usados itens e seqüências de passos em vez de texto para listar fatos e detalhes referentes a um assunto específico.
- Muitas ilustrações são usadas para explicar os princípios dos métodos numéricos.
- Muitos exemplos e problemas de final de capítulo envolvem problemas reais na ciência e na engenharia.
- O MATLAB está integrado ao longo do texto e nos exemplos. Um fundo cor de laranja suave é usado quando a sintaxe do MATLAB é exibida.
- Comentários explicando os comandos são colocados ao lado da sintaxe do MATLAB.
- Funções residentes do MATLAB, associadas aos métodos numéricos discutidos, são apresentadas em detalhe.
- Os problemas de final de capítulo são divididos em três grupos:
  - (a) **Problemas de solução manual:** Problemas relacionados à melhoria do entendimento dos métodos numéricos. Nesses problemas, pede-se que os estudantes respondam a questões relacionadas aos fundamentos dos métodos numéricos e que realizem alguns passos desses métodos manualmente.
  - (b) **Problemas de programação do MATLAB:** Problemas elaborados para fornecer ao estudante a possibilidade de melhorar suas habilidades de programação. Nesses problemas, pede-se que seja usado o MATLAB para escrever programas de computador implementando vários métodos numéricos (programas e funções em arquivo).
  - (c) **Problemas na matemática, na ciência e na engenharia:** Problemas na ciência e na engenharia que devem ser resolvidos com o uso de métodos numéricos. O objetivo é treinar os estudantes no uso de métodos numéricos para resolver problemas que eles verão em cursos futuros ou na prática. Espera-se que os estudantes usem os programas apresentados no livro, programas que eles escreveram e funções residentes do MATLAB.

### Organização do livro

**Capítulo 1:** Fornece uma introdução geral sobre os métodos numéricos e sobre a maneira pela qual computadores armazenam números e realizam operações numéricas. Inclui também uma seção de erros em soluções numéricas e uma seção dedicada aos computadores e à programação.

**Capítulo 2:** O segundo capítulo apresenta uma revisão dos conceitos matemáticos fundamentais usados nos capítulos seguintes, que cobrem os métodos numéricos. Ele deve ser usado como um lembrete dos conceitos com os quais os estudantes devem estar familiarizados após os seus cursos de primeiro e segundo ano em matemática. Como muitos desses tópicos estão associados a vários métodos

numéricos, sentimos que é preferível ter os fundamentos matemáticos reunidos em um único capítulo a tê-los dispersos ao longo do livro. Muitos dos tópicos tratados no Capítulo 2 e que são essenciais para a explicação de um determinado método numérico são repetidos em outros capítulos, no ponto onde esse determinado método é apresentado. Muitos professores provavelmente não abordarão todo o conteúdo do Capítulo 2 em classe, mas mencionarão um tópico e pedirão aos estudantes que o consultem quando necessário.

**Capítulos 3 a 9:** Esses sete capítulos apresentam vários métodos numéricos na ordem comumente seguida em um curso sobre métodos numéricos. Esses capítulos adotam o formato explicado a seguir.

### Organização de um capítulo típico

Uma lista dos tópicos tratados no capítulo é mostrada após o título do capítulo. A lista está dividida entre tópicos *centrais* e *complementares*. Os *tópicos centrais* são aqueles mais essenciais ao assunto do capítulo. Os *tópicos complementares* incluem temas mais avançados. Obviamente, essa divisão é subjetiva. A intenção é ajudar os professores no planejamento de seu curso quando não houver tempo para falar de todos os tópicos. Na prática, se todos os tópicos forem cobertos, a divisão pode ser ignorada.

A primeira seção do capítulo fornece uma visão geral com exemplos ilustrativos de problemas na ciência e na engenharia em que os métodos descritos no capítulo podem ser utilizados. Esta seção também explica as idéias básicas contidas na classe específica de métodos descritos no capítulo. As seções seguintes falam dos tópicos centrais do capítulo. Em seguida, uma seção especial discute as funções do MATLAB que implementam os métodos numéricos descritos no capítulo e como eles podem ser usados para resolver problemas. As últimas seções do capítulo tratam dos tópicos complementares.

### Ordem dos tópicos

É praticamente impossível escrever um livro-texto em que a ordem dos tópicos satisfaça a todos os professores. Neste livro, os assuntos principais são apresentados em uma ordem que é típica de um curso inicial sobre métodos numéricos. O Capítulo 3 trata da solução de equações não-lineares. Ele lida principalmente com a solução de apenas uma equação, o que é uma aplicação simples de métodos numéricos. Este capítulo também inclui, como um tópico complementar, uma seção sobre solução de sistemas de equações não-lineares. O Capítulo 4 aborda a solução de sistemas de equações lineares. Um tópico complementar presente nesse capítulo trata do problema dos autovalores. O Capítulo 5 é sobre ajuste de curvas e da interpolação, e os Capítulos 6 e 7 tratam da diferenciação e da integração, respectivamente. Finalmente, a solução de equações diferenciais ordinárias (EDOs) é apresentada nos dois capítulos finais. O Capítulo 8 trata da solução de problemas de valor inicial (de primeira ordem, sistemas e de ordem superior) e o Capítulo 9 analisa problemas de valor de contorno.

## Programas de MATLAB

Este livro contém vários programas de MATLAB. Eles são claramente identificados como funções criadas pelos usuários ou como arquivos de programa. Todos os programas estão listados no Apêndice B. Os programas, ou códigos, são escritos de uma maneira simples e fácil de seguir. A ênfase desses programas está voltada aos fundamentos e em como programar um algoritmo referente a um método numérico específico. Obviamente, os programas não são gerais e por isso não cobrem todas as circunstâncias possíveis quando executados. Eles não são escritos na perspectiva de serem mais curtos, mais rápidos ou mais eficientes. Eles são escritos de forma a serem fáceis de seguir. Assume-se que a maioria dos estudantes tenha apenas um entendimento limitado sobre o MATLAB e sobre a programação, e a apresentação do MATLAB dessa maneira permitirá um avanço em suas habilidades de programação. Usuários mais avançados do MATLAB são encorajados a escrever programas mais sofisticados e eficientes e a comparar o desempenho desses programas com aqueles apresentados no livro.

## Promessa da primeira edição

### A. Precisão do texto

O manuscrito foi revisado por mais de 20 professores de engenharia. Dois revisores (além dos autores) verificaram todos os exemplos e problemas de final de capítulo de forma independente. Um editor revisou o manuscrito procurando problemas de gramática e de pontuação. O texto final foi impresso diretamente pelos autores, o que elimina a introdução de potenciais erros durante a composição do texto final.

### B. Precisão da solução

Os autores escreveram todas as soluções dos problemas de final de capítulo. Elas foram verificadas de forma independente por dois revisores.

### C. Confiabilidade

O manuscrito foi testado em classe por mais de 200 estudantes em três cursos separados para garantir a confiabilidade, a facilidade de leitura e a utilidade do texto. As sugestões desses estudantes foram incorporadas na versão final do texto.

### D. Material de apoio\*

- (a) um manual completo de soluções cuja precisão foi verificada três vezes;
- (b) plano de estudos com tarefas sugeridas para ajudar à fácil integração do texto em seu curso;

---

\* Acesse a Área do Professor no site [www.bookman.com.br](http://www.bookman.com.br) e veja quais materiais estão disponíveis.

- (c) guias de conversão de outros textos importantes sobre métodos numéricos para mostrar onde cada seção deste livro que você utiliza é tratada naquele texto, o que o ajuda a passar rapidamente do velho para o novo;
- (d) versões eletrônicas de todas as figuras e tabelas do texto, que podem ser usadas em slides e em questões de prova baseados em imagens do livro;
- (e) arquivos.m de todos os programas apresentados no texto.

Muitas pessoas ajudaram na preparação deste livro. Gostaríamos de agradecer aos revisores pelos vários comentários e sugestões feitas.

Lawrence K. Agbezuge, *Rochester Institute of Technology*

David Alciatore, *Colorado State University*

Salame Amr, *Virginia State University*

John R. Cotton, *Virginia Polytechnic Institute and State University*

David Dux, *Purdue University*

Venkat Ganesan, *University of Texas-Austin*

Michael R. Gustafson II, *Duke University*

Alain Kassab, *University of Central Florida*

Tribikram Kundu, *University of Arizona*

Ronald A. Mann, *University of Louisville*

Peter O. Orono, *Indiana University Purdue University Indianapolis*

Charles Ritz, *California State Polytechnic University-Pomona*

Douglas E. Smith, *University of Missouri-Columbia*

Anatoliy Swishchuk, *University of Calgary*

Ronald F. Taylor, *Wright State University*

Brian Vick, *Virginia Polytechnic Institute and State University*

Também gostaríamos de agradecer a Joseph Hayton, editor de aquisição, Lisa Wojcik, editora de produção, Sandra Kim, assistente editorial e Harry Nolan, diretor de projeto, todos da Wiley. Agradecimentos especiais às filhas do Professor Subramaniam, Sonya e Priya, que digitaram e revisaram rascunhos iniciais de alguns capítulos.

Nossa intenção foi escrever um livro que fosse ao mesmo tempo útil para os estudantes e para os professores. Gostaríamos de receber comentários que contribuam para melhorias em edições futuras.

Amos Gilat ([gilat.1@osu.edu](mailto:gilat.1@osu.edu))

Vish Subramaniam ([subramaniam.1@osu.edu](mailto:subramaniam.1@osu.edu))

Columbus, Ohio

Fevereiro, 2007

# Sumário resumido

---

<b>Capítulo 1</b>	<i>Introdução</i>	<b>21</b>
<b>Capítulo 2</b>	<i>Fundamentos matemáticos</i>	<b>41</b>
<b>Capítulo 3</b>	<i>Resolvendo equações não-lineares</i>	<b>73</b>
<b>Capítulo 4</b>	<i>Resolvendo um sistema de equações lineares</i>	<b>113</b>
<b>Capítulo 5</b>	<i>Ajuste de curvas e interpolação</i>	<b>199</b>
<b>Capítulo 6</b>	<i>Diferenciação numérica</i>	<b>253</b>
<b>Capítulo 7</b>	<i>Integração numérica</i>	<b>287</b>
<b>Capítulo 8</b>	<i>Equações diferenciais ordinárias: problemas de valor inicial</i>	<b>327</b>
<b>Capítulo 9</b>	<i>Equações diferenciais ordinárias: problemas de valor de contorno</i>	<b>407</b>
<b>Apêndice A</b>	<i>Introdução ao MATLAB</i>	<b>441</b>
<b>Apêndice B</b>	<i>Programas de MATLAB</i>	<b>471</b>
<b>Índice</b>		<b>475</b>

# Sumário

---

<b>Capítulo 1</b>	<b><i>Introdução</i></b>	<b>21</b>
1.1	Fundamentos	<b>21</b>
1.2	Representação de números em um computador	<b>24</b>
1.3	Erros em soluções numéricas	<b>29</b>
1.3.1	<i>Erros de arredondamento</i>	<b>30</b>
1.3.2	<i>Erros de truncamento</i>	<b>33</b>
1.3.3	<i>Erro total</i>	<b>34</b>
1.4	Computadores e programação	<b>34</b>
1.5	Problemas	<b>37</b>
<b>Capítulo 2</b>	<b><i>Fundamentos matemáticos</i></b>	<b>41</b>
2.1	Fundamentos	<b>41</b>
2.2	Conceitos de cálculo e pré-cálculo	<b>42</b>
2.3	Vetores	<b>46</b>
2.3.1	<i>Operações com vetores</i>	<b>48</b>
2.4	Matrizes e álgebra linear	<b>50</b>
2.4.1	<i>Operações com matrizes</i>	<b>51</b>
2.4.2	<i>Matrizes especiais</i>	<b>53</b>
2.4.3	<i>Inversa de uma matriz</i>	<b>53</b>
2.4.4	<i>Propriedades de matrizes</i>	<b>54</b>
2.4.5	<i>Determinante de uma matriz</i>	<b>55</b>
2.4.6	<i>A regra de Cramer e a solução de sistemas de equações lineares simultâneas</i>	<b>56</b>
2.4.7	<i>Normas</i>	<b>56</b>
2.5	Equações diferenciais ordinárias (EDO)	<b>58</b>
2.6	Funções de duas ou mais variáveis independentes	<b>61</b>
2.6.1	<i>Definição da derivada parcial</i>	<b>62</b>
2.6.2	<i>Regras da cadeia</i>	<b>62</b>
2.6.3	<i>O Jacobiano</i>	<b>64</b>
2.7	Expansão de funções em série de Taylor	<b>64</b>
2.7.1	<i>Série de Taylor para uma função de uma variável</i>	<b>65</b>
2.7.2	<i>Série de Taylor para funções de duas variáveis</i>	<b>65</b>
2.8	Problemas	<b>67</b>

---

<b>Capítulo 3</b>	<b><i>Resolvendo equações não-lineares</i></b>	<b>73</b>
3.1	Fundamentos	73
3.2	Estimação de erros em soluções numéricas	75
3.3	Método da bisseção	77
3.4	Método regula falsi	80
3.5	Método de Newton	81
3.6	Método da secante	87
3.7	Método da iteração de ponto fixo	88
3.8	Uso de funções residentes do MATLAB para resolver equações não-lineares	93
3.8.1	<i>O comando fzero</i>	93
3.8.2	<i>O comando roots</i>	94
3.9	Equações com múltiplas soluções	94
3.10	Sistemas de equações não-lineares	96
3.10.1	<i>Método de Newton para a solução de sistemas de equações não-lineares</i>	97
3.10.2	<i>Método da iteração de ponto fixo para a solução de sistemas de equações não-lineares</i>	101
3.11	Problemas	102
<b>Capítulo 4</b>	<b><i>Resolvendo um sistema de equações lineares</i></b>	<b>113</b>
4.1	Fundamentos	113
4.1.1	<i>Revisão de métodos numéricos para resolver sistemas de equações lineares algébricas</i>	114
4.2	Método de eliminação de Gauss	116
4.2.1	<i>Potenciais dificuldades encontradas com a aplicação do método de eliminação de Gauss</i>	124
4.3	Eliminação de Gauss com pivotação	126
4.4	Método de eliminação de Gauss-Jordan	129
4.5	Método de decomposição LU	132
4.5.1	<i>Decomposição LU usando o procedimento de eliminação de Gauss</i>	134
4.5.2	<i>Decomposição LU usando o método de Crout</i>	135
4.5.3	<i>Decomposição LU com pivotação</i>	142
4.6	Inversa de uma matriz	142
4.6.1	<i>Cálculo da inversa com o método de decomposição LU</i>	143
4.6.2	<i>Cálculo da inversa usando o método de Gauss-Jordan</i>	145
4.7	Métodos iterativos	145
4.7.1	<i>Método iterativo de Jacobi</i>	147
4.7.2	<i>Método iterativo de Gauss-Seidel</i>	147

4.8	Uso de funções residentes do MATLAB para solucionar sistemas de equações lineares	150
4.8.1	<i>Solução de um sistema de equações usando as divisões à esquerda e à direita do MATLAB</i>	150
4.8.2	<i>Solução de sistemas de equações usando a operação inversa disponível no MATLAB</i>	151
4.8.3	<i>Função residente do MATLAB disponível para o cálculo da decomposição LU</i>	151
4.8.4	<i>Funções residentes do MATLAB adicionais</i>	153
4.9	Sistemas tridiagonais de equações	154
4.10	Erro, resíduo, normas e número condição	159
4.10.1	<i>Erro e resíduo</i>	159
4.10.2	<i>Normas e número condição</i>	161
4.11	Sistemas mal condicionados	166
4.12	Autovalores e autovetores	167
4.12.1	<i>O método da potência básico</i>	171
4.12.2	<i>O método da potência inverso</i>	174
4.12.3	<i>O método da potência deslocado</i>	175
4.12.4	<i>O Método QR de fatoração e iteração</i>	176
4.12.5	<i>Uso de funções residentes do MATLAB para determinar autovalores e autovetores</i>	185
4.13	Problemas	187
<b>Capítulo 5</b>	<b>Ajuste de curvas e interpolação</b>	<b>199</b>
5.1	Fundamentos	199
5.2	Ajuste de curvas com equações lineares	201
5.2.1	<i>Medição da qualidade de um ajuste</i>	201
5.2.2	<i>Regressão linear por mínimos quadrados</i>	203
5.3	Ajuste de curvas com a linearização de equações não-lineares	207
5.4	Ajuste de curvas com polinômios quadráticos e de ordem superior	211
5.5	Interpolação usando um único polinômio	216
5.5.1	<i>Polinômios interpoladores de Lagrange</i>	218
5.5.2	<i>Polinômios interpoladores de Newton</i>	222
5.6	Interpolação por partes (Spline)	228
5.6.1	<i>Splines lineares</i>	228
5.6.2	<i>Splines quadráticas</i>	230
5.6.3	<i>Splines cúbicas</i>	234
5.7	Uso de funções residentes do MATLAB para fazer o ajuste de curvas e a interpolação	240

5.8	Ajuste de curvas usando uma combinação linear de funções não-lineares	242
5.9	Problemas	245

**Capítulo 6    *Diferenciação numérica*    253**

6.1	Fundamentos	253
6.2	Aproximação da derivada por diferenças finitas	255
6.3	Fórmulas de diferenças finitas usando a expansão em série de Taylor	260
6.3.1	<i>Fórmulas de diferenças finitas para a derivada primeira</i>	260
6.3.2	<i>Fórmulas de diferenças finitas para a derivada segunda</i>	264
6.4	Resumo de fórmulas de diferenças finitas para a diferenciação numérica	267
6.5	Fórmulas de diferenciação usando polinômios de Lagrange	269
6.6	Diferenciação usando o ajuste de curvas	270
6.7	Uso de funções residentes do MATLAB para realizar a diferenciação numérica	270
6.8	Extrapolação de Richardson	272
6.9	Erros na diferenciação numérica	274
6.10	Diferenciação parcial numérica	275
6.11	Problemas	279

**Capítulo 7    *Integração numérica*    287**

7.1	Fundamentos	287
7.1.1	<i>Visão geral sobre as abordagens usadas na integração numérica</i>	288
7.2	Métodos do retângulo e do ponto central	290
7.3	Método trapezoidal	292
7.3.1	<i>Método trapezoidal composto</i>	293
7.4	Métodos de Simpson	296
7.4.1	<i>Método de Simpson 1/3</i>	296
7.4.2	<i>Método de Simpson 3/8</i>	299
7.5	Quadratura de Gauss	301
7.6	Avaliação de integrais múltiplas	306
7.7	Uso de funções residentes do MATLAB para a integração	307
7.8	Estimação de erro na integração numérica	309
7.9	Extrapolação de Richardson	311
7.10	Integração de Romberg	313

7.11	Integrais impróprias	<b>317</b>
7.11.1	<i>Integrais com singularidades</i>	<b>317</b>
7.11.2	<i>Integrais com limites infinitos</i>	<b>317</b>
7.12	Problemas	<b>318</b>
<b>Capítulo 8</b>	<b><i>Equações diferenciais ordinárias: problemas de valor inicial</i></b>	<b>327</b>
8.1	Fundamentos	<b>327</b>
8.2	Métodos de Euler	<b>332</b>
8.2.1	<i>Método explícito de Euler</i>	<b>332</b>
8.2.2	<i>Análise do erro de truncamento no método explícito de Euler</i>	<b>336</b>
8.2.3	<i>Método implícito de Euler</i>	<b>340</b>
8.3	Método de Euler modificado	<b>343</b>
8.4	Método do ponto central	<b>346</b>
8.5	Métodos de Runge-Kutta	<b>347</b>
8.5.1	<i>Métodos de Runge-Kutta de segunda ordem</i>	<b>348</b>
8.5.2	<i>Métodos de Runge-Kutta de terceira ordem</i>	<b>352</b>
8.5.3	<i>Métodos de Runge-Kutta de quarta ordem</i>	<b>353</b>
8.6	Métodos multipasso	<b>359</b>
8.6.1	<i>Método de Adams-Bashforth</i>	<b>360</b>
8.6.2	<i>Método de Adams-Moulton</i>	<b>361</b>
8.7	Métodos preditor-corretor	<b>362</b>
8.8	Sistemas de equações diferenciais ordinárias de primeira ordem	<b>364</b>
8.8.1	<i>Solução de sistemas de EDOs de primeira ordem usando o método explícito de Euler</i>	<b>366</b>
8.8.2	<i>Solução de sistemas de EDOs de primeira ordem usando o método de Runge-Kutta de segunda ordem (versão método de Euler modificado)</i>	<b>346</b>
8.8.3	<i>Solução de sistemas de EDOs de primeira ordem usando o método de Runge-Kutta de quarta ordem clássico</i>	<b>373</b>
8.9	Solução de problemas de valor inicial de ordem mais elevada	<b>374</b>
8.10	Uso de funções residentes do MATLAB para resolver problemas de valor inicial	<b>379</b>
8.10.1	<i>Solução de uma única EDO de primeira ordem usando o MATLAB</i>	<b>380</b>
8.10.2	<i>Solução de sistemas de EDOs de primeira ordem usando o MATLAB</i>	<b>386</b>

8.11	Erro de truncamento local no método de Runge-Kutta de segunda ordem	<b>388</b>
8.12	Largura do passo de integração para se obter a precisão desejada	<b>390</b>
8.13	Estabilidade	<b>393</b>
8.14	Equações diferenciais ordinárias rígidas	<b>396</b>
8.15	Problemas	<b>398</b>

**Capítulo 9    *Equações diferenciais ordinárias: problemas de valor de contorno*** **407**

9.1	Fundamentos	<b>407</b>
9.2	O método do tiro	<b>410</b>
9.3	O método das diferenças finitas	<b>418</b>
9.4	Uso de funções residentes do MATLAB para resolver problemas de valor de contorno	<b>428</b>
9.5	Erro e estabilidade na solução numérica de problemas de valor de contorno	<b>433</b>
9.6	Problemas	<b>434</b>

**Apêndice A    *Introdução ao MATLAB*** **441**

A.1	Fundamentos	<b>441</b>
A.2	Começando com o MATLAB	<b>441</b>
A.3	Arranjos	<b>446</b>
A.4	Operações matemáticas com arranjos	<b>451</b>
A.5	Arquivos de programa	<b>455</b>
A.6	Arquivos de função	<b>458</b>
A.7	Programação no MATLAB	<b>460</b>
	<i>A.7.1 Operadores lógicos e relacionais</i>	<b>460</b>
	<i>A.7.2 Declarações condicionais, estruturas if-else-end</i>	<b>462</b>
	<i>A.7.3 Laços de repetição</i>	<b>464</b>
A.8	Traçado de gráficos	<b>465</b>
A.9	Problemas	<b>467</b>

**Apêndice B    *Programas de MATLAB*** **471****Índice** **475**

## Introdução

### Tópicos centrais

Representação de números em um computador Computadores e programação (1.4)  
(1.2)

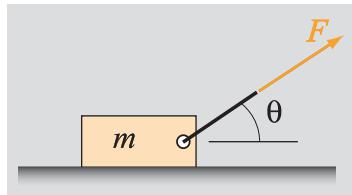
Erros em soluções numéricas, erros de arredondamento e erros de truncamento (1.3)

### 1.1 FUNDAMENTOS

Métodos numéricos são técnicas matemáticas usadas na solução de problemas matemáticos que não podem ser resolvidos ou que são difíceis de se resolver analiticamente. Uma solução analítica é uma resposta exata na forma de uma expressão matemática escrita em termos das variáveis associadas ao problema que está sendo resolvido. Uma solução numérica é um valor numérico aproximado para a solução (ou seja, um número). Embora soluções numéricas sejam uma aproximação, elas podem ser muito precisas. Em muitos métodos numéricos, os cálculos são executados de maneira iterativa até que a precisão desejada seja obtida.

Por exemplo, a Fig. 1-1 mostra um bloco com massa  $m$  sendo puxado por uma força  $F$  aplicada em um ângulo  $\theta$ . Utilizando as equações de equilíbrio, a relação entre a força e o ângulo é dada por:

$$F = \frac{\mu mg}{\cos \theta + \mu \sin \theta} \quad (1.1)$$



**Figura 1-1** Movimento de um bloco em uma superfície com atrito.

onde  $\mu$  é o coeficiente de atrito e  $g$  é a aceleração da gravidade. Para um dado valor de  $F$ , o ângulo necessário para mover o bloco pode ser determinado com a solução da Eq.

(1.1) em função de  $\theta$ . A Eq. (1.1), no entanto, não pode ser

resolvida analiticamente para  $\theta$ . Usando métodos numéricos, uma solução aproximada pode ser determinada com o grau de precisão especificado. Isso significa que, quando a solução numérica para  $\theta$  é substituída de volta na Eq. (1.1), o valor de  $F$  obtido a partir da expressão no lado direito da equação não é exatamente igual ao valor de  $F$ , embora possa ser bastante próximo.

Técnicas numéricas para resolver problemas matemáticos foram desenvolvidas e utilizadas centenas e mesmo milhares de anos atrás. A implementação dessas técnicas era difícil, já que os cálculos tinham que ser realizados manualmente ou com o emprego de dispositivos mecânicos simples. Isso limitava o número de cálculos que podiam ser realizados, além de reduzir a sua velocidade e a sua precisão. Hoje em dia, métodos numéricos são utilizados em rápidos computadores

digitais. Esses computadores permitem a execução de um grande número de cálculos tediosos e repetitivos em um curto espaço de tempo, produzindo soluções precisas, ainda que inexatas.

### **Resolvendo um problema na ciência e na engenharia**

O processo de solução de um problema na ciência e na engenharia é influenciado pelas ferramentas (métodos matemáticos) disponíveis para a solução desse problema. O processo pode ser dividido nos passos a seguir:

#### ***Declaração do problema***

A declaração define o problema. Ela dá a sua descrição, lista as variáveis envolvidas e identifica as restrições na forma de condições iniciais e/ou de contorno.

#### ***Formulação da solução***

A formulação da solução consiste no modelo (lei física, ou leis) que se usa para representar o problema e na dedução das equações a serem resolvidas. Exemplos de tais leis são as leis de Newton, a lei da conservação da massa e as leis da termodinâmica. Os modelos usados (escolhidos) para resolver o problema precisam ser consistentes com os métodos que serão usados em seguida na resolução das equações. Se o emprego de métodos analíticos é esperado para a solução, as equações que governam o problema devem permitir uma solução analítica. Se necessário, a formulação tem que ser simplificada para que as equações possam ser resolvidas analiticamente. Se são usados métodos numéricos na solução, os modelos e as equações podem ser mais complicados. Ainda assim, podem existir algumas limitações. Por exemplo, se a formulação é tal que uma solução numérica requer um longo tempo de processamento, pode ser necessário simplificar a formulação para que a solução seja obtida em um tempo razoável. Um exemplo disso é a previsão do tempo. O problema a ser resolvido é grande, e os métodos numéricos utilizados são muito complicados. A simulação numérica da previsão do tempo, contudo, não pode ir além do período no qual a previsão é necessária.

#### ***Programação (da solução numérica)***

Se o problema é resolvido numericamente, deve-se selecionar o método numérico a ser empregado. Para cada tipo de problema matemático, há várias (ou muitas) técnicas numéricas que podem ser usadas. As técnicas diferem em precisão, tempo de processamento e dificuldade de programação. Assim que um método numérico é selecionado, ele pode ser implementado em um computador. A implementação consiste em um **algoritmo**, que é um plano detalhado que descreve como o método numérico deve ser realizado, e em um programa, que é uma lista de comandos que permitem que o computador execute o algoritmo para obter a solução.

#### ***Interpretação da solução***

Já que soluções numéricas são uma aproximação (erros são discutidos na Seção 1.4), e como o programa de computador que executa o método numérico pode ter erros (ou *bugs*), uma solução numérica precisa ser examinada de perto. Isso

pode ser feito de várias formas, dependendo do problema. Por exemplo, se o método numérico é usado para resolver uma equação não-linear algébrica, a validade da solução pode ser verificada com a substituição da solução de volta na equação. Em problemas mais complicados, como a solução de uma equação diferencial, a solução numérica pode ser comparada com a solução conhecida de um problema similar, ou o problema pode ser resolvido várias vezes usando diferentes condições iniciais (ou de contorno) e diferentes métodos numéricos, com a verificação das diferenças subsequentes nas soluções.

Uma ilustração dos primeiros dois passos no processo de solução de um problema é mostrada no Exemplo 1-1.

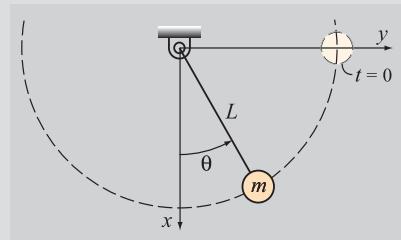
### Exemplo 1-1: Formulação do problema

Considere o seguinte enunciado:

Um pêndulo de massa  $m$  é preso a uma haste rígida com comprimento  $L$ , como mostra a figura. O pêndulo é deslocado da posição vertical de forma tal que o ângulo entre a haste e o eixo  $x$  seja  $\theta_0$ , e então solto. Formule o problema para determinar o ângulo  $\theta$  em função do tempo  $t$  após a soltura do pêndulo. Inclua na formulação uma força de amortecimento que seja proporcional à velocidade do pêndulo.

Formule a solução para dois casos:

(a)  $\theta_0 = 5^\circ$  e (b)  $\theta_0 = 90^\circ$



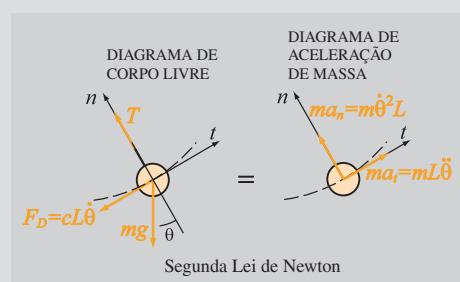
## SOLUÇÃO

### Lei física

A lei física utilizada para resolver o problema é a segunda lei de Newton da mecânica, que diz que, à medida que o pêndulo balança de um lado para o outro, a soma das forças que atuam na massa é igual à massa vezes a aceleração.

$$\sum \bar{F} = m\bar{a} \quad (1.2)$$

Isso pode ser visualizado a partir do desenho de um diagrama de massa livre e de um diagrama de aceleração de massa, que são mostrados ao lado. A constante  $c$  é o coeficiente de amortecimento. Deve ficar claro que a massa da haste é desprezada na presente solução.



### Equação que descreve o problema

A equação que descreve o problema é deduzida com a aplicação da segunda lei de Newton na direção tangencial:

$$\sum F_t = -cL \frac{d\theta}{dt} - mg \sin \theta = mL \frac{d^2\theta}{dt^2} \quad (1.3)$$

A Eq. (1.3), que é uma equação diferencial linear ordinária de segunda ordem, pode ser escrita na forma:

$$mL \frac{d^2\theta}{dt^2} + cL \frac{d\theta}{dt} + mg \sin \theta = 0 \quad (1.4)$$

As condições iniciais são obtidas com o conhecimento de que, quando o movimento do pêndulo se inicia ( $t = 0$ ), o pêndulo tem ângulo  $\theta_0$  e possui velocidade nula (se solto a partir do repouso):

$$\theta(0) = \theta_0 \quad \text{e} \quad \left. \frac{d\theta}{dt} \right|_{t=0} = 0 \quad (1.5)$$

### Método de solução

A Equação (1.4) é uma equação não-linear e não pode ser resolvida analiticamente. Entretanto, na letra (a), o deslocamento inicial do pêndulo é  $\theta_0 = 5^\circ$ . Logo, assim que é solto, o seu ângulo não ultrapassa  $5^\circ$  durante o processo oscilatório. Nesse caso, a Eq. (1.4) pode ser linearizada assumindo-se que  $\sin \theta \approx \theta$ . Com essa aproximação, a equação a ser resolvida é linear e possui solução analítica:

$$mL \frac{d^2\theta}{dt^2} - cL \frac{d\theta}{dt} - mg\theta = 0 \quad (1.6)$$

com as condições iniciais dadas na Eq. (1.5).

Na letra (b), o deslocamento inicial do pêndulo é  $\theta_0 = 90^\circ$  e a equação deve ser resolvida numericamente. Uma verdadeira solução para esse problema é mostrada no Exemplo 8-8.

## 1.2 REPRESENTAÇÃO DE NÚMEROS EM UM COMPUTADOR

### Representação decimal e binária

Números podem ser representados de várias formas. O familiar sistema decimal (base 10) usa dez algarismos 0, 1, ..., 9. Um número é escrito como uma seqüência de algarismos que correspondem a múltiplos de potências de 10. Como mostrado na Fig. 1-2, o primeiro algarismo à esquerda da vírgula decimal corresponde a  $10^0$ . O algarismo à esquerda deste corresponde a  $10^1$ , o algarismo à esquerda deste, a  $10^2$ , e assim por diante. Da mesma forma, o primeiro algarismo à direita da vírgula decimal corresponde a  $10^{-1}$ , o algarismo à direita deste, a  $10^{-2}$ , e assim por diante.

Em geral, no entanto, um número pode ser representado utilizando-se outras bases. Uma forma que pode ser facilmente implementada em computadores é o sistema binário (base 2). No sistema binário, um número é representado pelos algarismos 0 e 1. Portanto, um número é escrito como uma seqüência de zeros e uns que corresponda à multiplicação de potências de 2. O primeiro algarismo à esquerda da vírgula decimal corresponde a  $2^0$ . O algarismo à

$10^4$	$10^3$	$10^2$	$10^1$	$10^0$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$
↓	↓	↓	↓	↓	↓	↓	↓	↓
6	0	7	2	4 . 3	1	2	5	

$$6 \times 10^4 + 0 \times 10^3 + 7 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 + 3 \times 10^{-1} + 1 \times 10^{-2} + 2 \times 10^{-3} + 5 \times 10^{-4} = 60.724,3125$$

**Figura 1-2** Representação do número 60.724,3125 no sistema decimal (base 10).

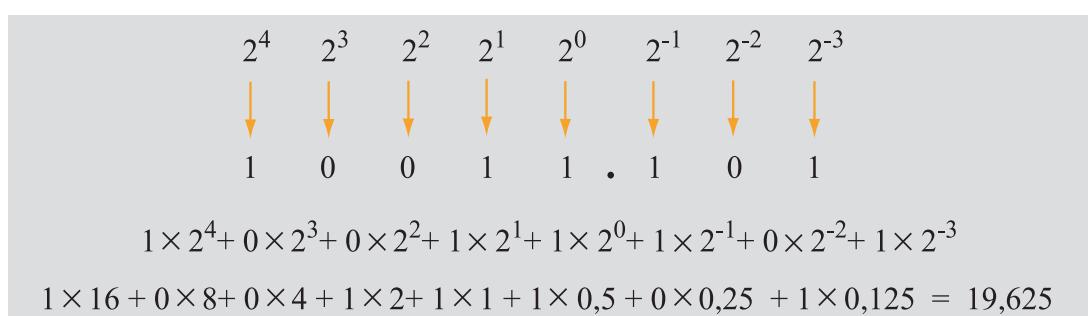
Base 10	Base 2			
	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

**Figura 1-3** Representação de números nas formas decimal e binária.

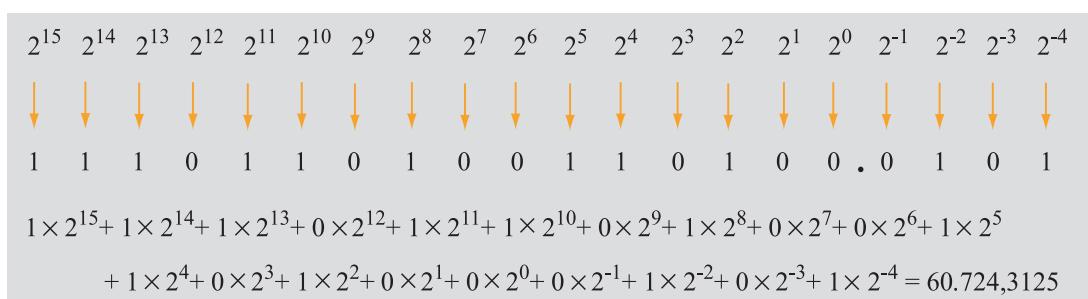
esquerda deste corresponde a 2<sup>1</sup>, o algarismo à sua esquerda corresponde a 2<sup>2</sup>, e assim por diante. Da mesma forma, o primeiro algarismo à direita da vírgula decimal corresponde a 2<sup>-1</sup>, o próximo algarismo à direita a 2<sup>-2</sup>, e assim por diante. Os primeiros dez algarismos (1, 2, 3,..., 10 na base 10) e a sua representação na base 2 são mostrados na Fig. 1-3. A representação do número 19,625 no sistema binário é mostrada na Fig. 1-4.

Outro exemplo é mostrado na Fig. 1-5, onde se escreve o número 60.724,3125 na forma binária.

Computadores armazenam e processam números na forma binária (base 2). Cada algarismo binário (um ou zero) é chamado de *bit* (termo que vem de *binary digit*). A aritmética binária é usada por computadores porque transistores modernos podem ser usados como chaves extremamente rápidas. Portanto, uma rede com tais transistores pode ser usada para representar cadeias de números, com o “1” se referindo à chave na posição “ligada”, e o “0” se referindo à posição “desligada”. Várias operações são então realizadas com base nessas seqüências de uns e zeros.



**Figura 1-4** Representação do número 19,625 no sistema binário (base 2).



**Figura 1-5** Representação do número 60.724,3125 no sistema binário (base 2).

### Representação em ponto flutuante

Para acomodar números grandes e pequenos, números reais são escritos na representação em ponto flutuante\*. A representação decimal em ponto flutuante (também chamada de notação científica) tem a forma:

$$d,dddddd \times 10^p \quad (1.7)$$

Nessa representação, um algarismo é escrito à esquerda da vírgula decimal, e o resto dos algarismos significativos é escrito à direita da vírgula. O número  $0,dddddd$  é chamado de *mantissa*. Seguem dois exemplos:

$$6519,23 \text{ é escrito como } 6,51923 \times 10^3$$

$$0,00000391 \text{ é escrito como } 3,91 \times 10^{-6}$$

A potência de 10,  $p$ , representa a ordem de grandeza do número, desde que o número que a precede seja menor que 5. Do contrário, diz-se que o número tem ordem  $p + 1$ . Logo, o número  $3,91 \times 10^{-6}$  é da ordem de  $10^{-6}$ ,  $O(10^{-6})$ , e o número  $6,51923 \times 10^3$  é da ordem de  $10^4$  (escrita como  $O(10^4)$ ).

A representação binária em ponto flutuante tem a forma:

$$1,bbbbbb \times 2^{bbb} \quad (b \text{ é um algarismo decimal}) \quad (1.8)$$

Nessa forma, a mantissa é  $0,bbbbbb$ , e a potência de 2 é chamada de *expoente*. Tanto a mantissa quanto o expoente são escritos na forma binária. A forma na Eq. (1.8) é obtida com a normalização do número (quando escrito na forma decimal) em relação à maior potência de 2 menor que o próprio número. Por exemplo, para escrever o número 50 na representação binária em ponto flutuante, o número é dividido (e multiplicado) por  $2^5 = 32$  (que é a maior potência de 2 menor que 50):

$$50 = \frac{50}{2^5} \times 2^5 = 1,5625 \times 2^5, \text{ que na forma binária é } 1,1001 \times 2^{101}$$

Mais dois exemplos são:

$$1344 = \frac{1344}{2^{10}} \times 2^{10} = 1,3125 \times 2^{10}, \text{ que na forma binária é } 1,0101 \times 2^{1001}$$

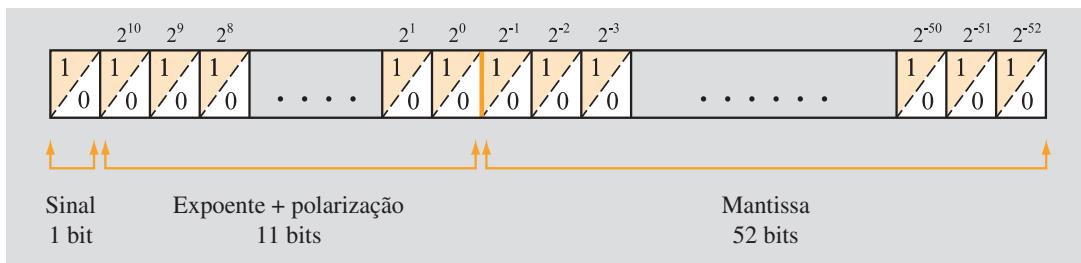
$$0,3125 = \frac{0,3125}{2^{-2}} \times 2^{-2} = 1,25 \times 2^{-2}, \text{ que na forma binária é } 1,01 \times 2^{-10}$$

### Armazenando um número na memória do computador

Uma vez colocado na representação binária em ponto flutuante, o número é armazenado no computador. O computador armazena os valores do expoente e da mantissa separadamente, não sendo armazenado o primeiro 1 à frente da vírgula

\* N. de T.: É necessário chamar a atenção, neste momento, para o uso do termo ponto flutuante. A tradução mais adequada para *floating point* talvez fosse vírgula flutuante. Isto porque, no Brasil, a vírgula é utilizada como separador decimal. A manutenção do uso do termo ponto flutuante neste texto segue a tendência verificada na literatura especializada no Brasil. Também vale a pena destacar que vários exemplos e problemas utilizando o programa MATLAB são apresentados ao longo do texto. O leitor deve estar atento ao fato de que tal programa utiliza o padrão da língua inglesa para representar números. Com isso, sempre que forem descritos programas, entradas de dados e resultados de simulação relacionados ao MATLAB, os números que porventura aparecerem serão escritos tendo o ponto como separador decimal, e não a vírgula.

decimal. Conforme já mencionado, um bit é um algarismo binário. A memória do computador é organizada em *bytes*, onde cada byte corresponde a 8 bits. De acordo com a norma IEEE<sup>1</sup>-754, computadores armazenam números e realizam cálculos em *precisão simples*<sup>2</sup> ou em *precisão dupla*<sup>3</sup>. Na precisão simples, os números são armazenados em uma cadeia de 32 bits (4 bytes), e, na precisão dupla, em uma cadeia de 64 bits (8 bytes). Em ambos os casos, o primeiro bit armazena o sinal do número (0 corresponde a + e 1 corresponde a -). Os próximos 8 bits na precisão simples (11 na precisão dupla) são usados para armazenar o expoente. Os 23 bits seguintes na precisão simples (52 bits na precisão dupla) são usados para armazenar a mantissa. Isso é ilustrado para a precisão dupla na Fig. 1-6.

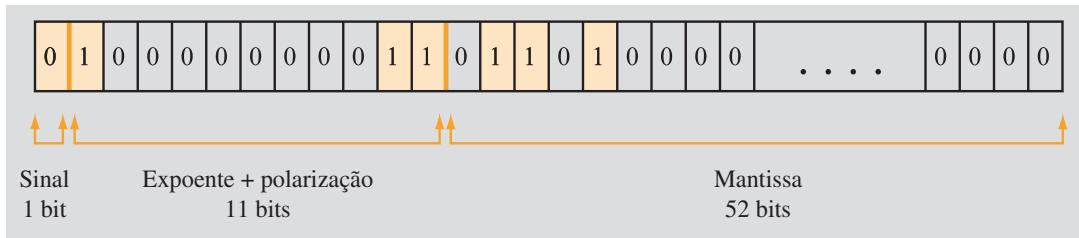


**Figura 1-6** Armazenando em precisão dupla um número escrito na representação binária em ponto flutuante.

O valor da mantissa é fornecido na forma binária. Ao valor do expoente é acrescida uma polarização (*bias*). A polarização corresponde à adição de uma constante ao valor do expoente. A polarização é introduzida para se evitar o uso de um dos bits para representar o sinal do expoente (já que o expoente pode ser positivo ou negativo). Na notação binária, o maior número que pode ser escrito com 11 bits é 2047 (quando todos os 11 dígitos são iguais a 1). A polarização utilizada é 1023. Isso significa que, se, por exemplo, o expoente for 4, então o valor armazenado é  $4 + 1023 = 1027$ . Logo, o menor expoente que pode ser armazenado pelo computador é  $-1023$  (que é armazenado como 0), e o maior é 1024 (que é armazenado como 2047). Na precisão simples, 8 bits são alocados para o valor do expoente, e a polarização é 127.

1. A sigla IEEE representa *Institute of Electrical and Electronics Engineers* (Instituto de Engenheiros Eletricistas e Eletrônicos).
2. O termo *precisão* se refere ao número de algarismos significativos de um número real que podem ser armazenados em um computador. Por exemplo, o número  $1/3 = 0,333333\dots$  só pode ser representado no computador em uma forma cortada ou arredondada utilizando um número finito de algarismos binários, pois a quantidade de memória disponível para armazenar esses bits é finita. Quanto mais algarismos armazenados no lado direito da vírgula decimal, mais precisa é a representação de um número real no computador.
3. Esse nome é meio enganador. A precisão de um número escrito em precisão dupla não é realmente o dobro daquela de um número escrito em precisão simples. Ao invés disso, “dupla”, no termo precisão dupla, refere-se ao fato de duas vezes mais bits serem usados (*64 versus 32*) para representar um número real neste caso do que na representação em precisão simples.

Como um exemplo, considere o armazenamento do número 22,5 em precisão dupla de acordo com a norma IEEE-754. Primeiramente, o número é normalizado,  $(22,5/2^4) \times 2^4 = 1,40625 \times 2^4$ . Em precisão dupla, o expoente com a polarização é  $4 + 1023 = 1027$ , que é armazenado na forma binária como 10000000011. A mantissa é 0,40625, que é armazenada na forma binária como 0,01101000....000. O armazenamento do número é ilustrado na Fig. 1-7.



**Figura 1-7** Armazenamento do número 22,5 em precisão dupla de acordo com a norma IEEE-754.

### Notas adicionais

- O menor número positivo que pode ser expressado em precisão dupla é:

$$2^{-1023} \approx 1,1 \times 10^{-308}$$

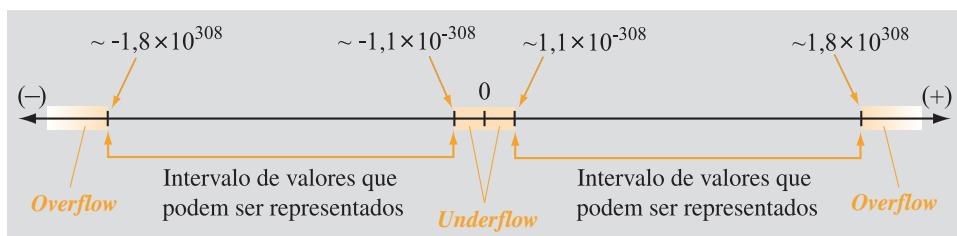
Isso significa que há um (pequeno) intervalo entre zero e o menor número que pode ser armazenado em um computador. Tentativas de se definir um número dentro desse intervalo causam um erro de *underflow* (da mesma forma, o número negativo mais próximo de zero é  $-1,1 \times 10^{-308}$ ).

- O maior número positivo que pode ser expressado em precisão dupla é aproximadamente:

$$2^{1024} \approx 1,8 \times 10^{308}$$

Tentativas de se definir um número maior que esse causam um erro de *overflow* (o mesmo se aplica a números menores que  $-2^{1024}$ ).

O intervalo de números que podem ser representados em precisão dupla é mostrado na Fig. 1-8.



**Figura 1-8** Intervalo de números que podem ser representados em precisão dupla.

- Como se usa um número finito de bits, nem todo número pode ser escrito de forma precisa no formato binário. Em outras palavras, apenas um número finito de valores exatos na forma decimal pode ser armazenado na forma binária. Por exemplo, o número 0,1 não pode ser representado exatamente no formato binário finito quando se usa a precisão simples. Para se escrever 0,1 na representação binária em ponto flutuante, é feita a normalização desse número:  $0,1 = 1,6 \times 2^{-4}$ . O expoente -4 (com uma polarização) pode ser armazenado de forma exata, mas a mantissa, 0,6, não pode ser escrita de forma exata no formato binário com 23 bits. Além disso, números irracionais não podem ser representados de forma exata em nenhum formato. Isso significa que, em muitos casos, os valores exatos devem ser aproximados. Os erros introduzidos são pequenos em um passo como esse, mas quando muitas operações são executadas, os erros podem crescer de tal forma que a resposta final pode ser afetada. Esses e outros erros são discutidos na próxima seção.
- O intervalo entre números que podem ser representados depende de sua ordem de grandeza. Na precisão dupla, a menor mantissa que pode ser armazenada é igual a  $2^{-52} \approx 2,22 \times 10^{-16}$ . Esta também é a menor diferença possível entre a mantissa de dois números. A ordem de grandeza do número real associado a essa mantissa, no entanto, depende do expoente. Para números da ordem de 1, a menor diferença que pode ser representada entre dois números é então  $2,22 \times 10^{-16}$ . Esse valor também é definido como o **épsilon da máquina** em precisão dupla. No MATLAB, esse valor é atribuído à variável predefinida `eps`. Conforme mostrado a seguir, quando o nome da variável `eps` é digitado na janela de comando, o seu valor é mostrado.

```
>> eps  
ans =  
2.220446049250313e-016
```

### 1.3 ERROS EM SOLUÇÕES NUMÉRICAS

Soluções numéricas podem ser muito precisas, mas em geral são inexatas. Dois tipos de erros são introduzidos quando métodos numéricos são usados na solução de um problema. Um deles, mencionado na seção anterior, ocorre em função da maneira pela qual computadores digitais armazenam números e executam operações numéricas. Estes são chamados de **erros de arredondamento**. O segundo tipo de erro é introduzido pelo método numérico usado na solução. Estes são chamados de **erros de truncamento**. Métodos numéricos usam aproximações para resolver problemas. Os erros introduzidos por essas aproximações são os erros de truncamento. Juntos, os dois erros constituem o **erro total** da solução numérica, que é a diferença (que pode ser definida de várias formas) entre a solução verdadeira (exata, e que é usualmente desconhecida) e a solução numérica aproximada. Os erros de arredondamento, de truncamento e total são abordados a seguir.

### 1.3.1 Erros de arredondamento

Os números são representados em um computador através de um número finito de bits (veja Seção 1.2). Conseqüentemente, números reais que têm uma mantissa mais longa do que o número de bits disponíveis para representá-los têm que ser encurtados. Esse requisito se aplica aos números irracionais, que devem ser encurtados em qualquer sistema, aos números finitos que são muito longos e aos números finitos na forma decimal que não podem ser representados de forma exata na forma binária. Um número pode ser encurtado seja *cortando*, ou descartando, os algarismos a mais, ou fazendo-se um *arredondamento*. No corte, os algarismos na mantissa além do comprimento que pode ser armazenado são simplesmente deixados de fora. No arredondamento, o último algarismo armazenado é arredondado.

Como uma simples ilustração, considere o número  $2/3$  (por simplicidade, é usado o formato decimal). Na forma decimal com quatro algarismos significativos,  $2/3$  pode ser escrito como 0,6666 ou 0,6667. No primeiro caso, o número verdadeiro foi cortado, enquanto no último o número verdadeiro foi arredondado. De qualquer maneira, o corte ou o arredondamento de números reais leva a erros nos cálculos numéricos, especialmente quando muitas operações são realizadas. Esse tipo de erro numérico (seja ele devido ao corte ou ao arredondamento) é conhecido como erro de arredondamento. O Exemplo 1-2 mostra a diferença entre corte e arredondamento.

#### Exemplo 1-2: Erros de arredondamento

Considere dois números quase iguais,  $p = 9890,9$  e  $q = 9887,1$ . Use a representação decimal em ponto flutuante (notação científica) com três algarismos significativos na mantissa para calcular a diferença desses dois números,  $(p - q)$ . Calcule-a primeiro usando o corte e depois o arredondamento.

#### SOLUÇÃO

Na representação decimal em ponto flutuante, os dois números são:

$$p = 9,8909 \times 10^3 \text{ e } q = 9,8871 \times 10^3$$

Se apenas três algarismos significativos são permitidos na mantissa, os números têm que ser encurtados. Se o corte é usado, os números se tornam:

$$p = 9,890 \times 10^3 \text{ e } q = 9,887 \times 10^3$$

O uso desses valores na subtração fornece:

$$q = 9,890 \times 10^3 - 9,887 \times 10^3 = 0,003 \times 10^3 = 3$$

Se o arredondamento é usado, os números se tornam:

$$p = 9,891 \times 10^3 \text{ e } q = 9,887 \times 10^3 \text{ (} q \text{ é o mesmo de antes)}$$

O uso desses valores na subtração fornece:

$$q = 9,891 \times 10^3 - 9,887 \times 10^3 = 0,004 \times 10^3 = 4$$

A diferença verdadeira (exata) entre os números é de 3,8. Esses resultados mostram que, no presente problema, o arredondamento leva a um valor mais próximo à resposta verdadeira.

A ordem de grandeza dos erros de arredondamento depende da ordem de grandeza dos números envolvidos, já que, conforme explicado na seção anterior, o intervalo entre os números que podem ser representados em um computador depende de sua grandeza. Erros de arredondamento normalmente ocorrem quando os números envolvidos nos cálculos diferem significativamente em magnitude e quando dois números que são quase idênticos são subtraídos um do outro.

Por exemplo, considere a equação quadrática:

$$x^2 - 100,0001x + 0,01 = 0 \quad (1.9)$$

para a qual as soluções exatas são  $x_1 = 100$  e  $x_2 = 0,0001$ . As soluções podem ser calculadas com a fórmula quadrática:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{e} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (1.10)$$

O uso da janela de comandos do MATLAB para calcular  $x_1$  e  $x_2$  fornece:

```
>> format long
>> a = 1; b = -100.0001; c = 0.01;
>> RaizesDis = sqrt(b^2 - 4*a*c)
RaizesDis =
99.999900000000000
```

```
>> x1 = (-b + RaizesDis)/(2*a)
x1 =
100
>> x2 = (-b - RaizesDis)/(2*a)
x2 =
1.0000000003197e-004
```

O valor calculado no MATLAB para  $x_2$  não é exato devido a erros de arredondamento. Tais erros ocorrem no numerador da expressão de  $x_2$ . Como  $b$  é negativo, o numerador envolve a subtração de dois números que são quase iguais.

Em muitos casos, a forma das expressões matemáticas que contêm a subtração de duas grandezas que são quase iguais pode ser mudada para uma forma diferente, menos propensa a erros de arredondamento. Na expressão de  $x_2$  na Eq. (1.10), isso pode ser feito com a multiplicação por  $(-b + \sqrt{b^2 - 4ac})/(-b + \sqrt{b^2 - 4ac})$ :

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \cdot \frac{(-b + \sqrt{b^2 - 4ac})}{(-b + \sqrt{b^2 - 4ac})} = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \quad (1.11)$$

O cálculo de  $x_2$  no MATLAB com a Eq. (1.11) fornece:

```
>> x2Mod = (2*c)/(-b+RaizesDis)
x2Mod =
1.000000000000000e-004
```

Agora, o valor calculado para  $x_2$  não contém erros. Outro exemplo de erros de arredondamento é mostrado no Exemplo 1-3.

### Exemplo 1-3: Erros de arredondamento

Considere a função:

$$f(x) = x(\sqrt{x} - \sqrt{x-1}) \quad (1.12)$$

- (a) Use o MATLAB para calcular o valor de  $f(x)$  para os três valores de  $x$  a seguir:  $x = 10$ ,  $x = 1000$  e  $x = 100000$ .
- (b) Use o formato decimal com seis algarismos significativos para calcular  $f(x)$  para os valores de  $x$  indicados na letra (a). Compare os resultados com os valores obtidos na letra (a).
- (c) Mude a forma de  $f(x)$  multiplicando essa função por  $\frac{\sqrt{x} + \sqrt{x-1}}{\sqrt{x} + \sqrt{x-1}}$ . Usando a nova forma com números no formato decimal e seis algarismos significativos, calcule o valor de  $f(x)$  para os três valores de  $x$ . Compare os resultados com aqueles obtidos na letra (a).

### SOLUÇÃO

(a)

```
>> format long g
>> x = [10 1000 100000];
>> Fx = x.*(sqrt(x) - sqrt(x-1))
Fx =
    1.6227766016838      15.8153431255776      158.114278298171
```

- (b) O uso do formato decimal com seis algarismos significativos na Eq. (1.12) fornece os seguintes valores para  $f(x)$ :

$$f(10) = 10(\sqrt{10} - \sqrt{10-1}) = 10(3,16228 - 3) = 1,62280$$

Esse valor concorda com o valor obtido na letra (a), quando se faz o arredondamento para seis algarismos significativos.

$$f(1000) = 1000(\sqrt{1000} - \sqrt{1000-1}) = 1000(31,6228 - 31,6070) = 15,8$$

Quando arredondado para seis algarismos significativos, o valor obtido na letra (a) é 15,8153.

$$f(100000) = 100000(\sqrt{100000} - \sqrt{100000-1}) = 100000(316,228 - 316,226) = 200$$

Quando arredondado para seis algarismos significativos, o valor obtido na letra (a) é 158,114. Esses resultados mostram que o erro de arredondamento causado pelo uso de seis algarismos significativos aumenta à medida que o valor de  $x$  aumenta e a diferença relativa entre  $\sqrt{x}$  e  $\sqrt{x-1}$  diminui.

- (c) Multiplicando o lado direito da Eq. (1.12) por  $\frac{\sqrt{x} + \sqrt{x-1}}{\sqrt{x} + \sqrt{x-1}}$ , tem-se:

$$f(x) = x(\sqrt{x} - \sqrt{x-1}) \frac{\sqrt{x} + \sqrt{x-1}}{\sqrt{x} + \sqrt{x-1}} = \frac{x[x - (x-1)]}{\sqrt{x} + \sqrt{x-1}} = \frac{x}{\sqrt{x} + \sqrt{x-1}} \quad (1.13)$$

O cálculo de  $f(x)$  usando a Eq. (1.13) para  $x = 10$ ,  $x = 1000$  e  $x = 100000$  resulta em:

$$f(10) = \frac{10}{\sqrt{10} + \sqrt{10 - 1}} = \frac{10}{3,16228 + 3} = 1,62278$$

$$f(1000) = \frac{1000}{\sqrt{1000} + \sqrt{1000 - 1}} = \frac{1000}{31,6228 + 31,6070} = 15,8153$$

$$f(100000x) = \frac{100000}{\sqrt{100000} + \sqrt{100000 - 1}} = \frac{100000}{316,228 + 316,226} = 158,114$$

Agora, os valores de  $f(x)$  são os mesmos calculados na letra (a).

### 1.3.2 Erros de truncamento

Os erros de truncamento ocorrem quando os métodos numéricos usados na solução de um problema matemático adotam um procedimento matemático aproximado. Um exemplo simples é a avaliação numérica de  $\sin(x)$ , que pode ser feita a partir da expansão em série de Taylor (a série de Taylor é revisada no Capítulo 2):

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots \quad (1.14)$$

O valor de  $\sin\left(\frac{\pi}{6}\right)$  pode ser determinado de forma exata com a Eq. (1.14) se um número infinito de termos for usado. O seu valor pode ser aproximado com o uso de apenas um número finito de termos. A diferença entre o valor verdadeiro (exato) e o valor aproximado é o erro de truncamento, denotado por  $E^{TR}$ . Por exemplo, se apenas o primeiro termo for usado:

$$\sin\left(\frac{\pi}{6}\right) = \frac{\pi}{6} = 0,5235988 \quad E^{TR} = 0,5 - 0,5235988 = -0,0235988$$

Se dois termos da série de Taylor forem usados:

$$\sin\left(\frac{\pi}{6}\right) = \frac{\pi}{6} - \frac{(\pi/6)^3}{3!} = 0,4996742$$

$$E^{TR} = 0,5 - 0,4996742 = 0,0003258$$

Outro exemplo de erro de truncamento com o qual o leitor está provavelmente familiarizado é o cálculo aproximado de derivadas. O valor da derivada de uma função  $f(x)$  em um ponto  $x_1$  pode ser aproximado pela expressão:

$$\left. \frac{df(x)}{dx} \right|_{x=x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (1.15)$$

onde  $x_2$  é um ponto próximo a  $x_1$ . A diferença entre o valor da derivada real e o valor calculado com a Eq. (1.15) é chamada de **erro de truncamento**.

O erro de truncamento é dependente do método numérico específico ou do algoritmo usado na solução do problema. Detalhes a respeito do erro de truncamento são discutidos ao longo do livro à medida que se apresentam os diversos métodos numéricos. O erro de truncamento é independente do erro de arredondamento; ele existe mesmo quando as operações matemáticas são exatas.

### 1.3.3 Erro total

A solução numérica é uma aproximação. Ela sempre inclui erros de arredondamento e, dependendo do método numérico utilizado, também pode incluir erros de truncamento. Juntos, os erros de arredondamento e de truncamento resultam no erro numérico total incluído na solução numérica. Esse erro total, também chamado de erro real, é a diferença entre a solução verdadeira (exata) e a solução numérica:

$$\text{ErroReal} = \text{SoluçãoExata} - \text{SoluçãoNumérica} \quad (1.16)$$

O valor absoluto da razão entre o erro real e a solução exata é chamado de erro relativo real:

$$\text{ErroRelativoReal} = \left| \frac{\text{SoluçãoExata} - \text{Solução Numérica}}{\text{SoluçãoExata}} \right| \quad (1.17)$$

Essa grandeza adimensional e independente de escalas indica quanto grande é o erro em relação à solução exata.

O erro real e o erro relativo real das Eqs. (1.16) e (1.17) não podem ser de fato determinados em problemas cuja solução requer o uso de métodos numéricos, já que a solução verdadeira não é conhecida. Contudo, essas grandezas podem ser úteis na verificação da precisão de diferentes métodos numéricos. Isso é feito com o emprego do método numérico na solução de problemas que têm solução analítica, avaliando-se com isso os erros reais.

Como os erros reais não podem ser calculados na maioria dos casos, a precisão de uma solução numérica pode ser avaliada de diferentes maneiras. Isso depende do método específico e é discutido em maior detalhe nos capítulos seguintes. Em alguns métodos, o erro numérico pode ser colocado em limites conhecidos; outros métodos permitem uma estimativa da ordem de grandeza do erro. Em aplicações práticas, soluções numéricas também podem ser comparadas com resultados experimentais, mas é importante lembrar que dados experimentais também apresentam erros e incertezas.

## 1.4 COMPUTADORES E PROGRAMAÇÃO

Conforme mencionado na Seção 1.1, os fundamentos dos métodos numéricos aplicáveis à solução de problemas matemáticos sem solução analítica foram desenvolvidos e já utilizados muitos anos atrás. A introdução dos computadores digitais modernos melhorou a precisão desses métodos e permitiu a sua aplicação em problemas que requerem um grande número de cálculos repetitivos. Um computador pode armazenar uma grande quantidade de números e pode executar operações de forma muito rápida. Para realizar os cálculos necessários para a implementação de um método numérico específico, deve-se fornecer ao computador uma série de instruções, que se chama de programa. Como o formato binário é usado nas operações matemáticas e no armazenamento de números, as instruções devem ser escritas nessa forma, requerendo o que é chamado de linguagem de máquina. Nos primeiros dias dos computadores, programas eram escritos em

linguagens computacionais de baixo nível (uma linguagem chamada de *assembler*). A programação dessa forma era tediosa e propensa a erros, porque tinha que ser muito detalhada, além de ser feita de uma forma muito diferente daquela usada na matemática do dia-a-dia.

Mais tarde, foram introduzidos os sistemas operacionais. Tais sistemas podem ser vistos como interfaces ou cascas que permitem o contato e a comunicação mais fácil entre humanos e a linguagem de máquina do computador. As instruções escritas na linguagem do sistema operacional são convertidas por esse sistema para comandos em linguagem de máquina que são executados pelo computador. Exemplos de sistemas operacionais são o Unix (escrito em uma linguagem de programação chamada de C), desenvolvido pela Bell Laboratories nos anos setenta, e o DOS (Disk Operating System), usado pela Microsoft Inc. Embora os sistemas operacionais simplifiquem a comunicação com o computador, seu uso ainda é relativamente difícil, requerendo longos códigos. Além disso, eles não são escritos para as necessidades especiais de engenheiros e cientistas.

Programas de computador usados por cientistas e engenheiros são freqüentemente escritos em linguagens de programação que operam no topo do sistema operacional. Essas linguagens computacionais de nível mais alto são mais fáceis de usar e permitem que o engenheiro ou cientista se concentre no problema ao invés de perder tempo com uma programação tediosa. Fortran, C e C++ são linguagens computacionais freqüentemente usadas na ciência e na engenharia. Em geral, para uma mesma tarefa, programas de computador escritos em linguagens computacionais de alto nível são mais curtos (requerem menos comandos) do que programas escritos em linguagens de baixo nível. Este livro usa o MATLAB, que é uma linguagem de alto nível para a computação técnica. Por exemplo, a multiplicação de duas matrizes no MATLAB é feita com a aplicação da operação de multiplicação, enquanto outras linguagens requerem a escrita de um *loop* de repetição com várias linhas de código.

## Algoritmo

Quando um computador é usado para fornecer a solução numérica de um problema, um programa computacional concebido pelo usuário realiza as operações associadas ao método numérico específico utilizado. Alguns métodos numéricos podem ser facilmente implementados, mas às vezes é possível encontrar procedimentos numéricos complicados e difíceis de se programar.

Antes de fazer a programação de um método numérico, é útil planejar todos os passos que têm que ser seguidos para que esse método possa ser implementado de forma bem-sucedida. Tal plano é chamado de *algoritmo*, que é definido como um conjunto de instruções passo a passo descrevendo como a solução pode ser obtida. Algoritmos podem ser escritos em vários níveis de detalhe. Como os métodos numéricos são o foco deste livro, o termo *algoritmo* é usado aqui apenas no contexto das instruções necessárias para a implementação desses métodos.

Como um exemplo simples, considere um algoritmo para a solução da equação quadrática:

$$ax^2 + bx + c = 0 \quad (1.18)$$

cuja solução no caso de raízes reais é dada pela fórmula quadrática:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad (1.19)$$

### Algoritmo para se obter as raízes reais de uma equação quadrática

São dadas as três constantes da equação quadrática,  $a$ ,  $b$  e  $c$ .

1. Calcule o valor do discriminante  $D = b^2 - 4ac$ .
2. Se  $D \geq 0$ , calcule as duas raízes usando a Eq. (1.19).
3. Se  $D = 0$ , calcule a raiz:  $x = \frac{-b}{2a}$  e mostre a mensagem: “A equação possui raiz única”.
4. Se  $D < 0$ , mostre a mensagem: “A equação não possui raízes reais”.

Uma vez planejado, o algoritmo pode ser implementado em um programa de computador.

### Programas de computador

Um programa de computador (código) é um conjunto (lista) de comandos (operações) que devem ser executados pelo computador. Diferentes linguagens de programação usam diferentes sintaxes para os comandos, mas, em geral, os comandos podem ser agrupados em várias categorias:

- Comandos de entrada e saída de dados. Esses comandos são usados para importar dados para o computador, mostrá-los no monitor, ou para armazenar resultados numéricos em arquivos.
- Comandos de definição de variáveis.
- Comandos que executam operações matemáticas. Estes incluem operações padrão (adição, multiplicação, potenciação, etc.) e comandos que calculam os valores de funções comumente usadas (trigonométricas, exponenciais, logarítmicas, etc.).
- Comandos que controlam a ordem de execução de comandos e que permitem ao computador executar ou não certos grupos de comandos em diferentes circunstâncias. Esses comandos estão tipicamente associados a declarações condicionais que permitem a tomada de decisões, como por exemplo a ordem em que determinados comandos devem ser executados. Muitas linguagens têm comandos “se-senão” para este propósito, mas muitos outros comandos existem com a mesma finalidade.
- Comandos que permitem ao computador repetir seções do programa. Em muitas linguagens, esses comandos são chamados de *loops*. Eles são muito úteis para a programação de métodos numéricos, já que muitos desses métodos usam iterações para obter soluções precisas.
- Comandos que criam figuras e gráficos mostrando resultados.

Um programa de computador pode ser escrito como uma longa lista de comandos, mas em geral é dividido em partes menores e bem definidas

(subprogramas). Essas partes são programas auto-suficientes que conduzem uma parcela do total de operações a serem realizadas. Com essa abordagem, as várias partes podem ser escritas e testadas de forma independente. Em muitas linguagens de computador, os subprogramas são chamados de sub-rotinas ou funções.

Conforme já mencionado, métodos numéricos são implementados neste livro usando o MATLAB, que é uma linguagem relativamente nova para a computação técnica. O MATLAB é poderoso e fácil de usar. Ele contém muitas funções residentes que são extremamente úteis na resolução de problemas na ciência e na engenharia.

Assume-se que o leitor deste livro tenha pelo menos algum conhecimento de MATLAB e de programação. Para aqueles que não o têm, uma introdução ao MATLAB é apresentada no Apêndice. Ela inclui uma seção sobre declarações condicionais e *loops*, que são os blocos construtivos básicos da programação. Para uma introdução mais detalhada à programação, o leitor deve recorrer a livros dedicados à programação de computadores. Para ajudar o leitor a seguir os programas de MATLAB listados neste livro, comentários e explicações são colocados logo após os programas.

## 1.5 PROBLEMAS

### Problemas de solução manual

*Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos.*

- 1.1** Converta o número binário 1011101 para o formato decimal.
- 1.2** Converta o número binário 11000101.101 para o formato decimal.
- 1.3** Converta o número binário 10010101110001.01110101 para o formato decimal.
- 1.4** Escreva o número 81 nos seguintes formatos (na letra *(c)*, siga a norma IEEE-754): *(a)* Formato binário; *(b)* Representação em ponto flutuante na base 2. *(c)* Cadeia de 32 bits em precisão simples.
- 1.5** Escreva o número 66,25 nos seguintes formatos (na letra *(c)*, siga a norma IEEE-754): *(a)* Formato binário; *(b)* Representação em ponto flutuante na base 2. *(c)* Cadeia de 32 bits em precisão simples.
- 1.6** Escreva o número -0,625 nos seguintes formatos (na letra *(c)*, siga a norma IEEE-754): *(a)* Formato binário; *(b)* Representação em ponto flutuante na base 2. *(c)* Cadeia de 32 bits em precisão simples.
- 1.7** Escreva o número 0,533203125 nos seguintes formatos (na letra *(c)*, siga a norma IEEE-754): *(a)* Formato binário; *(b)* Representação em ponto flutuante na base 2. *(c)* Cadeia de 32 bits em precisão simples.

- 1.8** Escreva o número 256,1875 nos seguintes formatos (na letra (c), siga a norma IEEE–754): (a) Formato binário; (b) Representação em ponto flutuante na base 2. (c) Cadeia de 64 bits em precisão dupla.
- 1.9** Escreva o número –30952 nos seguintes formatos (na letra (c), siga a norma IEEE–754): (a) Formato binário; (b) Representação em ponto flutuante na base 2. (c) Cadeia de 64 bits em precisão dupla.
- 1.10** Escreva o número 0,33203125 nos seguintes formatos (na letra (c), siga a norma IEEE–754): (a) Formato binário; (b) Representação em ponto flutuante na base 2. (c) Cadeia de 64 bits em precisão dupla.
- 1.11** Escreva o número 0,001220703125 nos seguintes formatos (na letra (c), siga a norma IEEE–754): (a) Formato binário; (b) Representação em ponto flutuante na base 2. (c) Cadeia de 64 bits em precisão dupla.
- 1.12** Escreva o número 0,2 no formato binário usando um número suficiente de algarismos de forma que o erro relativo real seja menor que 0,005.
- 1.13** Considere a função  $f(x) = \frac{1 - \cos(x)}{\sin(x)}$ .
- Use o formato decimal com seis algarismos significativos (aplique arredondamento) para calcular, usando uma calculadora,  $f(x)$  para  $x = 0,007$ .
  - Use o MATLAB (usando `format long`) para calcular o valor de  $f(x)$  e o erro relativo real causado pelo arredondamento no valor de  $f(x)$  obtido na letra (a).
  - Multiplique  $f(x)$  por  $\frac{1 + \cos(x)}{1 + \cos(x)}$  para obter uma forma de  $f(x)$  menos propensa a erros de arredondamento. Com essa nova forma, use o formato decimal com seis algarismos significativos (aplique arredondamento) para calcular, usando uma calculadora,  $f(x)$  para  $x = 0,007$ . Compare o valor obtido com aqueles obtidos nas letras (a) e (b).
- 1.14** Considere a função  $f(x) = \frac{\sqrt{9+x} - 3}{x}$ .
- Use o formato decimal com seis algarismos significativos (aplique arredondamento) para calcular, usando uma calculadora,  $f(x)$  para  $x = 0,005$ .
  - Use o MATLAB (usando `format long`) para calcular o valor de  $f(x)$  e o erro relativo real causado pelo arredondamento no valor de  $f(x)$  obtido na letra (a).
  - Multiplique  $f(x)$  por  $\frac{\sqrt{9+x} + 3}{\sqrt{9+x} + 3}$  para obter uma forma de  $f(x)$  menos propensa a erros de arredondamento. Com essa nova forma, use o formato decimal com seis algarismos significativos (aplique arredondamento) para calcular, usando uma calculadora,  $f(x)$  para  $x = 0,005$ . Compare o valor obtido com aqueles obtidos nas letras (a) e (b).
- 1.15** Considere a função  $f(x) = \frac{e^x - 1}{x}$ .
- Use o formato decimal com seis algarismos significativos (aplique arredondamento) para calcular, usando uma calculadora,  $f(x)$  para  $x = 0,00275$ .

- (b) Use o MATLAB (usando `format long`) para calcular o valor de  $f(x)$  e o erro relativo real causado pelo arredondamento no valor de  $f(x)$  obtido na letra (a).

**1.16** A expansão em série de Taylor de  $\cos(x)$  é dada por:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \frac{x^{10}}{10!} + \dots \quad (1.20)$$

Use os três primeiros termos na Eq. (1.20) para calcular o valor de  $\cos\left(\frac{\pi}{3}\right)$ . Calcule o erro de truncamento. Use o formato decimal com seis algarismos significativos (aplique arredondamento).

**1.17** A expansão em série de Taylor da função  $f(x) = e^x$  é:

$$f(x) = e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots \quad (1.21)$$

Use a Eq. (1.21) para calcular o valor de  $e^{-2}$  para os seguintes casos. Em cada caso também calcule o erro relativo real (use `format long` no MATLAB para calcular o valor verdadeiro de  $e^{-2}$ ). Use números decimais com seis algarismos significativos (aplique arredondamento).

- (a) Use os quatro primeiros termos. (b) Use os seis primeiros termos. (c) Use os oito primeiros termos.

**1.18** Use os sete primeiros termos na Eq. (1.21) para calcular um valor estimado para  $e$ . Realize as contas no MATLAB (use `format long` para mostrar os números). Determine o erro relativo real. Para o valor exato de  $e$ , use `exp(1)` no MATLAB.

**1.19** Desenvolva um algoritmo para determinar se um dado número inteiro é um número primo.

**1.20** Desenvolva um algoritmo para somar todos os números primos entre 0 e um dado número.

**1.21** Desenvolva um algoritmo para converter inteiros dados na forma decimal para o formato binário.

### Problemas de programação no MATLAB

*Resolva os seguintes problemas usando o ambiente MATLAB. Não use funções residentes do MATLAB para mudar o formato dos números.*

**1.22** Escreva um programa que implemente o algoritmo desenvolvido no Problema 1.19. O programa deve começar atribuindo um valor para a variável  $x$ . Quando o programa é executado, uma mensagem deve ser mostrada dizendo se o valor atribuído a  $x$  é ou não é um número primo. Execute o programa com  $x = 79$ ,  $x = 126$  e  $x = 367$ .

**1.23** Escreva no MATLAB uma função que implemente o algoritmo desenvolvido no Problema 1.20. Chame a função de `sp = somaprimos(int)`, onde o argumento de entrada `int` deve ser um número maior que 1, e o argumento

de saída `sp` é a soma de todos os números primos menores que `int`. Use essa função para calcular a soma de todos os números primos entre 0 e 30.

- 1.24** Escreva no MATLAB uma função que converta números inteiros para o formato binário. Chame a função de `b = integerTObina(d)`, onde o argumento de entrada `d` é o inteiro a ser convertido e o argumento de saída `b` é um vetor com 1s e 0s que representa o número binário. O maior número que pode ser convertido com o emprego da função deve ser um número binário com vinte 1s. Se um número maior for entrado como `d`, a função deve exibir uma mensagem de erro. Use a função para converter os números 81, 30952 e 1500000.
- 1.25** Escreva no MATLAB uma função que converta números reais do formato decimal para o formato binário. Chame a função de `b = deciTObina(d)`, onde o argumento de entrada `d` é o inteiro a ser convertido e o argumento de saída `b` é um vetor de 30 posições representando o número binário (contendo 1s e 0s). Os primeiros 15 elementos de `b` armazenam os algarismos à esquerda da vírgula decimal, e os seus últimos 15 elementos armazenam os algarismos à direita da vírgula decimal. Se mais de 15 posições forem necessárias no formato binário para representar os algarismos à direita da vírgula decimal, os algarismos devem ser cortados. Se o número `d` for maior do que a capacidade de armazenamento de `b`, a função deve exibir uma mensagem de erro. Use essa função na janela de comandos do MATLAB para converter os números 85,321; 0,00671 e 3006,42.

- 1.26** O valor de  $\pi$  pode ser calculado com a série:

$$\pi = 4 \sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{2n-1} = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots\right) \quad (1.22)$$

Escreva um programa no MATLAB que calcule o valor de  $\pi$  usando  $n$  termos da série e calcule o erro relativo real correspondente (para o valor verdadeiro de  $\pi$ , use a variável `pi` predefinida no MATLAB). Use o programa para calcular  $\pi$  e o erro relativo real para:

(a)  $n = 10$    (b)  $n = 20$    (c)  $n = 40$

# Fundamentos Matemáticos

### Tópicos centrais

Conceitos de cálculo (2.2)

Vetores (2.3)

Matrizes e álgebra linear (2.4)

Equações diferenciais ordinárias (EDO) (2.5)

Funções de duas ou mais variáveis

independentes, derivadas parciais (2.6)

Série de Taylor (2.7)

### 2.1 FUNDAMENTOS

Este livro se concentra no emprego de métodos numéricos para resolver problemas que não possuem solução analítica (ou que sejam difíceis de serem resolvidos analiticamente). Assume-se que o leitor tenha um conhecimento prévio em cálculo, álgebra linear e equações diferenciais. Este capítulo tem dois objetivos. O primeiro deles é apresentar, como referência, fundamentos que ajudem na análise de métodos numéricos. O segundo objetivo consiste em revisar conceitos e termos fundamentais do cálculo que possam ajudar na dedução desses métodos. Este capítulo também define muitos dos termos matemáticos e da notação usados no restante do livro.

Deve-se enfatizar que os tópicos apresentados nesta seção devem ser vistos como uma revisão, e que por esse motivo não serão apresentados de forma detalhada. Esta seção não substitui de forma alguma os livros de cálculo nos quais tais assuntos são tratados de forma mais rigorosa. Em cada tópico, o objetivo é fornecer uma definição básica seguida de uma curta explicação. Isto servirá como um lembrete dos conceitos com os quais se assume que o leitor esteja familiarizado. Se necessário, o leitor deve usar essa informação como uma base para a busca de livros de referência em álgebra linear e cálculo que possuam uma exposição mais rigorosa desses tópicos.

Alguns dos assuntos discutidos neste capítulo são repetidos mais adiante à medida que os métodos numéricos são apresentados. Um capítulo típico começa com uma seção explicando os fundamentos analíticos e o porquê do método numérico correspondente ser necessário. A repetição é necessária para que o método numérico possa ser apresentado de forma completa.

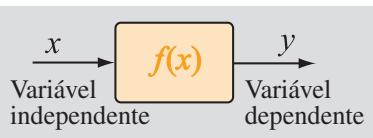
Este capítulo é organizado da seguinte forma. Alguns conceitos fundamentais da disciplina de cálculo (e anteriores a essa disciplina) são cobertos na

Seção 2.2. Vetores e os conceitos de independência e dependência linear são discutidos na Seção 2.3. Em seguida, apresenta-se uma revisão de matrizes e álgebra linear elementar na Seção 2.4. Equações diferenciais ordinárias são discutidas brevemente na Seção 2.5. Funções com duas ou mais variáveis independentes e derivadas parciais são revisadas na Seção 2.6. Finalmente, a expansão em série de Taylor de funções de uma e duas variáveis é descrita na Seção 2.7.

## 2.2 CONCEITOS DE CÁLCULO E PRÉ-CÁLCULO

### Função

Uma função escrita como  $y = f(x)$  associa um único número  $y$  (**variável dependente**) a cada valor de  $x$  (**variável independente**) (Fig. 2-1). A faixa de valores que  $x$  pode ter desde seu mínimo até seu máximo é chamada de **domínio**, e a faixa de valores correspondentes de  $y$  é chamada de **imagem**. O domínio e a imagem das variáveis também são chamados de intervalos. Quando o intervalo inclui os pontos finais (o primeiro e o último ponto da variável), ele é chamado de **intervalo fechado**;



**Figura 2-1** Uma função.

quando os pontos finais não estão incluídos, o intervalo é chamado de **intervalo aberto**. Se os pontos finais do intervalo de  $x$  são  $a$  e  $b$ , então o intervalo fechado de  $x$  é escrito como  $[a, b]$ , e o intervalo aberto, como  $(a, b)$ . Uma função pode ter mais de uma variável independente. Por exemplo, a função  $T = f(x, y, z)$  possui três variáveis independentes, com um único número  $T$  associado a cada conjunto de valores de  $x, y$  e  $z$ .

### Límite de uma função

Se uma função  $f(x)$  se aproxima arbitrariamente de um número  $L$  à medida que  $x$  tende a um número  $a$  vindo da direita ou da esquerda, então diz-se que o limite de  $f(x)$  tende a  $L$  à medida que  $x$  tende a  $a$ . Simbolicamente, o limite é expresso por:

$$\lim_{x \rightarrow a} f(x) = f(a) = L \quad (2.1)$$

A definição formal diz que, se  $f(x)$  é uma função definida em um intervalo aberto contendo  $a$ , e  $L$  é um número real, então para cada número  $\epsilon > 0$ , existe um número  $\delta > 0$  tal que se  $0 < |x - a| < \delta$  então  $|f(x) - L| < \epsilon$ . Como é possível escolher  $\delta$  arbitrariamente pequeno, pode-se fazer  $f(x)$  se aproximar do limite  $L$  o quanto se desejar. A Eq. (2.1), escrita dessa maneira, implica que o limite existe. Isto nem sempre é o caso e às vezes as funções não possuem limites em certos pontos. Entretanto, aquelas que possuem limites não podem ter dois limites diferentes à medida que  $x \rightarrow a$ . Em outras palavras, se o limite de uma função existe, então ele é único.

## Continuidade de uma função

Diz-se que uma função  $f(x)$  é **contínua** em  $x = a$  se as três seguintes condições forem satisfeitas:

$$(1) f(a) \text{ existe, } (2) \lim_{x \rightarrow a} f(x) \text{ existe e } (3) \lim_{x \rightarrow a} f(x) = f(a)$$

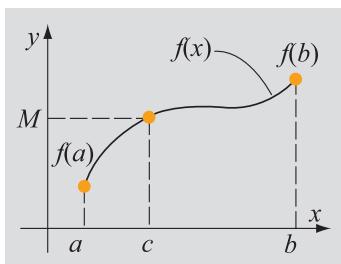
Uma função é contínua em um intervalo aberto  $(a, b)$  se for contínua em cada ponto do intervalo. Uma função contínua em todo o eixo real  $(-\infty, \infty)$  é chamada de **contínua em todo o domínio**.

Numericamente, a continuidade significa que pequenas variações na variável independente levam a pequenas variações na variável dependente.

## Teorema do valor intermediário

O teorema do valor intermediário é um teorema útil sobre o comportamento

de uma função em um intervalo fechado. Formalmente, ele diz que se a função  $f(x)$  é contínua no intervalo fechado  $[a, b]$ , então existe pelo menos um número  $c$  em  $[a, b]$  tal que  $f(c) = M$  (Fig. 2-2). Note que este teorema diz que existe pelo menos um  $c$ , mas ele não fornece um método para que o valor de  $c$  seja encontrado. Tal teorema é chamado de teorema da existência. O teorema do valor intermediário implica que o gráfico de uma função contínua não pode apresentar um salto vertical.



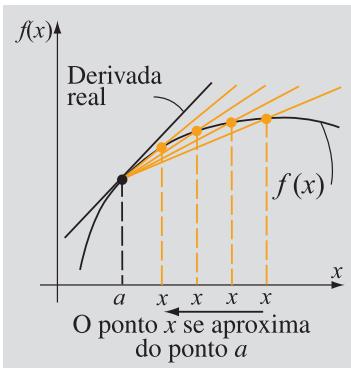
**Figura 2-2** Teorema do valor intermediário.

## Derivadas de uma função

A derivada ordinária, derivada primeira, ou, simplesmente, a derivada de uma função  $y = f(x)$  em um ponto  $x = a$  no domínio de  $f$  é representada por  $\frac{dy}{dx}$ ,  $y'$ ,  $\frac{df}{dx}$ , ou  $f'(a)$ , e é definida como:

$$\left. \frac{dy}{dx} \right|_{x=a} = f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} \quad (2.2)$$

A Eq. (2.2) define uma derivada ordinária porque a função  $f(x)$  é uma função de uma única variável independente. Note que a grandeza  $\frac{f(x) - f(a)}{x - a}$  representa a **inclinação** da linha secante que conecta os pontos  $(a, f(a))$  e  $(x, f(x))$ . No limite em que  $x \rightarrow a$ , pode ser visto a partir da Fig. 2-3 que o limite é a reta tangente ao ponto  $(a, f(a))$ . Portanto, a derivada da função  $f(x)$  no ponto  $x = a$  é a inclinação da reta tangente à curva  $y = f(x)$  naquele ponto. Uma função deve ser contínua antes de ser derivável. Uma função contínua e derivável ao longo de um certo intervalo é chamada de **suave**.



**Figura 2-3** Derivada de uma função.

Há duas maneiras importantes de se interpretar a derivada primeira de uma função. Uma delas já foi mencionada anteriormente, e trata a derivada como a inclinação da reta tangente à curva descrita por  $y = f(x)$  em um ponto. Essa primeira interpretação é especialmente útil na obtenção do máximo e do mínimo da curva  $y = f(x)$ , pois a inclinação (e portanto a derivada primeira) devem ser nulas em tais pontos. A segunda interpretação da derivada relaciona essa operação à taxa de variação da função  $y = f(x)$  em relação a  $x$ . Em outras palavras,  $\frac{dy}{dx}$  representa quanto rapidamente  $y$  varia à medida que  $x$  varia.

Derivadas de ordem mais elevada podem ser obtidas com a aplicação sucessiva da definição (2.2) em cada derivada. Em outras palavras, a derivada segunda  $\frac{d^2y}{dx^2}$  é obtida com a diferenciação da derivada primeira, isto é  $\frac{d}{dx}\left(\frac{dy}{dx}\right)$ . De forma similar, a derivada terceira é a derivada primeira da derivada segunda, ou  $\frac{d^3y}{dx^3} = \frac{d}{dx}\left(\frac{d^2y}{dx^2}\right)$ , e assim por diante.

### A regra da cadeia para a derivação ordinária

A regra da cadeia é útil na diferenciação de funções cujos argumentos também são funções. Por exemplo, se  $y = f(u)$ , onde  $u = g(x)$ , então  $y = f(g(x))$  também é derivável e a seguinte regra da cadeia pode ser aplicada:

$$\frac{dy}{dx} = \left(\frac{dy}{du}\right)\left(\frac{du}{dx}\right) \quad (2.3)$$

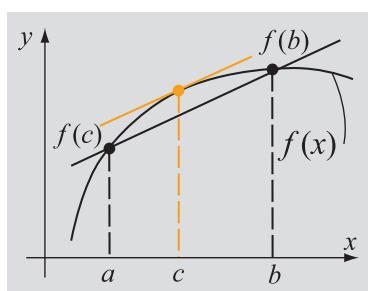
### Teorema do valor médio para derivadas

O teorema do valor médio é muito útil na determinação de valores limite para o erro associado à aplicação de diferentes métodos numéricos. Formalmente, ele

diz que se  $f(x)$  é uma função contínua no intervalo fechado  $[a, b]$  e derivável no intervalo aberto  $(a, b)$ , então existe um número  $c$  dentro do intervalo,  $c \in (a, b)$ , tal que:

$$f'(c) = \frac{dy}{dx}\Big|_{x=c} = \frac{f(b) - f(a)}{b - a} \quad (2.4)$$

Enunciado de forma simplificada, o teorema do valor médio para derivadas diz, conforme ilustrado na Fig. 2-4, que dentro do intervalo existe um ponto  $c$  tal que o valor da derivada de  $f(x)$  é exatamente igual à linha secante que junta os pontos finais  $(a, f(a))$  e  $(b, f(b))$ .



**Figura 2-4** Teorema do valor médio para as derivadas.

### Integral de uma função

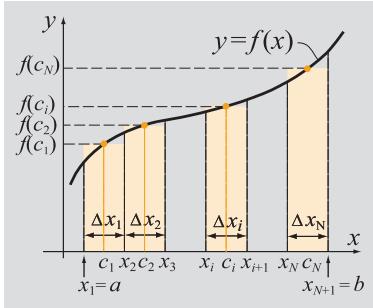
Há dois tipos de integral de uma função de uma variável, a *integral indefinida* e a *integral definida*. A **integral indefinida** é o oposto ou inverso da derivada, sendo portanto também chamada de **antiderivada**. Por exemplo, se  $g(x)$  é a derivada de

$f(x)$ , isto é, se  $g(x) = \frac{df(x)}{dx}$ , então a antiderivada ou integral indefinida de  $g(x)$  é  $f(x)$ , sendo escrita como:

$$f(x) = \int g(x) dx \quad (2.5)$$

Uma **integral definida**,  $I$ , é representada por:

$$I = \int_a^b f(x) dx \quad (2.6)$$



**Figura 2-5** Integral definida usando uma soma de Riemann.

A integral definida é um número, e existe em um intervalo fechado  $[a, b]$ . Ela possui limites de integração inferior e superior  $a$  e  $b$ , respectivamente; a função  $f(x)$  escrita logo após o sinal de integração  $\int$  é chamada de integrando. Uma integral pode ser definida usando-se a soma de Riemann. Considere uma função  $f(x)$  definida e contínua no intervalo  $[a, b]$ . O domínio pode ser dividido em  $n$  subintervalos definidos por  $\Delta x_i = x_{i+1} - x_i$ , onde  $i = 1, \dots, n$ . A soma de Riemann para  $f(x)$  em  $[a, b]$  é definida como

$$\sum_{i=1}^n f(c_i) \Delta x_i, \text{ onde } c_i \text{ é um número pertencente ao subintervalo } [x_i, x_{i+1}]$$

como o limite da soma de Riemann quando a largura de todos os subintervalos de  $[a, b]$  tende a zero:

$$I = \int_a^b f(x) dx = \lim_{\Delta x_i \rightarrow 0} \sum_{i=1}^n f(c_i) \Delta x_i \quad (2.7)$$

Pode-se ver a partir da Fig. 2-5 que o valor de  $\lim_{\Delta x_i \rightarrow 0} \sum_{i=1}^n f(c_i) \Delta x_i$  é igual à área sob a curva especificada por  $y = f(x)$ . Essa interpretação da integral como sendo a área sob a curva é útil no desenvolvimento de métodos aproximados para a integração numérica de funções (ver Capítulo 7).

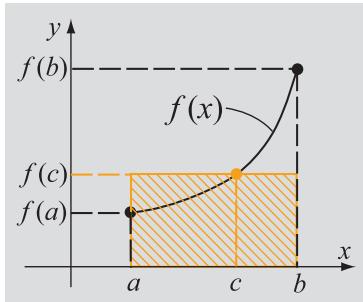
### Teorema fundamental do cálculo

A conexão entre a diferenciação e a integração é expressa pelo teorema fundamental do cálculo, que diz que, se uma função  $f(x)$  é contínua ao longo do intervalo fechado  $[a, b]$  e  $F(x)$  é a antiderivada de  $f(x)$  ao longo desse mesmo intervalo, então

$$\int_a^b f(x) dx = F(b) - F(a) \quad (2.8)$$

### Teorema do valor médio para integrais

Uma maneira de se interpretar a integral definida de uma função  $f(x)$  monotonicamente crescente em um intervalo  $[a, b]$  é dizer que ela corresponde à área sob



**Figura 2-6** Teorema do valor médio para uma integral.

a curva  $y = f(x)$ . Pode ser mostrado que a área sob a curva é limitada pela área do retângulo mais baixo  $f(a)(b - a)$  e a área do retângulo superior  $f(b)(b - a)$ . O teorema do valor médio para integrais diz que, em algum lugar entre esse dois retângulos, existe um retângulo cuja área é igual à área sob a curva (Fig. 2-6). Formalmente, o teorema diz que, se a função  $f(x)$  é contínua no intervalo fechado  $[a, b]$ , então existe um número  $c \in [a, b]$  tal que:

$$\int_a^b f(x)dx = f(c)(b - a) \quad (2.9)$$

### Valor médio de uma função

O valor  $f(c)$  que aparece no teorema do valor médio para as integrais, descrito na Eq. (2.9), é chamado de valor médio da função  $f(x)$  no intervalo  $[a, b]$ . Logo, o valor médio de  $f(x)$  ao longo de  $[a, b]$  é representado por  $\langle f \rangle$  e definido como:

$$\langle f \rangle = \frac{1}{(b-a)} \int_a^b f(x)dx \quad (2.10)$$

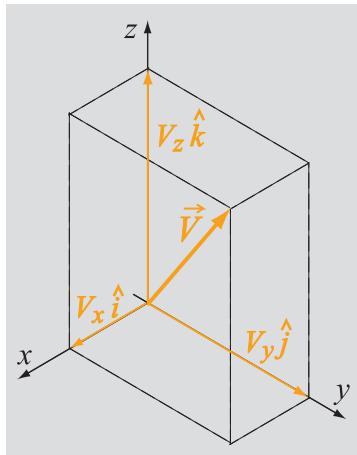
### Segundo teorema fundamental do cálculo

Este teorema permite que se avalie a derivada de uma integral definida. Formalmente, ele diz que, se  $f(x)$  é uma função contínua ao longo de um intervalo aberto contendo o número  $a$ , então para cada  $x$  no intervalo,

$$\frac{d}{dx} \left[ \int_a^x f(\xi)d\xi \right] = f(x) \quad (2.11)$$

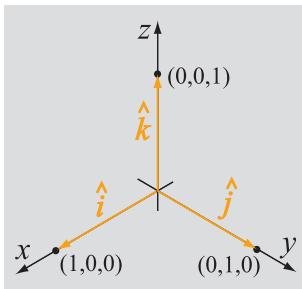
onde  $\xi$  é uma variável muda representando a coordenada ao longo do intervalo.

## 2.3 VETORES



**Figura 2-7** Vetor  $V$  com componentes  $V_x$ ,  $V_y$  e  $V_z$ .

Vetores são grandezas (matemáticas ou físicas) que possuem dois atributos: módulo e direção. Em contraste, grandezas com apenas um atributo (como o módulo) são chamadas de *escalares*. Exemplos de grandezas escalares são massa, comprimento e volume. Exemplos de vetores são força, momento e aceleração. Uma das maneiras usadas para representar uma grandeza vetorial implica o uso de uma pequena seta (ou linha curta) sobre uma letra qualquer, como  $\vec{V}$  (em muitos livros, nomes de vetores são escritos em negrito). O módulo de um vetor é representado pela própria letra,  $V$ , ou escrito como  $|V|$ . Um vetor é usualmente definido em relação a um sistema de coordenadas. Uma vez escolhido o sistema de coordenadas, um vetor pode ser representado graficamente em tal espaço por um segmento de reta orientado (isto é, uma linha com uma seta) (Fig. 2-7). As projeções do vetor em cada um



**Figura 2-8** Vetores unitários.

dos eixos das coordenadas definem os componentes do vetor. Se  $V_x \hat{i}$  é o componente  $x$  (isto é, a projeção do vetor no eixo  $x$ ),  $V_y \hat{j}$  é o componente  $y$  e  $V_z \hat{k}$  é o componente  $z$ , então o vetor  $\vec{V}$  pode ser escrito como:

$$\vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k} \quad (2.12)$$

onde  $\hat{i}$ ,  $\hat{j}$  e  $\hat{k}$  são os vetores unitários nas direções  $x$ ,  $y$  e  $z$ , respectivamente. Vetores unitários, mostrados na Fig. 2-8, são vetores que possuem uma direção específica e módulo 1. Além disso, um vetor pode ser escrito listando-se os módulos de seus componentes em uma linha ou em uma coluna:

$$\vec{V} = \begin{bmatrix} V_x & V_y & V_z \end{bmatrix} \quad \text{ou} \quad \vec{V} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \quad (2.13)$$

Às vezes, um vetor é representado por  $V_i$ , onde  $i = x, y, z$ , ou  $i = 1, 2, 3$ .

O módulo de um vetor no espaço tridimensional cartesiano é o seu comprimento, determinado por:

$$|\vec{V}| = \sqrt{V_x^2 + V_y^2 + V_z^2} \quad (2.14)$$

A direção do vetor pode ser especificada pelo vetor unitário (um vetor unitário na direção do vetor). Esse vetor unitário, escrito como  $\hat{V}$ , é obtido com a divisão da Eq. (2.12) pelo módulo (comprimento) dado pela Eq. (2.14):

$$\hat{V} = \frac{\vec{V}}{|\vec{V}|} = \frac{V_x \hat{i} + V_y \hat{j} + V_z \hat{k}}{\sqrt{V_x^2 + V_y^2 + V_z^2}} = l \hat{i} + m \hat{j} + n \hat{k} \quad (2.15)$$

onde  $l = \frac{V_x}{\sqrt{V_x^2 + V_y^2 + V_z^2}}$ ,  $m = \frac{V_y}{\sqrt{V_x^2 + V_y^2 + V_z^2}}$  e  $n = \frac{V_z}{\sqrt{V_x^2 + V_y^2 + V_z^2}}$  são chamados

de cossenos direcionais e correspondem ao cosseno dos ângulos entre o vetor e os eixos do sistema de coordenadas,  $x$ ,  $y$  e  $z$ , respectivamente.

Em situações físicas, os vetores estão restritos a no máximo três dimensões. A idéia de vetores, no entanto, pode ser generalizada na matemática (álgebra linear) a dimensões maiores que três. Um vetor se torna então uma lista (ou conjunto) de  $n$  números (elementos ou componentes) escritos em uma linha ou coluna. Seu nome é representado entre colchetes:

$$[V] = \begin{bmatrix} V_1 & V_2 & \dots & V_n \end{bmatrix} \quad \text{ou} \quad [V] = \begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_n \end{bmatrix} \quad (2.16)$$

O elemento de um vetor é chamado de  $V_i$ , onde o subscrito denota a posição do elemento na linha ou coluna. Quando os componentes são escritos como uma

linha, o vetor é chamado de ***vetor linha***. Quando escritos como uma coluna, o vetor é chamado de ***vetor coluna***. Na Eq. (2.16), o vetor linha é chamado de vetor ( $1 \times n$ ), indicando que ele possui uma linha e  $n$  colunas. O vetor coluna na Eq. (2.16) é chamado de vetor ( $n \times 1$ ).

### 2.3.1 Operações com vetores

Dois vetores são iguais se forem do mesmo tipo (linha ou coluna) e se todos os seus elementos em posições equivalentes forem iguais. Algumas das operações fundamentais da matemática são definidas para os vetores, enquanto outras não. Por exemplo, vetores podem ser somados, subtraídos e multiplicados de certa maneira, mas não podem ser divididos. Também há operações únicas para vetores. As operações básicas são resumidas a seguir.

#### Soma e subtração de dois vetores

Dois vetores podem ser somados ou subtraídos apenas se forem do mesmo tipo (isto é, ambos vetores linha ou vetores coluna) e do mesmo tamanho (isto é, com o mesmo número de componentes ou elementos). Dados dois vetores (linha ou coluna)  $\vec{V} = [V_i] = [V_1, \dots, V_n]$  e  $\vec{U} = [U_i] = [U_1, \dots, U_n]$ , a sua soma é:

$$\vec{V} + \vec{U} = [V_i + U_i] = [V_1 + U_1, V_2 + U_2, \dots, V_n + U_n] \quad (2.17)$$

De forma similar, para a subtração:

$$\vec{V} - \vec{U} = [V_i - U_i] = [V_1 - U_1, V_2 - U_2, \dots, V_n - U_n] \quad (2.18)$$

#### Multiplicação de um vetor por um escalar

Quando um vetor é multiplicado por um escalar, cada um de seus elementos deve ser multiplicado por esse escalar. Dados o vetor  $\vec{V} = [V_i] = [V_1, V_2, \dots, V_n]$  e o escalar  $\alpha$ , a multiplicação de ambos resulta em:

$$\alpha \vec{V} = \alpha [V_i] = [\alpha V_1, \alpha V_2, \dots, \alpha V_n] \quad (2.19)$$

A mesma propriedade se aplica quando  $\vec{V}$  é um vetor coluna.

#### Transposta de um vetor

A operação transposta transforma um vetor linha em um vetor coluna, e vice-versa. Por exemplo, se  $\vec{V} = [V_1, V_2, \dots, V_n]$  é um vetor linha ( $1 \times n$ ), então a transposta de  $\vec{V}$ , escrita como  $\vec{V}^T$ , é dada pelo seguinte vetor coluna ( $n \times 1$ ):

$$\vec{V}^T = \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{bmatrix} \quad (2.20)$$

#### Multiplicação de dois vetores

Há diferentes maneiras de multiplicar dois vetores. Duas das formas que produzem resultados com sentido físico são o produto interno e o produto cruzado. O

produto interno resulta em uma grandeza escalar, enquanto o produto cruzado resulta em uma grandeza vetorial.

### Produto interno ou escalar de dois vetores

O produto interno de dois vetores  $\vec{V} = [V_i]$  e  $\vec{U} = [U_i]$  é definido como:

$$\vec{V} \bullet \vec{U} = [V_i][U_i] = V_1 U_1 + V_2 U_2 + \dots V_n U_n \quad (2.21)$$

O resultado de tal multiplicação é um número, ou escalar. Às vezes, o produto interno é escrito em notação resumida:

$$\vec{V} \bullet \vec{U} = V_i U_i \quad (2.22)$$

onde os subscritos repetidos implicam a soma de todos os valores possíveis referentes àquele subscrito, isto é,  $V_i U_i = \sum_{i=1}^n V_i U_i$ .

Pode ser dada ao produto interno uma interpretação geométrica quando os dois vetores são desenhados em um sistema de coordenadas. Pode-se mostrar a partir de geometria e trigonometria elementares que:

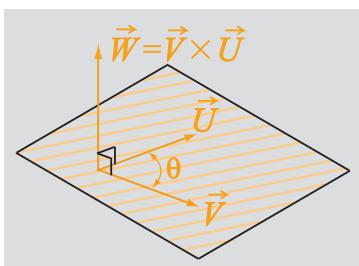
$$\vec{V} \bullet \vec{U} = |\vec{V}| |\vec{U}| \cos \theta \quad (2.23)$$

onde  $|\vec{V}|$  e  $|\vec{U}|$  são os módulos dos vetores (ver Eq. (2.14)) e  $\theta$  é o ângulo formado pelos dois vetores.

### Produto cruzado ou vetorial de dois vetores

O produto cruzado, ou vetorial, de dois vetores resulta em um outro vetor. Para dois vetores  $\vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k}$  e  $\vec{U} = U_x \hat{i} + U_y \hat{j} + U_z \hat{k}$  definidos em um sistema de coordenadas cartesiano tridimensional, o produto cruzado  $\vec{W} = \vec{V} \otimes \vec{U}$  é definido como:

$$\vec{W} = \vec{V} \otimes \vec{U} = (V_y U_z - V_z U_y) \hat{i} + (V_z U_x - V_x U_z) \hat{j} + (V_x U_y - V_y U_x) \hat{k} \quad (2.24)$$



**Figura 2-9** Produto cruzado.

Conforme ilustrado na Fig. 2-9, o vetor  $\vec{W}$  é perpendicular ao plano formado por  $\vec{V}$  e  $\vec{U}$  e a magnitude de  $\vec{W}$  é dada por:

$$|\vec{W}| = |\vec{V}| |\vec{U}| \sin \theta \quad (2.25)$$

onde  $\theta$  é o ângulo formado pelos dois vetores,  $\vec{V}$  e  $\vec{U}$ .

### Dependência e independência linear de um conjunto de vetores

Diz-se que um conjunto de vetores  $\vec{V}_1, \vec{V}_2, \dots, \vec{V}_n$  é **linearmente independente** se a expressão

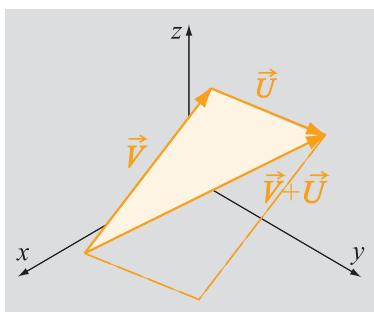
$$\alpha_1 \vec{V}_1 + \alpha_2 \vec{V}_2 + \dots + \alpha_n \vec{V}_n = 0 \quad (2.26)$$

é satisfeita se e somente se  $\alpha_1 = \alpha_2 = \dots = \alpha_n = 0$ . Do contrário, diz-se que os vetores são **linearmente dependentes**. Em outras palavras, se *qualquer* um dos números  $\alpha_1, \alpha_2, \dots, \alpha_n$  não for identicamente zero, então o conjunto de vetores é linearmente

dependente. Como um exemplo, considere os vetores coluna  $\vec{V} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ ,  $\vec{U} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$  e  $\vec{W} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$ . Por inspeção, pode-se ver que  $\vec{V}$  e  $\vec{U}$  são linearmente independentes.

Entretanto, como  $2\vec{V} + 3\vec{U} - \vec{W} = 0$ ,  $\vec{W}$  é linearmente dependente de  $\vec{V}$  e  $\vec{U}$ . A Eq. (2.26) é chamada de **combinação linear** de vetores. Um vetor é portanto linearmente dependente de outro conjunto de vetores se puder ser expresso como uma combinação linear desses vetores.

### Desigualdade triangular



**Figura 2-10** Desigualdade triangular.

A soma de dois vetores  $\vec{V}$  e  $\vec{U}$  pode ser representada geometricamente (Fig. 2-10) por um paralelogramo cujos lados são os vetores sendo somados; a soma resultante,  $(\vec{V} + \vec{U})$ , é a diagonal principal, conforme mostrado na Fig. (2-10). A desigualdade triangular se refere ao fato de a soma dos comprimentos dos dois lados de um triângulo ser sempre maior que, ou pelo menos igual, ao comprimento do terceiro lado. Ela é escrita como:

$$|\vec{V} + \vec{U}| \leq |\vec{V}| + |\vec{U}| \quad (2.27)$$

Esta propriedade também é útil para as matrizes.

## 2.4 MATRIZES E ÁLGEBRA LINEAR

Uma **matriz** é um arranjo retangular de números. O **tamanho** de uma matriz se refere ao número de linhas e colunas que ela contém. Uma matriz  $(m \times n)$  (“matriz  $m$  por  $n$ ”), tem  $m$  linhas e  $n$  colunas:

$$[a] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (2.28)$$

O nome de uma matriz é escrito entre colchetes. Um elemento (ou campo) de uma matriz é chamado de  $a_{ij}$ , onde os subscritos  $i$  e  $j$  denotam o número da linha e da coluna onde o elemento está posicionado.

As matrizes são úteis na análise de sistemas de equações lineares e em outras aplicações. As matrizes podem ser somadas, subtraídas, multiplicadas e usadas em operações matemáticas que lhes são específicas.

### Relação entre matrizes e vetores

Há uma relação próxima entre matrizes e vetores. A matriz pode ser vista como sendo composta por vetores linha, ou, alternativamente, por vetores coluna. Por outro lado, um vetor é um caso especial de uma matriz. Um vetor linha é

simplesmente uma matriz com uma linha e várias colunas, e um vetor coluna é simplesmente uma matriz com várias linhas e uma coluna.

### 2.4.1 Operações com matrizes

Operações matemáticas realizadas com matrizes caem na área geral da matemática conhecida como **álgebra linear**. Da mesma forma que para com os vetores, apenas certas operações matemáticas são definidas para as matrizes. Essas operações incluem a multiplicação por um escalar, a adição, a subtração e a multiplicação. Além disso, a divisão não é uma operação permitida. Duas matrizes são iguais se tiverem o mesmo tamanho e se todos os seus elementos localizados nas mesmas posições forem iguais.

#### Multiplicação por um escalar

Se  $[a] = [a_{ij}]$  é uma matriz e  $\alpha$  é um escalar, então obtém-se  $\alpha[a] = [\alpha a_{ij}]$  com a multiplicação de cada campo ou elemento da matriz pelo número  $\alpha$ .

#### Soma e subtração de duas matrizes

Duas matrizes podem ser somadas ou subtraídas somente se tiverem o mesmo tamanho. As matrizes  $[a]$  e  $[b]$  (ambas  $(n \times m)$ ) são somadas (ou subtraídas) com a soma (ou subtração) dos elementos correspondentes em ambas. O resultado  $[c]$  é uma matriz de mesmo tamanho, onde:

$$[c_{ij}] = [a_{ij}] + [b_{ij}] \quad (2.29)$$

**Figura 2-11** Soma de matrizes.

para a adição, conforme ilustrado na Fig. 2-11, e:

$$[c_{ij}] = [a_{ij}] - [b_{ij}] \quad (2.30)$$

para a subtração.

#### Transposta de uma matriz

A operação de transposição de uma matriz rearranja essa matriz de forma tal que suas linhas são transformadas em colunas (ou vice-versa, suas colunas são transformadas em linhas) (Fig. 2-12). Em outras palavras, a posição (número da linha, número da coluna) de cada elemento da matriz é trocada. A transposta de  $[a]$  é escrita como  $[a]^T$ . Por exemplo, o elemento  $[a_{12}]$  se torna  $[a_{21}^T]$ , e assim por diante. Em geral:

$$[a]^T = [a_{ij}^T] = [a_{ji}] \quad (2.31)$$

**Figura 2-12** Transposta de uma matriz.

$$\begin{bmatrix} 2 & -1 & 0 \\ 5 & 3 & 1 \\ 6 & 1 & -4 \\ 7 & -2 & 9 \end{bmatrix}^T = \begin{bmatrix} 2 & 5 & 6 & 7 \\ -1 & 3 & 1 & -2 \\ 0 & 1 & -4 & 9 \end{bmatrix}$$

Logo, a transposta da matriz  $(3 \times 4) [a] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$  é a

$$\text{matriz } (4 \times 3): [a]^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \\ a_{14} & a_{24} & a_{34} \end{bmatrix}.$$

### Multiplicação de matrizes

A multiplicação  $[c] = [a][b]$  de uma matriz  $[a]$  por uma matriz  $[b]$  é definida apenas quando o número de colunas da matriz  $[a]$  é igual ao número de linhas da matriz  $[b]$ . Não há restrições quanto ao número de linhas de  $[a]$  ou ao número de colunas de  $[b]$ . O resultado da multiplicação é uma matriz  $[c]$  que possui o mesmo número de linhas de  $[a]$  e o mesmo número de colunas de  $[b]$ . Então, se a matriz  $[a]$  é  $(m \times q)$  e a matriz  $[b]$  é  $(q \times n)$ , então a matriz  $[c]$  é  $(m \times n)$  (Fig. 2-13). Por exemplo, conforme mostrado na Eq. (2.32), se  $[a]$  é  $(3 \times 4)$  e  $[b]$  é  $(4 \times 2)$ , então  $[c]$  é  $(3 \times 2)$ .

$$[c]_{mn} = [a]_{mq}[b]_{qn}$$

**Figura 2-13** Multiplicação de matrizes.

$$\begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} \quad (2.32)$$

Os elementos da matriz  $[c]$  são calculados com a multiplicação das linhas de  $[a]$  pelas colunas de  $[b]$ . Começando com a primeira linha, o valor do elemento  $c_{11}$  é obtido com a multiplicação da primeira linha de  $[a]$  pela primeira coluna de  $[b]$  da seguinte maneira:

$$c_{11} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + a_{14}b_{41} \quad (2.33)$$

O valor do elemento  $c_{12}$  é obtido com a multiplicação da primeira linha de  $[a]$  pela segunda coluna de  $[b]$  da seguinte maneira:

$$c_{12} = a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} + a_{14}b_{42} \quad (2.34)$$

Na segunda linha de  $[c]$ , o valor do elemento  $c_{21}$  é obtido com a multiplicação da segunda linha de  $[a]$  pela primeira coluna de  $[b]$ :

$$c_{21} = a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} + a_{24}b_{41} \quad (2.35)$$

O procedimento de multiplicação continua até que o elemento  $c_{32}$  seja calculado. Em geral, a regra da multiplicação é dada por:

$$c_{ij} = a_{ik}b_{kj} = \sum_{k=1}^q a_{ik}b_{kj} \quad (2.36)$$

Um exemplo numérico da multiplicação é mostrado na Fig. 2-14.

$$\begin{bmatrix} 2 & -1 \\ 8 & 3 \\ 6 & 7 \end{bmatrix} \begin{bmatrix} 4 & 9 & 1 & -3 \\ -5 & 2 & 4 & 6 \end{bmatrix} = \begin{bmatrix} (2 \cdot 4 + -1 \cdot -5) & (2 \cdot 9 + -1 \cdot 2) & (2 \cdot 1 + -1 \cdot 4) & (2 \cdot -3 + -1 \cdot 6) \\ (8 \cdot 4 + 3 \cdot -5) & (8 \cdot 9 + 3 \cdot 2) & (8 \cdot 1 + 3 \cdot 4) & (8 \cdot -3 + 3 \cdot 6) \\ (6 \cdot 4 + 7 \cdot -5) & (6 \cdot 9 + 7 \cdot 2) & (6 \cdot 1 + 7 \cdot 4) & (6 \cdot -3 + 7 \cdot 6) \end{bmatrix}$$

$$= \begin{bmatrix} 13 & 16 & -2 & -12 \\ 17 & 78 & 20 & -6 \\ -11 & 68 & 34 & 24 \end{bmatrix}$$

**Figura 2-14** Exemplo numérico da multiplicação de matrizes.

### 2.4.2 Matrizes especiais

Matrizes com estruturas ou propriedades especiais aparecem quando métodos numéricos são empregados na solução de problemas. Apresenta-se a seguir uma lista contendo tais matrizes, com uma breve descrição de cada uma delas.

#### Matriz quadrada

Uma matriz que tem o mesmo número de linhas e colunas é chamada de matriz quadrada. Em tais matrizes, os campos ou elementos ao longo da diagonal da matriz,  $a_{ii}$ , isto é,  $a_{11}$ ,  $a_{22}$  e assim por diante, são conhecidos como elementos **diagonais**, enquanto todos os demais campos são chamados de elementos **fora da diagonal**. Em uma matriz quadrada, os campos (ou elementos) acima da diagonal, isto é,  $[a_{ij}]$  para  $j > i$ , são chamados de **elementos superdiagonais**, ou **elementos acima da diagonal**. Os campos abaixo da diagonal, isto é,  $[a_{ij}]$  para  $i > j$ , são chamados de **elementos subdiagonais** ou **elementos abaixo da diagonal**.

#### Matriz diagonal

Uma matriz quadrada com elementos diagonais diferentes de zero e todos os elementos fora da diagonal nulos é chamada de **matriz diagonal** e representada por  $[D]$ .

#### Matriz triangular superior

Uma matriz quadrada em que todos os elementos subdiagonais são nulos é chamada de **matriz triangular superior** e representada por  $[U]$ .

#### Matriz triangular inferior

Uma matriz quadrada em que todos os elementos superdiagonais são nulos é chamada de **matriz triangular inferior** e representada por  $[L]$ .

#### Matriz identidade

A **matriz identidade**  $[I]$  é uma matriz quadrada em que todos os elementos diagonais são iguais a 1 e todos os elementos fora da diagonal são iguais a 0. A matriz identidade é análoga ao número 1 para matrizes. Qualquer matriz que seja multiplicada pela matriz identidade permanece inalterada:

$$[a][I] = [a] \quad (2.37)$$

#### Matriz zero

A **matriz zero** é uma matriz em que todos os elementos são nulos.

#### Matriz simétrica

A **matriz simétrica** é uma matriz quadrada na qual  $[a_{ij}] = [a_{ji}]$ . Para uma matriz simétrica, a transposta da matriz é igual à própria matriz:

$$[a]^T = [a] \quad (2.38)$$

### 2.4.3 Inversa de uma matriz

A divisão é uma importante operação não definida para matrizes. Entretanto, uma operação definida para matrizes que serve para um propósito equivalente à

divisão é a **inversa** de uma matriz. Uma matriz quadrada  $[a]$  pode ser invertida desde que exista uma matriz  $[b]$  de mesmo tamanho tal que  $[a][b] = [I]$ , onde  $[I]$  é a matriz identidade. A matriz  $[b]$  é chamada de **inversa** de  $[a]$ , sendo escrita como  $[a]^{-1}$ . Logo:

$$[a][a]^{-1} = [a]^{-1}[a] = [I] \quad (2.39)$$

O Exemplo 2-1 ilustra a propriedade expressa pela Eq. (2.39).

#### 2.4.4 Propriedades de matrizes

A seguir são listadas propriedades gerais de matrizes:

- $[a] + [b] = [b] + [a]$
- $([a] + [b]) + [c] = [a] + ([b] + [c])$
- $\alpha([a] + [b]) = \alpha[a] + \alpha[b]$ , onde  $\alpha$  é um escalar.
- $(\alpha + \beta)[a] = \alpha[a] + \beta[a]$ , onde  $\alpha$  e  $\beta$  são escalares.

As propriedades acima também se aplicam à subtração.

- Se  $[a]$  e  $[b]$  são matrizes quadradas, então em geral  $[a][b] \neq [b][a]$  (a menos que uma seja a inversa da outra). Se  $[a]$  ou  $[b]$  não for quadrada, e o produto  $[a][b]$  existir, então o produto  $[b][a]$  não é definido e não existe. Em outras palavras, quando matrizes estão envolvidas, a ordem da multiplicação é importante.
- $([a] + [b])[c] = [a][c] + [b][c]$ , sendo a ordem da multiplicação importante.
- $[a]([b] + [c]) = [a][b] + [a][c]$ .
- $\alpha([a][b]) = (\alpha[a])[b] = [a](\alpha[b])$ , onde  $\alpha$  é um escalar.

#### Exemplo 2-1: Inversa de uma matriz

Mostre que a matriz  $[b] = \begin{bmatrix} 0,1 & 0,2 & 0 \\ 0,4 & 0,1 & 0,2 \\ 0,2 & 0,1 & 0,8 \end{bmatrix}$  é a inversa da matriz  $[a] = \begin{bmatrix} -1,2 & 3,2 & -0,8 \\ 5,6 & -1,6 & 0,4 \\ -0,4 & -0,6 & 1,4 \end{bmatrix}$ .

#### SOLUÇÃO

Para mostrar que a matriz  $[b]$  é a inversa da matriz  $[a]$ , é feita a multiplicação das duas matrizes.

$$[a][b] = \begin{bmatrix} -1,2 & 3,2 & -0,8 \\ 5,6 & -1,6 & 0,4 \\ -0,4 & -0,6 & 1,4 \end{bmatrix} \begin{bmatrix} 0,1 & 0,2 & 0 \\ 0,4 & 0,1 & 0,2 \\ 0,2 & 0,1 & 0,8 \end{bmatrix} =$$

$$\begin{bmatrix} (1,2 \cdot 0,1 + 3,2 \cdot 0,4 + -0,8 \cdot 0,2) & (1,2 \cdot 0,2 + 3,2 \cdot 0,1 + -0,8 \cdot 0,1) & (1,2 \cdot 0 + 3,2 \cdot 0,2 + -0,8 \cdot 0,8) \\ (5,6 \cdot 0,1 + -1,6 \cdot 0,4 + 0,4 \cdot 0,2) & (5,6 \cdot 0,2 + -1,6 \cdot 0,1 + 0,4 \cdot 0,1) & (5,6 \cdot 0 + -1,6 \cdot 0,2 + 0,4 \cdot 0,8) \\ (-0,4 \cdot 0,1 + -0,6 \cdot 0,4 + 1,4 \cdot 0,2) & (-0,4 \cdot 0,2 + -0,6 \cdot 0,1 + 1,4 \cdot 0,1) & (-0,4 \cdot 0 + -0,6 \cdot 0,2 + 1,4 \cdot 0,8) \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Se  $[a]$  e  $[b]$  são matrizes para as quais  $[a][b]$  é definida e existe, então  $([a][b])^T = [b]^T[a]^T$ . Note que a ordem de multiplicação é alterada.
- Para qualquer matriz  $[a]$ ,  $([a]^T)^T = [a]$ .
- Para uma matriz  $[a]$  que possui inversa,  $([a]^{-1})^{-1} = [a]$ .
- Se  $[a]$  e  $[b]$  são duas matrizes quadradas de mesmo tamanho que possuem inversas, então  $([a][b])^{-1} = [b]^{-1}[a]^{-1}$ .

#### 2.4.5 Determinante de uma matriz

O determinante, definido apenas para matrizes quadradas, é uma grandeza que aparece de forma proeminente na obtenção da inversa de uma matriz e que fornece informações úteis quanto à existência ou não de soluções para um conjunto de equações simultâneas. O cálculo do determinante de uma matriz é freqüentemente difícil se o tamanho da matriz for maior que  $(3 \times 3)$  ou  $(4 \times 4)$ .

O determinante é um número. Ele corresponde à soma de todos os produtos possíveis formados por um elemento de cada linha e de cada coluna, considerando o sinal apropriado. O sinal apropriado de cada termo é obtido escrevendo os termos individuais em cada produto e contando o número de trocas necessárias para colocar os subscritos na ordem  $1, 2, \dots, n$ . Se o número necessário para a realização de tais trocas for par, então o sinal é  $+$ ; se ele for ímpar, então o sinal é  $-$ . Formalmente, o determinante de uma matriz  $[a]_{n \times n}$  é chamado de  $\det(a)$  ou  $|a|$ , e é definido como:

$$\det(A) = |A| = \sum_j (-1)^k a_{1,j_1} a_{2,j_2} \dots a_{n,j_n} \quad (2.40)$$

onde a soma é realizada para todas as  $n!$  permutações de grau  $n$  e  $k$  é o número de mudanças necessárias para que os segundos subscritos sejam colocados na ordem  $1, 2, 3, \dots, n$ . O uso da Eq. (2.40) é ilustrado para  $n = 1$ ,  $n = 2$  e  $n = 3$ .

Para  $n = 1$ , a matriz é  $(1 \times 1)$ ,  $[a] = [a_{11}]$  e o determinante é:

$$\det(a) = a_{11}$$

Para  $n = 2$ , a matriz é  $(2 \times 2)$ ,  $[a] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$  e o determinante é:

$$\det(a) = (-1)^0 a_{11}a_{22} + (-1)^1 a_{12}a_{21} = a_{11}a_{22} - a_{12}a_{21}$$

Para  $n = 3$ , a matriz é  $(3 \times 3)$ ,  $[a] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$  e o determinante é:

$$\begin{aligned} \det(A) &= (-1)^0 a_{11}a_{22}a_{33} + (-1)^1 a_{11}a_{23}a_{32} + (-1)^1 a_{12}a_{21}a_{33} \\ &\quad + (-1)^2 a_{12}a_{23}a_{31} + (-1)^2 a_{13}a_{21}a_{32} + (-1)^3 a_{13}a_{22}a_{31} \\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \end{aligned}$$

Pode-se ver que a avaliação do determinante de matrizes grandes é pouco prática tanto manualmente quanto usando um computador, em função do grande número de operações necessárias para que as  $n!$  permutações sejam levadas em consideração.

#### 2.4.6 A regra de Cramer e a solução de sistemas de equações lineares simultâneas

Um conjunto de  $n$  equações lineares simultâneas com  $n$  incógnitas  $x_1, x_2, \dots, x_n$  é dado por:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\ \dots + \dots + \dots + \dots &= \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (2.41)$$

O sistema pode ser escrito de forma compacta com o uso de matrizes:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} \quad (2.42)$$

A Eq. (2.42) também pode ser escrita como:

$$[a][x] = [b] \quad (2.43)$$

onde  $[a]$  é a matriz de coeficientes,  $[x]$  é o vetor contendo as  $n$  incógnitas e  $[b]$  é o vetor contendo o lado direito de cada equação. A regra de Cramer diz que a solução da Eq. (2.41), se existir, é dada por:

$$x_j = \frac{\det(a'_j)}{\det(a)} \text{ para } j = 1, 2, \dots, n \quad (2.44)$$

onde  $a'_j$  é a matriz formada pela troca da  $j$ -ésima coluna da matriz  $[a]$  pelo vetor coluna contendo os lados direitos das equações do sistema original (2.42). Fica claro, a partir da Eq. (2.44), que soluções para (2.42) existem apenas se  $\det(a) \neq 0$ . A única maneira de  $\det(a)$  ser nulo ocorre se uma ou mais colunas ou linhas de  $[a]$  forem idênticas ou se uma ou mais colunas (ou linhas) de  $[a]$  forem linearmente dependentes de outras colunas (ou linhas).

#### 2.4.7 Normas

Na Seção 2.3, atribuiu-se aos vetores um módulo usualmente especificado pela Eq. (2.14). Na geometria Euclidiana, esse módulo pode servir como uma medida do tamanho de um vetor (não confundir com o tamanho ou número de elementos que ele contém). O módulo de um vetor é útil na comparação de vetores, tornando possível determinar se um vetor é maior que o outro. Tal medida equivalente para

**Exemplo 2-2:** Solução de um sistema de equações lineares usando a regra de Cramer

Obtenha a solução para o seguinte sistema de equações usando a regra de Cramer.

$$\begin{aligned} 2x + 3y - z &= 5 \\ 4x + 4y - 3z &= 3 \\ -2x + 3y - z &= 1 \end{aligned} \quad (2.45)$$

**SOLUÇÃO**

**Passo 1** Escreva o sistema de equações na forma matricial  $[a][x] = [b]$ .

$$\begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} \quad (2.46)$$

**Passo 2** Calcule o determinante da matriz de coeficientes.

$$\begin{aligned} \det(A) &= 2[(4 \times -1) - (-3 \times 3)] - 3[(4 \times -1) - (-3 \times -2)] - 1[(4 \times 3) - (4 \times -2)] \\ &= 2(5) - 3(-10) - 1(20) = 10 + 30 - 20 = 20 \end{aligned}$$

**Passo 3** Aplique a Eq. (2.44) para obter  $x$ ,  $y$  e  $z$ . Para se obter  $x$ , a matriz modificada  $a'_x$  é criada com a troca da primeira coluna de  $[a]$  por  $[b]$ .

$$x = \frac{\det\begin{pmatrix} 5 & 3 & -1 \\ 3 & 4 & -3 \\ 1 & 3 & -1 \end{pmatrix}}{20} = \frac{(5 \cdot 5) - (3 \cdot 0) - (1 \cdot 5)}{20} = 1$$

Da mesma forma, para se obter  $y$ , a matriz modificada  $a'_y$  é criada com a troca da segunda coluna de  $[a]$  por  $[b]$ .

$$y = \frac{\det\begin{pmatrix} 2 & 5 & -1 \\ 4 & 3 & -3 \\ -2 & 1 & -1 \end{pmatrix}}{20} = \frac{(20 \cdot 0) - (5 \cdot -10) - (1 \cdot 10)}{20} = 2$$

Finalmente, para se obter  $z$ , a matriz modificada  $a'_z$  é criada com a troca da terceira coluna de  $[a]$  por  $[b]$ .

$$z = \frac{\det\begin{pmatrix} 2 & 3 & 5 \\ 4 & 4 & 3 \\ -2 & 3 & 1 \end{pmatrix}}{20} = \frac{(2 \cdot -5) - (3 \cdot 10) - (5 \cdot 20)}{20} = 3$$

Para verificar a resposta, basta multiplicar a matriz de coeficientes  $[a]$  pela solução:

$$\begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 + 6 - 3 \\ 4 + 8 - 9 \\ -2 + 6 - 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix}$$

O lado direito é igual a  $[b]$ , o que confirma que a solução está correta.

o “módulo” de uma matriz também é útil na comparação de diferentes matrizes; ela é chamada de *Norma*, e representada como  $\|[a]\|$ . Não há uma forma única de se determinar o “módulo” ou norma de uma matriz. Várias definições de normas são apresentadas na Seção 4.9. A norma basicamente associa um número real a uma matriz (ou vetor).

Uma norma deve satisfazer a certas propriedades, já que ela representa para uma matriz algo análogo ao módulo ou comprimento de um vetor. Essas propriedades são:

- (1)  $\|[a]\| \geq 0$  e  $\|[a]\| = 0$  se e somente se  $[a] = [0]$  (isto é, se  $[a]$  é a matriz zero).
- (2) Para todos os números  $\alpha$ ,  $\|\alpha[a]\| = |\alpha|\|[a]\|$ .
- (3) Para quaisquer duas matrizes (ou vetores)  $[a]$  e  $[b]$ , a seguinte condição deve ser satisfeita:  $\|[a] + [b]\| \leq \|[a]\| + \|[b]\|$ .

A condição (1) diz que o “módulo” de uma matriz ou vetor, de acordo com a norma, deve ser uma grandeza positiva como qualquer comprimento usado para medir o tamanho de um vetor. A condição (2) diz que, para matrizes, assim como para vetores,  $\|[a]\|$  e  $\|[-a]\|$  têm o mesmo “módulo”. Isso é fácil de se ver no caso de vetores, já que o comprimento do vetor não muda com a inversão de sua direção. A condição (3) é tão somente a desigualdade triangular que é facilmente visualizada para vetores na geometria Euclidiana. As várias normas vetoriais e matriciais são discutidas em maior detalhe na Seção 4.10.

## 2.5 EQUAÇÕES DIFERENCIAIS ORDINÁRIAS (EDO)

Uma equação diferencial ordinária (EDO) é uma equação que contém uma variável dependente, uma variável independente e derivadas ordinárias da variável dependente. Se  $x$  é a variável independente e  $y$  é a variável dependente, uma EDO tem termos contendo  $x$ ,  $y$ ,  $\frac{dy}{dx}$ ,  $\frac{d^2y}{dx^2}$ , ...,  $\frac{d^n y}{dx^n}$ . As EDOs podem ser lineares ou não-lineares. Uma EDO linear é aquela em que a dependência em  $y$  e em suas derivadas é linear. Uma EDO *linear* pode ser escrita na seguinte *forma padronizada*, ou *canônica*:

$$a_{n+1}(x)\frac{d^n y}{dx^n} + a_n(x)\frac{d^{n-1}y}{dx^{n-1}} + \dots + a_3(x)\frac{d^2y}{dx^2} + a_2(x)\frac{dy}{dx} + a_1(x)y = r(x) \quad (2.47)$$

Note que todos os coeficientes na Eq. (2.47) são funções apenas da variável independente  $x$ . Exemplos de EDOs lineares são:

$$\frac{dy}{dx} = 10x$$

$$c\frac{dx}{dt} + kx = -m\frac{d^2x}{dt^2}$$

onde  $m$ ,  $k$  e  $c$  são constantes.

### EDOs homogêneas/não-homogêneas

Uma EDO pode ser homogênea ou não-homogênea. Quando escrita na forma padronizada (Eq. (2.47)), a EDO é **homogênea** se no lado direito  $r(x) = 0$ . Senão, se  $r(x) \neq 0$ , diz-se que a EDO é **não-homogênea**.

### Ordem de uma EDO

A ordem de uma EDO é determinada pela ordem da maior derivada que aparece na equação. A ordem de uma EDO pode fornecer informações importantes. Quando uma EDO é resolvida, constantes arbitrárias ou de integração aparecem na solução. O número de constantes a serem determinadas é igual à ordem da EDO. Por exemplo, a solução de uma EDO de segunda ordem tem duas constantes indeterminadas. Isso significa que duas restrições devem ser especificadas para que essas duas constantes indeterminadas possam ser obtidas. Quando a variável independente é a posição e as restrições são especificadas em duas diferentes posições, elas são chamadas de **condições de contorno**. Quando a variável independente é o tempo e as restrições são especificadas em um único instante de tempo, elas são chamadas de **condições iniciais**.

### EDO não-linear

Uma EDO é não-linear se os coeficientes da Eq. (2.47) forem funções de  $y$  ou de suas derivadas, se o lado direito  $r$  for uma função não-linear de  $y$ , ou se o termo linear  $a_1(x)y$  for trocado por uma função não-linear de  $y$ . As seguintes EDOs são exemplos de EDOs não-lineares:

$$\frac{d^2y}{dt^2} + \operatorname{sen} y = 4$$

$$y \frac{d^2y}{dt^2} + 3y = 8$$

$$\left(\frac{dy}{dt}\right) \frac{d^2y}{dt^2} + y = 9$$

$$\frac{d^2y}{dt^2} + 8y = \operatorname{tg} y$$

### Solução analítica de algumas importantes EDOs lineares

Certas EDOs lineares de primeira e segunda ordem aparecem recorrentemente em muitos ramos da ciência e da engenharia. Por causa de sua importância, as soluções dessas EDOs são dadas aqui como um lembrete. A **solução geral** de uma EDO **não-homogênea linear** é a soma da **solução homogênea** e da **solução particular**. A solução homogênea é a solução da EDO homogênea (isto é, da EDO com  $r(x)$  na Eq. (2.47) igualada a zero), e a solução particular é uma solução que, quando substituída na EDO, satisfaz ao lado direito da equação (isto é, leva a  $r(x)$ ). Apenas após a obtenção da solução geral, as restrições (isto é, as condições de contorno ou condições iniciais) devem ser substituídas para que as constantes indeterminadas sejam encontradas.

### Solução geral de uma EDO não-homogênea linear de primeira ordem

A solução geral de uma EDO não-homogênea linear de primeira ordem da forma:

$$\frac{dy}{dx} + P(x)y = Q(x) \quad (2.48)$$

é obtida com a multiplicação de ambos os lados da equação pelo seguinte fator de integração:

$$\mu(x) = e^{\int P(x)dx} \quad (2.49)$$

Quando isto é feito, a Eq. (2.48) pode ser escrita na seguinte forma integrável:

$$\frac{d}{dx}(y\mu) = Q(x)\mu(x) \quad (2.50)$$

Como  $P(x)$  e  $Q(x)$  são funções conhecidas, a Eq. (2.50) pode ser resolvida com a multiplicação de ambos os lados por  $dx$  e a integração:

$$y(x)\mu(x) = \int Q(x)\mu(x)dx + C_1 \quad (2.51)$$

Dividindo por  $\mu(x)$ , obtém-se:

$$y(x) = \frac{1}{\mu(x)} \int Q(x)\mu(x)dx + \frac{C_1}{\mu(x)} \quad (2.52)$$

A constante de integração  $C_1$  deve ser determinada com base em alguma restrição, que depende do problema em questão.

### Solução geral de uma EDO homogênea linear de segunda ordem com coeficientes constantes

Uma EDO homogênea linear de segunda ordem com coeficientes constantes pode ser escrita na forma:

$$\frac{d^2y}{dx^2} + b\frac{dy}{dx} + cy = 0 \quad (2.53)$$

onde  $b$  e  $c$  são constantes. A solução geral dessa equação é obtida com a substituição de  $y = e^{sx}$ . A equação resultante é chamada de equação característica:

$$s^2 + bs + c = 0 \quad (2.54)$$

A solução da Eq. (2.54) é obtida a partir da fórmula quadrática:

$$s = \frac{-b \pm \sqrt{b^2 - 4c}}{2} \quad (2.55)$$

A solução geral da Eq. (2.53) é, portanto:

$$y(x) = e^{-bx/2} \left[ C_1 e^{\frac{x}{2}\sqrt{b^2 - 4c}} + C_2 e^{-\frac{x}{2}\sqrt{b^2 - 4c}} \right] \quad (2.56)$$

onde  $C_1$  e  $C_2$  são constantes de integração determinadas a partir das restrições do problema particular sendo resolvido (isto é, condições iniciais ou de contorno).

Há dois casos especiais importantes na Eq. (2.56). No primeiro, o discriminante é positivo, isto é,  $b^2 > 4c$ , e a solução é igual à mostrada na Eq. (2.56). No segundo caso, o discriminante é negativo, isto é,  $b^2 < 4c$ , e a solução se torna:

$$y(x) = e^{-bx/2} \left[ C_1 e^{\frac{i}{2}x\sqrt{b^2 - 4c}} + C_2 e^{-\frac{i}{2}x\sqrt{b^2 - 4c}} \right] \quad (2.57)$$

onde  $i = \sqrt{-1}$ . Usando a fórmula de Euler,  $e^{iz} = \cos(z) + i\sin(z)$ , a Eq. (2.57) pode ser escrita em uma forma um pouco mais familiar:

$$\begin{aligned} y(x) = & e^{-bx/2} \left[ C_1 \cos\left(x \frac{\sqrt{4c - b^2}}{2}\right) + C_1 i \sin\left(x \frac{\sqrt{4c - b^2}}{2}\right) \right. \\ & \left. + C_2 \cos\left(x \frac{\sqrt{4c - b^2}}{2}\right) - C_2 i \sin\left(x \frac{\sqrt{4c - b^2}}{2}\right) \right] \end{aligned} \quad (2.58)$$

que pode ser combinada, levando a:

$$y(x) = e^{-bx/2} \left[ (C_1 + C_2) \cos\left(x \frac{\sqrt{4c - b^2}}{2}\right) + (C_1 - C_2) i \sin\left(x \frac{\sqrt{4c - b^2}}{2}\right) \right] \quad (2.59)$$

Como  $C_1$  e  $C_2$  são constantes arbitrárias,  $(C_1 + C_2)$  e  $(C_1 - C_2)$  também são constantes arbitrárias. Portanto, a Eq. (2.58) pode ser escrita como:

$$y(x) = e^{-bx/2} \left[ D_1 \sin\left(x \frac{\sqrt{4c - b^2}}{2}\right) + D_2 \cos\left(x \frac{\sqrt{4c - b^2}}{2}\right) \right] \quad (2.60)$$

onde  $D_1 = i(C_1 - C_2)$  e  $D_2 = C_1 + C_2$ .

Detalhes adicionais sobre métodos de solução analítica de EDOs encontram-se disponíveis em muitos livros de cálculo e de equações diferenciais. Muitas EDOs que aparecem em aplicações práticas, contudo, não podem ser resolvidas analiticamente. A sua solução requer o emprego de métodos numéricos.

## 2.6 FUNÇÕES DE DUAS OU MAIS VARIÁVEIS INDEPENDENTES

Uma função possui apenas uma variável dependente, mas pode ter uma, duas ou mais variáveis independentes. Por exemplo, a função  $z = f(x, y) = \frac{x^2}{2^2} + \frac{y^2}{3^2}$  (equação de um parabolóide elíptico) tem duas variáveis independentes ( $x$  e  $y$ ) e uma única variável dependente  $z$ . A função associa um único número  $z$  (**variável dependente**) a cada combinação de valores de  $x$  e  $y$  (**variáveis independentes**). Esta seção revisa vários tópicos relacionados ao cálculo da derivada de funções com duas ou mais variáveis independentes.

### 2.6.1 Definição da derivada parcial

Para uma função  $z = f(x, y)$ , a derivada parcial de  $f$  em relação a  $x$  é representada por  $\frac{\partial f}{\partial x}$ , ou  $f_x$ , e é definida por:

$$\frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \quad (2.61)$$

desde que, naturalmente, o limite exista. De forma similar, a derivada parcial de  $f$  em relação a  $y$  é representada por  $\frac{\partial f}{\partial y}$ , ou  $f_y$ , e é definida por:

$$\frac{\partial f}{\partial y} = \lim_{\Delta y \rightarrow 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \quad (2.62)$$

novamente, desde que o limite exista. Na prática, as definições das Eqs. (2.61) e (2.62) implicam que, se  $z = f(x, y)$ , então  $f_x$  é determinada com o cálculo da derivada da função em relação a  $x$  assumindo que  $y$  seja uma constante. Da mesma forma,  $f_y$  é determinada com o cálculo da derivada da função em relação a  $y$  assumindo que  $x$  seja uma constante.

### Derivadas parciais de ordem elevada

É possível calcular derivadas parciais de segunda ordem, de terceira ordem e de ordem mais elevada de funções de muitas variáveis (desde que elas existam). Por exemplo, a função  $f(x, y)$  de duas variáveis tem duas derivadas parciais,  $f_x$  e  $f_y$ . Cada uma delas tem duas derivadas parciais. A função  $f_x$  pode ser derivada em

relação a  $x$  para fornecer  $f_{xx} = \frac{\partial^2 f}{\partial x^2}$ , ou em relação a  $y$  para fornecer  $f_{xy} = \frac{\partial^2 f}{\partial y \partial x}$ .

Da mesma forma,  $f_y$  pode ser derivada em relação a  $y$  para fornecer  $f_{yy} = \frac{\partial^2 f}{\partial y^2}$ , ou em relação a  $x$  para fornecer  $f_{yx} = \frac{\partial^2 f}{\partial x \partial y}$ . As segundas derivadas parciais são cha-

madas de derivadas parciais mistas de  $f(x, y)$ . Se a função  $f(x, y)$  e suas derivadas parciais mistas forem contínuas, então pode-se mostrar que a ordem de derivação

não é importante, isto é,  $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$ .

### 2.6.2 Regras da cadeia

A **diferencial total** de uma função de duas variáveis, por exemplo  $f(x, y)$ , é dada por:

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy \quad (2.63)$$

A Eq. (2.63) é válida mesmo que as variáveis  $x$  e  $y$  não sejam independentes uma da outra. Só é necessário que as derivadas parciais da Eq. (2.63) sejam con-

tínuas. Note que a Eq. (2.63) pode ser facilmente generalizada para funções com mais de duas variáveis.

A função  $f(x, y)$  pode depender de seus argumentos  $x$  e  $y$  de diferentes maneiras. Primeiramente,  $x$  e  $y$  podem depender de uma única variável independente, tal como  $t$ . Em outras palavras,  $f(x, y) = f(x(t), y(t))$ . Em segundo lugar,  $x$  e  $y$  podem depender de duas outras variáveis independentes, como  $u$  e  $v$ . Nesse caso,  $f(x, y) = f(x(u, v), y(u, v))$ . Em terceiro lugar,  $x$  pode ser a variável independente e  $y$  pode depender de  $x$ , levando a  $f(x, y) = f(x, y(x))$ . Em quarto lugar, a função  $f$  pode depender de três variáveis,  $x$ ,  $y$  e  $z$ , com  $z$  dependendo de  $x$  e  $y$ , que são independentes entre si. O cálculo da derivada nesses casos é considerado a seguir.

- (1) Se as variáveis  $x$  e  $y$  dependerem cada uma delas de uma única variável  $t$ , então  $f(x, y)$  pode ser considerada função apenas da variável independente  $t$ . Nesse caso, a derivada total de  $f$  com relação a  $t$  pode ser simplesmente determinada por:

$$\frac{df}{dt} = \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} \quad (2.64)$$

- (2) Se as variáveis  $x$  e  $y$  dependerem cada uma delas de duas variáveis independentes  $u$  e  $v$ , então a derivada parcial de  $f$  em relação a  $v$  com  $u$  constante, isto é,  $\left. \frac{\partial f}{\partial v} \right|_u$  é obtida com:

$$\left. \frac{\partial f}{\partial v} \right|_u = \left. \frac{\partial f}{\partial x} \frac{\partial x}{\partial v} \right|_u + \left. \frac{\partial f}{\partial y} \frac{\partial y}{\partial v} \right|_u \quad (2.65)$$

Usualmente, o fato de se manter  $u$  constante durante o cálculo da derivada parcial em relação a  $v$  fica subentendido, e o subscrito  $u$  é retirado da Eq. (2.65):

$$\frac{\partial f}{\partial v} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial v} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial v} \quad (2.66)$$

- (3) Se  $y$  depender de  $x$ , a função  $f(x, y)$  é realmente uma função de  $x$ , e a derivada total de  $f$  em relação a  $x$  pode ser definida:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial x} \quad (2.67)$$

- (4) Se  $f$  é função de  $x$ ,  $y$  e  $z$ , e se  $z$  depender de  $x$  e  $y$ , que são independentes, então a derivada parcial de  $f$  em relação a  $x$  é:

$$\left. \frac{\partial f}{\partial x} \right|_y = \frac{\partial f}{\partial x} + \left. \frac{\partial f}{\partial z} \frac{\partial z}{\partial x} \right|_y \quad (2.68)$$

ou, simplesmente:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial x} \quad (2.69)$$

onde o termo  $\frac{\partial f}{\partial y}$  não aparece porque é multiplicado por  $\frac{\partial z}{\partial x}$ , que é nulo já que  $x$

e  $y$  são variáveis independentes.

Da mesma forma, a derivada parcial de  $f$  em relação a  $y$  é:

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial y} + \frac{\partial f \partial z}{\partial z \partial y} \quad (2.70)$$

### 2.6.3 O Jacobiano

O Jacobiano é uma grandeza que aparece na solução de sistemas de equações não-lineares simultâneas. Se  $f_1(x, y) = a$  e  $f_2(x, y) = b$  são duas equações simultâneas que precisam ser resolvidas para  $x$  e  $y$ , onde  $a$  e  $b$  são constantes, então a matriz Jacobiana é definida como:

$$[J] = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} \quad (2.71)$$

O **determinante Jacobiano**, ou simplesmente **Jacobião**,  $J(f_1, f_2)$ , é apenas o determinante da matriz Jacobiana:

$$J(f_1, f_2) = \det \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \left( \frac{\partial f_1}{\partial x} \right) \left( \frac{\partial f_2}{\partial y} \right) - \left( \frac{\partial f_1}{\partial y} \right) \left( \frac{\partial f_2}{\partial x} \right) \quad (2.72)$$

O Jacobiano pode ser facilmente generalizado para um sistema de  $n$  equações:

$$J(f_1, f_2, \dots, f_n) = \det \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \quad (2.73)$$

## 2.7 EXPANSÃO DE FUNÇÕES EM SÉRIE DE TAYLOR

A expansão de uma função em série de Taylor é uma maneira de encontrar o valor dessa função em um ponto próximo a algum ponto conhecido, isto é, um ponto onde se conhece o valor da função. A função é representada pela soma de termos de uma série convergente. Em alguns casos (se a função for um polinômio), a série de Taylor pode fornecer o valor exato da função. Em muitos casos, contudo, a soma de um número infinito de termos é necessária para que o valor exato seja obtido. Se apenas alguns poucos termos forem utilizados, o valor da função obtido a partir da série de Taylor é uma aproximação. A expansão de funções em série de Taylor é muito usada em métodos numéricos.

### 2.7.1 Série de Taylor para uma função de uma variável

Dada uma função  $f(x)$  derivável  $(n+1)$  vezes em um intervalo contendo um ponto  $x = x_0$ , o teorema de Taylor diz que, para cada  $z$  contido nesse intervalo, existe um valor  $x = \xi$  entre  $x$  e  $x_0$  tal que:

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0) \frac{df}{dx} \Big|_{x=x_0} + \frac{(x - x_0)^2}{2!} \frac{d^2f}{dx^2} \Big|_{x=x_0} + \frac{(x - x_0)^3}{3!} \frac{d^3f}{dx^3} \Big|_{x=x_0} \\ &\quad + \dots + \frac{(x - x_0)^n}{n!} \frac{d^n f}{dx^n} \Big|_{x=x_0} + R_n(x) \end{aligned} \quad (2.74)$$

onde  $R_n$ , chamado de resíduo, é dado por:

$$R_n = \frac{(x - x_0)^{n+1}}{(n+1)!} \frac{d^{n+1}f}{dx^{n+1}} \Big|_{x=\xi} \quad (2.75)$$

A prova deste teorema pode ser encontrada em qualquer livro de cálculo.

Note que, para  $n = 1$ , o teorema de Taylor se reduz a:

$$f(x) = f(x_0) + (x - x_0) \frac{df}{dx} \Big|_{x=\xi} \quad \text{ou} \quad \frac{df}{dx} \Big|_{x=\xi} = \frac{f(x) - f(x_0)}{(x - x_0)} \quad (2.76)$$

que é o teorema do valor médio para as derivadas (Eq. (2.4)), dado na Seção 2.2.

O valor do resíduo,  $R_n$ , não pode ser calculado na prática porque o valor de  $\xi$  não é conhecido. Quando a série de Taylor é usada na obtenção do valor aproximado de uma função no ponto  $x$ , dois ou mais termos são usados. A precisão da aproximação depende de quantos termos da série de Taylor forem usados e na proximidade do ponto  $x$  em relação ao ponto  $x_0$ . A precisão aumenta à medida que  $x$  se torna mais próximo de  $x_0$  e também com o aumento do número de termos. Isso é ilustrado na Exemplo 2-3, onde a série de Taylor é usada como uma aproximação para a função  $y = \sin(x)$ .

### 2.7.2 Série de Taylor para funções de duas variáveis

A expansão em série de Taylor de uma função de duas variáveis é feita da mesma forma que para uma função de uma variável independente, exceto pelo fato de a diferenciação envolver derivadas parciais. A fórmula de Taylor para a expansão de  $f(x, y)$  em torno de um ponto  $(x_0, y_0)$  é dada por:

$$\begin{aligned} f(x, y) &= f(x_0, y_0) + \frac{1}{1!} \left[ (x - x_0) \frac{\partial f}{\partial x} \Big|_{x_0, y_0} + (y - y_0) \frac{\partial f}{\partial y} \Big|_{x_0, y_0} \right] + \\ &\quad \frac{1}{2!} \left[ (x - x_0)^2 \frac{\partial^2 f}{\partial x^2} \Big|_{x_0, y_0} + 2(x - x_0)(y - y_0) \frac{\partial^2 f}{\partial x \partial y} \Big|_{x_0, y_0} + (y - y_0)^2 \frac{\partial^2 f}{\partial y^2} \Big|_{x_0, y_0} \right] + \\ &\quad + \dots + \frac{1}{n!} \left[ \sum_{k=0}^n \frac{n!}{k!(n-k)!} (x - x_0)^k (y - y_0)^{n-k} \frac{\partial^n f}{\partial x^k \partial y^{n-k}} \Big|_{x_0, y_0} \right] \end{aligned} \quad (2.78)$$

**Exemplo 2-3: Aproximação de uma função com a expansão em série de Taylor**

Aproxime a função  $y = \sin(x)$  empregando a expansão em série de Taylor em torno do ponto  $x = 0$ , usando dois, quatro e seis termos.

(a) Em cada caso, calcule o valor aproximado da função em  $x = \frac{\pi}{12}$  e em  $x = \frac{\pi}{2}$ .

(b) Usando o MATLAB, trace a função e as suas três aproximações para  $0 \leq x \leq \pi$ .

**SOLUÇÃO**

As primeiras cinco derivadas da função  $y = \sin(x)$  são:

$$y' = \cos(x), \quad y'' = -\sin(x), \quad y^{(3)} = -\cos(x), \quad y^{(4)} = \sin(x) \quad \text{e} \quad y^{(5)} = \cos(x)$$

Em  $x = 0$ , os valores dessas derivadas são:

$$y' = 1, \quad y'' = 0, \quad y^{(3)} = -1, \quad y^{(4)} = 0 \quad \text{e} \quad y^{(5)} = 1$$

Substituindo essa informação e  $y(0) = \sin(0) = 0$  na Eq. (2.74), tem-se:

$$y(x) = 0 + x + 0 - \frac{x^3}{3!} + 0 + \frac{x^5}{5!} \quad (2.77)$$

(a) Para  $x = \frac{\pi}{12}$ , o valor exato da função é  $y = \sin\left(\frac{\pi}{12}\right) = \frac{1}{4}(\sqrt{6} + \sqrt{2}) = 0,2588190451$

Os valores aproximados usando dois, quatro ou seis termos na expansão em série de Taylor são:

Dois termos na Eq. (2.77):  $y(x) = x = \frac{\pi}{12} = 0,2617993878$

Quatro termos na Eq. (2.77):  $y(x) = x - \frac{x^3}{3!} = \frac{\pi}{12} - \frac{(\pi/12)^3}{3!} = 0,2588088133$

Seis termos na Eq. (2.77):  $y(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} = \frac{\pi}{12} - \frac{(\pi/12)^3}{3!} + \frac{(\pi/12)^5}{5!} = 0,2588190618$

Para  $x = \frac{\pi}{2}$ , o valor exato da função é  $y = \sin\left(\frac{\pi}{2}\right) = 1$

O uso de dois termos na Eq. (2.77) resulta em:  $y(x) = x = \frac{\pi}{2} = 1,570796327$

O uso de quatro termos na Eq. (2.77) resulta em:  $y(x) = x - \frac{x^3}{3!} = \frac{\pi}{2} - \frac{(\pi/2)^3}{3!} = 0,9248322293$

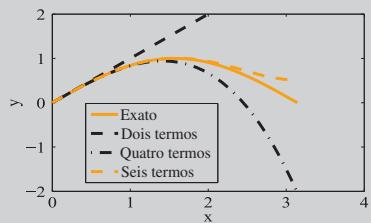
O uso de seis termos na Eq. (2.77) resulta em:

$$y(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} = \frac{\pi}{2} - \frac{(\pi/2)^3}{3!} + \frac{(\pi/2)^5}{5!} = 1,004524856$$

(b) Usando um programa no MATLAB, listado no seguinte arquivo texto, a função e as três aproximações foram calculadas no domínio  $0 \leq x \leq \pi$ . O programa também traça os resultados.

```
x = linspace(0,pi,40);
y = sin(x);
y2 = x;
y4 = x - x.^3/factorial(3);
y6 = x - x.^3/factorial(3) + x.^5/factorial(5);
plot(x,y,'r',x,y2,'k--',x,y4,'k-.',x,y6,'r--')
axis([0,4,-2,2])
legend('Exato','Dois termos','Quatro termos','Seis termos')
xlabel('x'); ylabel('y')
```

O gráfico produzido pelo programa é mostrado à direita. Os resultados de ambas as letras mostram, conforme esperado, que a aproximação da função usando a série de Taylor é mais precisa quando mais termos são usados e quando o ponto no qual se deseja calcular a função é mais próximo do ponto em torno do qual a função é expandida.



## 2.8 PROBLEMAS

### Problemas de solução manual

*Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos.*

- 2.1 Aplique o teorema do valor intermediário para mostrar que o polinômio  $f(x) = -x^4 + 2x + 4$  tem uma raiz no intervalo  $[1, 2]$ .
- 2.2 Aplique o teorema do valor intermediário para mostrar que a função  $f(x) = \cos x - x^2$  tem uma raiz no intervalo  $[0, \pi/2]$ .
- 2.3 Use a definição formal da derivada (Eq. (2.2)) e a terminologia associada para mostrar que a derivada de  $\cos x$  é  $-\sin x$ .
- 2.4 Use a definição da derivada (Eq. (2.2)) para mostrar que:
 
$$(a) \frac{d}{dx}(u(x)v(x)) = u\frac{dv}{dx} + v\frac{du}{dx}.$$

$$(b) \frac{d}{dx}\left(\frac{u(x)}{v(x)}\right) = \frac{v\frac{du}{dx} - u\frac{dv}{dx}}{v^2}.$$
- 2.5 Use a regra da cadeia (Eq. (2.3)) para determinar a derivada segunda de  $f(x) = \operatorname{sen}(e^x)$ . (Dica: defina  $u(x) = e^x$  e então aplique a regra da cadeia.)
- 2.6 Como patrulheiro rodoviário, você está participando de uma blitz. Um carro passa por sua viatura andando a 88 km/h. Um minuto mais tarde, seu parceiro em outra viatura localizada 1,6 km adiante vê o mesmo carro andando a 80 km/h. Usando o teorema do valor médio para as derivadas (Eq. (2.4)), mostre que o carro deve ter excedido o limite de velocidade de 88 km/h em algum ponto durante o minuto que ele percorreu entre as duas viaturas.
- 2.7 A tosse provoca a contração da glote e força o ar a passar com uma velocidade crescente. Suponha que a velocidade  $v$  do ar durante a tosse seja dada por:

$$v = C(R - r)r^2$$

onde  $C$  é uma constante,  $R$  é o raio normal da glote (isto é, quando não há tosse), que também é uma constante, e  $r$  é o raio variável da glote durante a tosse. Determine o raio da glote que produz o fluxo de ar com a maior velocidade durante a tosse.

**2.8** Usando o teorema do valor médio para as integrais, determine o valor médio da função  $f(x) = \sin x$  no intervalo  $[0, \pi]$ . Mostre que o produto desse valor médio pela largura do intervalo é igual à área sob a curva.

**2.9** Use o segundo teorema fundamental do cálculo e a regra da cadeia para determinar  $\frac{d}{dx} \left[ \int_1^{\sqrt{x}} \sin t dt \right]$ .

**2.10** Dado o seguinte sistema de equações

$$\begin{aligned} 2x + 2y - 3z &= 2 \\ -1x + 3y + 2z &= 0 \\ 3x + y - 3z &= 1 \end{aligned}$$

Determine as incógnitas  $x, y$  e  $z$  usando a regra de Cramer.

**2.11** A distribuição de temperaturas em um sólido é dada por  $T(x, y) = e^{-y^2} \sin x$ . O fluxo de calor na direção  $y$  é dado por  $q_y = -k \frac{\partial T}{\partial y}$ . Usando a definição da derivada parcial (Eq. (2.62)), determine o fluxo de calor  $q_y$  no ponto  $(1, 2)$ .

**2.12** Dada a função  $f(x) = (\sin x)(\cos y) \ln z$ , determine a derivada total em relação a  $x$ ,  $\frac{df}{dx}$ , no ponto  $(1, 2, 3)$ .

**2.13** Calcule o determinante da seguinte matriz:

$$\begin{bmatrix} 2 & 0 & 0 & 3 \\ 1 & 1 & 1 & 0 \\ 5 & 1 & 1 & 9 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

**2.14** Determine a ordem das seguintes EDOs e também se elas são lineares, não-lineares, homogêneas ou não-homogêneas.

(a)  $EI \frac{d^2y}{dx^2} = M(x)$ , onde  $E$  e  $I$  são constantes e  $M(x)$  é uma função conhecida de  $x$ .

(b)  $\frac{d\Phi}{dt} + \frac{\Phi}{\tau} = \frac{CF(t)}{\tau}$ , onde  $C$  e  $\tau$  são constantes, e  $F(t)$  é uma função conhecida de  $t$ .

(c)  $\frac{dh}{dt} = \frac{-r^2 \sqrt{2gh}}{2hR - h^2}$ , onde  $r, g$  e  $R$  são constantes.

**2.15** Na transformação de coordenadas cartesianas  $(x, y)$  para coordenadas polares  $(r, \theta)$ , as seguintes relações são válidas:  $x(r, \theta) = r\cos\theta$  e  $y(r, \theta) = r\sin\theta$ .

Determine a matriz Jacobiana  $[J] = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{bmatrix}$ . Qual é o determinante do Jacobiano?

**2.16** Escreva a expansão em série de Taylor da função  $f(x) = \ln(x + n)$  em torno de  $x = 0$ , onde  $n \neq 0$  é uma constante conhecida.

**2.17** Escreva a expansão em série de Taylor da função  $f(x, y) = e^{-x^2} \sin y$  em torno do ponto  $(1, 3)$ .

### Problemas de programação no MATLAB

*Resolva os seguintes problemas no MATLAB. Não use funções residentes para executar as operações solicitadas.*

**2.18** Escreva um programa que avalie a derivada da função  $f(x) = x^3 \sin x$  no ponto  $x = 3$  usando a Eq. (2.2). O valor da derivada deve ser calculado dezesseis vezes usando  $a = 3$  e dezesseis valores de  $x$ ,  $x = 2,6; 2,65; \dots, 2,95; \dots, 3,35; 3,4$ . O programa também deve traçar um gráfico com os valores das derivadas em função de  $x$ .

**2.19** Escreva no MATLAB uma função que avalie a integral definida de uma expressão matemática utilizando a soma de Riemann (ver Eq. (2.7)). Para a função, use o nome e os argumentos a seguir:  $I = \text{SomaRiemann}(\text{'NomeFun'}, a, b)$ . '*NomeFun*' é uma *string* com o nome de uma função que calcula o valor da expressão a ser integrada em um dado valor de  $x$ .  $a$  e  $b$  são os limites de integração, e  $I$  é o valor da integral. A soma de Riemann é calculada com a divisão do intervalo de integração  $[a, b]$  em dez subintervalos. Use *SomaRiemann* para avaliar a integral definida  $\int_0^1 e^x dx$ . Compare o resultado com o valor exato da integral: 1,71828.

**2.20** Escreva no MATLAB uma função que realize a multiplicação de duas matrizes  $[c] = [a][b]$ . Para a função, use o nome e os argumentos a seguir:  $C = \text{MatrizMult}(A, B)$ . Os argumentos de entrada  $A$  e  $B$  são as matrizes a serem multiplicadas. O argumento de saída  $C$  é o resultado. Não use a multiplicação de matrizes do MATLAB. A função *MatrizMult* deve primeiro verificar se as duas matrizes podem ser multiplicadas. Caso não possam, a saída  $C$  deve ser a mensagem “As matrizes não podem ser multiplicadas porque o número de linhas em  $[b]$  não é igual ao número de colunas em  $[a]$ ”. Use *MatrizMult* para realizar a multiplicação ilustrada na Fig. 2.14.

**2.21** Escreva no MATLAB uma função que determine o produto cruzado de dois vetores  $\vec{W} = \vec{V} \otimes \vec{U}$ . Para a função, use o nome e os argumentos a seguir:  $W = \text{Cruzado}(V, U)$ . Os argumentos de entrada  $V$  e  $U$  são os vetores a serem multiplicados. O argumento de saída  $W$  é o resultado (vetor com três elementos). Use *Cruzado* para determinar o produto cruzado dos vetores  $v = 3i + 6,5j - 2k$  e  $u = -5i + 4k + 10k$ .

**2.22** Escreva no MATLAB uma função que decomponha uma matriz  $m \times n$  em um vetor de comprimento  $1 \times m \cdot n$ . O vetor consiste nas linhas da matriz

colocadas em ordem. Por exemplo, se a matriz é  $\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ , então o vetor

é  $[a_{11} \ a_{12} \ a_{13} \ a_{21} \ a_{22} \ a_{23} \ a_{31} \ a_{32} \ a_{33}]$ . Para a função, use o nome e os argumentos a seguir: `v=decompor(A)`, onde o argumento de entrada A é uma matriz de qualquer tamanho, e o argumento de saída v é o vetor.

Use a função na janela de comandos do MATLAB para decompor a matriz:

$$\begin{bmatrix} 2 & -5 & 1 & 0 & 6 \\ -9 & 4 & 6 & 10 & -4 \\ 11 & 0 & -12 & 7 & 3 \end{bmatrix}$$

**2.23** Escreva no MATLAB uma função que determine a transposta de uma matriz  $m \times n$ . Para a função, use o nome e os argumentos a seguir: `At=transp(A)`, onde o argumento de entrada A é uma matriz de qualquer tamanho, e o argumento de saída At é a transposta de A.

Use a função na janela de comandos do MATLAB para determinar a transposta da matriz:

$$\begin{bmatrix} 2 & -5 & 1 & 0 & 6 \\ -9 & 4 & 6 & 10 & -4 \\ 11 & 0 & -12 & 7 & 3 \end{bmatrix}$$

**2.24** Escreva no MATLAB uma função que calcule o determinante de uma matriz quadrada ( $n \times n$ ), onde  $n$  pode ser 2, 3, ou 4. Para a função, use o nome e os argumentos a seguir: `D=Determinante(A)`. O argumento de entrada A é a matriz cujo determinante deve ser calculado. A função Determinante deve primeiro verificar se a matriz é quadrada. Se não for, a saída D deve ser a mensagem “A matriz deve ser quadrada”.

Use Determinante na janela de comandos do MATLAB para calcular o determinante das duas matrizes a seguir:

$$(a) \begin{bmatrix} 2 & 2 & -3 \\ -1 & 3 & 2 \\ 3 & 1 & -1 \end{bmatrix} \quad (b) \begin{bmatrix} 2 & 1 & 4 & -2 \\ -3 & 4 & 2 & -1 \\ 3 & 5 & -2 & 1 \\ -2 & 3 & 2 & 4 \end{bmatrix}$$

### Problemas na matemática, ciência e engenharia

Resolva os problemas a seguir usando o ambiente MATLAB.

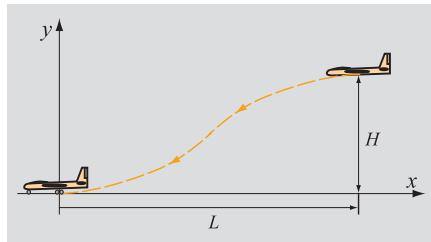
**2.25** Uma importante aplicação envolvendo o diferencial total de uma função de muitas variáveis é a estimativa da incerteza.

- (a) A potência elétrica dissipada por uma resistência  $R$  está relacionada à tensão  $V$  e à resistência por  $P = \frac{V^2}{R}$ . Escreva o diferencial total  $dP$  em termos dos diferenciais  $dV$  e  $dR$ , usando a Eq. (2.63).

- (b)  $dP$  é interpretado como a incerteza na potência,  $dV$  como a incerteza na tensão e  $dR$  como a incerteza na resistência. Usando a resposta obtida na letra (a), determine o máximo percentual de incerteza na potência  $P$  para  $V = 200$  V com uma incerteza de 2%, e  $R = 5000 \Omega$  com uma incerteza de 3%.

- 2.26** Uma aeronave começa a sua aterrissagem a uma distância  $x = L$  ( $x = 0$  é a marca na qual o avião toca o solo) e a uma altura  $H$ . Suponha que um polinômio cúbico com a forma a seguir seja usado para descrever a aterrissagem:

$$y = ax^3 + bx^2 + cx + d$$



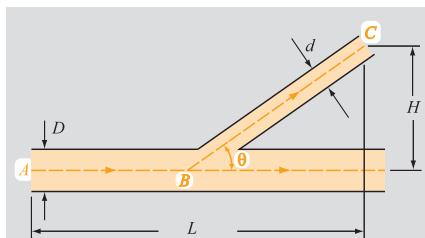
onde  $y$  é a altitude e  $x$  é a distância horizontal da aeronave. A aeronave começa a descer de uma altura inicial constante e pousa em uma altura final constante.

- (a) Obtenha os coeficientes  $a, b, c$  e  $d$ .
- (b) Se a aeronave mantiver uma velocidade constante ( $\frac{dx}{dt} = u = \text{constante}$ ) e o módulo da aceleração ( $\frac{d^2y}{dt^2}$ ) não puder ultrapassar uma constante  $A$ , mostre que  $\frac{6Hu^2}{L^2} \leq A$ .
- (c) Se  $A = 0,09 \text{ m/s}^2$ ,  $H = 3050 \text{ m}$  e  $u = 240 \text{ km/h}$ , a qual a distância do aeroporto o piloto deve começar a descida?

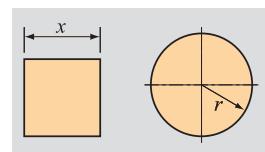
- 2.27** Uma artéria que deriva de outra artéria mais importante tem uma resistência ao fluxo de sangue dada por:

$$R_{\text{fluxo}} = K \left( \frac{L - H \cot \theta}{D^4} + \frac{H \csc \theta}{d^4} \right)$$

onde  $R_{\text{fluxo}}$  é a resistência ao fluxo de sangue da artéria maior para a artéria menor ao longo do caminho  $ABC$  (veja o diagrama),  $d$  é o diâmetro da artéria menor, em ramificação,  $D$  é o diâmetro da artéria maior,  $\theta$  é o ângulo que a veia menor faz com o eixo da artéria maior e  $L$  e  $H$  são as distâncias mostradas na figura. Determine o ângulo  $\theta$  que minimiza a resistência ao fluxo em termos de  $d$  e  $D$ .



- 2.28** Uma corda com comprimento de 10 m deve ser usada para envolver uma área quadrada com lado  $x$  e uma área circular com raio  $r$ . Que comprimento de corda deve ser usado para o quadrado e para o círculo se a área total delimitada pelas duas formas geométricas deve ser máxima?



**2.29** Há sistemas mecânicos, elétricos e químicos que são descritos pelas mesmas expressões matemáticas representando um movimento harmônico amortecido forçado de segunda ordem. A equação diferencial resultante obtida após a aplicação do balanço de forças ou da conservação do momento é da forma:

$$m \frac{d^2x}{dt^2} + \gamma \frac{dx}{dt} + kx = A_0 \operatorname{sen}(\omega t)$$

onde  $x$  é o deslocamento,  $t$  é o tempo,  $m$  é a massa,  $\gamma$  é o coeficiente de amortecimento,  $k$  é a constante da mola (força de restauração),  $A_0$  é a amplitude da força aplicada e  $\omega$  é a freqüência dessa força.

- (a) Determine a ordem da EDO e também se ela é linear, não-linear, homogênea ou não-homogênea.
- (b) Determine manualmente a solução homogênea para a EDO.
- (c) Determine manualmente a solução particular para a EDO. Determine  $x(t)$  após um longo período de tempo ( $t \rightarrow \infty$ ). Isso é às vezes chamado de resposta em “regime permanente”, mesmo que na realidade a resposta varie com o tempo.
- (d) Usando o MATLAB, faça um gráfico com a amplitude máxima de  $x(t)$  a partir do regime permanente em função da freqüência de excitação  $\omega$  ( $0 \leq \omega \leq 5$  rad/s) para  $A_0 = 1$  N,  $k = 1$  N/m e  $m = 1$  kg, para três valores de  $\gamma$  ( $\gamma = 0,5$ ,  $\gamma = 1,0$  e  $\gamma = 2,0$  N.s/m) (três gráficos na mesma figura). Discuta os resultados. O que acontece em  $\omega = \sqrt{\frac{k}{m}}$  quando  $\gamma = 0$ ?

## Resolvendo Equações Não-Lineares

### Tópicos centrais

- Estimação de erros em soluções numéricas (3.2)
- Método da Bisseção (3.3)
- Método Regula Falsi (3.4)
- Método de Newton (3.5)
- Método da Secante (3.6)
- Método da iteração de ponto fixo (3.7)

Uso de funções residentes do MATLAB para resolver equações não-lineares (3.8)

### Tópicos complementares

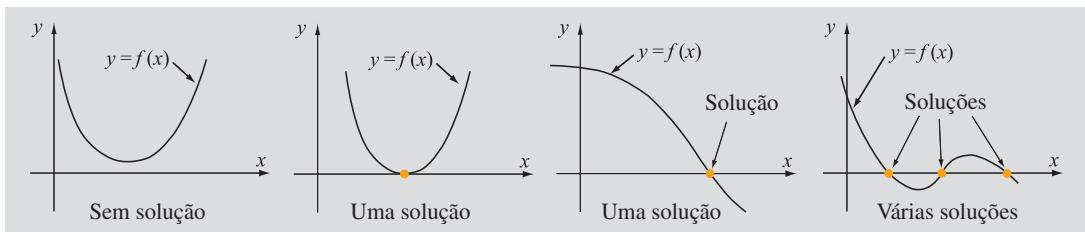
- Equações com múltiplas raízes (3.9)
- Sistemas de equações não-lineares (3.10)

### 3.1 FUNDAMENTOS

Equações precisam ser resolvidas em todas as áreas da ciência e da engenharia. Uma equação de uma única variável pode ser escrita na forma:

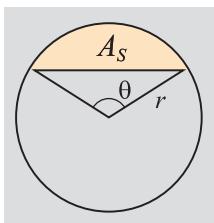
$$f(x) = 0 \quad (3.1)$$

A solução dessa equação (também chamada de *raiz*) é um valor numérico de  $x$  que satisfaz à equação. Graficamente, conforme mostrado na Fig. 3-1, a solução é o ponto onde a função  $y=f(x)$  cruza ou toca o eixo  $x$ . Uma equação pode não ter solução ou ter uma ou várias raízes (possivelmente muitas).



**Figura 3-1** Ilustração de equações com nenhuma, uma ou várias soluções.

Quando a equação é simples, o valor de  $x$  pode ser determinado analiticamente. Esse é o caso quando se escreve  $x$  explicitamente após a aplicação de operações matemáticas, ou quando uma fórmula conhecida (como a fórmula usada na resolução de equações quadráticas) pode ser usada para determinar o valor exato de  $x$ . Em muitas situações, no entanto, é impossível determinar analiticamente a



**Figura 3-2** Segmento de um círculo.

raiz de uma equação. Por exemplo, a área do segmento  $A_S$  de um círculo com raio  $r$  (área sombreada na Fig. 3-2) é dada por:

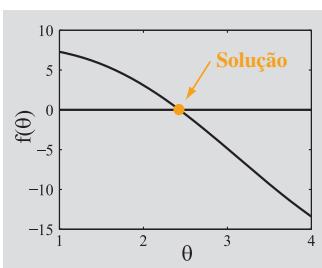
$$A_S = \frac{1}{2}r^2(\theta - \sin \theta) \quad (3.2)$$

Para determinar o ângulo  $\theta$  se  $A_S$  e  $r$  são conhecidos, a Eq. (3.2) deve ser resolvida para  $\theta$ . Obviamente, não é possível escrever  $\theta$  explicitamente em termos de  $A_S$  e  $r$ , e a equação não pode ser resolvida analiticamente.

A solução numérica de uma equação  $f(x) = 0$  é um valor de  $x$  que satisfaz à equação de forma aproximada. Isso significa que, quando  $x$  é substituído na equação, o valor de  $f(x)$  é próximo de zero, mas não exatamente zero. Por exemplo, para determinar o ângulo  $\theta$  para um círculo com  $r = 3$  m e  $A_S = 8 \text{ m}^2$ , a Eq. (3.2) pode ser escrita na forma:

$$f(\theta) = 8 - 4,5(\theta - \sin \theta) = 0 \quad (3.3)$$

Um gráfico de  $f(\theta)$  (ver Fig. 3-3) mostra que a solução está entre 2 e 3. A substituição de  $\theta = 2,4$  rad na Eq. (3.3) leva a  $f(\theta) = 0,2396$ , e a solução  $\theta = 2,43$  rad resulta em  $f(\theta) = 0,003683$ . Obviamente, a última é uma solução mais precisa, embora não seja exata. É possível determinar valores de  $\theta$  que levem a valores de  $f(\theta)$  mais próximos de zero, mas é impossível determinar um valor numérico de  $\theta$  que leve  $f(\theta)$  exatamente a zero. Quando uma equação é resolvida numericamente, é necessário selecionar a precisão desejada para a solução.

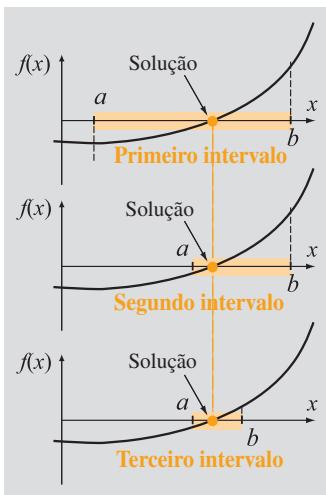


**Figura 3-3** Gráfico de  $f(\theta) = 8 - 4,5(\theta - \sin \theta)$ .

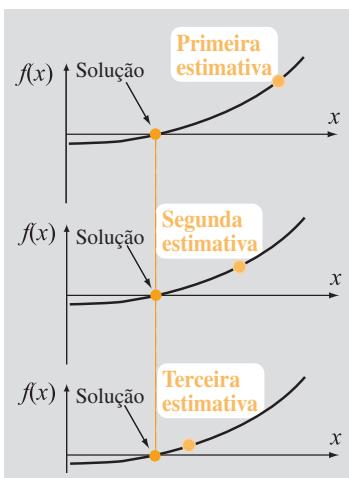
### Abordagens usadas na solução numérica de equações

O processo de solução numérica de uma equação é diferente do procedimento usado na obtenção de uma solução analítica. Uma solução analítica é obtida com a dedução de uma expressão que leva a um valor numérico exato. Uma solução numérica é obtida em um processo que começa com a determinação de uma solução aproximada e é seguido de um procedimento numérico no qual se determina uma solução mais precisa. A solução numérica inicial de uma equação na forma  $f(x) = 0$  pode ser estimada com o traçado de um gráfico  $f(x)$  versus  $x$  e com a verificação do ponto onde o gráfico cruza o eixo  $x$ . Também é possível escrever e executar um programa de computador que procure um domínio que contenha uma solução. Tal programa busca uma solução com a avaliação de  $f(x)$  em diferentes valores de  $x$ . Ele começa em um valor de  $x$  e então muda esse valor em pequenos incrementos. Uma mudança no sinal de  $f(x)$  indica a existência de uma raiz dentro do último incremento. Em muitos casos, quando a equação que se resolve está relacionada a alguma aplicação na ciência ou na engenharia, a faixa de valores de  $x$  que inclui a solução pode ser estimada e usada no traçado do gráfico inicial de  $f(x)$ , ou na busca numérica de um domínio pequeno que contenha uma solução. Quando uma equação tem mais de uma raiz, a solução numérica indica uma raiz de cada vez.

Os métodos numéricos usados para resolver equações podem ser divididos em dois grupos: **métodos de confinamento** e **métodos abertos**. Nos métodos de



**Figura 3-4** Ilustração de um método de confinamento.



**Figura 3-4** Ilustração de um método aberto.

confinamento, ilustrados na Fig. 3-4, identifica-se um intervalo que inclui a solução. Por definição, os pontos finais do intervalo são os limites superior e inferior da solução. Então, usando um esquema numérico, o tamanho do intervalo é reduzido sucessivamente até que a distância entre os pontos finais seja menor que a precisão desejada para a solução. Nos métodos abertos, ilustrados na Fig. 3-5, assume-se uma estimativa inicial para a solução (um ponto). O valor dessa tentativa inicial para a solução deve ser próximo à solução real. Então, usando um esquema numérico, valores melhores (mais precisos) são calculados para a solução. Métodos de confinamento sempre convergem para uma solução. Métodos abertos são usualmente mais eficientes, mas às vezes podem não levar à solução.

Conforme mencionado anteriormente, já que métodos numéricos não são exatos, é necessário estimar os erros envolvidos. Várias opções são apresentadas na Seção 3.2. As Seções

3.3 a 3.7 descrevem quatro métodos numéricos utilizados na determinação da raiz de uma única equação. Dois métodos de confinamento, o método da bisseção e o método regula falsi, são apresentados nas Seções 3.3 e 3.4, respectivamente. Três métodos abertos, o método de Newton, o método da secante e o método da iteração de ponto fixo, são introduzidos nas três seções seguintes. A Seção 3.8 descreve o emprego de funções residentes do MATLAB na obtenção de soluções numéricas, e a Seção 3.9 discute como lidar com equações que possuem múltiplas raízes. A última seção deste capítulo (3.10) lida com métodos numéricos para a solução de sistemas de equações não-lineares. A necessidade de resolver tais sistemas surge em muitos problemas na ciência e na engenharia, e também no emprego de métodos numéricos para resolver equações diferenciais ordinárias (ver Seção 9.3).

## 3.2 ESTIMAÇÃO DE ERROS EM SOLUÇÕES NUMÉRICAS

Como soluções numéricas não são exatas, algum critério deve ser aplicado para determinar se uma solução estimada é suficientemente precisa. Várias medidas podem ser usadas para estimar a precisão de uma solução aproximada. A escolha de uma dessas medidas depende da aplicação em questão e fica a cargo de quem resolve a equação.

Seja  $x_{TS}$  a solução verdadeira (exata) tal que  $f(x_{TS}) = 0$ , e seja  $x_{NS}$  uma solução numérica aproximada tal que  $f(x_{NS}) = \varepsilon$  (onde  $\varepsilon$  é um número pequeno). Quatro medidas podem ser consideradas para se estimar o erro:

**Erro real:** O erro real é a diferença entre a solução exata,  $x_{TS}$ , e a solução numérica,  $x_{NS}$ :

$$ErroReal = x_{TS} - x_{NS} \quad (3.4)$$

Infelizmente, no entanto, o erro real não pode ser calculado porque geralmente a solução exata não é conhecida.

**Tolerância em  $f(x)$ :** Em vez de considerar o erro na solução, é possível considerar o desvio de  $f(x_{NS})$  em relação a zero (o valor de  $f(x)$  no ponto  $x_{TS}$  é obviamente zero). A tolerância em  $f(x)$  é definida como o valor absoluto da diferença entre  $f(x_{TS})$  e  $f(x_{NS})$ :

$$TolerânciaEmf = |f(x_{TS}) - f(x_{NS})| = |0 - \varepsilon| = |\varepsilon| \quad (3.5)$$

A tolerância em  $f(x)$  é o valor absoluto da função em  $x_{NS}$ .

**Tolerância na solução:** A tolerância é a máxima quantidade na qual a solução exata pode desviar de uma solução numérica aproximada. A tolerância é útil na estimativa do erro quando métodos de confinamento são usados na determinação da solução numérica. Nesse caso, se é sabido que a solução está contida no domínio  $[a, b]$ , pode-se assumir que a solução numérica seja o ponto central desse intervalo:

$$x_{NS} = \frac{a + b}{2} \quad (3.6)$$

mais ou menos uma tolerância que é igual à metade da distância entre  $a$  e  $b$ :

$$Tolerância = \left| \frac{b - a}{2} \right| \quad (3.7)$$

**Erro relativo:** Se  $x_{NS}$  é uma solução numérica estimada, então o **erro relativo real** é dado por:

$$ErroRelativoReal = \left| \frac{x_{TS} - x_{NS}}{x_{TS}} \right| \quad (3.8)$$

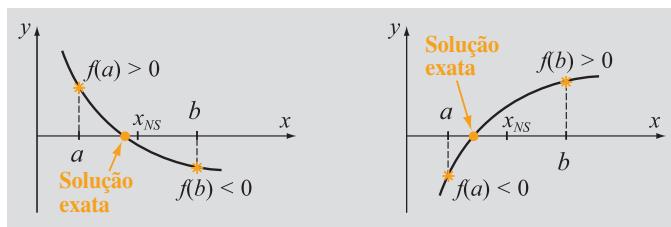
Esse erro relativo real não pode ser calculado, já que a solução  $x_{TS}$  não é conhecida. No entanto, é possível calcular o **erro relativo estimado** quando se tem duas estimativas numéricas para a solução. Esse é o caso quando soluções numéricas são calculadas iterativamente, onde em cada nova iteração se obtém uma solução mais precisa. Se  $x_{NS}^{(n)}$  é a solução numérica estimada na última iteração e  $x_{NS}^{(n-1)}$  é a solução numérica estimada na penúltima iteração, então o erro relativo estimado pode ser definido por:

$$ErroRelativoEstimado = \left| \frac{x_{NS}^{(n)} - x_{NS}^{(n-1)}}{x_{NS}^{(n-1)}} \right| \quad (3.9)$$

Quando as soluções numéricas estimadas se aproximam da solução exata, prevê-se que a diferença  $x_{NS}^{(n)} - x_{NS}^{(n-1)}$  seja pequena em comparação com o valor de  $x_{NS}^{(n)}$  e o erro relativo estimado se torna aproximadamente igual ao erro real relativo.

### 3.3 MÉTODO DA BISSEÇÃO

O método da bisseção é um método de confinamento usado para se obter a solução de uma equação na forma  $f(x) = 0$  quando se sabe que, dentro de um dado intervalo  $[a, b]$ ,  $f(x)$  é contínua e a equação possui uma solução. Quando esse é o caso,  $f(x)$  tem sinais opostos nos pontos finais do intervalo. Conforme mostrado na Fig. 3-6, se  $f(x)$  é contínua e tem uma solução entre os pontos  $x = a$  e  $x = b$ , então ou  $f(a) > 0$  e  $f(b) < 0$  ou  $f(a) < 0$  e  $f(b) > 0$ . Em outras palavras, se há uma solução entre  $x = a$  e  $x = b$ , então  $f(a)f(b) < 0$ .



**Figura 3-6** Solução de  $f(x) = 0$  entre  $x = a$  e  $x = b$ .

O processo de solução usando o método da bisseção é ilustrado na Fig. 3-7. Ele começa com a determinação dos pontos  $a$  e  $b$  que definem um intervalo onde

existe uma solução. Tal intervalo é encontrado ou com o traçado de um gráfico de  $f(x)$  e a identificação de um cruzamento por zero, ou com o exame da função buscando uma mudança de sinal. O ponto central do intervalo,  $x_{NS1}$ , é então tomado como sendo a primeira estimativa da solução numérica. A solução exata está contida ou na seção entre  $a$  e  $x_{NS1}$  ou na seção entre os pontos  $x_{NS1}$  e  $b$ . Se a solução numérica não for suficientemente precisa, define-se um novo intervalo que contenha a solução exata, e seu ponto central é escolhido como a nova (segunda) estimativa da solução numérica. O processo continua até que a solução numérica seja suficientemente precisa de acordo com o critério selecionado.

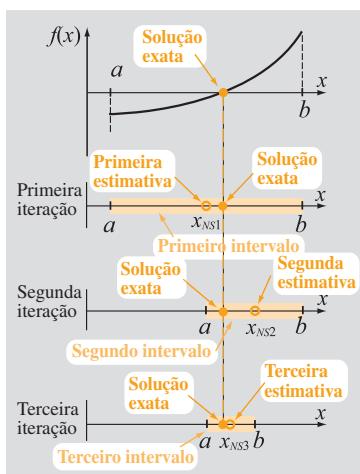
O procedimento (ou algoritmo) para se obter uma solução numérica com o método da bisseção é resumido a seguir.

#### Algoritmo para o método da bisseção

1. Escolha o primeiro intervalo encontrando os pontos  $a$  e  $b$  entre os quais existe uma solução. Isso significa que  $f(a)$  e  $f(b)$  têm sinais diferentes, de forma que  $f(a)f(b) < 0$ . Os pontos podem ser determinados a partir de um gráfico de  $f(x)$  versus  $x$ .

2. Calcule a primeira estimativa da solução numérica  $x_{NS1}$  usando:

$$x_{NS1} = \frac{(a + b)}{2}$$



**Figura 3-7** Método da bisseção.

3. Determine se a solução exata está entre  $a$  e  $x_{NS1}$ , ou entre  $x_{NS1}$  e  $b$ . Isso é feito com a verificação do sinal do produto  $f(a) \cdot f(x_{NS1})$ :

Se  $f(a) \cdot f(x_{NS1}) < 0$ , a solução exata está entre  $a$  e  $x_{NS1}$ .

Se  $f(a) \cdot f(x_{NS1}) > 0$ , a solução exata está entre  $x_{NS1}$  e  $b$ .

4. Selecione o subintervalo que contém a solução exata ( $a$  até  $x_{NS1}$ , ou  $x_{NS1}$  até  $b$ ) como o novo intervalo  $[a, b]$  e volte para o passo 2.

Os passos 2 a 4 são repetidos até que a tolerância especificada seja satisfeita ou um determinado limite de erro seja atingido.

### Quando o processo da bisseção deve ser interrompido?

Ideialmente, o processo da bisseção deve ser interrompido com a obtenção da solução exata. Isso significa que o valor de  $x_{NS}$  deve ser tal que  $f(x_{NS}) = 0$ . Na realidade, contudo, conforme discutido na Seção 3.1, a solução exata em geral não pode ser obtida computacionalmente. Na prática, portanto, o processo deve ser interrompido quando o erro estimado, de acordo com alguma das medidas listadas na Seção 3.1, for menor que algum valor predeterminado. A escolha do critério de interrupção pode depender do problema a ser resolvido.

Um programa escrito para determinar a solução numérica com a aplicação do método da bisseção no MATLAB é mostrado no exemplo a seguir (a escrita do programa na forma de uma função é deixada como um problema para casa).

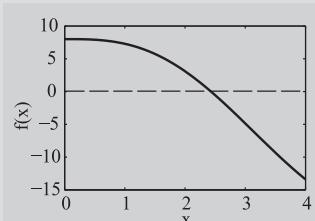
### Exemplo 3-1: Solução de uma equação não-linear usando o método da bisseção

Escreva um programa no MATLAB que determine a solução da equação  $8 - 4,5(x - \sin x) = 0$  usando o método da bisseção. A solução deve ter uma tolerância menor que 0,001 rad. Crie uma tabela que mostre os valores de  $a$ ,  $b$ ,  $x_{NS}$ ,  $f(x_{NS})$  e a tolerância para cada iteração do processo de bisseção.

#### SOLUÇÃO

Para se obter a localização aproximada da solução, um gráfico da função  $f(x) = 8 - 4,5(x - \sin x)$  é feito usando o comando `fplot` do MATLAB. O gráfico (Fig. 3-8) mostra que a solução está entre  $x = 2$  e  $x = 3$ . O intervalo inicial é escolhido como  $a = 2$  e  $b = 3$ .

O programa de MATLAB escrito para resolver este problema é apresentado a seguir.



**Figura 3-8** Gráfico da função  $f(x) = 8 - 4,5(x - \sin x)$ .

#### Programa 3-1 Arquivo texto. Método da bisseção.

```
clear all
F = inline('8 - 4.5*(x - sin(x))');
a = 2; b = 3; imax = 20; tol = 0.001;
Fa = F(a); Fb = F(b);
if Fa*Fb > 0
    disp('Erro: A função tem o mesmo sinal nos pontos a e b.')
else
    disp('iteration   a       b   (xNS) Solução   f(xNS)   Tolerância')
```

Define  $f(x)$  como uma função.

Atribui valores iniciais a  $a$  e  $b$ ; define número máximo de iterações e tolerância.

Interrompe o programa se a função tiver o mesmo sinal nos pontos  $a$  e  $b$ .

```

for i = 1:imax
    xNS = (a + b)/2;
    toli = (b - a)/2;
    FxNS = F(xNS);
    fprintf('%3i %11.6f %11.6f %11.6f %11.6f\n', i, a, b, xNS, FxNS, toli)
    if FxNS == 0
        fprintf('Uma solução exata x=%11.6f foi obtida',xNS)
        break
    end
    if toli < tol
        break
    end
    if i == imax
        fprintf('Solução não obtida em %i iterações',imax)
        break
    end
    if F(a)*FxNS < 0
        b = xNS;
    else
        a = xNS;
    end
end
end

```

**Calcula a solução numérica da iteração,  $x_{NS}$ .**

**Calcula a tolerância atual.**

**Calcula o valor de  $f(x_{NS})$  da iteração.**

**Interrompe o programa se a solução exata,  $f(x) = 0$ , for determinada.**

**Interrompe as iterações se a tolerância da iteração for menor que a tolerância desejada.**

**Interrompe as iterações se a solução não tiver sido obtida e o número de iterações atingir  $i_{max}$ .**

**Determina se a solução exata está entre  $a$  e  $x_{NS}$ , ou entre  $x_{NS}$  e  $b$ , e seleciona  $a$  e  $b$  para a próxima iteração.**

Quando o programa é executado, aparece a seguinte tabela na janela de comandos:

iteração	a	b	(xNS)	Solução	f(xNS)	Tolerância
1	2.000000	3.000000	2.500000	-0.556875	0.500000	
2	2.000000	2.500000	2.250000	1.376329	0.250000	
3	2.250000	2.500000	2.375000	0.434083	0.125000	
4	2.375000	2.500000	2.437500	-0.055709	0.062500	
5	2.375000	2.437500	2.406250	0.190661	0.031250	
6	2.406250	2.437500	2.421875	0.067838	0.015625	
7	2.421875	2.437500	2.429688	0.006154	0.007813	
8	2.429688	2.437500	2.433594	-0.024755	0.003906	
9	2.429688	2.433594	2.431641	-0.009295	0.001953	
10	2.429688	2.431641	2.430664	-0.001569	0.000977	

Solução numérica.

Valor da função para a solução numérica.

A última tolerância (satisfaz a tolerância prescrita).

A saída mostra que a solução com a tolerância desejada é obtida na 10ª iteração.

### Notas adicionais a respeito do método da bisseção

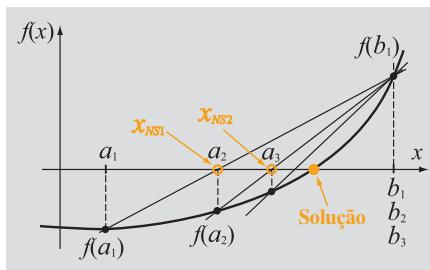
- O método sempre converge para uma resposta, desde que uma raiz esteja contida no intervalo inicial  $[a, b]$ .

- O método pode falhar quando a função é tangente ao eixo  $x$ , não o cruzando em  $f(x) = 0$ .
- A convergência do método é lenta em comparação com outros métodos.

### 3.4 MÉTODO REGULA FALSI

O método regula falsi (também chamado de método da falsa posição ou de interpolação linear) é um método de confinamento usado para se obter a solução de uma equação na forma  $f(x) = 0$  quando se sabe que, dentro de um dado intervalo  $[a, b]$ ,  $f(x)$  é contínua e a equação possui uma solução.

Conforme ilustrado na Fig. 3-9, a solução tem início com a obtenção de um intervalo  $[a_1, b_1]$  que confine a solução. Os valores da função nos pontos finais são  $f(a_1)$  e  $f(b_1)$ .



**Figura 3-9** Método regula falsi.

Os pontos finais são então conectados por uma linha reta, e a primeira estimativa da solução numérica,  $x_{NS1}$ , é o ponto onde a linha reta cruza o eixo  $x$ . Isso contrasta com o método da bissecção, onde o ponto central do intervalo foi escolhido como solução. Para a segunda iteração, define-se um novo intervalo  $[a_2, b_2]$ . Esse novo intervalo corresponde à subseção do primeiro intervalo que contém a solução. Ele é  $[a_1, x_{NS1}]$  ( $a_1$  atribuído a  $a_2$  e  $x_{NS1}$  a  $b_2$ ) ou  $[x_{NS1}, b_1]$  ( $x_{NS1}$  atribuído a  $a_2$  e  $b_1$  a  $b_2$ ). Os pontos finais do segundo intervalo são em seguida conectados por uma linha reta, e o ponto onde essa nova reta cruza o eixo  $x$  se torna a segunda solução estimada,  $x_{NS2}$ . Para a terceira iteração, um novo subintervalo  $[a_3, b_3]$  é selecionado, e as iterações continuam da mesma forma até que a solução numérica seja considerada suficientemente precisa.

Para um dado intervalo  $[a, b]$ , a equação da linha reta que conecta os pontos  $(b, f(b))$  e  $(a, f(a))$  é dada por:

$$y = \frac{f(b) - f(a)}{b - a}(x - b) + f(b) \quad (3.10)$$

O ponto  $x_{NS}$  onde a reta cruza o eixo  $x$  é determinado pela substituição de  $y = 0$  na Eq. (3.10) e a solução dessa equação para  $x$ :

$$x_{NS} = \frac{af(b) - bf(a)}{f(b) - f(a)} \quad (3.11)$$

O procedimento (ou algoritmo) usado para se obter uma solução com o método regula falsi é quase o mesmo empregado no método da bisseção.

### Algoritmo para o método regula falsi

1. Escolha o primeiro intervalo encontrando os pontos  $a$  e  $b$  entre os quais existe uma solução. Isso significa que  $f(a)$  e  $f(b)$  têm sinais diferentes, de forma que  $f(a)f(b) < 0$ . Os pontos podem ser determinados a partir de um gráfico de  $f(x)$  versus  $x$ .
2. Calcule a primeira estimativa da solução numérica  $x_{NS1}$  usando a Eq. (3.11)
3. Determine se a solução exata está entre  $a$  e  $x_{NS1}$ , ou entre  $x_{NS1}$  e  $b$ . Isso é feito com a verificação do sinal do produto  $f(a) \cdot f(x_{NS1})$ :  
Se  $f(a) \cdot f(x_{NS1}) < 0$ , a solução exata está entre  $a$  e  $x_{NS1}$ .  
Se  $f(a) \cdot f(x_{NS1}) > 0$ , a solução exata está entre  $x_{NS1}$  e  $b$ .
4. Selecione o subintervalo que contém a solução ( $a$  até  $x_{NS1}$ , ou  $x_{NS1}$  até  $b$ ) como o novo intervalo  $[a, b]$  e volte para o passo 2.

Os passos 2 a 4 são repetidos até que uma tolerância especificada ou um determinado limite de erro sejam atingidos.

### Quando as interações devem ser interrompidas?

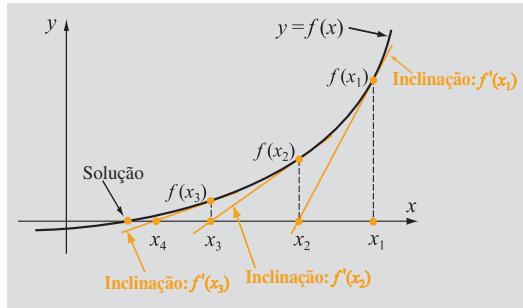
As interações são interrompidas quando o erro estimado, de acordo com alguma das medidas listadas na Seção 3.2, for menor que algum valor predeterminado.

### Notas adicionais a respeito do método regula falsi

- O método sempre converge para uma resposta, desde que uma raiz esteja contida no intervalo inicial  $[a, b]$ .
- Freqüentemente, como no caso mostrado na Fig. 3-9, a função no intervalo  $[a, b]$  é côncava para cima ou para baixo. Nesse caso, um dos pontos finais do intervalo permanece o mesmo em todas as iterações, enquanto o outro avança em direção à raiz. Em outras palavras, a solução numérica avança em direção à raiz apenas de um lado. A convergência em direção à solução poderia ser mais rápida se o outro ponto final também se “movesse” em direção à raiz. Várias modificações têm sido introduzidas no método regula falsi para fazer com que o subintervalo nas iterações sucessivas se aproxime da raiz de ambos os lados (veja Problema 3.18).

## 3.5 MÉTODO DE NEWTON

O método de Newton (também chamado de método de Newton-Raphson) é um esquema usado para se obter a solução numérica de uma equação na forma  $f(x) = 0$ , onde  $f(x)$  é contínua e diferenciável e sua equação possui uma solução próxima a um ponto dado. O método é ilustrado na Fig. 3-10.



**Figura 3-10** Método de Newton.

O processo de solução começa com a escolha do ponto  $x_1$  como a primeira estimativa da solução. A segunda estimativa,  $x_2$ , é obtida a partir do cruzamento com o eixo  $x$  da reta tangente a  $f(x)$  no ponto  $(x_1, f(x_1))$ . A estimativa seguinte,  $x_3$ , é a interseção com o eixo  $x$  da reta tangente a  $f(x)$  no ponto  $(x_2, f(x_2))$ , e assim por diante. Matematicamente, na primeira iteração, a inclinação  $f'(x_1)$  da tangente no ponto  $(x_1, f(x_1))$  é dada por:

$$f'(x_1) = \frac{f(x_1) - 0}{x_1 - x_2} \quad (3.12)$$

Resolvendo a Eq. (3.12) para  $x_2$ , obtém-se:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \quad (3.13)$$

A Eq. (3.13) pode ser generalizada para que a “próxima” solução  $x_{i+1}$  seja obtida a partir da solução atual:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (3.14)$$

A Eq. (3.14) é a fórmula iterativa geral do método de Newton. Ela é chamada de fórmula iterativa porque a solução é obtida com a aplicação repetida da Eq. (3.14) em cada valor sucessivo de  $i$ .

O método de Newton também pode ser deduzido a partir da série de Taylor. A expansão em série de Taylor de  $f(x)$  em torno de  $x_1$  é dada por:

$$f(x) = f(x_1) + (x - x_1)f'(x_1) + \frac{1}{2!}(x - x_1)^2f''(x_1) + \dots \quad (3.15)$$

Se  $x_2$  é uma solução da equação  $f(x) = 0$  e  $x_1$  é um ponto próximo a  $x_2$ , então:

$$f(x_2) = 0 = f(x_1) + (x_2 - x_1)f'(x_1) + \frac{1}{2!}(x_2 - x_1)^2f''(x_1) + \dots \quad (3.16)$$

Considerando apenas os dois primeiros termos da série, uma solução aproximada pode ser determinada resolvendo a Eq. (3.16) para  $x_2$ :

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \quad (3.17)$$

O resultado é o mesmo dado pela Eq. (3.13). Na iteração seguinte, a expansão em série de Taylor é escrita em torno do ponto  $x_2$ , e uma solução aproximada  $x_3$  é calculada. A fórmula geral é igual àquela dada pela Eq. (3.14).

### Algoritmo para o método de Newton

1. Escolha um ponto  $x_1$  como tentativa inicial da solução.
2. Para  $i = 1, 2, \dots$ , até que o erro seja menor que um valor especificado, calcule  $x_{i+1}$  usando a Eq. (3.14).

### Quando as iterações devem ser interrompidas?

Idealmente, as iterações devem ser interrompidas quando uma solução exata é obtida. Isso significa que o valor de  $x$  deve ser tal que  $f(x) = 0$ . Geralmente, conforme discutido na Seção 3.1, a solução exata não pode ser obtida computacionalmente. Na prática, portanto, as iterações são interrompidas quando o erro estimado for menor que algum valor predeterminado. Uma tolerância não pode ser calculada como se faz no método da bisseção, já que os limites não são conhecidos. Duas estimativas de erro tipicamente utilizadas pelo método de Newton são:

**Erro relativo estimado:** As iterações são interrompidas quando o erro relativo estimado (Eq. (3.9)) é menor que um valor especificado  $\varepsilon$ :

$$\left| \frac{x_{i+1} - x_i}{x_i} \right| \leq \varepsilon \quad (3.18)$$

**Tolerância em  $f(x)$ :** As iterações são interrompidas quando o valor absoluto de  $f(x_i)$  é menor que algum número  $\delta$ :

$$|f(x_i)| \leq \delta \quad (3.19)$$

A programação do método de Newton é muito simples. Uma função (chamada de `NewtonRaiz`) que faz a determinação da raiz de  $f(x) = 0$  no MATLAB é apresentada na Fig. 3-11. O programa consiste em uma repetição na qual a solução seguinte  $X_{i+1}$  é calculada a partir da solução atual  $X_i$  usando a Eq. (3.14). As repetições param se o erro for pequeno o suficiente de acordo com a Eq. (3.18). Para evitar a situação em que as repetições podem continuar indefinidamente (porque a solução não converge ou por causa de algum erro de programação), o número de passos na repetição é limitado a `imax`. As funções  $f(x)$  e  $f'(x)$  (que aparecem na Eq. (3.14)) devem ser fornecidas em separado como funções definidas pelo usuário. Seus nomes são digitados nos argumentos de `NewtonRaiz` como variáveis *string*. O Exemplo 3-2 mostra como usar a Eq. (3.14), e como a função `NewtonRaiz` pode ser usada para resolver um problema específico.

### Notas a respeito do método de Newton

- O método de Newton, quando bem-sucedido, converge rapidamente. A não convergência usualmente ocorre porque o ponto de partida não está suficientemente próximo da solução. Problemas de convergência ocorrem tipicamente quando o valor de  $f'(x)$  é próximo de zero na vizinhança da solução (onde  $f(x) = 0$ ). É possível mostrar que o método de Newton converge se a função  $f(x)$  e as suas derivadas primeira e segunda,  $f'(x)$  e  $f''(x)$ , forem contínuas, se

**Programa 3-2 Função definida pelo usuário. Método de Newton.**

```

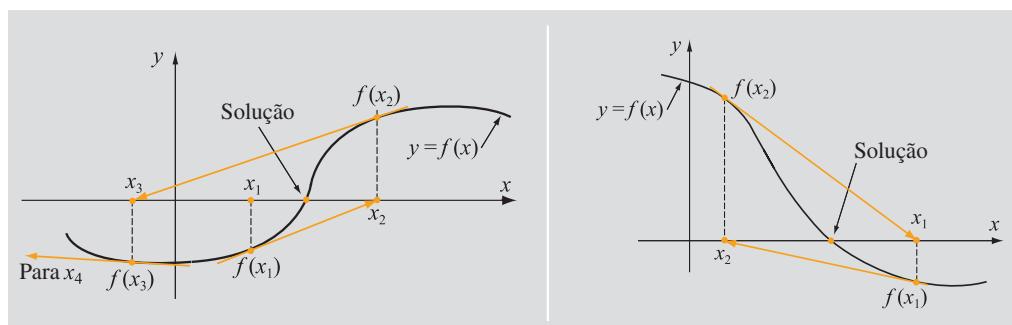
function Xs = NewtonRaiz(Fun,FunDer,Xest,Err,imax)
% NewtonRaiz determina a raiz de Fun = 0 na vizinhança do ponto Xest usando
% o método de Newton.
% Variáveis de entrada:
% Fun Nome (string) da função que calcula Fun para um dado x.
% FunDer Nome (string) da função que calcula a derivada
% de Fun para um dado x.
% Xest Estimativa inicial da função.
% Err Erro máximo.
% imax Número máximo de iterações
% Variável de saída:
% Xs Solução

for i = 1:imax
    Xi = Xest - feval(Fun,Xest)/feval(FunDer,Xest); Eq. (3.14).
    if abs((Xi - Xest)/Xest) < Err Eq. (3.18).
        Xs = Xi;
        break
    end
    Xest = Xi;
end
if i == imax
    fprintf('A solução não foi obtida em %i iterações.\n',imax)
    Xs = ('Sem resposta');
end

```

**Figura 3-11** Arquivo de função no MATLAB para resolver equações usando o método de Newton.

$f'(x)$  for diferente de zero na solução, e se o ponto de partida  $x_1$  estiver próximo da solução exata. Ilustrações de dois casos em que o método de Newton não converge (isto é, diverge) são mostradas na Fig. 3-12.



**Figura 3-12** Casos em que o método de Newton diverge.

### Exemplo 3-2: Solução de equações usando o método de Newton

Obtenha a solução da equação  $8 - 4,5(x - \sin x) = 0$  (a mesma equação resolvida no Exemplo 3-1) usando o método de Newton nas duas maneiras a seguir:

- Usando uma calculadora não-programável, calcule no papel as duas primeiras iterações usando seis algarismos significativos.
- Usando a função `NewtonRaiz` listada na Fig. 3-11 no MATLAB, use um erro relativo máximo de 0,0001 e 10 para o número máximo de iterações.

Nas duas maneiras, use  $x = 2$  como tentativa inicial da solução.

### SOLUÇÃO

No presente problema,  $f(x) = 8 - 4,5(x - \sin x)$  e  $f'(x) = -4,5(1 - \cos x)$ .

- Para iniciar as iterações,  $f(x)$  e  $f'(x)$  são substituídas na Eq. (3.14):

$$x_{i+1} = x_i - \frac{8 - 4,5(x_i - \sin x_i)}{-4,5(1 - \cos x_i)} \quad (3.20)$$

Na primeira iteração,  $i = 1$  e  $x_1 = 2$ , e a Eq. (3.20) resulta em:

$$x_2 = 2 - \frac{8 - 4,5(2 - \sin(2))}{-4,5(1 - \cos(2))} = 2,48517 \quad (3.21)$$

Na segunda iteração,  $i = 2$  e  $x_2 = 2,48517$ , e a Eq. (3.20) resulta em:

$$x_3 = 2,48517 - \frac{8 - 4,5(2,48517 - \sin(2,48517))}{-4,5(1 - \cos(2,48517))} = 2,43099 \quad (3.22)$$

- Para a resolver a equação no MATLAB usando a função `NewtonRaiz`, o usuário deve criar funções para  $f(x)$  e  $f'(x)$ . As duas funções, chamadas de `FunExemplo2` e `FunDerExemplo2`, são:

```
function y = FunExemplo2(x)
y = 8 - 4.5*(x - sin(x));
```

e

```
function y = FunDerExemplo2(x)
y = -4.5 + 4.5*cos(x);
```

Uma vez criadas e salvas as funções acima, a função `NewtonRaiz` pode ser usada na janela de comandos do MATLAB:

```
>> format long
>> xSolution = NewtonRaiz('FunExemplo2','FunDerExemplo2',2,0.0001,10)
xSolution =
2.43046574172363
```

Uma comparação entre os resultados obtidos nas letras *a* e *b* mostra que os quatro primeiros algarismos da solução (2,430) são obtidos na segunda iteração (na letra *b*, o processo de solução é interrompido na quarta iteração; veja o Problema 3.19). Isso mostra, conforme mencionado anteriormente, que o método de Newton usualmente apresenta uma convergência rápida. No Exemplo 3-1 (método da bissecção), os quatro primeiros algarismos apenas foram obtidos após 10 bissecções.

- Uma função  $f'(x)$ , que é a derivada da função  $f(x)$ , deve ser substituída na fórmula iterativa, a Eq (3.14). Em muitos casos, é simples escrever a derivada, mas às vezes a sua determinação pode ser difícil. Quando não se tem uma expressão para a derivada, é possível obter a inclinação numericamente ou empregar o método da secante (Seção 3.6), que é de certa forma similar ao método de Newton mas não requer uma expressão para a derivada.

A seguir, o Exemplo 3-3 ilustra o efeito que o ponto de partida pode ter na solução numérica com o método de Newton.

### Exemplo 3-3: Convergência do método de Newton

Determine a solução da equação  $\frac{1}{x} - 2 = 0$  usando o método de Newton. Como ponto de partida (estimativa inicial da solução), use:

$$(a) x = 1,4, \quad (b) x = 1 \quad \text{e} \quad (c) x = 0,4.$$

#### SOLUÇÃO

A equação pode ser resolvida analiticamente, e a solução exata é  $x = 0,5$ .

Para uma solução numérica empregando o método de Newton, a função  $f(x) = \frac{1}{x} - 2$  e sua derivada  $f'(x) = -\frac{1}{x^2}$  são substituídas na Eq. (3.14):

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{\frac{1}{x_i} - 2}{-\frac{1}{x_i^2}} = 2(x_i - x_i^2) \quad (3.23)$$

(a) Quando o ponto de partida das iterações é  $x_1 = 1,4$ , as duas iterações seguintes, usando a Eq. (3.23), são:

$$x_2 = 2(x_1 - x_1^2) = 2(1,4 - 1,4^2) = -1,12 \quad \text{e} \quad x_3 = 2(x_2 - x_2^2) = 2[(-1,12) - (-1,12)^2] = -4,7488$$

Esses resultados indicam que o método de Newton diverge. Este caso é ilustrado na Fig. 3-13a.

(b) Quando o ponto de partida das iterações é  $x = 1$ , as duas iterações seguintes, usando a Eq. (3.23), são:

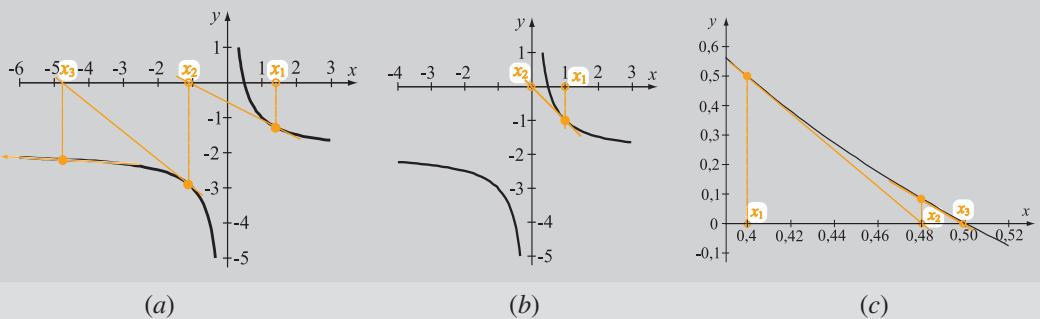
$$x_2 = 2(x_1 - x_1^2) = 2(1 - 1^2) = 0 \quad \text{e} \quad x_3 = 2(x_2 - x_2^2) = 2(0 - 0^2) = 0$$

Nesses resultados, a solução aparentemente converge para  $x = 0$ , que não é uma solução. Em  $x = 0$ , a função é, na realidade, indefinida (este é um ponto de singularidade). Obtém-se uma solução a partir da Eq. (3.23) porque a equação foi simplificada. Este caso é ilustrado na Fig. 3-13b.

(c) Quando o ponto de partida das iterações é  $x = 0,4$ , as duas iterações seguintes, usando a Eq. (3.23), são:

$$x_2 = 2(x_1 - x_1^2) = 2(0,4 - 0,4^2) = 0,48 \quad \text{e} \quad x_3 = 2(x_2 - x_2^2) = 2(0,48 - 0,48^2) = 0,4992$$

Neste caso, ilustrado na Fig. 3-13c, o método de Newton converge para a solução correta. Este exemplo também mostra que, se o ponto de partida for suficientemente próximo da solução exata, o método de Newton converge.



**Figura 3-13** Solução com o método de Newton usando diferentes pontos de partida.

### 3.6 MÉTODO DA SECANTE

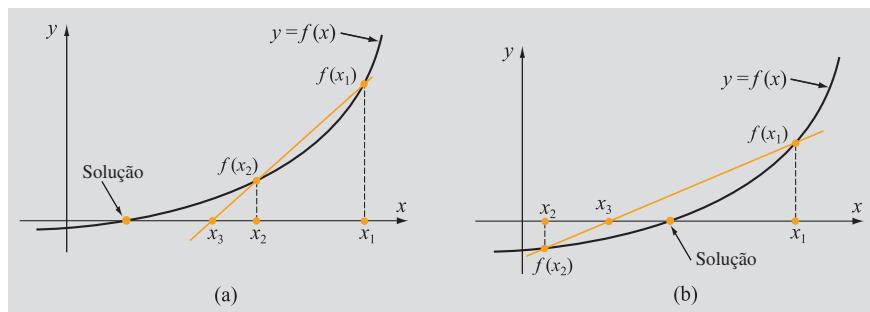
O método da secante é um esquema usado para se obter a solução numérica de uma equação na forma  $f(x) = 0$ . O método usa dois pontos na vizinhança da solução para determinar a nova solução estimada (Fig. 3-14). Os dois pontos (marcados como  $x_1$  e  $x_2$  na figura) são usados para definir uma linha reta (reta secante), e o ponto onde essa reta intercepta o eixo  $x$  (marcado como  $x_3$  na figura) é a nova solução estimada. Conforme ilustrado, ambos os pontos podem estar de um lado da solução (Fig. 3-14a), ou a solução pode estar entre os dois pontos (Fig. 3-14b). A inclinação da reta secante é dada por:

$$\frac{f(x_1) - f(x_2)}{x_1 - x_2} = \frac{f(x_2) - 0}{x_2 - x_3} \quad (3.24)$$

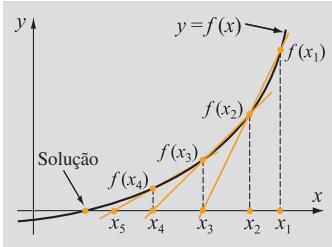
que pode ser resolvida para  $x_3$ :

$$x_3 = x_2 - \frac{f(x_2)(x_1 - x_2)}{f(x_1) - f(x_2)} \quad (3.25)$$

Assim que o ponto  $x_3$  é determinado, ele é usado juntamente com o ponto  $x_2$  para calcular a próxima estimativa da solução,  $x_4$ . A Eq. (3.25) pode ser generalizada para gerar uma fórmula iterativa na qual a nova estimativa da solução  $x_{i+1}$  é determinada a partir das duas soluções anteriores,  $x_i$  e  $x_{i-1}$ .



**Figura 3-14** Método da secante.



**Figura 3-15** Método da secente.

A Fig. 3-15 ilustra o processo iterativo com o método da secante.

### Relação com o método de Newton

A análise do método da secante mostra que, quando os dois pontos que definem a reta secante são próximos entre si, esse método é na realidade uma forma aproximada do método de Newton. Isso pode ser visto rescrevendo a Eq. (3.26) na forma:

$$x_{i+1} = x_i - \frac{f(x_i)}{\frac{f(x_{i-1}) - f(x_i)}{(x_{i-1} - x_i)}} \quad (3.27)$$

Essa equação é quase idêntica à Eq. (3.14) do método de Newton. Na Eq. (3.27), o denominador do segundo termo no lado direito é uma aproximação do valor da derivada de  $f(x)$  em  $x_i$ . Na Eq. (3.14), o denominador é realmente a derivada  $f'(x_i)$ . No método da secante (diferentemente do método de Newton), não é necessário conhecer a forma analítica de  $f'(x)$ .

A programação do método da secante é muito similar àquela do método de Newton. A Fig. 3-16 lista uma função (chamada de SecanteRaiz) que pode ser usada na determinação da raiz de  $f(x) = 0$  no MATLAB. O programa consiste em um laço de repetição no qual a solução seguinte  $X_i$  é calculada a partir das duas soluções anteriores,  $X_b$  e  $X_a$ , usando a Eq. (3.26). A repetição é interrompida se o erro for pequeno o suficiente de acordo com a Eq. (3.18). A função  $f(x)$  (que é usada na Eq. (3.26)) deve ser fornecida em separado como uma função definida pelo usuário. Seu nome é digitado no argumento de SecanteRaiz como uma string.

Como um exemplo, a equação dos Exemplos 3-1 e 3-2 é resolvida com a função SecanteRaiz. Foram escolhidos como pontos de partida  $a = 2$  e  $b = 3$ .

```
>> format long
>> SecanteRaiz('FunExemplo2',2,3,0.0001,10)
ans =
2.43046572658875
```

A função SecanteRaiz também é usada na solução do Exemplo 3-4.

## 3.7 MÉTODO DA ITERAÇÃO DE PONTO FIXO

O método da iteração de ponto fixo é um método usado para resolver uma equação na forma  $f(x) = 0$ . O método é implementado rescrevendo a equação como:

$$x = g(x) \quad (3.28)$$

**Programa 3-3 Função definida pelo usuário. Método da secante.**

```

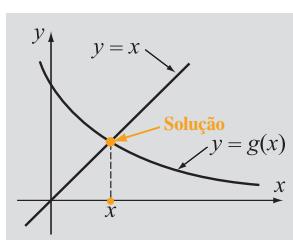
function Xs = SecanteRaiz(Fun,Xa,Xb,Err,imax)
% SecanteRaiz determina a raiz de Fun = 0 usando o método da Secante.
% Variáveis de entrada:
% Fun Nome (string) da função que calcula Fun para um dado x.
% a, b Dois pontos na vizinhança da raiz (ambos de um mesmo lado da raiz
% ou com a raiz entre si).
% Err Erro máximo.
% imax Número máximo de iterações
% Variável de saída:
% Xs Solução

for i = 1:imax
    FunXb = feval(Fun,Xb);
    Xi = Xb - FunXb*(Xa - Xb)/(feval(Fun,Xa) - FunXb); Eq. (3.26).
    if abs((Xi - Xb)/Xb) < Err Eq. (3.18).
        Xs = Xi;
        break
    end
    Xa = Xb;
    Xb = Xi;
end
if i == imax
    sprintf('A solução não foi obtida em %i iterações.\n',imax)
    Xs = ('Sem resposta');
end

```

**Figura 3-16** Arquivo de função no MATLAB para resolver equações usando o método da secante.

Obviamente, quando  $x$  é a solução de  $f(x) = 0$ , o lado esquerdo e o lado direito da Eq. (3.28) são iguais. Isso é ilustrado graficamente com o traçado de  $y = x$  e  $y = g(x)$ , como mostra a Fig. 3-17. O ponto de interseção entre os dois gráficos, chamado de **ponto fixo**, é a solução. O valor numérico da solução é determinado por meio de um processo iterativo. Ele começa com a escolha de um valor de  $x$  próximo ao ponto fixo. Esse valor é a primeira tentativa da solução e é substituído em  $g(x)$ . O valor obtido em  $g(x)$  é a nova (segunda) estimativa da solução. Esse segundo valor é então substituído novamente em  $g(x)$ , o que leva à terceira solução estimada. Com isso, a fórmula iterativa é dada por:

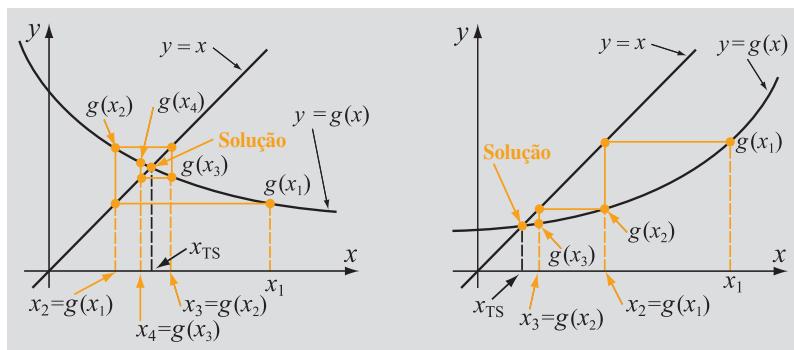


**Figura 3-17** Método da iteração de ponto fixo.

$$x_{i+1} = g(x_i) \quad (3.29)$$

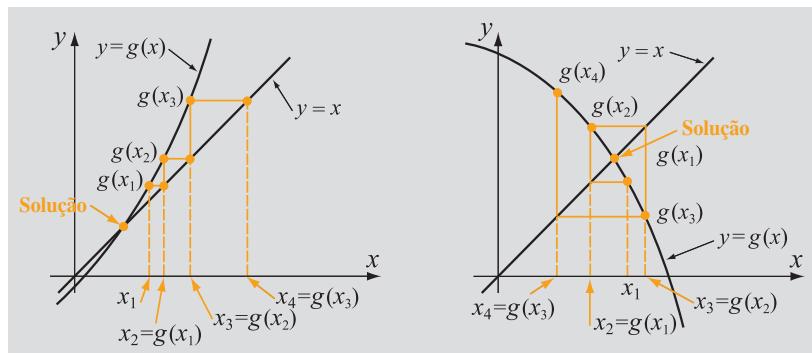
A função  $g(x)$  é chamada de **função de iteração**.

- Quando o método funciona, os valores de  $x$  obtidos são iterações sucessivas que convergem progressivamente em direção à solução. Dois casos como esse são ilustrados graficamente na Fig. 3-18. O processo de solução começa com a escolha do ponto  $x_1$  no eixo  $x$  e o traçado de uma reta vertical que cruze a curva  $y = g(x)$  no ponto  $g(x_1)$ . Como  $x_2 = g(x_1)$ , uma reta horizontal é traçada a partir do ponto  $(x_1, g(x_1))$  em direção à reta  $y = x$ . O ponto de interseção fornece a localização de  $x_2$ . A partir de  $x_2$ , uma reta vertical é traçada em direção à curva  $y = g(x)$ . O ponto de interseção é agora  $(x_2, g(x_2))$ , e  $g(x_2)$  é também o valor de  $x_3$ . A partir do ponto  $(x_2, g(x_2))$ , uma reta horizontal é novamente traçada em direção a  $y = x$ , e o ponto de interseção fornece a localização de  $x_3$ . À medida que o processo continua, os pontos de interseção convergem em direção ao ponto fixo, ou seja, à solução exata  $x_{TS}$ .



**Figura 3-18** Convergência do método da iteração de ponto fixo.

- É possível, no entanto, que as iterações não converjam para o ponto fixo, mas, ao contrário, divirjam conforme ilustrado na Fig. 3-19. Essa figura mostra que, mesmo que o ponto de partida esteja próximo da solução, os pontos subsequentes podem se afastar da solução.



**Figura 3-19** Divergência do método da iteração de ponto fixo.

- Às vezes, a forma  $f(x) = 0$  não se presta à dedução de uma fórmula iterativa  $x = g(x)$ . Em tais casos, sempre é possível somar e subtrair  $x$  de  $f(x)$  para obter  $x + f(x) - x = 0$ . A última equação pode ser rescrita na forma apropriada para a utilização do método da iteração de ponto fixo:

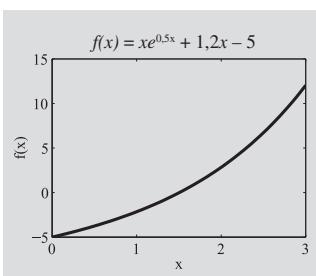
$$x = x + f(x) = g(x)$$

### Escolhendo uma função de iteração $g(x)$ apropriada

Para uma dada equação  $f(x) = 0$ , a função de iteração não é única, já que essa equação pode ser modificada de formas diferentes para assumir a forma  $x = g(x)$ . Isso significa que várias funções de iteração  $g(x)$  podem ser escritas para a mesma equação. A função  $g(x)$  a ser usada no processo iterativo da Eq. (3.29) é aquela na qual as iterações convergem em direção à solução. Pode haver mais de uma forma possível, ou mesmo nenhuma forma possível. Neste último caso, o método da iteração de ponto fixo não pode ser usado para resolver a equação. Em casos onde há múltiplas soluções, uma dada função de iteração pode levar a uma raiz, enquanto uma diferente função pode levar a outras raízes. Na realidade, é possível determinar antecipadamente se as iterações convergem ou divergem para uma função  $g(x)$  específica.

O método da iteração de ponto fixo converge se, na vizinhança do ponto fixo, a derivada de  $g(x)$  possuir um valor absoluto menor que 1 (condição também conhecida como Lipschitz contínua):

$$|g'(x)| < 1 \quad (3.30)$$



**Figura 3-20** Gráfico de  $xe^{x/2} + 1,2x - 5$ .

Como um exemplo, considere a equação:

$$xe^{0.5x} + 1,2x - 5 = 0 \quad (3.31)$$

Um gráfico da função  $f(x) = xe^{0.5x} + 1,2x - 5 = 0$  (ver Fig. 3-20) mostra que a equação tem uma solução entre  $x = 1$  e  $x = 2$ .

A Eq. (3.31) pode ser rescrita na forma  $x = g(x)$  de diferentes maneiras. Três possibilidades são discutidas a seguir:

**Caso a:**  $x = \frac{5 - xe^{0.5x}}{1,2}$  (3.32)

Neste caso,  $g(x) = \frac{5 - xe^{0.5x}}{1,2}$  e  $g'(x) = -(e^{0.5x} + 0,5xe^{0.5x})/1,2$ .

Os valores de  $g'(x)$  nos pontos  $x = 1$  e  $x = 2$ , que estão na vizinhança da solução, são:

$$g'(1) = -(e^{0.5 \cdot 1} + 0,5 \cdot 1e^{0.5 \cdot 1})/1,2 = -2,0609$$

$$g'(2) = -(e^{0.5 \cdot 2} + 0,5 \cdot 2e^{0.5 \cdot 2})/1,2 = -4,5305$$

**Caso b:**  $x = \frac{5}{e^{0.5x} + 1,2}$  (3.33)

Neste caso,  $g(x) = \frac{5}{e^{0.5x} + 1,2}$  e  $g'(x) = \frac{-5e^{0.5x}}{2(e^{0.5x} + 1,2)^2}$ .

Os valores de  $g'(x)$  nos pontos  $x = 1$  e  $x = 2$ , que estão na vizinhança da solução, são:

$$g'(1) = \frac{-5e^{0.5 \cdot 1}}{2(e^{0.5 \cdot 1} + 1.2)^2} = -0,5079$$

$$g'(2) = \frac{-5e^{0.5 \cdot 2}}{2(e^{0.5 \cdot 2} + 1.2)^2} = -0,4426$$

**Caso c:**

$$x = \frac{5 - 1,2x}{e^{0.5x}} \quad (3.34)$$

$$\text{Neste caso, } g(x) = \frac{5 - 1,2x}{e^{0.5x}} \quad \text{e} \quad g'(x) = \frac{-3,7 + 0,6x}{e^{0.5x}}.$$

Os valores de  $g'(x)$  nos pontos  $x = 1$  e  $x = 2$ , que estão na vizinhança da solução, são:

$$g'(1) = \frac{-3,7 + 0,6 \cdot 1}{e^{0.5 \cdot 1}} = -1,8802$$

$$g'(2) = \frac{-3,7 + 0,6 \cdot 2}{e^{0.5 \cdot 2}} = -0,9197$$

Esses resultados mostram que a função de iteração  $g(x)$  do Caso b é aquela que deve ser usada, pois, neste caso,  $|g'(1)| < 1$  e  $|g'(2)| < 1$ . A substituição de  $g(x)$  do Caso b na Eq. (3.29) resulta em:

$$x_{i+1} = \frac{5}{e^{0.5x_i} + 1,2} \quad (3.35)$$

Começando com  $x_1 = 1$ , as primeiras iterações são:

$$x_2 = \frac{5}{e^{0.5 \cdot 1} + 1,2} = 1,7552 \quad x_3 = \frac{5}{e^{0.5 \cdot 1,7552} + 1,2} = 1,3869$$

$$x_4 = \frac{5}{e^{0.5 \cdot 1,3869} + 1,2} = 1,5622 \quad x_5 = \frac{5}{e^{0.5 \cdot 1,5622} + 1,2} = 1,4776$$

$$x_6 = \frac{5}{e^{0.5 \cdot 1,4776} + 1,2} = 1,5182 \quad x_7 = \frac{5}{e^{0.5 \cdot 1,5182} + 1,2} = 1,4986$$

Conforme esperado, os valores calculados nas iterações convergem em direção à solução exata, que é  $x = 1,5050$ .

Do contrário, se a função  $g(x)$  do Caso a for usada, as primeiras iterações são:

$$x_2 = \frac{5 - 1e^{0.5 \cdot 1}}{1,2} = 2,7927 \quad x_3 = \frac{5 - 2,7927e^{0.5 \cdot 2,7927}}{1,2} = -5,2364$$

$$x_4 = \frac{5 - (-5,2364)e^{0.5 \cdot (-5,2364)}}{1,2} = 4,4849$$

$$x_5 = \frac{5 - 4,4849e^{0.5 \cdot 4,4849}}{1,2} = -31,0262$$

Nesse caso, as iterações fornecem valores que divergem da solução.

### Quando as iterações devem ser interrompidas?

O erro real (a diferença entre a solução exata e a solução estimada) não pode ser calculado, já que a solução exata é geralmente desconhecida. Da mesma forma

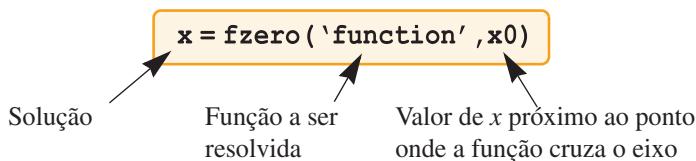
que no método de Newton, as iterações podem ser interrompidas quando o erro relativo ou a tolerância em  $f(x)$  for menor que algum valor predeterminado (Eqs. (3.18) ou (3.19)).

### 3.8 USO DE FUNÇÕES RESIDENTES DO MATLAB PARA RESOLVER EQUAÇÕES NÃO-LINEARES

O MATLAB possui duas funções que podem ser usadas para resolver equações com uma variável. O comando `fzero` pode ser usado na obtenção da raiz de qualquer equação, e o comando `roots` pode ser usado na obtenção das raízes de um polinômio.

#### 3.8.1 O comando `fzero`

O comando `fzero` pode ser usado para resolver equações (na forma  $f(x) = 0$ ) com apenas uma variável. O usuário precisa saber aproximadamente onde está a solução, ou, se houver múltiplas soluções, qual delas é a desejada. A forma do comando é:



- $x$  é a solução, que é um escalar.
- '`function`' é a função cuja raiz é desejada. Ela pode ser fornecida de três diferentes maneiras:
  1. A maneira mais simples é fornecer a expressão matemática no formato *string*.
  2. A expressão matemática também pode ser fornecida como uma função definida pelo usuário, e o nome dessa função deve ser digitado no formato *string*.
  3. A expressão matemática pode ser definida como uma função *inline*, e seu nome é então digitado no formato *string*.
- A função deve ser escrita na forma padrão. Por exemplo, se a função a ser resolvida for  $xe^{-x} = 0,2$ , ela deve ser escrita como  $f(x) = xe^{-x} - 0,2 = 0$ . Se essa função for usada como argumento de entrada da função `fzero`, ela deve ser digitada como: '`x*exp (-x) -0.2`'.
- Quando uma função é fornecida como um *string*, ela não pode incluir variáveis predefinidas. Por exemplo, se a função a ser resolvida for  $f(x) = xe^{-x} - 0,2$ , não é possível predefinir `b=0.2` no MATLAB e entrar com '`x*exp (-x) -b`'.
- $x0$  pode ser um escalar ou um vetor com dois elementos. Se ele for fornecido como um escalar, ele deve ser um valor de  $x$  próximo ao ponto onde a função cruza o eixo  $x$ . Se  $x0$  for fornecido como vetor, os dois elementos devem ser

pontos nos lados opostos da solução de forma tal que  $f(x_0(1))$  tenha um sinal diferente de  $f(x_0(2))$ . Quando uma função tem mais de uma solução, cada solução pode ser determinada separadamente com o uso da função `fzero` e o emprego de valores de  $x_0$  que sejam próximos a cada uma das soluções.

O uso do comando `fzero` é ilustrado a seguir na solução da equação dos Exemplos 3-1 e 3-2:

```
>> format long
>> Sol=fzero('8 - 4.5*(x - sin(x))',2)
Sol =
2.43046574172363
```

Resolve a equação do Exemplo 3-1.

### 3.8.2 O comando `roots`

O comando `roots` pode ser usado na obtenção das raízes de um polinômio. A forma do comando é:

**`r = roots(p)`**

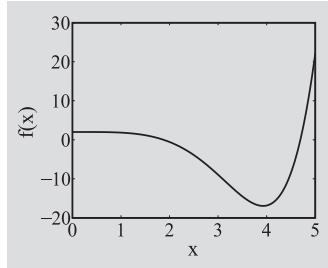
$r$  é um vetor coluna contendo as raízes do polinômio       $p$  é um vetor linha com os coeficientes do polinômio

## 3.9 EQUAÇÕES COM MÚLTIPHAS SOLUÇÕES

Muitas equações não-lineares na forma  $f(x) = 0$  têm múltiplas soluções ou raízes.

Por exemplo, considere a equação a seguir:

$$f(x) = \cos(x)\cosh(x) + 1 \quad (3.36)$$



**Figura 3-21** Gráfico da Eq. (3.36).

Um gráfico dessa função gerado no MATLAB é mostrado na Fig. 3-21, no intervalo  $[0, 5]$ . Como se vê, a função apresenta cruzamentos por zero entre  $x = 1$  e  $x = 2$ , e entre  $x = 4$  e  $x = 5$ . A existência de múltiplas raízes é típica de equações não-lineares. Uma estratégia geral que pode ser empregada na obtenção das raízes no intervalo  $[0, 5]$  é a seguinte:

- Determine a localização aproximada das raízes definindo pequenos intervalos dentro dos quais as raízes estão localizadas. Isso pode ser feito com o traçado da função (conforme mostrado na Fig. 3-21) ou com a avaliação da função ao longo de um conjunto de pontos sucessivos procurando mudanças de sinal.
- Aplique qualquer um dos métodos descritos nas Seções 3.3 a 3.7 em um subintervalo restrito. Por exemplo, a primeira raiz, contida no intervalo  $[1, 2]$ , pode ser obtida com o emprego do método da bisseção ou de um método similar com  $a = 1$  e  $b = 2$ . Alternativamente, um valor inicial ou uma tentativa

inicial pode ser usado(a) no método de Newton ou no método da iteração de ponto fixo para se determinar a raiz. A função `fzero` do MATLAB também pode ser utilizada. O processo pode então ser repetido ao longo do intervalo [4, 5] seguinte para se encontrar a próxima raiz.

O exemplo a seguir apresenta a solução da função na Eq. (3.36) em uma situação prática.

### Exemplo 3-4: Solução de uma equação com raízes múltiplas

As freqüências naturais ( $\omega_n$ ) de vibração de uma viga em balanço são determinadas a partir das raízes da equação:

$$f(k_n L) = \cos(k_n L) \cosh(k_n L) + 1 = 0 \quad (3.37)$$

onde  $L$  é o comprimento da viga e a freqüência  $\omega_n$  é dada por:

$$\omega_n = (k_n L)^2 \sqrt{\frac{EI}{\rho A L^4}}$$

onde  $E$  é o módulo elástico,  $I$  é o momento de inércia,  $A$  é a área da seção reta e  $\rho$  é a densidade por unidade de comprimento.

- (a) Determine o valor da primeira raiz definindo intervalos pequenos ao longo dos quais as raízes existem e empregando o método da secante.
- (b) Escreva um programa no MATLAB que determine o valor de  $k_n L$  para as quatro primeiras raízes.

### SOLUÇÃO

A Eq. (3.37) é idêntica à Eq. (3.36), e um gráfico que mostra a localização das duas primeiras raízes é apresentado na Fig. 3-21. A localização das duas raízes seguintes é mostrada na figura à direita, na qual a função é traçada ao longo do intervalo [7; 11,2]. Essa figura mostra que a terceira raiz está localizada em torno de 8 e que a quarta raiz está próxima de 11.

- (a) O valor da primeira raiz é determinado com o uso da função `SecanteRaiz`, listada na Fig. 3-16. Primeiramente, no entanto, é necessário escrever uma função no MATLAB que represente a função  $f(k_n L)$  da Eq. (3.37) (o nome dessa função é `FunExemplo3`):

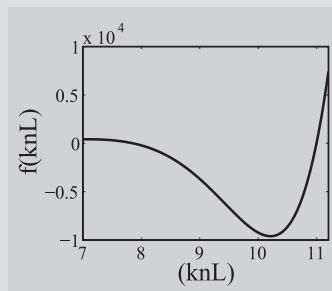
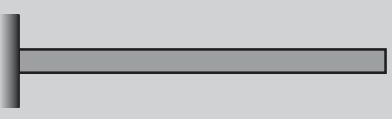
```
function y = FunExemplo3(x)
y = cos(x)*cosh(x) + 1;
```

A primeira raiz está entre 1 e 2. Para determinar numericamente esta solução, a função `SecanteRaiz` (listada na Fig. 3-16) é usada com  $a = 1$ ,  $b = 2$ ,  $\text{Err} = 0,0001$  e  $\text{imax} = 10$ :

```
>> FirstSolution = SecanteRaiz('FunExample3',1,2,0.0001,10)
```

FirstSolution =

1.87510406460241



- (b) A seguir, escreve-se um programa no MATLAB que encontra automaticamente as quatro raízes. O programa avalia a função ao longo de um conjunto de intervalos sucessivos, procurando

**Usar `format long` no MATLAB**

mudanças de sinal. Ele começa em  $k_n L = 0$  e usa um incremento de 0,2 até que o valor de  $k_n L$  seja igual a 11,2. Se uma mudança de sinal é detectada, determina-se a raiz no interior do intervalo usando a função `fzero` do MATLAB.

```
clear all
F = inline('cos(x)*cosh(x)+1');
Inc = 0.2;

i = 1;
KnLa = 0;
KnLb = KnLa + Inc;
while KnLb <= 11.2
    if F(KnLa)*F(KnLb) < 0
        Roots(i) = fzero(F,[KnLa,KnLb]);
        i = i + 1;
    end
    KnLa = KnLb;
    KnLb = KnLa + Inc;
end
Roots
```

Define o ponto à esquerda do primeiro incremento.

Define o ponto à direita do primeiro incremento.

Procura uma mudança de sinal no valor da função.

Determina a raiz dentro do intervalo se uma mudança de sinal for identificada.

Define o ponto à esquerda do próximo incremento.

Define o ponto à direita do próximo incremento.

Quando o programa é executado, a janela de comandos do MATLAB exibe:

```
Roots =
1.87510406871196 4.69409113297418 7.85475743823761 10.99554073487547
```

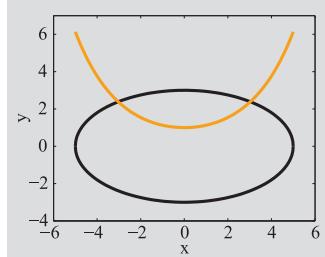
Esses são os valores das quatro primeiras raízes.

### 3.10 SISTEMAS DE EQUAÇÕES NÃO-LINEARES

Um sistema de equações não-lineares consiste em duas, ou mais, equações não-lineares que devem ser solucionadas simultaneamente. Por exemplo, a Fig. 3-22 mostra uma curva catenária (cabo pendurado) dada pela equação  $y = \frac{1}{2}(e^{x/2} + e^{(-x)/2})$  e uma elipse especificada pela equação  $\frac{x^2}{5^2} + \frac{y^2}{3^2} = 1$ . O ponto de interseção entre as duas curvas é dado pela solução do seguinte sistema de equações não-lineares:

$$f_1(x, y) = y - \frac{1}{2}(e^{x/2} + e^{(-x)/2}) = 0 \quad (3.38)$$

$$f_2(x, y) = 9x^2 + 25y^2 - 225 = 0 \quad (3.39)$$



**Figura 3-22** Gráfico das Eqs. (3.38) e (3.39).

A análise de muitos problemas na ciência e na engenharia requer a solução de sistemas de equações não-lineares. Além disso, conforme mostrado no Capítulo 9, um dos mais populares métodos numéricos usados na solução de equações

diferenciais ordinárias não-lineares (o método das diferenças finitas) requer a solução de um sistema de equações algébricas não-lineares.

Nesta seção, são apresentados dois métodos para a solução de equações não-lineares. O método de Newton (também chamado de método de Newton-Raphson), adequado para a solução de sistemas pequenos, é descrito na Seção 3.10.1. O método da iteração de ponto fixo, que também pode ser usado para solucionar grandes sistemas, é discutido na Seção 3.10.2.

### 3.10.1 Método de Newton para a solução de sistemas de equações não-lineares

O método de Newton para a solução de sistemas de equações não-lineares é uma extensão do método usado para resolver uma única equação (Seção 3.5). O método é primeiramente deduzido em detalhe para a solução de um sistema com duas equações não-lineares. Subseqüentemente, uma formulação geral é apresentada para o caso de um sistema com  $n$  equações não-lineares.

#### Resolvendo um sistema com duas equações não-lineares

Um sistema com duas equações e duas incógnitas  $x$  e  $y$  pode ser escrito como:

$$\begin{aligned} f_1(x, y) &= 0 \\ f_2(x, y) &= 0 \end{aligned} \quad (3.40)$$

O processo de solução começa com a escolha de uma solução estimada  $x_1$  e  $y_1$ . Se  $x_2$  e  $y_2$  são a solução exata do sistema (desconhecida) e estão suficientemente próximos de  $x_1$  e  $y_1$ , então os valores de  $f_1$  e  $f_2$  nos pontos  $x_2$  e  $y_2$  podem ser expressos usando a expansão em série de Taylor das funções  $f_1(x, y)$  e  $f_2(x, y)$  em torno de  $(x_1, y_1)$  (ver Seção 2.7.2):

$$f_1(x_2, y_2) = f_1(x_1, y_1) + (x_2 - x_1) \frac{\partial f_1}{\partial x} \Big|_{x_1, y_1} + (y_2 - y_1) \frac{\partial f_1}{\partial y} \Big|_{x_1, y_1} + \dots \quad (3.41)$$

$$f_2(x_2, y_2) = f_2(x_1, y_1) + (x_2 - x_1) \frac{\partial f_2}{\partial x} \Big|_{x_1, y_1} + (y_2 - y_1) \frac{\partial f_2}{\partial y} \Big|_{x_1, y_1} + \dots \quad (3.42)$$

Como  $x_2$  e  $y_2$  estão próximos de  $x_1$  e  $y_1$ , valores aproximados para  $f_1(x_2, y_2)$  e  $f_2(x_2, y_2)$  podem ser calculados sem que sejam considerados termos de ordem mais elevada. Além disso, como  $f_1(x_2, y_2) = 0$  e  $f_2(x_2, y_2) = 0$ , as Eqs. (3.41) e (3.42) podem ser rescritas como:

$$\frac{\partial f_1}{\partial x} \Big|_{x_1, y_1} \Delta x + \frac{\partial f_1}{\partial y} \Big|_{x_1, y_1} \Delta y = -f_1(x_1, y_1) \quad (3.43)$$

$$\frac{\partial f_2}{\partial x} \Big|_{x_1, y_1} \Delta x + \frac{\partial f_2}{\partial y} \Big|_{x_1, y_1} \Delta y = -f_2(x_1, y_1) \quad (3.44)$$

onde  $\Delta x = x_2 - x_1$  e  $\Delta y = y_2 - y_1$ . Como todos os termos nas Eqs. (3.43) e (3.44) são conhecidos, exceto as incógnitas  $\Delta x$  e  $\Delta y$ , essas equações formam um sistema de

duas equações não-lineares. O sistema pode ser solucionado com o emprego da regra de Cramer (ver Seção 2.4.6):

$$\Delta x = \frac{-f_1(x_1, y_1) \frac{\partial f_2}{\partial y} \Big|_{x_1, y_1} + f_2(x_1, y_1) \frac{\partial f_1}{\partial y} \Big|_{x_1, y_1}}{J(f_1(x_1, y_1), f_2(x_1, y_1))} \quad (3.45)$$

$$\Delta y = \frac{-f_2(x_1, y_1) \frac{\partial f_1}{\partial x} \Big|_{x_1, y_1} + f_1(x_1, y_1) \frac{\partial f_2}{\partial x} \Big|_{x_1, y_1}}{J(f_1(x_1, y_1), f_2(x_1, y_1))} \quad (3.46)$$

onde

$$J(f_1, f_2) = \det \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} \quad (3.47)$$

é o Jacobiano (ver Seção 2.6.3). Uma vez obtidos os valores de  $\Delta x$  e  $\Delta y$ , os valores de  $x_2$  e  $y_2$  são calculados com:

$$\begin{aligned} x_2 &= x_1 + \Delta x \\ y_2 &= y_1 + \Delta y \end{aligned} \quad (3.48)$$

Obviamente, os valores de  $x_2$  e  $y_2$  obtidos não são a solução real, pois foram desconsiderados termos de ordem mais elevada nas Eqs. (3.41) e (3.42). Entretanto, espera-se que esses valores estejam próximos da solução exata  $x_1$  e  $y_1$ .

O processo de solução continua com o uso de  $x_2$  e  $y_2$  como a nova estimativa da solução e o emprego das Eqs. (3.43) e (3.44) para determinar novos valores de  $\Delta x$  e  $\Delta y$  que levem a  $x_3$  e  $y_3$ . As iterações continuam até que duas respostas sucessivas difiram de uma quantidade menor que um valor desejado.

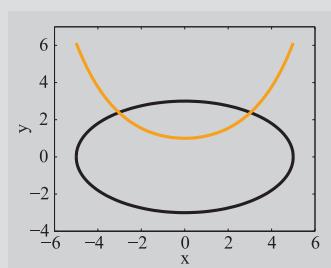
### Exemplo 3-5: Solução de um sistema de equações não-lineares usando o método de Newton

As equações da curva catenária e da elipse, que são mostradas na figura, são dadas por:

$$f_1(x, y) = y - \frac{1}{2}(e^{x/2} + e^{(-x)/2}) = 0 \quad (3.49)$$

$$f_2(x, y) = 9x^2 + 25y^2 - 225 = 0 \quad (3.50)$$

Use o método de Newton para determinar o ponto de interseção entre as curvas que reside no primeiro quadrante do sistema de coordenadas.



### SOLUÇÃO

As Eqs. (3.49) e (3.50) são um sistema de duas equações não-lineares. Os pontos de interseção são dados pela solução do sistema. A solução pelo método de Newton é obtida empregando-se as Eqs. (3.43) e (3.44). No presente problema, as derivadas parciais das equações são dadas por:

$$\frac{\partial f_1}{\partial x} = -\frac{1}{4}(e^{x/2} - e^{(-x)/2}) \quad \text{e} \quad \frac{\partial f_1}{\partial y} = 1 \quad (3.51)$$

$$\frac{\partial f_2}{\partial x} = 18x \quad \text{e} \quad \frac{\partial f_2}{\partial y} = 50y \quad (3.52)$$

O Jacobiano é dado por:

$$J(f_1, f_2) = \det \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix} = \det \begin{bmatrix} -\frac{1}{4}(e^{x/2} - e^{(-x)/2}) & 1 \\ 18x & 50y \end{bmatrix} = -\frac{1}{4}(e^{x/2} - e^{(-x)/2})50y - 18x \quad (3.53)$$

A substituição das Eqs. (3.51)–(3.53) nas Eqs. (3.45) e (3.46) fornece a solução para  $\Delta x$  e  $\Delta y$ .

O problema é solucionado no programa de MATLAB listado abaixo. A ordem das operações no programa é:

- A solução começa com a tentativa inicial,  $x_i = 2,5$ ,  $y_i = 2,0$ .
- As iterações têm início.  $\Delta x$  e  $\Delta y$  são determinados com a substituição de  $x_i$  e  $y_i$  nas Eqs. (3.45) e (3.46).
- $x_{i+1} = x_i + \Delta x$ , e  $y_{i+1} = y_i + \Delta y$  são determinados.
- Se o erro relativo estimado (Eq. (3.9)) para ambas as variáveis for menor que 0,001, as iterações param. Caso contrário, os valores de  $x_{i+1}$  e  $y_{i+1}$  são atribuídos a  $x_i$  e  $y_i$ , respectivamente, e a próxima iteração tem início.

O programa também mostra a solução e o erro em cada iteração.

```
% Solução do Exemplo 5 do Capítulo 3
F1 = inline('y - 0.5*(exp(x/2) + exp(-x/2))');
F2 = inline('9*x^2 + 25*y^2 - 225');
F1x = inline('-(exp(x/2) + exp(-x/2))/4');
F2x = inline('18*x');
F2y = inline('50*y');
Jacob = inline('-(exp(x/2) + exp(-x/2))/4*50*y - 18*x');
xi = 2.5; yi = 2; Err = 0.001;
for i = 1:5
    Jac = Jacob(xi,yi);
    Delx = (-F1(xi,yi)*F2y(yi) + F2(xi,yi))/Jac;
    Dely = (-F2(xi,yi)*F1x(xi) + F1(xi,yi)*F2x(xi))/Jac;
    xip1 = xi + Delx;
    yip1 = yi + Dely;
    Errx = abs((xip1 - xi)/xi);
    Erry = abs((yip1 - yi)/yi);
    fprintf('i = %2.0f x = %-7.4f y = %-7.4f Erro em x = %-7.4f Erro em y = %-7.4f\n',i,xip1,yip1,Errx,Erry)
    if Errx < Err & Erry < Err
        break
    else
        xi = xip1; yi = yip1;
    end
end
```

Atribui a primeira estimativa da solução.

As iterações têm início.

Calcula  $\Delta x$  e  $\Delta y$  com as Eqs. (3.45) e (3.46).

Calcula  $x_{i+1}$  e  $y_{i+1}$ .

Se o erro não for pequeno o suficiente, atribui-se  $x_{i+1}$  a  $x_i$  e  $y_{i+1}$  a  $y_i$ .

Quando o programa é executado, aparecem as seguintes informações na janela de comando do MATLAB:

```
i = 1 x = 3.1388 y = 2.4001 Erro em x = 0.25551 Erro em y = 0.20003
i = 2 x = 3.0339 y = 2.3855 Erro em x = 0.03340 Erro em y = 0.00607
i = 3 x = 3.0312 y = 2.3859 Erro em x = 0.00091 Erro em y = 0.00016
```

Esses resultados mostram que os valores convergem rapidamente para a solução.

A aplicação do método de Newton é ilustrada no Exemplo 3-5, onde se determina o ponto de interseção entre a curva catenária e a elipse da Fig. 3-22.

### Resolvendo um sistema com $n$ equações não-lineares

O método de Newton pode ser facilmente generalizado para o caso de um sistema com  $n$  equações não-lineares. Com  $n$  incógnitas,  $x_1, x_2, \dots, x_n$ , um sistema de  $n$  equações não-lineares simultâneas tem a forma:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\dots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \tag{3.54}$$

Os valores das funções na próxima aproximação da solução,  $x_{1,i+1}, x_{2,i+1}, \dots, x_{n,i+1}$ , são então obtidos usando-se a expansão em série de Taylor em torno do valor atual da solução aproximada,  $x_{1,i}, x_{2,i}, \dots, x_{n,i}$ . Se o mesmo procedimento que leva às Eqs. (3.43) e (3.44) for seguido, obtém-se como resultado o seguinte sistema com  $n$  equações lineares para as incógnitas  $\Delta x_1, \Delta x_2, \dots, \Delta x_n$ :

$$\left[ \begin{array}{cccc} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{array} \right] \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \dots \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} -f_1 \\ -f_2 \\ \dots \\ -f_n \end{bmatrix} \tag{3.55}$$

(O determinante da matriz contendo as derivadas parciais das funções no lado esquerdo da equação é chamado de Jacobiano, ver Seção 2.6.3). Assim que o sistema da Eq. (3.55) é resolvido, a nova solução aproximada é obtida a partir de:

$$\begin{aligned} x_{1,i+1} &= x_{1,i} + \Delta x_1 \\ x_{2,i+1} &= x_{2,i} + \Delta x_2 \\ &\dots \\ x_{n,i+1} &= x_{n,i} + \Delta x_n \end{aligned} \tag{3.56}$$

Da mesma forma que no método de Newton para uma única equação não-linear, a convergência não é garantida. O procedimento iterativo de Newton converge

durante a solução de um sistema de equações não-lineares tipicamente se as três condições a seguir forem satisfeitas:

- (i) As funções  $f_1, f_2, \dots, f_n$  e suas derivadas devem ser contínuas e limitadas na vizinhança da solução (raiz).
- (ii) O Jacobiano deve ser diferente de zero, isto é,  $J(f_1, f_2, \dots, f_n) \neq 0$ , na vizinhança da solução.
- (iii) A estimativa inicial (tentativa) da solução deve estar suficientemente próxima da solução exata.

O método de Newton para resolver um sistema com  $n$  equações não-lineares é resumido no algoritmo a seguir.

### Algoritmo para o método de Newton usado na solução de sistemas com $n$ equações não-lineares

Dado um sistema com  $n$  equações não-lineares,

1. Estime (chute) uma solução inicial,  $x_{1,i}, x_{2,i}, \dots, x_{n,i}$ .
2. Calcule o Jacobiano e o valor das funções no lado direito da Eq. (3.55).
3. Resolva a Eq. (3.55) para  $\Delta x_1, \Delta x_2, \dots, \Delta x_n$ .
4. Calcule a nova solução estimada,  $x_{1,i+1}, x_{2,i+1}, \dots, x_{n,i+1}$ , usando a Eq. (3.56).
5. Calcule o erro. Se a nova solução não for suficientemente precisa, atribua os valores de  $x_{1,i+1}, x_{2,i+1}, \dots, x_{n,i+1}$  a  $x_{1,i}, x_{2,i}, \dots, x_{n,i}$  e inicie uma iteração começando do passo 2.

### Comentários adicionais a respeito do método de Newton usado na solução de sistemas com $n$ equações não-lineares

- O método, quando bem-sucedido, converge rapidamente. A não convergência ocorre porque a tentativa inicial não está suficientemente próxima da solução.
- As derivadas parciais (os elementos da matriz Jacobiana) devem ser determinadas. Isso pode ser feito analiticamente ou numericamente (a diferenciação numérica é tratada no Capítulo 6). Entretanto, para um grande sistema de equações, a determinação do Jacobiano pode ser difícil.
- Quando o sistema de equações consiste em mais de três equações, a solução da Eq. (3.55) deve ser feita numericamente. Métodos para resolver sistemas de equações lineares são descritos no Capítulo 4.

### 3.10.2 Método da iteração de ponto fixo para a solução de sistemas de equações não-lineares

O método da iteração de ponto fixo discutido na Seção 3.7 para solucionar uma única equação não-linear pode ser estendido ao caso de um sistema de equações não-lineares. Um sistema com  $n$  equações não-lineares e incógnitas  $x_1, x_2, \dots, x_n$ , tem a forma:

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\dots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \tag{3.57}$$

O sistema pode ser rescrito na forma:

$$\begin{aligned} x_1 &= g_1(x_1, x_2, \dots, x_n) \\ x_2 &= g_2(x_1, x_2, \dots, x_n) \\ &\dots \\ x_n &= g_n(x_1, x_2, \dots, x_n) \end{aligned} \tag{3.58}$$

onde os  $g_s$  são as funções de iteração. O processo de solução tem início com a tentativa  $x_{1,i}, x_{2,i}, \dots, x_{n,i}$ , que é substituída no lado direito das Eqs. (3.58). Os valores calculados pelas Eqs. (3.58) são a nova (segunda) estimativa da solução,  $x_{1,i+1}, x_{2,i+1}, \dots, x_{n,i+1}$ . A nova estimativa é substituída novamente no lado direito das Eqs. (3.58) para determinar a nova solução, e assim por diante. Quando o método funciona, as novas estimativas da solução convergem em direção à solução exata. Neste caso, o processo continua até que se atinja a precisão desejada. Por exemplo, o erro relativo estimado é calculado para cada uma das variáveis, e as iterações são interrompidas quando o maior erro relativo for menor que um valor especificado.

A convergência do método depende da forma das funções de iteração. Para um dado problema, diferentes funções de iteração podem ser usadas, já que a escrita das Eqs. (3.57) na forma das Eqs. (3.58) não é única. Em geral, várias formas podem ser apropriadas para uma solução, ou, quando existirem várias soluções, diferentes funções de iteração podem ser necessárias para que as múltiplas soluções sejam obtidas. Quando se utiliza o método da iteração de ponto fixo, é possível tentar funções de iteração com várias formas diferentes ou, em alguns casos, determinar antecipadamente se a solução convergirá para uma escolha específica dos  $g_s$ .

O método da iteração do ponto fixo aplicado a um conjunto de equações não-lineares simultâneas converge se as seguintes condições suficientes (mas não necessárias) forem atingidas:

(i) Se  $g_1, \dots, g_n, \frac{\partial g_1}{\partial x_1}, \dots, \frac{\partial g_1}{\partial x_n}, \frac{\partial g_2}{\partial x_1}, \dots, \frac{\partial g_2}{\partial x_n}, \dots, \frac{\partial g_n}{\partial x_1}, \dots, \frac{\partial g_n}{\partial x_n}$  forem contínuas na vizinhança da solução.

$$\begin{aligned} (ii) \quad & \left| \frac{\partial g_1}{\partial x_1} \right| + \left| \frac{\partial g_1}{\partial x_2} \right| + \dots + \left| \frac{\partial g_1}{\partial x_n} \right| \leq 1 \\ & \left| \frac{\partial g_2}{\partial x_1} \right| + \left| \frac{\partial g_2}{\partial x_2} \right| + \dots + \left| \frac{\partial g_2}{\partial x_n} \right| \leq 1 \\ & \dots \\ & \left| \frac{\partial g_n}{\partial x_1} \right| + \left| \frac{\partial g_n}{\partial x_2} \right| + \dots + \left| \frac{\partial g_n}{\partial x_n} \right| \leq 1 \end{aligned}$$

(iii) Se a tentativa inicial,  $x_{1,1}, x_{2,1}, \dots, x_{n,1}$ , estiver suficientemente próxima da solução.

### 3.11 PROBLEMAS

#### Problemas de solução manual

*Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos. Não utilize funções residentes do MATLAB.*

- 3.1** A tolerância  $tol$  da solução no método da bisseção é dada por  $tol = 0,5(b_n - a_n)$ , onde  $a_n$  e  $b_n$  são os pontos finais do intervalo após a  $n$ -ésima iteração. O número  $n$  de iterações necessárias para se obter uma solução com uma tolerância menor ou igual a uma tolerância especificada pode ser determinado antes de se calcular a solução. Mostre que  $n$  é dado por:

$$n \geq \frac{\log(b-a) - \log(tol)}{\log 2}$$

onde  $a$  e  $b$  são os pontos finais do intervalo inicial e  $tol$  é uma tolerância especificada pelo usuário.

- 3.2** Determine a raiz de  $f(x) = x^2 - e^{-x}$ :

- (a) Usando o método da bisseção. Comece com  $a = 0$  e  $b = 1$  e realize as primeiras cinco iterações.
- (b) Usando o método da secante. Comece com os pontos  $x_1 = 0$  e  $x_2 = 1$  e realize as cinco primeiras iterações.
- (c) Usando o método de Newton. Comece em  $x_1 = 0$  e realize as cinco primeiras iterações.

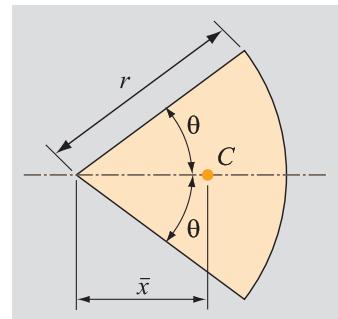
- 3.3** A localização  $\bar{x}$  do centróide de um setor circular é dada por:

$$\bar{x} = \frac{2r \sin \theta}{3\theta}$$

Determine o ângulo  $\theta$  para o qual  $\bar{x} = \frac{r}{2}$ .

Primeiramente, deduza a equação a ser resolvida e então determine a raiz usando os seguintes métodos:

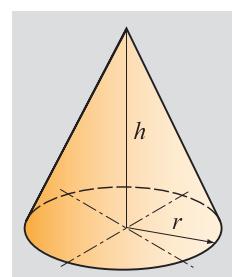
- (a) Método da bisseção. Comece com  $a = 1$  e  $b = 2$  e realize as primeiras cinco iterações.
- (b) Método da secante. Comece com os pontos  $x_1 = 1$  e  $x_2 = 2$  e realize as cinco primeiras iterações.
- (c) Método de Newton. Comece em  $x_1 = 1$  e realize as cinco primeiras iterações.



- 3.4** A área  $S$  da superfície lateral de um cone é dada por:

$$S = \pi r \sqrt{r^2 + h^2}$$

onde  $r$  é o raio da base e  $h$  é a altura. Determine o raio de um cone que tenha uma área superficial de  $1200 \text{ m}^2$  e uma altura de  $20 \text{ m}$ , calculando cinco iterações com o método da iteração de ponto fixo. Use  $r = S / (\pi \sqrt{r^2 + h^2})$  como função de iteração. Comece com  $r = 17 \text{ m}$ .



- 3.5** Determine a raiz cúbica de  $155$  obtendo a solução numérica da equação  $x^3 - 155 = 0$ . Use o método de Newton. Comece em  $x = 155$  e realize as cinco primeiras iterações.

**3.6** Determine a raiz positiva do polinômio  $x^3 + 3,8x^2 - 8,6x - 24,4$ .

- (a) Faça um gráfico do polinômio e escolha um ponto próximo à raiz como primeira estimativa da solução. Usando o método de Newton, determine a solução aproximada em cinco iterações. Comece com  $x = 2$ .
- (b) A partir do gráfico da letra (a), escolha dois pontos próximos à raiz para iniciar o processo de solução com o método da secante. Determine a solução aproximada nas primeiras cinco iterações.

**3.7** Determine todas as raízes da equação  $x^3 + 12x^2 - 100x - 6 = 0$  usando o método da bisseção (cinco iterações para cada raiz).

**3.8** Determine todas as raízes da equação  $x^3 - 2,2x^2 - 2,15x + 5,1 = 0$  usando o método da bisseção (cinco iterações para cada raiz).

**3.9** A equação  $f(x) = -x^{1/3} + 0,5x^2 - 2 = 0$  possui uma raiz entre  $x = 2$  e  $x = 3$ . Para determinar essa raiz usando o método da iteração de ponto fixo, a equação deve ser escrita na forma  $x = g(x)$ . Deduza duas formas possíveis para  $g(x)$  – uma resolvendo para  $x$  a partir do primeiro termo da equação e a outra resolvendo para  $x$  a partir do segundo termo da equação.

- (a) Determine que forma deveria ser usada de acordo com a condição dada pela Eq. (3.30).
- (b) Confirme a escolha feita na letra (a) realizando cinco iterações usando ambas as formas de  $g(x)$ .

**3.10** Determine quatro diferentes funções de iteração para resolver a equação  $x^2 - 3x + e^x - 2 = 0$  usando o método da iteração de ponto fixo.

**3.11** Determine quatro diferentes funções de iteração para resolver a equação  $x - \operatorname{tg} x = 0$  usando o método da iteração de ponto fixo.

**3.12** Resolva o seguinte sistema de equações não-lineares:

$$\begin{aligned} 4x^2 - y^3 + 28 &= 0 \\ 3x^3 + 4y^2 - 145 &= 0 \end{aligned}$$

- (a) Use o método de Newton. Comece em  $x = 1$ ,  $y = 1$  e realize as cinco primeiras iterações.
- (b) Use o método da iteração de ponto fixo. Use as funções de iteração  $y = (4x^2 + 28)^{1/3}$  e  $x = \left[ \frac{(145 - 4y^2)}{3} \right]^{1/3}$ . Comece em  $x = 1$ ,  $y = 1$  e realize as cinco primeiras iterações.

### Problemas de programação no MATLAB

Resolva os seguintes problemas no MATLAB. Não use funções residentes para resolver as equações não-lineares.

**3.13** No programa do Exemplo 3-1, as iterações são executadas em um laço de repetição *for-end*. Nesse laço de repetição, a função *inline F* é usada

duas vezes (uma vez no comando `FxNS=F(xNS)` e outra no comando `if F(a) * FxNS < 0`). Rescreva o programa de tal forma que a função *inline* apareça no laço de repetição apenas uma vez. Execute o novo programa e mostre que o resultado é o mesmo obtido no exemplo.

- 3.14** Escreva uma função no MATLAB que determine a raiz de uma equação não-linear  $f(x) = 0$  usando o método da bisseção. Chame a função de `Xs = BissecaoRaiz(Fun, a, b, TolMax)`. O argumento de saída `Xs` é a solução. O argumento de entrada `Fun` é o nome (*string*) de uma função em arquivo que calcula  $f(x)$  para um dado  $x$ , `a` e `b` são dois pontos que confinam a raiz, e `TolMax` é a tolerância máxima.

O programa deve conter as seguintes características:

- Verificar se os pontos `a` e `b` estão dos lados opostos da solução. Se não estiverem, o programa deve parar e mostrar uma mensagem de erro.
- O número de iterações deve ser determinado usando a equação no Problema 3.1, antes que as iterações sejam realizadas.

Use `BissecaoRaiz` para resolver a equação no Problema 3.1.

- 3.15** A determinação da raiz quadrada de um número  $p$ ,  $\sqrt{p}$ , é o mesmo que obter uma solução para a equação  $f(x) = x^2 - p = 0$ . Escreva uma função no MATLAB que determine a raiz quadrada de um número positivo usando o método de Newton. Chame a função de `[Xs] = RaizQuadrada(p)`. O argumento `Xs` é a resposta e o argumento de entrada `p` é o número cuja raiz quadrada deseja-se determinar. O programa deve incluir as seguintes características:

- Ele deve verificar se o número é positivo. Se não for, o programa deve parar e mostrar uma mensagem de erro.
- O valor inicial de  $x$  para as iterações deve ser  $x = p$ .
- As iterações devem parar quando o erro relativo estimado (Eq. (3.9)) for menor que 0,00001.
- O número de iterações deve ser limitado a 20. Se a solução não for obtida em 20 iterações, o programa deve parar e mostrar uma mensagem de erro.

- 3.16** Propõe-se um novo método para a solução de equações não-lineares, chamado de método da trisseção. Esse método é similar ao método da bisseção. A solução começa com a determinação de um intervalo  $[a, b]$  que confine a solução. A primeira estimativa da solução é o ponto central entre  $x = a$  e  $x = b$ . Então, conforme sugerido pelo nome do método, o intervalo  $[a, b]$  é dividido em três seções iguais. A seção que contém a raiz é escolhida como o novo intervalo para a iteração seguinte.

Escreva uma função no MATLAB que resolva uma equação não-linear usando o novo método. Chame a função de `Xs = TrissecaoRaiz(Fun, a, b, TolMax)`, onde o argumento de saída `Xs` é a solução. O argumento de entrada `Fun` é o nome (*string*) de uma função em arquivo que calcula  $f(x)$  para um dado  $x$ , `a` e `b` são dois pontos que confinam a raiz, e `TolMax` é a tolerância máxima.

Use a função TrissecaoRaiz para solucionar as equações dos Problemas 3.2 e 3.3. Para os valores iniciais de  $a$  e  $b$ , use os valores listados na letra (a) dos dois problemas, e, para TolMax, use 0,0001.

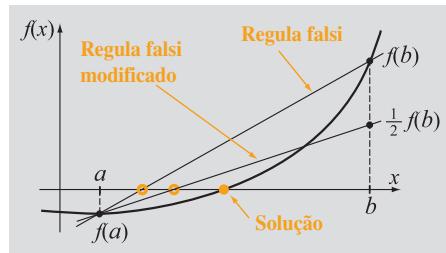
- 3.17** Escreva uma função no MATLAB que resolva uma equação não-linear  $f(x) = 0$  usando o método regula falsi. Chame a função de  $Xs = \text{RegulaRaiz}(\text{Fun}, a, b, \text{ErrMax})$ , onde  $Xs$  é a solução. Os argumentos de entrada são  $\text{Fun}$ , nome (string) de uma função em arquivo que calcula  $f(x)$  para um dado  $x$ ,  $a$  e  $b$ , que são dois pontos que confinam a raiz, e  $\text{ErrMax}$ , que é o erro máximo de acordo com a Eq. (3.9).

O programa deve ter as seguintes características:

- Verificar se os pontos  $a$  e  $b$  estão dos lados opostos da solução. Se não estiverem, o programa deve parar e mostrar uma mensagem de erro.
- O número de iterações deve estar limitado em 100 (para evitar a ocorrência de um *loop* infinito). Se uma solução com a precisão desejada não tiver sido obtida em 100 iterações, o programa deve parar e mostrar uma mensagem de erro.

Use a função RegulaRaiz para solucionar a equação do Problema 3.3 (use  $a = 1$ ,  $b = 2$ ). Para ErrMax, use 0,0001.

- 3.18** Freqüentemente, no método regula falsi (ver Seção 3.4), um dos pontos finais do intervalo não se altera durante as iterações, enquanto o outro avança em direção à raiz. No método regula falsi modificado, quando essa situação ocorre, a linha reta que conecta os pontos finais do intervalo é trocada por uma reta com inclinação menor. Conforme mostrado na figura, isso é feito com a divisão do valor da função no ponto que não se move por 2. Conseqüentemente, a reta cruza o eixo  $x$  em um ponto mais próximo à solução. Escreva uma função no MATLAB que resolva uma equação não-linear  $f(x) = 0$  com o método regula falsi modificado. O programa deve verificar se os pontos finais não se movem; se um ponto final não se mover em três iterações, o valor da função naquele ponto deve ser dividido por 2 quando a Eq. (3.11) for usada. Se o mesmo ponto não se mover nas três iterações seguintes, o valor da função naquele ponto deve ser dividido novamente por 2 (o valor real está agora dividido por 4), e assim por diante. Chame a função de  $Xs = \text{RegulaRaizMod}(\text{Fun}, a, b, \text{ErrMax})$ . Os argumentos da função e as suas características são os mesmos do Problema 3.17.



Use a função RegulaRaizMod para resolver a equação do Problema 3.3 (use  $a = 1$ ,  $b = 2$ ). Para ErrMax, use 0,0001.

- 3.19** Modifique a função NewtonRaiz listada na Fig. 3-11 de tal forma que a saída possua três argumentos. Chame a função de  $[Xs, FXs, iact] = \text{NewtonRaizMod}(\text{Fun}, \text{FunDer}, Xest, \text{Err}, \text{imax})$ . O primeiro argumento é a solução, o segundo é o valor da função na solução e o terceiro é o número

real de iterações realizadas para se obter a solução. Use a função Newton-RaizMod para resolver a equação que aparece no Exemplo 3-2.

**3.20** Escreva uma função no MATLAB que determine todas as raízes reais contidas em um domínio específico de uma função não-linear  $f(x) = 0$  usando o método da bisseção. Chame a função de `R=BisTodRaizes(fun,a,b,TolMax)`. O argumento de saída `R` é um vetor cujos elementos são os valores das raízes. O argumento de entrada `Fun` é o nome (*string*) de uma função em arquivo que calcula  $f(x)$  para um dado  $x$ . Os argumentos `a` e `b` definem o domínio e `TolMax` é a tolerância máxima usada pelo método da bisseção quando o valor de cada raiz é calculado. Use o seguinte algoritmo:

1. Divida o domínio  $[a, b]$  em 10 subintervalos iguais de largura  $h$ , de forma que  $h = (b - a)/10$ .
2. Procure por mudanças de sinal em  $f(x)$  nos pontos finais de cada subintervalo.
3. Se uma mudança de sinal for identificada em um subintervalo, use o método da bisseção para determinar a raiz naquele subintervalo.
4. Divida o domínio  $[a, b]$  em 100 subintervalos iguais de largura  $h$ , de forma que  $h = (b - a)/100$ .
5. Repita o passo 2. Se uma mudança de sinal for identificada em um subintervalo, verifique se ela contém uma raiz que já tenha sido obtida. Senão, use o método da bisseção para determinar a raiz naquele subintervalo.
6. Se nenhuma nova raiz tiver sido identificada, interrompa o programa.
7. Se uma ou mais raízes tiverem sido identificadas, repita os passos 4-6, onde em cada repetição o número de subintervalos deve ser multiplicado por 10.

Use a função `BisTodRaizes`, com `TolMax = 0,0001`, para encontrar todas as raízes da equação  $x^4 - 5,5x^3 - 7,2x^2 + 43x + 36 = 0$ .

**3.21** Examine as diferenças entre o Erro Relativo Real, Eq. (3.8), e o Erro Relativo Estimado, Eq. (3.9), resolvendo numericamente a equação  $f(x) = 0,5e^{(2+x)} - 40 = 0$ . A solução exata da equação é  $x = \ln(80) - 2$ . Escreva um programa no MATLAB que resolva a equação usando o método de Newton. Inicie as iterações em  $x = 4$  e execute 11 iterações. Em cada iteração, calcule o Erro Relativo Real (ERR) e o Erro Relativo Estimado (ERE). Mostre os resultados em uma tabela com quatro colunas (crie um arranjo bidimensional), com o número de iterações na primeira coluna, a solução numérica estimada na segunda, e o ERR e o ERE na terceira e quarta colunas, respectivamente.

### Problemas na matemática, ciência e engenharia

Resolva os seguintes problemas no ambiente MATLAB. Use programas apresentados neste capítulo, programas desenvolvidos em problemas já resolvidos ou funções residentes do MATLAB.

**3.22** Quando se calcula o pagamento de uma hipoteca, a relação entre o montante do empréstimo `Empr`, o pagamento mensal `Mpag`, a duração do empréstimo

em anos *Anos*, e a taxa de juros anual *Taxa*, é dada pela equação (equação de anuidade):

$$MPag = \frac{Empr \cdot Taxa}{12 \left( 1 - \frac{1}{\left( 1 + \frac{Taxa}{12} \right)^{12 \cdot Anos}} \right)}$$

Determine a *Taxa* de um empréstimo de R\$ 170.000 por 20 anos se o pagamento mensal for de R\$ 1.250.

- (a) Use a função *NewtonRaiz* dada no Programa 3-2. Use 0,001 para *Err*, e 1 para *Xest*.
- (b) Use a função *fzero* do MATLAB.

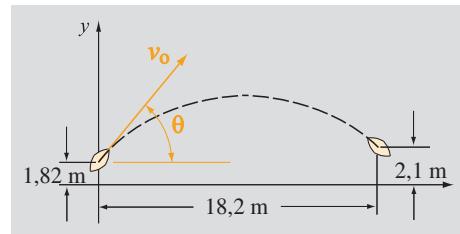
**3.23** A resistência elétrica  $R(T)$  de um termistor varia com a temperatura de acordo com:

$$R(T) = 100(1 + AT - BT^2)$$

onde  $R$  é dada em  $\Omega$ ,  $A = 3,90802 \times 10^{-3}$ ,  $B = 0,580195 \times 10^{-6} \text{ } ^\circ\text{C}^{-2}$  e  $T$  é a temperatura em graus Celsius. Determine a temperatura correspondente a uma resistência de 200  $\Omega$ .

- (a) Use a função *BissecaoRaiz* do Problema 3.14. Use 0,001 para *TolMax*.
- (b) Use a função *fzero* do MATLAB.

**3.24** Um jogador de futebol americano está prestes a fazer um lançamento para um outro jogador de seu time. O lançador tem uma altura de 1,82 m e o outro jogador está 18,2 m afastado. A expressão que descreve o movimento da bola é a familiar equação da física que descreve o movimento de um projétil:



$$y = x \tan(\theta) - \frac{1}{2} \frac{x^2 g}{v_0^2 \cos^2(\theta)} + h$$

onde  $x$  e  $y$  são as distâncias horizontal e vertical, respectivamente,  $g = 9,8 \text{ m/s}^2$  é a aceleração da gravidade,  $v_0$  é a velocidade inicial da bola quando deixa a mão do lançador e  $\theta$  é o ângulo que a bola faz com o eixo horizontal nesse mesmo instante. Para  $v_0 = 15,2 \text{ m/s}$ ,  $x = 18,2 \text{ m}$ ,  $h = 1,82 \text{ m}$  e  $y = 2,1 \text{ m}$ , determine o ângulo  $\theta$  no qual o jogador deve lançar a bola.

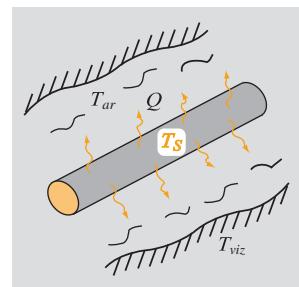
- (a) Use a função *BissecaoRaiz* desenvolvida no Problema 3.14. Use  $10^{-3} \text{ rad}$  como tolerância máxima.
- (b) Use a função *fzero* do MATLAB.

**3.25** Um cano de comprimento  $L = 25 \text{ m}$  e diâmetro  $d = 10 \text{ cm}$  conduzindo vapor perde calor para o ar ambiente e para as superfícies em sua vizinhança por

convecção e radiação. Se o fluxo total de calor por unidade de tempo  $Q$  emanando da superfície do cano for medida, então a temperatura superficial  $T_s$  do cano pode ser determinada pela seguinte equação:

$$Q = \pi dL[h(T_s - T_{ar}) + \varepsilon\sigma_{SB}(T_s^4 - T_{viz}^4)]$$

onde  $\varepsilon = 0,8$  é a emissividade da superfície do cano, e  $\sigma_{SB} = 5,67 \times 10^{-8} \text{ W/m}^2/\text{K}^4$  é a constante de Stefan-Boltzmann. Se  $Q = 18405 \text{ W}$ ,  $h = 10 \text{ W/m}^2/\text{K}$  e  $T_{ar} = T_{viz} = 298 \text{ K}$ , determine a temperatura superficial do cano,  $T_s$ .

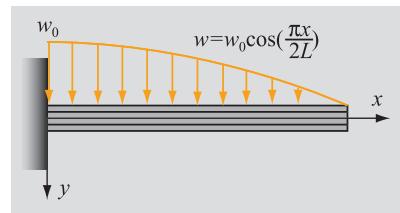


- (a) Use a função BissecãoRaiz desenvolvida no Problema 3.14 com um intervalo inicial  $[0, 1000]$ . Use  $10^{-2}$  rad como tolerância máxima.
- (b) Use a função fzero do MATLAB.

**3.26** Uma viga em balanço sustenta uma carga distribuída, conforme mostrado. A deflexão  $y$  da linha central da viga em função da posição  $x$  é dada pela equação:

$$y = \frac{w_0 L}{3\pi^4 EI} \left( 48L^3 \cos\left(\frac{\pi}{2L}x\right) - 48L^3 + 3\pi^3 Lx^2 - \pi^3 x^3 \right)$$

onde  $L = 3 \text{ m}$  é o comprimento,  $E = 70 \text{ GPa}$  é o módulo elástico,  $I = 52,9 \times 10^{-6} \text{ m}^4$  é o momento de inércia e  $w_0 = 15 \text{ kN/m}$ .

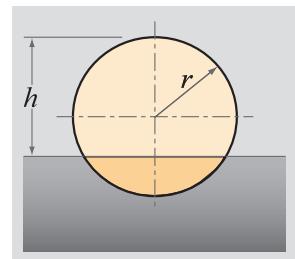


Determine a posição  $x$  onde a deflexão da viga é de 9 mm.

- (a) Use a função NewtonRaiz dada no Programa 3-2. Use Err = 0,0001 e 1,8 para Zest.
- (b) Use a função SecanteRaiz dada no Programa 3-3. Use Err = 1,5, Xa = 1,5 e Xb = 2,5.
- (c) Use a função fzero do MATLAB.

**3.27** De acordo com o princípio de Arquimedes, a força de empuxo agindo em um objeto parcialmente imerso em um fluido é igual ao peso deslocado pela porção submersa do objeto.

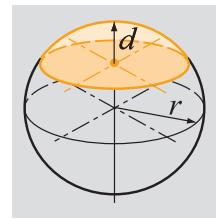
Um flutuador esférico com massa  $m_f = 70 \text{ kg}$  e diâmetro de 90 cm é colocado no oceano (a densidade da água do mar é de aproximadamente  $\rho = 1030 \text{ kg/m}^3$ ). A altura  $h$  da porção do flutuador acima da água pode ser determinada com a solução de uma equação que iguala a massa do flutuador à massa da água deslocada pela porção submersa do flutuador:



$$\rho V_{casca} = m_f \quad (3.59)$$

onde, para uma esfera de raio  $r$ , o volume de uma casca com profundidade  $d$  é dado por:

$$V_{\text{casca}} = \frac{1}{3}\pi d^2(3r - d)$$



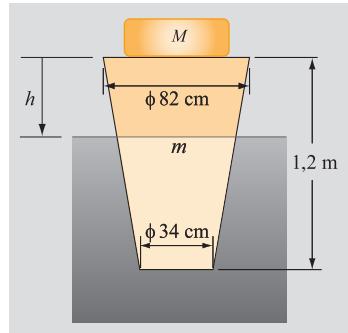
Escreva a Eq. (3.59) em termos de  $h$  e a resolva usando:

- (a) A função `NewtonRaiz` dada no Programa 3-2. Use  $\text{Err} = 0,0001$  e  $\text{Xest} = 0,8$ .
- (b) A função `fzero` do MATLAB.

- 3.28** De acordo com o princípio de Arquimedes, a força de empuxo agindo em um objeto parcialmente imerso em um fluido é igual ao peso deslocado pela porção submersa do objeto.

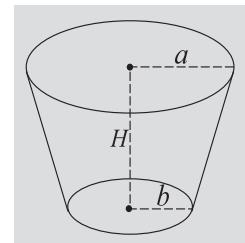
Uma bóia na forma de um cone cortado com massa  $m = 50$  kg é colocada no oceano (a densidade da água do mar é de aproximadamente  $\rho = 1030$  kg/m<sup>3</sup>). Um equipamento com massa  $M$  é colocado sobre a bóia, conforme mostrado na figura. A altura  $h$  da porção da bóia acima da água pode ser determinada com a solução de uma equação que iguala a massa total ( $m + M$ ) à massa de água deslocada pelo volume  $V$  da bóia que se encontra submerso:

$$m + M = \rho V \quad (3.60)$$



O volume de um cone cortado com raios  $a$  e  $b$  e altura  $H$  é dado por:

$$V = \frac{1}{3}\pi H(a^2 + ab + b^2)$$

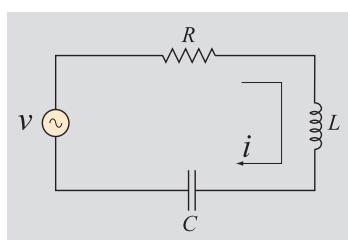


Escreva a Eq. (3.60) em termos de  $h$ .

- (a) Determine (manualmente) a maior massa  $M$  que pode ser colocada sobre a bóia sem que  $M$  toque a água (a massa para a qual  $h = 0$ ).
- (b) Escreva um programa no MATLAB que determine (usando `fzero`) o valor de  $h$  para diferentes valores de  $M$ , começando com  $M = 0$  e usando incrementos de 1 kg até que se atinja a máxima massa  $M$  permitida. Em seguida, trace um gráfico de  $h$  versus  $M$ .

- 3.29** Um circuito *RLC* consiste em um resistor  $R$ , um indutor  $L$  e um capacitor  $C$  conectados em série com uma fonte de tensão alternada  $V$ . A amplitude da corrente  $i_m$  é dada por:

$$i_m = \frac{v_m}{\sqrt{R^2 + [\omega L - 1/(\omega C)]^2}}$$



onde  $\omega$ , a freqüência angular, está relacionada à freqüência  $f$  por  $\omega = 2\pi f$ . Determine  $f$  para um circuito com  $R = 140 \Omega$ ,  $L = 260 \text{ mH}$ ,  $C = 25 \mu\text{F}$ ,  $v_m = 24 \text{ V}$  e  $i_m = 0,15 \text{ A}$  usando:

- A função SecanteRaiz dada no Programa 3-3. Use Err = 0,0001, Xa = 30 e Xb = 50.
- A função fzero do MATLAB.

**3.30** A potência de saída de uma célula solar varia com a tensão que ela fornece. A saída  $V_{mp}$  para a qual a potência de saída é máxima é dada pela equação:

$$e^{(qV_{mp}/k_B T)} \left(1 + \frac{qV_{mp}}{k_B T}\right) = e^{(qV_{OC}/k_B T)}$$

onde  $V_{OC}$  é a tensão de circuito aberto,  $T$  é a temperatura em Kelvin,  $q = 1,6022 \times 10^{-19} \text{ C}$  é a carga de um elétron e  $k_B = 1,3806 \times 10^{-23} \text{ J/K}$  é a constante de Boltzmann. Para  $V_{OC} = 0,5 \text{ V}$  e uma temperatura  $T = 297 \text{ K}$ , determine a tensão  $V_{mp}$  na qual a potência de saída da célula solar é máxima.

- Escreva um programa no MATLAB que use o método da iteração de ponto fixo para obter a raiz. Utilize como ponto de partida  $V_{mp} = 0,5 \text{ V}$ . Para interromper as iterações, use a Eq. (3.18) com  $\epsilon = 0,001$ .
- Use a função fzero do MATLAB.

**3.31** Perdas friccionais ou de viscosidade no fluxo de líquidos em um cano resultam em quedas de pressão que devem ser superadas. Estas são expressas em termos de um fator de atrito,  $f$ . Para o fluxo turbulento em canos,  $f$  é calculado a partir da equação de Colebrook:

$$\frac{1}{f^{0.5}} = -2 \log_{10} \left( \frac{e/D}{3,7} + \frac{2,51}{(Re)f^{0.5}} \right)$$

onde  $e$  é a aspereza,  $D$  é o diâmetro do cano, e  $Re$  é o número de Reynolds, que é adimensional. Para  $e/D = 0,004$  e  $Re = 2 \times 10^5$ , resolva essa equação para  $f$ .

- Use o método da iteração de ponto fixo considerando a seguinte função de iteração:

$$g(f) = \frac{1}{4 \left[ \log_{10} \left( \frac{e/D}{3,7} + \frac{2,51}{(Re)f^{0.5}} \right) \right]^2}$$

Interrompa as iterações quando o erro relativo máximo estimado de acordo com a Eq. (3.18) for menor que 0,001.

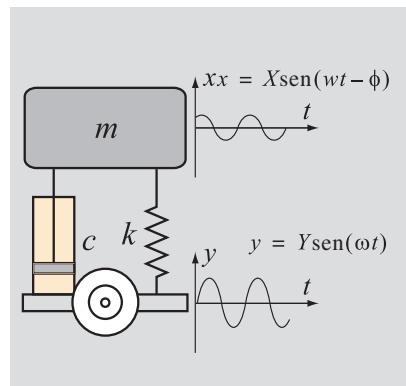
- Use a função fzero do MATLAB.

**3.32** Um modelo simplificado para a suspensão de um automóvel consiste em uma massa  $m$ , uma mola com constante elástica  $k$  e um amortecedor com constante de amortecimento  $c$ , conforme mostrado na figura. Uma estrada esburacada pode ser modelada assumindo-se que a roda se move para cima e para baixo de acordo com a equação  $y = Y_{sen}(\omega t)$ . A partir da solução dessa

equação, o movimento do carro (massa) para cima e para baixo é dado por  $x = X \sin(\omega t - \phi)$ . A razão entre a amplitude  $X$  e a amplitude  $Y$  é dada por:

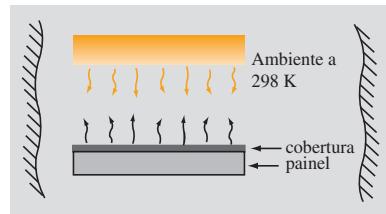
$$\frac{X}{Y} = \sqrt{\frac{mc\omega^3}{k(k-m\omega^2) + (\omega c)^2}}$$

Assumindo  $m = 2000$  kg,  $k = 500$  kN/m e  $c = 38 \times 10^3$  N·s/m, determine a freqüência  $\omega$  na qual  $X / Y = 0,2$ . Rescreva a equação para que ela tenha a forma de um polinômio em  $\omega$  e a resolva usando:



- (a) A função `BissecaoRaiz` desenvolvida no Problema 3.14. Use  $10^{-2}$  como tolerância máxima.
- (b) A função `fzero` do MATLAB.

**3.33** A cobertura de um painel é submetida à energia de radiação emanada de um aquecedor. A temperatura da cobertura é determinada pelos processos de transferência de calor na forma de radiação e convecção. Se a radiação for tratada como difusa e cinza, o seguinte sistema de equações não-lineares simultâneas permite o cálculo das incógnitas  $J_h, T_h, J_c, T_c$ :



$$5,67 \times 10^{-8} T_c^4 + 17,41 T_c - J_c = 5188,18$$

$$J_c - 0,71 J_h + 7,46 T_c = 2352,71$$

$$5,67 \times 10^{-8} T_h^4 + 1,865 T_h - J_h = 2250$$

$$J_h - 0,71 J_c + 7,46 T_h = 11093$$

onde  $J_h$  e  $J_c$  são as radiosidades das superfícies do aquecedor e da cobertura, respectivamente, e  $T_h$  e  $T_c$  são as respectivas temperaturas.

- (a) Mostre que as seguintes funções de iteração podem ser usadas para resolver o sistema de equações não-lineares com o método da iteração de ponto fixo:

$$T_c = \left[ \frac{J_c - 17,41 T_c + 5188,18}{5,67 \times 10^{-8}} \right]^{1/4} \quad T_h = \left[ \frac{2250 + J_h - 1,865 T_h}{5,67 \times 10^{-8}} \right]^{1/4}$$

$$J_c = 2352,71 + 0,71 J_h - 7,46 T_c \quad J_h = 11093 + 0,71 J_c - 7,46 T_h$$

- (b) Resolva o sistema de equações não-lineares com o método da iteração de ponto fixo usando as funções de iteração da letra (a). Utilize os seguintes valores iniciais:  $T_h = T_c = 298$  K,  $J_c = 3000$  W/m<sup>2</sup> e  $J_h = 5000$  W/m<sup>2</sup>. Calcule 100 iterações e trace um gráfico com os respectivos valores para observar a sua convergência. As respostas finais devem ser  $T_c = 481$  K,  $J_c = 6222$  W/m<sup>2</sup>,  $T_h = 671$  K,  $J_h = 10504$  W/m<sup>2</sup>.

## Resolvendo um Sistema de Equações Lineares

### Tópicos centrais

- Método de eliminação de Gauss (4.2)
- Eliminação de Gauss com pivotação (4.3)
- Método de eliminação de Gauss-Jordan (4.4)
- Método de decomposição LU (4.5)
- Inversa de uma Matriz (4.6)
- Métodos iterativos (Jacobi, Gauss-Seidel) (4.7)

Uso de funções residentes do MATLAB para resolver sistemas de equações lineares (4.8)

### Tópicos complementares

- Sistemas tridiagonais de equações (4.9)
- Erro, resíduo, normas e número condição (4.10)
- Sistemas mal condicionados (4.11)
- Autovalores e autovetores (4.12)

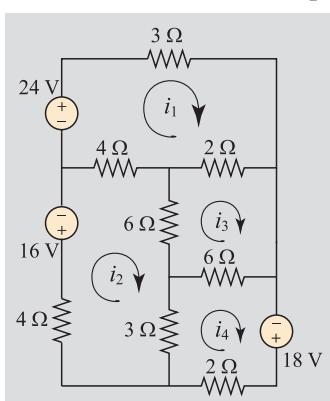
### 4.1 FUNDAMENTOS

Sistemas de equações lineares aparecem em problemas que contêm muitas variáveis dependentes. Tais problemas ocorrem não apenas na engenharia e na ciência, que são o foco deste livro, mas também em virtualmente todas as demais disciplinas (negócios, estatística, economia, etc.). Um sistema de duas (ou três) equações com duas (ou três) incógnitas pode ser resolvido manualmente por substituição ou com o uso de métodos matemáticos (por exemplo, a regra de Cramer, Seção 2.4.6). Resolver um sistema dessa maneira é praticamente impossível se o número de equações (e incógnitas) for maior que três.

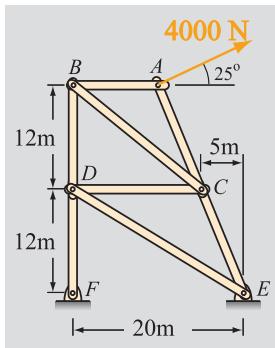
Um exemplo de um problema na engenharia elétrica que requer a solução de um sistema de equações é mostrado na Fig. 4-1. Usando a lei de Kirchhoff, as correntes  $i_1$ ,  $i_2$ ,  $i_3$  e  $i_4$  podem ser determinadas com a solução do seguinte sistema de quatro equações:

$$\begin{aligned} 9i_1 - 4i_2 - 2i_3 &= 24 \\ -4i_1 + 17i_2 - 6i_3 - 3i_4 &= -16 \\ -2i_1 - 6i_2 + 14i_3 - 6i_4 &= 0 \\ -3i_2 - 6i_3 + 11i_4 &= 18 \end{aligned} \quad (4.1)$$

Obviamente, circuitos mais complicados podem requerer solução de sistemas com um número maior de equações. Outro exemplo é o cálculo da força nos membros de uma treliça. As forças nos oito membros da treliça mostrada na Fig. 4-2



**Figura 4-1** Circuito elétrico.



**Figura 4-2** Treliça com oito membros.

são determinadas a partir da solução do seguinte sistema de oito equações (equações de equilíbrio dos pinos A, B, C e D):

$$\begin{aligned} 0,9231F_{AC} &= 1690 & -F_{AB} - 0,3846F_{AC} &= 3625 \\ F_{AB} - 0,7809F_{BC} &= 0 & 0,6247F_{BC} - F_{BD} &= 0 \\ F_{CD} + 0,8575F_{DE} &= 0 & F_{BD} - 0,5145F_{DE} - F_{DF} &= 0 \quad (4.2) \\ 0,3846F_{CE} - 0,3846F_{AC} - 0,7809F_{BC} - F_{CD} &= 0 \\ 0,9231F_{AC} + 0,6247F_{BC} - 0,9231F_{CE} &= 0 \end{aligned}$$

Há aplicações, por exemplo, na análise por diferenças e elementos finitos, onde o sistema de equações a ser resolvido contém milhares (ou mesmo milhões) de equações simultâneas.

#### 4.1.1 Revisão de métodos numéricos para resolver sistemas de equações lineares algébricas

A forma geral de um sistema contendo  $n$  equações lineares algébricas é:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad \begin{array}{lllll} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n & = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n & = b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n & = b_n \end{array} \quad (4.3)$$

**Figura 4-3** Um sistema com  $n$  equações lineares algébricas.

A forma matricial das equações é mostrada na Fig. 4-3. Dois tipos de métodos numéricos, **diretos** e **iterativos**, são usados para resolver sistemas de equações lineares algébricas. Nos métodos diretos, a solução é obtida com a realização de operações

algébricas nas equações. Nos métodos iterativos, uma solução inicial aproximada é assumida e então utilizada em um processo iterativo para que soluções mais precisas sejam obtidas sucessivamente.

#### Métodos diretos

Nos métodos diretos, o sistema de equações inicial dado na forma das Eqs. (4.3) é manipulado até se transformar em um sistema equivalente de fácil resolução que tenha, por exemplo, as formas **triangular superior**, **triangular inferior** ou **diagonal**.

A forma **triangular superior** é mostrada nas Eqs. (4.4) e escrita na forma matricial para um sistema de quatro equações na Fig. 4-4.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \quad \begin{array}{lllll} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n & = b_1 \\ a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n & = b_2 \\ a_{33}x_3 + \dots + a_{3n}x_n & = b_3 \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n & = b_{n-1} \\ a_{nn}x_n & = b_n \end{array} \quad (4.4)$$

**Figura 4-4** Sistema de quatro equações na forma triangular superior.

O sistema nessa forma tem todos os coeficientes nulos abaixo da diagonal e é resolvido por um procedimento chamado de

**substituição regressiva.** Ele começa com a última equação, que é resolvida para  $x_n$ . O valor de  $x_n$  é então substituído na equação anterior, que é resolvida para  $x_{n-1}$ . O processo continua da mesma maneira até alcançar a primeira equação. No caso de quatro equações, a solução é dada por:

$$\begin{aligned}x_4 &= \frac{b_4}{a_{44}}, & x_3 &= \frac{b_3 - a_{34}b_4}{a_{33}}, & x_2 &= \frac{b_2 - (a_{23}b_3 + a_{24}b_4)}{a_{22}} \\x_4 &= \frac{b_4 - (a_{12}b_2 + a_{13}b_3 + a_{14}b_4)}{a_{11}}\end{aligned}\quad \text{e}$$

Para um sistema com  $n$  equações na forma triangular superior, uma fórmula geral para a solução usando a substituição regressiva é:

$$\begin{aligned}x_n &= \frac{b_n}{a_{nn}} \\x_i &= \frac{b_i - \sum_{j=i+1}^{j=n} a_{ij}b_j}{a_{ii}} \quad i = n-1, n-2, \dots, 1\end{aligned}\quad (4.5)$$

Na Seção 4.2, a forma triangular superior e a substituição regressiva são usadas no método de eliminação de Gauss.

A forma **triangular inferior** é mostrada na Eq. (4.6) e escrita na forma matricial para um sistema de quatro equações na Fig. 4-5.

$$\left[\begin{array}{cccc} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\right] = \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ b_4 \end{array}\right]$$

$$\begin{aligned}a_{11}x_1 &= b_1 \\a_{21}x_1 + a_{22}x_2 &= b_2 \\a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \\&\vdots &&\vdots \\a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n\end{aligned}\quad (4.6)$$

**Figura 4-5** Sistema de quatro equações na forma triangular inferior.

O sistema nessa forma tem todos os coeficientes nulos acima da diagonal e é resolvido da mesma maneira que a forma triangular superior, porém no sentido oposto. O procedimento é chamado de **substituição progressiva**. Ele começa com a primeira equação, que é resolvida para  $x_1$ . O valor de  $x_1$  é então substituído na segunda equação, que é resolvida para  $x_2$ . O processo continua da mesma maneira até alcançar a última equação. No caso de quatro equações, a solução é dada por:

$$\begin{aligned}x_1 &= \frac{b_1}{a_{11}}, & x_2 &= \frac{b_2 - a_{21}b_1}{a_{22}}, & x_3 &= \frac{b_3 - (a_{31}b_1 + a_{32}b_2)}{a_{33}} \\x_4 &= \frac{b_4 - (a_{41}b_1 + a_{42}b_2 + a_{43}b_3)}{a_{44}}\end{aligned}\quad \text{e}$$

Para um sistema com  $n$  equações na forma triangular inferior, uma fórmula geral para a solução usando a substituição progressiva é:

$$x_1 = \frac{b_1}{a_{11}}$$

$$x_i = \frac{b_i - \sum_{j=i}^{i-1} a_{ij}b_j}{a_{ii}} \quad i = 2, 3, \dots, n \quad (4.8)$$

Na Seção 4.5, a forma triangular inferior é usada em conjunto com a forma triangular superior no método de decomposição *LU*.

A forma **diagonal** de um sistema de equações lineares é mostrada nas Eqs. (4.9) e escrita na forma matricial para um sistema de quatro equações na Fig. 4-6.

$$\begin{bmatrix} a_{11} & a & a & a \\ a & a_{22} & a & a \\ a & a & a_{33} & a \\ a & a & a & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

**Figura 4-6** Sistema de quatro equações na forma diagonal.

Um sistema na forma diagonal possui coeficientes diferentes de zero ao longo da diagonal e iguais a zero em todas as demais posições. Obviamente, um sistema nessa forma pode ser facilmente resolvido. Uma forma similar é usada no método de Gauss-Jordan, que é apresentado na Seção 4.4.

A partir das três formas apresentadas para as equações lineares simultâneas (triangular superior, triangular inferior e diagonal), pode parecer que a mudança de um sistema de equações para a forma diagonal seja a melhor escolha, porque o sistema diagonalizado é o de mais fácil resolução. Na realidade, no entanto, o número total de operações necessárias para resolver um sistema é menor quando outros métodos são usados.

Três métodos diretos para resolver sistemas de equações – a eliminação de Gauss (Seções 4.2 e 4.3), o método de Gauss-Jordan (Seção 4.4) e a decomposição *LU* (Seção 4.5) – e dois métodos indiretos (iterativos) – de Jacobi e de Gauss-Seidel – são descritos neste capítulo.

## 4.2 MÉTODO DE ELIMINAÇÃO DE GAUSS

O método de eliminação de Gauss é um procedimento usado para resolver sistemas de equações lineares. Nesse procedimento, um sistema de equações genérico é manipulado até apresentar a forma **triangular superior**, que é então resolvida com o emprego da substituição regressiva (ver Seção 4.1.1). Para um conjunto de quatro equações com quatro incógnitas, a forma geral é dada por:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 \quad (4.10a)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \quad (4.10b)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3 \quad (4.10c)$$

$$a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4 \quad (4.10d)$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

**Figura 4-7** Forma matricial de um sistema de quatro equações.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a'_{33} & a'_{34} \\ 0 & 0 & 0 & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

**Figura 4-8** Forma matricial do sistema equivalente.

da nas Eqs. (4.11). Uma em particular, o método de eliminação de Gauss, é descrita a seguir. Esse procedimento pode ser facilmente programado em um computador.

### Procedimento de eliminação de Gauss (eliminação progressiva)

O procedimento de eliminação de Gauss é ilustrado inicialmente para um sistema de quatro equações com quatro incógnitas. O ponto de partida é o conjunto de equações dado pelas Eqs. (4.10). A conversão desse sistema de equações para a forma dada nas Eqs. (4.11) é feita em passos.

**Passo 1** No primeiro passo, a primeira equação não é alterada e os termos que incluem a variável  $x_1$  nas demais equações são eliminados. Isso é feito uma equação por vez usando a primeira equação, que é chamada de **equação pivô**. O coeficiente  $a_{11}$  é chamado de **coeficiente pivô**, ou de elemento pivô. Para eliminar o termo  $a_{21}x_1$  na Eq. (4.10b), a equação pivô, Eq. (4.10a), é multiplicada por  $m_{21} = a_{21}/a_{11}$  e então subtraída da Eq. (4.10b):

$$\begin{aligned} - & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \\ m_{21}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) &= m_{21}b_1 \\ 0 + (a_{22} - m_{21}a_{12})x_2 + (a_{23} - m_{21}a_{13})x_3 + (a_{24} - m_{21}a_{14})x_4 &= b_2 - m_{21}b_1 \end{aligned}$$

Deve-se enfatizar aqui que a equação pivô, Eq. (4.10a), não é alterada. A forma matricial das equações após essa operação é mostrada na Fig. 4-9.

Em seguida, elimina-se o termo  $a_{31}x_1$  da Eq. (4.10c). A equação pivô, Eq. (4.10a), é multiplicada por  $m_{31} = a_{31}/a_{11}$  e então subtraída da Eq. (4.10c):

$$\begin{array}{rcl}
 - & a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3 \\
 & m_{31}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) = m_{31}b_1 \\
 \hline
 0 + (a_{32} - m_{31}a_{12})x_2 + (a_{33} - m_{31}a_{13})x_3 + (a_{34} - m_{31}a_{14})x_4 = b_3 - m_{31}b_1 \\
 \quad \quad \quad \underbrace{a'_{32}} \quad \quad \quad \underbrace{a'_{33}} \quad \quad \quad \underbrace{a'_{34}} \quad \quad \quad \underbrace{b'_3}
 \end{array}$$

$$\left[ \begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[ \begin{array}{c} b_1 \\ b'_2 \\ b'_3 \\ b_4 \end{array} \right]$$

**Figura 4-10** Forma matricial do sistema após a eliminação de  $a_{31}$ .

A forma matricial das equações após essa operação é mostrada na Fig. 4-10.

Em seguida, elimina-se o termo  $a_{41}x_1$  da Eq. (4.10d). A equação pivô, Eq. (4.10a), é multiplicada por  $m_{41} = a_{41}/a_{11}$  e então subtraída da Eq. (4.10d):

$$\begin{array}{rcl}
 - & a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4 \\
 & m_{41}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) = m_{41}b_1 \\
 \hline
 0 + (a_{42} - m_{41}a_{12})x_2 + (a_{43} - m_{41}a_{13})x_3 + (a_{44} - m_{41}a_{14})x_4 = b_4 - m_{41}b_1 \\
 \quad \quad \quad \underbrace{a'_{42}} \quad \quad \quad \underbrace{a'_{43}} \quad \quad \quad \underbrace{a'_{44}} \quad \quad \quad \underbrace{b'_4}
 \end{array}$$

Este é o final do **Passo 1**. O sistema de equações tem agora a seguinte forma:

$$\left[ \begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[ \begin{array}{c} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{array} \right]$$

**Figura 4-11** Forma matricial do sistema após a eliminação de  $a_{41}$ .

$$\begin{array}{ll}
 a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 & (4.12a) \\
 0 + a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 = b'_2 & (4.12b) \\
 0 + a'_{32}x_2 + a'_{33}x_3 + a'_{34}x_4 = b'_3 & (4.12c) \\
 0 + a'_{42}x_2 + a'_{43}x_3 + a'_{44}x_4 = b'_4 & (4.12d)
 \end{array}$$

A forma matricial das equações após essa operação é mostrada na Fig. 4-11. Note que o resultado da operação de eliminação é a redução dos campos da primeira coluna, exceção  $a_{11}$  (o elemento pivô), a zero.

**Passo 2** Neste passo, as Eqs. (4.12a) e (4.12b) não são alteradas, e os termos que incluem a variável  $x_2$  nas Eqs. (4.12c) e (4.12d) são eliminados. Neste passo, a Eq. (4.12b) é a equação pivô, e o coeficiente  $a'_{22}$  é o coeficiente pivô. Para eliminar o termo  $a'_{32}x_2$  na Eq. (4.12c), a equação pivô, Eq. (4.12b), é multiplicada por  $m_{32} = a'_{32}/a'_{22}$  e então subtraída da Eq. (4.12c):

$$\begin{array}{rcl}
 - & a'_{32}x_2 + a'_{33}x_3 + a'_{34}x_4 = b'_3 \\
 & m_{32}(a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4) = m_{32}b'_2
 \end{array}$$

$$\left[ \begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[ \begin{array}{c} b_1 \\ b'_2 \\ b''_3 \\ b'_4 \end{array} \right]$$

**Figura 4-12** Forma matricial do sistema após a eliminação de  $a'_{32}$ .

$$\begin{array}{rcl}
 0 + (a'_{33} - m_{32}a'_{23})x_3 + (a'_{34} - m_{32}a'_{24})x_4 = b'_3 - m_{32}b'_2 \\
 \quad \quad \quad \underbrace{a''_{33}} \quad \quad \quad \underbrace{a''_{34}} \quad \quad \quad \underbrace{b''_3}
 \end{array}$$

A forma matricial das equações após essa operação é mostrada na Fig. 4-12.

Em seguida, elimina-se o termo  $a'_{42}x_2$ . A equação pivô, Eq. (4.12b), é multiplicada por  $m_{42} = a'_{42}/a'_{22}$  e então subtraída da Eq. (4.12d):

$$a'_{42}x_2 + a'_{43}x_3 + a'_{44}x_4 = b'_4$$

$$m_{42}(a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4) = m_{42}b'_2$$

$$\begin{array}{rcl} 0 + (a'_{43} - m_{42}a'_{23})x_3 + (a'_{44} - m_{42}a'_{24})x_4 & = & b'_4 - m_{42}b'_2 \\ \underbrace{a''_{43}} & \quad \underbrace{a''_{44}} & \quad \underbrace{b''_4} \end{array}$$

Este é o final do **Passo 2**. O sistema de equações tem agora a seguinte forma:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 \quad (4.13a)$$

$$0 + a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 = b'_2 \quad (4.13b)$$

$$0 + 0 + a''_{33}x_3 + a''_{34}x_4 = b''_3 \quad (4.13c)$$

$$0 + 0 + a''_{43}x_3 + a''_{44}x_4 = b''_4 \quad (4.13d)$$

A forma matricial das equações no final do **Passo 2** é mostrada na Fig. 4-13.

**Passo 3** Neste passo, as Eqs. (4.13a), (4.13b) e (4.13c) não são alteradas, e o termo que inclui a variável  $x_3$  nas Eqs. (4.13d) é eliminado. Agora, a Eq. (4.13c) é a equação pivô, e o coeficiente  $a''_{33}$  é o coeficiente pivô. Para eliminar o termo  $a''_{43}x_3$  da Eq. (4.13d), a equação pivô é multiplicada por  $m_{43} = a''_{43}/a''_{33}$  e então subtraída da Eq. (4.13d):

$$a''_{43}x_3 + a''_{44}x_4 = b''_4$$

$$m_{43}(a''_{33}x_3 + a''_{34}x_4) = m_{43}b''_3$$

$$\begin{array}{rcl} (a''_{44} - m_{43}a''_{34})x_4 & = & b''_4 - m_{43}b''_3 \\ \underbrace{a'''_{44}} & \quad \underbrace{b'''_4} & \end{array}$$

Este é o final do **Passo 3**. O sistema de equações está agora na forma triangular superior:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1 \quad (4.14a)$$

$$0 + a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 = b'_2 \quad (4.14b)$$

$$0 + 0 + a''_{33}x_3 + a''_{34}x_4 = b''_3 \quad (4.14c)$$

$$0 + 0 + 0 + a''_{44}x_4 = b''_4 \quad (4.14d)$$

A forma matricial das equações é mostrada na Fig. 4-14. Uma vez transformada para a forma triangular superior, as equações podem ser facilmente resolvidas com o uso da substituição regressiva. Os três passos do processo de eliminação de Gauss são ilustrados conjuntamente na Fig. 4-15.

$$\left[ \begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a''_{43} & a''_{44} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[ \begin{array}{c} b_1 \\ b'_2 \\ b''_3 \\ b''_4 \end{array} \right]$$

**Figura 4-13** Forma matricial do sistema após a eliminação de  $a'_{42}$ .

$a''_{43}x_3$  da Eq. (4.13d), a equação pivô é multiplicada por  $m_{43} = a''_{43}/a''_{33}$  e então subtraída da Eq. (4.13d):

$$\left[ \begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a''_{44} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[ \begin{array}{c} b_1 \\ b'_2 \\ b''_3 \\ b''_4 \end{array} \right]$$

**Figura 4-14** Forma matricial do sistema após a eliminação de  $a''_{43}$ .

$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$ <p>Conjunto inicial de equações</p>	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ \cancel{a_{21}} & a'_{22} & a'_{23} & a'_{24} \\ \cancel{a_{31}} & a'_{32} & a'_{33} & a'_{34} \\ \cancel{a_{41}} & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$ <p>Passo 1</p>	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a'''_{43} & a'''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x'_2 \\ x''_3 \\ x'''_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$ <p>Passo 2</p>
$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a'''_{43} & a'''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x'_2 \\ x''_3 \\ x'''_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$ <p>Passo 3</p>	<p>Equações na forma triangular superior</p>	

**Figura 4-15** Procedimento de eliminação de Gauss.

O Exemplo 4-1 mostra a aplicação manual do método de eliminação de Gauss na solução de um sistema de quatro equações.

### Exemplo 4-1: Solução de um conjunto de quatro equações usando a eliminação de Gauss

Resolva o seguinte sistema de quatro equações usando o método de eliminação de Gauss.

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ -6x_1 + 7x_2 + 6,5x_3 - 6x_4 &= -6,5 \\ x_1 + 7,5x_2 + 6,25x_3 + 5,5x_4 &= 16 \\ -12x_1 + 22x_2 + 15,5x_3 - x_4 &= 17 \end{aligned}$$

#### SOLUÇÃO

A solução segue os passos apresentados nas páginas anteriores.

**Passo 1** A primeira equação é a equação pivô, e 4 é o coeficiente pivô.

Multiplique a equação pivô por  $m_{21} = (-6)/4 = -1,5$  e a subtraia da segunda equação:

$$\begin{array}{r} -6x_1 + 7x_2 + 6,5x_3 - 6x_4 = -6,5 \\ \hline (-1,5)(4x_1 - 2x_2 - 3x_3 + 6x_4) = (-6/4) \cdot 12 \\ \hline 0x_1 + 4x_2 + 2x_3 + 3x_4 = 11,5 \end{array}$$

Multiplique a equação pivô por  $m_{31} = (1/4) = 0,25$  e a subtraia da terceira equação:

$$\begin{array}{r} x_1 + 7,5x_2 + 6,25x_3 + 5,5x_4 = 16 \\ \hline (0,25)(4x_1 - 2x_2 - 3x_3 + 6x_4) = (1/4) \cdot 12 \\ \hline 0x_1 + 8x_2 + 7x_3 + 4x_4 = 13 \end{array}$$

Multiplique a equação pivô por  $m_{41} = (-12)/4 = -3$  e a subtraia da quarta equação:

$$\begin{array}{r} -12x_1 + 22x_2 + 15,5x_3 - x_4 = 17 \\ \hline (-3)(4x_1 - 2x_2 - 3x_3 + 6x_4) = -3 \cdot 12 \\ \hline 0x_1 + 16x_2 + 6,5x_3 + 17x_4 = 53 \end{array}$$

No final do **Passo 1**, as quatro equações têm a forma:

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ 4x_2 + 2x_3 + 3x_4 &= 11,5 \\ 8x_2 + 7x_3 + 4x_4 &= 13 \\ 16x_2 + 6,5x_3 + 17x_4 &= 53 \end{aligned}$$

**Passo 2** A segunda equação é a equação pivô, e 4 é o coeficiente pivô.

Multiplique a equação pivô por  $m_{32} = 8/4 = 2$  e a subtraia da terceira equação:

$$\begin{array}{r} 8x_2 + 7x_3 + 4x_4 = 13 \\ - 2(4x_2 + 2x_3 + 3x_4) = 2 \cdot 11,5 \\ \hline 0x_2 + 3x_3 - 2x_4 = -10 \end{array}$$

Multiplique a equação pivô por  $m_{42} = 16/4 = 4$  e a subtraia da quarta equação:

$$\begin{array}{r} 16x_2 + 6,5x_3 + 17x_4 = 53 \\ - 4(4x_2 + 2x_3 + 3x_4) = 4 \cdot 11,5 \\ \hline 0x_2 - 1,5x_3 + 5x_4 = 7 \end{array}$$

No final do **Passo 2**, as quatro equações têm a forma:

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ 4x_2 + 2x_3 + 3x_4 &= 11,5 \\ 3x_3 - 2x_4 &= -10 \\ -1,5x_3 + 5x_4 &= 7 \end{aligned}$$

**Passo 3** A terceira equação é a equação pivô, e 3 é o coeficiente pivô.

Multiplique a equação pivô por  $m_{43} = (-1,5)/3 = -0,5$  e a subtraia da quarta equação:

$$\begin{array}{r} -1,5x_3 + 5x_4 = 7 \\ -0,5(3x_3 - 2x_4) = -0,5 \cdot -10 \\ \hline 0x_3 + 4x_4 = 2 \end{array}$$

No final do **Passo 3**, as quatro equações têm a forma:

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ 4x_2 + 2x_3 + 3x_4 &= 11,5 \\ 3x_3 - 2x_4 &= -10 \\ 4x_4 &= 2 \end{aligned}$$

Com as equações escritas nessa forma, a solução pode ser determinada com o emprego da substituição regressiva. O valor de  $x_4$  é determinado com a solução da quarta equação:

$$x_4 = 2/4 = 0,5$$

Em seguida,  $x_4$  é substituído na terceira equação, que é resolvida para  $x_3$ :

$$x_3 = \frac{-10 + 2x_4}{3} = \frac{-10 + 2 \cdot 0,5}{3} = -3$$

Em seguida,  $x_4$  e  $x_3$  são substituídos na segunda equação, que é resolvida para  $x_2$ :

$$x_2 = \frac{11,5 - 2x_3 - 3x_4}{4} = \frac{11,5 - (2 \cdot -3) - (3 \cdot 0,5)}{4} = 4$$

Finalmente,  $x_4$ ,  $x_3$  e  $x_2$  são substituídos na primeira equação, que é resolvida para  $x_1$ :

$$x_1 = \frac{12 + 2x_2 + 3x_3 - 6x_4}{4} = \frac{12 + 2 \cdot 4 + 3 \cdot -3 - (6 \cdot 0,5)}{4} = 2$$

A extensão do procedimento de eliminação de Gauss a um sistema com um número  $n$  de equações é simples. O procedimento de eliminação começa com a escolha da primeira linha como linha pivô e continua linha após linha até alcançar a penúltima linha. Em cada passo, a linha pivô é usada para eliminar os termos abaixo do elemento pivô em todas as linhas inferiores. Assim que o sistema original de equações é mudado para a forma triangular superior, a substituição regressiva é usada para determinar a solução.

Quando o método de eliminação de Gauss é programado, é conveniente e mais eficiente criar uma matriz que inclua a matriz de coeficientes  $[a]$  e o vetor  $[b]$  que contém os lados direitos das equações. Isso é feito acrescentando-se o vetor  $[b]$  à matriz  $[a]$ , conforme mostrado no Exemplo 4-2, onde o método de eliminação de Gauss é programado no MATLAB.

### Exemplo 4-2: Função criada no MATLAB para resolver um sistema de equações usando a eliminação de Gauss

Escreva uma função no MATLAB para resolver um sistema de equações lineares,  $[a][x] = [b]$ , usando o método de eliminação de Gauss. Use  $x = \text{Gauss}(a, b)$  como nome de função e argumentos, onde  $a$  é a matriz de coeficientes,  $b$  é o vetor coluna contendo os termos no lado direito das equações, e  $x$  é um vetor coluna contendo a solução.

Use a função Gauss para:

- Resolver o sistema de equações do Exemplo 1-1.
- Resolver o sistema de Eqs. (4.1)

### SOLUÇÃO

A seguinte função resolve um sistema de equações lineares. O programa começa com a incorporação do vetor  $[b]$  à matriz  $[a]$ . A nova matriz aumentada, chamada no programa de  $ab$ , tem a forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix}$$

Em seguida, aplica-se o procedimento de eliminação de Gauss (eliminação progressiva). A matriz é alterada para que todos os elementos abaixo da diagonal principal se anulem:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ 0 & 0 & a_{33} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{nn} & b_n \end{bmatrix}$$

No final do programa, a substituição regressiva é usada para resolver para as incógnitas e os resultados são atribuídos ao vetor coluna  $x$ .

**Programa 4-1 Função definida pelo usuário. Eliminação de Gauss.**

```
function x = Gauss(a,b)
% A função resolve um sistema de equações lineares [a][x]=[b] usando o
% método de eliminação de Gauss.
% Variáveis de entrada:
% a Matriz de coeficientes.
% b Vetor coluna contendo as constantes do lado direito do sistema.
% Variável de saída:
% x Vetor coluna com a solução.
```

```
ab = [a,b];
[R, C] = size(ab);
for j = 1:R - 1
    for i = j + 1:R
        ab(i,j:C) = ab(i,j:C) - ab(i,j)/ab(j,j)*ab(j,j:C);
    end
end
x = zeros(R,1);
x(R) = ab(R,C)/ab(R,R);
for i = R - 1:-1:1
    x(i) = (ab(i,C) - ab(i,i + 1:R)*x(i + 1:R))/ab(i,i);
end
```

A função Gauss é em seguida usada na tela de comandos do MATLAB, inicialmente para resolver o sistema de equações do Exemplo 4-1, e então para resolver o sistema de Eqs. (4.1).

```
>> A=[4 -2 -3 6; -6 7 6.5 -6; 1 7.5 6.25 5.5; -12 22 15.5 -1];
>> B = [12; -6.5; 16; 17];
>> sola = Gauss(A,B)
sola =
    2.0000
    4.0000
   -3.0000
    0.5000
>> C = [9 -4 -2 0; -4 17 -6 -3; -2 -6 14 -6; 0 -3 -6 11];
>> D = [24; -16; 0; 18];
>> solb = Gauss(C,D)
solb =
    4.0343
    1.6545
    2.8452
    3.6395
```

Solução da letra (a).

Solução da letra (b).

#### 4.2.1 Potenciais dificuldades encontradas com a aplicação do método de eliminação de Gauss

##### O elemento pivô é zero

Como a linha pivô é dividida pelo elemento pivô, surge um problema durante a execução do procedimento de eliminação de Gauss se o valor do elemento pivô for igual a zero. Conforme mostrado na seção a seguir, esse problema pode ser corrigido com a mudança da ordem das linhas. Em um processo chamado pivotação, a linha pivô que contém o elemento pivô nulo é trocada por outra linha possuindo um elemento pivô diferente de zero.

##### O elemento pivô é pequeno em relação aos demais termos na linha pivô

Erros de arredondamento significativos podem ocorrer quando o elemento pivô é pequeno em relação aos demais elementos na linha pivô. Isso é ilustrado no próximo exemplo.

Considere o seguinte sistema de equações simultâneas envolvendo as incógnitas  $x_1$  e  $x_2$ :

$$\begin{aligned} 0,0003x_1 + 12,34x_2 &= 12,343 \\ 0,4321x_1 + x_2 &= 5,321 \end{aligned} \quad (4.15)$$

A solução exata do sistema é  $x_1 = 10$  e  $x_2 = 1$ .

O erro de arredondamento é ilustrado com a solução do sistema em um computador com precisão limitada a quatro algarismos significativos. Quando se entra com a primeira das Eqs. (4.15), a constante no lado direito é arredondada para 12,34.

A solução começa com o uso da primeira equação como pivô e  $a_{11} = 0,0003$  como o coeficiente pivô. No primeiro passo, a equação pivô é multiplicada por  $m_{21} = 0,4321/0,0003 = 1440$ . Com quatro algarismos significativos e o arredondamento, essa operação resulta em:

$$(1440)(0,0003x_1 + 12,34x_2) = 1440 \cdot 12,34$$

ou

$$0,4320x_1 + 17770x_2 = 17770$$

O resultado é em seguida subtraído da segunda das Eqs. (4.15):

$$\begin{array}{r} - \\ 0,4321x_1 + x_2 = 5,321 \\ \hline 0,4320x_1 + 17770x_2 = 17770 \\ \hline 0,0001x_1 - 17770x_2 = -17760 \end{array}$$

Após essa operação, o sistema é:

$$0,0003x_1 + 12,34x_2 = 12,34$$

$$0,0001x_1 - 17770x_2 = -17760$$

Note que o elemento  $a_{21}$  é diferente de zero, mas é muito pequeno. Em seguida, o valor de  $x_2$  é calculado a partir da segunda equação:

$$x_2 = \frac{-17760}{-17770} = 0,9994$$

Então,  $x_2$  é substituído na primeira equação, que é resolvida para  $x_1$ :

$$x_1 = \frac{12,34 - (12,34 \cdot 0,9994)}{0,0003} = \frac{12,34 - 12,33}{0,0003} = \frac{0,01}{0,0003} = 33,33$$

A solução obtida para  $x_1$  é obviamente incorreta. Isso ocorre porque a ordem de grandeza de  $a_{11}$  é pequena se comparada à ordem de grandeza de  $a_{12}$ . Conseqüentemente, um erro relativamente pequeno no valor de  $x_2$  (causado pelo arredondamento proveniente da precisão finita do computador) pode levar a um grande erro no valor de  $x_1$ .

O problema pode ser facilmente remediado com a troca da ordem das duas equações na Eq. (4.15):

$$\begin{aligned} 0,4321x_1 + x_2 &= 5,321 \\ 0,0003x_1 + 12,34x_2 &= 12,343 \end{aligned} \quad (4.16)$$

Agora com a primeira equação como pivô, o coeficiente pivô é  $a_{11} = 0,4321$ . No primeiro passo, a equação pivô é multiplicada por  $m_{21} = 0,0003/0,4321 = 0,0006943$ . Com quatro algarismos significativos e o arredondamento, essa operação resulta em:

$$(0,0006943)(0,4321x_1 + x_2 = 5,321) = 0,0006943 \cdot 5,321$$

ou

$$0,0003x_1 + 0,0006943x_2 = 0,003694$$

O resultado é em seguida subtraído da segunda das Eqs. (4.16):

$$\begin{array}{r} 0,0003x_1 + 12,34x_2 = 12,34 \\ - 0,0003x_1 + 0,0006943x_2 = 0,003694 \\ \hline 12,34x_2 = 12,34 \end{array}$$

Após essa operação, o sistema é:

$$\begin{aligned} 0,4321x_1 + x_2 &= 5,321 \\ 0x_1 + 12,34x_2 &= 12,34 \end{aligned}$$

Em seguida, o valor de  $x_2$  é calculado a partir da segunda equação:

$$x_2 = \frac{12,34}{12,34} = 1$$

Depois,  $x_2$  é substituído na primeira equação, que é resolvida para  $x_1$ :

$$x_1 = \frac{5,321 - 1}{0,4321} = 10$$

A solução obtida é agora a solução exata.

Em geral, obtém-se uma solução mais precisa quando as equações são arranjadas (e rearranjadas cada vez que uma nova equação pivô é utilizada) de tal forma que a equação pivô tenha o maior elemento pivô possível. Isso é explicado em maior detalhe na próxima seção.

Erros de arredondamento também podem ser significativos quando grandes sistemas de equações são resolvidos, mesmo quando todos os coeficientes da

linha pivô possuem a mesma ordem de grandeza. Isso pode ser causado pelo grande número de operações (multiplicação, divisão, adição e subtração) associado à solução de grandes sistemas.

### 4.3 ELIMINAÇÃO DE GAUSS COM PIVOTAÇÃO

No procedimento de eliminação de Gauss, a equação pivô é dividida pelo coeficiente pivô. Isto, no entanto, não pode ser feito se o coeficiente pivô for igual a zero. Por exemplo, para o seguinte sistema de três equações:

$$\begin{aligned} 0x_1 + 2x_2 + 3x_3 &= 46 \\ 4x_1 - 3x_2 + 2x_3 &= 16 \\ 2x_1 + 4x_2 - 3x_3 &= 12 \end{aligned}$$

o procedimento tem início com a escolha da primeira equação como equação pivô e do coeficiente de  $x_1$ , que é 0, como coeficiente pivô. Para eliminar o termo  $4x_1$  da segunda equação, a equação pivô deve ser multiplicada por 4/0 e então subtraída da segunda equação. Obviamente, isso não é possível quando o elemento pivô é igual a zero. A divisão por zero pode ser evitada se a ordem na qual as equações são escritas for alterada de tal forma que, na primeira equação, o primeiro coeficiente seja diferente de zero. Por exemplo, isso pode ser feito no sistema visto com o intercâmbio das duas primeiras equações.

Após o primeiro passo, a segunda equação tem um elemento pivô igual a zero.

$$\left[ \begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[ \begin{array}{c} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{array} \right]$$

Usando a pivotação, troca-se a segunda pela quarta equação.

$$\left[ \begin{array}{cccc} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & 0 & a'_{23} & a'_{24} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \right] = \left[ \begin{array}{c} b_1 \\ b'_3 \\ b'_2 \\ b'_4 \end{array} \right]$$

**Figura 4-16** Ilustração da pivotação.

No procedimento geral de eliminação de Gauss, uma equação (ou linha) pode ser usada como a equação pivô (linha pivô) apenas se o coeficiente pivô (elemento pivô) for diferente de zero. Se o elemento pivô for nulo, a equação (isto é, a linha) deve ser trocada por uma das demais equações (linhas) que possuir um coeficiente pivô diferente de zero. Essa troca de linhas, ilustrada na Fig. 4-16, é chamada de **pivotação**.

#### Comentários adicionais sobre a pivotação

- Se durante o procedimento de eliminação de Gauss uma equação pivô tiver um elemento pivô nulo e o sistema de equações sendo resolvido tiver solução, uma equação com um elemento pivô diferente de zero sempre poderá ser encontrada.
- Os cálculos numéricos são menos propensos a erros e apresentam menores erros de arredondamento (ver Seção 4.2.1) se o elemento pivô possuir um valor numérico absoluto grande em comparação com os demais elementos na mesma linha. Conseqüentemente, entre todas as equações que podem ser trocadas para assumir a função de equação pivô, é melhor selecionar a equação cujo elemento pivô possui o maior valor numérico absoluto. Além disso, é sempre bom empregar a pivotação para se ter um elemento pivô com o maior valor possível (mesmo quando a pivotação não for necessária).

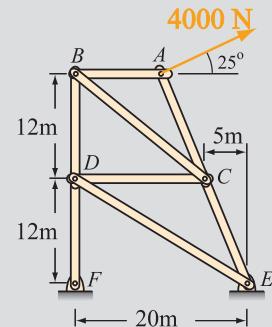
**Exemplo 4-3: Função criada no MATLAB para resolver um sistema de equações usando a eliminação de Gauss com pivotação**

Escreva uma função no MATLAB que resolva um sistema de equações lineares,  $[a][x] = [b]$ , usando o método de eliminação de Gauss com pivotação. Chame a função de  $x=GaussPivot(a,b)$ , onde  $a$  é a matriz de coeficientes,  $b$  é o vetor coluna contendo os termos no lado direito das equações, e  $x$  é um vetor coluna contendo a solução. Use essa função para determinar as forças na treliça com oito membros mostrada na figura (que é igual à Fig. 4-2).

### SOLUÇÃO

As forças nos oito membros da treliça são determinadas a partir do conjunto de oito equações descrito pela Eq. (4.2). As equações são deduzidas a partir do traçado dos diagramas de corpo livre dos pinos  $A, B, C$  e  $D$ , e a aplicação das equações de equilíbrio. As equações são rescritas aqui na forma matricial (intencionalmente, as equações são escritas em uma ordem que requer a pivotação):

$$\begin{bmatrix} 0 & 0,9231 & 0 & 0 & 0 & 0 & 0 \\ -1 & -0,3846 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0,8575 \\ 1 & 0 & -0,7809 & 0 & 0 & 0 & 0 \\ 0 & -0,3846 & -0,7809 & 0 & -1 & 0,3846 & 0 \\ 0 & 0,9231 & 0,6247 & 0 & 0 & -0,9231 & 0 \\ 0 & 0 & 0,6247 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -0,5145 \end{bmatrix} \begin{bmatrix} F_{AB} \\ F_{AC} \\ F_{BC} \\ F_{BD} \\ F_{CD} \\ F_{CE} \\ F_{DE} \\ F_{DF} \end{bmatrix} = \begin{bmatrix} 1690 \\ 3625 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.17)$$



A função `GaussPivot` é criada com a modificação da função `Gauss` listada na solução do Exemplo 4-2.

**Programa 4-2 Função definida pelo usuário. Eliminação de Gauss com pivotação.**

```
function x = GaussPivot(a,b)
% A função resolve um sistema de equações lineares [a][x]=[b] usando o
% método de eliminação de Gauss com pivotação.
% Variáveis de entrada:
% a Matriz de coeficientes.
% b Vetor coluna contendo as constantes do lado direito do sistema.
% Variável de saída:
% x Vetor coluna com a solução.
```

```
ab = [a,b];
[R, C] = size(ab);
for j = 1:R - 1
    % Começa a seção de pivotação
    if ab(j,j) == 0
```

Verifica se o elemento pivô é nulo.

```

for k = j + 1:R
    if ab(k,j) ~ = 0
        abTemp = ab(j,:);
        ab(j,:) = ab(k,:);
        ab(k,:) = abTemp;
        break
    end
end
end
% Termina a seção de pivotação
for i = j + 1:R
    ab(i,j:C) = ab(i,j:C) - ab(i,j)/ab(j,j)*ab(j,j:C);
end
x = zeros(R,1);
x(R) = ab(R,C)/ab(R,R);
for i = R - 1:-1:1
    x(i) = (ab(i,C) - ab(i,i + 1:R)*x(i + 1:R))/ab(i,i);
end

```

Se a pivotação for necessária, procura-se uma linha contendo um elemento pivô diferente de zero.

Troca a linha que tem o elemento pivô nulo pela linha que tem o elemento pivô não-nulo.

Termina a busca por uma linha com elemento pivô diferente de zero.

A função GaussPivot é a seguir usada em um programa para resolver o sistema de equações da Eq. (4.17).

% Exemplo 4-3

```

a = [0 0.9231 0 0 0 0 0; -1 -0.3846 0 0 0 0 0; 0 0 0 0 1 0 0.8575 0; 1 0 -0.7809 0 0 0 0
     0 -0.3846 -0.7809 0 -1 0.3846 0 0; 0 0.9231 0.6247 0 0 -0.9231 0 0
     0 0 0.6247 -1 0 0 0 0; 0 0 0 1 0 0 -0.5145 -1];
b = [1690;3625;0;0;0;0;0;0];
Forcas = GaussPivot(a,b)

```

Quando se executa o programa, a seguinte solução é exibida na janela de comandos:

```

Forcas =
-4.3291e+003
1.8308e+003
-5.5438e+003
-3.4632e+003
2.8862e+003
-1.9209e+003
-3.3659e+003
-1.7315e+003
>>

```

$$\begin{bmatrix} F_{AB} \\ F_{AC} \\ F_{BC} \\ F_{BD} \\ F_{CD} \\ F_{CE} \\ F_{DE} \\ F_{DF} \end{bmatrix}$$

A inclusão da pivotação na programação do método de eliminação de Gauss é mostrada no Exemplo 4-3. A inclusão da pivotação com o intuito de selecionar como equação pivô aquela que possui o maior elemento pivô possível é deixada como exercício no Problema 4.23.

## 4.4 MÉTODO DE ELIMINAÇÃO DE GAUSS-JORDAN

O método de eliminação de Gauss-Jordan é um procedimento utilizado para resolver sistemas de equações lineares na forma  $[a][x] = [b]$ . Nesse procedimento, um sistema de equações fornecido na forma geral é manipulado até que se obtenha um sistema equivalente na forma diagonal (ver Seção 4.1.1), com elementos normalizados. Isso significa que, quando a forma diagonal da matriz de coeficientes  $[a]$  é reduzida à matriz identidade, o novo vetor  $[b']$  é a solução. O ponto de partida do procedimento é um sistema de equações genérico (a ilustração a seguir se refere a um sistema com quatro equações):

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

**Figura 4-17** Forma matricial de um sistema de quatro equações.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 & (4.18a) \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 & (4.18b) \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 & (4.18c) \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4 & (4.18d) \end{aligned}$$

A forma matricial do sistema é mostrada na Fig. 4-17. No método de eliminação de Gauss-Jordan, o sistema de equações é manipulado até apresentar a forma diagonal a seguir:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

**Figura 4-18** Forma matricial do sistema equivalente após a aplicação do método de Gauss-Jordan.

$$\begin{aligned} x_1 + 0 + 0 + 0 &= b'_1 & (4.19a) \\ 0 + x_2 + 0 + 0 &= b'_2 & (4.19b) \\ 0 + 0 + x_3 + 0 &= b'_3 & (4.19c) \\ 0 + 0 + 0 + x_4 &= b'_4 & (4.19d) \end{aligned}$$

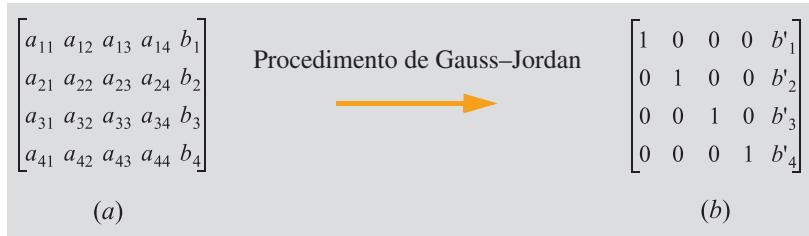
A forma matricial do sistema equivalente é mostrada na Fig. 4-18. Os termos no lado direito das equações (coluna  $[b']$ ) são a solução. Na forma matricial, a matriz dos coeficientes é transformada em uma matriz identidade.

### Procedimento de eliminação de Gauss-Jordan

O procedimento de eliminação de Gauss-Jordan utilizado para transformar o sistema de Equações (4.18) para a forma apresentada nas Eqs. (4.19) é idêntico ao procedimento de eliminação de Gauss, exceto pelas duas diferenças a seguir:

- A equação pivô é normalizada com a divisão de todos os seus termos pelo coeficiente pivô. Isso faz com que o coeficiente pivô seja igual a 1.
- A equação pivô é utilizada na eliminação dos elementos fora da diagonal principal em **TODAS** as demais equações. Isso significa que o processo de eliminação é aplicado às equações (linhas) que estão acima e abaixo da equação pivô (no método de eliminação de Gauss, apenas elementos abaixo do elemento pivô são eliminados).

Quando se programa o procedimento de Gauss-Jordan, é conveniente e mais eficiente criar uma única matriz que inclua a matriz de coeficientes  $[a]$  e o vetor  $[b]$ . Isso é feito com a incorporação do vetor  $[b]$  à matriz  $[a]$ . A matriz aumentada é o ponto de partida de um procedimento mostrado na Fig. 4-19a (para um sistema de quatro equações). No final do procedimento, mostrado na Fig. 4-19b, os elementos de  $[a]$  são substituídos por uma matriz identidade, e a coluna  $[b']$  é a solução.

**Figura 4-19** Ilustração esquemática do método de Gauss-Jordan.

O método de Gauss-Jordan também pode ser usado para resolver vários sistemas de equações  $[a][x] = [b]$  que tenham os mesmos coeficientes  $[a]$  mas diferentes vetores  $[b]$ . Isso é feito aumentando a matriz  $[a]$  de forma a incluir todos os vetores  $[b]$  desejados. Na Seção 4.6.2, o método de Gauss-Jordan é usado dessa forma para calcular a inversa de uma matriz.

O método de eliminação de Gauss-Jordan é demonstrado no Exemplo 4-4, que ilustra a sua aplicação na solução do conjunto de equações já resolvido no Exemplo 4-1.

#### Exemplo 4-4: Solução de um conjunto de quatro equações usando a eliminação de Gauss-Jordan

Resolva o seguinte conjunto de quatro equações utilizando o método de eliminação de Gauss-Jordan.

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ -6x_1 + 7x_2 + 6,5x_3 - 6x_4 &= -6,5 \\ x_1 + 7,5x_2 + 6,25x_3 + 5,5x_4 &= 16 \\ -12x_1 + 22x_2 + 15,5x_3 - x_4 &= 17 \end{aligned}$$

#### SOLUÇÃO

A solução é realizada utilizando a forma matricial das equações. Na forma matricial, o sistema é:

$$\left[ \begin{array}{cccc|c} 4 & -2 & -3 & 6 & 12 \\ -6 & 7 & 6,5 & -6 & -6,5 \\ 1 & 7,5 & 6,25 & 5,5 & 16 \\ -12 & 22 & 15,5 & -1 & 17 \end{array} \right]$$

No procedimento numérico, uma nova matriz é criada com a inclusão do lado direito da equação na matriz de coeficientes:

$$\left[ \begin{array}{cccc|c} 4 & -2 & -3 & 6 & 12 \\ -6 & 7 & 6,5 & -6 & -6,5 \\ 1 & 7,5 & 6,25 & 5,5 & 16 \\ -12 & 22 & 15,5 & -1 & 17 \end{array} \right]$$

A primeira linha de pivotação é a primeira linha, e o primeiro elemento da linha é o elemento pivô. A linha é normalizada com a sua divisão pelo elemento pivô.

$$\left[ \begin{array}{ccccc} \frac{4}{4} & -\frac{2}{4} & -\frac{3}{4} & \frac{6}{4} & \frac{12}{4} \\ -6 & 7 & 6,5 & -6 & -6,5 \\ 1 & 7,5 & 6,25 & 5,5 & 16 \\ -12 & 22 & 15,5 & -1 & 17 \end{array} \right] = \left[ \begin{array}{ccccc} 1 & -0,5 & -0,75 & 1,5 & 3 \\ -6 & 7 & 6,5 & -6 & -6,5 \\ 1 & 7,5 & 6,25 & 5,5 & 16 \\ -12 & 22 & 15,5 & -1 & 17 \end{array} \right]$$

Em seguida, todos os primeiros elementos nas linhas 2, 3 e 4 são eliminados:

$$\left[ \begin{array}{ccccc} 1 & -0,5 & -0,75 & 1,5 & 3 \\ -6 & 7 & 6,5 & -6 & -6,5 \\ 1 & 7,5 & 6,25 & 5,5 & 16 \\ -12 & 22 & 15,5 & -1 & 17 \end{array} \right] \xrightarrow{\begin{array}{l} -(6)[1 \ -0,5 \ -0,75 \ 1,5 \ 3] \\ -(1)[1 \ -0,5 \ -0,75 \ 1,5 \ 3] \\ -(-12)[1 \ -0,5 \ -0,75 \ 1,5 \ 3] \end{array}} = \left[ \begin{array}{ccccc} 1 & -0,5 & -0,75 & 1,5 & 3 \\ 0 & 4 & 2 & 3 & 11,5 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6,5 & 17 & 53 \end{array} \right]$$

A próxima linha pivô é a segunda linha, e seu segundo elemento é utilizado como pivô. A linha é normalizada com a sua divisão pelo elemento pivô:

$$\left[ \begin{array}{ccccc} 1 & -0,5 & -0,75 & 1,5 & 3 \\ 0 & \frac{4}{4} & \frac{2}{4} & \frac{3}{4} & \frac{11,5}{4} \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6,5 & 17 & 53 \end{array} \right] = \left[ \begin{array}{ccccc} 1 & -0,5 & -0,75 & 1,5 & 3 \\ 0 & 1 & 0,5 & 0,75 & 2,875 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6,5 & 17 & 53 \end{array} \right]$$

Em seguida, todos os segundos elementos nas linhas 1, 3 e 4 são eliminados:

$$\left[ \begin{array}{ccccc} 1 & -0,5 & -0,75 & 1,5 & 3 \\ 0 & 1 & 0,5 & 0,75 & 2,875 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6,5 & 17 & 53 \end{array} \right] \xrightarrow{\begin{array}{l} -(0,5)[0 \ 1 \ 0,5 \ 0,75 \ 2,875] \\ -(8)[0 \ 1 \ 0,5 \ 0,75 \ 2,875] \\ -(-16)[0 \ 1 \ 0,5 \ 0,75 \ 2,875] \end{array}} = \left[ \begin{array}{ccccc} 1 & 0 & -0,5 & 1,875 & 4,4375 \\ 0 & 1 & 0,5 & 0,75 & 2,875 \\ 0 & 0 & 3 & -2 & -10 \\ 0 & 0 & -1,5 & 5 & 7 \end{array} \right]$$

A próxima linha pivô é a terceira linha, e seu terceiro elemento é usado como pivô. A linha é normalizada com a sua divisão pelo elemento pivô:

$$\left[ \begin{array}{ccccc} 1 & 0 & -0,5 & 1,875 & 4,4375 \\ 0 & 1 & 0,5 & 0,75 & 2,875 \\ 0 & 0 & \frac{3}{3} & -\frac{2}{3} & \frac{-10}{3} \\ 0 & 0 & -1,5 & 5 & 7 \end{array} \right] = \left[ \begin{array}{ccccc} 1 & 0 & -0,5 & 1,875 & 4,4375 \\ 0 & 1 & 0,5 & 0,75 & 2,875 \\ 0 & 0 & 1 & -0,667 & -3,333 \\ 0 & 0 & -1,5 & 5 & 7 \end{array} \right]$$

Em seguida, todos os terceiros elementos nas linhas 1, 2 e 4 são eliminados:

$$\left[ \begin{array}{ccccc} 1 & 0 & -0,5 & 1,875 & 4,4375 \\ 0 & 1 & 0,5 & 0,75 & 2,875 \\ 0 & 0 & 1 & -0,667 & -3,333 \\ 0 & 0 & -1,5 & 5 & 7 \end{array} \right] \xrightarrow{\begin{array}{l} -(0,5)[0 \ 0 \ 1 \ -0,667 \ -3,333] \\ -(0,5)[0 \ 0 \ 1 \ -0,667 \ -3,333] \\ -(-1,5)[0 \ 0 \ 1 \ -0,667 \ -3,333] \end{array}} = \left[ \begin{array}{ccccc} 1 & 0 & 0 & 1,5417 & 2,7708 \\ 0 & 1 & 0 & 1,0833 & 4,5417 \\ 0 & 0 & 1 & -0,667 & -3,333 \\ 0 & 0 & 0 & 4 & 2 \end{array} \right]$$

A próxima linha pivô é a quarta linha, sendo o seu quarto elemento usado como pivô. A linha é normalizada com a sua divisão pelo elemento pivô:

$$\left[ \begin{array}{ccccc} 1 & 0 & 0 & 1,5417 & 2,7708 \\ 0 & 1 & 0 & 1,0833 & 4,5417 \\ 0 & 0 & 1 & -0,667 & -3,333 \\ 0 & 0 & 0 & \frac{4}{4} & \frac{2}{4} \end{array} \right] = \left[ \begin{array}{ccccc} 1 & 0 & 0 & 1,5417 & 2,7708 \\ 0 & 1 & 0 & 1,0833 & 4,5417 \\ 0 & 0 & 1 & -0,667 & -3,333 \\ 0 & 0 & 0 & 1 & 0,5 \end{array} \right]$$

Em seguida, todos os quartos elementos nas linhas 1, 2 e 3 são eliminados

$$\left[ \begin{array}{ccccc} 1 & 0 & 0 & 1,5417 & 2,7708 \\ 0 & 1 & 0 & 1,0833 & 4,5417 \\ 0 & 0 & 1 & -0,667 & -3,333 \\ 0 & 0 & 0 & 1 & 0,5 \end{array} \right] \xrightarrow{-(1,5417)[0\ 0\ 0\ 1\ 0,5]} \left[ \begin{array}{ccccc} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & -3 \\ 0 & 0 & 0 & 1 & 0,5 \end{array} \right]$$

A solução é:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ -3 \\ 0,5 \end{bmatrix}$$

### Método de eliminação de Gauss-Jordan com pivotação

Pode ser que as equações estejam escritas em uma ordem tal que, durante o procedimento de eliminação, uma equação pivô tenha um elemento pivô igual a zero. Obviamente, nesse caso é impossível normalizar a linha pivô (dividi-la pelo elemento pivô). Da mesma forma que no método de eliminação de Gauss, o problema pode ser corrigido usando a pivotação. Isso é deixado como exercício no Problema 4.24.

## 4.5 MÉTODO DE DECOMPOSIÇÃO LU

### Fundamentos

O método de eliminação de Gauss consiste em duas partes. A primeira corresponde ao procedimento de eliminação no qual um sistema de equações lineares dado na forma geral,  $[a][x] = [b]$ , é transformado em um sistema equivalente de equações  $[a'][x] = [b']$  no qual a matriz de coeficientes  $[a']$  é triangular superior. Na segunda parte, o sistema equivalente é solucionado com o emprego da substituição regressiva. O procedimento de eliminação requer operações matemáticas e também um tempo computacional muito mais significativos do que os cálculos realizados na substituição regressiva. Durante o procedimento de eliminação, a matriz de coeficientes  $[a]$  e o vetor  $[b]$  são alterados. Isso significa que, se houver a necessidade de resolver sistemas de equações possuindo os mesmos termos no lado esquerdo (uma mesma matriz de coeficientes  $[a]$ ), mas diferentes constantes do lado direito (diferentes vetores  $[b]$ ), o procedimento de eliminação deve ser feito novamente para cada  $[b]$ . Idealmente, seria melhor se as operações realizadas na matriz de coeficientes  $[a]$  fossem dissociadas daquelas realizadas no vetor

de constantes  $[b]$ . Dessa forma, o procedimento de eliminação em  $[a]$  seria feito apenas uma vez e então utilizado para solucionar sistemas de equações com diferentes vetores  $[b]$ .

Uma opção para resolver sistemas de equações  $[a][x] = [b]$  com mesmas matrizes de coeficientes  $[a]$  mas diferentes vetores de constantes  $[b]$  é primeiro calcular a inversa da matriz  $[a]$ . Uma vez conhecida a matriz inversa  $[a]^{-1}$ , a solução pode ser calculada com:

$$[x] = [a]^{-1}[b]$$

O cálculo da inversa de uma matriz, contudo, requer muitas operações matemáticas e é computacionalmente ineficiente. Um método de solução mais eficiente para este caso é o método de decomposição *LU*.

No método de decomposição *LU*, as operações com a matriz  $[a]$  são feitas sem que o vetor  $[b]$  seja utilizado ou alterado. Este só é usado na parte de substituição da solução. O método de decomposição *LU* pode ser usado para resolver um único sistema de equações lineares. Porém, ele é especialmente vantajoso na solução de sistemas que têm as mesmas matrizes de coeficientes  $[a]$  mas diferentes vetores de constantes  $[b]$ .

### O método de decomposição *LU*

O método de decomposição *LU* é um método empregado na solução de sistemas de equações lineares na forma  $[a][x] = [b]$ . Neste método, a matriz de coeficientes  $[a]$  é decomposta (fatorada) em um produto de duas matrizes  $[L]$  e  $[U]$ :

$$[a] = [L][U] \quad (4.20)$$

onde a matriz  $[L]$  é uma matriz triangular inferior e a matriz  $[U]$  é uma matriz triangular superior. Com essa decomposição, o sistema de equações a ser resolvido tem a forma:

$$[L][U][x] = [b] \quad (4.21)$$

Para resolver essa equação, o produto  $[U][x]$  é definido como

$$[U][x] = [y] \quad (4.22)$$

e substituído na Eq. (4.21), levando a:

$$[L][y] = [b] \quad (4.23)$$

A solução  $[x]$  é agora obtida em dois passos. Primeiramente, a Eq. (4.23) é resolvida para  $[y]$ . Em seguida, a solução  $[y]$  é substituída na Eq. (4.22) e esta equação é resolvida para  $[x]$ .

Como a matriz  $[L]$  é uma matriz triangular inferior, a solução  $[y]$  na Eq. (4.23) é obtida com o uso do método da substituição progressiva. Assim que  $[y]$  é conhecido e substituído na Eq. (4.22), esta equação é resolvida com o emprego da substituição regressiva, pois  $[U]$  é uma matriz triangular superior.

Para uma dada matriz  $[a]$ , vários métodos podem ser usados para determinar as matrizes correspondentes  $[L]$  e  $[U]$ . Dois desses métodos, um relacionado ao método de eliminação de Gauss e o outro chamado de método de Crout, são descritos a seguir.

#### 4.5.1 Decomposição LU usando o procedimento de eliminação de Gauss

Quando o procedimento de eliminação de Gauss é aplicado em uma matriz  $[a]$ , os elementos das matrizes  $[L]$  e  $[U]$  já são calculados automaticamente. A matriz triangular superior  $[U]$  é a matriz de coeficientes  $[a]$  obtida ao final do procedimento, conforme mostrado nas Figs. 4-8 e 4-14. A matriz triangular inferior  $[L]$  não é escrita explicitamente durante o procedimento, mas os elementos que a formam são na realidade calculados ao longo do caminho. Os elementos diagonais de  $[L]$  são todos iguais a 1. Os elementos abaixo da diagonal são os multiplicadores  $m_{ij}$  que multiplicam a equação pivô quando ela é usada para eliminar os elementos abaixo do coeficiente pivô (ver o *procedimento de eliminação de Gauss* na Seção 4.2). No caso de um sistema com quatro equações, a matriz de coeficientes  $[a]$  é  $(4 \times 4)$ , e a decomposição tem a forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{bmatrix} \quad (4.24)$$

Um exemplo numérico ilustrando a decomposição  $LU$  é dado a seguir. Ele usa as informações referentes à solução do Exemplo 4-1, em que um sistema de quatro equações é resolvido com o uso do método de eliminação de Gauss. A matriz  $[a]$  é escrita a partir do conjunto original de equações fornecido no enunciado do programa, e a matriz  $[U]$  é escrita a partir do conjunto de equações no final do **Passo 3** (p. 121). A matriz  $[L]$  é escrita a partir dos multiplicadores calculados durante a solução. A decomposição tem a forma:

$$\begin{bmatrix} 4 & -2 & -3 & 6 \\ -6 & 7 & 6,5 & -6 \\ 1 & 7,5 & 6,25 & 5,5 \\ -12 & 22 & 15,5 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1,5 & 1 & 0 & 0 \\ 0,25 & 2 & 1 & 0 \\ -3 & 4 & -0,5 & 1 \end{bmatrix} \begin{bmatrix} 4 & -2 & -3 & 6 \\ 0 & 4 & 2 & 3 \\ 0 & 0 & 3 & -2 \\ 0 & 0 & 0 & 4 \end{bmatrix} \quad (4.25)$$

A decomposição na Eq. (4.25) pode ser verificada no MATLAB:

```
>> L = [1,0,0,0;-1.5,1,0,0;0.25,2,1,0;-3,4,-0.5,1]
L =
    1.0000      0      0      0
   -1.5000    1.0000      0      0
    0.2500    2.0000    1.0000      0
   -3.0000    4.0000   -0.5000    1.0000
>> U = [4,-2,-3,6;0,4,2,3;0,0,3,-2;0,0,0,4]
```

```

U =
 4 -2 -3 6
 0 4 2 3
 0 0 3 -2
 0 0 0 4
>> L*U
ans =
 4.0000 -2.0000 -3.0000 6.0000
 -6.0000 7.0000 6.5000 -6.0000
 1.0000 7.5000 6.2500 5.5000
 -12.0000 22.0000 15.5000 -1.0000

```

A multiplicação das matrizes  $L$  e  $U$  verifica se a resposta é a matriz  $[a]$ .

### 4.5.2 Decomposição LU usando o método de Crout

Neste método, decompõe-se a matriz  $[a]$  no produto  $[L][U]$ , no qual os elementos diagonais da matriz  $[U]$  são todos iguais a 1. Os elementos de ambas as matrizes podem ser determinados com o uso de fórmulas de fácil programação. Isso é ilustrado a seguir para um sistema de quatro equações. No método de Crout, a decomposição  $LU$  tem a forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \begin{bmatrix} 1 & U_{12} & U_{13} & U_{14} \\ 0 & 1 & U_{23} & U_{24} \\ 0 & 0 & 1 & U_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.26)$$

A execução da multiplicação matricial no lado direito da equação resulta em:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} L_{11} & (L_{11}U_{12}) & (L_{11}U_{13}) & (L_{11}U_{14}) \\ L_{21} & (L_{21}U_{12}+L_{22}) & (L_{21}U_{13}+L_{22}U_{23}) & (L_{21}U_{14}+L_{22}U_{24}) \\ L_{31} & (L_{31}U_{12}+L_{32}) & (L_{31}U_{13}+L_{32}U_{23}+L_{33}) & (L_{31}U_{14}+L_{32}U_{24}+L_{33}U_{34}) \\ L_{41} & (L_{41}U_{12}+L_{42}) & (L_{41}U_{13}+L_{42}U_{23}+L_{43}) & (L_{41}U_{14}+L_{42}U_{24}+L_{43}U_{34}+L_{44}) \end{bmatrix} \quad (4.27)$$

Os elementos das matrizes  $[L]$  e  $[U]$  podem ser determinados com a solução da Eq (4.27). Para isso, basta igualar os elementos correspondentes em ambos os lados da equação. É possível calcular os elementos das matrizes  $[L]$  e  $[U]$  com facilidade, linha após linha, a partir dos elementos conhecidos de  $[a]$  e dos elementos de  $[L]$  e  $[U]$  já calculados. Começando com a primeira linha, o valor de  $L_{11}$  é dado por  $L_{11} = a_{11}$ . Uma vez conhecido  $L_{11}$ , os valores de  $U_{12}$ ,  $U_{13}$  e  $U_{14}$  são calculados por:

$$U_{12} = \frac{a_{12}}{L_{11}} \quad U_{13} = \frac{a_{13}}{L_{11}} \quad \text{e} \quad U_{14} = \frac{a_{14}}{L_{11}} \quad (4.28)$$

Movendo para a segunda linha, o valor de  $L_{21}$  é calculado a partir da igualdade  $L_{21} = a_{21}$  e o valor de  $L_{22}$  é calculado de:

$$L_{22} = a_{22} - L_{21}U_{12} \quad (4.29)$$

Com os valores de  $L_{21}$  e  $L_{22}$  conhecidos, os valores de  $U_{23}$  e  $U_{24}$  são determinados de:

$$U_{23} = \frac{a_{23} - L_{21}U_{13}}{L_{22}} \quad \text{e} \quad U_{24} = \frac{a_{24} - L_{21}U_{14}}{L_{22}} \quad (4.30)$$

Na terceira linha:

$$L_{31} = a_{31}, \quad L_{32} = a_{32} - L_{31}U_{12} \quad \text{e} \quad L_{33} = a_{33} - L_{31}U_{13} - L_{32}U_{23} \quad (4.31)$$

Uma vez conhecidos os valores de  $L_{31}$ ,  $L_{32}$  e  $L_{33}$ , o valor de  $U_{34}$  é calculado de:

$$U_{34} = \frac{a_{34} - L_{31}U_{14} - L_{32}U_{24}}{L_{33}} \quad (4.32)$$

Na quarta coluna, os valores de  $L_{41}$ ,  $L_{42}$ ,  $L_{43}$  e  $L_{44}$  são calculados de:

$$\begin{aligned} L_{41} &= a_{41}, \quad L_{42} = a_{42} - L_{41}U_{12}, \quad L_{43} = a_{43} - L_{41}U_{13} - L_{42}U_{23} \quad \text{e} \\ L_{44} &= a_{44} - L_{41}U_{14} - L_{42}U_{24} - L_{43}U_{34} \end{aligned} \quad (4.33)$$

Um procedimento para determinar os elementos das matrizes  $[L]$  e  $[U]$  pode ser escrito com base nos cálculos apresentados nas Eqs. (4.28) a (4.33). Se  $[a]$  é uma matriz  $(n \times n)$ , os elementos de  $[L]$  e  $[U]$  são calculados da seguinte maneira:

**Passo 1** Cálculo da primeira coluna de  $[L]$ :

$$\text{para } i = 1, 2, \dots, n \quad L_{i1} = a_{i1} \quad (4.34)$$

**Passo 2** Substituição de (1s) na diagonal de  $[U]$ :

$$\text{para } i = 1, 2, \dots, n \quad U_{ii} = 1 \quad (4.35)$$

**Passo 3** Cálculo dos elementos da primeira linha de  $[U]$  (exceto  $U_{11}$ , que já foi calculado):

$$\text{para } j = 2, 3, \dots, n \quad U_{1j} = \frac{a_{1j}}{L_{11}} \quad (4.36)$$

**Passo 4** Cálculo dos demais elementos linha após linha ( $i$  é o número da linha e  $j$  é o número da coluna). Os elementos de  $[L]$  são calculados primeiro porque eles são usados no cálculo dos elementos de  $[U]$ :

$$\text{para } i = 2, 3, \dots, n$$

$$\text{para } j = 2, 3, \dots, i \quad L_{ij} = a_{ij} - \sum_{k=1}^{j-1} L_{ik}U_{kj} \quad (4.37)$$

$$\text{para } j = (i+1), (i+2), \dots, n$$

$$U_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} L_{ik}U_{kj}}{L_{ii}} \quad (4.38)$$

Os Exemplos 4-5 e 4-6 mostram como empregar a decomposição  $LU$  com o método de Crout na solução de sistemas de equações. No Exemplo 4-5, os cálculos são feitos manualmente, e, no Exemplo 4-6, a decomposição é feita usando um programa no MATLAB.

**Exemplo 4-5: Resolvendo um conjunto de quatro equações usando a decomposição LU com o método de Crout**

Resolva o seguinte conjunto de quatro equações (o mesmo do Exemplo 4-1) usando a decomposição LU com o método de Crout.

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ -6x_1 + 7x_2 + 6,5x_3 - 6x_4 &= -6,5 \\ x_1 + 7,5x_2 + 6,25x_3 + 5,5x_4 &= 16 \\ -12x_1 + 22x_2 + 15,5x_3 - x_4 &= 17 \end{aligned}$$

## SOLUÇÃO

Primeiramente, as equações são escritas na forma matricial:

$$\left[ \begin{array}{cccc} 4 & -2 & -3 & 6 \\ -6 & 7 & 6,5 & -6 \\ 1 & 7,5 & 6,25 & 5,5 \\ -12 & 22 & 15,5 & -1 \end{array} \right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ -6,5 \\ 16 \\ 17 \end{bmatrix} \quad (4.39)$$

Em seguida, a matriz de coeficientes  $[a]$  é decomposta no produto  $[L][U]$  como mostra a Eq. (4.27). A decomposição é feita seguindo os passos listados na página anterior:

**Passo 1** Cálculo da primeira coluna de  $[L]$ :

para  $i = 1, 2, 3, 4$      $L_{i1} = a_{ii}$ :     $L_{11} = 4$ ,     $L_{21} = -6$ ,     $L_{31} = 1$ ,     $L_{41} = -12$

**Passo 2** Substituição de (1s) na diagonal de  $[U]$ :

para  $i = 1, 2, 3, 4$      $U_{ii} = 1$ :     $U_{11} = 1$ ,     $U_{22} = 1$ ,     $U_{33} = 1$ ,     $U_{44} = 1$ .

**Passo 3** Cálculo dos elementos da primeira linha de  $[U]$  (exceto  $U_{11}$ , que já foi calculado):

para  $j = 2, 3, 4$      $U_{1j} = \frac{a_{1j}}{L_{11}}$ :     $U_{12} = \frac{a_{12}}{L_{11}} = \frac{-2}{4} = -0,5$ ,     $U_{13} = \frac{a_{13}}{L_{11}} = \frac{-3}{4} = -0,75$ ,  
 $U_{14} = \frac{a_{14}}{L_{11}} = \frac{6}{4} = 1,5$

**Passo 4** Cálculo dos demais elementos linha apóis linha ( $i$  é o número da linha e  $j$  é o número da coluna). No presente problema, há quatro linhas, e com isso  $i$  começa em 2 e termina em 4. Para cada valor de  $i$  (cada linha), os elementos de  $[L]$  são calculados primeiro e os elementos de  $[U]$  são calculados em seguida. A forma geral das equações é (Eqs. (4.37) e (4.38)):

para  $i = 2, 3, 4$

$$\text{para } j = 2, 3, \dots, i \quad L_{ij} = a_{ij} - \sum_{k=1}^{j-1} L_{ik} U_{kj} \quad (4.40)$$

$$\text{para } j = (i+1), (i+2), \dots, n \quad U_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} L_{ik} U_{kj}}{L_{ii}} \quad (4.41)$$

Começando com a segunda linha,  $i = 2$ :

$$\text{para } j = 2: \quad L_{22} = a_{22} - \sum_{k=1}^{1} L_{2k} U_{k2} = a_{22} - L_{21} U_{12} = 7 - (-6 \cdot -0,5) = 4$$

para  $j = 3, 4$  :

$$U_{23} = \frac{a_{23} - \sum_{k=1}^{k=1} L_{2k} U_{k3}}{L_{22}} = \frac{a_{23} - (L_{21} U_{13})}{L_{22}} = \frac{6,5 - (-6 \cdot -0,75)}{4} = 0,5$$

$$U_{24} = \frac{a_{24} - \sum_{k=1}^{k=1} L_{2k} U_{k4}}{L_{22}} = \frac{a_{24} - (L_{21} U_{14})}{L_{22}} = \frac{-6 - (-6 \cdot 1,5)}{4} = 0,75$$

Em seguida, para a terceira linha,  $i = 3$ :

para  $j = 2, 3$  :

$$L_{32} = a_{32} - \sum_{k=1}^{k=1} L_{3k} U_{k2} = a_{32} - L_{31} U_{12} = 7,5 - (1 \cdot -0,5) = 8$$

$$L_{33} = a_{33} - \sum_{k=1}^{k=2} L_{3k} U_{k3} = a_{33} - (L_{31} U_{13} + L_{32} U_{23}) = 6,25 - (1 \cdot -0,75 + 8 \cdot 0,5) = 3$$

para

$$j = 4 : U_{34} = \frac{a_{34} - \sum_{k=1}^{k=2} L_{3k} U_{k4}}{L_{33}} = \frac{a_{34} - (L_{31} U_{14} + L_{32} U_{24})}{L_{33}} = \frac{5,5 - (1 \cdot 1,5 + 8 \cdot 0,75)}{3} = -0,6667$$

Para a última linha,  $i = 4$

para

$$j = 2, 3, 4 : L_{42} = a_{42} - \sum_{k=1}^{k=1} L_{4k} U_{k2} = a_{42} - L_{41} U_{12} = 22 - (-12 \cdot -0,5) = 16$$

$$L_{43} = a_{43} - \sum_{k=1}^{k=2} L_{4k} U_{k3} = a_{43} - (L_{41} U_{13} + L_{42} U_{23}) = 15,5 - (-12 \cdot -0,75 + 16 \cdot 0,5) = -1,5$$

$$L_{44} = a_{44} - \sum_{k=1}^{k=4} L_{4k} U_{k4} = a_{44} - (L_{41} U_{14} + L_{42} U_{24} + L_{43} U_{34}) = -1 - (-12 \cdot 1,5 + 16 \cdot 0,75 + -1,5 \cdot -0,6667) = 4$$

Escrevendo as matrizes  $[L]$  e  $[U]$  na forma matricial:

$$L = \begin{bmatrix} 4 & 0 & 0 & 0 \\ -6 & 4 & 0 & 0 \\ 1 & 8 & 3 & 0 \\ -12 & 16 & -1,5 & 4 \end{bmatrix} \quad \text{e} \quad U = \begin{bmatrix} 1 & -0,5 & -0,75 & 1,5 \\ 0 & 1 & 0,5 & 0,75 \\ 0 & 0 & 1 & -0,6667 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para verificar se as duas matrizes estão corretas, é feita a sua multiplicação no MATLAB:

```
>> L = [4 0 0 0; -6,4 0 0; 1 8 3 0; -12 16 -1.5 4];
>> U = [1 -0.5 -0.75 1.5; 0 1 0.5 0.75; 0 0 1 -0.6667; 0 0 0 1];
>> L*U
ans =
```

4.0000	-2.0000	-3.0000	6.0000
-6.0000	7.0000	6.5000	-6.0000
1.0000	7.5000	6.2500	5.4999
-12.0000	22.0000	15.5000	-1.0000

 Esta matriz é idêntica à matriz de coeficientes na Eq. (4.39) (exceto por erros de arredondamento).

Assim que a decomposição é concluída, obtém-se a solução usando as Eqs. (4.22) e (4.23). Primeiramente, a matriz  $[L]$  e o vetor  $[b]$  são usados na Eq. (4.23),  $[L][y] = [b]$ , para se obter  $[y]$ :

$$\left[ \begin{array}{cccc|c} 4 & 0 & 0 & 0 & y_1 \\ -6 & 4 & 0 & 0 & y_2 \\ 1 & 8 & 3 & 0 & y_3 \\ -12 & 16 & -1,5 & 4 & y_4 \end{array} \right] = \left[ \begin{array}{c} 12 \\ -6,5 \\ 16 \\ 17 \end{array} \right] \quad (4.42)$$

Usando a substituição progressiva, a solução é:

$$y_1 = \frac{12}{4} = 3, \quad y_2 = \frac{-6,5 + 6y_1}{4} = 2,875, \quad y_3 = \frac{16 - y_1 - 8y_2}{3} = -3,333 \text{ e}$$

$$y_4 = \frac{17 + 12y_1 - 16y_2 + 1,5y_3}{4} = 0,5$$

Em seguida, a matriz  $[U]$  e o vetor  $[y]$  são usados na Eq. (4.22),  $[U][x] = [y]$ , para se obter  $[x]$ :

$$\left[ \begin{array}{cccc|c} 1 & -0,5 & -0,75 & 1,5 & x_1 \\ 0 & 1 & 0,5 & 0,75 & x_2 \\ 0 & 0 & 1 & -0,6667 & x_3 \\ 0 & 0 & 0 & 1 & x_4 \end{array} \right] = \left[ \begin{array}{c} 3 \\ 2,875 \\ -3,333 \\ 0,5 \end{array} \right] \quad (4.43)$$

Usando a substituição regressiva, a solução é:

$$x_4 = \frac{0,5}{1} = 0,5, \quad x_3 = -3,333 + 0,6667x_4 = -3, \quad x_2 = 2,875 - 0,5x_3 - 0,75x_4 = 4 \quad \text{e}$$

$$x_1 = 3 + 0,5x_2 + 0,75x_3 - 1,5x_4 = 2$$

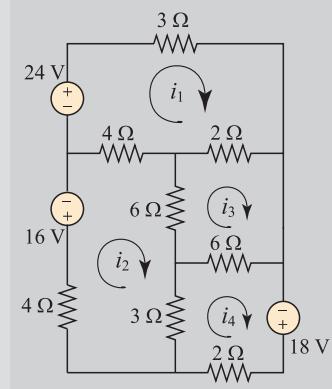
### Exemplo 4-6: Função criada no MATLAB para resolver sistemas de equações usando a decomposição LU com o método de Crout

Determine as correntes  $i_1, i_2, i_3$  e  $i_4$  no circuito mostrado na figura (que é igual à Fig. 4-1). Escreva o sistema de equações a ser resolvido na forma  $[a][i] = [b]$ . Resolva o sistema usando o método de decomposição LU e faça a decomposição empregando o método de Crout.

#### SOLUÇÃO

As correntes são determinadas a partir do conjunto de quatro equações dado na Eq. (4.1). As equações são deduzidas com o emprego da lei de Kirchhoff. Na forma matricial,  $[a][i] = [b]$ , as equações são:

$$\left[ \begin{array}{cccc|c} 9 & -4 & -2 & 0 & i_1 \\ -4 & 17 & -6 & -3 & i_2 \\ -2 & -6 & 14 & -6 & i_3 \\ 0 & -3 & -6 & 11 & i_4 \end{array} \right] = \left[ \begin{array}{c} 24 \\ -16 \\ 0 \\ 18 \end{array} \right] \quad (4.44)$$



Para resolver esse sistema de equações, três funções são criadas. As funções são as seguintes:

$[L \ U] = LUdecompCrout(A)$  Esta função decompõe a matriz A em matrizes L e U triangular inferior e superior, respectivamente.

y=ProgresSub(L, b) Esta função resolve um sistema de equações na forma triangular inferior.

y=RegresSub(L, b) Esta função resolve um sistema de equações na forma triangular superior.

Código da função LUdecompCrout:

**Programa 4-3 Função definida pelo usuário. Decomposição LU usando o método de Crout.**

```
function [L, U] = LUdecompCrout(A)
% A função decompõe a matriz A em uma matriz triangular inferior L e
% em uma matriz triangular superior U, usando o método de Crout de forma que A = LU.
% Variáveis de entrada:
% A Matriz de coeficientes.
% b vetor coluna de constantes.
% Variáveis de saída:
% L Matriz triangular inferior.
% U Matriz triangular superior.
```

```
[R, C] = size(A);
```

```
for i = 1:R
```

```
    L(i,1) = A(i,1);
```

Eq. (4.34).

```
    U(i,i) = 1;
```

Eq. (4.35).

```
end
```

```
for j = 2:R
```

```
    U(1,j) = A(1,j)/L(1,1);
```

Eq. (4.36).

```
end
```

```
for i = 2:R
```

```
    for j = 2:i
```

```
        L(i,j) = A(i,j) - L(i,1:j - 1)*U(1:j - 1,j);
```

Eq. (4.37).

```
    end
```

```
    for j = i + 1:R
```

```
        U(i,j) = (A(i,j) - L(i,1:i - 1)*U(1:i - 1,j))/L(i,i);
```

Eq. (4.38).

```
    end
```

```
end
```

Código da função ProgresSub:

**Programa 4-4 Função definida pelo usuário. Substituição progressiva.**

```
function y = ProgresSub(a,b)
```

% A função resolve um sistema de equações lineares  $ax = b$ , onde a é um

% matriz triangular inferior, usando a substituição progressiva.

% Variáveis de entrada:

% a Matriz de coeficientes.

% b Vetor coluna de constantes.

% Variável de saída:

% y Vetor coluna com a solução.

```
n = length(b);
y(1,1) = b(1)/a(1,1);
for i = 2:n
    y(i,1) = (b(i) - a(i,1:i - 1)*y(1:i - 1,1))./a(i,i);
end
```

Eq. (4.8).

Código da função RegressSub:

**Programa 4-5 Função definida pelo usuário. Substituição regressiva.**

```
function y = RegresSub(a,b)
% A função resolve um sistema de equações lineares ax = b, onde a é uma
% matriz triangular superior, usando a substituição regressiva.
% Variáveis de entrada:
% a Matriz de coeficientes.
% b Vetor coluna de constantes.
% Variável de saída:
% y Vetor coluna com a solução.

n = length(b);
y(n,1) = b(n)/a(n,n);
for i = n - 1:-1:1
    y(i,1) = (b(i) - a(i,i + 1:n)*y(i + 1:n,1))./a(i,i);
end
```

Eq. (4.5).

As funções são em seguida usadas em um programa de computador para resolver o problema no MATLAB. Tal programa segue os passos a seguir:

- A matriz de coeficientes  $[a]$  é decomposta em matrizes triangular superior  $[U]$  e triangular inferior  $[L]$  (usando a função LUdecompCrout).
- A matriz  $[L]$  e o vetor  $[b]$  são usados na Eq. (4.23),  $[L][y] = [b]$ , para determinar  $[y]$ , (usando a função ProgresSub).
- A solução  $[y]$  e a matriz  $[U]$  são usadas na Eq. (4.22),  $[U][i] = [y]$ , para determinar  $[i]$  (usando a função RegresSub).

Código do programa:

**Programa 4-6 Arquivo texto. Solução de sistemas com a decomposição LU usando o método de Crout.**

```
% Este programa resolve um sistema de equações usando a
% decomposição LU e o método de Crout.
a = [9 -4 -2 0; -4 17 -6 -3; -2 -6 14 -6; 0 -3 -6 11];
b = [24; -16; 0; 18];
[L, U] = LUdecompCrout(a);
y = ProgresSub(L,b);
i = RegresSub(U,y)
```

Quando se executa o programa, a seguinte solução é mostrada na janela de comandos do MATLAB:

```
i =
4.0343
1.6545
2.8452
3.6395
```

O arquivo texto pode ser facilmente modificado para resolver o sistema de equações  $[a][i] = [b]$  para a mesma matriz  $[a]$ , mas diferentes valores de  $[b]$ . A decomposição  $LU$  é feita apenas uma vez, e somente os dois últimos passos devem ser executados para cada  $[b]$ .

### 4.5.3 Decomposição LU com pivotação

A decomposição de uma matriz  $[a]$  em matrizes  $[L]$  e  $[U]$  significa que  $[a] = [L][U]$ . Na apresentação dos métodos de decomposição de Gauss e de Crout, assume-se que seja possível realizar todos os cálculos sem fazer pivotação. Na realidade, conforme discutido na Seção 4.3, a pivotação pode ser necessária para que o método de Gauss seja executado com sucesso. A pivotação também pode ser necessária no método de Crout. Se a pivotação for utilizada, então as matrizes  $[L]$  e  $[U]$  obtidas não correspondem à decomposição da matriz original  $[a]$ . O produto  $[L][U]$  fornece uma matriz com linhas que têm os mesmos elementos de  $[a]$ , mas que, em função da pivotação, estão em uma ordem diferente. Quando a pivotação é utilizada no procedimento de decomposição, as mudanças feitas devem ser gravadas e armazenadas. Isto é feito criando-se uma matriz  $[P]$ , chamada de matriz de permutação, tal que:

$$[P][a] = [L][U] \quad (4.45)$$

Se as matrizes  $[L]$  e  $[U]$  forem usadas para resolver um sistema de equações  $[a][x] = [b]$  (utilizando as Eqs. (4.23) e (4.22)), então a ordem das linhas de  $[b]$  deve ser alterada de uma forma tal que seja consistente com a pivotação. Isso é feito com a multiplicação de  $[b]$  pela matriz de permutação,  $[P][b]$ . O uso da matriz de permutação é mostrado na Seção 4.8.3, onde a decomposição é feita por uma função residente do MATLAB.

## 4.6 INVERSA DE UMA MATRIZ

A inversa de uma matriz quadrada  $[a]$  é a matriz  $[a]^{-1}$  tal que o produto das duas matrizes forneça a matriz identidade  $[I]$ .

$$[a][a]^{-1} = [a]^{-1}[a] = [I] \quad (4.46)$$

O processo de cálculo da inversa de uma matriz é essencialmente idêntico ao processo de solução de um sistema de equações lineares. Isso é ilustrado para o caso de uma matriz  $(4 \times 4)$ . Se  $[a]$  é uma dada matriz e  $[x]$  é inversa desconhecida de  $[a]$ , então:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.47)$$

A Eq. (4.47) pode ser rescrita como quatro sistemas de equações diferentes. Em cada sistema, uma coluna da matriz  $[x]$  é a incógnita:

$$\begin{array}{lcl} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ x_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, & \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{14} \\ x_{24} \\ x_{34} \\ x_{44} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{array} \quad (4.48)$$

A solução dos quatro sistemas de equações na Eq. (4.48) fornece as quatro colunas da inversa de  $[a]$ . Os sistemas de equações podem ser resolvidos por qualquer um dos métodos já introduzidos neste capítulo (ou outros métodos). Dois desses métodos, o método de decomposição *LU* e o método de eliminação de Gauss-Jordan, são descritos a seguir de forma mais detalhada.

#### 4.6.1 Cálculo da inversa com o método de decomposição *LU*

O método de decomposição *LU* é especialmente adequado para o cálculo da inversa de uma matriz. Conforme mostrado nas Eqs. (4.48), a matriz de coeficientes nos quatro sistemas de equações é a mesma. Consequentemente, a decomposição *LU* da matriz  $[a]$  é calculada apenas uma vez. Então, cada um dos sistemas é resolvido usando primeiramente a Eq. (4.23) (substituição progressiva) e então a Eq. (4.22) (substituição regressiva). Isso é ilustrado, utilizando o MATLAB, no Exemplo 4-7.

#### Exemplo 4-7: Determinação da inversa de uma matriz usando o método de decomposição *LU*

Determine a inversa da matriz  $[a]$  usando o método de decomposição *LU*.

$$[a] = \begin{bmatrix} 0,2 & -5 & 3 & 0,4 & 0 \\ -0,5 & 1 & 7 & -2 & 0,3 \\ 0,6 & 2 & -4 & 3 & 0,1 \\ 3 & 0,8 & 2 & -0,4 & 3 \\ 0,5 & 3 & 2 & 0,4 & 1 \end{bmatrix} \quad (4.49)$$

Faça os cálculos escrevendo uma função no MATLAB. Chame a função de `invA = InversaLU(A)`, onde  $A$  é a matriz a ser invertida e  $invA$  é a sua inversa. Na função, use as funções `LUdecompCrout`, `ProgresSub` e `RegresSub` escritas no Exemplo 4-6.

## SOLUÇÃO

Se a inversa de  $[a]$  é  $[x]$  ( $[x] = [a]^{-1}$ ), então  $[a][x] = [I]$ , o que forma cinco conjuntos de cinco equações a serem resolvidos. Em cada conjunto de equações, calcula-se uma coluna da inversa.

$$\begin{bmatrix} 0,2 & -5 & 3 & 0,4 & 0 \\ -0,5 & 1 & 7 & -2 & 0,3 \\ 0,6 & 2 & -4 & 3 & 0,1 \\ 3 & 0,8 & 2 & -0,4 & 3 \\ 0,5 & 3 & 2 & 0,4 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \\ x_{51} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0,2 & -5 & 3 & 0,4 & 0 \\ -0,5 & 1 & 7 & -2 & 0,3 \\ 0,6 & 2 & -4 & 3 & 0,1 \\ 3 & 0,8 & 2 & -0,4 & 3 \\ 0,5 & 3 & 2 & 0,4 & 1 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \\ x_{52} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0,2 & -5 & 3 & 0,4 & 0 \\ -0,5 & 1 & 7 & -2 & 0,3 \\ 0,6 & 2 & -4 & 3 & 0,1 \\ 3 & 0,8 & 2 & -0,4 & 3 \\ 0,5 & 3 & 2 & 0,4 & 1 \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ x_{43} \\ x_{53} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0,2 & -5 & 3 & 0,4 & 0 \\ -0,5 & 1 & 7 & -2 & 0,3 \\ 0,6 & 2 & -4 & 3 & 0,1 \\ 3 & 0,8 & 2 & -0,4 & 3 \\ 0,5 & 3 & 2 & 0,4 & 1 \end{bmatrix} \begin{bmatrix} x_{14} \\ x_{24} \\ x_{34} \\ x_{44} \\ x_{54} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0,2 & -5 & 3 & 0,4 & 0 \\ -0,5 & 1 & 7 & -2 & 0,3 \\ 0,6 & 2 & -4 & 3 & 0,1 \\ 3 & 0,8 & 2 & -0,4 & 3 \\ 0,5 & 3 & 2 & 0,4 & 1 \end{bmatrix} \begin{bmatrix} x_{15} \\ x_{25} \\ x_{35} \\ x_{45} \\ x_{55} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.50)$$

A solução é obtida com a função `InversaLU` listada abaixo. A função pode ser usada para calcular a inversa de uma matriz quadrada de qualquer tamanho.

A função executa as seguintes operações:

- A matriz  $[a]$  é decomposta em matrizes  $[L]$  e  $[U]$  com a aplicação do método de Crout. Isso é feito com o emprego da função `LUdecompCrout` escrita no Exemplo 4-6.
- Cada sistema de equações nas Eqs. (4.50) é resolvido com o uso das Eqs. (4.23) e (4.22). Isso é feito primeiro usando a função `ProgresSub` e depois a função `RegresSub` (ver Exemplo 4-6).

### Programa 4-7 Função definida pelo usuário. Inversa de uma matriz.

```
function invA = InversaLU(A)
```

```
% Esta função calcula a inversa de uma matriz
```

```
% Variáveis de entrada:
```

```
% A Matriz a ser invertida.
```

```
% Variável de saída:
```

```
% invA Inversa de A.
```

```
[nR nC] = size(A);
```

```
I = eye(nR);
```

```
[L U] = LUdecompCrout(A);
```

```
for j = 1:nC
```

```
    y = ProgresSub(L,I(:,j));
```

```
    invA(:,j) = RegresSub(U,y);
```

```
end
```

Cria uma matriz identidade com o mesmo tamanho de  $[A]$ .

Decomposição de  $[A]$  em  $[L]$  e  $[U]$ .

Em cada passo do *loop*, um conjunto de equações na Eq. (4.50) é resolvido. Cada solução corresponde a uma coluna na inversa da matriz.

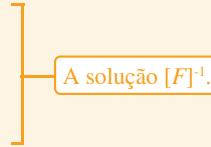
A função é então utilizada na janela de comandos do MATLAB para solucionar o problema.

```
>> F = [0.2 -5 3 0.4 0; -0.5 1 7 -2 0.3; 0.6 2 -4 3 0.1; 3 0.8 2 -0.4 3; 0.5 3 2 0.4 1];
```

```
>> invF = InversaLU(F)
```

```
invF =
```

-0.7079	2.5314	2.4312	0.9666	-3.9023
-0.1934	0.3101	0.2795	0.0577	-0.2941
0.0217	0.3655	0.2861	0.0506	-0.2899
0.2734	-0.1299	0.1316	-0.1410	0.4489
0.7815	-2.8751	-2.6789	-0.7011	4.2338



```
>> invF*F
```

```
ans =
```

1.0000	-0.0000	0.0000	-0.0000	-0.0000
0.0000	1.0000	0.0000	-0.0000	0
0	-0.0000	1.0000	-0.0000	-0.0000
-0.0000	0.0000	-0.0000	1.0000	-0.0000
-0.0000	0.0000	-0.0000	-0.0000	1.0000

Verifica se  $[F][F]^{-1} = [I]$ .

#### 4.6.2 Cálculo da inversa usando o método de Gauss-Jordan

O método de Gauss-Jordan pode ser facilmente adaptado para calcular a inversa de uma matriz quadrada  $[a]$  ( $n \times n$ ). Isso é feito primeiramente com a incorporação de uma matriz identidade  $[I]$  de mesmo tamanho à matriz  $[a]$ , o que é mostrado de forma esquemática para uma matriz ( $4 \times 4$ ) na Fig. 4-20a. Em seguida, o procedimento de Gauss-Jordan é aplicado de tal forma que os elementos da matriz  $[a]$  (o lado esquerdo da matriz aumentada) sejam convertidos em 1s ao longo da diagonal principal e em 0s nos demais campos. Durante esse processo, os termos da matriz identidade na Fig. 4-20a (o lado direito da matriz aumentada) são modificados e se tornam os elementos  $[a']$  na Fig. 4.20b, que constitui a inversa de  $[a]$ .

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 1 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 1 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 & 0 & 1 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 & 0 & 0 & 1 \end{bmatrix} \xrightarrow{\text{Procedimento de Gauss-Jordan}} \begin{bmatrix} 1 & 0 & 0 & 0 & a'_{11} & a'_{12} & a'_{13} & a'_{14} \\ 0 & 1 & 0 & 0 & a'_{21} & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & 1 & 0 & a'_{31} & a'_{32} & a'_{33} & a'_{34} \\ 0 & 0 & 0 & 1 & a'_{41} & a'_{42} & a'_{43} & a'_{44} \end{bmatrix}$$

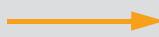
(a) (b)

**Figura 4-20** Cálculo da inversa com o método de Gauss-Jordan.

#### 4.7 MÉTODOS ITERATIVOS

Sistemas de equações lineares também podem ser resolvidos com o uso de uma abordagem iterativa. O processo, em princípio, é idêntico ao método da iteração de

ponto fixo usado para resolver uma única equação não-linear (ver Seção 3.7). Na solução de sistemas de equações usando processos iterativos, as equações são colocadas em uma forma explícita na qual cada incógnita é escrita em termos das demais incógnitas. A forma explícita de um sistema de equações é ilustrada na Fig. 4-21.

$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = b_1$ $a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2$ $a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3$ $a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4$	 Escrevendo as equações em uma forma explícita	$x_1 = [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11}$ $x_2 = [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22}$ $x_3 = [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33}$ $x_4 = [b_4 - (a_{41}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44}$
<b>(a)</b>		<b>(b)</b>

**Figura 4-21** Forma padronizada **(a)** e explícita **(b)** de um sistema de quatro equações.

O processo de solução começa com a escolha de valores iniciais para as incógnitas (primeira solução estimada). Na primeira iteração, a primeira solução assumida é substituída no lado direito das equações, e os novos valores calculados para as incógnitas formam a segunda solução estimada. Na segunda iteração, a segunda solução é substituída de volta nas equações para que novos valores sejam obtidos para as incógnitas, e isso constitui a terceira solução estimada. As iterações continuam da mesma forma e, quando o método dá certo, as soluções obtidas durante as iterações sucessivas convergem para a solução real. Em um sistema com  $n$  equações, as equações explícitas para as incógnitas  $[x_i]$  são:

$$x_i = \frac{1}{a_{ii}} \left[ b_i - \left( \sum_{\substack{j=1 \\ j \neq i}}^{j=n} a_{ij} x_j \right) \right] \quad i = 1, 2, \dots, n \quad (4.51)$$

### Condição para a convergência

Para um sistema de  $n$  equações  $[a][x] = [b]$ , uma condição suficiente para a convergência ocorre se, em cada uma das linhas da matriz de coeficientes  $[a]$ , o valor absoluto do elemento diagonal for maior que a soma dos valores absolutos dos elementos fora da diagonal.

$$|a_{ii}| > \sum_{j=1, j \neq i}^{j=n} |a_{ij}| \quad (4.52)$$

Essa condição é suficiente mas não necessária para a convergência do método iterativo. Quando a condição (4.52) é satisfeita, a matriz  $[a]$  é classificada como **diagonalmente dominante**, e o processo iterativo converge para a solução. A solução, no entanto, pode convergir mesmo quando a Eq. (4.52) não é satisfeita.

Dois métodos iterativos específicos são apresentados a seguir, os métodos de Jacobi e de Gauss-Seidel. A diferença entre esses dois métodos está na maneira pela qual os novos valores calculados para as incógnitas são utilizados. No método de Jacobi, os valores das incógnitas no lado direito da Eq. (4.51) são atualizados todos de uma vez no final de cada iteração. No método de Gauss-Seidel, o

valor de cada incógnita é atualizado (e usado no cálculo da nova estimativa das demais incógnitas dentro da mesma iteração) assim que se calcula uma nova estimativa para essa incógnita.

#### 4.7.1 Método iterativo de Jacobi

No método de Jacobi, um valor inicial é escolhido para cada uma das incógnitas,  $x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$ . Se não houver nenhuma informação a respeito dos valores aproximados das incógnitas, pode-se assumir que o valor inicial de todas elas seja igual a zero. A segunda estimativa da solução,  $x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}$ , é calculada com a substituição da primeira estimativa no lado direito das Eqs. (4.51):

$$x_i^{(2)} = \frac{1}{a_{ii}} \left[ b_i - \left( \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(1)} \right) \right] \quad i = 1, 2, \dots, n \quad (4.53)$$

Em geral, a  $(k+1)$ -ésima estimativa da solução é calculada a partir da  $k$ -ésima estimativa usando:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[ b_i - \left( \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} \right) \right] \quad i = 1, 2, \dots, n \quad (4.54)$$

As iterações continuam até que as diferenças entre os valores obtidos nas iterações sucessivas sejam pequenas. As iterações podem ser interrompidas quando o valor absoluto do erro relativo estimado (ver Seção 3.2) de todas as incógnitas for menor que algum valor predeterminado:

$$\left| \frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k)}} \right| < \varepsilon \quad i = 1, 2, \dots, n \quad (4.55)$$

#### 4.7.2 Método iterativo de Gauss-Seidel

No método de Gauss-Seidel, valores iniciais são assumidos para as incógnitas  $x_2, x_3, \dots, x_n$  (todas as incógnitas, exceto  $x_1$ ). Se não houver nenhuma informação a respeito dos valores aproximados das incógnitas, pode-se assumir que o valor inicial de todas elas seja igual a zero. Os primeiros valores assumidos para as incógnitas são substituídos na Eq. (4.51) com  $i = 1$  para calcular o valor de  $x_1$ . Em seguida, a Eq. (4.51) com  $i = 2$  é usada no cálculo de um novo valor para  $x_2$ . Isso é seguido do uso da Eq. (4.51) com  $i = 3$  para calcular um novo valor para  $x_3$ , o que continua até que  $i = n$ , o que representa o final da primeira iteração. Em seguida, a segunda iteração começa com  $i = 1$ , onde um novo valor é calculado para  $x_1$ , e assim por diante. No método de Gauss-Seidel, os valores atuais das incógnitas são utilizados no cálculo do novo valor da próxima incógnita. Em outras palavras, à medida que um novo valor é calculado, ele é imediatamente utilizado na próxima aplicação da Eq. (4.51) (no método de Jacobi, os valores das incógnitas obtidos em uma iteração

são utilizados como um conjunto completo no cálculo dos novos valores das incógnitas na próxima iteração. Os valores das incógnitas não são atualizados no meio da iteração).

A aplicação da Eq. (4.51) no método de Gauss-Seidel leva à fórmula iterativa:

$$\begin{aligned}x_1^{(k+1)} &= \frac{1}{a_{11}} \left[ b_1 - \sum_{j=2}^{j=n} a_{1j} x_j^{(k)} \right] \\x_i^{(k+1)} &= \frac{1}{a_{ii}} \left[ b_i - \left( \sum_{j=1}^{j=i-1} a_{ij} x_j^{(k+1)} + \sum_{j=i+1}^{j=n} a_{ij} x_j^{(k)} \right) \right] \quad i = 2, 3, \dots, n-1 \\x_n^{(k+1)} &= \frac{1}{a_{nn}} \left[ b_n - \sum_{j=1}^{j=n-1} a_{nj} x_j^{(k+1)} \right]\end{aligned}\quad (4.56)$$

Note que os valores das incógnitas na iteração  $k+1$ ,  $x_i^{(k+1)}$ , são calculados usando os valores  $x_j^{(k+1)}$  obtidos na iteração  $k+1$  para  $j < i$  e usando os valores  $x_j^{(k)}$  para  $j > i$ . O critério de interrupção das iterações é o mesmo utilizado no método de Jacobi, a Eq. (4.55). O método de Gauss-Seidel converge mais rápido do que o método de Jacobi e requer menos memória computacional quando programado. O método é ilustrado para um sistema de quatro equações no Exemplo 4-8.

### Exemplo 4-8: Solução de um conjunto de quatro equações lineares usando o método de Gauss-Seidel

Resolva o seguinte conjunto de quatro equações lineares usando o método iterativo de Gauss-Seidel.

$$\begin{aligned}9x_1 - 2x_2 + 3x_3 + 2x_4 &= 54,5 \\2x_1 + 8x_2 - 2x_3 + 3x_4 &= -14 \\-3x_1 + 2x_2 + 11x_3 - 4x_4 &= 12,5 \\-2x_1 + 3x_2 + 2x_3 + 10x_4 &= -21\end{aligned}$$

### SOLUÇÃO

Primeiramente, as equações são escritas de forma explícita (ver Fig. 4.21):

$$\begin{aligned}x_1 &= [54,5 - (-2x_2 + 3x_3 + 2x_4)]/9 \\x_2 &= [-14 - (2x_1 - 2x_3 + 3x_4)]/8 \\x_3 &= [12,5 - (-3x_1 + 2x_2 - 4x_4)]/11 \\x_4 &= [-21 - (-2x_1 + 3x_2 + 2x_3)]/10\end{aligned}\quad (4.57)$$

Como ponto de partida, assume-se que o valor inicial de todas as incógnitas,  $x_1^{(1)}$ ,  $x_2^{(1)}$ ,  $x_3^{(1)}$  e  $x_4^{(1)}$ , seja igual a zero. As duas primeiras iterações são calculadas manualmente, e então um programa é usado no MATLAB para calcular os valores das incógnitas em sete iterações.

#### Cálculo manual das duas primeiras iterações:

A segunda estimativa da solução ( $k = 2$ ) é calculada na primeira iteração utilizando as Eqs. (4.57). Os valores substituídos em  $x_i$  no lado direito das equações são os mais recentes. Isso significa que, quando se usa a primeira equação para calcular  $x_1^{(2)}$ , todos os valores  $x_i$  são nulos. Assim, quando

a segunda equação é usada para calcular  $x_2^{(2)}$ , o novo valor  $x_1^{(2)}$  é substituído em  $x_1$ , mas os valores antigos  $x_3^{(1)}$  e  $x_4^{(1)}$  são usados em  $x_3$  e  $x_4$ , e assim por diante:

$$x_1^{(2)} = [54,5 - (-2 \cdot 0 + 3 \cdot 0 + 2 \cdot 0)]/9 = 6,056$$

$$x_2^{(2)} = [-14 - (2 \cdot 6,056 - (2 \cdot 0) + 3 \cdot 0)]/8 = -3,264$$

$$x_3^{(2)} = [12,5 - (-3 \cdot 6,056 + 2 \cdot -3,264 - (4 \cdot 0))]/11 = 3,381$$

$$x_4^{(2)} = [-21 - (-2 \cdot 6,056 + 3 \cdot -3,264 + 2 \cdot 3,381)]/10 = -0,5860$$

A terceira estimativa da solução ( $k = 3$ ) é calculada na segunda iteração:

$$x_1^{(3)} = [54,5 - (-2 \cdot -3,264 + 3 \cdot 3,381 + 2 \cdot -0,5860)]/9 = 4,333$$

$$x_2^{(3)} = [-14 - (2 \cdot 4,333 - (2 \cdot 3,381) + 3 \cdot -0,5860)]/8 = -1,768$$

$$x_3^{(3)} = [12,5 - (-3 \cdot 4,333 + 2 \cdot -1,768 - (4 \cdot -0,5860))]/11 = 2,427$$

$$x_4^{(3)} = [-21 - (-2 \cdot 4,333 + 3 \cdot -1,768 + 2 \cdot 2,427)]/10 = -1,188$$

#### **Programa de MATLAB que calcula as sete primeiras iterações:**

O seguinte programa de MATLAB calcula as sete primeiras iterações da solução usando as Eqs. (4.57):

#### **Programa 4-8 Arquivo texto. Iteração de Gauss-Seidel.**

```
k = 1; x1 = 0; x2 = 0; x3 = 0; x4 = 0;
disp(' k      x1      x2      x3      x4')
fprintf('%2.0f    %-8.5f %-8.5f %-8.5f %-8.5f\n', k, x1, x2, x3, x4)
for k = 2 : 8
    x1 = (54.5 - (-2*x2 + 3*x3 + 2*x4))/9;
    x2 = (-14 - (2*x1 - 2*x3 + 3*x4))/8;
    x3 = (12.5 - (-3*x1 + 2*x2 - 4*x4))/11;
    x4 = (-21 - (-2*x1 + 3*x2 + 2*x3))/10;
    fprintf('%2.0f    %-8.5f %-8.5f %-8.5f %-8.5f\n', k, x1, x2, x3, x4)
end
```

Quando o programa é executado, os seguintes resultados aparecem na janela de comandos do MATLAB.

k	x1	x2	x3	x4
1	0.00000	0.00000	0.00000	0.00000
2	6.05556	-3.26389	3.38131	-0.58598
3	4.33336	-1.76827	2.42661	-1.18817
4	5.11778	-1.97723	2.45956	-0.97519
5	5.01303	-2.02267	2.51670	-0.99393
6	4.98805	-1.99511	2.49806	-1.00347
7	5.00250	-1.99981	2.49939	-0.99943
8	5.00012	-2.00040	2.50031	-0.99992

Os resultados mostram que a solução converge para a solução exata, que é:

$$x_1 = 5, x_2 = -2, x_3 = 2,5 \quad \text{e} \quad x_4 = -1.$$

## 4.8 USO DE FUNÇÕES RESIDENTES DO MATLAB PARA SOLUCIONAR SISTEMAS DE EQUAÇÕES LINEARES

O MATLAB dispõe de operações matemáticas e de funções residentes que podem ser utilizadas para solucionar sistemas de equações lineares e para realizar as operações matriciais descritas neste capítulo.

### 4.8.1 Solução de um sistema de equações usando as divisões à esquerda e à direita do MATLAB

**Divisão à esquerda \:** A divisão à esquerda pode ser usada para resolver um sistema de  $n$  equações escritas na forma matricial  $[a][x] = [b]$ , onde  $[a]$  é a matriz de coeficientes ( $n \times n$ ),  $[x]$  é um vetor coluna ( $n \times 1$ ) com as incógnitas e  $[b]$  é um vetor coluna ( $n \times 1$ ) de constantes.

$$\mathbf{x} = \mathbf{a} \backslash \mathbf{b}$$

Por exemplo, a solução do sistema de equações nos Exemplos 4-1 e 4-2 é calculada na janela de comandos do MATLAB:

```
>> a = [4 -2 -3 6 ; -6 7 6.5 -6 ; 1 7.5 6.25 5.5 ; -12 22 15.5 -1];
>> b = [12; -6.5; 16; 17];
>> x = a\b
x =
    2.0000
    4.0000
   -3.0000
    0.5000
```

**Divisão à direita / :** A divisão à direita pode ser usada para resolver um sistema de  $n$  equações escritas na forma matricial  $[x][a] = [b]$ , onde  $[a]$  é a matriz de coeficientes ( $n \times n$ ),  $[x]$  é um vetor linha ( $1 \times n$ ) com as incógnitas e  $[b]$  é um vetor linha ( $1 \times n$ ) de constantes.

$$\mathbf{x} = \mathbf{b} / \mathbf{a}$$

Por exemplo, a solução do sistema de equações nos Exemplos 4-1 e 4-2 pode ser calculada na janela de comandos do MATLAB da seguinte maneira:

```
>> a = [4 -6 1 -12 ; -2 7 7.5 22 ; -3 6.5 6.25 15.5 ; 6 -6 5.5 -1];
>> b = [12 -6.5 16 17];
>> x = b/a
x =
    2.0000    4.0000   -3.0000    0.5000
```

Note que a matriz  $[a]$  usada na divisão à direita é a transposta da matriz usada nos cálculos com a divisão à esquerda.

### 4.8.2 Solução de sistemas de equações usando a operação inversa disponível no MATLAB

Na forma matricial, o vetor  $[x]$  no sistema de equações  $[a][x] = [b]$  pode ser obtido com a multiplicação de ambos os lados do sistema por  $[a]^{-1}$  (a inversa de  $[a]$ ), resultando em:

$$[a]^{-1}[a][x] = [a]^{-1}[b] \quad (4.58)$$

Como  $[a]^{-1}[a] = [I]$  (matriz identidade), e  $[I][x] = [x]$ , a Eq. (4.58) se reduz a:

$$[x] = [a]^{-1}[b] \quad (4.59)$$

No MATLAB, a inversa de uma matriz  $[a]$  pode ser calculada com a elevação da matriz à potência de  $-1$  ou com o uso da função `inv(a)`. Uma vez calculada a inversa, a solução é obtida com a sua multiplicação pelo vetor  $[b]$ . Isso é demonstrado para a solução do sistema dos Exemplos 4-1 e 4-2.

```
>> a = [4 -2 -3 6; -6 7 6.5 -6; 1 7.5 6.25 5.5; -12 22 15.5 -1];
>> b = [12; -6.5; 16; 17];
>> x = a^-1*b
x =
    2.0000
    4.0000
   -3.0000
    0.5000
```

O mesmo resultado é obtido digitando-se: `>>x = inv(a)*b.`

### 4.8.3 Função residente do MATLAB disponível para o cálculo da decomposição LU

O MATLAB possui uma função residente, chamada de `lu`, que decompõe uma matriz  $[a]$  no produto  $[L][U]$ , tal que  $[a] = [L][U]$ , onde  $[L]$  é uma matriz triangular inferior e  $[U]$  é uma matriz triangular superior. Uma forma da função é:

$$[L, U, P] = lu(a)$$

$L$  é uma matriz triangular inferior

$U$  é uma matriz triangular superior

$P$  é a matriz de permutação

$a$  é a matriz a ser decomposta

O MATLAB usa a pivotação parcial para realizar a fatoração. Conseqüentemente, as matrizes  $[L]$  e  $[U]$  determinadas pelo MATLAB resultam da fatoração de uma matriz com linhas que podem estar em uma ordem diferente daquela de  $[a]$ . A matriz de permutação  $[P]$  (uma matriz com 1s e 0s) contém a informação sobre a pivotação. A multiplicação de  $[a]$  pela matriz  $[P]$  fornece uma matriz cuja decomposição é dada por  $[L]$  e  $[U]$  (ver Seção 4.5.3):

$$[L][U] = [P][a] \quad (4.60)$$

A matriz  $[P][a]$  tem as mesmas linhas de  $[a]$ , mas em ordem diferente. Se o MATLAB não usar a pivotação parcial ao utilizar a função `lu`, então a matriz de permutação  $[P]$  é a matriz identidade.

Se as matrizes  $[L]$  e  $[U]$  determinadas pela função `lu` forem subsequentemente utilizadas para resolver um sistema de equações  $[a][x] = [b]$  (usando as Eqs. (4.23) e (4.22)), então o vetor  $[b]$  deve ser multiplicado pela matriz de permutação  $[P]$ . Isso faz com que as linhas de  $[b]$  se tornem consistentes com a pivotação em  $[a]$ . O seguinte exemplo mostra a solução no MATLAB do sistema de equações dos Exemplos 4-1 e 4-2 usando a função `lu`.

```
>> a = [4 -2 -3 6 ; -6 7 6.5 -6 ; 1 7.5 6.25 5.5 ; -12 22 15.5 -1];
>> b = [12; -6.5; 16; 17];
```

```
>> [L, U, P] = lu(a)
```

Decomposição de  $[a]$  no MATLAB usando a função `lu`.

$L =$

1.0000	0	0	0
-0.0833	1.0000	0	0
-0.3333	0.5714	1.0000	0
0.5000	-0.4286	-0.9250	1.0000

$U =$

-12.0000	22.0000	15.5000	-1.0000
0	9.3333	7.5417	5.4167
0	0	-2.1429	2.5714
0	0	0	-0.800

$P =$

0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0

```
>> y = L\ (P*b)
```

Resolve para  $y$  na Eq. (4.23).

A multiplicação  $[P][a]$  fornece a matriz pivotada  $[a]$ : `>> P*a`

ans =

-12.0000	22.0000	15.5000	-1.0000
1.0000	7.5000	6.2500	5.5000
4.0000	-2.0000	-3.0000	6.0000
-6.0000	7.0000	6.5000	-6.0000

$y =$

17.0000

17.4167

7.7143

-0.4000

```
>> x = U\y
```

Resolve para  $x$  na Eq. (4.22).

$x =$

2.0000

4.0000

-3.0000

0.5000

O vetor  $[b]$  é multiplicado pela matriz de permutação.

#### 4.8.4 Funções residentes do MATLAB adicionais

O MATLAB possui muitas funções residentes que podem ser úteis na análise de sistemas de equações. Várias dessas funções são apresentadas na Tabela 4-1. Note que as operações que estão relacionadas a algumas das funções na tabela são discutidas na Seção 4.9.

**Tabela 4-1** Funções disponíveis no MATLAB para realizar operações matriciais e a análise de matrizes

Função	Descrição	Exemplo
inv (A)	Inversa de uma matriz. A é uma matriz quadrada. Retorna a inversa de A.	<pre>&gt;&gt; A = [-3 1 0.6; 0.2 -4 3; 0.1 0.5 2]; &gt;&gt; Ain = inv(A) Ain = -0.3310 -0.0592 0.1882 -0.0035 -0.2111 0.3178 0.0174 0.0557 0.4111</pre>
d=det (A)	Determinante de uma matriz. A é uma matriz quadrada, d é o determinante de A.	<pre>&gt;&gt; A = [-3 1 0.6; 0.2 -4 3; 0.1 0.5 2]; &gt;&gt; d = det(A) d = 28.7000</pre>
n=norm (A)	Norma vetorial e matricial A é um vetor ou uma matriz, n é sua norma.	<pre>&gt;&gt; A = [2 0 7 -9]; &gt;&gt; n=norm(A,1) n = 18 &gt;&gt; n = norm(A,inf) n = 9 &gt;&gt; n = norm(A,2) n = 11.5758</pre>
n=norm (A, p)	<b>Quando A é um vetor:</b> norm (A, p) retorna: $\sum(\text{abs}(A.^p))^{(1/p)}$ . $p=\text{inf}$ A norma infinita (ver Equação (4.70)). norm (A) Retorna a norma 2 Euclidiana (ver Eq. (4.72)), o mesmo que norm (A, 2) .	<pre>&gt;&gt; A = [1 3 -2; 0 -1 4; 5 2 3]; &gt;&gt; n = norm(A,1) n = 9 &gt;&gt; n = norm(A,2) n = 6.4818 &gt;&gt; n = norm(A,inf) n = 10</pre>
c=cond (A)	<b>Quando A é uma matriz:</b> norm (A, p) retorna: $p=1$ A norma 1 (maior soma de colunas de A (ver Eq. (4.74)). $p=2$ O maior valor singular, o mesmo que norm (A) (ver Equação (4.75)). Esta não é a norma Euclidiana (ver Equação (4.76)). $p=\text{inf}$ A norma infinita (ver Equação (4.73)).	<pre>&gt;&gt; a = [9 -2 3 2; 2 8 -2 3; -3 2 11 -4; -2 3 2 10]; &gt;&gt; cond(a,inf) ans = 3.8039</pre>
c=cond (A, p)	Número condição (ver Equação (4.86)) A é uma matriz quadrada, c é o número condição de A. cond (A) O mesmo que $p=2$ . $p=1$ O número condição de norma 1. $p=2$ O número condição de norma 2. $p=\text{inf}$ O número condição de norma infinita.	<span style="border: 1px solid orange; padding: 2px;">Ver o final do Exemplo 4-10.</span>

$$\begin{bmatrix} A_{11} & A_{12} & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & 0 \\ 0 & A_{32} & A_{33} & A_{34} & 0 \\ 0 & 0 & A_{43} & A_{44} & A_{45} \\ 0 & 0 & 0 & A_{54} & A_{55} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{bmatrix}$$

**Figura 4-22** Sistema tridiagonal de cinco equações.

## 4.9 SISTEMAS TRIDIAGONAIS DE EQUAÇÕES

Sistemas tridiagonais de equações lineares possuem uma matriz de coeficientes repleta de zeros, exceto ao longo da diagonal principal e nos elementos logo acima e logo abaixo da diagonal principal. Um sistema tridiagonal com  $n$  equações é mostrado na forma matricial na Eq. (4.61) e ilustrado para um sistema com cinco equações na Fig. 4-22.

$$\begin{bmatrix} A_{11} & A_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & A_{34} & 0 & 0 & 0 & 0 \\ \dots & \dots \\ \dots & \dots \\ 0 & 0 & 0 & 0 & A_{n-2, n-3} & A_{n-2, n-2} & A_{n-2, n-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{n-1, n-2} & A_{n-1, n-1} & A_{n-1, n} \\ 0 & 0 & 0 & 0 & 0 & 0 & A_{n, n-1} & A_{n, n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \dots \\ \dots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ \dots \\ \dots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix} \quad (4.61)$$

A matriz de coeficientes de sistemas tridiagonais tem muitos elementos nulos (especialmente quando o sistema contém um grande número de equações). O sistema poderia ser resolvido com os métodos usuais (Gauss, Gauss-Jordan, decomposição LU), mas isso implicaria o armazenamento de um grande número de elementos nulos e também a execução de um grande número de operações desnecessárias (com zeros). Para economizar memória e tempo computacional, métodos numéricos especiais têm sido desenvolvidos para resolver sistemas tridiagonais de equações. Um desses métodos, o algoritmo de Thomas, é descrito nesta seção.

Muitas aplicações na engenharia e na ciência requerem a solução de sistemas tridiagonais de equações. Alguns métodos numéricos usados para resolver equações diferenciais também envolvem a solução de tais sistemas.

### Algoritmo de Thomas para resolver sistemas tridiagonais

O algoritmo de Thomas é um procedimento aplicado na solução de sistemas tridiagonais de equações. O método de solução no algoritmo de Thomas é similar ao método de eliminação de Gauss, no qual o sistema é primeiramente modificado para a forma triangular superior e então resolvido com o uso da substituição regressiva. O algoritmo de Thomas, no entanto, é muito mais eficiente porque apenas os elementos não-nulos da matriz são armazenados, e apenas as operações necessárias são executadas (operações desnecessárias com os elementos nulos são eliminadas).

O algoritmo de Thomas começa com o armazenamento dos elementos não-nulos da matriz de coeficientes [A] em três vetores. Os elementos diagonais  $A_{ii}$  são atribuídos ao vetor  $d$  (o nome  $d$  é usado para representar a diagonal), de forma que  $d_i = A_{ii}$ . Os elementos acima da diagonal  $A_{i, i+1}$  são atribuídos ao vetor  $a$  (de

above, ou acima da diagonal), de forma que  $a_i = A_{i,i+1}$ , e os elementos abaixo da diagonal,  $A_{i-1,i}$ , são atribuídos ao vetor  $b$  (de below, ou abaixo da diagonal), de forma que  $b_i = A_{i-1,i}$ . Com os elementos não-nulos da matriz de coeficientes armazenados como vetores, o sistema de equações tem a forma:

$$\begin{bmatrix} d_1 & a_1 & 0 & 0 & 0 & 0 & 0 \\ b_2 & d_2 & a_2 & 0 & 0 & 0 & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & b_{n-2} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \dots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix} \quad (4.62)$$

Deve-se enfatizar aqui que, na Eq. (4.62), a matriz de coeficientes é mostrada como uma matriz (com os 0s), mas no algoritmo de Thomas, apenas os vetores  $b$ ,  $d$  e  $a$  são armazenados.

Em seguida, a primeira linha é normalizada com a divisão da linha por  $d_1$ . Isso faz com que o elemento  $d_1$  (a ser usado como elemento pivô) seja igual a 1:

$$\begin{bmatrix} 1 & a'_1 & 0 & 0 & 0 & 0 & 0 \\ b_2 & d_2 & a_2 & 0 & 0 & 0 & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & b_{n-2} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} B'_1 \\ B_2 \\ B_3 \\ \dots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix} \quad (4.63)$$

onde  $a'_1 = a_1 / d_1$  e  $B'_1 = B_1 / d_1$ .

Agora, o elemento  $b_2$  é eliminado. A primeira linha (pivô) é multiplicada por  $b_2$  e então subtraída da segunda linha.

$$\begin{bmatrix} 1 & a'_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & d'_2 & a_2 & 0 & 0 & 0 & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & b_{n-2} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} B'_1 \\ B'_2 \\ B_3 \\ \dots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix} \quad (4.64)$$

onde  $d'_2 = d_2 - b_2 a'_1$  e  $B'_2 = B_2 - B_1 b_2$ .

As operações realizadas com a primeira e a segunda linha são repetidas com a segunda e a terceira linhas. A segunda linha é normalizada com a sua divisão por  $d'_2$ . Isso faz com que o elemento  $d'_2$  (a ser usado como pivô) seja igual a 1. A segunda linha é então usada para eliminar  $b_3$  na terceira linha.

Esse processo continua linha após linha até que o sistema de equações seja transformado para a forma triangular superior com 1s na diagonal:

$$\left[ \begin{array}{ccccccc} 1 & a'_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & a'_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & a'_3 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 1 & a'_{n-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{array} \right] = \left[ \begin{array}{c} B'_1 \\ B''_2 \\ B''_3 \\ \dots \\ B''_{n-2} \\ B''_{n-1} \\ B''_n \end{array} \right] \quad (4.65)$$

Assim que a matriz de coeficientes apresenta a forma triangular superior, os valores das incógnitas podem ser calculados utilizando a substituição regressiva.

Matematicamente, o algoritmo de Thomas pode ser resumido nos passos a seguir:

**Passo 1** Defina os vetores  $b = [0, b_2, b_3, \dots, b_n]$ ,  $d = [d_1, d_2, \dots, d_n]$ ,  $a = [a_1, a_2, \dots, a_{n-1}]$  e  $B = [B_1, B_2, \dots, B_n]$ .

**Passo 2** Calcule  $a_1 = \frac{a_1}{d_1}$  e  $B_1 = \frac{B_1}{d_1}$ .

**Passo 3** Para  $i = 2, 3, \dots, n-1$ , calcule:

$$a_i = \frac{a_i}{d_i - b_i a_{i-1}} \quad \text{e} \quad B_i = \frac{B_i - b_i B_{i-1}}{d_i - b_i a_{i-1}}$$

**Passo 4** Calcule:  $B_n = \frac{B_n - b_n B_{n-1}}{d_n - b_n a_{n-1}}$

**Passo 5** Calcule a solução usando a substituição regressiva:

$$x_n = B_n \quad \text{e, para } i = n-1, n-2, n-3, \dots, 2, 1, \quad x_i = B_i - a_i x_{i+1}$$

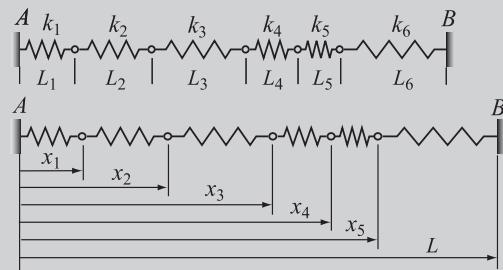
A solução de um sistema tridiagonal de equações é mostrada no Exemplo 4-9 usando uma função criada no MATLAB.

**Exemplo 4-9: Solução de um sistema tridiagonal de equações usando o algoritmo de Thomas**

Seis molas com diferentes constantes elásticas  $k_i$  e comprimentos  $L_i$  na condição de repouso são amarradas em série. O ponto final  $B$  é então deslocado de tal forma que a distância entre os pontos  $A$  e  $B$  seja de  $L = 1,2$  m. Determine as posições  $x_1, x_2, \dots, x_5$  dos pontos finais das molas.

As constantes das molas e os seus comprimentos em repouso são:

mola	1	2	3	4	5	6
$k$ (kN/m)	8	9	15	12	10	18
$L$ (m)	0,18	0,22	0,26	0,19	0,15	0,30



## SOLUÇÃO

A força  $F$  em uma mola é dada por:

$$F = k\delta$$

onde  $k$  é a constante da mola e  $\delta$  é sua extensão além de seu comprimento na condição de repouso. Como as molas estão amarradas em série, a força em todas as molas é a mesma. Conseqüentemente, é possível escrever cinco equações igualando a força em cada duas molas adjacentes. Por exemplo, a condição que a força na primeira mola é igual à força na segunda mola resulta em:

$$k_1(x_1 - L_1) = k_2[(x_2 - x_1) - L_2]$$

De forma similar, quatro equações adicionais podem ser escritas:

$$k_2[(x_2 - x_1) - L_2] = k_3[(x_3 - x_2) - L_3]$$

$$k_3[(x_3 - x_2) - L_3] = k_4[(x_4 - x_3) - L_4]$$

$$k_4[(x_4 - x_3) - L_4] = k_5[(x_5 - x_4) - L_5]$$

$$k_5[(x_5 - x_4) - L_5] = k_6[(L - x_5) - L_6]$$

As cinco equações formam um sistema tridiagonal. Na forma matricial, o sistema é:

$$\begin{bmatrix} k_1 + k_2 & -k_2 & 0 & 0 & 0 \\ -k_2 & k_2 + k_3 & -k_3 & 0 & 0 \\ 0 & -k_3 & k_3 + k_4 & -k_4 & 0 \\ 0 & 0 & -k_4 & k_4 + k_5 & -k_5 \\ 0 & 0 & 0 & -k_5 & k_5 + k_6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} k_1 L_1 - k_2 L_2 \\ k_2 L_2 - k_3 L_3 \\ k_3 L_3 - k_4 L_4 \\ k_4 L_4 - k_5 L_5 \\ k_5 L_5 + k_6 L - k_6 L_6 \end{bmatrix} \quad (4.66)$$

O sistema de Equações (4.66) é solucionado com o uso de uma função criada no MATLAB, chamada Tridiagonal, que é listada a seguir:

**Programa 4-9 Função definida pelo usuário. Solução de um sistema tridiagonal de equações.**

```
function x = Tridiagonal(A,B)
% A função soluciona um sistema tridiagonal de equações lineares [a][x]=[b]
% usando o algoritmo de Thomas.
% Variáveis de entrada:
```

```
% A Matriz de coeficientes.
% B Vetor coluna de constantes.
% Variável de saída:
% x Vetor coluna com a solução.

[nR, nC] = size(A);
for i = 1:nR
    d(i) = A(i,i); Define o vetor d com os elementos da diagonal.
end
for i = 1:nR - 1
    ad(i) = A(i,i + 1); Define o vetor ad com os elementos acima da diagonal.
end
for i = 2:nR
    bd(i) = A(i,i - 1); Define o vetor bd com os elementos abaixo da diagonal.
end
ad(1) = ad(1)/d(1);
B(1) = B(1)/d(1);
for i = 2:nR - 1
    ad(i) = ad(i)/(d(i) - bd(i)*ad(i - 1)); Passo 2.
    B(i) = (B(i) - bd(i)*B(i - 1))/(d(i) - bd(i)*ad(i - 1)); Passo 3.
end
B(nR) = (B(nR) - bd(nR)*B(nR - 1))/(d(nR) - bd(nR)*ad(nR - 1)); Passo 4.
x(nR,1) = B(nR);
for i = nR - 1:-1:1
    x(i,1) = B(i) - ad(i)*x(i + 1); Passo 5.
end
```

A função Tridiagonal é usada a seguir em um programa para resolver o sistema da Eq. (4.66).

% Exemplo 4-9

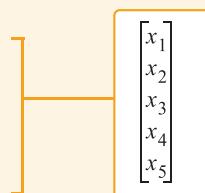
```
k1 = 8000; k2 = 9000; k3 = 15000; k4 = 12000; k5 = 10000; k6 = 18000;
L = 1.5; L1 = 0.18; L2 = 0.22; L3 = 0.26; L4 = 0.19; L5 = 0.15; L6 = 0.30;
a = [k1 + k2, -k2, 0, 0, 0; -k2, k2+k3, -k3, 0, 0; 0, -k3, k3+k4, -k4, 0
     0, 0, -k4, k4+k5, -k5; 0, 0, 0, -k5, k5+k6];
b = [k1*L1 - k2*L2; k2*L2 - k3*L3; k3*L3 - k4*L4; k4*L4 - k5*L5; k5*L5 + k6*L - k6*L6];
Xs = Tridiagonal(a,b)
```

Quando o arquivo é executado, a seguinte solução é mostrada na janela de comandos:

Xs =

```
0.2262
0.4872
0.7718
0.9926
1.1795
```

>>



## 4.10 ERRO, RESÍDUO, NORMAS E NÚMERO CONDIÇÃO

A solução numérica de um sistema de equações é raramente uma solução exata. Mesmo embora métodos diretos (Gauss, Gauss-Jordan, decomposição LU) possam ser exatos, eles não deixam de ser susceptíveis a erros de arredondamento quando implementados em um computador. Isso é especialmente verdadeiro no caso de grandes sistemas e também em sistemas mal condicionados (ver Seção 4.11). As soluções obtidas com métodos iterativos são aproximações por natureza. Esta seção descreve medidas que podem ser usadas para quantificar a precisão de uma solução numérica, ou para estimar a ordem de grandeza do erro.

### 4.10.1 Erro e resíduo

Se  $[x_{NS}]$  é uma solução numérica aproximada computada para um sistema de  $n$  equações  $[a][x] = [b]$  e  $[x_{TS}]$  é a solução exata, então o erro real é o vetor:

$$[e] = [x_{TS}] - [x_{NS}] \quad (4.67)$$

O erro real, no entanto, em geral não pode ser calculado porque a solução exata não é conhecida.

Uma medida alternativa da precisão de uma solução é o resíduo  $[r]$ , definido como:

$$[r] = [a][x_{TS}] - [a][x_{NS}] = [b] - [a][x_{NS}] \quad (4.68)$$

Colocando em palavras,  $[r]$  mede quanto bem o sistema de equações é satisfeito quando  $[X_{NS}]$  é substituído em  $[x]$  (isso é equivalente à tolerância em  $f(x)$  quando se considera a solução de uma única equação. Ver Eq. (3.5) na Seção 3.2). O vetor  $[r]$  tem  $n$  elementos, e se a solução numérica for próxima da solução real, então todos os elementos de  $[r]$  são pequenos. Deve-se lembrar que, na realidade,  $[r]$  não indica quanto pequenos são os erros na solução  $[x]$ .  $[r]$  mostra apenas quanto bem o lado direito das equações é satisfeito quando  $[X_{NS}]$  é substituído em  $[x]$  nas equações originais. Isso depende da ordem de grandeza dos elementos da matriz  $[a]$ . Conforme mostrado no Exemplo 4-10, é possível ter uma solução numérica aproximada apresentando um grande erro real mas um resíduo pequeno.

Uma estimativa mais precisa do erro em uma solução numérica pode ser obtida com o uso de grandezas que medem o tamanho de vetores e matrizes. No caso de números, é fácil determinar qual deles é maior ou menor a partir da comparação de seus valores absolutos. No entanto, é mais difícil medir o tamanho de vetores e matrizes. Isso é feito por uma grandeza chamada de **norma**, introduzida logo a seguir.

**Exemplo 4-10: Erro e resíduo**

A solução exata do sistema de equações:

$$1,02x_1 + 0,98x_2 = 2$$

$$0,98x_1 + 1,02x_2 = 2$$

$$\text{é } x_1 = x_2 = 1.$$

Calcule o erro real e o resíduo para as duas soluções aproximadas a seguir:

$$(a) \quad x_1 = 1,02, \quad x_2 = 1,02.$$

$$(b) \quad x_1 = 2, \quad x_2 = 0.$$

**SOLUÇÃO**

Na forma matricial, o sistema de equações original tem a forma  $[a][x] = [b]$ , onde  $[a] = \begin{bmatrix} 1,02 & 0,98 \\ 0,98 & 1,02 \end{bmatrix}$  e  $[b] = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ . A solução real é  $[x_{TS}] = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ .

O erro real e o resíduo são dados pelas Eqs. (4.67) e (4.68), respectivamente. A aplicação dessas equações nas duas soluções aproximadas resulta em:

(a) Neste caso,  $[x_{NS}] = \begin{bmatrix} 1,02 \\ 1,02 \end{bmatrix}$ . Conseqüentemente, o erro e o resíduo são:

$$[e] = [x_{TS}] - [x_{NS}] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1,02 \\ 1,02 \end{bmatrix} = \begin{bmatrix} -0,02 \\ -0,02 \end{bmatrix}$$

$$[r] = [b] - [a][x_{NS}] = [b] - [a][x_{NS}] = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1,02 & 0,98 \\ 0,98 & 1,02 \end{bmatrix} \begin{bmatrix} 1,02 \\ 1,02 \end{bmatrix} = \begin{bmatrix} -0,04 \\ -0,04 \end{bmatrix}$$

Neste caso, tanto o erro quanto o resíduo são pequenos.

(b) Neste caso,  $[x_{NS}] = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ . Conseqüentemente, o erro e o resíduo são:

$$[e] = [x_{TS}] - [x_{NS}] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$[r] = [b] - [a][x_{NS}] = [b] - [a][x_{NS}] = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1,02 & 0,98 \\ 0,98 & 1,02 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -0,04 \\ 0,04 \end{bmatrix},$$

Neste caso, o erro é grande mas o resíduo é pequeno.

Este exemplo mostra que um resíduo pequeno não necessariamente garante um erro pequeno. O fato de um resíduo pequeno resultar ou não em um erro pequeno depende da “magnitude” da matriz  $[a]$ .

#### 4.10.2 Normas e número condição

A **norma** é um número real atribuído a uma matriz ou a um vetor. Esse número satisfaz às quatro propriedades a seguir:

- (i) A norma de um vetor ou matriz, denotada por  $\|[a]\|$ , é uma grandeza positiva. Ela é igual a zero apenas se o objeto  $[a]$  for nulo. Em outras palavras,  $\|[a]\| \geq 0$  e  $\|[a]\| = 0$  apenas se  $[a] = 0$ . Isso significa que todos os vetores e matrizes, exceto o vetor zero e a matriz zero, têm tamanho positivo.
- (ii) Para todos os números  $\alpha$ ,  $\|\alpha[a]\| = |\alpha| \|[a]\|$ . Isso quer dizer que dois objetos  $[a]$  e  $[-a]$  possuem o mesmo tamanho e que o tamanho de  $[10a]$  é 10 vezes maior que o tamanho de  $[a]$ .
- (iii) Para matrizes e vetores,  $\|[a][x]\| \leq \|[a]\| \|[x]\|$ , o que significa que a norma do produto de duas matrizes é menor ou igual ao produto das normas de cada matriz.
- (iv) Para quaisquer dois vetores ou matrizes  $[a]$  e  $[b]$ :

$$\|[a + b]\| \leq \|[a]\| + \|[b]\| \quad (4.69)$$

Este enunciado é conhecido como **desigualdade triangular** porque, para os vetores  $[a]$  e  $[b]$ , ele diz que a soma dos comprimentos dos dois lados de um triângulo nunca pode ser menor que o comprimento do terceiro lado.

Qualquer norma de vetor ou matriz deve satisfazer às quatro propriedades acima para ser qualificada como uma medida de seu “tamanho”. Diferentes normas de vetores ou matrizes são descritas a seguir.

#### Normas de vetores

Para um dado vetor  $[v]$  com  $n$  elementos, a **norma infinita**, escrita como  $\|v\|_\infty$ , é definida como:

$$\|v\|_\infty = \max_{1 \leq i \leq n} |v_i| \quad (4.70)$$

Colocando em palavras,  $\|v\|_\infty$  é um número igual ao elemento  $v_i$  com o maior valor absoluto.

A **norma 1**, escrita como  $\|v\|_1$ , é definida como:

$$\|v\|_1 = \sum_{i=1}^n |v_i| \quad (4.71)$$

Colocando em palavras,  $\|v\|_1$  é a soma dos valores absolutos dos elementos do vetor.

A **norma 2 Euclidiana**, escrita como  $\|v\|_2$ , é definida como:

$$\|v\|_2 = \left( \sum_{i=1}^n v_i^2 \right)^{1/2} \quad (4.72)$$

Colocando em palavras,  $\|v\|_2$  é a raiz quadrada da soma dos quadrados dos elementos. Ela é também chamada de módulo do vetor  $[v]$ .

#### Normas de matrizes

A **norma infinita** de uma matriz é dada por:

$$\|[a]\|_{\infty} = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| \quad (4.73)$$

Colocando em palavras, na norma infinita de matrizes é feita a soma dos valores absolutos dos elementos de cada linha da matriz. O valor da maior soma é atribuído a  $\|[a]\|_{\infty}$ .

A **norma 1** de uma matriz é calculada como:

$$\|[a]\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| \quad (4.74)$$

Esta norma é similar à norma infinita, exceto pelo fato de que a soma dos valores absolutos dos elementos é feita para cada coluna, e o valor da maior soma é atribuído a  $\|[a]\|_1$ .

A **norma 2** de uma matriz é calculada como uma norma espectral:

$$\|[a]\|_2 = \max \left( \frac{\|[a][v]\|}{\|[v]\|} \right) \quad (4.75)$$

onde  $[v]$  é um autovetor da matriz  $[a]$  correspondente a um autovalor  $\lambda$  (autovetores e autovalores são introduzidos na Seção 4.12). O cálculo da norma 2 de uma matriz no MATLAB usa uma técnica chamada de decomposição de valor singular, na qual a matriz  $[a]$  é fatorada na forma  $[a] = [u][d][v]$ , onde  $[u]$  e  $[v]$  são matrizes ortogonais (matrizes especiais que satisfazem à propriedade  $[u]^{-1} = [u]^T$ ), e  $[d]$  é uma matriz diagonal. O maior valor dos elementos diagonais de  $[d]$  é usado como a norma 2 da matriz  $[a]$ .

A **norma Euclidiana** de uma matriz  $[a] m \times n$  (que é diferente da norma 2 de uma matriz) é dada por:

$$\|[a]\|_{Euclidiana} = \left( \sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2} \quad (4.76)$$

### Uso de normas para determinar limites no erro de soluções numéricas

A partir das Eqs. (4.67) e (4.68), o resíduo pode ser escrito em termos do erro  $[e]$  como:

$$[r] = [a][x_{TS}] - [a][x_{NS}] = [a]([x_{TS}] - [x_{NS}]) = [a][e] \quad (4.77)$$

Se a matriz  $[a]$  tiver inversa (do contrário o sistema de equações não tem uma solução), o erro pode ser expresso como:

$$[e] = [a]^{-1}[r] \quad (4.78)$$

A aplicação da propriedade (iii) da norma matricial na Eq. (4.78) resulta em:

$$\|[e]\| = \|[a]^{-1}[r]\| \leq \|[a]^{-1}\| \|[r]\| \quad (4.79)$$

Da Eq. (4.77), o resíduo  $[r]$  é:

$$[r] = [a][e] \quad (4.80)$$

A aplicação da propriedade (iii) das normas matriciais na Eq. (4.80) resulta em:

$$\|[r]\| = \|[a][e]\| \leq \|[a]\| \|[e]\| \quad (4.81)$$

A última equação pode ser reescrita como:

$$\frac{\|[r]\|}{\|[a]\|} \leq \|[e]\| \quad (4.82)$$

As Eqs. (4.79) e (4.82) podem ser combinadas e escritas na forma:

$$\frac{\|[r]\|}{\|[a]\|} \leq \|[e]\| = \| [a]^{-1} [r] \| \leq \| [a]^{-1} \| \|[r]\| \quad (4.83)$$

Para usar a Eq. (4.83), são definidas duas novas grandezas. Uma delas é o **erro relativo**, definido como  $\|[e]\| / \|[x_{TS}]\|$ , e a segunda é o **resíduo relativo**, definido como  $\|[r]\| / \|[b]\|$ . Em uma solução numérica aproximada, o resíduo pode ser calculado a partir da Eq. (4.68). A partir do resíduo, a Eq. (4.83) pode ser usada para determinar os limites superior e inferior do erro relativo em termos do resíduo relativo. Isso é feito com a divisão da Eq. (4.83) por  $\|[x_{TS}]\|$  e escrevendo a equação na forma:

$$\frac{1}{\|[a]\|} \frac{\|[b]\|}{\|[x_{TS}]\|} \frac{\|[r]\|}{\|[b]\|} \leq \frac{\|[e]\|}{\|[x_{TS}]\|} \leq \| [a]^{-1} \| \frac{\|[b]\|}{\|[x_{TS}]\|} \frac{\|[r]\|}{\|[b]\|} \quad (4.84)$$

Como  $[a][x_{TS}] = [b]$ , a propriedade (iii) das normas matriciais leva a  $\|[b]\| \leq \|[a]\| \|[x_{TS}]\|$  ou  $\frac{\|[b]\|}{\|[x_{TS}]\|} \leq \|[a]\|$ , e isso significa que  $\frac{\|[b]\|}{\|[x_{TS}]\|}$  pode ser substituído por  $\|[a]\|$  no lado direito da Eq. (4.84). De forma similar, como  $[x_{TS}] = [a]^{-1}[b]$ , a propriedade (iii) das normas matriciais leva a  $\|[x_{TS}]\| \leq \| [a]^{-1} \| \|[b]\|$  ou

$\frac{1}{\|[a]^{-1}\|} \leq \frac{\|[b]\|}{\|[x_{TS}]\|}$ , e isso significa que  $\frac{1}{\|[a]^{-1}\|}$  pode ser substituído por  $\frac{\|[b]\|}{\|[x_{TS}]\|}$  no

lado esquerdo da Eq. (4.84). Com essas substituições, a Eq. (4.84) vira:

$$\frac{1}{\|[a]\| \| [a]^{-1} \|} \frac{\|[r]\|}{\|[b]\|} \leq \frac{\|[e]\|}{\|[x_{TS}]\|} \leq \| [a]^{-1} \| \| [a] \| \frac{\|[r]\|}{\|[b]\|} \quad (4.85)$$

A Eq. (4.85) é o principal resultado desta seção. Ela permite delimitar o erro na solução numérica de um sistema de equações. A Eq. (4.85) diz que o erro

relativo real  $\frac{\|[e]\|}{\|[x_{TS}]\|}$  (que é desconhecido) está limitado entre  $\frac{1}{\|[a]\| \| [a]^{-1} \|}$  vezes o resíduo relativo  $\frac{\|[r]\|}{\|[b]\|}$  (limite inferior), e  $\| [a]^{-1} \| \| [a] \|$  vezes o resíduo relativo

(limite superior). O resíduo relativo pode ser calculado a partir da solução numérica aproximada, e com isso pode-se limitar o erro relativo real se a grandeza  $\|[a]\| \| [a]^{-1} \|$  (chamada de número condição) puder ser calculada.

### Número condição

O número  $\|[a]\|\|[a]^{-1}\|$  é chamado de **número condição** da matriz  $[a]$ . Ele é escrito como:

$$\text{Cond}[a] = \|[a]\|\|[a]^{-1}\| \quad (4.86)$$

- O número condição da matriz identidade é igual a 1. O número condição de qualquer outra matriz é maior ou igual a 1.
- Se o número condição for aproximadamente igual a 1, então o erro relativo real tem a mesma ordem de grandeza do resíduo relativo.
- Se o número condição for muito maior que 1, então um pequeno resíduo relativo não necessariamente implica um pequeno erro relativo real.
- Para uma dada matriz, o valor do número condição depende da norma matricial utilizada.
- A inversa de uma matriz deve ser conhecida para que o número condição seja calculado.

O Exemplo 4-11 ilustra o cálculo do erro, do resíduo, das normas e do número condição.

#### Exemplo 4-11: Cálculo do erro, resíduo, normas e número condição

Considere o seguinte conjunto de quatro equações (o mesmo resolvido no Exemplo 4-8).

$$\begin{aligned} 9x_1 - 2x_2 + 3x_3 + 2x_4 &= 54,5 \\ 2x_1 + 8x_2 - 2x_3 + 3x_4 &= -14 \\ -3x_1 + 2x_2 + 11x_3 - 4x_4 &= 12,5 \\ -2x_1 + 3x_2 + 2x_3 + 10x_4 &= -21 \end{aligned}$$

A solução exata do sistema é:  $x_1 = 5$ ,  $x_2 = -2$ ,  $x_3 = 2,5$  e  $x_4 = -1$ . Quando esse sistema foi resolvido no Exemplo 4-8 com o método iterativo de Gauss-Seidel, a solução numérica na sexta iteração era:  $x_1 = 4,98805$ ,  $x_2 = -1,99511$ ,  $x_3 = 2,49806$  e  $x_4 = -1,00347$ .

- Determine o erro real  $[e]$  e o resíduo  $[r]$ .
- Determine as normas infinitas da solução exata,  $[x_{TS}]$ , do erro,  $[e]$ , do resíduo,  $[r]$  e do vetor  $[b]$ .
- Determine a inversa de  $[a]$ , a norma infinita de  $[a]$  e  $[a]^{-1}$  e o número condição da matriz  $[a]$ .
- Substitua as grandezas das letras (b) e (c) na Eq. (4.85) e discuta os resultados.

### SOLUÇÃO

Primeiramente, as equações são escritas na forma matricial:

$$\begin{bmatrix} 9 & -2 & 3 & 2 \\ 2 & 8 & -2 & 3 \\ -3 & 2 & 11 & -4 \\ -2 & 3 & 2 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 54,5 \\ -14 \\ 12,5 \\ -21 \end{bmatrix}$$

- (a) A solução real é  $x_{TS} = \begin{bmatrix} 5 \\ -2 \\ 2,5 \\ -1 \end{bmatrix}$  e a solução numérica aproximada é  $x_{NS} = \begin{bmatrix} 4,98805 \\ -1,99511 \\ 2,49806 \\ -1,00347 \end{bmatrix}$ .

$$\text{O erro é então: } [e] = [x_{TS}] - [x_{NS}] = \begin{bmatrix} 5 \\ -2 \\ 2,5 \\ -1 \end{bmatrix} - \begin{bmatrix} 4,98805 \\ -1,99511 \\ 2,49806 \\ -1,00347 \end{bmatrix} = \begin{bmatrix} 0,0119 \\ -0,0049 \\ 0,0019 \\ 0,0035 \end{bmatrix}.$$

O resíduo é dado pela Eq. (4.77),  $[r] = [a][e]$ . Ele é calculado na janela de comandos do MATLAB:

```
>> a = [9 -2 3 2; 2 8 -2 3; -3 2 11 -4; -2 3 2 10];
```

```
>> e = [0.0119; -0.0049; 0.0019; 0.0035];
```

```
>> r = a*e
```

```
r =
```

```
0.130090000000000  
-0.008690000000000  
-0.038170000000000  
0.000010000000000
```

(b) A norma infinita de um vetor é definida na Eq. (4.70):  $\|v\|_\infty = \max_{1 \leq i \leq n} |v_i|$ . O uso dessa equação para

calcular a norma infinita da solução exata, do resíduo e do vetor  $[b]$  resulta em:

$$\|x_{TS}\|_\infty = \max_{1 \leq i \leq 4} |x_{TS,i}| = \max[|5|, |-2|, |2,5|, |-1|] = 5$$

$$\|e\|_\infty = \max_{1 \leq i \leq 4} |e_i| = \max[|0,0119|, |-0,0049|, |0,0019|, |0,0035|] = 0,0119$$

$$\|r\|_\infty = \max_{1 \leq i \leq 4} |r_i| = \max[|0,13009|, |-0,00869|, |-0,03817|, 0,00001] = 0,13009$$

$$\|b\|_\infty = \max_{1 \leq i \leq 4} |b_i| = \max[|54,5|, |-14|, |12,5|, |-21|] = 54,5$$

(c) A inversa de  $[a]$  é calculada usando a função `inv` na janela de comandos do MATLAB:

```
>> aINV = inv(a)
```

```
aINV =
```

```
0.0910 0.0386 -0.0116 -0.0344  
-0.0206 0.1194 0.0308 -0.0194  
0.0349 -0.0200 0.0727 0.0281  
0.0174 -0.0241 -0.0261 0.0933
```

As normas infinitas de  $[a]$  e  $[a]^{-1}$  são calculadas com a Eq. (4.73)  $\|[a]\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$ :

$$\|[a]\|_\infty = \max_{1 \leq i \leq 4} \sum_{j=1}^n |a_{ij}| = \max[|9| + |-2| + |3| + |2|, |2| + |8| + |-2| + |3|, |-3| + |2| + |11| + |-4|, |9| + |-2| + |3| + |2|] = 20$$

$$\|[a]\|_\infty = \max[16, 15, 20, 16] = 20$$

$$\|[a]^{-1}\|_\infty = \max_{1 \leq i \leq 4} \sum_{j=1}^n |a_{ij}^{-1}| = \max[|0,091| + |0,0386| + |-0,0116| + |-0,0344|, |-0,0206| + |0,1194| + |0,0308| + |-0,0194|, |0,0349| + |-0,02| + |0,0727| + |0,0281|, |0,0174| + |-0,0241| + |-0,0261| + |0,0933|] = 0,1902$$

$$\|[a]^{-1}\|_\infty = \max[0,1756, 0,1902, 0,1557, 0,1609] = 0,1902$$

O número condição da matriz  $[a]$  é calculado usando a Eq. (4.86):

$$\text{Cond}[a] = \|[a]\| \|[a]^{-1}\| = 20 \cdot 0,1902 = 3,804$$

(d) A substituição de todas as variáveis calculadas nas letras (b) e (c) na Eq. (4.85) resulta em:

$$\frac{1}{\|[a]\|\|[a]^{-1}\|\|[b]\|} \frac{\|[r]\|}{\|[e]\|} \leq \frac{\|[e]\|}{\|[x_{TS}]\|} \leq \|[a]^{-1}\| \|[a]\| \frac{\|[r]\|}{\|[b]\|}$$

$$\frac{1}{3,804} \frac{0,13009}{54,5} \leq \frac{\|[e]\|}{\|[x_{TS}]\|} \leq 3,804 \frac{0,13009}{54,5}$$

$$\frac{1}{3,804} 0,002387 \leq \frac{\|[e]\|}{\|[x_{TS}]\|} \leq 3,804 \cdot 0,002387, \quad \text{ou} \quad 6,275 \times 10^{-4} \leq \frac{\|[e]\|}{\|[x_{TS}]\|} \leq 0,00908$$

Esses resultados indicam que a ordem de grandeza do erro relativo está entre  $6,275 \times 10^{-4}$  e 0,00908.

Neste problema, o erro relativo real pode ser calculado porque a solução exata é conhecida.

O erro relativo real é:

$$\frac{\|[e]\|}{\|[x_{TS}]\|} = \frac{0,0119}{5} = 0,00238, \quad \text{que está dentro dos limites calculados pela Eq. (4.85).}$$

## 4.11 SISTEMAS MAL CONDICIONADOS

Um sistema de equações mal condicionado é aquele no qual pequenas variações nos coeficientes causam grandes mudanças na solução. A matriz de coeficientes de sistemas mal condicionados tem, geralmente, um número condição significativamente maior que 1. Por exemplo, considere o sistema:

$$\begin{aligned} 6x_1 - 2x_2 &= 10 \\ 11,5x_1 - 3,85x_2 &= 17 \end{aligned} \tag{4.87}$$

A solução desse sistema é:

$$x_1 = \frac{a_{12}b_2 - a_{22}b_1}{a_{12}a_{21} - a_{11}a_{22}} = \frac{-2 \cdot 17 - (-3,85 \cdot 10)}{-2 \cdot 11,5 - (6 \cdot -3,85)} = \frac{4,5}{0,1} = 45$$

$$x_2 = \frac{a_{21}b_1 - a_{11}b_2}{a_{12}a_{21} - a_{11}a_{22}} = \frac{11,5 \cdot 10 - (6 \cdot 17)}{-2 \cdot 11,5 - (6 \cdot -3,85)} = \frac{13}{0,1} = 130$$

Se uma pequena mudança for feita no sistema com a troca de  $a_{22}$  por 3,84:

$$\begin{aligned} 6x_1 - 2x_2 &= 10 \\ 11,5x_1 - 3,84x_2 &= 17 \end{aligned} \tag{4.88}$$

então a solução é:

$$x_1 = \frac{a_{12}b_2 - a_{22}b_1}{a_{12}a_{21} - a_{11}a_{22}} = \frac{-2 \cdot 17 - (-3,84 \cdot 10)}{-2 \cdot 11,5 - (6 \cdot -3,84)} = \frac{4,4}{0,04} = 110$$

$$x_2 = \frac{a_{21}b_1 - a_{11}b_2}{a_{12}a_{21} - a_{11}a_{22}} = \frac{11,5 \cdot 10 - (6 \cdot 17)}{-2 \cdot 11,5 - (6 \cdot -3,84)} = \frac{13}{0,04} = 325$$

Pode-se observar que há uma diferença muito grande entre as soluções dos dois sistemas. Um exame cuidadoso das soluções das Eqs. (4.87) e (4.88) mostra que o numerador da equação referente a  $x_2$  em ambas as soluções é o mesmo, e que há apenas uma pequena diferença no numerador da equação referente a  $x_1$ . Ao mesmo tempo, há uma grande diferença (um fator de 2,5) entre os denominadores

das duas equações. O denominador de ambas as equações são os determinantes das matrizes de coeficientes  $[a]$ .

O fato de o sistema da Eq. (4.87) ser mal condicionado fica evidente a partir do valor do número condição. Para esse sistema:

$$[a] = \begin{bmatrix} 6 & -2 \\ 11,5 & -3,85 \end{bmatrix} \quad \text{e} \quad [a]^{-1} = \begin{bmatrix} 38,5 & -20 \\ 115 & -60 \end{bmatrix}$$

Usando a norma infinita dada pela Eq. (4.73), o número condição do sistema é:

$$\text{Cond}[a] = \| [a] \| \| [a]^{-1} \| = 15,35 \cdot 175 = 2686,25$$

Usando a norma 1, o número condição do sistema é:

$$\text{Cond}[a] = \| [a] \| \| [a]^{-1} \| = 17,5 \cdot 153,5 = 2686,25$$

Usando a norma 2, o número condição do sistema é (as normas foram calculadas no MATLAB utilizando a função residente `norm(a, 2)`):

$$\text{Cond}[a] = \| [a] \| \| [a]^{-1} \| = 13,6774 \cdot 136,774 = 1870,7$$

Estes resultados mostram que, com o uso de qualquer norma, o número condição da matriz de coeficientes do sistema da Eq. (4.87) é muito maior que 1. Isso significa que o sistema tem tudo para ser mal condicionado.

Quando se resolve numericamente um sistema de equações mal condicionado, há uma alta probabilidade de a solução obtida apresentar um erro grande ou mesmo de não se obter uma solução. Em geral, é difícil quantificar o valor do número condição que identifica precisamente um sistema mal condicionado. Isso depende da precisão do computador utilizado e de outros fatores. Logo, na prática, deve-se preocupar apenas com o fato de o número condição ser muito maior que 1 ou não, e não com o seu valor exato. Além disso, de qualquer forma, pode não ser possível calcular o determinante e o número condição de um sistema mal condicionado, porque as operações matemáticas realizadas nestes cálculos são similares às operações necessárias durante a solução do sistema.

## 4.12 AUTOVALORES E AUTOVETORES

Para uma dada matriz  $[a]$  ( $n \times n$ ), o número  $\lambda$  é um autovalor da matriz se:

$$[a][u] = \lambda[u] \tag{4.89}$$

O vetor  $[u]$  é um vetor coluna com  $n$  elementos chamado de autovetor associado ao autovalor  $\lambda$ .

A Eq. (4.89) pode ser vista de uma forma mais geral. A multiplicação  $[a]$   $[u]$  é uma operação matemática e pode ser vista como a matriz  $[a]$  operando no operador  $[u]$ . Com essa terminologia, a Eq. (4.89) pode ser lida como: “[ $a$ ] opera em  $[u]$  para resultar em  $\lambda$  vezes  $[u]$ ”, e a Eq. (4.89) pode ser generalizada para qualquer operação matemática como:

$$Lu = \lambda u \tag{4.90}$$

onde  $L$  é um operador que pode representar multiplicação por uma matriz, diferenciação, integração, e assim por diante,  $u$  é um vetor ou uma função, e  $\lambda$  é uma constante escalar. Por exemplo, se  $L$  representar a derivada segunda em relação a  $x$ ,  $y$  for uma função de  $x$ , e  $k$  for uma constante, então a Eq. (4.90) pode ter a forma:

$$\frac{d^2y}{dx^2} = k^2 y \quad (4.91)$$

A Eq. (4.90) é a declaração geral de um **problema de autovalores**, onde  $\lambda$  é chamado de **autovalor** associado ao operador  $L$  e  $u$  é o **autovetor** ou **autofunção** correspondente ao autovalor  $\lambda$  e ao operador  $L$ .

Autovalores e autovetores aparecem em métodos numéricos e têm importância especial na ciência e na engenharia. Por exemplo, no estudo de vibrações, os autovalores representam as freqüências naturais de um sistema ou de um componente, e os autovetores representam os modos dessas vibrações. É importante identificar tais freqüências naturais porque, quando o sistema é submetido a forças periódicas nessas freqüências ou em freqüências próximas, a ressonância pode levar à falha da estrutura ao amplificar o seu movimento. Na mecânica dos materiais, os esforços principais são os autovalores da matriz de tensões, e as direções principais são as direções dos autovetores associados. Na mecânica quântica, autovalores são especialmente importantes. Na formulação de Heisenberg na mecânica quântica, existe um operador  $L$  correspondente a todas as grandezas observáveis (isto é, quaisquer grandezas que podem ser medidas ou inferidas experimentalmente, como posição, velocidade ou energia). Este operador  $L$  atua em um operando  $\Psi$  chamado de função de onda. Se o resultado for proporcional à função de onda, isto é, se  $L\Psi = c\Psi$ , então o valor da grandeza observável  $c$  corresponde ao autovalor<sup>1</sup> e é chamado de *certo* (isto é, pode ser conhecido de forma muito precisa). Em outras palavras, os autovalores  $c$  associados à grandeza observável são os valores dessa grandeza que têm uma probabilidade de ocorrência diferente de zero (ou seja, de a grandeza ser observada). Exemplos de tais operadores na

mecânica quântica são:  $\frac{i\hbar}{2\pi}\frac{\partial\Psi}{\partial t} = E\Psi$ , onde  $\frac{i\hbar}{2\pi}\frac{\partial}{\partial t}$  é o operador energia e  $E$  é a energia;  $-i\frac{\hbar}{2\pi}\vec{\nabla}\Psi = \vec{p}\Psi$ , onde  $-i\frac{\hbar}{2\pi}\vec{\nabla}$  é o operador momento,  $\vec{p}$  é o momento linear,  $i = \sqrt{-1}$  e  $\hbar$  é a constante de Planck. Os autovetores, também conhecidos como autoestados, representam um dos muitos estados no qual um objeto ou sistema pode existir dado um autovalor particular.

Há um elo entre problemas de autovalores envolvendo equações diferenciais e problemas de autovalores envolvendo matrizes, que são o foco desta seção. A solução numérica de problemas de autovalores envolvendo equações diferenciais ordinárias (EDO) resulta em sistemas de equações simultâneas com a forma (4.89). Em outras palavras, a determinação numérica dos autovalores em um

<sup>1</sup> A palavra autovalor deriva da palavra alemã *eigenwert*, traduzida para o inglês como *eigenvalue*, que significa “valor próprio ou característico”.

problema envolvendo uma EDO reduz-se à determinação dos autovalores de uma matriz associada  $[a]$ , resultando em um problema na forma da Eq. (4.89).

Além da importância física dos autovalores na ciência e na engenharia, os autovalores de uma matriz também podem fornecer informações importantes sobre as propriedades dessa matriz em cálculos numéricos que a envolvam. A Seção 4.7 mostrou que os métodos iterativos de Jacobi e de Gauss-Seidel podem ser escritos na forma:

$$x_i^{(k+1)} = b'_i - [a]x_i^{(k)}$$

No final, a convergência ou não desses métodos iterativos depende dos autovetores da matriz  $[a]$ . Além disso, a rapidez na qual as iterações convergem depende da ordem de grandeza dos autovalores de  $[a]$ .

### Determinação de autovalores e autovetores

A determinação dos autovalores de uma matriz a partir da Eq. (4.89) é realizada escrevendo essa equação na forma:

$$[a - \lambda I][u] = 0 \quad (4.92)$$

onde  $[I]$  é uma matriz identidade com as mesmas dimensões de  $[a]$ . Escrevendo tal equação nesta forma homogênea, pode-se ver que, se a matriz  $[a - \lambda I]$  for não-singular (isto é, se ela possuir inversa), a multiplicação de ambos os lados de (4.92) por  $[a - \lambda I]^{-1}$  leva à solução trivial  $[u] = 0$ . Por outro lado, se  $[a - \lambda I]$  for singular, isto é, se ela não possuir inversa, então é possível encontrar uma solução não-trivial para  $[u]$ . Outra forma de enunciar esse critério se baseia na regra de Cramer (ver Capítulo 2): a matriz  $[a - \lambda I]$  é singular se o seu determinante for igual a zero:

$$\det[a - \lambda I] = 0 \quad (4.93)$$

A Eq. (4.93) é chamada de equação característica. Para uma dada matriz  $[a]$ , ela leva a uma equação polinomial em função de  $\lambda$  cujas raízes são os autovalores. Uma vez conhecidos os autovalores, os autovetores podem ser determinados. Isso é feito com a substituição dos autovalores (um de cada vez) na Eq. (4.89) e com a solução da equação para  $[u]$ . Para uma matriz  $[a]$  pequena ( $(2 \times 2)$  ou  $(3 \times 3)$ ), os autovalores podem ser determinados diretamente a partir do cálculo do determinante e das raízes da equação característica. Isso é mostrado no Exemplo 4-12, onde a abordagem do problema de autovalor é usada no cálculo dos momentos de inércia principais e das direções dos eixos principais de uma área com seção reta assimétrica.

A determinação dos autovalores de matrizes maiores é mais difícil. No entanto, existem vários métodos numéricos para a solução de problemas com tais características. Dois deles, o método da potência e o método da fatoração QR, são descritos a seguir.

**Exemplo 4-12: Momentos de inércia principais**

Determine os momentos de inércia principais e a orientação dos eixos de inércia principais da seção reta mostrada.

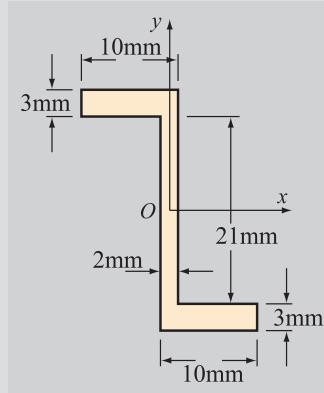
Os momentos de inércia  $I_x$ ,  $I_y$  e o produto de inércia  $I_{xy}$  são:

$$I_x = 10228,5 \text{ mm}^4, \quad I_y = 1307,34 \text{ mm}^4 \quad \text{e} \quad I_{xy} = -2880 \text{ mm}^4$$

**SOLUÇÃO**

Na forma matricial, a matriz bidimensional que representa o momento de inércia é dada por:

$$I_{Iner} = \begin{bmatrix} I_x & -I_{xy} \\ -I_{xy} & I_y \end{bmatrix} = \begin{bmatrix} 10228,5 & 2880 \\ 2880 & 1307,34 \end{bmatrix} \quad (4.94)$$



Os momentos de inércia principais e a orientação dos eixos de inércia principais podem ser calculados com a solução do seguinte problema de autovalores:

$$[I_{Iner}][u] = \lambda[u] \quad (4.95)$$

onde os autovalores  $\lambda$  são os momentos de inércia principais e os autovetores  $[u]$  são os vetores unitários na direção dos eixos de inércia principais. Os autovalores são determinados com o cálculo do determinante expresso pela Eq. (4.93):

$$\det[I_{Iner} - \lambda I] = 0 \quad (4.96)$$

$$\det \begin{bmatrix} (10228,5 - \lambda) & 2880 \\ 2880 & (1307,34 - \lambda) \end{bmatrix} = 0 \quad (4.97)$$

A equação polinomial em função de  $\lambda$  é:

$$(10228,5 - \lambda)(1307,34 - \lambda) - 2880^2 = 0 \quad \text{ou} \quad \lambda^2 - 11535,84\lambda + 5077727,19 = 0 \quad (4.98)$$

As soluções da equação polinomial quadrática são os autovalores  $\lambda_1 = 11077,46 \text{ mm}^4$  e  $\lambda_2 = 458,38 \text{ mm}^4$ , que são os momentos de inércia principais.

Os autovetores correspondentes a cada um dos autovalores são calculados com a substituição dos autovalores na Eq. (4.95). Para o primeiro autovetor  $u^{(1)}$ :

$$\begin{bmatrix} 10228,5 & 2880 \\ 2880 & 1307,34 \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{bmatrix} = 11077,46 \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{bmatrix} \quad \text{ou} \quad \begin{bmatrix} -848,96 & 2880 \\ 2880 & -9770,12 \end{bmatrix} \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.99)$$

As duas Eqs. (4.99) levam a  $u_2^{(1)} = 0,29478u_1^{(1)}$ . Usando a condição adicional de que o autovetor neste problema é um vetor unitário,  $(u_1^{(1)})^2 + (u_2^{(1)})^2 = 1$ , o autovetor associado ao primeiro autovetor  $\lambda_1 = 11077,46$  é  $u^{(1)} = 0,95919i + 0,28275j$ .

Para o segundo autovetor  $u^{(2)}$ :

$$\begin{bmatrix} 10228,5 & 2880 \\ 2880 & 1307,34 \end{bmatrix} \begin{bmatrix} u_1^{(2)} \\ u_2^{(2)} \end{bmatrix} = 458,38 \begin{bmatrix} u_1^{(2)} \\ u_2^{(2)} \end{bmatrix} \quad \text{ou} \quad \begin{bmatrix} 9770,12 & 2880 \\ 2880 & 848,96 \end{bmatrix} \begin{bmatrix} u_1^{(2)} \\ u_2^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.100)$$

As duas Eqs. (4.100) levam a  $u_2^{(2)} = -3,3924u_1^{(2)}$ . Usando a condição adicional de que o autovetor é um vetor unitário,  $(u_1^{(2)})^2 + (u_2^{(2)})^2 = 1$ , o autovetor associado ao segundo autovalor  $\lambda_2 = 458,38$  é determinado como sendo  $u^{(1)} = -0,28275i + 0,95919j$ .

### 4.12.1 O método da potência básico

O método da potência é um procedimento iterativo usado para determinar o maior autovalor real de uma matriz e seu correspondente autovetor. Considere uma matriz  $[a]$  ( $n \times n$ ) com  $n$  autovalores reais e distintos  $\lambda_1, \lambda_2, \dots, \lambda_n$  e  $n$  autovetores associados  $[u]_1, [u]_2, \dots, [u]_n$ . Os autovalores são numerados do maior para o menor de acordo com o critério:

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| \quad (4.101)$$

Como os autovetores são linearmente independentes, eles formam uma base. Isso significa que qualquer vetor pertencente ao mesmo espaço (isto é, grupo) dos autovetores pode ser escrito como uma combinação linear dos vetores base. Suponha que  $[x]$  seja um vetor coluna pertencente ao mesmo espaço dos autovetores  $[u]_1, [u]_2, \dots, [u]_n$ . Esse vetor  $[x]$  pode ser expresso como uma combinação dos autovetores:

$$[x] = c_1[u]_1 + c_2[u]_2 + \dots + c_n[u]_n \quad (4.102)$$

onde os escalares  $c_i \neq 0$  são constantes. Seja  $[x]_1 = [x]$ . A multiplicação da Eq. (4.102) por  $[a]$  resulta em:

$$[a][x]_1 = c_1[a][u]_1 + c_2[a][u]_2 + \dots + c_n[a][u]_n = \lambda_1 c_1[x]_2 \quad (4.103)$$

onde  $[x]_2 = [u]_1 + \frac{c_2\lambda_2}{c_1\lambda_1}[u]_2 + \dots + \frac{c_n\lambda_n}{c_1\lambda_1}[u]_n$ . Como os termos  $[u]_i$  são os autovetores,  $[a][u]_i = \lambda_i[u]_i$ , e com isso:

$$\begin{aligned} [a][x]_2 &= \lambda_1[u]_1 + \frac{c_2\lambda_2^2}{c_1\lambda_1}[u]_2 + \dots + \frac{c_n\lambda_n^2}{c_1\lambda_1}[u]_n \\ &= \lambda_1 \left\{ [u]_1 + \frac{c_2\lambda_2^2}{c_1\lambda_1^2}[u]_2 + \dots + \frac{c_n\lambda_n^2}{c_1\lambda_1^2}[u]_n \right\} = \lambda_1[x]_3 \end{aligned} \quad (4.104)$$

onde  $[x]_3 = [u]_1 + \frac{c_2\lambda_2^2}{c_1\lambda_1^2}[u]_2 + \dots + \frac{c_n\lambda_n^2}{c_1\lambda_1^2}[u]_n$ . Multiplicando a Eq. (4.104) novamente por  $[a]$  e usando  $[a][u]_i = \lambda_i[u]_i$ , obtém-se:

$$[a][x]_3 = \lambda_1[u]_1 + \frac{c_2\lambda_2^3}{c_1\lambda_1^2}[u]_2 + \dots + \frac{c_n\lambda_n^3}{c_1\lambda_1^2}[u]_n = \lambda_1[x]_4 \quad (4.105)$$

onde  $[x]_4 = [u]_1 + \frac{c_2\lambda_2^3}{c_1\lambda_1^3}[u]_2 + \dots + \frac{c_n\lambda_n^3}{c_1\lambda_1^3}[u]_n$ .

Pode-se ver que cada iteração sucessiva resulta em:

$$[x]_{k+1} = [u]_1 + \frac{c_2 \lambda_2^k}{c_1 \lambda_1^k} [u]_2 + \dots + \frac{c_n \lambda_n^k}{c_1 \lambda_1^k} [u]_n \quad (4.106)$$

Lembre-se que  $\lambda_1$  é o maior autovalor (ver Eq. (4.101)), o que significa que  $\frac{\lambda_i}{\lambda_1} < 1$  para todo  $i > 1$ . Logo, quando  $k$  é suficientemente grande, todos os termos que contêm  $\left(\frac{\lambda_i}{\lambda_1}\right)^k$  no lado direito da Eq. (4.106) podem ser desprezados se comparados com o termo  $[u]_1$ , e assim:

$$[a][x]_{k+1} \rightarrow \lambda_1 [u]_1 \text{ à medida que } k \rightarrow \infty \text{ e } [x]_{k+1} \rightarrow [u]_1 \quad (4.107)$$

A Eq. (4.107) mostra que o vetor  $[x]_k$  obtido a partir da Eq. (4.106) é igual a  $[u]_1$  (o autovetor). Quando se implementa o método da potência, o vetor  $[x]_k$  é normalizado em cada passo com a divisão de seus elementos pelo valor do maior elemento (ver Eqs. (4.102) a (4.106)). Isso faz com que o maior elemento do vetor seja igual a 1. É por causa desse ajuste de escala em cada passo que o método da potência fornece simultaneamente o autovalor e o autovetor associado.

Um procedimento numérico para a determinação do maior autovalor de uma matriz  $[a]$  de ordem  $(n \times n)$  utilizando o método da potência é dado no algoritmo a seguir.

### Algoritmo para o método da potência

1. Comece com um vetor coluna  $[x]_i$  de comprimento  $n$ . Esse vetor pode ser qualquer vetor não-nulo.
2. Multiplique o vetor  $[x]_i$  pela matriz  $[a]$ . O resultado é um vetor coluna  $[x]_{i+1}$ ,  $[x]_{i+1} = [a][x]_i$ .
3. Normalize o vetor resultante  $[x]_{i+1}$ . Isto é feito colocando em evidência o maior elemento do vetor. O resultado dessa operação é um fator multiplicativo (um escalar) vezes um vetor normalizado. O vetor normalizado tem o valor 1 no campo do maior elemento, enquanto o valor absoluto dos demais elementos é menor que 1.
4. Carregue o vetor normalizado (sem o fator multiplicativo) em  $[x]_i$  e volte ao passo 1.

As iterações continuam desta maneira até que a diferença entre o vetor  $[x]_i$  e o vetor normalizado  $[x]_{i+1}$  seja menor do que alguma tolerância especificada. A diferença pode ser medida de maneiras distintas. Uma possibilidade é usar a norma infinita (ver Seção 4.10.2):

$$\|[x]_{i+1} - [x]_i\|_\infty \leq \text{Tolerância} \quad (4.108)$$

O último fator multiplicativo é o maior autovalor, e o vetor normalizado é o autovetor associado.

O Exemplo 4-13 ilustra o funcionamento do método da potência.

**Exemplo 4-13: Uso do método da potência para determinar o maior autovalor de uma matriz**

Determine o maior autovalor da seguinte matriz:

$$\begin{bmatrix} 4 & 2 & -2 \\ -2 & 8 & 1 \\ 2 & 4 & -4 \end{bmatrix} \quad (4.109)$$

Use o método da potência e comece com o vetor  $x = [1, 1, 1]^T$ .

**SOLUÇÃO**

Começando com  $i = 1$ ,  $x_1 = [1, 1, 1]^T$ . No método da potência, o vetor  $[x]_2$  é primeiramente calculado a partir de  $[x]_2 = [a][x]_1$  (Passo 2) e então normalizado (Passo 3):

$$[x]_2 = [a][x]_1 = \begin{bmatrix} 4 & 2 & -2 \\ -2 & 8 & 1 \\ 2 & 4 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 2 \end{bmatrix} = 7 \begin{bmatrix} 0,5714 \\ 1 \\ 0,2857 \end{bmatrix} \quad (4.110)$$

Para  $i = 2$ , o vetor normalizado  $[x]_2$  (sem o fator multiplicativo) é multiplicado por  $[a]$ . Isto resulta no vetor  $[x]_3$ , que é então normalizado:

$$[x]_3 = [a][x]_2 = \begin{bmatrix} 4 & 2 & -2 \\ -2 & 8 & 1 \\ 2 & 4 & -4 \end{bmatrix} \begin{bmatrix} 0,5714 \\ 1 \\ 0,2857 \end{bmatrix} = \begin{bmatrix} 3,7143 \\ 7,1429 \\ 4 \end{bmatrix} = 7,1429 \begin{bmatrix} 0,52 \\ 1 \\ 0,56 \end{bmatrix} \quad (4.111)$$

As próximas três iterações são:

$$i = 3 : \quad [x]_4 = [a][x]_3 = \begin{bmatrix} 4 & 2 & -2 \\ -2 & 8 & 1 \\ 2 & 4 & -4 \end{bmatrix} \begin{bmatrix} 0,52 \\ 1 \\ 0,56 \end{bmatrix} = \begin{bmatrix} 2,96 \\ 7,52 \\ 2,8 \end{bmatrix} = 7,52 \begin{bmatrix} 0,3936 \\ 1 \\ 0,3723 \end{bmatrix} \quad (4.112)$$

$$i = 4 : \quad [x]_5 = [a][x]_4 = \begin{bmatrix} 4 & 2 & -2 \\ -2 & 8 & 1 \\ 2 & 4 & -4 \end{bmatrix} \begin{bmatrix} 0,3936 \\ 1 \\ 0,3723 \end{bmatrix} = \begin{bmatrix} 2,8298 \\ 7,5851 \\ 3,2979 \end{bmatrix} = 7,5851 \begin{bmatrix} 0,3731 \\ 1 \\ 0,4348 \end{bmatrix} \quad (4.113)$$

$$i = 5 : \quad [x]_6 = [a][x]_5 = \begin{bmatrix} 4 & 2 & -2 \\ -2 & 8 & 1 \\ 2 & 4 & -4 \end{bmatrix} \begin{bmatrix} 0,3731 \\ 1 \\ 0,4348 \end{bmatrix} = \begin{bmatrix} 2,6227 \\ 7,6886 \\ 3,0070 \end{bmatrix} = 7,6886 \begin{bmatrix} 0,3411 \\ 1 \\ 0,3911 \end{bmatrix} \quad (4.114)$$

Após mais três iterações, os resultados são:

$$i = 8 \quad [x]_9 = [a][x]_8 = \begin{bmatrix} 4 & 2 & -2 \\ -2 & 8 & 1 \\ 2 & 4 & -4 \end{bmatrix} \begin{bmatrix} 0,3272 \\ 1 \\ 0,3946 \end{bmatrix} = \begin{bmatrix} 2,5197 \\ 7,7401 \\ 3,0760 \end{bmatrix} = 7,7401 \begin{bmatrix} 0,3255 \\ 1 \\ 0,3974 \end{bmatrix} \quad (4.115)$$

Os resultados mostram que as diferenças entre o vetor  $[x]_i$  e o vetor normalizado  $[x]_{i+1}$  se tornam cada vez menores. O valor do fator multiplicativo (7,7401) é uma estimativa do maior autovalor. Conforme mostrado na Seção 4.12.3, o valor 7,7504 é obtido para este autovalor utilizando a função residente do MATLAB `eig`.

### Convergência do método da potência

O método da potência em geral converge lentamente, a menos que o vetor de partida  $[x]$  esteja próximo do autovetor. Pode-se ver a partir da Eq. (4.106) que a razão entre os dois maiores autovalores determina quão rapidamente o método da potência converge para uma resposta. Um problema pode surgir quando o vetor de partida  $[x]$  é tal que o valor de  $c_1$  na Eq. (4.102) é igual a zero. Isso significa que  $[x]$  não possui componentes na direção do autovetor correspondente  $[u]_1$ . Teoricamente, o método da potência falharia neste caso. Na prática, contudo, o método pode ainda assim convergir (muito lentamente) porque o acúmulo dos erros de arredondamento durante a multiplicação repetida pela matriz  $[a]$  tende a produzir componentes na direção de  $[u]_1$ .

### Quando é possível utilizar o método da potência?

O método da potência pode ser usado nas seguintes condições:

- Deseja-se apenas o maior autovalor da matriz.
- O maior autovalor não pode ser uma raiz repetida da equação característica. Em outras palavras, não pode haver outros autovalores com módulo igual ao maior autovalor.
- O maior autovalor deve ser real. Essa condição também está implícita no item anterior, porque, se o maior autovalor for complexo, então o complexo conjugado também é um autovalor. Isso significa que há dois autovalores com o mesmo módulo.

### Nota adicional sobre o método da potência

A matriz cujo autovalor deseja-se determinar não pode sofrer qualquer modificação (isto é, ela não pode ser passada para a forma triangular superior, triangular inferior, etc.) antes que o maior autovalor seja determinado. A modificação da matriz seguida da aplicação do método da potência resulta em uma matriz diferente da matriz original, com diferentes autovalores.

#### 4.12.2 O método da potência inverso

O método da potência inverso pode ser usado para determinar o menor autovalor. Isso é feito com a aplicação do método da potência na inversa da matriz  $[a]$  (isto é, em  $[a]^{-1}$ ). Isso funciona porque os autovalores da matriz  $[a]^{-1}$  são o inverso dos autovalores de  $[a]$ . Começando de  $[a][x] = \lambda[x]$  e multiplicando ambos os lados dessa equação por  $[a]^{-1}$ , obtém-se:

$$[a]^{-1}[a][x] = [a]^{-1}\lambda[x] = \lambda[a]^{-1}[x] \quad (4.116)$$

Como  $[a]^{-1}[a] = [I]$ , a Eq. (4.116) se reduz a:

$$[x] = \lambda[a]^{-1}[x] \text{ ou } [a]^{-1}[x] = \frac{[x]}{\lambda} \quad (4.117)$$

Isso mostra que  $1/\lambda$  é o autovalor da matriz inversa  $[a]^{-1}$ . Logo, o método da potência pode ser aplicado na determinação do maior autovalor de  $[a]^{-1}$ . O

resultado será o maior valor de  $1/\lambda$ , que corresponde ao menor valor de  $\lambda$  referente à matriz  $[a]$ . A aplicação do método da potência na inversa de  $[a]$  é chamada de **método da potência inverso**.

O procedimento de aplicação do método da potência inverso é, em princípio, igual ao método da potência. Um vetor inicial  $[x]_i$  é multiplicado por  $[a]^{-1}$  para se obter  $[x]_{i+1}$ , que é então normalizado e novamente multiplicado:

$$[x]_{i+1} = [a]^{-1}[x]_i \quad (4.118)$$

Obviamente, a matriz inversa  $[a]^{-1}$  deve ser calculada antes que as iterações referentes à Eq. (4.118) comecem. Métodos numéricos para o cálculo da inversa de uma matriz são descritos na Seção 4.6. Na prática, no entanto, o cálculo da inversa de uma matriz é computacionalmente ineficiente e indesejável. Para evitar a necessidade de se calcular a inversa de  $[a]$ , a Eq. (4.118) pode ser rescrita como:

$$[a][x]_{i+1} = [x]_i \quad (4.119)$$

Agora, para um dado  $[x]_i$ , pode-se obter  $[x]_{i+1}$  a partir da Eq. (4.119). Isso pode ser feito com o uso do método da decomposição LU (Seção 4.5).

Até agora, o método da potência foi usado na obtenção do maior e do menor autovalor de uma matriz. Em alguns casos, é necessário determinar todos os autovalores. As próximas duas seções descrevem dois métodos numéricos, o método da potência deslocado e a fatoração QR, que podem ser utilizados na obtenção de todos os autovalores.

#### 4.12.3 O método da potência deslocado

Uma vez conhecidos o menor e o maior autovalor, o método da potência deslocado pode ser usado na obtenção dos demais autovalores. Esse método usa uma importante propriedade das matrizes e de seus autovalores. Dado  $[a][x] = \lambda[x]$ , se  $\lambda_1$  é o maior (ou menor) autovalor obtido com o uso do método da potência (ou do método da potência inverso), então os autovalores de uma nova matriz **deslocada** formada por  $[a - \lambda_1 I]$  são  $0, \lambda_2 - \lambda_1, \lambda_3 - \lambda_1, \lambda_4 - \lambda_1, \dots, \lambda_n - \lambda_1$ . Isto pode ser visto facilmente porque os autovalores de  $[a - \lambda_1 I]$  são obtidos de:

$$[a - \lambda_1 I][x] = \alpha[x] \quad (4.120)$$

onde os  $\alpha$ s são os autovalores da matriz deslocada  $[a - \lambda_1 I]$ . Mas  $[a][x] = \lambda[x]$ , e então a Eq. (4.120) se torna:

$$(\lambda - \lambda_1)[x] = \alpha[x] \quad (4.121)$$

onde  $\lambda = \lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ . Portanto, os autovalores da matriz deslocada são  $\alpha = 0, \lambda_2 - \lambda_1, \lambda_3 - \lambda_1, \lambda_4 - \lambda_1, \dots, \lambda_n - \lambda_1$ . Os autovetores da matriz deslocada  $[a - \lambda_1 I]$  são iguais aos autovetores da matriz original  $[a]$ . Se o método da potência básico for aplicado à matriz deslocada  $[a - \lambda_1 I]$  (após a sua aplicação em  $[a]$  para determinar  $\lambda_1$ ), torna-se possível portanto determinar o maior autovalor dessa matriz,  $\alpha_k$ . Com isso, o autovalor  $\lambda_k$  pode ser determinado, pois  $\alpha_k = \lambda_k - \lambda_1$ . Todos os demais autovalores podem ser determinados com a repetição desse processo  $k - 2$  vezes, onde em cada vez a matriz deslocada é igual a  $[a - \lambda_k I]$  e  $\lambda_k$  é o autovalor obtido no deslocamento anterior.

O método da potência deslocado é um processo tedioso e ineficiente. Um método mais adequado à obtenção de todos os autovalores de uma matriz é o método de fatoração QR, descrito na próxima seção.

#### 4.12.4 O método QR de fatoração e iteração

O método QR de fatoração e iteração é um método popularmente usado na determinação de todos os autovalores de uma matriz. O método se baseia no fato de que matrizes similares (ver definição a seguir) têm os mesmos autovalores e autovetores associados, e também no fato de os autovalores de uma matriz triangular superior serem os elementos da diagonal. Para determinar os autovalores (todos reais) de uma matriz  $[a]$ , a estratégia do método de fatoração QR é transformar a matriz em uma matriz similar na forma triangular superior. Na realidade, isso não é feito em um único passo, mas sim, conforme descrito mais tarde, em um processo iterativo.

O método de fatoração QR determina todos os autovalores de uma matriz, mas não pode encontrar os autovetores correspondentes. Se os autovalores de uma dada matriz são reais, o método QR fatora essa matriz e a transforma em duas matrizes, uma ortogonal e outra na forma triangular superior. Se os autovalores são complexos (tópico não coberto neste livro), a matriz é fatorada em uma matriz ortogonal e em uma matriz diagonal em blocos ( $2 \times 2$ ) (isto é, uma matriz cujos elementos diagonais são matrizes  $(2 \times 2)$ ).

#### Matrizes similares

Duas matrizes quadradas  $[a]$  e  $[b]$  são similares se:

$$[a] = [c]^{-1}[b][c] \quad (4.122)$$

onde  $[c]$  é uma matriz que possui inversa. A operação na Eq. (4.122) é chamada de **transformação de similaridade**. Matrizes similares possuem os mesmos autovalores e autovetores associados.

#### Procedimento do método QR de fatoração e iteração

O procedimento de fatoração do método QR tem início com a matriz  $[a]_1$ , cujos autovalores e autovetores devem ser determinados. A matriz é fatorada em duas matrizes  $[Q]_1$  e  $[R]_1$ :

$$[a]_1 = [Q]_1[R]_1 \quad (4.123)$$

onde  $[Q]_1$  é uma matriz ortogonal e  $[R]_1$  é uma matriz triangular superior (uma matriz ortogonal é aquela cuja inversa é igual à transposta,  $[Q]^{-1} = [Q]^T$ , isto é,  $[Q]^T[Q] = [Q]^{-1}[Q] = [I]$ ).

A matriz  $[R]_1$  é então multiplicada à direita por  $[Q]_1$  para dar origem à matriz  $[a]_2$

$$[a]_2 = [R]_1[Q]_1 \quad (4.124)$$

Como a partir da Eq. (4.123)  $[R]_1 = [Q]_1^T[a]_1$ , a Eq. (4.124) se reduz a:

$$[a]_2 = [Q]_1^T[a]_1[Q]_1 \quad (4.125)$$

Isso significa (ver Eq. (4.122)) que as matrizes  $[a]_1$  e  $[a]_2$  são similares, possuindo portanto os mesmos autovalores. Isto completa a primeira iteração no método QR.

A segunda iteração tem início com a fatoração da matriz  $[a]_2$  em  $[Q]_2$  (ortogonal) e  $[R]_2$  (triangular superior), de forma que  $[a]_2 = [Q]_2[R]_2$ , e então com o cálculo de  $[a]_3$  usando  $[a]_3 = [R]_2[Q]_2$ . Novamente, como  $[R]_2 = [Q]_2^T[a]_1$ , a matriz  $[a]_3$  é dada por  $[a]_3 = [Q]_2^T[R]_2[Q]_2$ . As matrizes  $[a]_3$  e  $[a]_2$  são similares, e possuem portanto os mesmos autovalores (que são iguais aos autovalores de  $[a]_1$ ).

As iterações continuam até que a seqüência gerada de matrizes,  $[a]_1$ ,  $[a]_2$ ,  $[a]_3$ ,..., resulte em uma matriz triangular na forma:

$$\begin{bmatrix} \lambda_1 & X & X & X \\ 0 & \lambda_2 & X & X \\ 0 & 0 & \lambda_3 & X \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix}$$

onde os autovalores da matriz original  $[a]_1$  aparecem ao longo da diagonal. Os autovalores,  $\lambda_1, \lambda_2, \lambda_3, \dots$ , não estão em nenhuma ordem particular.

Em cada iteração do método QR, a fatoração da matriz  $[a]$  em uma matriz ortogonal  $[Q]$  e em uma matriz triangular superior  $[R]$ , tal que  $[a] = [Q][R]$ , é feita em passos com o uso de uma matriz especial  $[H]$ , chamada de **matriz de Householder**.

### A matriz de Householder $[H]$

A matriz de Householder  $[H]$  de ordem  $(n \times n)$  tem a forma:

$$[H] = [I] - \frac{2}{[v]^T[v]}[v][v]^T \quad (4.126)$$

onde  $[I]$  é a matriz identidade  $(n \times n)$  e  $[v]$  é um vetor coluna com  $n$  elementos dado por:

$$[v] = [c] + \|c\|_2[e] \quad (4.127)$$

Na Eq. (4.127),  $[e]$  e  $[c]$  são vetores coluna com  $n$  elementos, e  $\|c\|_2$  é a norma Euclidiana (comprimento) de  $[c]$ :

$$\|c\|_2 = \sqrt{c_1^2 + c_2^2 + c_3^2 + \dots + c_n^2} \quad (4.128)$$

Note que  $[v]^T[v]$  é um escalar (número) e  $[v][v]^T$  é uma matriz  $(n \times n)$ . Os vetores  $[c]$  e  $[e]$  são descritos em detalhe na próxima subseção.

A matriz de Householder tem propriedades especiais. Em primeiro lugar, ela é simétrica. Além disso, ela é ortogonal. Logo,  $[H]^{-1} = [H]^T = H$ . Isso significa que  $[H][a][H]$  resulta em uma matriz que é similar a  $[a]$ .

### Fatoração de uma matriz $[a]$ em uma matriz ortogonal $[Q]$ e em uma matriz triangular superior $[R]$

A fatoração de uma matriz  $[a]$  de ordem  $(n \times n)$  em uma matriz ortogonal  $[Q]$  e em uma matriz triangular superior  $[R]$ , tal que  $[a] = [Q][R]$ , é feita em  $(n - 1)$  passos.

**Passo 1** O vetor  $[c]$ , que tem  $n$  elementos, é definido como a primeira coluna da matriz  $[a]$ :

$$[c] = \begin{bmatrix} a_{11} \\ a_{21} \\ \dots \\ a_{n1} \end{bmatrix} \quad (4.129)$$

O vetor  $[e]$  é definido como o seguinte vetor coluna de comprimento  $n$ :

$$[e] = \begin{bmatrix} \pm 1 \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (4.130)$$

O primeiro elemento de  $[e]$  é igual a +1 se o primeiro elemento de  $[c]$  ( $a_{11}$ ) for positivo e é igual a -1 se o primeiro elemento de  $[c]$  for negativo. Os demais elementos são iguais a zero.

Uma vez identificados  $[c]$  e  $[e]$ , a matriz de Householder  $[H]^{(1)}$  de ordem  $(n \times n)$  pode ser construída com o emprego das Eqs. (4.126)-(4.128). Usando  $[H]^{(1)}$ , a matriz  $[a]$  é fatorada em  $[Q]^{(1)} [R]^{(1)}$ , onde:

$$\begin{bmatrix} R_{11}^{(1)} & R_{12}^{(1)} & R_{13}^{(1)} & R_{14}^{(1)} & R_{15}^{(1)} \\ 0 & R_{22}^{(1)} & R_{23}^{(1)} & R_{24}^{(1)} & R_{25}^{(1)} \\ 0 & R_{32}^{(1)} & R_{33}^{(1)} & R_{34}^{(1)} & R_{35}^{(1)} \\ 0 & R_{42}^{(1)} & R_{43}^{(1)} & R_{44}^{(1)} & R_{45}^{(1)} \\ 0 & R_{52}^{(1)} & R_{53}^{(1)} & R_{54}^{(1)} & R_{55}^{(1)} \end{bmatrix}$$

**Figura 4-23** A matriz  $[R]^{(1)}$  após o Passo 1.

$$[Q]^{(1)} = [H]^{(1)} \quad (4.131)$$

e

$$[R]^{(1)} = [H]^{(1)}[a] \quad (4.132)$$

A matriz  $[Q]^{(1)}$  é ortogonal por ser uma matriz de Householder, e  $[R]^{(1)}$  é uma matriz com zeros nos elementos pertencentes à primeira coluna que estão abaixo do campo (1, 1). A matriz  $[R]^{(1)}$  é ilustrada na Fig. 4-23 para o caso de uma matriz  $(5 \times 5)$ .

**Passo 2** O vetor  $[c]$ , que possui  $n$  elementos, é definido como a segunda coluna da matriz  $[R]^{(1)}$  com o seu primeiro campo igualado a zero:

$$[c] = \begin{bmatrix} 0 \\ R_{22}^{(1)} \\ R_{32}^{(1)} \\ \dots \\ R_{n2}^{(1)} \end{bmatrix} \quad (4.133)$$

O vetor  $[e]$  é definido como o seguinte vetor coluna de comprimento  $n$ :

$$[e] = \begin{bmatrix} 0 \\ \pm 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (4.134)$$

O segundo elemento de  $[e]$  é igual a  $+1$  se o segundo elemento de  $[c]$  na Eq. (4.133) (que é  $R_{22}^{(1)}$ ) for positivo e é igual a  $-1$  se o segundo elemento de  $[c]$  for negativo. Os demais elementos são iguais a zero.

Uma vez identificados  $[c]$  e  $[e]$ , a próxima matriz de Householder  $[H]^{(2)}$  pode ser construída com o emprego das Eqs. (4.126)-(4.128). Em seguida, usando  $[H]^{(2)}$ , a matriz  $[a]$  é fatorada em  $[Q]^{(2)} [R]^{(2)}$ , onde:

$$\begin{bmatrix} R_{11}^{(2)} & R_{12}^{(2)} & R_{13}^{(2)} & R_{14}^{(2)} & R_{15}^{(2)} \\ 0 & R_{22}^{(2)} & R_{23}^{(2)} & R_{24}^{(2)} & R_{25}^{(2)} \\ 0 & 0 & R_{33}^{(2)} & R_{34}^{(2)} & R_{35}^{(2)} \\ 0 & 0 & R_{43}^{(2)} & R_{44}^{(2)} & R_{45}^{(2)} \\ 0 & 0 & R_{53}^{(2)} & R_{54}^{(2)} & R_{55}^{(2)} \end{bmatrix}$$

**Figura 4-24** A matriz  $[R]^{(2)}$  após o Passo 2.

$$[Q]^{(2)} = [Q]^{(1)} [H]^{(2)} \quad (4.135)$$

e

$$[R]^{(2)} = [H]^{(2)} [R]^{(1)} \quad (4.136)$$

A matriz  $[Q]^{(2)}$  é ortogonal e  $[R]^{(2)}$  é uma matriz com zeros nos elementos pertencentes à primeira e à segunda coluna que estão abaixo dos elementos diagonais dessas colunas. A matriz  $[R]^{(2)}$  é ilustrada na Fig. 4-24 para o caso de uma matriz  $(5 \times 5)$ .

**Passo 3** Passando para a terceira coluna de  $[a]$ , o vetor  $[c]$ , que tem  $n$  elementos, é definido como:

$$[c] = \begin{bmatrix} 0 \\ 0 \\ R_{33}^{(2)} \\ R_{34}^{(2)} \\ \dots \\ R_{n3}^{(2)} \end{bmatrix} \quad (4.137)$$

O vetor  $[e]$  é definido como o seguinte vetor coluna de comprimento  $n$ :

$$[e] = \begin{bmatrix} 0 \\ 0 \\ \pm 1 \\ 0 \\ \dots \\ 0 \end{bmatrix} \quad (4.138)$$

O terceiro elemento de  $[e]$  é igual a  $+1$  se o terceiro elemento de  $[c]$  na Eq. (4.133) (que é  $R_{33}^{(2)}$ ) for positivo e é igual a  $-1$  se o terceiro elemento de  $[c]$  for negativo. Os demais elementos são iguais a zero.

Uma vez identificados  $[c]$  e  $[e]$ , a próxima matriz de Householder  $[H]^{(3)}$  de ordem  $(n \times n)$  pode ser construída com o emprego das Eqs. (4.126)-(4.128). Em seguida, usando  $[H]^{(3)}$ , a matriz  $[a]$  é fatorada em  $[Q]^{(3)} [R]^{(3)}$ , onde:

$$[Q]^{(3)} = [Q]^{(2)} [H]^{(3)} \quad (4.139)$$

e

$$[R]^{(3)} = [H]^{(3)} [R]^{(2)} \quad (4.140)$$

**Figura 4-25** A matriz  $[R]^{(3)}$  após o Passo 3.

A matriz  $[Q]^{(3)}$  é ortogonal e  $[R]^{(3)}$  é uma matriz com zeros nos elementos pertencentes à primeira, à segunda e à terceira coluna que estão abaixo dos elementos diagonais dessas colunas. A matriz  $[R]^{(3)}$  é ilustrada na Fig. 4-25 para o caso de uma matriz  $(5 \times 5)$ .

**Passos 4 até  $(n - 1)$**  A fatoração da matriz  $[a]$  em matrizes ortogonais e triangulares superiores continua da mesma forma que nos passos 1 a 3. Em um passo genérico  $i$ , o vetor  $[c]$  tem zeros nos elementos 1 até  $i - 1$ , e, nos elementos  $i$  até  $n$ , esse vetor contém os valores dos elementos  $R_{ii}^{(i-1)}$  até  $R_{ni}^{(i-1)}$  da matriz  $[R]^{(i-1)}$  calculada no passo anterior. O vetor  $[e]$  tem +1 no  $i$ -ésimo elemento se o  $i$ -ésimo elemento de  $[c]$  for positivo e -1 se o  $i$ -ésimo elemento de  $[c]$  for negativo. Os demais elementos são iguais a zero.

Uma vez identificados  $[c]$  e  $[e]$ , a próxima matriz de Householder  $[H]^{(i)}$  de ordem  $(n \times n)$  pode ser construída com o emprego das Eqs. (4.126)-(4.128). Em seguida, usando  $[H]^{(i)}$ , a matriz  $[a]$  é fatorada em  $[Q]^{(i)} [R]^{(i)}$ , onde:

$$[Q]^{(i)} = [Q]^{(i-1)} [H]^{(i)} \quad (4.141)$$

e

$$[R]^{(i)} = [H]^{(i)} [R]^{(i-1)} \quad (4.142)$$

A matriz  $[Q]^{(i)}$  é ortogonal e  $[R]^{(i)}$  é uma matriz com zeros nos elementos abaixo da diagonal principal nas colunas 1 a  $i$ .

Após o último passo (Passo  $n - 1$ ), a matriz  $[R]^{(n-1)}$  é triangular superior. As matrizes  $[Q]^{(n-1)}$  e  $[R]^{(n-1)}$  obtidas no último passo são as matrizes ortogonal e triangular superior nas quais a matriz  $[A]$  é fatorada ao longo do processo iterativo.

$$[A] = [Q]^{(n-1)} [R]^{(n-1)} \quad (4.143)$$

O Exemplo 4-14 mostra o cálculo manual da fatoração QR de uma matriz.

### Exemplo 4-14: Fatoração QR de uma matriz

Fatore a seguinte matriz  $[a]$  em uma matriz ortogonal  $[Q]$  e uma matriz triangular  $[R]$ :

$$[a] = \begin{bmatrix} 6 & -7 & 2 \\ 4 & -5 & 2 \\ 1 & -1 & 1 \end{bmatrix} \quad (4.144)$$

### SOLUÇÃO

A solução segue os passos listados nas páginas anteriores. Como a matriz  $[a]$  é  $(3 \times 3)$ , a fatoração requer apenas 2 passos.

**Passo 1** Define-se o vetor  $[c]$  como a primeira coluna da matriz  $[a]$ :

$$[c] = \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix}$$

O vetor  $[e]$  é definido como o seguinte vetor coluna com três elementos:

$$[e] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Usando a Eq. (4.128), a norma Euclidiana,  $\|c\|_2$ , de  $[c]$ , é:

$$\|c\|_2 = \sqrt{c_1^2 + c_2^2 + c_3^2} = \sqrt{6^2 + 4^2 + 1^2} = 7,2801$$

Usando a Eq. (4.127), o vetor  $[v]$  é:

$$[v] = [c] + \|c\|_2 [e] = \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} + 7,2801 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 13,2801 \\ 4 \\ 1 \end{bmatrix}$$

Em seguida, os produtos  $[v]^T[v]$  e  $[v][v]^T$  são calculados:

$$[v]^T[v] = \begin{bmatrix} 13,2801 & 4 & 1 \end{bmatrix} \begin{bmatrix} 13,2801 \\ 4 \\ 1 \end{bmatrix} = 193,3611$$

$$[v][v]^T = \begin{bmatrix} 13,2801 \\ 4 \\ 1 \end{bmatrix} \begin{bmatrix} 13,2801 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 176,3611 & 53,1204 & 13,2801 \\ 53,1204 & 16 & 4 \\ 13,2801 & 4 & 1 \end{bmatrix}$$

A matriz de Householder  $[H]^{(1)}$  é então:

$$[H]^{(1)} = [I] - \frac{2}{[v]^T[v]} [v][v]^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{2}{193,3611} \begin{bmatrix} 176,3611 & 53,1204 & 13,2801 \\ 53,1204 & 16 & 4 \\ 13,2801 & 4 & 1 \end{bmatrix} = \begin{bmatrix} -0,8242 & -0,5494 & -0,1374 \\ -0,5494 & 0,8345 & -0,0414 \\ -0,1374 & -0,0414 & 0,9897 \end{bmatrix}$$

Uma vez construída a matriz de Householder  $[H]^{(1)}$ ,  $[a]$  pode ser fatorada em  $[Q]^{(1)}[R]^{(1)}$ , onde:

$$[Q]^{(1)} = [H]^{(1)} = \begin{bmatrix} -0,8242 & -0,5494 & -0,1374 \\ -0,5494 & 0,8345 & -0,0414 \\ -0,1374 & -0,0414 & 0,9897 \end{bmatrix}$$

e

$$[R]^{(1)} = [H]^{(1)}[a] = \begin{bmatrix} -0,8242 & -0,5494 & -0,1374 \\ -0,5494 & 0,8345 & -0,0414 \\ -0,1374 & -0,0414 & 0,9897 \end{bmatrix} \begin{bmatrix} 6 & -7 & 2 \\ 4 & -5 & 2 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} -7,2801 & 8,6537 & -2,8846 \\ 0 & -0,2851 & 0,5288 \\ 0 & 0,1787 & 0,6322 \end{bmatrix}$$

Isso completa o primeiro passo.

**Passo 2** O vetor  $[c]$ , que possui três elementos, é definido como:

$$[c] = \begin{bmatrix} 0 \\ R_{22}^{(1)} \\ R_{32}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ -0,2851 \\ 0,1787 \end{bmatrix}$$

O vetor  $[e]$  é definido como o seguinte vetor coluna com três elementos:

$$[e] = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

Usando a Eq. (4.128), a norma Euclidiana,  $\|c\|_2$ , de  $[c]$ , é:

$$\|c\|_2 = \sqrt{c_1^2 + c_2^2 + c_3^2} = \sqrt{0^2 + (-0,2851)^2 + 0,1787^2} = 0,3365$$

Usando a Eq. (4.127), o vetor  $[v]$  é:

$$[v] = [c] + \|c\|_2[e] = \begin{bmatrix} 0 \\ -0,2851 \\ 0,1787 \end{bmatrix} + 0,3365 \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -0,6215 \\ 0,1787 \end{bmatrix}$$

Em seguida, os produtos  $[v]^T[v]$  e  $[v][v]^T$  são calculados:

$$[v]^T[v] = \begin{bmatrix} 0 & -0,6215 & 0,1787 \end{bmatrix} \begin{bmatrix} 0 \\ -0,6215 \\ 0,1787 \end{bmatrix} = 0,4183$$

$$[v][v]^T = \begin{bmatrix} 0 \\ -0,6215 \\ 0,1787 \end{bmatrix} \begin{bmatrix} 0 & -0,6215 & 0,1787 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0,3864 & -0,1111 \\ 0 & 0,1111 & 0,0319 \end{bmatrix}$$

A matriz de Householder  $[H]^{(2)}$  é então:

$$[H]^{(2)} = [I] - \frac{2}{[v]^T[v]} [v][v]^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{2}{0,4183} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0,3864 & -0,1111 \\ 0 & 0,1111 & 0,0319 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0,8474 & 0,5311 \\ 0 & 0,5311 & 0,8473 \end{bmatrix}$$

Uma vez construída a matriz de Householder  $[H]^{(2)}$ ,  $[a]$  pode ser fatorada em  $[Q]^{(2)}[R]^{(2)}$ , onde:

$$[Q]^{(2)} = [Q]^{(1)}[H]^{(2)} = \begin{bmatrix} -0,8242 & -0,5494 & -0,1374 \\ -0,5494 & 0,8345 & -0,0414 \\ -0,1374 & -0,0414 & 0,9897 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0,8474 & 0,5311 \\ 0 & 0,5311 & 0,8473 \end{bmatrix} = \begin{bmatrix} -0,8242 & 0,3927 & -0,4082 \\ -0,5494 & -0,7291 & 0,4082 \\ -0,1374 & 0,5607 & 0,8166 \end{bmatrix}$$

e

$$[R]^{(2)} = [H]^{(2)}[R]^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0,8474 & 0,5311 \\ 0 & 0,5311 & 0,8473 \end{bmatrix} \begin{bmatrix} -7,2801 & 8,6537 & -2,8846 \\ 0 & -0,2851 & 0,5288 \\ 0 & 0,1787 & 0,6322 \end{bmatrix} = \begin{bmatrix} -7,2801 & 8,6537 & -2,8846 \\ 0 & 0,3365 & -0,1123 \\ 0 & 0 & 0,8165 \end{bmatrix}$$

Isso completa a fatoração, o que significa que:

$$[a] = [Q]^{(2)}[R]^{(2)} \quad \text{ou} \quad \begin{bmatrix} 6 & -7 & 2 \\ 4 & -5 & 2 \\ 1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} -0,8242 & 0,3927 & -0,4082 \\ -0,5494 & -0,7291 & 0,4082 \\ -0,1374 & 0,5607 & 0,8166 \end{bmatrix} \begin{bmatrix} -7,2801 & 8,6537 & -2,8846 \\ 0 & 0,3365 & -0,1123 \\ 0 & 0 & 0,8165 \end{bmatrix}$$

Os resultados podem ser conferidos no MATLAB. Primeiramente, verifica-se que a matriz  $[Q]^{(2)}$  é ortogonal. Isso é feito com o cálculo da inversa de  $[Q]^{(2)}$  com a função `inv` do MATLAB, e com a verificação de que essa matriz é igual à transposta de  $[Q]^{(2)}$ . Em seguida, multiplica-se  $[Q]^{(2)}[R]^{(2)}$  no MATLAB e compara-se o resultado com  $[a]$ .

```

>> Q2 = [-0.8242 0.3927 -0.4082; -0.5494 -0.7291 0.4084; -0.1374 0.5607 0.8166];
>> R2 = [-7.2801 8.6537 -2.8846; 0 0.3365 -0.1123; 0 0 0.8165];
>> invQ2 = inv(Q2)
invQ2 =
-0.8242 -0.5494 -0.1372
 0.3924 -0.7290  0.5607
 -0.4081  0.4081  0.8165
>> a = Q2*R2
a =
 6.0003 -7.0002  2.0001
 3.9997 -4.9997  2.0001
 1.0003 -1.0003  1.0001

```

Os resultados verificam que a fatoração foi feita de forma adequada, a menos de erros de arredondamento.

O método de fatoração QR usado na obtenção dos autovalores de uma matriz é resumido no algoritmo a seguir:

### Algoritmo para determinar autovalores com o método de fatoração QR

Dada uma matriz  $[a]_1$  de ordem  $(n \times n)$  cujos autovalores deseja-se determinar:

1. Fatore  $[a]_1$  em uma matriz ortogonal  $[Q]_1$  e em uma matriz triangular superior  $[R]_1$ , tais que  $[a]_1 = [Q]_1[R]_1$ . Isso é feito em  $n - 1$  passos conforme descrito nas páginas 178-180 (*Passo 1*).
2. Calcule  $[a]_2$  usando  $[a]_2 = [R]_1[Q]_1$ .
3. Repita os dois primeiros passos para obter a seqüência de matrizes  $[a]_1$ ,  $[a]_3$ ,  $[a]_4$ ,..., até que a última matriz na seqüência seja triangular superior. Os elementos ao longo da diagonal principal são então os autovalores.

O Exemplo 4-15 mostra a implementação do método de fatoração QR no MATLAB.

### Exemplo 4-15: Cálculo de autovalores usando o método de fatoração QR

O esforço tridimensional em um ponto no interior de uma estrutura carregada é dada por:

$$\sigma_{ij} = \begin{bmatrix} 45 & 30 & -25 \\ 30 & -24 & 68 \\ -25 & 68 & 80 \end{bmatrix} \text{ MPa}$$

Determine os esforços principais neste ponto encontrando os autovalores da matriz de esforços. Use para isso o método QR de fatoração.

### SOLUÇÃO

O problema é resolvido no MATLAB. Primeiramente, escreve-se uma função denominada `QRFatora`. Em seguida, essa função é usada em um programa que determina autovalores usando o método QR de fatoração.

**Programa 4-10 Função definida pelo usuário. Fatoração QR de uma matriz.**

```

function [Q R] = QRFatora(R)
% A função fatora a matriz [A] em uma matriz ortogonal [Q]
% e em uma matriz triangular superior [R].
% Variáveis de entrada:
% A Matriz (quadrada) a ser fatorada.
% Variáveis de saída:
% Q Matriz ortogonal.
% R Matriz triangular superior.

nmatrix = size(R);
n = nmatrix(1);
I = eye(n);
Q = I;
for j = 1:n - 1
    c = R(:,j);
    c(1:j - 1) = 0;
    e(1:n,1) = 0;
    if c(j) > 0
        e(j) = 1;
    else
        e(j) = -1;
    end
    clenlength = sqrt(c'*c);
    v = c + clenlength*e;
    H = I - 2/(v'*v)*v*v';
    Q = Q*H;
    R = H*R;
end

```

Define o vetor  $[c]$ .

Define o vetor  $[e]$ .

Eq. (4.128).

Gera o vetor  $[v]$ , Eq. (4.127).

Constrói a matriz de Householder  $[H]$ , Eq. (4.126).

A determinação dos autovalores segue o procedimento indicado no algoritmo.

```

A = [45 30 -25; 30 -24 68; -25 68 80]
for i = 1:40
    [q R] = QRFatora(A);
    A = R*q;
end
A
e = diag(A)

```

O programa repete a fatoração QR 40 vezes e então exibe na tela de comandos do MATLAB a última matriz  $[A]$  obtida. Os elementos diagonais da matriz são os autovalores da matriz original  $[A]$ .

```

A =
45 30 -25
30 -24 68
-25 68 80

A =
114.9545 0.0000 0.0000
0.0000 -70.1526 -1.5563
0.0000 -1.5563 56.1981

e =
114.9545
-70.1526
56.1981

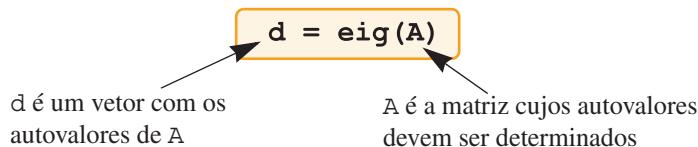
```

Os resultados mostram que, após 40 iterações, a matriz  $[A]$  é praticamente triangular superior. Na realidade, neste caso, a fatoração QR resulta em uma matriz diagonal porque a matriz  $[\sigma]$  é simétrica.

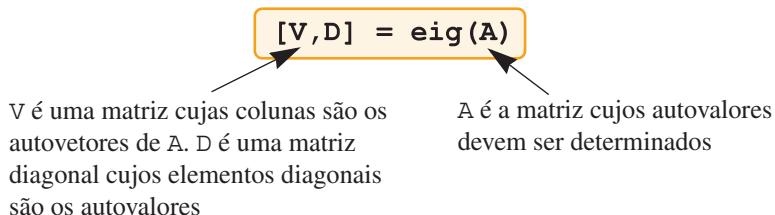
#### 4.12.5 Uso de funções residentes do MATLAB para determinar autovalores e autovetores

O MATLAB dispõe de muitas funções residentes para determinar os autovalores e os autovetores de uma matriz, e de uma função residente que realiza a fatoração QR.

Os autovalores e autovetores de uma matriz podem ser determinados com o emprego da função residente `eig`. Se apenas os autovalores são desejados, a função tem a forma:



Para determinar os autovalores e os autovetores, a função residente tem a seguinte forma:



Com essa notação,  $A \cdot V = V \cdot D$ . Por exemplo, se  $A = \begin{bmatrix} 6 & -7 & 2 \\ 4 & -5 & 2 \\ 1 & -1 & 1 \end{bmatrix}$ , os autovalores e os autovetores associados podem ser determinados digitando:

```
>> A = [6 -7 2; 4 -5 2; 1 -1 1];
>> lambdas = eig(A)
lambdas =
-1.0000
2.0000
1.0000
>> [eVectors eValues] = eig(A)
```

Usa `eig` para determinar apenas os autovalores.

```
eVectors =
```

Usa `eig` para determinar os autovalores e os autovetores.

```
0.7071 0.8018 0.5774
0.7071 0.5345 0.5774
0.0000 0.2673 0.5774
```

Cada coluna é um autovetor.

```
eValues =
```

```
-1.0000 0 0
0 2.0000 0
0 0 1.0000
```

Os autovalores são os elementos da diagonal.

O MATLAB também dispõe de uma função residente para realizar a fatoração QR de matrizes. A função é chamada de `qr`, e seu formato mais simples é:

$$[Q, R] = qr(A)$$

$Q$  é uma matriz ortogonal e  $R$  é uma matriz triangular superior tal que  
 $A=Q*R$

Como exemplo, a matriz que foi fatorada no Exemplo 4-14 é fatorada abaixo usando a função `qr`.

```
>> A = [6 -7 2; 4 -5 2; 1 -1 1];
>> [Q R] = qr(A)
Q =
-0.8242 0.3925 -0.4082
-0.5494 -0.7290 0.4082
-0.1374 0.5608 0.8165
R =
-7.2801 8.6537 -2.8846
0 0.3365 -0.1122
0 0 0.8165
```

Usa a função `qr` do MATLAB para fatorar a matriz  $A$ .

$Q$  é uma matriz ortogonal.

$R$  é uma matriz triangular superior.

## 4.13 PROBLEMAS

### Problemas de solução manual

Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos. Não utilize funções residentes do MATLAB.

- 4.1** Resolva o seguinte sistema de equações usando o método de eliminação de Gauss:

$$x_1 + 2x_2 - 2x_3 = 9$$

$$2x_1 + 3x_2 + x_3 = 23$$

$$3x_1 + 2x_2 - 4x_3 = 11$$

- 4.2** Dado o sistema de equações  $[a][x] = [b]$ , onde  $a = \begin{bmatrix} 2 & -4 & 1 \\ 6 & 2 & -1 \\ -2 & 6 & -2 \end{bmatrix}$ ,  $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$  e  $b = \begin{bmatrix} 4 \\ 10 \\ -6 \end{bmatrix}$ , determine a solução usando o método de eliminação de Gauss.

- 4.3** Considere o sistema de duas equações lineares a seguir:

$$0,0003x_1 + 1,566x_2 = 1,569$$

$$0,3454x_1 - 2,436x_2 = 1,018$$

- (a) Resolva o sistema usando o método de eliminação de Gauss arredondando em quatro algarismos significativos.

- (b) Troque a ordem das equações e resolva o sistema com o método de eliminação de Gauss arredondando em quatro algarismos significativos.

Verifique as respostas substituindo a solução de volta nas equações.

- 4.4** Resolva o sistema de equações a seguir usando o método de eliminação de Gauss:

$$4x_1 + 3x_2 + 2x_3 + x_4 = 1$$

$$3x_1 + 4x_2 + 3x_3 + 2x_4 = 1$$

$$2x_1 + 3x_2 + 4x_3 + 3x_4 = -1$$

$$x_1 + 2x_2 + 3x_3 + 4x_4 = -1$$

- 4.5** Resolva o sistema de equações a seguir usando o método de eliminação de Gauss:

$$2x_1 + x_2 + 4x_3 - 2x_4 = 19$$

$$-3x_1 + 4x_2 + 2x_3 - x_4 = 1$$

$$3x_1 + 5x_2 - 2x_3 + x_4 = 8$$

$$-2x_1 + 3x_2 + 2x_3 + 4x_4 = 13$$

**4.6** Resolva o sistema de equações a seguir usando o método de Gauss-Jordan:

$$x_1 + 2x_2 - 2x_3 = 9$$

$$2x_1 + 3x_2 + x_3 = 23$$

$$3x_1 + 2x_2 - 4x_3 = 11$$

**4.7** Dado o sistema de equações  $[a][x] = [b]$ , onde  $a = \begin{bmatrix} 2 & -4 & 1 \\ 6 & 2 & -1 \\ -2 & 6 & -2 \end{bmatrix}$ ,  $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$  e

$$b = \begin{bmatrix} 4 \\ 10 \\ -6 \end{bmatrix}, \text{ determine a solução usando o método de Gauss-Jordan.}$$

**4.8** Resolva o sistema de equações a seguir usando o método de eliminação de Gauss-Jordan:

$$2x_1 + x_2 + 4x_3 - 2x_4 = 19$$

$$-3x_1 + 4x_2 + 2x_3 - x_4 = 1$$

$$3x_1 + 5x_2 - 2x_3 + x_4 = 8$$

$$-2x_1 + 3x_2 + 2x_3 + 4x_4 = 13$$

**4.9** Determine a decomposição LU da matriz  $a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 3 & 2 & 2 \end{bmatrix}$  usando o procedimento de eliminação de Gauss.

**4.10** Determine a decomposição LU da matriz  $a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 3 & 2 & 2 \end{bmatrix}$  usando o método de Crout.

**4.11** Resolva o seguinte sistema com a decomposição LU usando o método de Crout.

$$\begin{bmatrix} 5 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 9 \\ 4 \\ -6 \end{bmatrix}$$

**4.12** Determine a inversa da matriz  $\begin{bmatrix} -4/5 & -3/5 & -2/5 \\ -3/5 & -6/5 & -4/5 \\ -2/5 & -4/5 & -6/5 \end{bmatrix}$  usando o método de Gauss-Jordan.

**4.13** Dada a matriz  $a = \begin{bmatrix} 6 & 3 & 11 \\ 3 & 2 & 7 \\ 3 & 2 & 6 \end{bmatrix}$ , determine a inversa de  $[a]$  usando o método de Gauss-Jordan.

**4.14** Realize as três primeiras iterações da solução do seguinte sistema de equações usando o método iterativo de Gauss-Seidel. Como primeira tentativa da solução, assuma que os valores das incógnitas sejam iguais a zero.

$$8x_1 + 2x_2 + 3x_3 = 51$$

$$2x_1 + 5x_2 + x_3 = 23$$

$$-3x_1 + x_2 + 6x_3 = 20$$

**4.15** Determine o número condição da matriz do Problema 4.13 usando a norma infinita.

**4.16** Determine o número condição da matriz do Problema 4.13 usando a norma 1.

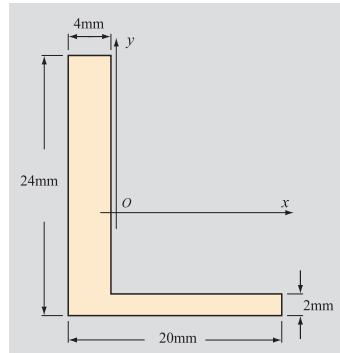
**4.17** Mostre que os autovalores da matriz identidade  $n \times n$  são números 1 repetidos  $n$  vezes.

**4.18** Mostre que os autovalores da seguinte matriz são  $10, \sqrt{2}, -\sqrt{2}$ .

$$\begin{bmatrix} 10 & 0 & 0 \\ 1 & -3 & -7 \\ 0 & 1 & 3 \end{bmatrix}$$

**4.19** Os momentos de inércia  $I_x$ ,  $I_y$  e o produto de inércia  $I_{xy}$  da área lateral mostrada na figura são  $I_x = 7523 \text{ mm}^4$ ,  $I_y = 3210 \text{ mm}^4$  e  $I_{xy} = -2640 \text{ mm}^4$ .

Os momentos de inércia principais são os autovalores da matriz  $\begin{bmatrix} 7523 & -2640 \\ -2640 & 3210 \end{bmatrix}$  e os eixos principais estão na direção dos autovetores. Determine os momentos de inércia principais resolvendo a equação característica. Determine a orientação dos eixos de inércia principais (vetores unitários na direção dos autovetores).



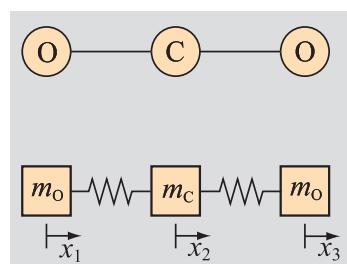
**4.20** Determine os momentos de inércia principais da área lateral mostrada no Problema 4.19 usando o método de fatoração QR. Realize as quatro primeiras iterações.

**4.21** A estrutura de uma molécula de  $\text{CO}_2$  pode ser idealizada como três massas conectadas por duas molas, onde as massas são os átomos de carbono e oxigênio e as molas representam a ligação química entre os átomos. A equação do movimento para cada átomo (massa) pode ser escrita como:

$$m_O \frac{d^2x_1}{dt^2} = -kx_1 + kx_2$$

$$m_C \frac{d^2x_2}{dt^2} = -2kx_2 + kx_1 + kx_3$$

$$m_O \frac{d^2x_3}{dt^2} = kx_2 - kx_3$$



onde  $k$  é a constante elástica das molas que representam as ligações C-O. Como a molécula pode vibrar livremente, vibrações normais (isto é, ao longo do eixo) podem ser examinadas com a substituição de  $x_j = A_j e^{i\omega t}$ , onde  $A_j$  é a amplitude da  $j$ -ésima massa,  $i = \sqrt{-1}$ ,  $\omega$  é a freqüência e  $t$  é o tempo. Isto resulta no seguinte sistema de equações:

$$\begin{aligned} -\omega^2 A_1 &= -\frac{k}{m_O} A_1 + \frac{k}{m_O} A_2 \\ -\omega^2 A_2 &= -\frac{2k}{m_C} A_2 + \frac{k}{m_C} A_1 + \frac{k}{m_C} A_3 \\ -\omega^2 A_3 &= \frac{k}{m_O} A_2 - \frac{k}{m_O} A_3 \end{aligned} \quad (4.145)$$

- (a) Rescreva o sistema de equações da Eq. (4.145) como um problema de autovalores e mostre que  $\omega_2$  é o autovalor.
- (b) Escreva a equação característica e a resolva analiticamente para diferentes freqüências.
- (c) Se  $k = 14,2 \times 10^2 \text{ kg/s}^2$ ,  $m_O = 16 \text{ amu}$  e  $m_C = 12 \text{ amu}$ , ( $1 \text{ amu} = 1,6605 \times 10^{-27} \text{ kg}$ ), determine os comprimentos de onda  $\lambda = 2\pi c/\omega$  (onde  $c = 3 \times 10^8 \text{ m/s}$  é a velocidade da luz) que correspondem às freqüências da letra (b).
- (d) Determine os autovetores correspondentes aos autovalores obtidos nas letras (b) e (c). A partir dos autovetores, deduza o movimento relativo dos átomos (se eles se aproximam ou se afastam uns dos outros).

**4.22** O esforço em um ponto é dado pelo tensor de esforços:

$$\sigma_{ij} = \begin{bmatrix} 30 & 15 & 20 \\ 15 & 22 & 26 \\ 20 & 26 & 40 \end{bmatrix}$$

Os esforços principais e as direções principais são dados pelos autovalores e autovetores. Use o método da potência para determinar o valor do maior esforço principal. Comece com um vetor coluna de 1s e realize as três primeiras iterações.

### Problemas de programação no MATLAB

*Resolva os seguintes problemas usando o MATLAB. Não use funções residentes para realizar operações com matrizes.*

**4.23** Modifique a função GaussPivot no Programa 4-2 (Exemplo 4-3) de forma tal que, em cada passo do processo de eliminação, a linha pivô seja trocada pela linha que tiver o elemento pivô com o maior valor absoluto. Como nome de função e argumentos, use  $x = \text{GaussPivotMaior}(a, b)$ , onde  $a$  é a matriz de coeficientes,  $b$  é a coluna de constantes no lado direito e  $x$  é a solução.

- (a) Use a função GaussPivotMaior para solucionar o sistema de equações na Eq. (4.17).
- (b) Use a função GaussPivotMaior para solucionar o sistema:

$$\left[ \begin{array}{cccccc} 0 & 3 & 8 & -5 & -1 & 6 \\ 3 & 12 & -4 & 8 & 5 & -2 \\ 8 & 0 & 0 & 10 & -3 & 7 \\ 3 & 1 & 0 & 0 & 0 & 4 \\ 0 & 0 & 4 & -6 & 0 & 2 \\ 3 & 0 & 5 & 0 & 0 & -6 \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \right] = \left[ \begin{array}{c} 34 \\ 20 \\ 45 \\ 36 \\ 60 \\ 28 \end{array} \right]$$

**4.24** Escreva uma função no MATLAB que resolva um sistema de  $n$  equações lineares,  $[a][x] = [b]$ , com o método de Gauss-Jordan. O programa deve incluir a pivotação, isto é, a linha pivô deve ser trocada pela linha cujo elemento pivô possui o maior valor absoluto. Como nome de função e argumentos, use  $x = \text{GaussJordan}(a, b)$ , onde  $a$  é a matriz de coeficientes,  $b$  é a coluna de constantes no lado direito e  $x$  é a solução.

(a) Use a função GaussJordan para resolver o sistema:

$$\begin{aligned} 4x_1 + 3x_2 + 2x_3 + x_4 &= 1 \\ x_1 + 4x_2 + 3x_3 + 2x_4 &= 1 \\ 2x_1 + 3x_2 + 4x_3 + 3x_4 &= -1 \\ x_1 + 2x_2 + 3x_3 + 4x_4 &= -1 \end{aligned}$$

(b) Use a função GaussJordan para resolver o sistema:

$$\left[ \begin{array}{cccccc} 0 & 3 & 8 & -5 & -1 & 6 \\ 3 & 12 & -4 & 8 & 5 & -2 \\ 8 & 0 & 0 & 10 & -3 & 7 \\ 3 & 1 & 0 & 0 & 0 & 4 \\ 0 & 0 & 4 & -6 & 0 & 2 \\ 3 & 0 & 5 & 0 & 0 & -6 \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \right] = \left[ \begin{array}{c} 34 \\ 20 \\ 45 \\ 36 \\ 60 \\ 28 \end{array} \right]$$

**4.25** Escreva uma função no MATLAB que determine a inversa de uma matriz usando o método de Gauss-Jordan. Como nome de função e argumentos, use  $Ainv = \text{Inversa}(A)$ , onde  $A$  é a matriz a ser invertida e  $Ainv$  é a inversa da matriz. Use a função Inversa para calcular a inversa da matriz:

$$\begin{bmatrix} -0,04 & 0,04 & 0,12 \\ 0,56 & -1,56 & 0,32 \\ -0,24 & 1,24 & -0,28 \end{bmatrix}$$

**4.26** Escreva uma função no MATLAB que calcule a norma 1 de qualquer matriz. Como nome de função e argumentos, use  $N = \text{NormaUm}(A)$ , onde  $A$  é a matriz e  $N$  é o valor da norma. Use essa função para calcular a norma 1 de:

$$(a) A = \begin{bmatrix} 6 & 3 & 11 \\ 3 & 2 & 7 \\ 3 & 2 & 6 \end{bmatrix}$$

$$(b) \quad B = \begin{bmatrix} 6 & 3 & 11 & -1 & 2 \\ 3 & -2 & 7 & 0 & 4 \\ 3 & 2 & -6 & 5 & -3 \\ -5 & 7 & 1 & -4 & 0 \end{bmatrix}$$

**4.27** Escreva uma função no MATLAB que calcule a norma infinita de qualquer matriz. Como nome de função e argumentos, use  $N = \text{NormaInfinita}(A)$ , onde  $A$  é a matriz e  $N$  é o valor da norma. Use essa função para calcular a norma infinita de:

$$(a) \quad A = \begin{bmatrix} 6 & 3 & 11 \\ 3 & 2 & 7 \\ 3 & 2 & 6 \end{bmatrix}$$

$$(b) \quad B = \begin{bmatrix} 6 & 3 & 11 & -1 & 2 \\ 3 & -2 & 7 & 0 & 4 \\ 3 & 2 & -6 & 5 & -3 \\ -5 & 7 & 1 & -4 & 0 \end{bmatrix}$$

**4.28** Escreva uma função no MATLAB que calcule o número condição de uma matriz ( $n \times n$ ) usando a norma 1. Como nome de função e argumentos, use  $c = \text{NumeroCond}(A)$ , onde  $A$  é a matriz e  $c$  é o valor do número condição. Dentro da função, use as funções Inversa do Problema 4.25 e NormaUm do Problema 4.26. Use a função NumeroCond para calcular o número condição da matriz de coeficientes do Problema 4.25.

**4.29** Escreva uma função no MATLAB que determine o maior autovalor de uma matriz ( $n \times n$ ) usando o método da potência. Como nome de função e argumentos, use  $e = \text{MaxEig}(A)$ , onde  $A$  é a matriz e  $e$  é o valor do maior autovalor. Use a função MaxEig para calcular o maior autovalor da matriz do Problema 4.22. Verifique a resposta usando a função residente do MATLAB que determina os autovalores de uma matriz.

**4.30** Escreva uma função no MATLAB que determine o menor autovalor de uma matriz ( $n \times n$ ) usando o método da potência inverso. Como nome de função e argumentos, use  $e = \text{MinEig}(A)$ , onde  $A$  é a matriz e  $e$  é o valor do menor autovalor. Dentro da função MinEig, use a função Inversa, que foi escrita no Problema 4.25, para calcular a inversa da matriz  $A$ . Use a função MinEig para calcular o menor autovalor da matriz do Problema 4.22. Verifique a resposta usando as funções residentes do MATLAB para determinar os autovalores de uma matriz.

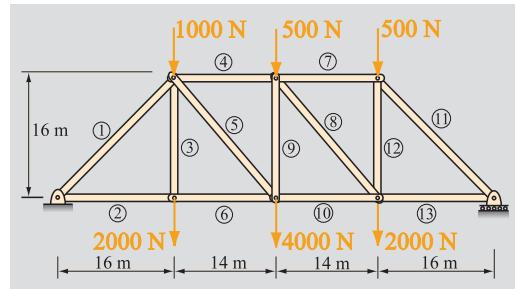
**4.31** Escreva uma função no MATLAB que determine todos os autovalores de uma matriz ( $n \times n$ ) usando o método QR de fatoração e iteração. Como nome de função e argumentos, use  $e = \text{TodosEig}(A)$ , onde  $A$  é a matriz e  $e$  é um vetor cujos elementos são os autovalores. Use a função TodosEig

para calcular os autovalores da matriz do Problema 4.22. Verifique a resposta usando a função residente do MATLAB que determina os autovalores de uma matriz.

### Problemas na matemática, na ciência e na engenharia

*Resolva os seguintes problemas no ambiente MATLAB. Use programas apresentados neste capítulo, programas desenvolvidos em problemas já resolvidos ou funções residentes do MATLAB.*

- 4.32** A força axial  $F_i$  em cada um dos 13 membros da treliça conectada por pinos mostrada na figura pode ser calculada com a solução do seguinte sistema de 13 equações:



$$F_2 + 0,7071F_1 = 0, \quad -F_2 + F_6 = 0, \quad F_3 - 2000 = 0$$

$$F_4 + 0,6585F_5 - 0,7071F_1 = 0$$

$$0,7071F_1 + F_3 + 0,7526F_5 + 1000 = 0$$

$$F_7 + 0,6585F_8 - F_4 = 0, \quad 0,7526F_8 + F_9 + 500 = 0$$

$$F_{10} - 0,6585F_5 - F_6 = 0, \quad F_9 + 0,7526F_5 - 4000 = 0$$

$$0,7071F_{11} - F_7 = 0, \quad 0,7071F_{11} + F_{12} + 500 = 0$$

$$F_{12} + 0,7526F_8 - 2000 = 0, \quad F_{13} + 0,7071F_{11} = 0$$

- (a) Resolva o sistema de equações usando a função `GaussPivotMaior` desenvolvida no Problema 4.23.
- (b) Resolva esse sistema usando a iteração de Gauss-Seidel. A solução converge se um vetor tentativa inicial com todos os elementos iguais a zero for utilizado?
- (c) Resolva esse sistema de equações usando a operação de divisão à esquerda do MATLAB.

- 4.33** A espectrometria da massa de uma amostra revela uma série de picos que representam as diversas massas dos íons que constituem essa amostra. A altura  $I_i$  de cada pico é influenciada pelas quantidades dos vários constituintes:

$$I_j = \sum_{i=1}^N C_{ij} n_j$$

onde  $C_{ij}$  é a contribuição de íons de espécies  $i$  com pico  $j$  e  $n_j$  é a quantidade de íons ou concentração de espécies  $j$ . Os coeficientes  $C_{ij}$  de cada pico são dados por:

Identidade do pico	Espécie				
	CH <sub>4</sub>	C <sub>2</sub> H <sub>4</sub>	C <sub>2</sub> H <sub>6</sub>	C <sub>3</sub> H <sub>6</sub>	C <sub>3</sub> H <sub>8</sub>
1	0.2	0.2	0.3	0.2	0.2
2	28	1	0	0	0.1
3		18	12	2.4	16
4			10	0	1
5				10	2
6					18

Se uma amostra produz um espectro de massas com picos  $I_1 = 3,4$ ,  $I_2 = 20,5$ ,  $I_3 = 170$ ,  $I_4 = 49$ ,  $I_5 = 39,8$  e  $I_6 = 96,3$ , determine as concentrações das diferentes espécies na amostra.

**4.34** A força axial  $F_i$  em cada um dos 21 membros da treliça conectada por pinos mostrada na figura pode ser calculada com a solução do seguinte sistema de 21 equações:

$$-F_1 - 0,7071F_3 = 0, \quad -F_2 + F_6 - 0,7071F_3 = 0$$

$$F_5 + 0,7071F_3 = 0, \quad -F_5 - 0,7071F_7 = 0$$

$$-F_4 + F_8 + 0,7071F_7 = 0, \quad -F_6 + F_{10} - 0,7071F_7 = 0$$

$$F_9 + 0,7071F_7 = 0, \quad 0,9806F_{14} + 0,7071F_{15} - 0,7071F_{11} - F_9 = 0$$

$$0,1961F_{14} + 0,7071F_{15} + 0,7071F_{11} + F_{13} - F_8 = 0$$

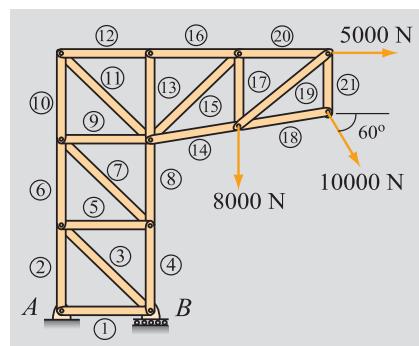
$$-F_{10} - 0,7071F_{11} = 0, \quad F_{12} + 0,7071F_{11} = 0$$

$$-F_{12} + F_{16} = 0, \quad F_{13} = 0,$$

$$0,9806F_{18} - 0,9806F_{14} + 0,7433F_{19} = 0, \quad 0,1961F_{18} - 0,1961F_{14} + F_{17} + 0,669F_{19} = 8000$$

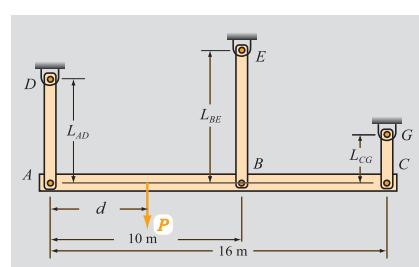
$$F_{20} - F_{16} - 0,7071F_{15} = 0, \quad -F_{17} - 0,7071F_{15} = 0, \quad -F_{20} - 0,7433F_{19} = -5000, \quad -F_{21} - 0,669F_{19} = 0$$

$$0,9806F_{11} = 10000\cos 60^\circ, \quad F_{21} - 0,1961F_{18} = 10000\sin 60^\circ$$



- (a) Resolva o sistema de equações usando a função GaussJordan desenvolvida no Problema 4.24.
- (b) Resolva esse sistema de equações usando a operação de divisão à esquerda do MATLAB.

**4.35** Uma barra rígida ABC é suportada por três barras conectadas por pinos, conforme mostrado. Uma força  $P = 40$  kN é aplicada na barra rígida em uma distância  $d$  a partir de A. As forças nas barras,  $F_{AD}$ ,  $F_{BE}$  e  $F_{CG}$ , podem ser determinadas com a solução do seguinte sistema de três equações:



$$F_{AD} + F_{BE} + F_{CG} = P, \quad 10F_{BE} + 16F_{CG} = d \cdot P$$

$$\frac{6L_{AD}}{E_{AD}A_{AD}}F_{AD} - \frac{16L_{BE}}{E_{BE}A_{BE}}F_{BE} + \frac{10L_{CG}}{E_{CG}A_{CG}}F_{CG} = 0$$

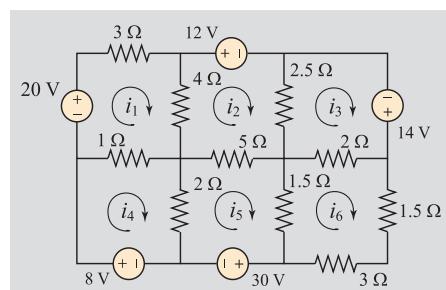
onde  $L$ ,  $E$  e  $A$  denotam o comprimento, o módulo elástico e a área da seção reta das barras, respectivamente. Uma vez calculada a força em cada uma das três barras, seu alongamento  $\delta$  pode ser determinado com o emprego da fórmula  $\delta = \frac{FL}{EA}$ .

Escreva um programa no MATLAB que determine as forças nas três barras e seu alongamento para  $0 \leq d \leq 16$ . O programa deve mostrar as três forças em função de  $d$  em um gráfico e o alongamento das barras em função de  $d$  em um segundo gráfico (dois gráficos na mesma página). Demais dados:

$$L_{BE} = 5 \text{ m}, \quad L_{CG} = 2 \text{ m}, \quad E_{AD} = 70 \text{ GPa}, \quad E_{BE} = 200 \text{ GPa}, \quad E_{CG} = 115 \text{ GPa} \text{ e} \\ A_{AD} = A_{BE} = A_{CG} = 5 \cdot 10^{-5} \text{ m}^2.$$

**4.36** Faz-se sorvete de forma tal que ele contenha uma quantidade desejada de gordura de leite, sólidos não-gordurosos do leite (msnf), água, açúcar, estabilizantes e emulsificadores. Suponha que 10 kg de sorvete devam ser feitos de forma a conter 18% de gordura, 15% de sacarose, 0,4 de estabilizantes e 1% de gema de ovos, usando os seguintes ingredientes: creme (contendo 35% de gordura), leite (contendo 3,5% de gordura), leite em pó desnatado (contendo 97% de msnf), sacarose, estabilizante e gema de ovos. Determine quanto de cada ingrediente deve ser usado (em kg). Assuma que 9% da nata do leite contenham msnf (dica: monte um sistema de três equações tendo como incógnitas a quantidade de leite em pó desnatado ( $x$ ), a quantidade de leite ( $y$ ) e a quantidade de creme ( $z$ ) usando (1) o fato de que a soma de todos os componentes deve ter 10 kg, (2) escrevendo uma equação para o conteúdo total de msnf e (3) escrevendo uma equação de balanço para a gordura).

**4.37** As correntes  $i_1, i_2, i_3, i_4, i_5$  e  $i_6$  no circuito mostrado podem ser determinadas a partir da solução do seguinte sistema de equações (obtido com a aplicação da lei de Kirchhoff):



$$8i_1 - 4i_2 - i_4 = 20, \quad -4i_1 + 11.5i_2 - 2.5i_3 - 5i_5 = -12$$

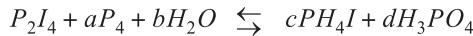
$$-2.5i_2 + 4.5i_3 - 2i_6 = 14, \quad -i_1 + 3i_4 - 2i_5 = 8$$

$$-5i_2 - 2i_4 + 8.5i_5 - 1.5i_6 = -30, \quad -2i_3 - 1.5i_5 + 8i_6 = 0$$

Resolva o sistema usando:

- (a) A função GaussJordan desenvolvida no Problema 4.24.
- (b) Funções residentes do MATLAB.

**4.38** Com o balanceamento da seguinte reação química, feito com a conservação do número de átomos de cada elemento entre reagentes e produtos



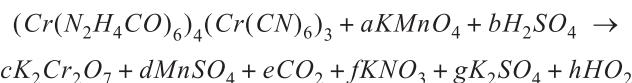
os coeficientes estequiométricos  $a, b, c$  e  $d$  são dados pela solução do seguinte sistema de equações:

$$\begin{bmatrix} -4 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & 4 & 3 \\ 0 & -1 & 0 & 4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

Resolva o sistema usando:

- (a) A função GaussJordan desenvolvida no Problema 4.24.
- (b) A divisão à esquerda do MATLAB.

**4.39** Com o balanceamento da seguinte reação química, feito com a conservação do número de átomos de cada elemento entre reagentes e produtos



os coeficientes estequiométricos  $a$  a  $h$  são dados pela solução do seguinte sistema de equações:

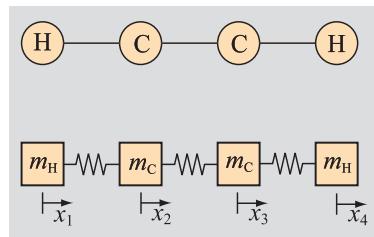
$$\begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -4 & -4 & 7 & 4 & 2 & 3 & 4 & 1 \\ -1 & 0 & 2 & 0 & 0 & 1 & 2 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{bmatrix} = \begin{bmatrix} 7 \\ 66 \\ 96 \\ 42 \\ 24 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Resolva o sistema usando:

- (a) A função GaussJordan desenvolvida no Problema 4.24.
- (b) A divisão à esquerda do MATLAB.

**4.40** Escreva uma função que determine os esforços principais e as direções das tensões principais referentes a um dado estado de esforços tridimensional. Como nome de função e argumentos, use  $[Ps \; Pd] = \text{PrinplTens}(S)$ , onde  $S$  é uma matriz  $(3 \times 3)$  com os valores do tensor de esforços,  $Ps$  é um vetor coluna com os valores dos esforços principais e  $Pd$  é uma matriz  $(3 \times 3)$  na qual cada linha contém um vetor unitário em uma direção principal. Use funções residentes do MATLAB. Use essa função para calcular os esforços e as direções principais referentes à matriz de estados dada no Problema 4.22:

$$\sigma_{ij} = \begin{bmatrix} 30 & 15 & 20 \\ 15 & 22 & 26 \\ 20 & 26 & 40 \end{bmatrix}$$



**4.41** A estrutura da molécula  $C_2H_2$  (acetileno) pode ser idealizada como quatro massas conectadas por duas molas (ver discussão no Problema 4.21). Com a aplicação da equação do movimento, o seguinte sistema de equações pode ser escrito para descrever a amplitude da vibração de cada átomo:

$$\begin{bmatrix} \frac{k_{CH}}{m_H} - \omega^2 & -\frac{k_{CH}}{m_H} & 0 & 0 \\ -\frac{k_{CH}}{m_C} & \frac{(k_{CH} + k_{CC})}{m_C} - \omega^2 & -\frac{k_{CC}}{m_C} & 0 \\ 0 & -\frac{k_{CC}}{m_C} & \frac{(k_{CH} + k_{CC})}{m_C} - \omega^2 & -\frac{k_{CH}}{m_C} \\ 0 & 0 & \frac{k_{CH}}{m_H} & \frac{k_{CH}}{m_H} - \omega^2 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

onde  $\omega$  é a freqüência,  $k_{CH} = 5,92 \times 10^2 \text{ kg/s}^2$  e  $k_{CC} = 15,8 \times 10^2 \text{ kg/s}^2$  são as constantes das molas representando as ligações C-H e C-C, respectivamente e  $m_H = 1 \text{ amu}$  e  $m_C = 12 \text{ amu}$  ( $\text{amu} = 1,6605 \times 10^{-27} \text{ kg}$ ) são as massas dos átomos.

- (a) Determine os autovalores (freqüências) e os comprimentos de onda correspondentes ( $\lambda = \frac{2\pi c}{\omega}$ , onde  $c = 3 \times 10^8 \text{ m/s}$  é a velocidade da luz)
- (b) Determine os autovetores correspondentes aos autovalores obtidos na letra (a). A partir dos autovetores, deduza o movimento relativo dos átomos (isto é, eles se afastam ou se aproximam uns dos outros?)

## Ajuste de Curvas e Interpolação

### Tópicos centrais

- |   |   |
|---|---|
| Ajuste de curvas com equações lineares (5.2)                          | Polinômios de Newton (5.5.2)  |
| Ajuste de curvas com a linearização de equações não-lineares (5.3)    | Interpolação por partes (spline) (5.6)  |
| Ajuste de curvas com polinômios quadráticos e de ordem superior (5.4) | Uso de funções residentes do MATLAB para realizar o ajuste de curvas e a interpolação (5.7) |
| Interpolação usando um único polinômio (5.5)                          | <b>Tópicos Complementares</b>   |
| Polinômios interpoladores de Lagrange (5.5.1)                         | Ajuste de curvas usando uma combinação linear de funções não-lineares (5.8)                 |

### 5.1 FUNDAMENTOS

Muitas observações científicas e de engenharia são feitas em experimentos nos quais grandezas físicas são medidas e gravadas. Tais registros são normalmente chamados de dados ou pontos experimentais. Por exemplo, a dureza de muitos metais depende do tamanho dos grãos que o compõem. O teste de espécimes com diferentes tamanhos de grãos resulta em um conjunto discreto de números ( $d$  – diâmetro médio do grão,  $\sigma_y$  – tensão limite de escoamento), conforme mostrado na Tabela 5-1.

**Tabela 5-1** Dados da dureza dos grãos em função de sua dimensão

<b>d (mm)</b>	0,005	0,009	0,016	0,025	0,040	0,062	0,085	0,110
<b><math>\sigma_y</math> (MPa)</b>	205	150	135	97	89	80	70	67

Às vezes, medições são realizadas e gravadas continuamente em dispositivos analógicos. No entanto, em muitos casos, especialmente nos últimos anos com o uso mais difundido de computadores, as grandezas medidas são digitalizadas e armazenadas como um conjunto de pontos discretos.

Cientistas e engenheiros podem usar dados experimentais de diferente maneiras. Freqüentemente, eles são usados no desenvolvimento ou na avaliação de fórmulas matemáticas (equações) que possam representá-los. Isso é feito com o traçado de curvas nas quais se assume uma forma de equação específica, com base ou não em algum tipo de teoria, e com a determinação dos parâmetros dessa equação de forma que as curvas traçadas representem da melhor forma possível o conjunto de dados. Às vezes, são usados dados experimentais na estimativa dos valores esperados entre os pontos medidos, um procedimento chamado de **interpolação**, ou na

predição de como esses dados poderiam ser estendidos além do intervalo no qual foram medidos, um procedimento chamado de **extrapolação**.

### Ajuste de curvas

O ajuste de curvas é um procedimento no qual uma fórmula matemática (equação) é usada para produzir uma curva que melhor represente um conjunto de dados.

O objetivo é encontrar uma equação que possa fazer isso de forma geral. Isso significa que a função não tem que fornecer o valor exato em cada ponto, mas sim representar o conjunto de dados de forma satisfatória como um todo. Por exemplo, a Fig. 5-1 mostra os pontos da Tabela 5-1 e uma curva descrita por uma função de potência ( $\sigma = Cd^m$ ) que melhor se ajusta a esse conjunto de dados. Pode-se observar que a curva reproduz a tendência geral dos dados, embora não seja exatamente igual a nenhum dos pontos medidos. O ajuste de curvas é tipicamente utilizado quando os valores dos dados medidos apresentam algum erro ou dispersão. Em

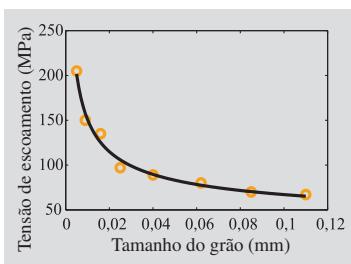


Figura 5-1 Ajuste de curvas.

geral, qualquer medição experimental apresenta erros ou incertezas inerentes, e a procura por uma curva que passe por todos os pontos medidos não traz consigo qualquer benefício. O procedimento de ajuste de curvas também é usado para determinar os valores dos parâmetros (coeficientes) nas equações. Isso pode ser feito com muitas funções diferentes e com polinômios de várias ordens.

### Interpolação

A interpolação é um procedimento empregado na estimativa de valores *entre* os pontos conhecidos de um conjunto de dados. Ela é feita primeiramente com a determinação de um polinômio que forneça o valor exato nos pontos conhecidos, e então com o uso desse polinômio para calcular valores entre esses pontos. Quando

um pequeno número de pontos está envolvido, um único polinômio pode ser suficiente para realizar a interpolação ao longo de todo o domínio de dados. Muitas vezes, no entanto, quando se tem um grande número de pontos, diferentes polinômios são usados nos intervalos entre os pontos. Esse processo é chamado de interpolação por partes, ou spline. Por exemplo, a Fig. 5-2 mostra um gráfico da relação tensão-deformação da borracha. As marcas mostram pontos experimentais medidos de forma muito precisa, e a curva em linha contínua foi obtida com o uso da interpolação por partes. Pode-se observar que a curva passa precisamente pelos pontos e fornece uma boa estimativa dos valores entre eles.

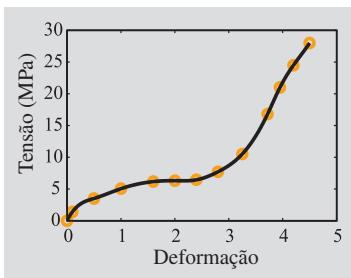


Figura 5-2 Interpolação.

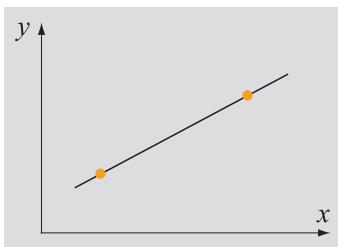
As três seções a seguir falam do ajuste de curvas. A Seção 5.2 descreve o

procedimento de ajuste de curvas usando a regressão linear com mínimos quadrados. Na Seção 5.3, o ajuste de curvas é feito com a linearização de funções não-lineares. Na Seção 5.4, tal ajuste é feito com polinômios de segunda ordem e de ordem superior. A interpolação é discutida nas duas seções seguintes. A Seção 5.5 mostra como encontrar a equação de um único polinômio que passe pelos pontos de um conjunto de dados (polinômios de Lagrange e de Newton), e a Seção 5.6

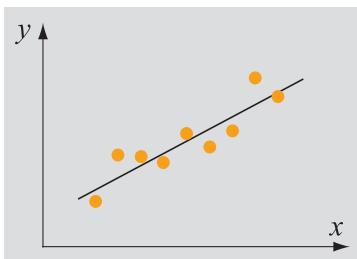
trata da interpolação por partes (spline), na qual diferentes polinômios são usados. A Seção 5.7 descreve ferramentas das quais o MATLAB dispõe para realizar o ajuste de curvas e a interpolação. Na Seção 5.8, o ajuste de curvas é feito de uma forma mais geral, sendo usada uma combinação linear de funções não-lineares.

## 5.2 AJUSTE DE CURVAS COM EQUAÇÕES LINEARES

O ajuste de curvas usando uma equação linear (polinômio de primeiro grau) é o processo pelo qual uma equação na forma:



**Figura 5-3** Dois pontos medidos.



**Figura 5-4** Vários pontos medidos.

$$y = a_1x + a_0 \quad (5.1)$$

é usada para promover o melhor ajuste de um conjunto de pontos. Isso é feito com a determinação das constantes  $a_1$  e  $a_0$  que fornecem o menor erro quando os pontos medidos são substituídos na Eq. (5.1). Se os dados compreenderem apenas dois pontos, as constantes podem ser determinadas de forma tal que a Eq. (5.1) forneça os valores exatos nos pontos. Graficamente, conforme mostrado na Fig. 5-3, isso significa que a linha reta correspondente à Eq. (5.1) passa pelos dois pontos.

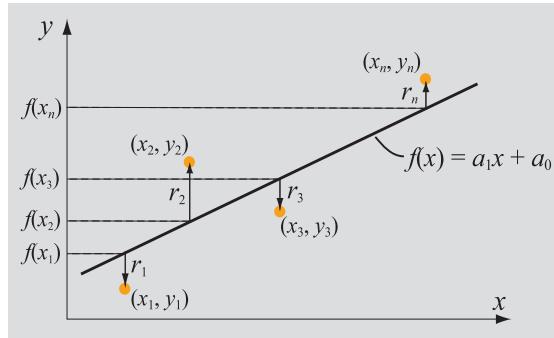
Quando os dados consistirem em mais de dois pontos, obviamente, uma linha reta não pode passar por todos os pontos. Neste caso, as constantes  $a_1$  e  $a_0$  são determinadas de tal forma que a linha reta promova o melhor ajuste como um todo, conforme ilustrado na Fig. 5-4.

O processo de obtenção das constantes que fornecem o melhor ajuste requer a definição do que vem a ser o melhor ajuste (Seção 5.2.1) e também de um procedimento matemático para a dedução do valor das constantes (Seção 5.2.2).

### 5.2.1 Medição da qualidade de um ajuste

Um critério que mede quanto bem uma função pode representar de forma aproximada um conjunto de dados é um número que quantifica a concordância geral entre os pontos pertencentes a esse conjunto de dados e a função utilizada. Um critério como esse é necessário por duas razões. Primeiramente, ele pode ser usado para comparar duas funções diferentes usadas no ajuste do mesmo conjunto de pontos. Em segundo lugar, e de forma ainda mais importante, tal critério pode ser usado para determinar os coeficientes da função que levem ao melhor ajuste. Isso é mostrado na Seção 5.2.2.

O ajuste entre um conjunto de dados e uma função linear aproximada é determinado primeiramente com o cálculo do erro, também chamado de resíduo, que é a diferença entre cada ponto pertencente ao conjunto de dados e o valor da função aproximada. Subseqüentemente, os resíduos são usados para calcular o erro total em todos os pontos. A Fig. 5-5 mostra o caso geral de uma função linear (linha reta) usada para ajustar um conjunto de  $n$  pontos. O resíduo  $r_i$  em um ponto  $(x_i, y_i)$  é a diferença entre o valor  $y_i$  do ponto medido e do valor da função  $f(x_i)$  usada para aproximar o conjunto de dados:

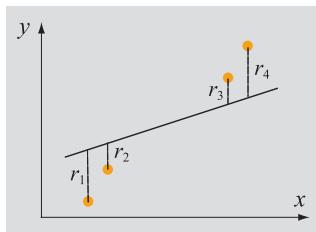


**Figura 5-5** Ajuste de curva usando uma equação linear.

$$r_i = y_i - f(x_i) \quad (5.2)$$

Um critério que mede quão bem a função aproximada é capaz de representar os dados fornecidos pode ser obtido com o cálculo do erro total  $E$  em termos dos resíduos. O erro global pode ser calculado de diferentes maneiras. Uma maneira simples é somar os resíduos de todos os pontos:

$$E = \sum_{i=1}^n r_i = \sum_{i=1}^n [y_i - (a_1 x_i + a_0)] \quad (5.3)$$

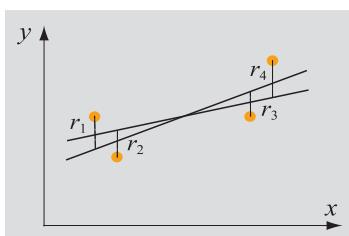


**Figura 5-6** Ajuste com erro nulo, de acordo com a Eq. (5.3).

$$\text{e } r_2 = -r_3.$$

Outra possibilidade é fazer com que o erro global  $E$  seja igual à soma do valor absoluto dos resíduos:

$$E = \sum_{i=1}^n |r_i| = \sum_{i=1}^n |y_i - (a_1 x_i + a_0)| \quad (5.4)$$



**Figura 5-7** Dois ajustes com o mesmo erro, de acordo com a Eq. (5.4).

Com essa definição, o erro total é sempre um número positivo, já que os resíduos não podem se cancelar. Um erro  $E$  menor na Eq. (5.4) indica um melhor ajuste. Essa medida pode ser usada para avaliar ou comparar ajustes propostos, mas não pode ser empregada na determinação das constantes da função que propicia o melhor ajuste. Isso ocorre porque essa medida não é única, o que significa que, para um mesmo conjunto de pontos, pode haver várias funções que resultem no mesmo erro total. Isso é mostrado na Fig. 5-7, onde o erro total  $E$  de acordo com a Eq. (5.4) é o mesmo para as duas retas usadas na aproximação.

Uma definição para o erro global  $E$  que fornece uma boa medida do erro total e que também pode ser usada para determinar uma única função linear que leve ao melhor ajuste (isto é, ao menor erro total) é obtida fazendo com que  $E$  seja igual à soma dos quadrados dos resíduos:

$$E = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n [y_i - (a_1 x_i + a_0)]^2 \quad (5.5)$$

Com essa definição, o erro global é sempre um número positivo (resíduos positivos e negativos não se cancelam). Além disso, maiores resíduos têm um efeito relativamente maior (peso) no erro total. Conforme já mencionado, a Eq. (5.5) pode ser usada para calcular os coeficientes  $a_1$  e  $a_0$  na função linear  $y = a_1 x + a_0$  que levem ao menor erro total. Isso é feito com o uso de um procedimento chamado regressão linear por mínimos quadrados, que é apresentado na próxima seção.

### 5.2.2 Regressão linear por mínimos quadrados

A regressão linear por mínimos quadrados é um procedimento no qual os coeficientes  $a_1$  e  $a_0$  da função linear  $y = a_1 x + a_0$  são determinados de tal forma que essa função leve ao melhor ajuste de um determinado conjunto de pontos. O melhor ajuste é definido como o menor erro total calculado com a soma dos quadrados dos resíduos de acordo com a Eq. (5.5).

Para um dado conjunto de  $n$  pontos  $(x_i, y_i)$ , o erro global calculado pela Eq. (5.5) é:

$$E = \sum_{i=1}^n [y_i - (a_1 x_i + a_0)]^2 \quad (5.6)$$

Como todos os valores  $x_i$  e  $y_i$  são conhecidos,  $E$  na Eq. (5.6) é uma função não-linear de duas variáveis,  $a_1$  e  $a_0$ . A função  $E$  tem um mínimo nos valores de  $a_1$  e  $a_0$  nos quais as derivadas parciais de  $E$  em relação a cada variável são iguais a zero. Calculando as derivadas parciais e as igualando a zero, obtém-se:

$$\frac{\partial E}{\partial a_0} = -2 \sum_{i=1}^n (y_i - a_1 x_i - a_0) = 0 \quad (5.7)$$

$$\frac{\partial E}{\partial a_1} = -2 \sum_{i=1}^n (y_i - a_1 x_i - a_0) x_i = 0 \quad (5.8)$$

As Eqs. (5.7) e (5.8) formam um sistema de duas equações lineares com incógnitas  $a_1$  e  $a_0$ , e podem ser escritas na forma:

$$na_0 + \left( \sum_{i=1}^n x_i \right) a_1 = \sum_{i=1}^n y_i \quad (5.9)$$

$$\left( \sum_{i=1}^n x_i \right) a_0 + \left( \sum_{i=1}^n x_i^2 \right) a_1 = \sum_{i=1}^n x_i y_i \quad (5.10)$$

A solução do sistema é:

$$a_1 = \frac{n \sum_{i=1}^n x_i y_i - \left( \sum_{i=1}^n x_i \right) \left( \sum_{i=1}^n y_i \right)}{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2} \quad (5.11)$$

$$a_0 = \frac{\left( \sum_{i=1}^n x_i^2 \right) \left( \sum_{i=1}^n y_i \right) - \left( \sum_{i=1}^n x_i y_i \right) \left( \sum_{i=1}^n x_i \right)}{n \sum_{i=1}^n x_i^2 - \left( \sum_{i=1}^n x_i \right)^2} \quad (5.12)$$

Como as Eqs. (5.11) e (5.12) contêm somas idênticas, é conveniente calculá-las primeiramente para então substituí-las nas equações. Para fazer isso, tais somas são definidas como:

$$S_x = \sum_{i=1}^n x_i, \quad S_y = \sum_{i=1}^n y_i, \quad S_{xy} = \sum_{i=1}^n x_i y_i, \quad S_{xx} = \sum_{i=1}^n x_i^2 \quad (5.13)$$

Com essas definições, as equações dos coeficientes  $a_1$  e  $a_0$  são:

$$a_1 = \frac{n S_{xy} - S_x S_y}{n S_{xx} - (S_x)^2} \quad a_0 = \frac{S_{xx} S_y - S_{xy} S_x}{n S_{xx} - (S_x)^2} \quad (5.14)$$

As Eqs. (5.14) fornecem os valores de  $a_1$  e  $a_0$  na função  $y = a_1 x + a_0$  que levam ao melhor ajuste dos  $n$  pontos do conjunto de dados. O Exemplo 5-1 mostra como usar as Eqs. (5.11) e (5.12) no ajuste de uma equação linear para representar um conjunto de dados experimentais.

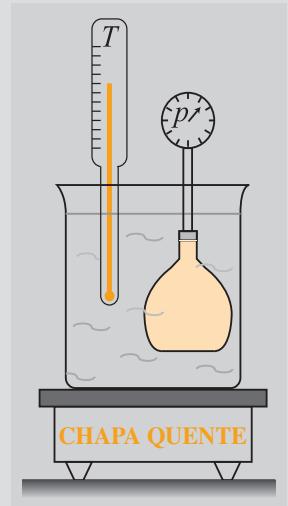
### Exemplo 5-1: Determinação da temperatura de zero absoluto

De acordo com a lei de Charles para um gás ideal em um volume constante, existe uma relação linear entre a pressão  $p$  e a temperatura  $T$ . No experimento mostrado na figura, um volume fixo de gás em um recipiente lacrado é submerso em água gelada ( $T = 0^\circ\text{C}$ ). A temperatura do gás é então elevada em incrementos de dez até alcançar  $T = 100^\circ\text{C}$ , o que é feito aquecendo-se a água, e a pressão do gás é medida em cada temperatura. Os dados obtidos no experimento são:

$T$ ( $^\circ\text{C}$ )	0	10	20	30	40	50	60
$p$ (atm.)	0,94	0,96	1,0	1,05	1,07	1,09	1,14
$T$ ( $^\circ\text{C}$ )	70	80	90	100			
$p$ (atm.)	1,17	1,21	1,24	1,28			

Extrapolie os dados e determine a temperatura de zero absoluto,  $T_0$ . Isso pode ser feito seguindo os seguintes passos:

- Trace um gráfico dos dados ( $p$  versus  $T$ ).



- (b) Use a regressão linear por mínimos quadrados para determinar uma função linear na forma  $p = a_1 T + a_0$  que melhor se ajuste aos pontos do conjunto de dados. Primeiramente, calcule os coeficientes manualmente usando apenas os quatro pontos a seguir: 0, 30, 70 e 100°C. Depois, escreva uma função no MATLAB que calcule os coeficientes da função linear para qualquer número de pontos e a utilize em toda a massa de dados fornecida no problema para determinar os coeficientes da função.
- (c) Trace um gráfico da função e estenda a reta (extrapole-a) até que ela cruze o eixo  $T$  horizontal. Esse ponto é uma estimativa da temperatura do zero absoluto. Determine o valor de  $T_0$  a partir da função.

## SOLUÇÃO

- (a) Cria-se um gráfico no MATLAB a partir dos dados fornecidos:

```
>> T = 0:10:100;
p = [0.94 0.96 1.0 1.05 1.07 1.09 1.14 1.17 1.21 1.24 1.28];
>> plot(T,p,'*r')
```

O gráfico obtido é mostrado à direita (os títulos dos eixos foram adicionados usando editor de gráficos do MATLAB). O gráfico mostra, conforme esperado, uma relação praticamente linear entre a pressão e a temperatura.

- (b) Cálculo manual da regressão por mínimos quadrados dos quatro pontos pertencentes ao conjunto de dados:

$(0; 0,94), (30; 1,05), (70; 1,17), (100; 1,28)$

Os coeficientes  $a_1$  e  $a_0$  da equação  $p = a_1 T + a_0$  que melhor se ajusta aos pontos são determinados usando a Eq. (5.14). As somas dadas pelas Eqs. (5.13) são calculadas primeiro:

$$S_x = \sum_{i=1}^4 x_i = 0 + 30 + 70 + 100 = 200 \quad S_y = \sum_{i=1}^4 y_i = 0,94 + 1,05 + 1,17 + 1,28 = 4,44$$

$$S_{xx} = \sum_{i=1}^4 x_i^2 = 0^2 + 30^2 + 70^2 + 100^2 = 15800$$

$$S_{xy} = \sum_{i=1}^4 x_i y_i = 0 \cdot 0,94 + 30 \cdot 1,05 + 70 \cdot 1,17 + 100 \cdot 1,28 = 241,4$$

A substituição das somas acima nas Eqs. (5.14) resulta em:

$$a_1 = \frac{nS_{xy} - S_x S_y}{nS_{xx} - (S_x)^2} = \frac{4 \cdot 241,4 - (200 \cdot 4,44)}{4 \cdot 15800 - 200^2} = 0,003345$$

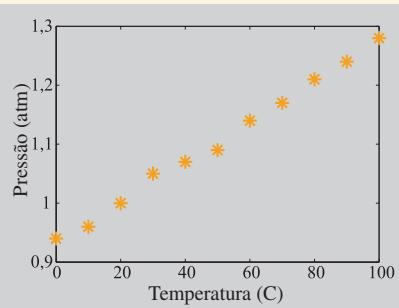
$$a_0 = \frac{S_{xx} S_y - S_{xy} S_x}{nS_{xx} - (S_x)^2} = \frac{15800 \cdot 4,44 - (241,4 \cdot 200)}{4 \cdot 15800 - 200^2} = 0,9428$$

A partir desses cálculos, a equação que melhor se ajusta aos dados é:  $p = 0,003345T + 0,9428$ .

A seguir, o problema é resolvido escrevendo-se uma função no MATLAB que calcula os coeficientes da função linear para qualquer número de pontos. As entradas da função são dois vetores com as coordenadas dos pontos. As saídas são os coeficientes  $a_1$  e  $a_0$  da equação linear, que são calculados com as Eqs. (5.14).

### Programa 5-1 Função definida pelo usuário. Regressão linear por mínimos quadrados.

```
function [a1,a0] = RegressaoLinear(x, y)
% RegressaoLinear calcula os coeficientes a1 e a0 da equação linear
```



```
% y = a1*x + a0 que melhor se ajusta aos n pontos do conjunto de dados.
```

```
% Variáveis de entrada:
```

```
% x    Vetor com as coordenadas x dos pontos.
```

```
% y    Vetor com as coordenadas y dos pontos.
```

```
% Variáveis de saída:
```

```
% a1   Coeficiente a1.
```

```
% a0   Coeficiente a0.
```

```
nx = length(x);
```

```
ny = length(y);
```

```
if nx ~ = ny
```

Verifica se os vetores x e y têm o mesmo número de elementos.

```
    disp('ERRO: O número de elementos em x deve ser o mesmo que em y.')
```

```
    a1 = 'Erro';
```

```
    a0 = 'Erro';
```

```
else
```

```
    Sx = sum(x);
```

```
    Sy = sum(y);
```

```
    Sxy = sum(x.*y);
```

```
    Sxx = sum(x.^2);
```

```
    a1 = (nx*Sxy - Sx*Sy)/(nx*Sxx - Sx.^2);
```

```
    a0 = (Sxx*Sy - Sxy*Sx)/(nx*Sxx - Sx.^2);
```

```
end
```

Se sim, o MATLAB exibe uma mensagem de erro e as constantes não são calculadas.

Calcula os termos com as somas nas Eqs. (5.13).

Calcula os coeficientes  $a_1$  e  $a_0$  nas Eqs. (5.14).

A função RegressaoLinear é então usada na janela de comandos do MATLAB para determinar a reta que melhor se ajusta aos pontos fornecidos no problema.

```
>> T = 0:10:100;
```

```
>> p = [0.94 0.96 1.0 1.05 1.07 1.09 1.14 1.17 1.21 1.24 1.28];
```

```
>> [a1, a0]=RegressaoLinear(T,p)
```

```
a1 =
```

```
0.0034
```

```
a0 =
```

```
0.9336
```

A equação que melhor se ajusta aos dados é:  
 $p = 0,0034T + 0,9336$

(c) A solução é obtida escrevendo o seguinte arquivo, que traça um gráfico da função e dos pontos e calcula o valor de  $T_0$  a partir dessa função.

```
T = 0:10:100;
```

```
p = [0.94 0.96 1.0 1.05 1.07 1.09 1.14 1.17 1.21 1.24 1.28];
```

```
Ti = [-300 100];
```

```
pi = 0.0034*Ti+0.9336;
```

```
plot(T,p,'r','markersize',12)
```

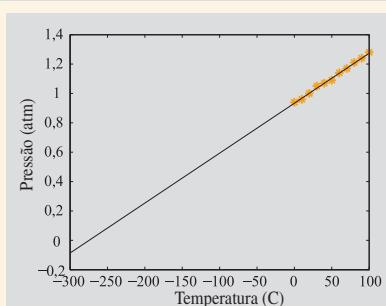
```
hold on
```

```
plot(Ti,pi,'k')
```

```
xlabel('Temperatura (C)','fontsize',20)
```

```
ylabel('Pressão (atm)','fontsize',20)
```

```
T0 = -0.9336/0.0034
```



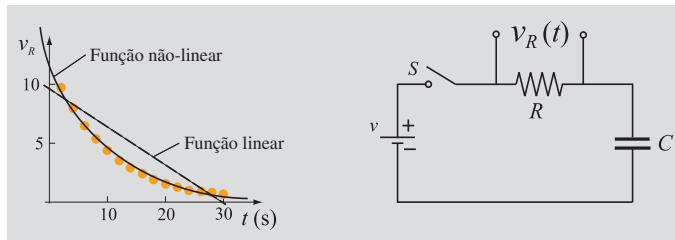
Quando esse programa é executado, aparece a figura ao lado e o valor da temperatura de zero absoluto é mostrada na janela de comandos do MATLAB, conforme mostrado abaixo:

```
T0 =
-274.5882
```

Esse resultado é próximo do valor encontrado nos livros, que é de  $-213,15^{\circ}\text{C}$ .

### 5.3 AJUSTE DE CURVAS COM A LINEARIZAÇÃO DE EQUAÇÕES NÃO-LINEARES

Muitas situações na ciência e na engenharia mostram que a relação entre as grandezas envolvidas não é linear. Por exemplo, a Fig. 5-8 mostra um gráfico de dados medidos em um experimento com um circuito  $RC$ . Neste experimento, a queda de tensão no resistor é medida em função do tempo a partir do instante em que a chave é fechada.



**Figura 5-8** Ajuste de curvas usando uma equação não-linear.

Os dados do experimento são listados no Exemplo 5-2. É óbvio, a partir do gráfico, que o uso de uma função não-linear leva a um ajuste muito melhor dos dados experimentais do que o uso de uma função linear.

Há muitos tipos de funções não-lineares. Esta seção apresenta funções não-lineares que podem ser escritas em uma forma tal que possibilite a determinação dos coeficientes que levam ao melhor ajuste com o emprego do método da regressão linear por mínimos quadrados. Exemplos de funções não-lineares usadas nesta seção no ajuste de curvas são:

$$y = bx^m \quad (\text{função de potência})$$

$$y = be^{mx} \text{ ou } y = b10^{mx} \quad (\text{função exponencial})$$

$$y = \frac{1}{mx + b} \quad (\text{função inversa})$$

Polinômios de segunda ordem ou de ordem superior também são funções não-lineares. O ajuste de curvas com esses polinômios é tratado em separado na Seção 5.4.

#### Escrevendo uma equação não-linear em uma forma linear

Para que a regressão linear possa ser utilizada, a equação não-linear de duas variáveis deve ser modificada de tal forma que a nova equação seja linear com termos

contendo as variáveis originais. Por exemplo, a função de potência  $y = bx^m$  pode ser linearizada calculando-se o logaritmo natural ( $\ln$ ) de ambos os lados:

$$\ln(y) = \ln(bx^m) = m \ln(x) + \ln(b) \quad (5.15)$$

Essa equação é linear com  $\ln(y)$  em termos de  $\ln(x)$ . A equação está na forma  $Y = a_1X + a_0$ , onde  $Y = \ln(y)$ ,  $a_1 = m$ ,  $X = \ln(x)$  e  $a_0 = \ln(b)$ :

$$\begin{array}{c} \ln(y) = m \ln(x) + \ln(b) \\ \hline Y = a_1 X + a_0 \end{array}$$

Isso significa que uma regressão linear por mínimos quadrados pode ser usada para fazer com que uma equação na forma  $y = bx^m$  se ajuste a um conjunto de pontos  $x_i, y_i$ . Isso é feito calculando  $a_1$  e  $a_0$  com as Eqs. (5.11) e (5.12) [ou (5.13) e (5.14)], com a substituição de  $y_i$  por  $\ln(y_i)$  e  $x_i$  por  $\ln(x_i)$ . Uma vez conhecidos  $a_1$  e  $a_0$ , as constantes  $b$  e  $m$  na equação exponencial são calculadas com:

$$m = a_1 \text{ e } b = e^{(a_0)} \quad (5.16)$$

Muitas outras equações não-lineares podem ser linearizadas de forma similar. A Tabela 5-2 lista várias dessas equações.

**Tabela 5-2** Linearização de equações não-lineares

Equação não-linear	Forma linear	Relação com $Y = a_1X + a_0$	Valores para a regressão linear por mínimos quadrados	Gráficos onde os dados medidos parecem se ajustar a uma linha reta
$y = bx^m$	$\ln(y) = m \ln(x) + \ln(b)$	$Y = \ln(y), X = \ln(x)$ $a_1 = m, a_0 = \ln(b)$	$\ln(x_i)$ e $\ln(y_i)$	Gráfico $y$ vs. $x$ em eixos $x$ e $y$ logarítmicos. Gráfico $\ln(y)$ vs. $\ln(x)$ em eixos $x$ e $y$ lineares.
$y = be^{mx}$	$\ln(y) = mx + \ln(b)$	$Y = \ln(y), X = x$ $a_1 = m, a_0 = \ln(b)$	$x_i$ e $\ln(y_i)$	Gráfico $y$ vs. $x$ em eixos $x$ linear e $y$ logarítmico. Gráfico $\ln(y)$ vs. $x$ em eixos $x$ e $y$ lineares.
$y = b10^{mx}$	$\log(y) = mx + \log(b)$	$Y = \log(y), X = x$ $a_1 = m, a_0 = \log(b)$	$x_i$ e $\log(y_i)$	Gráfico $y$ vs. $x$ em eixos $x$ linear e $y$ logarítmico. Gráfico $\log(y)$ vs. $x$ em eixos $x$ e $y$ lineares.
$y = \frac{1}{mx + b}$	$\frac{1}{y} = mx + b$	$Y = \frac{1}{y}, X = x$ $a_1 = m, a_0 = b$	$x_i$ e $1/y_i$	Gráfico $1/y$ vs. $x$ em eixos $x$ e $y$ lineares.
$y = \frac{mx}{b + x}$	$\frac{1}{y} = \frac{b}{mx} + \frac{1}{m}$	$Y = \frac{1}{y}, X = \frac{1}{x}$ $a_1 = \frac{b}{m}, a_0 = \frac{1}{m}$	$1/x_i$ e $1/y_i$	Gráfico $1/y$ vs. $1/x$ em eixos $x$ e $y$ lineares.

### Como escolher uma função não-linear apropriada para o ajuste de curvas

Um gráfico com os pontos do conjunto de dados pode fornecer uma indicação sobre a relação entre as grandezas. A determinação se a relação é linear ou não-linear pode ser feita com o traçado de um gráfico com eixos lineares. Se em tal gráfico os pontos se parecerem com uma linha reta, então a relação entre as grandezas é linear.

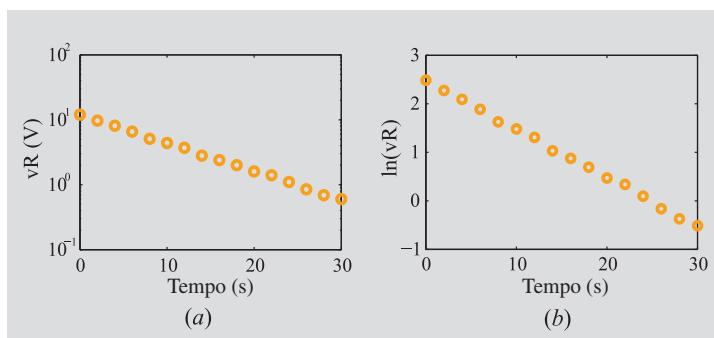
Um gráfico com eixos lineares no qual os pontos do conjunto de dados parecem se alinhar com uma curva indica uma relação não-linear entre as grandezas consideradas. A pergunta então é: qual função não-linear deve ser utilizada para fazer o ajuste da curva? Muitas vezes, na engenharia e na ciência, se conhece, a partir de alguma teoria fundamental, o fenômeno físico e a forma da equação matemática associada aos dados medidos. Por exemplo, o processo de carga de um capacitor mostrado na Fig. 5-8 é modelado com uma função exponencial. Se não se conhece qualquer forma possível para a equação, a escolha da função não-linear mais apropriada para fazer o ajuste de uma curva que represente os dados pode ser mais difícil.

Para um determinado conjunto de dados, é possível prever, até certo ponto, se uma função não-linear proposta tem potencial para fornecer um bom gráfico. Isso é feito com o traçado dos pontos medidos de uma maneira específica, verificando se esses pontos parecem formar uma linha reta. Para as funções listadas na Tabela 5-2, isso é mostrado na quinta (última) coluna da tabela. Para funções de potência e exponenciais, isso pode ser feito com o traçado dos dados usando diferentes combinações de eixos lineares e logarítmicos. Para todas as funções, isso pode ser feito com o traçado dos valores transformados do conjunto de dados em gráficos com eixos lineares.

Por exemplo, conforme mencionado anteriormente, espera-se que os dados do experimento mostrado na Fig. 5-8 se ajustem a uma função exponencial. Isso significa que o traçado de um gráfico da tensão  $v_R$  versus tempo  $t$  com eixo vertical logarítmico (para  $v_R$ ) e eixo horizontal linear (para  $t$ ) deve revelar o alinhamento dos pontos que compõem o conjunto de dados segundo uma linha reta. Outra opção é fazer um gráfico de  $\ln(v_R)$  versus  $t$  em eixos horizontal e vertical lineares, o que também se espera indicar o alinhamento dos pontos ao longo de uma linha reta. Ambos os gráficos são mostrados na Fig. 5-9. As figuras confirmam que os dados do experimento de carga do capacitor podem ser representados com o

O arquivo texto usado para gerar esse gráfico é:

```
tx=2:2:30;
vexp=[9.7 8.1 6.6 5.1 4.4 3.7 2.8
2.4 2.0 1.6 1.4 1.1 0.85 0.69 0.6];
vexpLOG=log(vexp)
subplot(1,2,1)
semilogy(tx,vexp,'or')
subplot(1,2,2)
plot(tx,vexpLOG,'or')
```



**Figura 5-9** (a) Gráfico de  $v_R$  vs.  $t$  utilizando eixo vertical logarítmico e eixo horizontal linear. (b) Gráfico de  $\ln(v_R)$  vs.  $t$  utilizando eixos vertical e horizontal lineares.

ajuste de uma curva descrita por uma função exponencial. O procedimento real de ajuste da curva é mostrado no Exemplo 5-2.

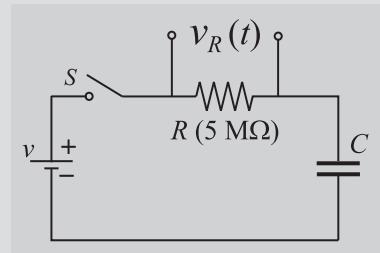
Outras considerações com relação à escolha da função não-linear adequada para o ajuste de uma curva são as seguintes:

- Funções exponenciais não podem passar pela origem.
- Funções exponenciais só são capazes de fazer o ajuste de dados nos quais todos os valores de  $y$  são positivos ou negativos.
- Funções logarítmicas não podem incluir  $x = 0$  ou valores negativos de  $x$ .
- Para função de potência  $y = 0$  quando  $x = 0$ .
- A equação inversa não pode incluir  $y = 0$ .

### Exemplo 5-2: Ajuste de curvas com a linearização de uma função não-linear

Um experimento com um circuito  $RC$  é usado para determinar a capacitância de um capacitor. No circuito, mostrado ao lado e na Fig. 5-8, um resistor de  $5 \text{ M}\Omega$  é conectado em série com o capacitor  $C$  e uma bateria. O experimento começa com o fechamento da chave e a medição da tensão  $v_R$  nos terminais do resistor em intervalos de 2 segundos, ao longo de 30 segundos. Os dados medidos no experimento são:

$t(\text{s})$	2	4	6	8	10	12	14	16	18
$v_R (\text{V})$	9,7	8,1	6,6	5,1	4,4	3,7	2,8	2,4	2,0
$t(\text{s})$	20	22	24	26	28	30			
$v_R (\text{V})$	1,6	1,4	1,1	0,85	0,69	0,6			



Teoricamente, a tensão no resistor em função do tempo é dada pela função exponencial:

$$v_R = ve^{(-t/(RC))} \quad (5.17)$$

Determine a capacitância do capacitor ajustando a função exponencial aos dados medidos.

### SOLUÇÃO

Mostrou-se na Fig. 5-9 que, conforme esperado, uma função exponencial se ajusta bem aos dados. O problema é resolvido primeiramente com a determinação das constantes  $b$  e  $m$  na função exponencial  $v = be^{mt}$  que melhor se ajusta aos dados. Isso é feito com a linearização dessa equação e com o uso da regressão linear por mínimos quadrados.

A regressão linear por mínimos quadrados é aplicada utilizando a função RegressaoLinear desenvolvida na solução do Exemplo 5-1. As entradas da função são os valores  $t_i$  e  $\ln(v_{ri})$ . De posse de  $b$  e  $m$ , o valor de  $C$  é determinado igualando-se os coeficientes no expoente de  $e$ :

$$\frac{-1}{RC} = m \text{ resolvendo para } C \text{ resulta em: } C = \frac{-1}{Rm} \quad (5.18)$$

Os cálculos são feitos com a execução do seguinte programa no MATLAB:

**Programa 5-2 Programa escrito em arquivo texto. Ajuste de curvas usando uma função não-linear.**

```

texp = 2:2:30;
vexp = [9.7 8.1 6.6 5.1 4.4 3.7 2.8 2.4 2.0 1.6 1.4 1.1 0.85 0.69 0.6]; Entra com os dados experimentais.
vexpLOG = log(vexp); Calcula  $\ln(y_i)$  dos dados (para uso na regressão linear).
R = 5E6; Calcula os coeficientes  $a_1$  e  $a_0$  com a função RegressaoLinear do Exemplo 5-1.
[a1,a0] = RegressaoLinear(texp, vexpLOG) Calcula  $b$ , sabendo que  $a_0 = \ln(b)$  (ver Tabela 5-2).
b = exp(a0) Calcula  $C$  usando a Eq. (5.18).
C = -1/(R*a1)
t = 0:0.5:30;
v = b*exp(a1*t); a1 é  $m$  na equação  $v = be^{mt}$ .
plot(t,v,texp,vexp,'or')

```

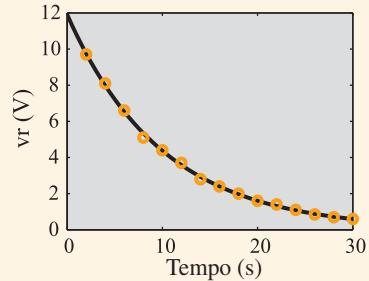
Quando o programa é executado, os valores a seguir são exibidos na janela de comandos do MATLAB. Além disso, mostra-se o seguinte gráfico com os pontos e a curva ajustada (os títulos dos gráficos foram adicionados posteriormente).

```

a1 =
-0.1002
a0 =
2.4776
b =
11.9131
C =
1.9968e-006

```

A capacidade é de aproximadamente 2  $\mu\text{F}$ .



## 5.4 AJUSTE DE CURVAS COM POLINÔMIOS QUADRÁTICOS E DE ORDEM SUPERIOR

### Fundamentos

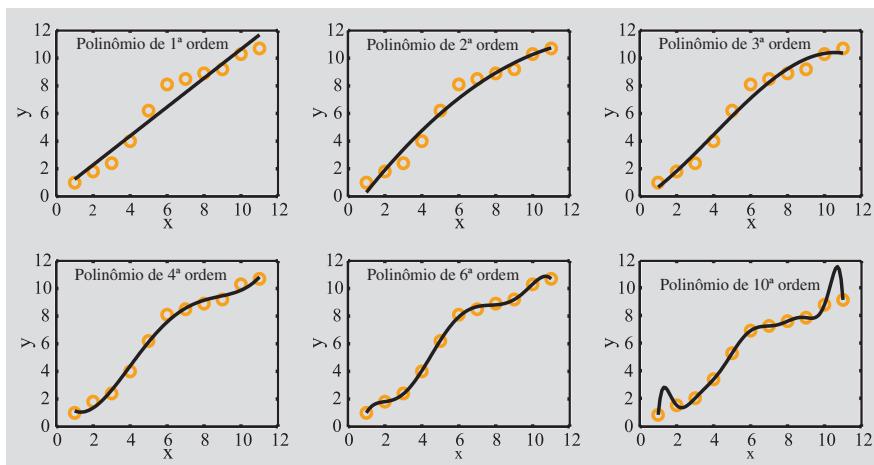
Polinômios são funções que têm a forma:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \quad (5.19)$$

Os coeficientes  $a_n, a_{n-1}, \dots, a_1, a_0$  são números reais, e  $n$ , que é um inteiro não-negativo, é o grau, ou ordem, do polinômio. O gráfico de um polinômio é uma curva. Um polinômio de primeira ordem é uma função linear, e seu gráfico é uma linha reta. Polinômios de ordem mais elevada são funções não-lineares, e seus gráficos são curvas. Um polinômio quadrático (de segunda ordem) gera uma curva (parábola) que é côncava para cima ou para baixo. Um polinômio de terceira ordem gera um ponto de inflexão que faz com que a curva associada seja côncava para cima (ou para baixo) em uma região, e côncava para baixo (ou para cima) em outra. Em geral, à medida que a ordem do polinômio aumenta, sua curva passa a ter mais “tortuosidades”.

Um determinado conjunto de dados contendo  $n$  pontos pode ser ajustado com polinômios de ordens diferentes até uma ordem ( $n - 1$ ). Conforme mostrado mais

adiante nesta seção, os coeficientes de um polinômio podem ser determinados de tal forma que ele forneça o melhor ajuste para um determinado conjunto de dados, o que é feito com a minimização do erro utilizando mínimos quadrados. A Fig. 5-10 mostra o ajuste de curvas com polinômios de diferentes ordens, tendo como referência um mesmo conjunto de 11 pontos. Os gráficos na figura mostram que, à medida que a ordem do polinômio aumenta, as curvas se aproximam dos pontos. Na realidade é possível ter um polinômio que passe exatamente por todos os pontos (em cada ponto, o valor do polinômio é igual ao valor do ponto). Para  $n$  pontos, isso ocorre com o polinômio de ordem  $(n - 1)$ . Na Fig. 5-10, isso corresponde a um polinômio de grau 10 (já que há 11 pontos).



**Figura 5-10** Ajuste de curvas de um mesmo conjunto de dados usando polinômios com diferentes graus.

A Fig. 5-10 mostra que o mesmo conjunto de dados pode ser ajustado por polinômios de ordem diferente. Dizer qual dos polinômios fornece o melhor ajuste não é fácil. Isso depende do tipo e da origem dos dados, de sua aplicação, e do propósito do ajuste. Por exemplo, se os dados não forem precisos (é possível haver grandes erros quando se mede uma grandeza), não faz muito sentido utilizar um polinômio de ordem elevada que siga os pontos de forma próxima. Por outro lado, se os valores dos pontos que constituem o conjunto de dados forem muito precisos e a curva ajustada for usada para representá-los, a utilização de um polinômio de ordem mais elevada pode ser apropriada. Entretanto, conforme explicado na **nota importante** a seguir, não se recomenda o uso de polinômios de ordem elevada no ajuste de curvas.

### Nota importante

Conforme já mencionado, para qualquer número  $n$  de pontos pertencentes a um conjunto de dados, é possível deduzir um polinômio de ordem  $n - 1$  que passe exatamente por todos os pontos. Entretanto, quando muitos pontos estão envolvidos, esse polinômio possui um grau elevado. Embora um polinômio de ordem elevada forneça os valores exatos em todos os pontos, muitas vezes ele apresenta um desvio significativo *entre* alguns dos pontos. Isso pode ser visto no gráfico

que ilustra o polinômio de décima ordem na Fig. 5-10, onde entre os dois primeiros pontos e entre os dois últimos pontos a curva do polinômio se afasta e não segue a tendência geral dos dados. Isso significa que, mesmo que um polinômio de ordem elevada forneça os valores exatos em todos os pontos, ele não pode ser usado de forma confiável para a interpolação ou a extrapolação dos dados. Métodos apropriados para a interpolação são descritos nas Seções 5.5 e 5.6.

### Regressão polinomial

A regressão polinomial é um procedimento usado na determinação dos coeficientes de um polinômio de segundo grau, ou de ordem maior, de forma que esse polinômio produza o melhor ajuste de um determinado conjunto de dados. Como na regressão linear, a dedução das equações utilizadas para determinar os coeficientes se baseia na minimização do erro total de acordo com a Eq. (5.5).

Se o polinômio de ordem  $m$  usado no ajuste da curva é:

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0 \quad (5.20)$$

então, para um dado conjunto de  $n$  pontos  $(x_i, y_i)$  ( $m$  é menor que  $n - 1$ ), o erro total calculado pela Eq. (5.5) é:

$$E = \sum_{i=1}^n [y_i - (a_m x_i^m + a_{m-1} x_i^{m-1} + \dots + a_1 x_i + a_0)]^2 \quad (5.21)$$

Como todos os valores  $x_i$  e  $y_i$  que constituem o conjunto de dados são conhecidos,  $E$  na Eq. (5.21) é uma função não-linear das  $m + 1$  variáveis (os coeficientes  $a_0$  a  $a_m$ ). A função  $E$  tem um mínimo nos valores de  $a_0$  a  $a_m$  nos quais as derivadas parciais de  $E$  em relação a cada uma das variáveis são iguais a zero. Calculando as derivadas parciais de  $E$  na Eq. (5.21) e as igualando a zero, obtém-se um conjunto de  $m + 1$  equações lineares para os coeficientes. Para simplificar a apresentação, a dedução para o caso  $m = 2$  (polinômio quadrático) é mostrada em detalhe. Neste caso, a Eq. (5.21) é:

$$E = \sum_{i=1}^n [y_i - (a_2 x_i^2 + a_1 x_i + a_0)]^2 \quad (5.22)$$

Calculando as derivadas parciais em relação a  $a_0$ ,  $a_1$  e  $a_2$  e igualando os resultados a zero, obtém-se:

$$\frac{\partial E}{\partial a_0} = -2 \sum_{i=1}^n (y_i - a_2 x_i^2 - a_1 x_i - a_0) = 0 \quad (5.23)$$

$$\frac{\partial E}{\partial a_1} = -2 \sum_{i=1}^n (y_i - a_2 x_i^2 - a_1 x_i - a_0) x_i = 0 \quad (5.24)$$

$$\frac{\partial E}{\partial a_2} = -2 \sum_{i=1}^n (y_i - a_2 x_i^2 - a_1 x_i - a_0) x_i^2 = 0 \quad (5.25)$$

As Eqs. (5.23) a (5.25) formam um sistema de três equações lineares em função das incógnitas  $a_0$ ,  $a_1$  e  $a_2$ , que pode ser resrito na forma:

$$na_0 + \left( \sum_{i=1}^n x_i \right) a_1 + \left( \sum_{i=1}^n x_i^2 \right) a_2 = \sum_{i=1}^n y_i \quad (5.26)$$

$$\left( \sum_{i=1}^n x_i \right) a_0 + \left( \sum_{i=1}^n x_i^2 \right) a_1 + \left( \sum_{i=1}^n x_i^3 \right) a_2 = \sum_{i=1}^n x_i y_i \quad (5.27)$$

$$\left( \sum_{i=1}^n x_i^2 \right) a_0 + \left( \sum_{i=1}^n x_i^3 \right) a_1 + \left( \sum_{i=1}^n x_i^4 \right) a_2 = \sum_{i=1}^n x_i^2 y_i \quad (5.28)$$

A solução do sistema de equações (5.26)-(5.28) fornece os valores dos coeficientes  $a_0$ ,  $a_1$  e  $a_2$  do polinômio  $y = a_2 x^2 + a_1 x + a_0$  que melhor se ajusta aos  $n$  pontos  $(x_i, y_i)$ .

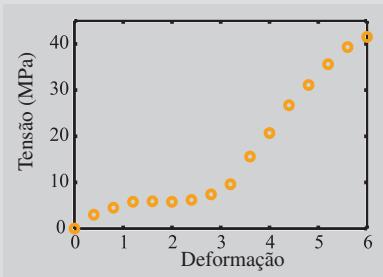
Os coeficientes de polinômios de ordem superior são deduzidos da mesma forma. Para um polinômio de ordem  $m$ , as Eqs. (5.26)-(5.28) são estendidas para formar um conjunto de  $m+1$  equações lineares em função dos  $m+1$  coeficientes. As equações para um polinômio de quarta ordem são mostradas no Exemplo 5-3.

### Exemplo 5-3: Uso da regressão polinomial no ajuste de uma curva tensão-deformação

Realiza-se um teste de tensão para determinar o comportamento tensão-deformação da borracha. Os dados coletados no teste são mostrados na figura e seus valores são fornecidos a seguir. Determine o polinômio de quarta ordem que faça o melhor ajuste dos pontos. Trace um gráfico que inclua esses pontos e a curva correspondente ao polinômio.

Deformação $\epsilon$	0	0,4	0,8	1,2	1,6	2,0	2,4
Tensão $\sigma$ (MPa)	0	3,0	4,5	5,8	5,9	5,8	6,2

Deformação $\epsilon$	2,8	3,2	3,6	4,0	4,4	4,8	5,2	5,6	6,0
Tensão $\sigma$ (MPa)	7,4	9,6	15,6	20,7	26,7	31,1	35,6	39,3	41,5



### SOLUÇÃO

Um polinômio de quarta ordem pode ser escrito como:

$$f(x) = a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0 \quad (5.29)$$

O ajuste de uma curva para representar os 16 pontos medidos é feito com o emprego da regressão polinomial. Os valores dos cinco coeficientes  $a_0$ ,  $a_1$ ,  $a_2$ ,  $a_3$  e  $a_4$  são obtidos com a solução de um sistema com cinco equações lineares. Estas cinco equações podem ser escritas a partir da extensão das Eqs. (5.26)-(5.28).

$$na_0 + \left( \sum_{i=1}^n x_i \right) a_1 + \left( \sum_{i=1}^n x_i^2 \right) a_2 + \left( \sum_{i=1}^n x_i^3 \right) a_3 + \left( \sum_{i=1}^n x_i^4 \right) a_4 = \sum_{i=1}^n y_i \quad (5.30)$$

$$\left( \sum_{i=1}^n x_i \right) a_0 + \left( \sum_{i=1}^n x_i^2 \right) a_1 + \left( \sum_{i=1}^n x_i^3 \right) a_2 + \left( \sum_{i=1}^n x_i^4 \right) a_3 + \left( \sum_{i=1}^n x_i^5 \right) a_4 = \sum_{i=1}^n x_i y_i \quad (5.31)$$

$$\left( \sum_{i=1}^n x_i^2 \right) a_0 + \left( \sum_{i=1}^n x_i^3 \right) a_1 + \left( \sum_{i=1}^n x_i^4 \right) a_2 + \left( \sum_{i=1}^n x_i^5 \right) a_3 + \left( \sum_{i=1}^n x_i^6 \right) a_4 = \sum_{i=1}^n x_i^2 y_i \quad (5.32)$$

$$\left( \sum_{i=1}^n x_i^3 \right) a_0 + \left( \sum_{i=1}^n x_i^4 \right) a_1 + \left( \sum_{i=1}^n x_i^5 \right) a_2 + \left( \sum_{i=1}^n x_i^6 \right) a_3 + \left( \sum_{i=1}^n x_i^7 \right) a_4 = \sum_{i=1}^n x_i^3 y_i \quad (5.33)$$

$$\left( \sum_{i=1}^n x_i^4 \right) a_0 + \left( \sum_{i=1}^n x_i^5 \right) a_1 + \left( \sum_{i=1}^n x_i^6 \right) a_2 + \left( \sum_{i=1}^n x_i^7 \right) a_3 + \left( \sum_{i=1}^n x_i^8 \right) a_4 = \sum_{i=1}^n x_i^4 y_i \quad (5.34)$$

Os cálculos e o traçado do gráfico são feitos no MATLAB utilizando o programa cujos passos são os seguintes:

**Passo 1:** Crie os vetores  $x$  e  $y$  contendo os pontos medidos.

**Passo 2:** Crie um vetor  $xsum$  cujos elementos são os termos das somas das potências de  $x_i$ . Por exemplo, o quarto elemento desse vetor é:  $xsum(4) = \sum_{i=1}^n x_i^4$

**Passo 3:** Coloque o sistema de cinco equações lineares (Eqs. (5.30)-(5.34)) na forma  $[a][p] = [b]$ , onde  $[a]$  é a matriz com os termos das somas das potências de  $x_i$ ,  $[p]$  é o vetor das incógnitas (os coeficientes do polinômio) e  $[b]$  é um vetor contendo os termos das somas no lado direito das Eqs. (5.30)-(5.34).

**Passo 4:** Resolva o sistema de cinco equações lineares  $[a][p] = [b]$  e obtenha  $[p]$  usando a divisão à esquerda do MATLAB. A solução é um vetor com os coeficientes do polinômio de quarta ordem que propicia o melhor ajuste do conjunto de dados.

**Passo 5:** Trace um gráfico com os pontos medidos e com a curva do polinômio de ajuste.

### Programa 5-3 Programa escrito em arquivo texto. Ajuste de curvas usando regressão polinomial.

```

clear all
x = 0:0.4:6; Atribui os dados experimentais aos vetores x e y.
y = [0 3 4.5 5.8 5.9 5.8 6.2 7.4 9.6 15.6 20.7 26.7 31.1 35.6 39.3 41.5];
n = length(x); n é o número de pontos.
m = 4; m é a ordem do polinômio.
for i = 1:2*m
    xsum(i) = sum(x.^i);
end Define um vetor com os termos das somas das potências de  $x_i$ .
% Início do Passo 3
a(1,1) = n;
b(1,1) = sum(y);
for j = 2:m + 1
    a(1,j) = xsum(j - 1);
end
for i = 2:m + 1
    for j = 1:m + 1
        a(i,j) = xsum(j + i - 2);
    end
    b(i,1) = sum(x.^i-1).*y;
end
% Passo 4
p = (a\b)' Resolve o sistema  $[a][p] = [b]$  para  $[p]$ . Transpõe a solução para que  $[p]$  se torne um vetor linha.

```

```

for i = 1:m + 1
    Pcoef(i) = p(m + 2 - i);
end
epsilon = 0:0.1:6;
stressfit = polyval(Pcoef,epsilon);
plot(x,y,'ro',epsilon,stressfit,'k','linewidth',2)
xlabel('Deformação','fontsize',20)
ylabel('Tensão (MPa)','fontsize',20)

```

Cria um novo vetor para os coeficientes do polinômio, a ser usado na função `polyval` do MATLAB (veja nota no final do exemplo).

Define um vetor de deformações a ser usado no traçado do polinômio.

Tensão calculada pelo polinômio.

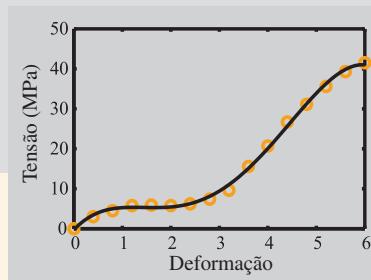
Traça um gráfico com os pontos e o polinômio.

Quando o programa é executado, a solução  $[p]$  é mostrada na janela de comandos do MATLAB. Além disso, também é mostrado o gráfico que contém os pontos do conjunto de dados e a curva obtida com o polinômio.

```
p =
-0.2746 12.8780 -10.1927 3.1185 -0.2644
```

O polinômio obtido após o ajuste é:

$$f(x) = (-0,2746)x^4 + 12,878x^3 - 10,1927x^2 + 3,1185x - 0,2644.$$

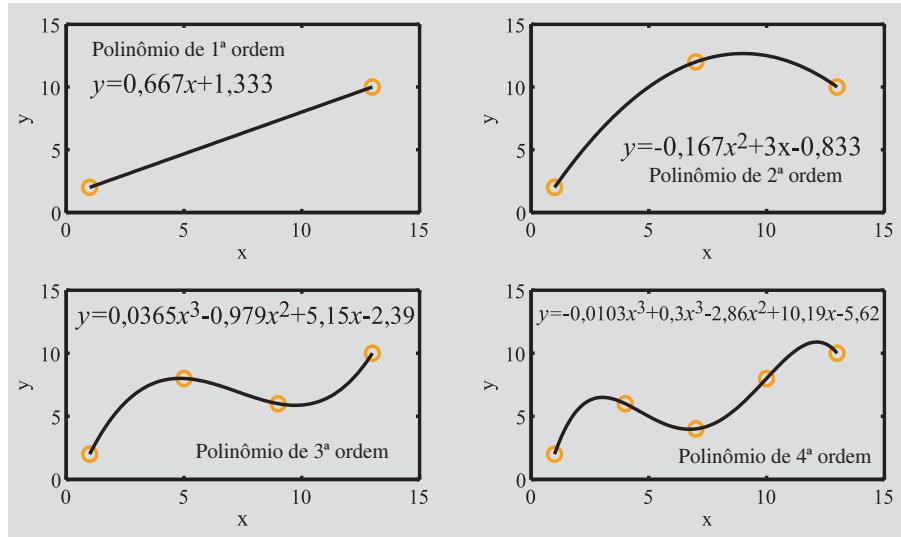


Nota: No MATLAB, um polinômio é representado por um vetor cujos elementos são os coeficientes desse polinômio. O primeiro elemento do vetor é o coeficiente de ordem mais elevada do polinômio, e seu último elemento é o termo  $a_0$ .

## 5.5 INTERPOLAÇÃO USANDO UM ÚNICO POLINÔMIO

A interpolação é um procedimento no qual uma fórmula matemática é usada para fornecer o valor exato dos pontos pertencentes a um conjunto de dados e um valor estimado *entre* esses pontos. Esta seção mostra como isso é feito empregando-se um único polinômio, independentemente do número de pontos envolvidos. Conforme mencionado na seção anterior, para qualquer número  $n$  de pontos existe um polinômio de ordem  $n - 1$  que passa por todos esses pontos. Para dois pontos, esse polinômio é de primeira ordem (uma linha reta os conectando). Para três pontos, o polinômio é de segunda ordem (uma parábola os conectando), e assim por diante. Isso é ilustrado na Fig. 5-11, que mostra polinômios de primeira, segunda, terceira e quarta ordem conectando, respectivamente, dois, três, quatro e cinco pontos.

Uma vez determinado o polinômio, ele pode ser usado para estimar os valores de  $y$  entre os pontos conhecidos, o que é feito simplesmente com a substituição da coordenada  $x$  desejada no polinômio. A interpolação usando um único polinômio fornece bons resultados apenas para um pequeno número de pontos. Para um grande número de pontos, a ordem do polinômio deve ser elevada e, embora esse polinômio passe por todos os pontos, ele pode apresentar um desvio significativo fora deles. Isso foi mostrado na Fig. 5-10 para um polinômio de grau 10 e é mostrado mais adiante na Fig. 5-17, onde um polinômio de 15<sup>a</sup> ordem é usado para interpolar um conjunto de dados formado por 16 pontos. Conseqüentemente, a interpolação com apenas um polinômio pode não ser apropriada para um número



**Figura 5-11** Polinômios de várias ordens.

elevado de pontos. Neste caso específico, melhores resultados podem ser obtidos com o uso da interpolação por partes (spline) (tratada na Seção 5.6), na qual diferentes polinômios de ordem inferior são usados na interpolação de diferentes pontos pertencentes a um mesmo conjunto de dados.

Para um dado conjunto de  $n$  pontos, apenas um (único) polinômio de ordem  $m$  ( $m = n - 1$ ) passa exatamente por todos os pontos. Esse polinômio, no entanto, pode ser escrito de diferentes formas matemáticas. Esta seção mostra como deduzir três dessas formas (padrão, Lagrange e Newton). As diferentes formas matemáticas são adequadas a diferentes circunstâncias.

A forma padrão de um polinômio de ordem  $m$  é:

$$f(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0 \quad (5.35)$$

Os coeficientes nesta forma são determinados com a solução de um sistema de  $m + 1$  equações lineares. As equações são obtidas escrevendo-se o polinômio explicitamente em cada ponto (substituindo cada ponto no polinômio). Por exemplo, os cinco pontos ( $n = 5$ ) dados pelo polinômio de quarto grau ( $m = 4$ ) mostrado na Fig. (5-11) são: (1, 2), (4, 6), (7, 4), (10, 8) e (13, 10). O uso da Eq. (5.35) em cada um desses pontos fornece o seguinte sistema de cinco equações com incógnitas  $a_0, a_1, a_2, a_3$  e  $a_4$ :

$$\begin{aligned} a_4 1^4 + a_3 1^3 + a_2 1^2 + a_1 1 + a_0 &= 2 \\ a_4 4^4 + a_3 4^3 + a_2 4^2 + a_1 4 + a_0 &= 6 \\ a_4 7^4 + a_3 7^3 + a_2 7^2 + a_1 7 + a_0 &= 4 \\ a_4 10^4 + a_3 10^3 + a_2 10^2 + a_1 10 + a_0 &= 8 \\ a_4 13^4 + a_3 13^3 + a_2 13^2 + a_1 13 + a_0 &= 10 \end{aligned} \quad (5.36)$$

A solução desse sistema de equações fornece os valores dos coeficientes. Uma solução da Eq. (5.36) usando o MATLAB é a seguinte:

```

>> a = [1 1 1 1 1; 4^4 4^3 4^2 4 1; 7^4 7^3 7^2 7 1; 10^4 10^3 10^2 10 1; 13^4 13^3 13^2 13
1]
a =
      1      1      1      1      1
    256     64     16      4      1
   2401    343     49      7      1
  10000   1000    100     10     1
  28561   2197    169     13     1
>> b = [2; 6; 4; 8; 10]
>> A = a\b
A =
-0.0103
0.3004
-2.8580
10.1893
-5.6214

```

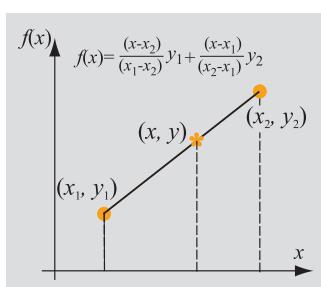
O polinômio que corresponde a estes coeficientes é  
 $y = -0,0103x^4 + 0,3x^3 - 2,86x^2 + 10,19x - 5,62$   
(ver Fig. 5-11).

Na prática, a solução de um sistema de equações não é eficiente, especialmente quando polinômios de ordem mais elevada estão envolvidos. Além disso, a matriz dos coeficientes é freqüentemente mal condicionada (ver Seção 4.11).

É possível escrever o polinômio em outras formas de uso mais fácil. Duas dessas formas, as formas de Lagrange e de Newton, são descritas nas duas subseções a seguir.

### 5.5.1 Polinômios interpoladores de Lagrange

Os polinômios interpoladores de Lagrange formam uma classe específica de polinômios que podem ser usados para fazer o ajuste de um determinado conjunto de dados simplesmente a partir dos valores dos pontos. Os polinômios podem ser escritos diretamente, e os coeficientes são determinados sem a necessidade de nenhum cálculo preliminar.



**Figura 5-12** Polinômio de Lagrange de primeira ordem.

Para dois pontos,  $(x_1, y_1)$  e  $(x_2, y_2)$ , o polinômio de Lagrange de primeira ordem (Fig. 5-12) tem a forma:

$$f(x) = y = a_1(x - x_2) + a_2(x - x_1) \quad (5.37)$$

Substituindo os dois pontos na Eq. (5.37), obtém-se:

$$y_1 = a_1(x_1 - x_2) + a_2(x_1 - x_1) \quad \text{ou} \quad a_1 = \frac{y_1}{(x_1 - x_2)} \quad (5.38)$$

e

$$y_2 = a_1(x_2 - x_2) + a_2(x_2 - x_1) \quad \text{ou} \quad a_2 = \frac{y_2}{(x_2 - x_1)} \quad (5.39)$$

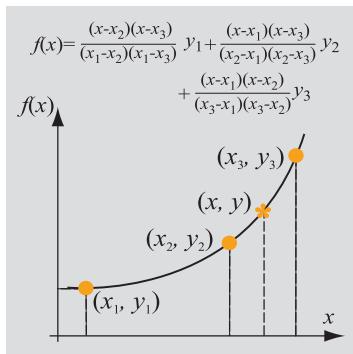
Substituindo os coeficientes  $a_1$  e  $a_2$  de volta na Eq. (5.37), obtém-se:

$$f(x) = \frac{(x-x_2)}{(x_1-x_2)}y_1 + \frac{(x-x_1)}{(x_2-x_1)}y_2 \quad (5.40)$$

A Eq. (5.40) é uma função linear de  $x$  (equação de uma linha reta conectando os dois pontos). É fácil ver que, se  $x = x_1$  for substituído na Eq. (5.40), o valor do polinômio é igual a  $y_1$ , e se  $x = x_2$  for substituído na Eq. (5.40), o valor do polinômio é igual a  $y_2$ . A substituição de um valor de  $x$  entre os pontos fornece um valor interpolado de  $y$ . A Eq. (5.40) pode também ser escrita na forma padrão  $f(x) = a_1x + a_0$ :

$$f(x) = \frac{(y_2-y_1)}{(x_2-x_1)}x + \frac{x_2y_1 - x_1y_2}{(x_2-x_1)} \quad (5.41)$$

Para três pontos,  $(x_1, y_1)$ ,  $(x_2, y_2)$  e  $(x_3, y_3)$ , o polinômio de Lagrange de segunda ordem (Fig. 5-13) tem a forma:



**Figura 5-13** Polinômio de Lagrange de segunda ordem.

$$f(x) = y = a_1(x-x_2)(x-x_3) + a_2(x-x_1)(x-x_3) + a_3(x-x_1)(x-x_2) \quad (5.42)$$

Uma vez determinados os coeficientes de forma que o coeficiente passe pelos três pontos, o polinômio é:

$$f(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3 \quad (5.43)$$

A Eq. (5.43) é uma função quadrática de  $x$ . Quando a coordenada  $x_1$ ,  $x_2$  ou  $x_3$  de um dos três pontos é substituída na Eq. (5.43), o valor do polinômio é igual a  $y_1$ ,  $y_2$  ou  $y_3$ , respectivamente. Isso ocorre porque o coeficiente na frente do termo  $y_i$  correspondente é igual a 1 e o coeficiente dos outros dois termos é igual a zero.

Seguindo o formato dos polinômios nas Eqs. (5.41) e (5.43), a fórmula geral de um polinômio de Lagrange de ordem  $n-1$  que passe por  $n$  pontos  $(x_1, y_1)$ ,  $(x_2, y_2)$ , ...,  $(x_n, y_n)$  é:

$$\begin{aligned} f(x) = & \frac{(x-x_2)(x-x_3)\dots(x-x_n)}{(x_1-x_2)(x_1-x_3)\dots(x_1-x_n)}y_1 + \frac{(x-x_1)(x-x_3)\dots(x-x_n)}{(x_2-x_1)(x_2-x_3)\dots(x_2-x_n)}y_2 + \\ & \dots + \frac{(x-x_1)(x-x_2)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_1)(x_i-x_2)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}y_i + \dots + \\ & \frac{(x-x_1)(x-x_2)\dots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\dots(x_n-x_{n-1})}y_n \end{aligned} \quad (5.44)$$

No lado direito da Eq. (5.44), o numerador do  $i$ -ésimo termo não contém  $(x-x_i)$ , e o denominador não contém  $(x_i-x_i)$ . Conseqüentemente, quando a coordenada  $x_i$  de um dos  $n$  pontos é substituída na Eq. (5.44), o valor do polinômio é igual a  $y_i$ . A Eq. (5.44) pode ser escrita de forma compacta usando a notação de soma e produto como:

$$f(x) = \sum_{i=1}^n y_i L_i(x) = \sum_{i=1}^n y_i \prod_{\substack{j=1 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} \quad (5.45)$$

onde as funções  $L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)}$  são chamadas de funções de Lagrange.

Essa forma pode ser facilmente implementada em um programa de computador, conforme mostrado no Exemplo 5-4.

### Notas adicionais sobre os polinômios de Lagrange

- O espaçamento entre os pontos que compõem o conjunto de dados não precisa ser igual.
- Para um dado conjunto de dados, deve-se calcular a expressão completa do polinômio interpolador para cada valor de  $x$ . Em outras palavras, os cálculos de interpolação para cada valor de  $x$  são independentes dos demais. Isso é diferente de outras formas (por exemplo, da Eq. (5.35)), onde uma vez determinados os coeficientes do polinômio, estes podem ser utilizados para calcular diferentes valores de  $x$ .
- Se um valor interpolado for calculado para um dado conjunto de dados e então esse conjunto de dados for ampliado para incluir pontos adicionais, todos os termos do polinômio de Lagrange devem ser calculados novamente. Conforme mostrado na Seção 5.5.2, isso é diferente do que ocorre nos polinômios de Newton, onde apenas os novos termos deverão ser calculados se mais pontos forem adicionados ao conjunto de dados.

A aplicação de um polinômio de Lagrange é mostrada no Exemplo 5-4.

### Exemplo 5-4: Polinômio interpolador de Lagrange

A partir do seguinte conjunto de dados:

$x$	1	2	4	5	7
$y$	52	5	-5	-40	10

- Determine o polinômio de Lagrange de quarta ordem que passa pelos cinco pontos.
- Use o polinômio obtido na letra (a) para determinar o valor interpolado em  $x = 3$ .
- Desenvolva uma função no MATLAB que interpole usando um polinômio de Lagrange. A entrada da função são as coordenadas dos pontos pertencentes ao conjunto de dados fornecido e a coordenada  $x$  onde se deseja calcular o valor interpolado de  $y$ . A saída da função é o valor interpolado de  $y$  em  $x = 3$ .

### SOLUÇÃO

- Seguindo a forma da Eq. (5.44), o polinômio de Lagrange para os cinco pontos dados é:

$$f(x) = \frac{(x-2)(x-4)(x-5)(x-7)}{(1-2)(1-4)(1-5)(1-7)} 52 + \frac{(x-1)(x-4)(x-5)(x-7)}{(2-1)(2-4)(2-5)(2-7)} 5 + \frac{(x-1)(x-2)(x-5)(x-7)}{(4-1)(4-2)(4-5)(4-7)} (-5) + \frac{(x-1)(x-2)(x-4)(x-7)}{(5-1)(5-2)(5-4)(5-7)} (-40) + \frac{(x-1)(x-2)(x-4)(x-5)}{(7-1)(7-2)(7-4)(7-5)} 10$$

- O valor interpolado em  $x = 3$  é obtido com a substituição de  $x$  no polinômio:

$$f(3) = \frac{(3-2)(3-4)(3-5)(3-7)}{(1-2)(1-4)(1-5)(1-7)} 52 + \frac{(3-1)(3-4)(3-5)(3-7)}{(2-1)(2-4)(2-5)(2-7)} 5 + \frac{(3-1)(3-2)(3-5)(3-7)}{(4-1)(4-2)(4-5)(4-7)} (-5) + \\ \frac{(3-1)(3-2)(3-4)(3-7)}{(5-1)(5-2)(5-4)(5-7)} (-40) + \frac{(3-1)(3-2)(3-4)(3-5)}{(7-1)(7-2)(7-4)(7-5)} 10$$

$$f(3) = -5,778 + 2,667 - 4,444 + 13,333 + 0,222 = 6$$

- (c) A função criada no MATLAB para realizar a interpolação usando polinômios de Lagrange é chamada de `Yint=LagrangeINT(x,y,Xint)`. `x` e `y` são vetores com as coordenadas dos pontos fornecidos, e `Xint` é a coordenada do ponto no qual `y` deve ser interpolado.
- O programa calcula primeiro os produtos presentes nas funções de Lagrange descritas na Eq. (5.45). Os termos são atribuídos a uma variável `L` (vetor).

$$L_i = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{(x-x_j)}{(x_i-x_j)} \quad \text{onde} \quad x = Xint$$

- O programa em seguida calcula o valor do polinômio em  $x = Xint$ .

$$f(x) = \sum_{i=1}^n y_i L_i$$

#### Programa 5-4 Função definida pelo usuário. Interpolação utilizando um polinômio de Lagrange.

```
function Yint = LagrangeINT(x,y,Xint)
% LagrangeINT ajusta um polinômio de Lagrange a um conjunto de pontos dado
% e usa o polinômio para determinar o valor interpolado de um ponto.
function Yint = LagrangeINT(x,y,Xint)
% LagrangeINT ajusta um polinômio de Lagrange a um conjunto de pontos dado
% e usa o polinômio para determinar o valor interpolado de um ponto.
% Variáveis de entrada:
% x Vetor com as coordenadas x dos pontos dados.
% y Vetor com as coordenadas y dos pontos dados.
% Xint A coordenada x do ponto a ser interpolado.
% Variável de saída:
% Yint O valor interpolado de Xint.
```

```
n = length(x);
for i = 1:n
    L(i) = 1;
    for j = 1:n
        if j ~= i
            L(i) = L(i)*(Xint - x(j))/(x(i) - x(j));
        end
    end
end
Yint = sum(y .* L);
```

O comprimento do vetor `x` fornece o número de termos do polinômio.

```
    L(i) = L(i)*(Xint - x(j))/(x(i) - x(j));
```

Calcula os termos  $L_i$  do produtório.

```
Yint = sum(y .* L);
```

Calcula o valor do polinômio  $f(x) = \sum_{i=1}^n y_i L_i$ .

A função `LagrangeInt(x,y,Xint)` é em seguida usada na janela de comandos do MATLAB para calcular o valor interpolado de  $x = 3$ .

```

>> x = [1 2 4 5 7];
>> y = [52 5 -5 -40 10];
>> Yinterpolated = LagrangeINT(x,y,3)
Yinterpolated =
    6.0000

```

### 5.5.2 Polinômios interpoladores de Newton

Os polinômios interpoladores de Newton são uma forma popularmente usada no ajuste exato de conjuntos de dados. A forma geral do polinômio de Newton de ordem  $n - 1$  que passa por  $n$  pontos é:

$$f(x) = a_1 + a_2(x-x_1) + a_3(x-x_1)(x-x_2) + \dots + a_n(x-x_1)(x-x_2)\dots(x-x_{n-1}) \quad (5.46)$$

A característica especial de um polinômio como esse está no fato de os coeficientes  $a_1$  a  $a_n$  poderem ser determinados a partir de um procedimento matemático simples (a determinação dos coeficientes não requer a solução de um sistema com  $n$  equações). Uma vez conhecidos os coeficientes, o polinômio pode ser usado para calcular um valor interpolado em qualquer  $x$ .

Os polinômios interpoladores de Newton têm características adicionais desejáveis que os fazem uma escolha popular. Os pontos do conjunto de dados não precisam estar ordenados de forma ascendente ou descendente, ou mesmo em qualquer ordem. Além disso, após a determinação dos  $n$  coeficientes de um polinômio interpolador de Newton de ordem  $n - 1$ , mais pontos podem ser adicionados ao conjunto de dados, sendo necessário apenas determinar os coeficientes adicionais.

#### Polinômio de Newton de primeira ordem

Para dois pontos,  $(x_1, y_1)$  e  $(x_2, y_2)$ , o polinômio de Newton de primeira ordem tem a forma:

$$f(x) = a_1 + a_2(x-x_1) \quad (5.47)$$

Conforme mostrado na Fig. 5-14, esta é a equação da linha reta que passa pelos dois pontos fornecidos. Os coeficientes  $a_1$  e  $a_2$  podem ser calculados utilizando a semelhança de triângulos na Fig. 5-14:

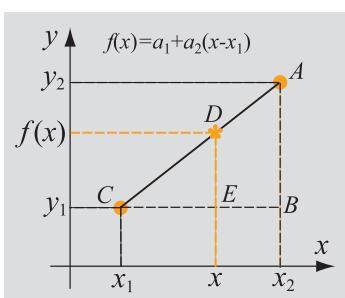
$$\frac{DE}{CE} = \frac{AB}{CB}, \quad \text{ou} \quad \frac{f(x)-y_1}{x-x_1} = \frac{y_2-y_1}{x_2-x_1} \quad (5.48)$$

Resolvendo para  $f(x)$  na Eq. (5.48), obtém-se:

$$f(x) = y_1 + \frac{y_2-y_1}{x_2-x_1}(x-x_1) \quad (5.49)$$

Comparando a Eq. (5.49) com a Eq. (5.47), obtém-se os valores dos coeficientes  $a_1$  e  $a_2$  em termos das coordenadas dos pontos:

$$a_1 = y_1 \quad \text{e} \quad a_2 = \frac{y_2-y_1}{x_2-x_1} \quad (5.50)$$



**Figura 5-14** Polinômio de Newton de primeira ordem.

Note que o coeficiente  $a_2$  é a inclinação da reta que conecta os dois pontos. Conforme mostrado no Capítulo 6,  $a_2$  é a aproximação da derivada primeira no ponto  $(x_1, y_1)$  usando a diferença progressiva com dois pontos.

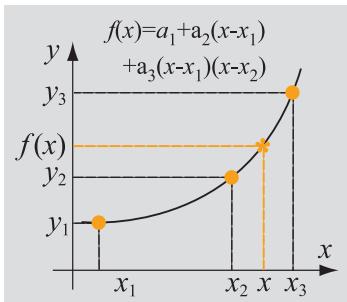
### Polinômio de Newton de segunda ordem

Para três pontos,  $(x_1, y_1)$ ,  $(x_2, y_2)$  e  $(x_3, y_3)$ , o polinômio de Newton de segunda ordem tem a forma:

$$f(x) = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) \quad (5.51)$$

Conforme mostrado na Fig. 5-15, a Eq. (5.51) descreve uma parábola que passa pelos três pontos dados. Os coeficientes  $a_1$ ,  $a_2$  e  $a_3$  podem ser determinados com a substituição dos três pontos na Eq. (5.51). A substituição de  $x = x_1$  e  $f(x_1) = y_1$  resulta em  $a_1 = y_1$ . A substituição do segundo ponto,  $x = x_2$  e  $f(x_2) = y_2$  (e  $a_1 = y_1$ ), na Eq. (5.51), resulta em:

$$y_2 = y_1 + a_2(x_2 - x_1) \quad \text{ou} \quad a_2 = \frac{y_2 - y_1}{x_2 - x_1} \quad (5.52)$$



**Figura 5-15** Polinômio de Newton de segunda ordem.

A substituição do terceiro ponto,  $x = x_3$  e  $f(x_3) = y_3$  (bem como  $a_1 = y_1$  e  $a_2 = \frac{y_2 - y_1}{x_2 - x_1}$ ), na Eq. (5.51), resulta em:

$$y_3 = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x_3 - x_1) + a_3(x_3 - x_1)(x_3 - x_2) \quad (5.53)$$

O termo  $a_3$  pode ser isolado na Eq. (5.53) após alguma álgebra:

$$a_3 = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{(x_3 - x_1)} \quad (5.54)$$

Os coeficientes  $a_1$  e  $a_2$  são os mesmos nos polinômios de primeira e de segunda ordem. Isso significa que, se dois pontos são dados e um polinômio de Newton de primeira ordem é usado para gerar uma reta passando por esses dois pontos, este polinômio pode ser modificado para se tornar um polinômio de segunda ordem passando por três pontos (caso um terceiro ponto seja adicionado) simplesmente com a determinação do coeficiente adicional.

### Polinômio de Newton de terceira ordem

Para quatro pontos,  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  e  $(x_4, y_4)$ , o polinômio de Newton de terceira ordem tem a forma:

$$f(x) = y = a_1 + a_2(x - x_1) + a_3(x - x_1)(x - x_2) + a_4(x - x_1)(x - x_2)(x - x_3) \quad (5.55)$$

As fórmulas para os coeficientes  $a_1$ ,  $a_2$  e  $a_3$  são as mesmas usadas para o polinômio de segunda ordem. A fórmula para determinar o coeficiente  $a_4$  pode ser obtida com a substituição de  $(x_4, y_4)$  na Eq. (5.55):

$$a_4 = \frac{\left( \frac{y_4 - y_3}{x_4 - x_3} - \frac{y_3 - y_2}{x_3 - x_2} \right) - \left( \frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1} \right)}{(x_4 - x_1)} \quad (5.56)$$

### Forma geral do polinômio de Newton e de seus coeficientes

Um exame cuidadoso das equações dos coeficientes  $a_2$  (Eq. (5.52)),  $a_3$  (Eq. (5.54)) e  $a_4$  (Eq. (5.56)) mostra que essas expressões seguem um certo modelo. Esse modelo fica mais claro com a definição das chamadas *diferenças divididas*.

Para dois pontos,  $(x_1, y_1)$  e  $(x_2, y_2)$ , a primeira diferença dividida, escrita como  $f[x_2, x_1]$ , é definida como a inclinação da reta que conecta os dois pontos:

$$f[x_2, x_1] = \frac{y_2 - y_1}{x_2 - x_1} = a_2 \quad (5.57)$$

A primeira diferença dividida é igual ao coeficiente  $a_2$ .

Para três pontos,  $(x_1, y_1)$ ,  $(x_2, y_2)$  e  $(x_3, y_3)$ , a segunda diferença dividida, escrita como  $f[x_3, x_2, x_1]$ , é definida como a diferença entre as primeiras diferenças divididas dos pontos  $(x_3, y_3)$  e  $(x_2, y_2)$  e dos pontos  $(x_2, y_2)$  e  $(x_1, y_1)$ , dividida por  $(x_3 - x_1)$ :

$$f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = \frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{(x_3 - x_1)} = a_3 \quad (5.58)$$

A segunda diferença dividida é portanto igual ao coeficiente  $a_3$ .

Para quatro pontos,  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  e  $(x_4, y_4)$ , a terceira diferença dividida, escrita como  $f[x_4, x_3, x_2, x_1]$ , é definida como a diferença entre as segundas diferenças divididas dos pontos  $(x_2, y_2)$ ,  $(x_3, y_3)$  e  $(x_4, y_4)$  e dos pontos  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ , dividida por  $(x_4 - x_1)$ :

$$\begin{aligned} f[x_4, x_3, x_2, x_1] &= \frac{f[x_4, x_3, x_2] - f[x_3, x_2, x_1]}{x_4 - x_1} \\ &= \frac{\frac{f[x_4, x_3] - f[x_3, x_2]}{x_4 - x_3} - \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_2}}{(x_4 - x_1)} \\ &= \frac{\left( \frac{y_4 - y_3}{x_4 - x_3} - \frac{y_3 - y_2}{x_3 - x_2} \right) - \frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{(x_4 - x_1)} \\ &= \frac{\left( \frac{y_4 - y_3}{x_4 - x_3} - \frac{y_3 - y_2}{x_3 - x_2} \right) - \frac{y_2 - y_1}{x_2 - x_1}}{(x_4 - x_1)} = a_4 \end{aligned} \quad (5.59)$$

A terceira diferença dividida é portanto igual ao coeficiente  $a_4$ .

A próxima (quarta) diferença dividida (quando cinco pontos são dados) é:

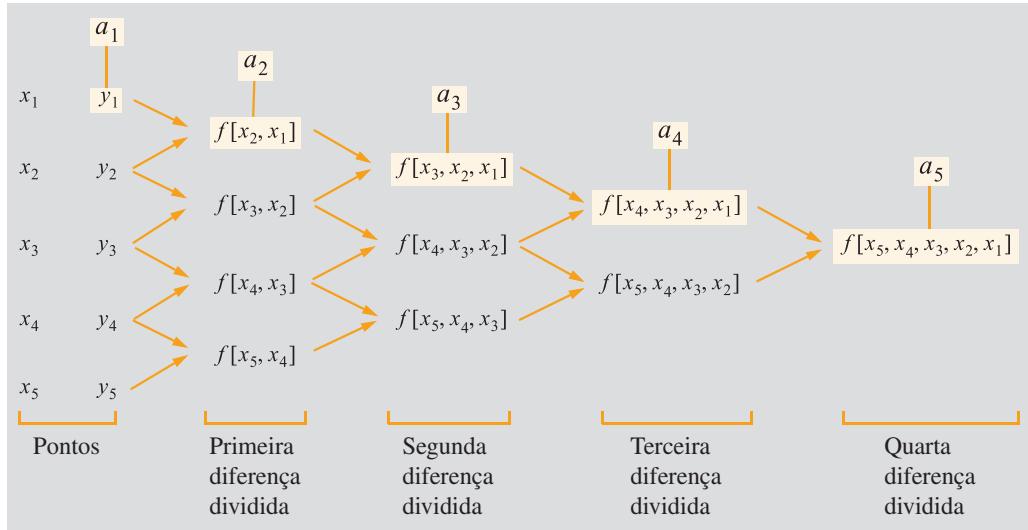
$$f[x_5, x_4, x_3, x_2, x_1] = \frac{f[x_5, x_4, x_3, x_2] - f[x_4, x_3, x_2, x_1]}{x_5 - x_1} = a_5 \quad (5.60)$$

Se mais pontos forem fornecidos, o procedimento para calcular diferenças maiores continua da mesma maneira. Em geral, quando  $n$  pontos são dados, o procedimento começa com o cálculo das  $(n - 1)$  primeiras diferenças divididas. Depois,  $(n - 2)$  segundas diferenças divididas são calculadas a partir das primeiras diferenças divididas. Este passo é sucedido do cálculo das  $(n - 3)$  terceiras diferenças divididas a partir das segundas diferenças divididas. O processo termina quando a  $n$ -ésima diferença dividida é calculada a partir das duas  $(n - 1)$  diferenças divididas para fornecer o coeficiente  $a_n$ .

O procedimento de determinação dos coeficientes usando as diferenças divididas pode ser acompanhado em uma tabela. Uma tabela como essa é ilustrada na Fig. 5-16 para o caso de um conjunto de dados com cinco pontos.

Em termos gerais, para  $n$  pontos,  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , as primeiras diferenças divididas entre dois pontos  $(x_i, y_i)$  e  $(x_j, y_j)$  são dadas por:

$$f[x_j, x_i] = \frac{y_j - y_i}{x_j - x_i} \quad (5.61)$$



**Figura 5-16** Tabela de diferenças divididas para um conjunto de dados com cinco pontos.

A  $k$ -ésima diferença dividida ordem 2 ou superior é dada por (equação válida até a diferença de ordem  $(n - 1)$ ):

$$f[x_k, x_{k-1}, \dots, x_2, x_1] = \frac{f[x_k, x_{k-1}, \dots, x_3, x_2] - f[x_{k-1}, x_{k-2}, \dots, x_2, x_1]}{x_k - x_1} \quad (5.62)$$

Com essas definições, o polinômio de Newton de ordem  $(n - 1)$ , Eq. (5.46), é dado por:

$$f(x) = y = y_1 + f[x_2, x_1](x-x_1) + f[x_3, x_2, x_1](x-x_1)(x-x_2) + \dots + f[x_n, x_{n-1}, \dots, x_2, x_1](x-x_1)(x-x_2)\dots(x-x_{n-1})$$

$\underbrace{a_1}_{\text{---}}$     $\underbrace{a_2}_{\text{---}}$     $\underbrace{a_3}_{\text{---}}$     $\dots$     $\underbrace{a_n}_{\text{---}}$

(5.63)

### Notas sobre os polinômios de Newton

- O espaçamento entre os pontos que compõem o conjunto de dados não precisa ser o mesmo.
- Em um dado conjunto de dados com  $n$  pontos, os coeficientes  $a_1$  a  $a_n$ , assim que determinados, podem ser usados para interpolar quaisquer dos pontos que compõem o conjunto de dados.

O Exemplo 5-5 mostra a aplicação dos polinômios interpoladores de Newton.

### Exemplo 5-5: Polinômio interpolador de Newton

A partir do seguinte conjunto de dados com cinco pontos:

$x$	1	2	4	5	7
$y$	52	5	-5	-40	10

- (a) Determine o polinômio de quarta ordem na forma de Newton que passe pelos pontos. Calcule os coeficientes usando uma tabela de diferenças divididas.
- (b) Escreva o polinômio obtido na letra (a) para determinar o valor interpolado em  $x = 3$ .
- (c) Escreva uma função no MATLAB que faça a interpolação usando o polinômio de Newton. A entrada da função deve conter as coordenadas dos pontos fornecidos e a coordenada  $x$  do ponto no qual  $y$  deve ser interpolado. A saída da função é o valor de  $y$  no ponto interpolado.

### SOLUÇÃO

- (a) O polinômio de Newton para os pontos dados tem a forma:

$$f(x) = y = a_1 + a_2(x-1) + a_3(x-1)(x-2) + a_4(x-1)(x-2)(x-4) + a_5(x-1)(x-2)(x-4)(x-5)$$

Os coeficientes podem ser determinados a partir da seguinte tabela de diferenças divididas:

$x_i$	$y_i$	$a_1 = 52$	$a_2 = -47$	$a_3 = 14$	$a_4 = -6$	$a_5 = 2$
1	52					
2	5	$\frac{5-52}{2-1} = -47$				
4	-5		$\frac{-5-(-47)}{4-1} = 14$			
5	-40			$\frac{-10-14}{5-1} = -6$		
7	10				$\frac{6-(-6)}{7-1} = 2$	

Com os coeficientes determinados, o polinômio é:

$$f(x) = y = 52 - 47(x-1) + 14(x-1)(x-2) - 6((x-1)(x-2)(x-4)) + 2(x-1)(x-2)(x-4)(x-5)$$

- (b) O valor interpolado em  $x = 3$  é obtido com a substituição de  $x$  no polinômio:

$$f(3) = y = 52 - 47(3-1) + 14(3-1)(3-2) - 6((3-1)(3-2)(3-4)) + 2(3-1)(3-2)(3-4)(3-5) = 6$$

- (c) A função criada no MATLAB para realizar a interpolação de Newton é chamada de `Yint=NewtonINT(x,y,Xint)`.  $x$  e  $y$  são vetores com as coordenadas dos pontos fornecidos, e  $Xint$  é a coordenada do ponto no qual  $y$  deve ser interpolado.

- O programa começa com o cálculo das primeiras diferenças divididas, que então são usadas para calcular as diferenças divididas de ordem superior.
- Os coeficientes do polinômio (primeira linha da tabela) são então atribuídos a um vetor  $a$ .
- O polinômio determinado é usado na interpolação.

### Programa 5-5 Função definida pelo usuário. Interpolação usando o polinômio de Newton.

```

function Yint = NewtonsINT(x,y,Xint)
% NewtonINT ajusta um polinômio de Newton a um dado conjunto de pontos e
% usa esse polinômio para determinar o valor interpolado de um ponto.
% Variáveis de entrada:
% x Vetor com as coordenadas x dos pontos dados.
% y Vetor com as coordenadas y dos pontos dados.
% Xint Coordenada x do ponto a ser interpolado.
% Variável de saída:
% Yint O valor interpolado de Xint.

n = length(x);          % O comprimento do vetor x fornece o número de coeficientes (e termos) do polinômio.
a(1) = y(1);            % O primeiro coeficiente  $a_1$ .
for i = 1:n - 1
    divDIF(i,1) = (y(i + 1) - y(i))/(x(i + 1) - x(i));
end
for j = 2:n - 1
    for i = 1:n - j
        divDIF(i,j) = (divDIF(i + 1,j - 1) - divDIF(i,j - 1))/(x(j + i) - x(i));
    end
end
for j = 2:n
    a(j) = divDIF(1,j - 1);
end
Yint = a(1);
xn = 1;
for k = 2:n
    xn = xn*(Xint - x(k - 1));
    Yint = Yint + a(k)*xn;
end

```

Calcula as diferenças divididas. Elas são armazenadas na primeira coluna de  $\text{divDIF}$ .

Calcula as diferenças divididas de ordem 2 e superior (até a ordem  $(n - 1)$ ). Os valores são atribuídos às colunas de  $\text{divDIF}$ .

Atribui os coeficientes  $a_2$  a  $a_n$  ao vetor  $a$ .

Calcula o valor interpolado de  $Xint$ . O primeiro termo no polinômio é  $a_1$ . Os termos seguintes são adicionados por meio de um *loop*.

A função `NewtonINT(x,y,Xint)` é então usada na janela de comandos do MATLAB para calcular o valor interpolado de  $x = 3$ .

```

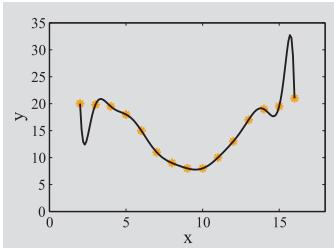
>> x = [1 2 4 5 7];
>> y = [52 5 -5 -40 10];
>> Yinterpolado=NewtonINT(x,y,3)
Yinterpolado =
    6

```

## 5.6 INTERPOLAÇÃO POR PARTES (SPLINE)

Quando um conjunto de dados contendo  $n$  pontos é dado e um único polinômio é usado para fazer a sua interpolação, esse polinômio fornece os valores exatos nos pontos e determina valores estimados (interpolados) entre eles.

Quando o número de pontos é pequeno, de forma que a ordem do polinômio seja baixa, os valores interpolados são tipicamente precisos. Entretanto, conforme mencionado na Seção 5.4, erros maiores podem ocorrer quando um polinômio de ordem elevada é usado para interpolar um grande número de pontos. Isso é mostrado na Fig. 5-17, onde um polinômio de 15<sup>a</sup> ordem é usado para interpolar um conjunto de 16 pontos. Fica claro, a partir da figura, que o polinômio apresenta um desvio significativo na região próxima às extremidades, não acompanhando a tendência dos dados. Por esse motivo, ele não pode ser usado de forma confiável na interpolação.



**Figura 5-17** Uso de um polinômio de 15<sup>a</sup> ordem para fazer o ajuste de 16 pontos.

Quando se trabalha com um grande número de pontos, uma melhor interpolação pode ser feita com o uso de muitos polinômios de baixa ordem ao invés de um único polinômio de ordem elevada. Cada polinômio de baixa ordem é válido em um intervalo entre dois ou vários pontos. Tipicamente, todos os polinômios utilizados têm a mesma ordem, mas os coeficientes são diferentes em cada intervalo. Quando polinômios de primeira ordem são utilizados, linhas retas conectam os pontos. Para polinômios de segunda ordem (quadráticos) e de terceira ordem (cúbicos), os pontos são conectados por curvas. A interpolação feita dessa forma é chamada de interpolação *por partes*, ou *spline*. Os pontos do conjunto de dados onde se encontram os polinômios de intervalos adjacentes são chamados de *nós*. O nome “spline” vem do termo em inglês usado para denominar a haste flexível que se utiliza no desenho técnico para interpolar fisicamente pontos discretos marcados por pinos.

Os três tipos de interpolação spline são a linear, a quadrática e a cúbica.

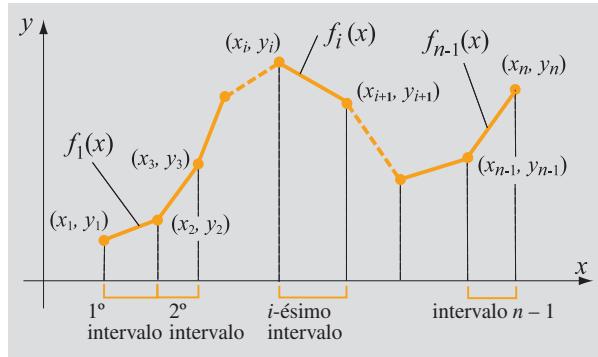
### 5.6.1 Splines lineares

Com splines lineares, a interpolação é feita usando um polinômio de primeira ordem (função linear), e os pontos são conectados por linhas retas, conforme mostrado na Fig. 5-18. Usando a forma de Lagrange, a equação da linhareta que conecta os dois primeiros pontos é dada por:

$$f_1(x) = \frac{(x - x_2)}{(x_1 - x_2)}y_1 + \frac{(x - x_1)}{(x_2 - x_1)}y_2 \quad (5.64)$$

Em um conjunto de  $n$  pontos, há  $n - 1$  intervalos. A interpolação no intervalo  $i$ , que está entre os pontos  $x_i$  e  $x_{i+1}$  ( $x_i \leq x \leq x_{i+1}$ ), é feita usando a equação da linha reta que conecta o ponto  $(x_i, y_i)$  ao ponto  $(x_{i+1}, y_{i+1})$ :

$$f_i(x) = \frac{(x - x_{i+1})}{(x_i - x_{i+1})}y_i + \frac{(x - x_i)}{(x_{i+1} - x_i)}y_{i+1} \text{ para } i = 1, 2, \dots, n - 1 \quad (5.65)$$



**Figura 5-18** Splines lineares.

É óbvio que splines lineares resultam em uma interpolação contínua, já que os dois polinômios adjacentes têm o mesmo valor em um nó comum. Há, no entanto, uma descontinuidade na inclinação das splines lineares nos nós.

A interpolação feita com splines lineares pode ser facilmente calculada e programada, e fornece bons resultados quando o intervalo entre os pontos é pequeno. O Exemplo 5-6 mostra a aplicação numérica das splines lineares usando cálculos manuais e o desenvolvimento de uma função no MATLAB.

### Exemplo 5-6: Splines lineares

A partir do seguinte conjunto de dados com quatro pontos:

$x$	8	11	15	18
$y$	5	9	10	8

- (a) Determine as splines lineares que fazem o ajuste dos dados.
- (b) Determine o valor interpolado em  $x = 12,7$ .
- (c) Escreva uma função no MATLAB que faça a interpolação usando splines lineares. A entrada da função deve conter as coordenadas dos pontos fornecidos e a coordenada  $x$  do ponto no qual  $y$  deve ser interpolado. A saída da função é o valor de  $y$  no ponto interpolado. Use essa função para determinar o valor interpolado de  $y$  em  $x = 12,7$ .

### SOLUÇÃO

- (a) Há quatro pontos, e portanto três splines. Usando a Eq. (5.65), as equações das splines são:

$$f_1(x) = \frac{(x - x_2)}{(x_1 - x_2)}y_1 + \frac{(x - x_1)}{(x_2 - x_1)}y_2 = \frac{(x - 11)}{(8 - 11)}5 + \frac{(x - 8)}{(11 - 8)}9 = \frac{5}{-3}(x - 11) + \frac{9}{2}(x - 8) \quad \text{para } 8 \leq x \leq 11$$

$$f_2(x) = \frac{(x - x_3)}{(x_2 - x_3)}y_2 + \frac{(x - x_2)}{(x_3 - x_2)}y_3 = \frac{(x - 15)}{(11 - 15)}9 + \frac{(x - 11)}{(15 - 11)}10 = \frac{9}{-4}(x - 15) + \frac{10}{4}(x - 11) \quad \text{para } 11 \leq x \leq 15$$

$$f_3(x) = \frac{(x - x_4)}{(x_3 - x_4)}y_3 + \frac{(x - x_3)}{(x_4 - x_3)}y_4 = \frac{(x - 18)}{(15 - 18)}10 + \frac{(x - 15)}{(18 - 15)}8 = \frac{10}{-3}(x - 18) + \frac{8}{3}(x - 15) \quad \text{para } 15 \leq x \leq 18$$

- (b) O valor interpolado de  $y$  em  $x = 12,7$  é obtido com a substituição do valor  $x$  na equação de  $f_2(x)$  acima:

$$f_2(x) = \frac{9}{4}(12,7 - 15) + \frac{10}{4}(12,7 - 11) = 9,425$$

- (c) A função criada no MATLAB para implementar a interpolação linear usando splines é chamada de `Yint=SplineLinear(x,y,Xint)`. `x` e `y` são vetores com as coordenadas dos pontos fornecidos, e `Xint` é a coordenada do ponto no qual `y` deve ser interpolado.

#### Programa 5-6 Função definida pelo usuário. Splines lineares.

```
function Yint = LinearSpline(x, y, Xint)
% SplineLinear calcula a interpolação usando splines lineares.
% Variáveis de entrada:
% x  Vetor com as coordenadas x dos pontos dados.
% y  Vetor com as coordenadas y dos pontos dados.
% Xint  Coordenada x do ponto a ser interpolado.
% Variável de saída:
% Yint  O valor interpolado de Xint.
```

```
n = length(x);
for i = 2:n
    if Xint < x(i)
        break
    end
end
Yint = (Xint - x(i))*y(i - 1)/(x(i - 1) - x(i)) + (Xint - x(i - 1))*y(i)/(x(i) - x(i - 1));
```

O comprimento do vetor `x` fornece o número de termos contidos nos dados.

Determina o intervalo que inclui `Xint`.

Calcula `Yint` com a Eq. (5.65).

A função `SplineLinear(x,y,Xint)` é usada em seguida na janela de comandos do MATLAB para calcular o valor interpolado de  $x = 12,7$ .

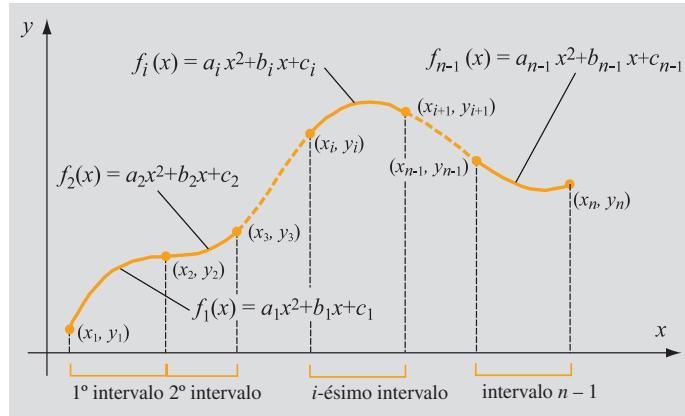
```
>> x = [8 11 15 18];
>> y = [5 9 10 8];
>> Yint = LinearSpline(x,y,12.7)
Yint =
    9.4250
```

#### 5.6.2 Splines quadráticas

Nas splines quadráticas, a interpolação é feita com polinômios de segunda ordem (Fig. 5-19). Em um conjunto de  $n$  pontos, há  $n - 1$  intervalos, e, usando a forma padrão, a equação do polinômio no  $i$ -ésimo intervalo, localizado entre os pontos  $x_i$  e  $x_{i+1}$ , é dada por:

$$f_i(x) = a_i x^2 + b_i x + c_i \quad \text{para } i = 1, 2, \dots, n - 1 \quad (5.66)$$

De forma geral, há  $n - 1$  equações. Como cada equação tem três coeficientes, um total de  $3(n - 1) = 3n - 3$  coeficientes têm que ser determinados. Os coeficientes são determinados com a aplicação das seguintes condições:



**Figura 5-19** Splines quadráticas.

- Cada polinômio  $f_i(x)$  deve passar pelos pontos finais do intervalo,  $(x_i, y_i)$  e  $(x_{i+1}, y_{i+1})$ , o que significa que  $f_i(x_i) = y_i$  e  $f_i(x_{i+1}) = y_{i+1}$ :

$$a_i x_i^2 + b_i x_i + c_i = y_i \quad \text{para } i = 1, 2, \dots, n-1 \quad (5.67)$$

$$a_i x_{i+1}^2 + b_i x_{i+1} + c_i = y_{i+1} \quad \text{para } i = 1, 2, \dots, n-1 \quad (5.68)$$

Como há  $n - 1$  intervalos, essa condição fornece  $2(n - 1) = 2n - 2$  equações.

- Nos nós internos, as inclinações (derivadas primeiras) dos polinômios de intervalos adjacentes são iguais. Isso significa que, com a transição da curva que passa por um nó interno de um polinômio para outro, a inclinação deve ser contínua. A derivada primeira do  $i$ -ésimo polinômio é:

$$f'(x) = \frac{df}{dx} = 2a_i x + b_i \quad (5.69)$$

Para  $n$  pontos, o primeiro ponto interno é  $i = 2$  e o último é  $i = n - 1$ . Igualando as derivadas primeiras em todos os pontos internos, obtém-se:

$$2a_{i-1}x_i + b_{i-1} = 2a_i x_i + b_i \quad \text{para } i = 2, 3, \dots, n-1 \quad (5.70)$$

Como há  $n - 2$  pontos internos, essa condição fornece  $n - 2$  equações.

Juntas, as duas condições fornecem  $3n - 4$  equações. Entretanto, os  $n - 1$  polinômios têm  $3n - 3$  coeficientes, de forma que uma equação adicional (condição) é necessária para que os coeficientes sejam obtidos. A condição comumente aplicada assume que a derivada segunda seja nula no primeiro ou no último ponto. Considere a primeira escolha, descrita em detalhe no item 3.

- A derivada segunda no primeiro ponto,  $(x_1, y_1)$ , é nula. O polinômio no primeiro intervalo (entre o primeiro e o segundo ponto) é:

$$f_1(x) = a_1 x^2 + b_1 x + c_1 \quad (5.71)$$

A derivada segunda do polinômio é  $f_1''(x) = 2a_1$ , que, quando igualada a zero, resulta em  $a_1 = 0$ . Essa condição significa, na realidade, que uma linha reta conecta os dois primeiros pontos (a inclinação é constante).

### Nota sobre as splines quadráticas e cúbicas

Splines quadráticas têm derivada primeira contínua em pontos internos (nós). Em um conjunto de  $n$  pontos, elas requerem a solução de um sistema linear com  $3n - 4$  equações para que os coeficientes dos polinômios sejam determinados. Conforme mostrado na próxima seção, splines cúbicas têm derivadas primeira e segunda contínuas nos pontos internos, e podem ser escritas em uma forma que requer a solução de um sistema linear com apenas  $n - 2$  equações em função dos coeficientes.

O Exemplo 5-7 mostra a aplicação das splines quadráticas na interpolação de um conjunto de dados com 5 pontos.

#### Exemplo 5-7: Splines quadráticas

A partir do conjunto de dados com cinco pontos a seguir:

$x$	8	11	15	18	22
$y$	5	9	10	8	7

- (a) Determine as splines quadráticas que fazem o ajuste dos dados.
- (b) Determine o valor interpolado de  $y$  em  $x = 12,7$ .
- (c) Trace um gráfico com os pontos do conjunto de dados e os polinômios interpoladores.

#### SOLUÇÃO

- (a) Há cinco pontos ( $n = 5$ ) e portanto quatro splines ( $i = 1, 2, \dots, 4$ ). A equação quadrática para a  $i$ -ésima spline é:

$$f_i(x) = a_i x^2 + b_i x + c_i$$

Há quatro polinômios, e, como cada polinômio tem três coeficientes, 12 coeficientes têm que ser determinados no total. Os coeficientes são  $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3, a_4, b_4$  e  $c_4$ . O coeficiente  $a_1$  é igual a zero (ver condição 3). Os outros 11 coeficientes são determinados a partir de um sistema linear de 11 equações.

Oito equações são obtidas a partir da condição descrita nas Eqs. (5.67) e (5.68), que diz que, em cada intervalo, o polinômio deve passar pelos pontos finais:

$$\begin{aligned} i = 1 \quad f_1(x) &= a_1 x_1^2 + b_1 x_1 + c_1 = b_1 8 + c_1 = 5 \\ f_1(x) &= a_1 x_2^2 + b_1 x_2 + c_1 = b_1 11 + c_1 = 9 \\ i = 2 \quad f_2(x) &= a_2 x_2^2 + b_2 x_2 + c_2 = a_2 11^2 + b_2 11 + c_2 = 9 \\ f_2(x) &= a_2 x_3^2 + b_2 x_3 + c_2 = a_2 15^2 + b_2 15 + c_2 = 10 \\ i = 3 \quad f_3(x) &= a_3 x_3^2 + b_3 x_3 + c_3 = a_3 15^2 + b_3 15 + c_3 = 10 \\ f_3(x) &= a_3 x_4^2 + b_3 x_4 + c_3 = a_3 18^2 + b_3 18 + c_3 = 8 \\ i = 4 \quad f_4(x) &= a_4 x_4^2 + b_4 x_4 + c_4 = a_4 18^2 + b_4 18 + c_4 = 8 \\ f_4(x) &= a_4 x_5^2 + b_4 x_5 + c_4 = a_4 22^2 + b_4 22 + c_4 = 7 \end{aligned}$$

Três equações são obtidas a partir da condição que diz que, nos nós interiores, as inclinações (derivadas primeiras) dos polinômios de intervalos adjacentes são iguais (Eq. (5.70)).

$$\begin{aligned}
 i = 2 \quad 2a_1x_2 + b_1 &= 2a_2x_2 + b_2 \longrightarrow b_1 = 2a_211 + b_2 \quad \text{ou:} \quad b_1 - 2a_211 - b_2 = 0 \\
 i = 3 \quad 2a_2x_3 + b_2 &= 2a_3x_3 + b_3 \longrightarrow 2a_215 + b_2 = 2a_315 + b_3 \quad \text{ou:} \quad 2a_215 + b_2 - 2a_315 - b_3 = 0 \\
 i = 4 \quad 2a_3x_4 + b_3 &= 2a_4x_4 + b_4 \longrightarrow 2a_318 + b_3 = 2a_418 + b_4 \quad \text{ou:} \quad 2a_318 + b_3 - 2a_418 - b_4 = 0
 \end{aligned}$$

O sistema de 11 equações lineares pode ser escrito na forma matricial:

$$\left[ \begin{array}{ccccccccc|c} 8 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & b_1 \\ 11 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_1 \\ 0 & 0 & 11^2 & 11 & 1 & 0 & 0 & 0 & 0 & a_2 \\ 0 & 0 & 15^2 & 15 & 1 & 0 & 0 & 0 & 0 & b_2 \\ 0 & 0 & 0 & 0 & 0 & 15^2 & 15 & 1 & 0 & c_2 \\ 0 & 0 & 0 & 0 & 0 & 18^2 & 18 & 1 & 0 & a_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18^2 & 18 & b_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22^2 & c_3 \\ 1 & 0 & -22 & -1 & 0 & 0 & 0 & 0 & 0 & a_4 \\ 0 & 0 & 30 & 1 & 0 & -30 & -1 & 0 & 0 & b_4 \\ 0 & 0 & 0 & 0 & 0 & 36 & 1 & 0 & -36 & c_4 \end{array} \right] = \begin{bmatrix} 5 \\ 9 \\ 9 \\ 10 \\ 10 \\ 8 \\ 8 \\ 7 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.72)$$

O sistema da Eq. (5.72) é resolvido no MATLAB:

```

>> A = [8 1 0 0 0 0 0 0 0 0 0; 11 1 0 0 0 0 0 0 0 0 0; 0 0 11^2 11 1 0 0 0 0 0 0
0 0 15^2 15 1 0 0 0 0 0 0; 0 0 0 0 0 15^2 15 1 0 0 0; 0 0 0 0 0 18^2 18 1 0 0 0
0 0 0 0 0 0 0 18^2 18 1; 0 0 0 0 0 0 0 22^2 22 1; 1 0 -22 -1 0 0 0 0 0 0 0
0 0 30 1 0 -30 -1 0 0 0 0; 0 0 0 0 0 36 1 0 -36 -1 0];
>> B = [5; 9; 10; 10; 8; 8; 7; 0; 0; 0];
>> coeficientes = (A\B)';
coeficientes =
1.3333 -5.6667 -0.2708 7.2917 -38.4375 0.0556 -2.5000 35.0000 0.0625 -2.7500 37.2500

```

$b_1$	$c_1$	$a_2$	$b_2$	$c_2$	$a_3$	$b_3$	$c_3$	$a_4$	$b_4$	$c_4$
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

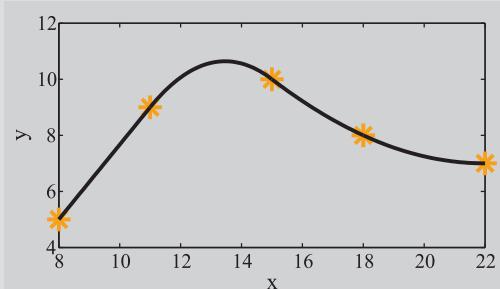
De posse dos coeficientes, os polinômios são:

$$\begin{aligned}
 f_1(x) &= 1,333x - 5,6667 \quad \text{para } 8 \leq x \leq 11, \quad f_2(x) = (-0,2708)x^2 + 7,2917x - 38,4375 \quad \text{para } 11 \leq x \leq 15 \\
 f_3(x) &= 0,0556x^2 - 2,5x + 35 \quad \text{para } 15 \leq x \leq 18, \quad f_4(x) = 0,0625x^2 - 2,75x + 37,25 \quad \text{para } 18 \leq x \leq 22
 \end{aligned}$$

(b) O valor interpolado de  $y$  em  $x = 12,7$  é calculado com a substituição do valor de  $x$  em  $f_2(x)$ :

$$f_2(12,7) = (-0,2708)12,7^2 + 7,2917 \cdot 12,7 - 38,4375 = 10,4898$$

(c) O gráfico ao lado mostra os pontos do conjunto de dados e as curvas descritas pelos polinômios. O gráfico mostra claramente que a primeira spline é uma linha reta (inclinação constante).



### 5.6.3 Splines cúbicas

Em splines cúbicas, a interpolação é feita com polinômios de terceira ordem. Para um conjunto de dados com  $n$  pontos, há  $n - 1$  intervalos. Como cada um dos polinômios de terceira ordem tem quatro coeficientes, a determinação de todos os coeficientes pode requerer um grande número de cálculos. Conforme explicado anteriormente neste capítulo, polinômios podem ser escritos de diversas maneiras (padrão, Lagrange, Newton) e, em tese, qualquer uma delas pode ser usada para representar as splines cúbicas. Na prática, contudo, a quantidade de cálculos varia bastante com a forma do polinômio utilizado. A apresentação a seguir mostra duas deduções de splines cúbicas. A primeira usa a forma padrão dos polinômios, e a segunda usa uma variação da forma de Lagrange. A dedução da forma padrão é mais fácil de acompanhar, de entender e de usar (ela é similar à dedução das splines quadráticas), mas requer a solução de um sistema com  $4n - 4$  equações. A dedução baseada na forma de Lagrange é mais sofisticada, mas requer a solução de apenas  $n - 2$  equações lineares.

#### Splines cúbicas com polinômios na forma padrão

Para um conjunto de dados com  $n$  pontos, conforme mostrado na Fig. 5-20, há  $n - 1$  intervalos. Usando a forma padrão, a equação do polinômio do  $i$ -ésimo intervalo, localizado entre os pontos  $x_i$  e  $x_{i+1}$ , é dada por:

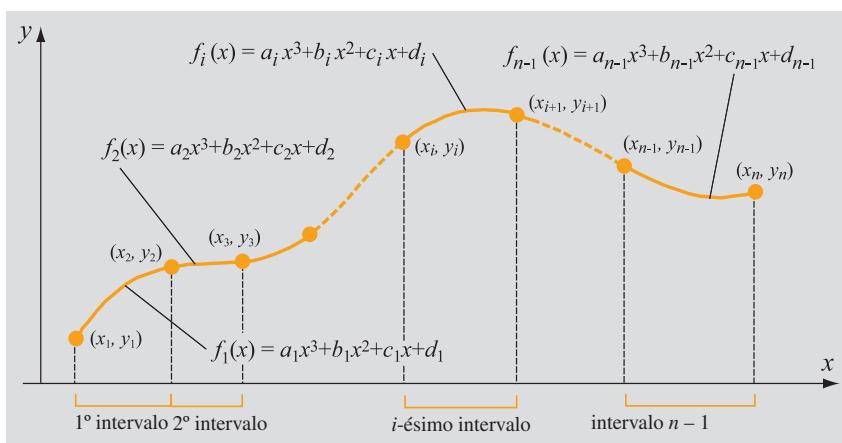
$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \quad (5.73)$$

De forma geral, há  $n - 1$  equações. Como cada equação tem quatro coeficientes,  $4(n - 1) = 4n - 4$  coeficientes têm que ser determinados no total. Os coeficientes são obtidos com a aplicação das seguintes condições:

1. Cada polinômio  $f_i(x)$  deve passar pelos pontos finais do intervalo,  $(x_i, y_i)$  e  $(x_{i+1}, y_{i+1})$ , o que significa que  $f_i(x_i) = y_i$  e  $f_i(x_{i+1}) = y_{i+1}$ :

$$a_i x_i^3 + b_i x_i^2 + c_i x_i + d_i = y_i \quad \text{para } i = 1, 2, \dots, n - 1 \quad (5.74)$$

$$a_i x_{i+1}^3 + b_i x_{i+1}^2 + c_i x_{i+1} + d_i = y_{i+1} \quad \text{para } i = 1, 2, \dots, n - 1 \quad (5.75)$$



**Figura 5-20** Splines cúbicas.

Como há  $n - 1$  intervalos, essa condição fornece  $2(n - 1) = 2n - 2$  equações.

2. Nos nós internos, as inclinações (derivadas primeiras) dos polinômios de intervalos adjacentes são iguais. Isso significa que, com a transição da curva que passa por um nó interno de um polinômio para outro, a inclinação deve ser contínua. A derivada primeira do  $i$ -ésimo polinômio é:

$$f_i'(x) = \frac{df_i}{dx} = 3a_i x^2 + 2b_i x + c_i \quad (5.76)$$

Para  $n$  pontos, o primeiro ponto interno é  $i = 2$  e o último é  $i = n - 1$ . Igualando as derivadas primeiras em cada um dos pontos internos, obtém-se:

$$3a_{i-1}x_i^2 + 2b_{i-1}x_i + c_{i-1} = 3a_i x_i^2 + 2b_i x_i + c_i \text{ para } i = 2, 3, \dots, n-1 \quad (5.77)$$

Como há  $n - 2$  pontos internos, essa condição fornece  $n - 2$  equações.

3. Nos nós internos, as derivadas segundas dos polinômios de intervalos adjacentes devem ser iguais. Isso significa que, com a transição da curva que passa por um nó interno de um polinômio para outro, a taxa de inclinação (curvatura) deve ser contínua. A derivada segunda do polinômio no  $i$ -ésimo intervalo é:

$$f_i''(x) = \frac{d^2f_i}{dx^2} = 6a_i x + 2b_i \quad (5.78)$$

Para  $n$  pontos, o primeiro ponto interno é  $i = 2$  e o último,  $i = n - 1$ . Igualando as derivadas segundas em todos os pontos internos, obtém-se:

$$6a_{i-1}x_i + 2b_{i-1} = 6a_i x_i + 2b_i \text{ para } i = 2, 3, \dots, n-1 \quad (5.79)$$

Como há  $n - 2$  pontos internos, essa condição fornece  $n - 2$  equações.

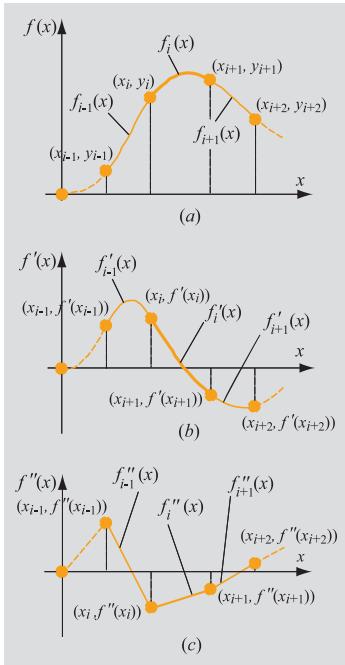
Juntas, as três condições fornecem  $4n - 6$  equações. Entretanto, os  $n - 1$  polinômios têm  $4n - 4$  coeficientes, e com isso duas equações (condições) adicionais são necessárias para que os coeficientes sejam obtidos. As condições geralmente escolhidas assumem que a derivada segunda seja nula no primeiro e no último ponto. Isso resulta em duas equações adicionais:

$$6a_1 x_1 + 2b_1 = 0 \text{ e } 6a_{n-1} x_n + 2b_{n-1} = 0 \quad (5.80)$$

Splines cúbicas com derivadas segundas igualadas a zero nos pontos finais do intervalo são chamadas de *splines cúbicas naturais*. A aplicação de todas as condições leva a um sistema de  $4n - 4$  equações com  $4n - 4$  coeficientes. Esse sistema pode ser resolvido com a aplicação de algum dos métodos do Capítulo 4.

### **Splines cúbicas baseadas em polinômios na forma de Lagrange**

A dedução de splines cúbicas usando a forma de Lagrange começa com a derivada segunda do polinômio. A Fig. 5.21 mostra a interpolação spline com polinômios cúbicos em (a), as derivadas primeiras dos polinômios em (b) e as suas derivadas segundas em (c). A figura mostra um  $i$ -ésimo intervalo com intervalos adjacentes  $i - 1$  e  $i + 1$ . A derivada segunda de um polinômio de terceira ordem é uma função linear. Isso significa que, dentro de cada spline, a derivada segunda é uma função



**Figura 5-21** (a) Polinômio de terceira ordem e (b) suas derivadas primeiras e (c) segundas.

linear de  $x$  (ver Fig. 5-21c). No  $i$ -ésimo intervalo, essa função linear pode ser escrita na forma de Lagrange:

$$f_i''(x) = \frac{x - x_{i+1}}{x_i - x_{i+1}} f_i''(x_i) + \frac{x - x_i}{x_{i+1} - x_i} f_i''(x_{i+1}) \quad (5.81)$$

onde os valores da derivada segunda do polinômio de segunda ordem nos pontos finais (nós) do intervalo são  $f_i''(x_i)$  e  $f_i''(x_{i+1})$ . O polinômio de terceira ordem no intervalo  $i$  pode ser determinado integrando-se a Eq. (5.81) duas vezes. A expressão resultante contém duas constantes de integração. Essas duas constantes podem ser determinadas a partir da condição que diz que os valores dos polinômios nos nós são conhecidos:

$$f_i(x_i) = (y_i) \quad \text{e} \quad f_i(x_{i+1}) = y_{i+1}$$

Uma vez determinadas as constantes de integração, a equação do polinômio de terceira ordem no intervalo  $i$  é dada por:

$$\begin{aligned} f_i(x) &= \frac{f_i''(x_i)}{6(x_{i+1} - x_i)} (x_{i+1} - x)^3 + \frac{f_i''(x_{i+1})}{6(x_{i+1} - x_i)} (x - x_i)^3 \\ &\quad + \left[ \frac{y_i}{x_{i+1} - x_i} - \frac{f_i''(x_i)(x_{i+1} - x_i)}{6} \right] (x_{i+1} - x) \\ &\quad + \left[ \frac{y_{i+1}}{x_{i+1} - x_i} - \frac{f_i''(x_{i+1})(x_{i+1} - x_i)}{6} \right] (x - x_i) \end{aligned} \quad (5.82)$$

para  $x_i \leq x \leq x_{i+1}$       e       $i = 1, 2, \dots, n - 1$

Para cada intervalo, a Eq. (5.82) contém duas incógnitas,  $f_i''(x_i)$  e  $f_i''(x_{i+1})$ . Esses são os valores da derivada segunda nos pontos finais do intervalo. As equações que relacionam os valores das derivadas segundas nos  $n - 2$  pontos internos podem ser deduzidas a partir da continuidade das derivadas primeiras dos polinômios de intervalos adjacentes nos pontos internos:

$$f'_i(x_{i+1}) = f'_{i+1}(x_{i+1}) \quad \text{para } i = 1, 2, \dots, n - 2 \quad (5.83)$$

Essa condição é aplicada usando a Eq. (5.82) para escrever as expressões em função de  $f_i(x)$  e  $f_{i+1}(x)$ , calculando as derivadas dessas expressões e substituindo essas derivadas na Eq. (5.83). Tais operações resultam (após um pouco de álgebra) nas equações a seguir:

$$\begin{aligned} &(x_{i+1} - x_i)f''(x_i) + 2(x_{i+2} - x_i)f''(x_{i+1}) + (x_{i+2} - x_{i+1})f''(x_{i+2}) \\ &= 6 \left[ \frac{y_{i+2} - y_{i+1}}{x_{i+2} - x_{i+1}} - \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right] \end{aligned} \quad (5.84)$$

para  $i = 1, 2, \dots, n - 2$

Esse é um sistema de  $n - 2$  equações lineares contendo  $n$  incógnitas.

### Como se determina o polinômio em cada intervalo?

- Para  $n$  pontos pertencentes a um conjunto de dados, há  $n - 1$  intervalos. O polinômio cúbico em cada intervalo é dado pela Eq. (5.82) (total de  $n - 1$  polinômios).
- Os  $n - 1$  polinômios contêm  $n$  coeficientes  $f''(x_1)$  a  $f''(x_n)$ . Estes são os valores das derivadas segundas dos polinômios nos pontos. Assume-se que a derivada segunda nos nós internos seja contínua. Isso significa que, nos nós internos, as derivadas segundas de polinômios de intervalos adjacentes são iguais. Conseqüentemente, para  $n$  pontos, há  $n$  valores (o valor da derivada segunda em cada ponto) que precisam ser determinados.
- As Eqs. (5.84) fornecem um sistema de  $n - 2$  equações lineares em função dos  $n$  coeficientes  $f''(x_1)$  a  $f''(x_n)$ . Para obter os valores dos coeficientes, duas relações adicionais são necessárias. Mais comumente, a derivada segunda nos pontos finais dos dados (o primeiro e o último ponto) é igualada a zero (splines cúbicas naturais):

$$f''(x_1) = 0 \quad \text{e} \quad f''(x_n) = 0 \quad (5.85)$$

Com essas condições, o sistema linear das Eqs. (5.84) pode ser resolvido, e os coeficientes podem ser substituídos nas equações dos polinômios (Eqs. (5.82)). Splines cúbicas com as derivadas segundas igualadas a zero nos pontos finais são chamadas de *splines cúbicas naturais*.

### Forma simplificada das equações

A forma das Eqs. (5.82) e (5.84) pode ser simplificada definindo-se  $h_i$  como o comprimento do  $i$ -ésimo intervalo (os intervalos não têm que ter o mesmo tamanho):

$$h_i = x_{i+1} - x_i \quad (5.86)$$

e  $a_i$  como a derivada segunda do polinômio no ponto  $x_i$ :

$$a_i = f''(x_i) \quad (5.87)$$

Com essas definições, a equação do polinômio no  $i$ -ésimo intervalo é:

$$\begin{aligned} f_i(x) &= \frac{a_i}{6h_i}(x_{i+1} - x)^3 + \frac{a_{i+1}}{6h_i}(x - x_i)^3 \\ &+ \left[ \frac{y_i}{h_i} - \frac{a_i h_i}{6} \right] (x_{i+1} - x) + \left[ \frac{y_{i+1}}{h_i} - \frac{a_{i+1} h_i}{6} \right] (x - x_i) \\ \text{para } x_i &\leq x \leq x_{i+1} \quad \text{e} \quad i = 1, 2, \dots, n-1 \end{aligned} \quad (5.88)$$

e o sistema de equações lineares que precisa ser resolvido para os termos  $a_i$  é dado por:

$$\begin{aligned} h_i a_i + 2(h_i + h_{i+1})a_{i+1} + h_{i+1}a_{i+2} &= 6 \left[ \frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i} \right] \\ \text{para } i &= 1, 2, \dots, n-2 \end{aligned} \quad (5.89)$$

Para realizar a interpolação com splines cúbicas, a Eq. (5.89) é usada para escrever um sistema de  $n - 2$  equações com  $n - 2$  incógnitas, de  $a_2$  a  $a_{n-1}$  (lembre

que, com splines cúbicas naturais,  $a_1$  e  $a_n$  são iguais a zero). A Eq. (5.89) resulta em um sistema tridiagonal de equações que pode ser resolvido de forma eficiente usando a Eq. (5.88). O Exemplo 5-8 mostra uma solução do problema do Exemplo 5-7 usando splines cúbicas.

### Nota sobre o uso de splines cúbicas no MATLAB

Há uma função disponível no MATLAB dedicada ao método das splines cúbicas. Entretanto, a opção 'cubic' (também chamada de 'pchip') não corresponde a esse método, sendo 'spline' a opção apropriada. Contudo, mesmo quando a opção 'spline' é usada, o usuário deve estar atento ao fato de que essa função não se refere às splines naturais descritas neste capítulo. As splines cúbicas disponíveis no MATLAB na opção 'spline' usam condições nó-a-nó nos pontos finais, isto é, no primeiro e no último ponto do conjunto de dados. A condição nó-a-nó se refere ao fato de que as derivadas terceiras são contínuas no segundo e no penúltimo ponto.

### Exemplo 5-8: Splines cúbicas

A partir do conjunto de dados com cinco pontos a seguir:

$x$	8	11	15	18	22
$y$	5	9	10	8	7

- (a) Determine as splines cúbicas naturais que fazem o ajuste dos dados.
- (b) Determine o valor interpolado de  $y$  em  $x = 12,7$ .
- (c) Trace um gráfico com os pontos do conjunto de dados e os polinômios interpoladores.

### SOLUÇÃO

- (a) Há cinco pontos ( $n = 5$ ), e portanto quatro splines ( $i = 1, \dots, 4$ ). A equação cúbica da  $i$ -ésima spline é:

$$f_i(x) = \frac{a_i}{6h_i}(x_{i+1} - x)^3 + \frac{a_{i+1}}{6h_i}(x - x_i)^3 + \left[ \frac{y_i}{h_i} - \frac{a_i h_i}{6} \right] (x_{i+1} - x) + \left[ \frac{y_{i+1}}{h_i} - \frac{a_{i+1} h_i}{6} \right] (x - x_i) \quad \text{para } i = 1, \dots, 4$$

onde  $h_i = x_{i+1} - x_i$ . As quatro equações contêm cinco coeficientes desconhecidos  $a_1, a_2, a_3, a_4$  e  $a_5$ . Nas splines cúbicas naturais, os coeficientes  $a_1$  e  $a_5$  são iguais a zero. Os outros três coeficientes são determinados a partir de um sistema linear de três equações dado pela Eq. (5.89).

Os valores dos termos  $h_i$  são:  $h_1 = x_2 - x_1 = 11 - 8 = 3$ ,  $h_2 = x_3 - x_2 = 15 - 11 = 4$   
 $h_3 = x_4 - x_3 = 18 - 15 = 3$ ,  $h_4 = x_5 - x_4 = 22 - 18 = 4$

$$\begin{aligned} i = 1 \quad h_1 a_1 + 2(h_1 + h_2)a_2 + h_2 a_3 &= 6 \left[ \frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1} \right] \\ 3 \cdot 0 + 2(3 + 4)a_2 + 4a_3 &= 6 \left[ \frac{10 - 9}{4} - \frac{9 - 5}{3} \right] \rightarrow 14a_2 + 4a_3 = -6,5 \end{aligned}$$

$$\begin{aligned} i = 2 \quad h_2 a_2 + 2(h_2 + h_3)a_3 + h_3 a_4 &= 6 \left[ \frac{y_4 - y_3}{h_3} - \frac{y_3 - y_2}{h_2} \right] \\ 4a_2 + 2(4 + 3)a_3 + 3a_4 &= 6 \left[ \frac{8 - 10}{3} - \frac{10 - 9}{4} \right] \rightarrow 4a_2 + 14a_3 + 3a_4 = -5,5 \end{aligned}$$

$$\begin{aligned} i = 3 \quad h_3 a_3 + 2(h_3 + h_4)a_4 + h_4 a_5 &= 6 \left[ \frac{y_5 - y_4}{h_4} - \frac{y_4 - y_3}{h_3} \right] \\ 3a_3 + 2(3+4)a_4 + 4 \cdot 0 &= 6 \left[ \frac{y_5 - y_4}{h_4} - \frac{y_4 - y_3}{h_3} \right] \quad \rightarrow \quad 3a_3 + 14a_4 = 2,5 \end{aligned}$$

O sistema de três equações lineares pode ser escrito na forma matricial:

$$\begin{bmatrix} 14 & 4 & 0 \\ 4 & 14 & 3 \\ 0 & 3 & 14 \end{bmatrix} \begin{bmatrix} a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} -6,5 \\ -5,5 \\ 2,5 \end{bmatrix} \quad (5.90)$$

O sistema da Eq. (5.90) é resolvido no MATLAB:

```
>> A = [14 4 0; 4 14 3; 0 3 14];
>> B = [-6.5; -5.5; 2.5];
>> coeficientes = (A\B)'
coeficientes =
```

-0.3665 -0.3421 0.2519



Conhecendo-se os coeficientes, os polinômios são (a partir da Eq. (5.88)):

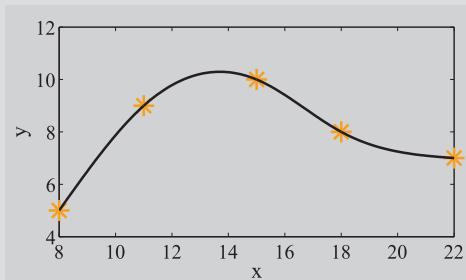
$$\begin{aligned} i = 1 \quad f_1(x) &= \frac{a_1}{6h_1}(x_2 - x)^3 + \frac{a_2}{6h_1}(x - x_1)^3 + \left[ \frac{y_1}{h_1} - \frac{a_1 h_1}{6} \right](x_2 - x) + \left[ \frac{y_2}{h_1} - \frac{a_2 h_1}{6} \right](x - x_1) \\ f_1(x) &= \frac{0}{6 \cdot 3}(11 - x)^3 + \frac{-0,3665}{6 \cdot 3}(x - 8)^3 + \left[ \frac{5}{3} - \frac{0 \cdot 3}{6} \right](11 - x) + \left[ \frac{9}{3} - \frac{-0,3665 \cdot 3}{6} \right](x - 8) \\ f_1(x) &= (-0,02036)(x - 8)^3 + 1,667(11 - x) + 3,183(x - 8) \quad \text{para } 8 \leq x \leq 11 \\ i = 2 \quad f_2(x) &= \frac{a_2}{6h_2}(x_3 - x)^3 + \frac{a_3}{6h_2}(x - x_2)^3 + \left[ \frac{y_2}{h_2} - \frac{a_2 h_2}{6} \right](x_3 - x) + \left[ \frac{y_3}{h_2} - \frac{a_3 h_2}{6} \right](x - x_2) \\ f_2(x) &= \frac{-0,3665}{6 \cdot 4}(15 - x)^3 + \frac{-0,3421}{6 \cdot 4}(x - 11)^3 + \left[ \frac{9}{4} - \frac{-0,3665 \cdot 4}{6} \right](15 - x) + \left[ \frac{10}{4} - \frac{-0,3421 \cdot 4}{6} \right](x - 11) \\ f_2(x) &= (-0,01527)(15 - x)^3 + (-0,01427)(x - 11)^3 + 2,494(15 - x) + 2,728(x - 11) \quad \text{para } 11 \leq x \leq 15 \\ i = 3 \quad f_3(x) &= \frac{a_3}{6h_3}(x_4 - x)^3 + \frac{a_4}{6h_3}(x - x_3)^3 + \left[ \frac{y_3}{h_3} - \frac{a_3 h_3}{6} \right](x_4 - x) + \left[ \frac{y_4}{h_3} - \frac{a_4 h_3}{6} \right](x - x_3) \\ f_3(x) &= \frac{-0,3421}{6 \cdot 3}(18 - x)^3 + \frac{0,2519}{6 \cdot 3}(x - 15)^3 + \left[ \frac{10}{3} - \frac{-0,3421 \cdot 3}{6} \right](18 - x) + \left[ \frac{8}{3} - \frac{0,2519 \cdot 3}{6} \right](x - 15) \\ f_3(x) &= (-0,019)(18 - x)^3 + 0,014(x - 15)^3 + 3,504(18 - x) + 2,5407(x - 15) \quad \text{para } 15 \leq x \leq 18 \\ i = 4 \quad f_4(x) &= \frac{a_4}{6h_4}(x_5 - x)^3 + \frac{a_5}{6h_4}(x - x_4)^3 + \left[ \frac{y_4}{h_4} - \frac{a_4 h_4}{6} \right](x_5 - x) + \left[ \frac{y_5}{h_4} - \frac{a_5 h_4}{6} \right](x - x_4) \\ f_4(x) &= \frac{0,2519}{6 \cdot 4}(22 - x)^3 + \frac{0}{6 \cdot 4}(x - 18)^3 + \left[ \frac{8}{4} - \frac{0,2519 \cdot 4}{6} \right](22 - x) + \left[ \frac{7}{4} - \frac{0 \cdot 4}{6} \right](x - 18) \\ f_4(x) &= 0,0105(22 - x)^3 + 1,832(22 - x) + 1,75(x - 18) \quad \text{para } 18 \leq x \leq 22 \end{aligned}$$

(b) O valor interpolado de  $y$  em  $x = 12,7$  é calculado com a substituição do valor de  $x$  em  $f_2(x)$ :

$$f_2(x) = (-0,01527)(15 - 12,7)^3 + (-0,01427)(12,7 - 11)^3 + 2,494(15 - 12,7) + 2,728(12,7 - 11)$$

$$f_2(x) = 10,11$$

(c) O gráfico ao lado mostra os pontos do conjunto de dados e o polinômio.

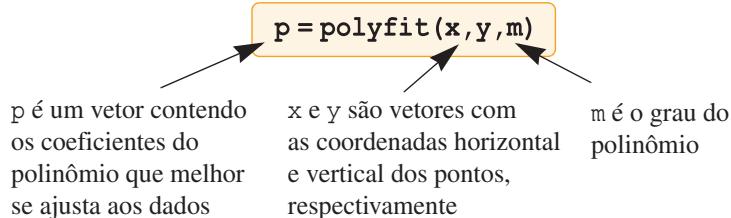


## 5.7 USO DE FUNÇÕES RESIDENTES DO MATLAB PARA FAZER O AJUSTE DE CURVAS E A INTERPOLAÇÃO

O MATLAB possui funções residentes para fazer o ajuste de curvas e a interpolação. Além disso, ele dispõe de uma ferramenta iterativa para fazer o ajuste de curvas, chamada de interface básica de ajuste. Esta seção descreve como usar as funções `polyfit` (para fazer o ajuste de curvas) e `interp1` (para interpolação). Os polinômios podem ser facilmente usados e manipulados matematicamente no MATLAB.

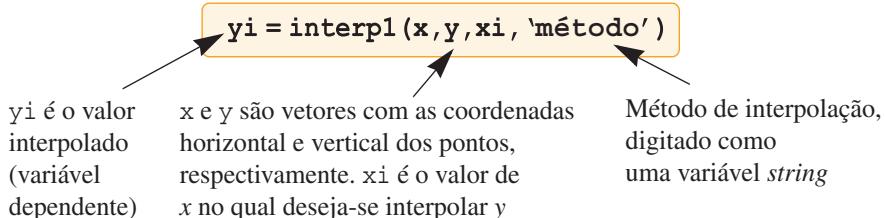
### O comando `polyfit`

O comando `polyfit` pode ser usado para fazer o ajuste de curvas em um conjunto de  $n$  pontos usando polinômios de vários graus e também para determinar o polinômio de ordem  $n - 1$  que passa por todos os pontos. A forma do comando é:



### O comando `interp1`

O comando `interp1` (o último caractere no comando é o número 1) executa a interpolação unidimensional em um ponto. O formato do comando é:



- O vetor  $x$  deve ser monotônico (os elementos devem estar na ordem ascendente ou descendente).
- $xi$  pode ser um escalar (interpolação em um ponto) ou um vetor (interpolação em vários pontos).  $yi$  é um escalar ou um vetor correspondendo aos valores interpolados nos pontos  $xi$ .
- O MATLAB pode interpolar usando um dentre vários métodos que podem ser especificados. Tais métodos incluem:
  - 'nearest' retorna o valor do ponto do conjunto de dados mais próximo ao ponto interpolado
  - 'linear' usa interpolação por spline linear.
  - 'spline' usa interpolação por spline cúbica com condições “nó-a-nó” onde as derivadas terceiras no segundo e no penúltimo ponto são contínuas. Esta não é a spline natural apresentada neste capítulo.
  - 'pchip' também chamada de 'cubic', usa a interpolação cúbica de Hermite por partes.
- Quando os métodos 'nearest' e 'linear' são usados, os valores de  $xi$  devem estar dentro do domínio de  $x$ . Se os métodos 'spline' ou 'pchip' forem usados,  $xi$  pode ter valores fora do domínio de  $x$  e a função `interp1` realiza a interpolação.
- O método 'spline' pode resultar em erros elevados se os dados de entrada estiverem não-uniformemente distribuídos, com alguns pontos mais próximos entre si do que outros.

A especificação do método é opcional. Se nenhum método for especificado, o padrão é 'linear'.

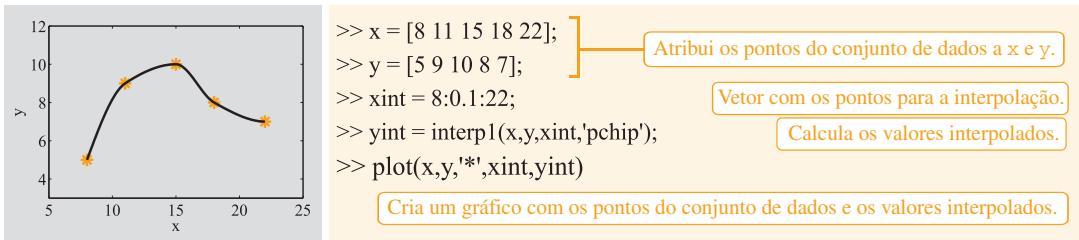
Dois exemplos do uso de funções residentes do MATLAB no ajuste de curvas e na interpolação são mostrados a seguir. Primeiramente, a função `polyfit` é usada na determinação do polinômio de quarta ordem que se ajusta aos pontos do Exemplo 5-3:

```
>> x = 0:0.4:6;
>> y = [0 3 4.5 5.8 5.9 5.8 6.2 7.4 9.6 15.6 20.7 26.7 31.1 35.6 39.3 41.5];
>> p = polyfit(x,y,4)
p =
-0.2644 3.1185 -10.1927 12.8780 -0.2746
```

O polinômio que corresponde a estes coeficientes é:

$$f(x) = -0,2644x^4 + 3,1185x^3 - 10,1927x^2 + 12,8780x - 0,2746.$$

No segundo exemplo, o comando `interp1` é usado para fazer a interpolação do Exemplo 5-8:



**Figura 5-22** Interpolação usando a função `interp1` do MATLAB.

O gráfico resultante é mostrado na Fig. 5-22. O MATLAB também tem uma ferramenta iterativa para fazer o ajuste de curvas e a interpolação, chamada de interface básica de ajuste de curvas. Para ativar essa interface, o usuário deve gerar um gráfico a partir do conjunto de dados e então selecionar, na janela da figura, **Basic Fitting** no menu **Tools** (uma descrição detalhada da interface básica de ajuste de curvas pode ser encontrada no livro: MATLAB, *An Introduction with Applications*, de Amos Gilat, Wiley, 2005).

## 5.8 AJUSTE DE CURVAS USANDO UMA COMBINAÇÃO LINEAR DE FUNÇÕES NÃO-LINEARES

O método dos mínimos quadrados, que foi aplicado na Seção 5.2 para fazer o ajuste de curvas usando funções lineares e também na Seção 5.4 usando polinômios quadráticos e de ordem superior, pode ser generalizado para permitir o ajuste de curvas a partir de uma combinação linear de funções não-lineares. Uma combinação linear de  $m$  funções não-lineares pode ser escrita como:

$$F(x) = C_1 f_1(x) + C_2 f_2(x) + C_3 f_3(x) + \dots + C_m f_m(x) = \sum_{j=1}^m C_j f_j(x) \quad (5.91)$$

onde  $f_1, f_2, \dots, f_m$  são funções prescritas, e  $C_1, C_2, \dots, C_m$  são coeficientes desconhecidos. Usando uma regressão por mínimos quadrados, a Eq. (5.91) é usada para fazer o ajuste de um conjunto de  $n$  pontos  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  minimizando o erro total dado pela soma dos quadrados dos resíduos:

$$E = \sum_{i=1}^n \left[ y_i - \sum_{j=1}^m C_j f_j(x_i) \right]^2 \quad (5.92)$$

A função  $E$  na Eq. (5.92) tem um mínimo nos valores dos coeficientes  $C_1, C_2, \dots, C_m$  onde a derivada parcial de  $E$  em relação a cada um dos coeficientes é igual a zero:

$$\frac{\partial E}{\partial C_k} = 0 \quad \text{para } k = 1, 2, \dots, m \quad (5.93)$$

A substituição da Eq. (5.92) na Eq. (5.93) resulta em:

$$\frac{\partial E}{\partial C_k} = \sum_{i=1}^n 2 \left[ y_i - \sum_{j=1}^m C_j f_j(x_i) \right] \left[ (-1) \frac{\partial}{\partial C_k} \left( \sum_{j=1}^m C_j f_j(x_i) \right) \right] = 0$$

para  $k = 1, 2, \dots, m$  (5.94)

Como os coeficientes  $C_1, C_2, \dots, C_m$  são independentes uns dos outros,

$$\frac{\partial}{\partial C_k} \left( \sum_{j=1}^m C_j f_j(x_i) \right) = f_k(x_i) \quad (5.95)$$

e com isso a Eq. (5.94) se torna:

$$\frac{\partial E}{\partial C_k} = - \sum_{i=1}^n 2 \left[ y_i - \sum_{j=1}^m C_j f_j(x_i) \right] f_k(x_i) = 0 \quad (5.96)$$

A última equação pode ser rescrita na forma:

$$\sum_{i=1}^n \sum_{j=1}^m C_j f_j(x_i) f_k(x_i) = \sum_{i=1}^n y_i f_k(x_i) \quad \text{para } k = 1, 2, \dots, m \quad (5.97)$$

Na Eq. (5.97),  $x_i, y_i$  e  $f_k(x_i)$  são grandezas conhecidas, e  $C_1, C_2, \dots, C_m$  são as incógnitas. O conjunto de Equações (5.97) é um sistema de  $m$  equações lineares em função das incógnitas  $C_1, C_2, \dots, C_m$ .

As funções  $f_k(x)$  podem ser funções quaisquer. Por exemplo, se  $F(x) = C_1 f_1(x) + C_2 f_2(x)$ , de tal forma que  $f_1(x) = 1$  e  $f_2(x) = x$ , então as Eqs. (5.97) se reduzem às Eqs. (5.9) e (5.10). Se as funções  $f_k(x)$  são escolhidas para que  $F(x)$  seja quadrática (isto é  $f_1(x) = 1$  e  $f_2(x) = x$  e  $f_3(x) = x^2$ ), então a Eq. (5.97) se reduz às Eqs. (5.23)-(5.25). Em geral, as funções  $f_k(x)$  são escolhidas em função de alguma teoria capaz de prever a tendência dos dados. O Exemplo 5-9 mostra como o método é usado para fazer o ajuste de um conjunto de dados empregando funções não-lineares.

### Exemplo 5-9: Ajuste de curvas usando uma combinação linear de funções não-lineares

Os seguintes dados são obtidos a partir de testes em túnel de vento, para a variação da relação entre a velocidade tangencial de um vórtice e a velocidade de fluxo  $y = V_\theta/V_\infty$  versus a relação entre a distância do núcleo do vórtice e o eixo principal da asa de um avião,  $x = R/C$ :

x	0,6	0,8	0,85	0,95	1,0	1,1	1,2	1,3	1,45	1,6	1,8
y	0,08	0,06	0,07	0,07	0,07	0,06	0,06	0,06	0,05	0,05	0,04

A teoria prediz que a relação entre  $x$  e  $y$  deve ter a forma  $y = \frac{A}{x} + \frac{Be^{-2x^2}}{x}$ . Determine os valores de  $A$  e  $B$  usando o método de mínimos quadrados para fazer o ajuste dos dados acima.

### SOLUÇÃO

Na notação da Eq. (5.91), a função de aproximação é  $F(x) = C_1 f_1(x) + C_2 f_2(x)$ , com  $F(x) = y$ ,  $C_1 = A$ ,  $C_2 = B$ ,  $f_1(x) = \frac{1}{x}$  e  $f_2(x) = \frac{e^{-2x^2}}{x}$ . A equação tem dois termos e, com isso,  $m = 2$ ; como há 11 pontos,  $n = 11$ . A substituição dessa informação na Eq. (5.97) fornece o seguinte sistema de duas equações lineares para  $A$  e  $B$ :

$$\begin{aligned} \sum_{i=1}^{11} A \frac{1}{x_i} \frac{1}{x_i} + \sum_{i=1}^{11} B \frac{e^{-2x_i^2}}{x_i} \frac{1}{x_i} &= \sum_{i=1}^{11} y_i \frac{1}{x_i} \quad \text{para } k = 1 \\ \sum_{i=1}^{11} A \frac{1}{x_i} \frac{e^{-2x_i^2}}{x_i} + \sum_{i=1}^{11} B \frac{e^{-2x_i^2}}{x_i} \frac{e^{-2x_i^2}}{x_i} &= \sum_{i=1}^{11} y_i \frac{e^{-2x_i^2}}{x_i} \quad \text{para } k = 2 \end{aligned}$$

Essas duas equações podem ser rescritas como:

$$\begin{aligned} A \sum_{i=1}^{11} \frac{1}{x_i^2} + B \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} &= \sum_{i=1}^{11} y_i \frac{1}{x_i} \\ A \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} + B \sum_{i=1}^{11} \frac{e^{-4x_i^2}}{x_i^2} &= \sum_{i=1}^{11} y_i \frac{e^{-2x_i^2}}{x_i} \end{aligned}$$

O sistema pode ser escrito na forma matricial:

$$\begin{bmatrix} \sum_{i=1}^{11} \frac{1}{x_i^2} & \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} \\ \sum_{i=1}^{11} \frac{e^{-2x_i^2}}{x_i^2} & \sum_{i=1}^{11} \frac{e^{-4x_i^2}}{x_i^2} \end{bmatrix} \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{11} y_i \frac{1}{x_i} \\ \sum_{i=1}^{11} y_i \frac{e^{-2x_i^2}}{x_i} \end{bmatrix}$$

O sistema é resolvido no MATLAB. O programa a seguir resolve o sistema e em seguida traça um gráfico contendo os pontos do conjunto de dados e a curva de ajuste.

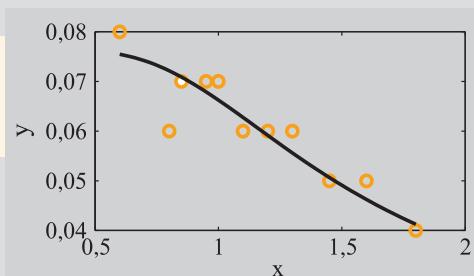
```
x = [0.6 0.8 0.85 0.95 1.0 1.1 1.2 1.3 1.45 1.6 1.8];
y = [0.08 0.06 0.07 0.07 0.06 0.06 0.06 0.05 0.05 0.04];
a(1,1) = sum(1./x.^2);
a(1,2) = sum(exp(-2*x.^2)./x.^2);
a(2,1) = a(1,2);
a(2,2) = sum(exp(-4*x.^2)./x.^2);
b(1,1) = sum(y./x);
b(2,1) = sum((y.*exp(-2*x.^2))./x);
AB = a\b;
xfit = 0.6:0.02:1.8;
yfit = AB(1)./xfit + AB(2)*exp(-2*xfit.^2)./xfit;
plot(x,y,'o',xfit,yfit)
```

Quando o programa é executado, os coeficientes obtidos são mostrados na janela de comandos do MATLAB (os dois elementos do vetor AB), e um gráfico contendo os pontos do conjunto de dados e a curva de ajuste é criado.

Janela de comandos:

```
AB =
0.0743
-0.0597
```

← Coeficiente A.  
← Coeficiente B.



## 5.9 PROBLEMAS

### Problemas de solução manual

Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos. Não utilize funções residentes do MATLAB para realizar o ajuste de curvas e interpolação.

**5.1** Com base no seguinte conjunto de dados:

<b>x</b>	2	5	6	8	9	13	15
<b>y</b>	7	8	10	11	12	14	15

- (a) Use a regressão linear por mínimos quadrados para determinar os coeficientes  $m$  e  $b$  da função  $y = mx + b$  que melhor se ajusta aos dados.
- (b) Use a Eq. (5.5) para determinar o erro global.

**5.2** Com base no seguinte conjunto de dados:

<b>x</b>	-7	-5	-1	0	2	5	6
<b>y</b>	15	12	5	2	0	-5	-9

- (a) Use a regressão linear por mínimos quadrados para determinar os coeficientes  $m$  e  $b$  da função  $y = mx + b$  que melhor se ajusta aos dados.
- (b) Use a Eq. (5.5) para determinar o erro global.

**5.3** Os dados a seguir fornecem a população aproximada do mundo em anos selecionados de 1850 até 2000.

<b>Ano</b>	1850	1900	1950	1980	2000
<b>População (Bilhões)</b>	1,3	1,6	3	4,4	6

Assuma que o crescimento da população possa ser modelado por uma função exponencial  $p = be^{mx}$ , onde  $x$  é o ano e  $p$  é a população em bilhões. Linearize essa função (Seção 5.3) e use a regressão linear por mínimos quadrados para determinar as constantes  $b$  e  $m$  para as quais a função fornece o melhor ajuste para os dados. Use essa equação para estimar a população em 1970.

**5.4** Com base no seguinte conjunto de dados:

<b>x</b>	-0,2	-0,1	0,2	0,7	1,3
<b>y</b>	5,2	3	0,6	0,4	0,2

Determine os coeficientes  $m$  e  $b$  da função  $y = \frac{1}{mx + b}$  que melhor se ajusta aos dados (linearize a equação (Seção 5.3) e use a regressão linear por mínimos quadrados para determinar o valor dos coeficientes).

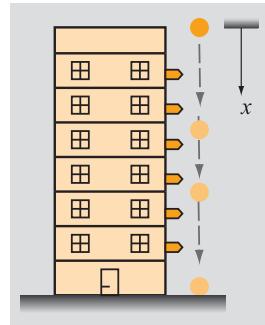
**5.5** Com base no seguinte conjunto de dados:

<b>x</b>	1	3	5	7	10
<b>y</b>	2,2	5,0	5,5	6,1	6,6

Determine os coeficientes  $m$  e  $b$  da função  $y = \frac{mx}{b+x}$  que melhor se ajusta aos dados (linearize a equação (Seção 5.3) e use a regressão linear por mínimos quadrados para determinar o valor dos coeficientes).

**5.6** Para medir  $g$  (a aceleração da gravidade), realize o experimento a seguir. Uma bola é solta do topo de um edifício de 30 m de altura. À medida que o objeto vai caindo, sua velocidade  $v$  vai sendo medida em várias alturas por sensores presos ao edifício. Os dados medidos no experimento são fornecidos na tabela.

<b>x (m)</b>	0	5	10	15	20	25
<b>v (m/s)</b>	0	9,85	14,32	17,63	19,34	22,41



Em termos das coordenadas mostradas na figura (positivo para baixo), a velocidade da bola em função da distância  $x$  é dada por  $v^2 = 2gx$ . Usando a regressão linear, determine o valor experimental de  $g$ .

**5.7** A pressão atmosférica  $p$  em função da altura  $h$  pode ser modelada por uma função exponencial na forma  $p = be^{-mh}$ . Os valores a seguir correspondem à pressão medida em diferentes alturas. Usando a regressão linear, determine as constantes  $m$  e  $b$  que fazem o melhor ajuste dos dados. Use a equação para estimar a pressão atmosférica em uma altura de 7000 m.

<b>h (m)</b>	0	5.000	10.000	15.000	20.000
<b>p (Pa)</b>	100.000	47.500	22.600	10.800	5.100

**5.8** No processo de fabricação de fibras eletroforéticas, o diâmetro da fibra  $d$  está relacionado à corrente  $I$ . Os seguintes dados são medidos durante a produção:

<b>I (nA)</b>	300	300	350	400	400	500	500	650	650
<b>d (μm)</b>	22	26	27	30	34	33	33,5	37	42

A relação entre a corrente e o diâmetro pode ser modelada com uma equação na forma  $d = a + b\sqrt{I}$ . Use os dados para determinar as constantes  $a$  e  $b$  que fazem o melhor ajuste dos dados.

**5.9** Determine os coeficientes do polinômio  $y = a_2x^2 + a_1x + a_0$  que fazem o melhor ajuste dos dados fornecidos no Problema 5.5.

**5.10** Usando o método dos mínimos quadrados (Seção 5.8), faça o ajuste dos dados a seguir usando a combinação de uma linha reta,  $\sin x$  e  $e^x$ .

<b>x</b>	0,1	0,4	0,5	0,7	0,7	0,9
<b>y</b>	0,61	0,92	0,99	1,52	1,47	2,03

**5.11** A economia de um carro (km/litro) varia com sua velocidade. Em um experimento, são feitas as cinco medições a seguir.

<b>Velocidade (km/h)</b>	16	40	64	88	112
<b>Economia (km/litro)</b>	4,2	9,2	10	10,7	8,6

Determine o polinômio de Lagrange de quarta ordem que passa pelos pontos. Use esse polinômio para calcular a economia de combustível a 105 km/h.

**5.12** Determine o polinômio interpolador de Newton de quarta ordem que passa pelos pontos dados no Problema 5.11. Use esse polinômio para calcular a economia de combustível a 48 km/h.

**5.13** Com base no seguinte conjunto de dados:

<b>x</b>	1	2,5	2	3	4	5
<b>y</b>	1	7	5	8	2	1

(a) Escreva o polinômio de Lagrange que passa pelos pontos e então use-o para calcular o valor interpolado de  $y$  em  $x = 3,5$ .

(b) Escreva o polinômio de Newton que passa pelos pontos e então use-o para calcular o valor interpolado de  $y$  em  $x = 3,5$ .

**5.14** Interpole os dados do Problema 5.11 usando splines lineares e calcule a economia de combustível à velocidade de

(a) 48 km/h (b) 105 km/h

**5.15** Interpole os dados do Problema 5.11 usando splines quadráticas e calcule a economia de combustível à velocidade de

(a) 48 km/h (b) 105 km/h

**5.16** Interpole os dados do Problema 5.11 usando splines cúbicas naturais (baseadas nos polinômios de Lagrange [Eqs. (5.86)-(5.89)]) e calcule a economia de combustível à velocidade de

(a) 48 km/h (b) 105 km/h

### Problemas de programação no MATLAB

Resolva os seguintes problemas usando o MATLAB. Não use funções residentes para realizar o ajuste de curvas e a interpolação.

**5.17** Modifique a função RegressaoLinear criada no Programa 5-1. Além de determinar as constantes  $a_1$  e  $a_0$ , a função modificada deve calcular o

erro global  $E$  de acordo com a Eq. (5.6). Chame a função de  $[a, Er] = \text{RegLin}(x, y)$ . Os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos. O argumento de saída  $a$  é um vetor com dois elementos contendo os valores das constantes  $a_1$  e  $a_0$ . O argumento de saída  $Er$  é o valor do erro global.

- (a) Use a função para resolver o Exemplo 5-1
- (b) Use a função para resolver o Problema 5.2

**5.18** Escreva uma função no MATLAB que determine o melhor ajuste de uma função exponencial na forma  $y = be^{-mx}$  a um determinado conjunto de dados. Chame a função de  $[b, m] = \text{ExpoFit}(x, y)$ , onde os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos e os argumentos de saída  $b$  e  $m$  são os valores dos coeficientes. A função `ExpoFit` deve usar a abordagem descrita na Seção 5.3 para determinar o valor dos coeficientes. Use a função para resolver o Problema 5.7.

**5.19** Escreva uma função no MATLAB que determine o melhor ajuste de uma função na forma  $y = bx^m$  a um determinado conjunto de dados. Chame a função de  $[b, m] = \text{PowerFit}(x, y)$ , onde os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos e os argumentos de saída  $b$  e  $m$  são os valores dos coeficientes. A função `PowerFit` deve usar a abordagem descrita na Seção 5.3 para determinar o valor dos coeficientes. Use a função para resolver o Problema 5.3.

**5.20** Escreva uma função no MATLAB que determine o melhor ajuste de uma função na forma  $f(x) = a_2x^2 + a_1x + a_0$  a um determinado conjunto de dados. Chame a função de  $a = \text{QuadFit}(x, y)$ , onde os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos e o argumento de saída  $a$  é um vetor de três elementos contendo os valores dos coeficientes  $a_2$ ,  $a_1$  e  $a_0$ .

- (a) Use a função para determinar o polinômio quadrático que faz o melhor ajuste dos dados do Exemplo 5-2.
- (b) Escreva um programa que faça o traçado dos pontos do conjunto de dados e do polinômio quadrático que faz o melhor ajuste.

**5.21** Escreva uma função no MATLAB que determine o melhor ajuste de uma função na forma  $f(x) = a_3x^3 + a_2x^2 + a_1x + a_0$  a um determinado conjunto de dados. A função também deve calcular o erro global  $E$  de acordo com a Eq. (5.6). Chame a função de  $[a, Er] = \text{CubicPolyFit}(x, y)$ , onde os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos e o argumento de saída  $a$  é um vetor de quatro elementos contendo os valores dos coeficientes  $a_3$ ,  $a_2$ ,  $a_1$  e  $a_0$ . O argumento de saída  $Er$  é o valor do erro global.

- (a) Use `CubicPolyFit` para determinar o polinômio cúbico que faz o melhor ajuste dos dados do Exemplo 5-3.

- (b) Escreva um programa que faça o traçado dos pontos do conjunto de dados e do polinômio cúbico que faz o melhor ajuste.

**5.22** Escreva uma função no MATLAB que faça interpolação usando splines cúbicas naturais. Chame a função de  $Yint = CubicSplines(x, y, Xint)$ , onde os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos e  $Xint$  é a coordenada  $x$  do ponto interpolado. O argumento de saída  $Yint$  é o valor de  $y$  no ponto interpolado.

- (a) Use a função nos dados do Exemplo 5-8 para calcular o valor interpolado em  $x = 12,7$ .
- (b) Use a função nos dados do Problema 5.27 para calcular a entalpia por unidade de massa em  $T = 14000$  K e  $T = 24000$  K.

### Problemas na matemática, na ciência e na engenharia

Resolva os seguintes problemas no ambiente MATLAB. Use programas apresentados neste capítulo, programas desenvolvidos em problemas já resolvidos ou funções residentes do MATLAB.

**5.23** Em um teste de tensão uniaxial, estica-se em uma máquina um corpo de prova no formato de um osso canino. Durante o teste, a força  $F$  é aplicada no corpo de prova e o comprimento  $L$  da deformação é medido. A tensão real,  $\sigma_t$ , e a deformação real,  $\varepsilon_t$ , são definidas por:

$$\sigma_t = \frac{F}{A_0 L_0} \quad \text{e} \quad \varepsilon_t = \ln \frac{L}{L_0}$$

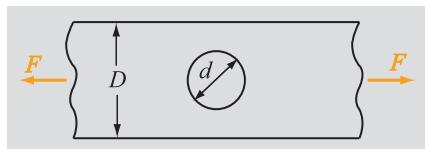
onde  $A_0$  e  $L_0$  são a área inicial da seção reta e a deformação inicial, respectivamente. A curva tensão-deformação real é freqüentemente modelada por:

$$\sigma_t = K \varepsilon_t^m$$

A tabela a seguir apresenta os valores de  $F$  e  $L$  medidos em um experimento. Use a abordagem da Seção 5.3 para determinar o valor dos coeficientes  $K$  e  $m$  que fazem o melhor ajuste dos dados. A área inicial da seção reta e a deformação inicial são  $A_0 = 1,25 \times 10^{-4} \text{ m}^2$  e  $L_0 = 0,0125 \text{ m}$ .

<b><math>F</math> (kN)</b>	24,6	29,3	31,5	33,3	34,8	35,7	36,6	37,5	38,8	39,6	40,4
<b><math>L</math> (mm)</b>	12,58	12,82	12,91	12,95	13,05	13,21	13,35	13,49	14,08	14,21	14,48

**5.24** O fator concentração de tensão  $k$  é a relação entre a tensão máxima  $\sigma_{\max}$  e a tensão média  $\sigma_{\text{med}}$ ,  $k = \sigma_{\max} / \sigma_{\text{med}}$ . Para uma placa com espessura  $D$  possuindo um furo central de diâmetro  $d$  carregada com uma força axial  $F$  (ver figura), a tensão máxima ocorre na



borda do buraco, e a tensão média é dada por  $\sigma_{\text{média}} = F/[t(D - d)]$ , onde  $t$  é a dureza da placa. O fator concentração de tensão medido em cinco testes com placas possuindo diferentes relações  $d/D$  é mostrado na tabela.

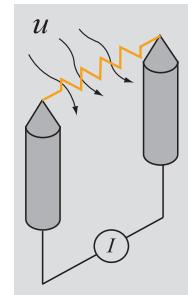
$d/D$	0,05	0,25	0,45	0,65	0,85
$k$	2,91	2,40	2,17	2,11	2,03

- (a) Use uma função exponencial  $k = be^{m(d/D)}$  para modelar a relação entre  $k$  e  $d/D$ . Determine os valores de  $b$  e  $m$  que fazem o melhor ajuste dos dados.
- (b) Trace um gráfico com os pontos do conjunto de dados e o modelo de ajuste.
- (c) Use o modelo para predizer o fator de concentração de tensão para  $d/D = 0,15$ .

**5.25** Um anemômetro de fio quente é um dispositivo usado para medir a velocidade de fluxo a partir do efeito de resfriamento causado pelo fluxo na resistência de um fio quente. Os dados a seguir são obtidos em testes de calibração.

$u$ (m/s)	1,44	3,81	6,11	8,63	11,42	12,63	14,75	16,78
$V$ (Volt)	7,18	7,3	7,37	7,42	7,47	7,5	7,53	7,55

$u$ (m/s)	20,35	18,03	16,03	14,39	13,03	9,97	7,75	2,49
$V$ (Volt)	7,58	7,56	7,55	7,53	7,51	7,47	7,44	7,28



Determine os coeficientes da função exponencial  $u = Ae^{BV}$  que faz o melhor ajuste dos dados,

- (a) usando a função *ExpoFit* desenvolvida no Problema 5.18.
- (b) usando funções residentes do MATLAB;

Em cada letra, trace um gráfico com os pontos do conjunto de dados e o modelo de ajuste.

**5.26** A tensão limite de escoamento  $\sigma_y$  de vários metais varia com o tamanho dos grãos. Muitas vezes, a relação entre o tamanho do grão e a tensão limite de escoamento é modelada com a equação de Hall-Petch:

$$\sigma_y = \sigma_0 + kd^{\left(\frac{-1}{2}\right)}$$

Os dados a seguir resultam da medição do tamanho médio dos grãos e da tensão limite de escoamento de uma amostra.

$d$ (mm)	0,006	0,011	0,017	0,025	0,039	0,060	0,081	0,105
$\sigma_y$ (MPa)	334	276	249	235	216	197	194	182

- (a) Determine as constantes  $\sigma_0$  e  $d$  tais que a equação de Hall-Petch faça o melhor ajuste dos dados. Trace um gráfico representando os pontos com

marcadores circulares e a equação de Hall-Petch como uma linha contínua. Use a equação de Hall-Petch para estimar a tensão de escoamento de um espécime com grãos de 0,05 mm.

- (b) Use a função QuadFit do Problema 5.20 para determinar a função quadrática que faz o melhor ajuste dos dados. Trace um gráfico representando os pontos com marcadores circulares e a função quadrática como uma linha contínua. Use a equação quadrática para estimar a tensão de escoamento de um espécime com grãos de 0,05 mm.

**5.27** Valores de entalpia por unidade de massa  $h$  de um plasma de Argônio em equilíbrio (íons Ar,  $\text{Ar}^+$ ,  $\text{A}^{++}$ ,  $\text{A}^{+++}$  e elétrons) *versus* temperatura são:

$T \times 10^3$ (K)	5	7,5	10	12,5	15	17,5	20	22,5	25	27,5	30
$h$ (MJ/kg)	3,3	7,5	41,8	51,8	61	101,1	132,9	145,5	171,4	225,8	260,9

Escreva um programa que use a interpolação para calcular  $h$  em temperaturas entre 5000 K e 30000 K em incrementos de 500 K. O programa deve gerar um gráfico que mostre os valores interpolados e os dados da tabela (use um asterisco para representá-los).

- (a) Para realizar a interpolação, use a função CubicSplines do Problema 5.22.  
 (b) Para realizar a interpolação, use a função residente do MATLAB `interp1` com a opção `spline`.

**5.28** Os dados a seguir correspondem à medição do coeficiente de taxa  $k$  para a reação  $\text{CH}_4 + \text{O} \rightarrow \text{CH}_3 + \text{OH}$  em diferentes temperaturas  $T$ .

$T$ (K)	595	623	761	849	989	1076	1146	1202	1382	1445	1562
$k \times 10^{20}$ (m <sup>3</sup> /s)	2,12	3,12	14,4	30,6	80,3	131	186	240	489	604	868

- (a) Use o método dos mínimos quadrados para fazer o ajuste de uma função na forma  $\ln(k) = C + b \ln(T) - \frac{D}{T}$  aos dados. Determine as constantes  $C$ ,  $b$ , e  $D$  realizando a combinação linear das funções  $f_1(T) = 1$ ,  $f_2(T) = \ln(T)$  e  $f_3(T) = -1/T$  (Seção 5.8).  
 (b) Usualmente, o coeficiente de taxa é expresso na forma da equação de Arrhenius  $k = AT^b e^{-E_a/(RT)}$ , onde  $A$  e  $b$  são constantes,  $R = 8,314$  J/mole/K é a constante universal dos gases, e  $E_a$  é a energia de ativação da reação. Tendo determinado as constantes  $C$ ,  $b$  e  $D$  na letra (a), deduza os valores de  $A$  (m<sup>3</sup>/s) e  $E_a$  (J/mole) na expressão de Arrhenius.

## Diferenciação Numérica

### Tópicos centrais

Aproximação da derivada por diferenças finitas (6.2)

Fórmulas de diferenças finitas usando a expansão em série de Taylor (6.3)

Resumo de fórmulas de diferenças finitas para a diferenciação numérica (6.4)

Fórmulas de diferenciação usando polinômios de Lagrange (6.5)

Diferenciação usando o ajuste de curvas (6.6)

Uso de funções residentes do MATLAB para realizar a diferenciação numérica (6.7)

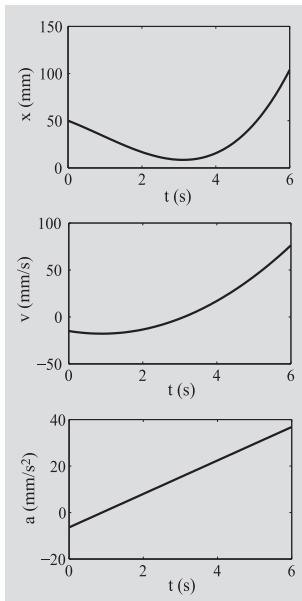
### Tópicos Complementares

Extrapolação de Richardson (6.8)

Erros na diferenciação numérica (6.9)

Diferenciação parcial numérica (6.10)

### 6.1 FUNDAMENTOS



**Figura 6-1** Posição, velocidade e aceleração em função do tempo.

A diferenciação dá uma medida da taxa na qual uma grandeza varia. Taxas de variação de grandes aparições em muitas disciplinas, especialmente na ciência e na engenharia. Uma das taxas de variação mais fundamentais é a relação entre posição, velocidade e aceleração. Se a posição  $x$  de um objeto que se move em linha reta é conhecida em função do tempo  $t$  (a curva de cima na Fig. 6-1):

$$x = f(t) \quad (6.1)$$

a velocidade do objeto,  $v(t)$ , é a derivada da posição em relação ao tempo (curva central na Fig. 6-1):

$$v = \frac{df(t)}{dt} \quad (6.2)$$

A velocidade  $v$  é a inclinação da curva posição-tempo. De forma similar, a aceleração do objeto,  $a(t)$ , é a derivada da velocidade em relação ao tempo (curva de baixo na Fig. 6-1):

$$a = \frac{dv(t)}{dt} \quad (6.3)$$

A aceleração  $a$  é a inclinação da curva velocidade-tempo.

Muitos modelos físicos e de engenharia são expressos em termos de taxas. Em um circuito elétrico, a corrente em um capacitor está relacionada à derivada temporal da tensão. Na análise da condução de calor, a quantidade de fluxo de calor é determinada a partir da derivada da temperatura.

A diferenciação também é usada na obtenção dos máximos e mínimos de uma função.

### A necessidade da diferenciação numérica

A função a ser diferenciada pode ser dada como uma expressão analítica ou como um conjunto de pontos discretos (dados tabulados). Quando a função é dada como uma expressão matemática simples, a derivada pode ser determinada analiticamente. Quando a diferenciação analítica da expressão é difícil ou impossível, deve-se utilizar a diferenciação numérica. Quando a função é especificada como um conjunto de pontos discretos, a diferenciação é feita com o uso de um método numérico.

A diferenciação numérica também é importante em alguns dos métodos numéricos empregados na solução de equações diferenciais, conforme mostrado nos Capítulos 8 e 9.

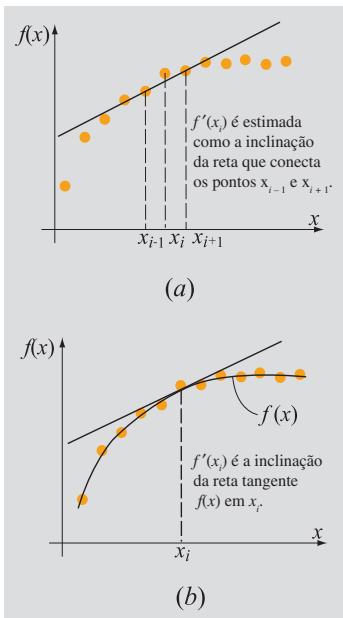
### Abordagens para a diferenciação numérica

A diferenciação numérica é realizada em dados especificados como um conjunto de pontos discretos. Em muitos casos, os dados são medidos ou gravados em experimentos, ou podem ser o resultado de cálculos numéricos em grande escala. Se houver a necessidade de se calcular a derivada de uma função dada na forma analítica, então a diferenciação é feita usando pontos discretos da função. Isso significa que, em todos os casos, a diferenciação numérica é feita usando os valores dos pontos.

Para um determinado conjunto de dados, duas abordagens podem ser usadas no cálculo da aproximação numérica da derivada em um ponto. Uma delas é a **aproximação por diferenças finitas**.

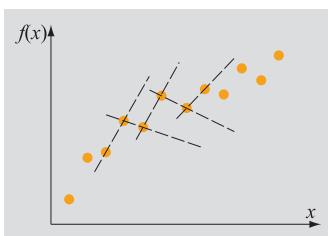
A aproximação da derivada em um ponto  $x_i$  por diferenças finitas se baseia nos valores dos pontos na vizinhança de  $x_i$ . Essa abordagem é usada na Fig. 6-2a, onde a derivada no ponto  $x_i$  é aproximada pela inclinação da reta que liga o ponto antes de  $x_i$  ao ponto depois de  $x_i$ . A precisão da aproximação por diferenças finitas depende da precisão dos pontos do conjunto de dados, do espaçamento entre os pontos e da fórmula específica usada na aproximação. A fórmula mais simples aproxima a derivada como sendo a inclinação da reta que conecta dois pontos adjacentes. A aproximação por diferenças finitas é tratada nas Seções 6.2 e 6.3.

A segunda abordagem corresponde à aproximação dos pontos utilizando uma expressão analítica que possa ser facilmente diferenciada, seguida do cálculo da derivada com a diferenciação dessa expressão analítica. A expressão analítica aproximada pode ser deduzida com o uso do ajuste de curvas. Essa abordagem é ilustrada na Fig. 6-2b, onde se faz o ajuste de uma curva  $f(x)$  para representar os pontos. A derivada no ponto  $x_i$  é obtida com a diferenciação analítica da função de aproximação e com a sua avaliação no ponto  $x_i$ . Essa abordagem é descrita na Seção 6.6.



**Figura 6-2** Diferenciação numérica usando (a) aproximação por diferenças finitas e (b) função de aproximação.

### Ruído e dispersão nos dados



**Figura 6-3** Diferenciação numérica de dados com dispersão.

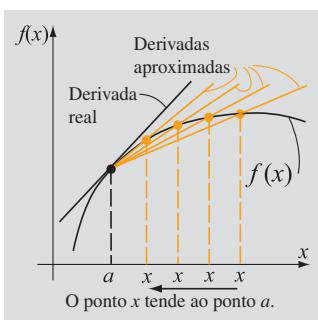
Quando os dados a serem diferenciados são obtidos a partir de medições experimentais, eles usualmente apresentam dispersão por causa de erros experimentais ou incertezas nas medições (por exemplo, ruído elétrico). Um conjunto de dados apresentando dispersão é mostrado esquematicamente na Fig. 6-3. Se esse conjunto de dados for diferenciado com a utilização da forma mais simples da aproximação por diferenças finitas (inclinação da reta que conecta dois pontos adjacentes), então grandes variações (positivas e negativas) serão vistas no valor da derivada de ponto a ponto. É óbvio, a partir dos dados na figura, que o valor de  $y$  geralmente aumenta com o aumento de  $x$ .

Isso significa que a derivada de  $y$  em relação a  $x$  é positiva. Melhores resultados podem ser obtidos com o uso de fórmulas de aproximação por diferenças finitas de ordem superior, que usem os valores de mais de dois pontos. Por exemplo (ver fórmulas na Seção 6.4), há fórmulas de diferenças finitas usando quatro, cinco e sete pontos. Conforme mencionado anteriormente, a diferenciação também pode ser feita com uso de uma função analítica que represente o conjunto de dados de forma aproximada, com a posterior diferenciação dessa função. Neste caso, os dados são suavizados antes da diferenciação, o que elimina o problema de inclinações serem incorretamente amplificadas em pontos sucessivos.

## 6.2 APROXIMAÇÃO DA DERIVADA POR DIFERENÇAS FINITAS

A derivada  $f'(x)$  de uma função  $f(x)$  no ponto  $x = a$  é definida como:

$$\frac{df(x)}{dx} \Big|_{x=a} = f'(a) = \lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} \quad (6.4)$$



**Figura 6-4** Definição de derivada.

Graficamente, a definição é ilustrada na Fig. 6-4. A derivada é o valor da inclinação da reta tangente à função em  $x = a$ . A derivada é obtida com a escolha de um ponto  $x$  próximo a  $x = a$  e o cálculo da inclinação da reta que conecta os dois pontos. A precisão do cálculo da derivada feito dessa forma aumenta à medida que o ponto  $x$  se aproxima do ponto  $a$ . No limite em que o ponto  $x$  tende ao ponto  $a$ , a derivada é a inclinação da reta tangente a  $f(x)$  em  $x = a$ . Na disciplina de Cálculo, a aplicação da condição limite descrita pela Eq. (6.4), que diz que  $x$  tende ao ponto  $a$ , é usada na dedução de regras de diferenciação que fornecem uma expressão analítica para a derivada.

Na aproximação de derivadas usando diferenças finitas, valores da função em diferentes pontos na vizinhança do ponto  $x = a$  são usados na estimativa da inclinação. Deve ser lembrado que a função sendo diferenciada é prescrita como um conjunto de pontos discretos. Existem várias fórmulas de aproximação por diferenças finitas. Três dessas fórmulas, nas quais a derivada é calculada a partir dos valores de dois pontos, são apresentadas nesta seção.

### Fórmulas de diferença progressiva, regressiva e central para a derivada primeira

As fórmulas de diferenças finitas progressiva, regressiva e central são as mais simples aproximações da derivada por diferenças finitas. Nessas aproximações, ilustradas na Fig. 6-5, a derivada no ponto ( $x_i$ ) é calculada a partir do valor de dois pontos. A derivada é estimada como a inclinação da reta que conecta esses dois pontos.

- A **diferença progressiva** é a inclinação da reta que conecta os pontos  $(x_i, f(x_i))$  e  $(x_{i+1}, f(x_{i+1}))$ :

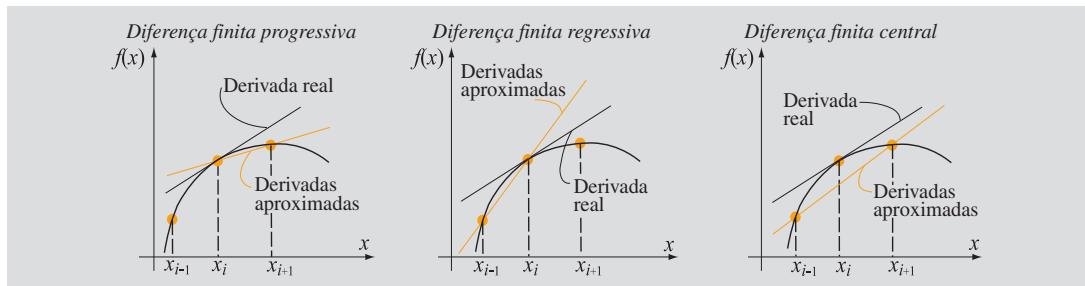
$$\left. \frac{df}{dx} \right|_{x=x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \quad (6.5)$$

- A **diferença regressiva** é a inclinação da reta que conecta os pontos  $(x_{i-1}, f(x_{i-1}))$  e  $(x_i, f(x_i))$ :

$$\left. \frac{df}{dx} \right|_{x=x_i} = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \quad (6.6)$$

- A **diferença central** é a inclinação da reta que conecta os pontos  $(x_{i-1}, f(x_{i-1}))$  e  $(x_{i+1}, f(x_{i+1}))$ :

$$\left. \frac{df}{dx} \right|_{x=x_i} = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} \quad (6.7)$$



**Figura 6-5** Aproximação da derivada por diferenças finitas.

Os primeiros dois exemplos mostram aplicações das fórmulas de diferenças finitas progressiva, regressiva e central. O Exemplo 6-1 compara a diferenciação numérica com a diferenciação analítica, e, no Exemplo 6-2, as fórmulas são usadas para realizar a diferenciação de dados discretos.

### Exemplo 6-1: Comparação entre as diferenciações numérica e analítica

Considere a função  $f(x) = x^3$ . Calcule numericamente a derivada primeira no ponto  $x = 3$  aplicando as fórmulas de diferenças finitas progressiva, regressiva e central, usando:

- os pontos  $x = 2, x = 3$  e  $x = 4$ .
- os pontos  $x = 2,75, x = 3$  e  $x = 3,25$ .

Compare os resultados com a derivada exata (analítica).

### SOLUÇÃO

**Diferenciação analítica:** A derivada da função é  $f'(x) = 3x^2$ , e o valor da derivada em  $x = 3$  é  $f'(3) = 3 \cdot 3^2 = 27$ .

#### Diferenciação numérica:

- Os pontos usados na diferenciação numérica são:

$x:$	2	3	4
$f(x):$	8	27	64

Usando as Eqs. (6.5) a (6.7), as derivadas usando as fórmulas de diferenças finitas progressiva, regressiva e central são:

#### Diferença finita progressiva:

$$\frac{df}{dx} \Big|_{x=3} = \frac{f(4) - f(3)}{4 - 3} = \frac{64 - 27}{1} = 37 \quad \text{erro} = \left| \frac{37 - 27}{27} \cdot 100 \right| = 37,04\%$$

#### Diferença finita regressiva:

$$\frac{df}{dx} \Big|_{x=3} = \frac{f(3) - f(2)}{3 - 2} = \frac{27 - 8}{1} = 19 \quad \text{erro} = \left| \frac{19 - 27}{27} \cdot 100 \right| = 29,63\%$$

#### Diferença finita central:

$$\frac{df}{dx} \Big|_{x=3} = \frac{f(4) - f(2)}{4 - 2} = \frac{64 - 8}{1} = 28 \quad \text{erro} = \left| \frac{28 - 27}{27} \cdot 100 \right| = 3,704\%$$

- Os pontos usados na diferenciação numérica são:

$x:$	2,75	3	3,25
$f(x)$	$2,75^3$	$3^3$	$3,25^3$

Usando as Eqs. (6.5) a (6.7), as derivadas obtidas com as fórmulas de diferenças finitas progressiva, regressiva e central são:

#### Diferença finita progressiva:

$$\frac{df}{dx} \Big|_{x=3} = \frac{f(3,25) - f(3)}{3,25 - 3} = \frac{3,25^3 - 27}{0,25} = 29,3125 \quad \text{erro} = \left| \frac{29,3125 - 27}{27} \cdot 100 \right| = 8,565\%$$

#### Diferença finita regressiva:

$$\frac{df}{dx} \Big|_{x=3} = \frac{f(3) - f(2,75)}{3 - 2,75} = \frac{27 - 2,75^3}{3 - 2,75} = 24,8125 \quad \text{erro} = \left| \frac{24,8125 - 27}{27} \cdot 100 \right| = 8,102\%$$

#### Diferença finita central:

$$\frac{df}{dx} \Big|_{x=3} = \frac{f(3,25) - f(2,75)}{3,25 - 2,75} = \frac{3,25^3 - 2,75^3}{3,25 - 2,75} = 27,0625 \quad \text{erro} = \left| \frac{27,0625 - 27}{27} \cdot 100 \right| = 0,2315\%$$

Os resultados mostram que a fórmula de diferença finita central fornece uma aproximação mais precisa. Isso é discutido em maiores detalhes na próxima seção. Além disso, uma menor separação entre os pontos resulta em uma aproximação significativamente mais precisa.

### Exemplo 6-2: Vibrações amortecidas

Em um experimento de vibração, um bloco de massa  $m$  é preso a uma mola com dureza  $k$  e a um amortecedor com coeficiente de amortecimento  $c$ , conforme mostrado na figura. Para que o experimento tenha início, o bloco é retirado da posição de equilíbrio e solto. A posição do bloco em função do tempo é gravada em uma freqüência de 5 Hz (5 vezes por segundo). Os dados gravados nos primeiros 10 s são mostrados na figura. Os dados no intervalo  $4 \leq t \leq 8$  s são dados na tabela a seguir.

- (a) A velocidade do bloco é a derivada da posição em relação ao tempo. Use a fórmula de diferença finita central para calcular a velocidade nos tempos  $t = 5$  s e  $t = 6$  s.
- (b) Escreva uma função no MATLAB que calcule a derivada de uma função descrita por um conjunto de pontos. Chame a função de `dx=derivada(x,y)`, onde  $x$  e  $y$  são vetores com as coordenadas dos pontos, e  $dx$  é um vetor com a derivada  $dy/dx$  em cada ponto. A função deve calcular a derivada no **primeiro** e no **último** ponto usando as fórmulas de **diferenças finitas progressiva e regressiva**, respectivamente, e usando a fórmula de diferença finita central nos demais pontos.

Use os pontos fornecidos para calcular a velocidade do bloco em  $4 \leq t \leq 8$  s. Calcule a aceleração do bloco a partir da diferenciação da velocidade. Trace um gráfico contendo deslocamento, velocidade e aceleração *versus* tempo para  $4 \leq t \leq 8$  s.

$t(s)$	4,0	4,2	4,4	4,6	4,8	5,0	5,2	5,4	5,6	5,8	6,0
$x(\text{cm})$	-5,87	-4,23	-2,55	-0,89	0,67	2,09	3,31	4,31	5,06	5,55	5,78
$t(s)$	6,2	6,4	6,6	6,8	7,0	7,2	7,4	7,6	7,8	8,0	
$x(\text{cm})$	5,77	5,52	5,08	4,46	3,72	2,88	2,00	1,10	0,23	-0,59	

### SOLUÇÃO

- (a) A velocidade é calculada usando a Eq. (6.7):

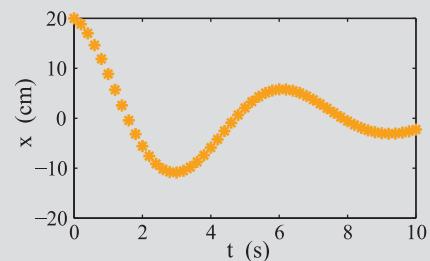
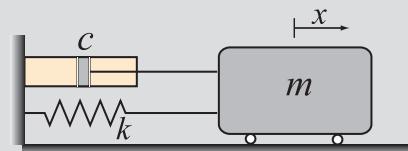
$$\text{para } t = 5 \text{ s:} \quad \frac{dx}{dt} \Big|_{x=5} = \frac{f(5,2) - f(4,8)}{5,2 - 4,8} = \frac{3,31 - 0,67}{0,4} = 6,6 \text{ cm/s}$$

$$\text{para } t = 6 \text{ s:} \quad \frac{dx}{dt} \Big|_{x=5} = \frac{f(6,2) - f(5,8)}{6,2 - 5,8} = \frac{5,77 - 5,55}{0,4} = 0,55 \text{ cm/s}$$

- (b) A função `dx=derivada(x,y)` listada a seguir calcula a derivada de uma função descrita por um conjunto discreto de pontos.

#### Programa 6-1 Arquivo de função. Derivada de uma função descrita por pontos.

```
function dx = derivada(x,y)
% Esta função calcula a derivada de uma função dada por um conjunto de
% pontos. A derivada no primeiro e no último ponto é calculada usando
% as fórmulas de diferenças finitas progressiva e regressiva respectivamente
% A derivada em todos os demais pontos é calculada pela
```

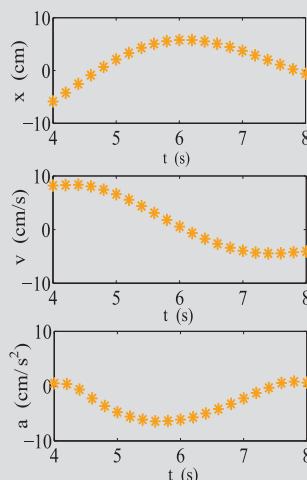


```
% fórmula de diferença finita central.
% Variáveis de entrada:
% x Vetor com as coordenadas x dos pontos.
% y Vetor com as coordenadas y dos pontos.
% Variável de saída:
% dx Vetor com o valor da derivada em cada ponto.
n = length(x);
dx(1) = (y(2) - y(1))/(x(2) - x(1));
for i = 2:n - 1
    dx(i) = (y(i + 1) - y(i - 1))/(x(i + 1) - x(i - 1));
end
dx(n) = (y(n) - y(n - 1))/(x(n) - x(n - 1));
```

A função derivada é usada no seguinte programa, escrito em arquivo texto. O programa determina a velocidade (a derivada do conjunto de dados fornecido) e a aceleração (a derivada da velocidade), gerando em seguida três gráficos.

```
t = 4:0.2:8;
x = [-5.87 -4.23 -2.55 -0.89 0.67 2.09 3.31 4.31 5.06 5.55 5.78 5.77 5.52 5.08 4.46 3.72 2.88 2.00 1.10
0.23 -0.59];
vel = derivada(t,x);
acc = derivada(t,vel);
subplot (3,1,1)
plot(t,x)
subplot (3,1,2)
plot(t,vel)
subplot (3,1,3)
plot(t,acc)
```

Quando o programa é executado, os seguintes gráficos são exibidos (os gráficos foram formatados na janela da figura).



### 6.3 FÓRMULAS DE DIFERENÇAS FINITAS USANDO A EXPANSÃO EM SÉRIE DE TAYLOR

As fórmulas de diferenças finitas progressiva, regressiva e central, bem como muitas outras fórmulas usadas para calcular derivadas de forma aproximada, podem ser deduzidas a partir da expansão em série de Taylor. Essas fórmulas fornecem uma estimativa da derivada em um ponto usando valores de pontos em sua vizinhança. O número de pontos usados nos cálculos varia com a fórmula, e os pontos podem estar à frente, atrás ou em ambos os lados do ponto onde se calcula a derivada. Uma vantagem do uso da expansão em série de Taylor na dedução das fórmulas está no fato de ela também fornecer uma estimativa do erro de truncamento presente na aproximação.

Nesta seção, são deduzidas muitas fórmulas de diferenças finitas. Embora essas fórmulas possam ser deduzidas para pontos não uniformemente distribuídos, a dedução aqui apresentada se restringe a pontos igualmente espaçados. A Seção 6.3.1 apresenta fórmulas para o cálculo aproximado da derivada primeira, e a Seção 6.3.2 lida com fórmulas de diferenças finitas usadas no cálculo da derivada segunda. Os métodos aqui utilizados para deduzir as fórmulas também podem ser usados na obtenção de fórmulas de diferenças finitas que aproximem derivadas de ordem mais elevada. Um resumo de fórmulas de diferenças finitas usadas para avaliar derivadas até a derivada quarta é apresentado na Seção 6.4.

#### 6.3.1 Fórmulas de diferenças finitas para a derivada primeira

Várias fórmulas usadas na aproximação da derivada primeira no ponto  $x_i$  com base nos valores de pontos próximos a  $x_i$  são deduzidas usando a expansão em série de Taylor. Todas as fórmulas deduzidas nesta seção se referem ao caso no qual os pontos estão igualmente espaçados.

##### Fórmula de diferença finita progressiva com dois pontos para a derivada primeira

O valor da função no ponto  $x_{i+1}$  pode ser aproximado usando a série de Taylor em termos do valor da função e de suas derivadas no ponto  $x_i$ :

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(x_i)}{4!}h^4 + \dots \quad (6.8)$$

onde  $h = x_{i+1} - x_i$  é o espaçamento entre os pontos. Usando a expansão da série de Taylor com dois termos e um resíduo (ver Capítulo 2), a Eq. (6.8) pode ser rescrita como:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(\xi)}{2!}h^2 \quad (6.9)$$

onde  $\xi$  é um valor de  $x$  entre  $x_i$  e  $x_{i+1}$ .

Resolvendo a Eq. (6.9) para  $f'(x_i)$ , obtém-se:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(\xi)}{2!}h \quad (6.10)$$

Um valor aproximado pode agora ser calculado para a derivada  $f'(x_i)$  se o segundo termo no lado direito da Eq. (6.10) for ignorado, o que introduz um erro de truncamento (discretização). Como esse termo é proporcional a  $h$ , diz-se que o erro de truncamento é da ordem de  $h$  (escrito como  $O(h)$ ):

$$\text{erro de truncamento} = \frac{f''(\xi)}{2!}h = O(h) \quad (6.11)$$

Deve-se frisar aqui que a ordem de grandeza do erro de truncamento não é de fato conhecida, já que não se sabe o valor de  $f''(\xi)$ . Entretanto, a Eq. (6.11) é valiosa por sugerir que um menor  $h$  resulta em um erro menor. Além disso, conforme mostrado mais adiante neste capítulo, ela possibilita a comparação da ordem de grandeza do erro presente em diferentes fórmulas de diferenças finitas.

Usando a notação da Eq. (6.11), o valor aproximado da derivada primeira é:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h) \quad (6.12)$$

A aproximação da Eq. (6.12) é igual à fórmula de diferença finita progressiva apresentada na Eq. (6.5).

### Fórmula de diferença finita regressiva com dois pontos para a derivada primeira

A fórmula de diferença finita regressiva também pode ser deduzida com a aplicação da expansão em série de Taylor. O valor da função no ponto  $x_{i-1}$  pode ser aproximado usando a série de Taylor em termos do valor da função e de suas derivadas no ponto  $x_i$ :

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(x_i)}{4!}h^4 + \dots \quad (6.13)$$

onde  $h = x_i - x_{i-1}$  é o espaçamento entre os pontos. Usando a expansão da série de Taylor com dois termos e um resíduo (ver Capítulo 2), a Eq. (6.13) pode ser rescrita como:

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(\xi)}{2!}h^2 \quad (6.14)$$

onde  $\xi$  é um valor de  $x$  entre  $x_{i-1}$  e  $x_i$ . Resolvendo a Eq. (6.14) para  $f'(x_i)$ , obtém-se:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + \frac{f''(\xi)}{2!}h \quad (6.15)$$

Um valor aproximado pode agora ser calculado para a derivada  $f'(x_i)$  se o segundo termo no lado direito da Eq. (6.15) for ignorado. Isso resulta em:

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + O(h) \quad (6.16)$$

A aproximação da Eq. (6.16) é igual à fórmula de diferença finita regressiva apresentada na Eq. (6.6).

### Fórmula de diferença finita central com dois pontos para a derivada primeira

A fórmula de diferença finita central também pode ser deduzida usando três termos na série de Taylor e um resíduo. O valor da função no ponto  $x_{i+1}$  em termos do valor da função e de suas derivadas no ponto  $x_i$  é dado por:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\xi_1)}{3!}h^3 \quad (6.17)$$

onde  $\xi_1$  é um valor de  $x$  entre  $x_i$  e  $x_{i+1}$ . O valor da função no ponto  $x_{i-1}$  em termos do valor da função e de suas derivadas no ponto  $x_i$  é dado por:

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(\xi_2)}{3!}h^3 \quad (6.18)$$

onde  $\xi_2$  é um valor de  $x$  entre  $x_{i-1}$  e  $x_i$ . Nas duas últimas equações, o espaçamento dos intervalos é igual, de forma que  $h = x_{i+1} - x_i = x_i - x_{i-1}$ . Subtraindo a Eq. (6.18) da Eq. (6.17), obtém-se:

$$f(x_{i+1}) - f(x_{i-1}) = 2f'(x_i)h + \frac{f'''(\xi_1)}{3!}h^3 + \frac{f'''(\xi_2)}{3!}h^3 \quad (6.19)$$

Uma estimativa para a derivada primeira é obtida resolvendo-se a Eq. (6.19) para  $f'(x_i)$  sem considerar os resíduos, o que introduz um erro de truncamento da ordem de  $h^2$ :

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2) \quad (6.20)$$

A aproximação na Eq. (6.20) é igual à fórmula de diferença central na Eq. (6.7) para intervalos igualmente espaçados. Uma comparação entre as Eqs. (6.12), (6.16) e (6.20) mostra que, nas diferenças finitas progressiva e regressiva, o erro de truncamento é da ordem de  $h$ , enquanto na aproximação por diferença central o erro de truncamento é da ordem de  $h^2$ . Isso indica que a aproximação por diferença central fornece uma aproximação mais precisa para a derivada. Isso pode ser observado esquematicamente na Fig. 6-5, onde a inclinação da reta que representa a derivada na aproximação pela diferença central parece estar mais próxima da inclinação da reta tangente do que as retas referentes às aproximações progressiva e regressiva.

### Fórmulas de diferenças finitas progressiva e regressiva com três pontos para a derivada primeira

As fórmulas de diferenças finitas progressiva e regressiva dadas nas Eqs. (6.12) e (6.16) fornecem uma estimativa para a derivada primeira com um erro de truncamento de  $O(h)$ . A fórmula de diferença progressiva avalia a derivada no ponto  $x_i$  com base nos valores nesse ponto e no ponto imediatamente à sua direita,  $x_{i+1}$ . A fórmula de diferença finita regressiva avalia a derivada no ponto  $x_i$  com base nos valores nesse ponto e naquele imediatamente à sua esquerda,  $x_{i-1}$ . Claramente, a fórmula de diferença finita progressiva pode ser usada na avaliação da derivada primeira do primeiro ponto  $x_1$  e em todos os pontos internos, enquanto a fórmula de diferença finita regressiva pode ser usada na avaliação da derivada primeira no

último ponto e em todos os pontos internos. A fórmula de diferença central, dada pela Eq. (6.20), fornece uma estimativa para a derivada primeira com um erro de  $O(h^2)$ . A fórmula de diferença central avalia a derivada primeira em um dado ponto  $x_i$  usando os pontos  $x_{i-1}$  e  $x_{i+1}$ . Conseqüentemente, para uma função dada por um conjunto discreto de  $n$  pontos, a fórmula da diferença central pode ser usada apenas nos pontos internos e não nos ponto finais ( $x_1$  ou  $x_n$ ). Uma estimativa para a derivada primeira nos pontos finais, com erro de  $O(h^2)$ , pode ser calculada com fórmulas de diferenças finitas progressiva e regressiva com três pontos, deduzidas a seguir.

A fórmula de **diferença finita progressiva com três pontos** calcula a derivada no ponto  $x_i$  usando o valor da função nesse ponto e nos dois pontos seguintes,  $x_{i+1}$  e  $x_{i+2}$ . Assume-se que os pontos estejam igualmente espaçados, logo  $h = x_{i+2} - x_{i+1} = x_{i+1} - x_i$  (o procedimento também pode ser aplicado em pontos não-uniformemente espaçados). A dedução da fórmula começa usando-se três termos da expansão em série de Taylor com um resíduo para escrever o valor da função nos pontos  $x_{i+1}$  e  $x_{i+2}$  em termos do valor da função e de suas derivadas no ponto  $x_i$ :

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(\xi_1)}{3!}h^3 \quad (6.21)$$

$$f(x_{i+2}) = f(x_i) + f'(x_i)2h + \frac{f''(x_i)}{2!}(2h)^2 + \frac{f'''(\xi_2)}{3!}(2h)^3 \quad (6.22)$$

onde  $\xi_1$  é um valor de  $x$  entre  $x_i$  e  $x_{i+1}$ , e  $\xi_2$  é um valor de  $x$  entre  $x_i$  e  $x_{i+2}$ . As Eqs. (6.21) e (6.22) são em seguida combinadas de tal forma que os termos com a derivada segunda desapareçam. Isso é feito multiplicando a Eq. (6.21) por 4 e subtraindo a Eq. (6.22):

$$4f(x_{i+1}) - f(x_{i+2}) = 3f(x_i) + 2f'(x_i)h + \frac{4f'''(\xi_1)}{3!}h^3 - \frac{f'''(\xi_2)}{3!}(2h)^3 \quad (6.23)$$

Uma estimativa para a derivada primeira é obtida resolvendo a Eq. (6.23) para  $f'(x_i)$  sem considerar os resíduos, o que introduz um erro de truncamento da ordem de  $h^2$ :

$$f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h} + O(h^2) \quad (6.24)$$

A Eq. (6.24) é a fórmula de diferença finita progressiva com três pontos que estima a derivada primeira no ponto  $x_i$  usando o valor da função nesse ponto e nos dois pontos seguintes,  $x_{i+1}$  e  $x_{i+2}$ , com um erro de  $O(h^2)$ . Essa fórmula pode ser usada para calcular a derivada do primeiro ponto de uma função descrita por um conjunto discreto de  $n$  pontos.

A fórmula de **diferença finita regressiva com três pontos** calcula a derivada no ponto  $x_i$  usando o valor desse ponto e dos dois pontos anteriores,  $x_{i-1}$  e  $x_{i-2}$ . A fórmula é deduzida da mesma forma que a Eq. (6.24). A expansão em série de Taylor com três termos e um resíduo é escrita para o valor da função nos pontos  $x_{i-1}$  e  $x_{i-2}$  em termos do valor da função e de suas derivadas no ponto  $x_i$ . As equações são então manipuladas para se obter uma equação sem os termos das derivadas segundas, que é então resolvida para  $f'(x_i)$ . A fórmula obtida é:

$$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h} + O(h^2) \quad (6.25)$$

onde  $h = x_i - x_{i-1} = x_{i-1} - x_{i-2}$  é a distância entre os pontos.

O Exemplo 6-3 mostra a aplicação da fórmula de diferença finita progressiva com três pontos no cálculo da derivada primeira.

### Exemplo 6-3: Comparação entre as diferenciações numérica e analítica

Considere a função  $f(x) = x^3$ . Calcule numericamente a derivada primeira no ponto  $x = 3$  aplicando as fórmulas de diferenças finitas progressiva, regressiva e central, usando:

- (a) os pontos  $x = 3, x = 4$  e  $x = 5$ .
- (b) os pontos  $x = 3, x = 3,25$  e  $x = 3,5$ .

Compare os resultados com o valor exato da derivada, obtido analiticamente.

#### SOLUÇÃO

**Diferenciação analítica:** A derivada da função é  $f'(x) = 3x^2$ , e o valor da derivada em  $x = 3$  é  $f'(3) = 3 \cdot 3^2 = 27$ .

#### Diferenciação numérica:

- (a) Os pontos usados na diferenciação numérica são:

$x:$	3	4	5
$f(x):$	27	64	125

Usando a Eq. (6.24), a derivada usando a fórmula de diferença finita progressiva com três pontos é:

$$f'(3) = \frac{-3f(3) + 4f(4) - f(5)}{2 \cdot 1} = \frac{-3 \cdot 27 + 4 \cdot 64 - 125}{2} = 25 \quad \text{erro} = \left| \frac{25 - 27}{27} \right| \cdot 100 = 7,4\%$$

- (b) Os pontos usados na diferenciação numérica são:

$x:$	3	3,25	3,5
$f(x)$	27	$3,25^3$	$3,5^3$

Usando a Eq. (6.24), a derivada obtida com a fórmula de diferença finita progressiva com três pontos é:

$$f'(3) = \frac{-3f(3) + 4f(3,25) - f(3,5)}{2 \cdot 0,25} = \frac{-3 \cdot 27 + 4 \cdot 3,25^3 - 3,5^3}{0,5} = 26,875$$

$$\text{erro} = \left| \frac{26,875 - 27}{27} \right| \cdot 100 = 0,46\%$$

Os resultados mostram que a fórmula de diferença finita progressiva com três pontos fornece uma derivada primeira muito mais precisa do que a fórmula de diferença finita progressiva com dois pontos usada no Exemplo 6-1. Para  $h = 1$ , o erro cai de 37,04% para 7,4%, e, para  $h = 0,25$ , o erro cai de 8,57% para 0,46%.

### 6.3.2 Fórmulas de diferenças finitas para a derivada segunda

A mesma abordagem usada na Seção 6.3.1 para desenvolver fórmulas de diferenças finitas para a derivada primeira pode ser usada para desenvolver expressões para derivadas de ordem superior. Nesta seção, expressões baseadas em diferen-

ças centrais, em diferenças progressivas unilaterais e em diferenças regressivas unilaterais são apresentadas para o cálculo da derivada segunda no ponto  $x_i$ .

### Fórmula de diferença central com três pontos para a derivada segunda

Fórmulas de diferença central podem ser desenvolvidas para a derivada segunda usando qualquer número de pontos em cada lado de  $x_i$ , no qual se deseja avaliar a derivada. As fórmulas são deduzidas escrevendo-se a expansão em série de Taylor com pontos em ambos os lados de  $x_i$  em termos do valor da função e de suas derivadas nesse ponto. Em seguida, as equações são combinadas de tal forma que os termos contendo as derivadas primeiras sejam eliminados. Por exemplo, para pontos  $x_{i+1}$  e  $x_{i-1}$ , a expansão em série de Taylor com um resíduo é:

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(\xi_1)}{4!}h^4 \quad (6.26)$$

$$f(x_{i-1}) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(\xi_2)}{4!}h^4 \quad (6.27)$$

onde  $\xi_1$  é um valor de  $x$  entre  $x_i$  e  $x_{i+1}$  e  $\xi_2$  é um valor de  $x$  entre  $x_i$  e  $x_{i-1}$ . Somando as Eqs. (6.26) e (6.27), obtém-se:

$$f(x_{i+1}) + f(x_{i-1}) = 2f(x_i) + 2\frac{f''(x_i)}{2!}h^2 + \frac{f^{(4)}(\xi_1)}{4!}h^4 + \frac{f^{(4)}(\xi_2)}{4!}h^4 \quad (6.28)$$

Pode-se obter uma estimativa para a derivada segunda resolvendo a Eq. (6.28) para  $f''(x_i)$  sem considerar os resíduos. Isso introduz um erro de truncamento da ordem de  $h^2$ .

$$f''(x_i) = \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1})}{h^2} + O(h^2) \quad (6.29)$$

A Eq. (6.29) é a fórmula de diferença central com três pontos que estima a derivada segunda no ponto  $x_i$  usando o valor da função nesse ponto, no ponto anterior,  $x_{i-1}$ , e no ponto seguinte,  $x_{i+1}$ , com um erro de truncamento de  $O(h^2)$ .

O mesmo procedimento pode ser usado para desenvolver uma fórmula de ordem superior (quarta ordem) mais precisa, envolvendo os pontos  $x_{i-2}$ ,  $x_{i-1}$ ,  $x_i$ ,  $x_{i+1}$ ,  $x_{i+2}$ :

$$f''(x_i) = \frac{-f(x_{i-2}) + 16f(x_{i-1}) - 30f(x_i) + 16f(x_{i+1}) - f(x_{i+2})}{12h^2} + O(h^4) \quad (6.30)$$

### Fórmulas de diferenças progressiva e regressiva com três pontos para a derivada segunda

A fórmula de *diferença progressiva com três pontos* que estima a derivada segunda no ponto  $x_i$  usando o valor da função nesse ponto e nos dois pontos seguintes,  $x_{i+1}$  e  $x_{i+2}$ , é desenvolvida multiplicando-se a Eq. (6.21) por 2 e subtraindo-a da Eq. (6.22). A equação resultante é então resolvida para  $f''(x_i)$ :

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2})}{h^2} + O(h) \quad (6.31)$$

A fórmula de **diferença regressiva com três pontos** que estima a derivada segunda no ponto  $x_i$  usando o valor da função nesse ponto e nos dois pontos anteriores,  $x_{i-1}$  e  $x_{i-2}$ , é deduzida de forma similar. Isto é feito escrevendo-se a expansão em série de Taylor com três termos e um resíduo para o valor da função nos pontos  $x_{i-1}$  e  $x_{i-2}$  em termos do valor da função e de suas derivadas no ponto  $x_i$ . As equações são em seguida manipuladas para se obter uma equação sem os termos que incluem a derivada primeira, que é então resolvida para  $f''(x_i)$ . A fórmula resultante é:

$$f''(x_i) = \frac{f(x_{i-2}) - 2f(x_{i-1}) + f(x_i)}{h^2} + O(h) \quad (6.32)$$

Fórmulas de ordem mais elevada podem ser deduzidas com o emprego dos mesmos métodos usados aqui para a derivada segunda. Uma lista com tais fórmulas é dada na próxima seção. O Exemplo 6-3 mostra a aplicação da fórmula de diferença finita com três pontos no cálculo da derivada segunda.

#### Exemplo 6-4: Comparação entre as diferenciações numérica e analítica

Considere a função  $f(x) = \frac{2^x}{x}$ . Calcule numericamente a derivada segunda no ponto  $x = 2$  aplicando a fórmula de diferença central com três pontos, usando:

- (a) os pontos  $x = 1,8, x = 2$  e  $x = 2,2$ .
- (b) os pontos  $x = 1,9, x = 2$  e  $x = 2,1$ .

Compare os resultados com a derivada exata (analítica).

#### SOLUÇÃO

**Diferenciação analítica:** A derivada segunda da função  $f(x) = \frac{2^x}{x}$  é:

$$f''(x) = \frac{2^x [\ln(2)]^2}{x} - \frac{2 \cdot 2^x \ln(2)}{x^2} + \frac{2 \cdot 2^x}{x^3}$$

e o valor da derivada em  $x = 2$  é  $f''(2) = 0,5746$ .

#### Diferenciação numérica

- (a) A diferenciação numérica é feita com a substituição dos valores dos pontos  $x = 1,8, x = 2$  e  $x = 2,2$  na Eq. (6.29). As operações são feitas na janela de comandos do MATLAB:

```
>> xa=[1.8 2 2.2];
>> ya=2.^xa./xa;
>> df=(ya(1)-2*ya(2)+ya(3))/0.2^2
df =
0.57748177389232
```

- (b) A diferenciação numérica é feita com a substituição dos valores dos pontos  $x = 1,9, x = 2$  e  $x = 2,1$  na Eq. (6.29). As operações são feitas na janela de comandos do MATLAB:

```
>> xb=[1.9 2 2.1];
>> yb=2.^xb./xb;
>> dfb=(yb(1)-2*yb(2)+yb(3))/0.1^2
dfb =
0.57532441566441
```

$$\text{Erro na letra (a): } erro = \frac{0,577482 - 0,5746}{0,5746} \cdot 100 = 0,5016\%$$

$$\text{Erro na letra (b): } erro = \frac{0,575324 - 0,5746}{0,5746} \cdot 100 = 0,126\%$$

Os resultados mostram que a fórmula da derivada central com três pontos fornece uma aproximação bastante precisa para o valor da derivada segunda.

## 6.4 RESUMO DE FÓRMULAS DE DIFERENÇAS FINITAS PARA A DIFERENCIADAÇÃO NUMÉRICA

A Tabela 6-1 lista fórmulas de diferenças finitas com precisão variável que podem ser usadas na avaliação numérica das derivadas primeira, segunda, terceira e quarta. As fórmulas podem ser utilizadas quando a função a ser diferenciada é especificada como um conjunto de pontos discretos com variável independente uniformemente espaçada.

**Tabela 6-1** Fórmulas de diferenças finitas

<i>Derivada primeira</i>		
<b>Método</b>	<b>Fórmula</b>	<b>Erro de truncamento</b>
Diferença progressiva com dois pontos	$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
Diferença progressiva com três pontos	$f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h}$	$O(h^2)$
Diferença regressiva com dois pontos	$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
Diferença regressiva com três pontos	$f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h}$	$O(h^2)$
Diferença central com dois pontos	$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
Diferença central com quatro pontos	$f'(x_i) = \frac{f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})}{12h}$	$O(h^4)$
<i>Derivada segunda</i>		
<b>Método</b>	<b>Fórmula</b>	<b>Erro de truncamento</b>
Diferença progressiva com três pontos	$f''(x_i) = \frac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2})}{h^2}$	$O(h)$

**Tabela 6-1** Fórmulas de diferenças finitas (*continuação*)

Diferença progressiva com quatro pontos	$f''(x_i) = \frac{2f(x_i) - 5f(x_{i+1}) + 4f(x_{i+2}) - f(x_{i+3})}{h^2}$	$O(h^2)$
Diferença regressiva com três pontos	$f''(x_i) = \frac{f(x_{i-2}) - 2f(x_{i-1}) + f(x_i)}{h^2}$	$O(h)$
Diferença regressiva com quatro pontos	$f''(x_i) = \frac{-f(x_{i-3}) + 4f(x_{i-2}) - 5f(x_{i-1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Diferença central com três pontos	$f''(x_i) = \frac{f(x_{i-1}) - 2f(x_i) + f(x_{i+1})}{h^2}$	$O(h^2)$
Diferença central com cinco pontos	$f''(x_i) = \frac{-f(x_{i-2}) + 16f(x_{i-1}) - 30f(x_i) + 16f(x_{i+1}) - f(x_{i+2})}{12h^2}$	$O(h^4)$

**Derivada terceira**

Método	Fórmula	Erro de truncamento
Diferença progressiva com quatro pontos	$f'''(x_i) = \frac{-f(x_i) + 3f(x_{i+1}) - 3f(x_{i+2}) + f(x_{i+3})}{h^3}$	$O(h)$
Diferença progressiva com cinco pontos	$f'''(x_i) = \frac{-5f(x_i) + 18f(x_{i+1}) - 24f(x_{i+2}) + 14f(x_{i+3}) - 3f(x_{i+4})}{2h^3}$	$O(h^2)$
Diferença regressiva com quatro pontos	$f'''(x_i) = \frac{-f(x_{i-3}) + 3f(x_{i-2}) - 3f(x_{i-1}) + f(x_i)}{h^3}$	$O(h)$
Diferença regressiva com cinco pontos	$f'''(x_i) = \frac{3f(x_{i-4}) - 14f(x_{i-3}) + 24f(x_{i-2}) - 18f(x_{i-1}) + 5f(x_i)}{2h^3}$	$O(h^2)$
Diferença central com quatro pontos	$f'''(x_i) = \frac{-f(x_{i-2}) + 2f(x_{i-1}) - 2f(x_{i+1}) + f(x_{i+2})}{2h^3}$	$O(h^2)$
Diferença central com seis pontos	$f'''(x_i) = \frac{f(x_{i-3}) - 8f(x_{i-2}) + 13f(x_{i-1}) - 13f(x_{i+1}) + 8f(x_{i+2}) - f(x_{i+3})}{8h^3}$	$O(h^4)$

**Derivada quarta**

Método	Fórmula	Erro de truncamento
Diferença progressiva com cinco pontos	$f^{iv}(x_i) = \frac{f(x_i) - 4f(x_{i+1}) + 6f(x_{i+2}) - 4f(x_{i+3}) + f(x_{i+4})}{h^4}$	$O(h)$
Diferença progressiva com seis pontos	$f^{iv}(x_i) = \frac{3f(x_i) - 14f(x_{i+1}) + 26f(x_{i+2}) - 24f(x_{i+3}) + 11f(x_{i+4}) - 2f(x_{i+5})}{h^4}$	$O(h^2)$
Diferença regressiva com cinco pontos	$f^{iv}(x_i) = \frac{f(x_{i-4}) - 4f(x_{i-3}) + 6f(x_{i-2}) - 4f(x_{i-1}) + f(x_i)}{h^4}$	$O(h)$
Diferença regressiva com seis pontos	$f^{iv}(x_i) = \frac{-2f(x_{i-5}) + 11f(x_{i-4}) - 24f(x_{i-3}) + 26f(x_{i-2}) - 14f(x_{i-1}) + 3f(x_i)}{h^4}$	$O(h^2)$

**Tabela 6-1** Fórmulas de diferenças finitas (*continuação*)

Diferença central com cinco pontos	$f^{(iv)}(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 6f(x_i) - 4f(x_{i+1}) + f(x_{i+2})}{h^4}$	$O(h^2)$
Diferença central com sete pontos	$f^{(iv)}(x_i) = \frac{f(x_{i-3}) + 12f(x_{i-2}) - 39f(x_{i-1}) + 56f(x_i) + 39f(x_{i+1}) + 12f(x_{i+2}) - f(x_{i+3})}{6h^4}$	$O(h^4)$

## 6.5 FÓRMULAS DE DIFERENCIAÇÃO USANDO POLINÔMIOS DE LAGRANGE

Fórmulas de diferenciação também podem ser deduzidas usando polinômios de Lagrange. Para a derivada primeira, as fórmulas de diferenças finitas central com dois pontos, progressiva com três pontos e regressiva com três pontos são obtidas considerando três pontos,  $(x_i, y_i)$ ,  $(x_{i+1}, y_{i+1})$ , e  $(x_{i+2}, y_{i+2})$ . O polinômio na forma de Lagrange que passa por esses pontos é dado por:

$$\begin{aligned} f(x) &= \frac{(x-x_{i+1})(x-x_{i+2})}{(x_i-x_{i+1})(x_i-x_{i+2})} y_i + \frac{(x-x_i)(x-x_{i+2})}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})} y_{i+1} \\ &\quad + \frac{(x-x_i)(x-x_{i+1})}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} y_{i+2} \end{aligned} \quad (6.33)$$

Derivando a Eq. (6.33), obtém-se:

$$\begin{aligned} f'(x) &= \frac{2x-x_{i+1}-x_{i+2}}{(x_i-x_{i+1})(x_i-x_{i+2})} y_i + \frac{2x-x_i-x_{i+2}}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})} y_{i+1} \\ &\quad + \frac{2x-x_i-x_{i+1}}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} y_{i+2} \end{aligned} \quad (6.34)$$

A derivada primeira em qualquer um dos três pontos é calculada com a substituição do valor de  $x$  correspondente ( $x_i$ ,  $x_{i+1}$  ou  $x_{i+2}$ ) na Eq. (6.34). Isto resulta nas três fórmulas a seguir para o cálculo da derivada primeira em cada um dos pontos.

$$\begin{aligned} f'(x_i) &= \frac{2x_i-x_{i+1}-x_{i+2}}{(x_i-x_{i+1})(x_i-x_{i+2})} y_i + \frac{x_i-x_{i+2}}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})} y_{i+1} \\ &\quad + \frac{x_i-x_{i+1}}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} y_{i+2} \end{aligned} \quad (6.35)$$

$$\begin{aligned} f'(x_{i+1}) &= \frac{x_{i+1}-x_{i+2}}{(x_i-x_{i+1})(x_i-x_{i+2})} y_i + \frac{2x_{i+1}-x_i-x_{i+2}}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})} y_{i+1} \\ &\quad + \frac{x_{i+1}-x_i}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} y_{i+2} \end{aligned} \quad (6.36)$$

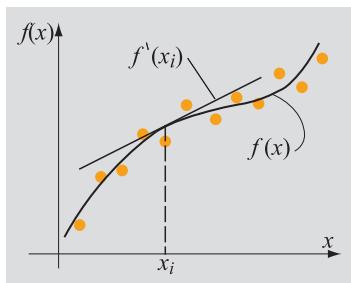
$$\begin{aligned} f'(x_{i+2}) &= \frac{x_{i+2}-x_{i+1}}{(x_i-x_{i+1})(x_i-x_{i+2})} y_i + \frac{x_{i+2}-x_i}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})} y_{i+1} \\ &\quad + \frac{2x_{i+2}-x_i-x_{i+1}}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} y_{i+2} \end{aligned} \quad (6.37)$$

Quando os pontos são igualmente espaçados, a Eq. (6.35) se reduz à formula de diferença progressiva com três pontos da Eq. (6.24), a Eq. (6.36) se reduz à formula de diferença central com dois pontos da Eq. (6.20) e a Eq. (6.37) se reduz à formula de diferença regressiva com três pontos da Eq. (6.25).

A Eq. (6.34) possui outras duas características importantes. Ela pode ser usada quando os pontos não estão uniformemente espaçados, e também para calcular a derivada primeira de qualquer ponto entre  $x_i$  e  $x_{i+2}$ .

Fórmulas de diferença usando mais pontos e/ou aplicáveis a derivadas de ordem mais elevada também podem ser deduzidas usando polinômios de Lagrange. O uso desses polinômios na dedução de fórmulas de diferenças finitas é às vezes mais fácil do que o uso da série de Taylor. Entretanto, a série de Taylor fornece uma estimativa do erro de truncamento.

## 6.6 DIFERENCIAÇÃO USANDO O AJUSTE DE CURVAS



**Figura 6-6** Diferenciação numérica usando o ajuste de curvas.

Uma abordagem diferente para o cálculo da derivada de dados especificados por um conjunto de pontos discretos consiste em primeiro obter uma função analítica aproximada que possa ser facilmente diferenciada. A função aproximada é então diferenciada para o cálculo da derivada em qualquer ponto (Fig. 6-6). O ajuste de curvas é descrito no Capítulo 5. Para dados que apresentam uma relação não-linear, o ajuste da curva é freqüentemente feito com o uso combinado da técnica de mínimos quadrados e de uma função exponencial, uma função de potência, um polinômio de baixa ordem ou uma combinação linear de funções não-lineares cujas derivadas sejam fáceis de se calcular. Este procedimento pode ser mais adequado quando os dados apresentarem dispersão, ou ruído, já que a função ajustada suaviza o ruído presente.

## 6.7 USO DE FUNÇÕES RESIDENTES DO MATLAB PARA REALIZAR A DIFERENCIAÇÃO NUMÉRICA

Em geral, recomenda-se que as técnicas descritas neste capítulo sejam usadas para desenvolver programas que realizem a diferenciação desejada. O MATLAB não possui funções residentes que realizem a diferenciação numérica de uma função arbitrária ou de dados discretos. Há, no entanto, uma função residente denominada `diff`, que pode ser usada para realizar a diferenciação numérica, e outra função residente denominada `polyder`, que determina a derivada de um polinômio.

### O comando `diff`

A função residente `diff` calcula as diferenças entre os elementos adjacentes de um vetor. A forma mais simples do comando é:

**d = diff (x)**

d é um vetor com as diferenças entre os elementos de x:  

$$d = [(x_2 - x_1), (x_3 - x_2), \dots, (x_n - x_{n-1})]$$

x é um vetor:  

$$[x_1, x_2, x_3, x_4, \dots, x_{n-1}, x_n]$$

O vetor d tem um elemento a menos que o vetor x.

Para uma função representada por um conjunto discreto de  $n$  pontos,  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$ , ...,  $(x_n, y_n)$ , a derivada primeira dada pela fórmula de diferença progressiva com dois pontos na Eq. (6.5) pode ser calculada usando o comando `diff` se `diff(y) ./ diff(x)` for digitado. O resultado é um vetor cujos elementos são:

$$\left[ \frac{(y_2 - y_1)}{(x_2 - x_1)}, \frac{(y_3 - y_2)}{(x_3 - x_2)}, \frac{(y_4 - y_3)}{(x_4 - x_3)}, \dots, \frac{(y_n - y_{n-1})}{(x_n - x_{n-1})} \right]$$

Quando o espaçamento entre os pontos é igual, de forma que  $h = (x_2 - x_1) = (x_3 - x_2) = \dots = (x_n - x_{n-1})$ , a derivada primeira dada pela fórmula de diferença progressiva com dois pontos na Eq. (6.5) pode ser calculada usando o comando `diff` se `diff(y) / h` for digitado.

O comando `diff` tem um argumento de entrada opcional que pode ser usado para o cálculo de derivadas de ordem mais elevada.

**d = diff (x, n)**

onde n é um número inteiro que especifica o número de vezes que o comando `diff` é aplicado recursivamente. Por exemplo, `diff(x, 2)` é igual a `diff(diff(x))`. Em outras palavras, para um vetor com  $n$  elementos  $x_1, \dots, x_n$ , `diff(x)` calcula um vetor com  $n - 1$  elementos:

$$x_{i+1} - x_i \quad \text{para } i = 1, \dots, n-1 \quad (6.38)$$

e `diff(x, 2)` retorna um vetor com  $n - 2$  elementos:

$$(x_{i+2} - x_{i+1}) - (x_{i+1} - x_i) = x_i - 2x_{i+1} + x_{i+2} \quad \text{para } i = 1, \dots, n-1 \quad (6.39)$$

O lado direito da Eq. (6.39) é igual ao numerador da fórmula de diferença finita progressiva com três pontos empregada no cálculo da derivada segunda em  $x = x_1$ , dada na Eq. (6.31). Consequentemente, para uma função representada por um conjunto de pontos discretos  $(x_i, y_i)$  cuja distância  $h$  entre pontos é a mesma, uma estimativa da derivada segunda de acordo com a fórmula de diferença progressiva com três pontos pode ser calculada no MATLAB digitando-se `diff(y, 2) / h^2`.

De forma similar, `diff(y, 3)` fornece o numerador da derivada terceira usando a fórmula de diferença progressiva com quatro pontos (ver Tabela 6-1). Em geral, `diff(y, n)` fornece o numerador da fórmula de diferença referente à  $n$ -ésima derivada.

### O comando polyder

A função residente `polyder` pode calcular a derivada de um polinômio (e também do produto e do quociente de dois polinômios). A forma mais simples do comando é:

**dp = polyder(p)**

dp é um vetor com os coeficientes do polinômio que corresponde à derivada do polinômio p

p é um vetor com os coeficientes do polinômio que é diferenciado

No MATLAB, polinômios são representados por um vetor linha cujos elementos são os coeficientes do polinômio ordenados do termo de maior ordem para o termo de ordem zero. Se  $p$  é um vetor de comprimento  $n$ , então  $dp$  é um vetor de comprimento  $n - 1$ . Por exemplo, para obter a derivada do polinômio  $f(x) = 4x^3 + 5x + 7$ , defina um vetor  $p = [4 0 5 7]$  e digite  $df = \text{polyder}(p)$ . A saída será  $df = [12 0 5]$ , representando  $12x^2 + 5$ , que é a derivada de  $f(x)$ .

```
>> p = [4 0 5 7];
>> dp = polyder(p)
dp =
    12    0    5
```

O comando `polyder` pode ser útil no cálculo da derivada quando uma função representada por um conjunto de pontos discretos for aproximada por um polinômio de ajuste.

## 6.8 EXTRAPOLAÇÃO DE RICHARDSON

A extrapolação de Richardson é um método usado para calcular uma aproximação mais precisa da derivada a partir de duas aproximações menos precisas dessa mesma derivada.

Em termos gerais, considere o valor  $D$  de uma derivada (desconhecida) calculada pela fórmula de diferença:

$$D = D(h) + k_2 h^2 + k_4 h^4 \quad (6.40)$$

onde  $D(h)$  é uma função que aproxima o valor da derivada e  $k_2 h^2$  e  $k_4 h^4$  são termos de erro, nos quais os coeficientes  $k_2$  e  $k_4$  são independentes do espaçamento  $h$ . O uso da mesma fórmula para calcular o valor de  $D$ , porém com um espaçamento de  $h/2$ , resulta em:

$$D = D\left(\frac{h}{2}\right) + k_2\left(\frac{h}{2}\right)^2 + k_4\left(\frac{h}{2}\right)^4 \quad (6.41)$$

A Eq. (6.41) pode ser rescrita como (após multiplicação por 4):

$$4D = 4D\left(\frac{h}{2}\right) + k_2 h^2 + k_4 \frac{h^4}{4} \quad (6.42)$$

A subtração da Eq. (6.40) da Eq. (6.42) elimina os termos com  $h^2$  e fornece:

$$3D = 4D\left(\frac{h}{2}\right) - D(h) - k_4 \frac{3h^4}{4} \quad (6.43)$$

Resolvendo a Eq. (6.43) para  $D$ , obtém-se uma nova aproximação para a derivada:

$$D = \frac{1}{3} \left( 4D\left(\frac{h}{2}\right) - D(h) \right) - k_4 \frac{h^4}{4} \quad (6.44)$$

O erro na Eq. (6.44) é agora  $O(h^4)$ . O valor  $D$  da derivada pode então ser aproximado por:

$$D = \frac{1}{3} \left( 4D\left(\frac{h}{2}\right) - D(h) \right) + O(h^4) \quad (6.45)$$

Isso significa que um valor aproximado de  $D$  com erro de  $O(h^4)$  pode ser obtido a partir de duas aproximações de ordem inferior ( $D(h)$  e  $D(h/2)$ ) calculadas com um erro de  $O(h^2)$ . A Eq. (6.45) pode ser usada para obter uma aproximação mais precisa para qualquer fórmula que calcule a derivada com um erro de  $O(h^2)$ . Para isso, utiliza-se a fórmula no cálculo de aproximações com espaçamentos  $h$  e  $h/2$ . As duas aproximações são então substituídas na Eq. (6.45), o que resulta em uma nova estimativa com um erro de  $O(h^4)$ . O procedimento é ilustrado no Exemplo 6-5.

A Eq. (6.45) também pode ser diretamente deduzida a partir de uma fórmula de diferenças finitas específica. Por exemplo, considere a fórmula de diferença central com três pontos usada no cálculo da derivada primeira de uma função com pontos igualmente espaçados (espaçamento  $h$ ), tal que  $x_{i+1} = x_i + h$  e  $x_{i-1} = x_i - h$ . Escrevendo a expansão em série de Taylor com cinco termos e um resíduo para o valor da função no ponto  $x_{i+1}$  em termos do valor dessa função e de suas derivadas no ponto  $x_i$ , obtém-se:

$$f(x_i + h) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^{IV}(x_i)}{4!}h^4 + \frac{f^V(\xi_1)}{5!}h^5 \quad (6.46)$$

onde  $\xi_1$  é um valor de  $x$  entre  $x_i$  e  $x_i + h$ . Da mesma maneira, o valor da função no ponto  $x_{i-1}$  é expresso em termos do valor da função e de suas derivadas no ponto  $x_i$ :

$$f(x_i - h) = f(x_i) - f'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(x_i)}{3!}h^3 + \frac{f^{IV}(x_i)}{4!}h^4 - \frac{f^V(\xi_2)}{5!}h^5 \quad (6.47)$$

onde  $\xi_2$  é um valor de  $x$  entre  $x_i - h$  e  $x_i$ . Subtraindo a Eq. (6.47) da Eq. (6.46), obtém-se:

$$f(x_i + h) - f(x_i - h) = 2f'(x_i)h + 2\frac{f'''(x_i)}{3!}h^3 + \frac{f^V(\xi_1)}{5!}h^5 + \frac{f^V(\xi_2)}{5!}h^5 \quad (6.48)$$

Assumindo que a derivada quinta seja contínua no intervalo  $[x_{i-1}, x_{i+1}]$ , os dois resíduos na Eq. (6.48) podem ser combinados e escritos como  $O(h^5)$ . Então, resolvendo a Eq. (6.48) para  $f'(x_i)$ , obtém-se:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h} - \frac{f'''(x_i)}{3!}h^2 + O(h^4) \quad (6.49)$$

que é a aproximação para a derivada primeira com um espaçamento entre pontos  $h$ .

A dedução das Eqs. (6.46)-(6.49) pode ser repetida se o espaçamento entre os pontos for alterado para  $h/2$ . Neste caso, a equação da derivada é:

$$f'(x_i) = \frac{f(x_i + h/2) - f(x_i - h/2)}{2(h/2)} - \frac{f'''(x_i)}{3!} \left(\frac{h}{2}\right)^2 + O(h^4) \quad (6.50)$$

ou

$$f'(x_i) = \frac{f(x_i + h/2) - f(x_i - h/2)}{h} - \frac{f'''(x_i)}{4 \cdot 3!} h^2 + O(h^4) \quad (6.51)$$

Multiplicando a Eq. (6.51) por 4, obtém-se:

$$4f'(x_i) = 4 \frac{f(x_i + h/2) - f(x_i - h/2)}{h} - \frac{f'''(x_i)}{3!} h^2 + O(h^4) \quad (6.52)$$

Subtraindo a Eq. (6.49) da Eq. (6.52) e dividindo o resultado por 3, obtém-se uma aproximação para a derivada primeira com erro  $O(h^4)$ :

$$f'(x_i) = \frac{1}{3} \left[ 4 \frac{f(x_i + h/2) - f(x_i - h/2)}{h} - \frac{f(x_i + h) - f(x_i - h)}{2h} \right] + O(h^4) \quad (6.53)$$

Derivada primeira calculada com a fórmula de diferença central com dois pontos, Eq. (6.20), com erro de  $O(h^2)$  para pontos com espaçamento  $h/2$ .

Derivada primeira calculada com a fórmula de diferença central com dois pontos, Eq. (6.20), com erro de  $O(h^2)$  para pontos com espaçamento  $h$ .

A Eq. (6.53) é um caso especial da Eq. (6.45) no qual as derivadas são calculadas com a fórmula de diferença central com dois pontos. A Eq. (6.45) pode ser usada com qualquer fórmula de diferença com erro  $O(h^2)$ .

O método da extrapolação de Richardson também pode ser usado juntamente com aproximações envolvendo erros de ordem superior. Duas aproximações com erro  $O(h^4)$  – uma calculada a partir de pontos com espaçamento  $h$  e outra a partir de pontos com espaçamento  $h/2$  – podem ser usadas no cálculo de uma aproximação mais precisa com erro  $O(h^6)$ . A fórmula para esse caso é:

$$D = \frac{1}{15} \left( 16D\left(\frac{h}{2}\right) - D(h) \right) + O(h^6) \quad (6.54)$$

A aplicação da extrapolação de Richardson é mostrada no Exemplo 6-5.

## 6.9 ERROS NA DIFERENCIACÃO NUMÉRICA

Ao longo deste capítulo, foram fornecidas expressões para o erro de truncamento, também conhecido como erro de discretização. Essas expressões foram geradas pelo esquema numérico particular usado na dedução de cada fórmula de diferenças finitas empregada no cálculo da derivada. Foi visto que, em cada caso, o erro de truncamento depende de  $h$  (o espaçamento entre os pontos) elevado a alguma potência. Claramente, reduzindo  $h$ , pode-se fazer com que o erro se torne

### Exemplo 6-5: Uso da extrapolação de Richardson na diferenciação

Use a extrapolação de Richardson nos resultados do Exemplo 6-4 para calcular uma aproximação mais precisa para a derivada da função  $f(x) = \frac{2^x}{x}$  no ponto  $x = 2$ .

Compare os resultados com a derivada exata (analítica).

### SOLUÇÃO

No Exemplo 6-4, foram calculadas duas aproximações para a derivada da função em  $x = 2$  usando a fórmula de diferença central na qual o erro é de  $O(h^2)$ . Em uma delas,  $h = 0,2$ , e na outra,  $h = 0,1$ . Os resultados do Exemplo 6-4 são:

para  $h = 0,2$ ,  $f'(2) = 0,577482$ . O erro nesta aproximação é de 0,5016%.

para  $h = 0,1$ ,  $f'(2) = 0,575324$ . O erro nesta aproximação é de 0,126%.

A extrapolação de Richardson pode ser usada com a substituição desses resultados na Eq. (6.45) (ou na Eq. (6.53)):

$$D = \frac{1}{3} \left( 4D\left(\frac{h}{2}\right) - D(h) \right) + O(h^4) = \frac{1}{3}(4 \cdot 0,575324 - 0,577481) = 0,574605$$

$$\text{Tem-se agora } \text{erro} = \frac{0,574605 - 0,5746}{0,5746} \cdot 100 = 0,00087 \%$$

Esse resultado mostra que uma aproximação muito mais precisa pode ser obtida com o uso da extrapolação de Richardson.

arbitrariamente pequeno. No entanto, quando a função a ser diferenciada é especificada como um conjunto de pontos discretos, o espaçamento é fixo e o erro de truncamento não pode ser diminuído com a redução de  $h$ . Nesse caso, um menor erro de truncamento pode ser obtido usando uma fórmula de diferenças finitas que tenha um erro de truncamento de ordem mais elevada.

Quando a função sendo diferenciada é dada por uma expressão matemática, o espaçamento  $h$  entre os pontos usados nas fórmulas de diferenças finitas é definido pelo usuário. Pode parecer que  $h$  pode ser reduzido arbitrariamente e que não há limite para quanto pequeno o erro pode ser feito. Isso, contudo, não é verdade porque o erro total é composto por duas partes. Uma delas é o erro de truncamento, que depende do método numérico empregado (a fórmula de diferenças finitas específica). A outra se refere ao erro de arredondamento, que depende da precisão finita de cada computador. Portanto, mesmo se o erro de truncamento puder ser praticamente anulado com a escolha de valores de  $h$  cada vez menores, ainda haverá o erro de arredondamento, que poderá até mesmo crescer com a redução de  $h$ . O Exemplo 6-6 ilustra esse ponto.

## 6.10 DIFERENCIAÇÃO PARCIAL NUMÉRICA

Todos os métodos de diferenciação apresentados até agora consideraram funções de apenas uma variável independente. No entanto, muitos problemas na engenharia e na ciência envolvem funções de muitas variáveis independentes, pois aplicações na vida real são bidimensionais ou tridimensionais, podendo, além disso,

### Exemplo 6-6: Comparação entre as diferenciações numérica e analítica

Considere a função  $f(x) = e^x$ . Escreva uma expressão para a derivada primeira da função em  $x = 0$  usando a fórmula de diferença central com dois pontos dada na Eq. (6.20). Investigue o efeito que o espaçamento  $h$  entre os pontos tem nos erros de truncamento e de arredondamento.

### SOLUÇÃO

A fórmula de diferença central com dois pontos dada pela Eq. (6.20) é:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - 2 \frac{f'''(\xi)}{3!} h^2$$

onde  $\xi$  é um valor de  $x$  entre  $x_{i-1}$  e  $x_{i+1}$ .

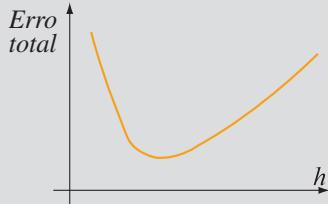
Os pontos usados para calcular a derivada de  $f(x) = e^x$  em  $x = 0$  são  $x_{i-1} = -h$  e  $x_{i+1} = h$ . Substituindo esses pontos na fórmula, obtém-se:

$$f'(0) = \frac{e^h - e^{-h}}{2h} - 2 \frac{f'''(\xi)}{3!} h^2 \quad (6.55)$$

Quando o computador calcula os valores de  $e^h$  e  $e^{-h}$ , introduz-se um erro de arredondamento porque o computador tem uma precisão finita. Consequentemente, os termos  $e^h$  e  $e^{-h}$  na Eq. (6.55) são substituídos por  $e^h + R_1$  e  $e^{-h} + R_2$ , onde  $e^h$  e  $e^{-h}$  são agora valores exatos, e  $R_1$  e  $R_2$  são os erros de arredondamento:

$$f'(0) = \frac{e^h + R_1 - e^{-h} - R_2}{2h} - 2 \frac{f'''(\xi)}{3!} h^2 = \frac{e^h - e^{-h}}{2h} + \frac{R_1 - R_2}{2h} - 2 \frac{f'''(\xi)}{3!} h^2 \quad (6.56)$$

Na Eq. (6.56), o último termo no lado direito é o erro de truncamento. Neste termo, o valor de  $f'''(\xi)$  não é conhecido, embora seja limitado. Isso significa que, à medida que  $h$  diminui, o erro também diminui. O erro de arredondamento é dado por  $(R_1 - R_2)/(2h)$ . À medida que  $h$  diminui, o erro de arredondamento aumenta. O erro total é a soma do erro de truncamento e do erro de arredondamento. Seu comportamento é mostrado esquematicamente na figura ao lado. À medida que  $h$  diminui, o erro total inicialmente diminui, mas, após um certo valor (que depende da precisão do computador utilizado), ele passa a crescer.



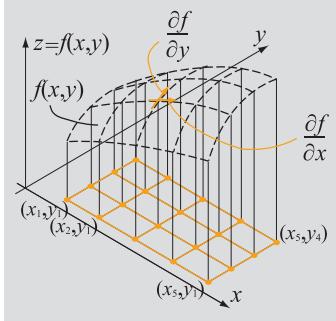
Depender do tempo. Por exemplo, a distribuição de temperaturas em um objeto é função das três coordenadas usadas para descrever esse objeto:  $T(x, y, z)$ ,  $T(r, \theta, z)$ , ou  $T(r, \theta, \phi)$ . A temperatura também pode ser uma função do tempo:  $T(x, y, z, t)$ . Se houver a necessidade de se avaliar o fluxo de calor em uma determinada direção, por exemplo na direção  $z$ , é necessário conhecer a derivada parcial da temperatura na direção  $z$ :  $\frac{\partial T(x, y, z, t)}{\partial z}$ . Outro exemplo é a determinação de deformações a partir de deslocamentos. Se deslocamentos bidimensionais forem medidos na superfície de uma estrutura, as deformações podem ser determinadas a partir das derivadas parciais dos deslocamentos.

Para uma função de muitas variáveis independentes, a derivada parcial da função em relação a uma das variáveis representa a taxa de variação da função

em relação a essa variável, sendo as demais variáveis mantidas constantes (ver Seção 2.6). Para uma função  $f(x, y)$  de duas variáveis independentes, as derivadas parciais em relação a  $x$  e  $y$  no ponto  $(a, b)$  são definidas como:

$$\frac{\partial f(x, y)}{\partial x} \Big|_{\substack{x=a \\ y=b}} = \lim_{x \rightarrow a} \frac{f(x, b) - f(a, b)}{x - a} \quad (6.57)$$

$$\frac{\partial f(x, y)}{\partial y} \Big|_{\substack{x=a \\ y=b}} = \lim_{y \rightarrow b} \frac{f(a, y) - f(a, b)}{y - b} \quad (6.58)$$



**Figura 6-7** Função de duas variáveis independentes.

Isso significa que as fórmulas de diferenças finitas usadas na aproximação das derivadas das funções de uma variável independente podem ser adotadas no cálculo das derivadas parciais. As fórmulas são aplicadas em cada uma das variáveis, enquanto as demais variáveis são mantidas constantes. Por exemplo, considere uma função de duas variáveis independentes  $f(x, y)$  especificada como um conjunto de  $m \cdot n$  pontos discretos  $(x_1, y_1), (x_1, y_2), \dots, (x_n, y_m)$ . O espaçamento entre os pontos em cada direção é constante, isto é,  $h_x = x_{i+1} - x_i$  e  $h_y = y_{i+1} - y_i$ . A Fig. 6-7 mostra um caso em que  $n = 5$  e  $m = 4$ . Uma aproximação para a derivada parcial no ponto  $(x_i, y_i)$  usando a fórmula de diferença progressiva com dois pontos é:

$$\frac{\partial f}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{h_x} \quad (6.59)$$

$$\frac{\partial f}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_i, y_{i+1}) - f(x_i, y_i)}{h_y} \quad (6.60)$$

Da mesma maneira, as fórmulas de diferenças regressiva e central com dois pontos são:

$$\frac{\partial f}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_i, y_i) - f(x_{i-1}, y_i)}{h_x} \quad \frac{\partial f}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_i, y_i) - f(x_i, y_{i-1})}{h_y} \quad (6.61)$$

$$\frac{\partial f}{\partial x} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_{i+1}, y_i) - f(x_{i-1}, y_i)}{2h_x} \quad \frac{\partial f}{\partial y} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_i, y_{i+1}) - f(x_i, y_{i-1})}{2h_y} \quad (6.62)$$

As segundas derivadas parciais usando a fórmula de diferença central com três pontos são dadas por:

$$\frac{\partial^2 f}{\partial x^2} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_{i-1}, y_i) - 2f(x_i, y_i) + f(x_{i+1}, y_i)}{h_x^2} \quad (6.63)$$

$$\frac{\partial^2 f}{\partial y^2} \Big|_{\substack{x=x_i \\ y=y_i}} = \frac{f(x_i, y_{i-1}) - 2f(x_i, y_i) + f(x_i, y_{i+1})}{h_y^2} \quad (6.64)$$

De forma similar, todas as fórmulas de diferenças finitas listadas na Seção 6.4 podem ser adaptadas ao cálculo de derivadas parciais de diferentes ordens em relação a uma das variáveis.

Uma derivada parcial de segunda ordem também pode ser mista,  $\frac{\partial^2 f}{\partial x \partial y}$ . Essa derivada é realizada sucessivamente  $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial y} \left( \frac{\partial f}{\partial x} \right) = \frac{\partial}{\partial x} \left( \frac{\partial f}{\partial y} \right)$ . Pode-se obter uma fórmula de diferenças finitas para a derivada mista usando as fórmulas de diferenças de primeira ordem empregadas no cálculo de derivadas parciais. Por exemplo, a fórmula de diferença central com quatro pontos usada no cálculo da derivada mista de segunda ordem é obtida a partir das Eqs. (6.62):

$$\left. \frac{\partial^2 f}{\partial x \partial y} \right|_{\substack{x=x_i \\ y=y_i}} = \frac{[f(x_{i+1}, y_{i+1}) - f(x_{i-1}, y_{i+1})] - [f(x_{i+1}, y_{i-1}) - f(x_{i-1}, y_{i-1})]}{2h_x \cdot 2h_y} \quad (6.65)$$

A aplicação de fórmulas de diferenças finitas no cálculo de derivadas parciais é mostrada no Exemplo 6-7.

### Exemplo 6-7: Diferenciação parcial numérica

Os seguintes dados bidimensionais descrevendo o componente  $x$  da velocidade  $u$  em função das coordenadas  $x$  e  $y$  são medidos em um experimento:

	$x = 1,0$	$x = 1,5$	$x = 2,0$	$x = 2,5$	$x = 3,0$
$y = 1,0$	163	205	250	298	349
$y = 2,0$	228	291	361	437	517
$y = 3,0$	265	350	448	557	676

- (a) Usando as aproximações de diferença central, calcule  $\partial u / \partial x$ ,  $\partial u / (\partial y)$ ,  $\partial^2 u / \partial x^2$  e  $\partial^2 u / \partial x \partial y$  no ponto  $(2, 2)$ .
- (b) Usando uma aproximação de diferença progressiva com três pontos, calcule  $\partial u / \partial x$  no ponto  $(2, 2)$ .
- (c) Usando uma aproximação de diferença progressiva com três pontos, calcule  $\partial u / (\partial y)$  no ponto  $(2, 1)$ .

### SOLUÇÃO

- (a) Neste caso específico,  $x_i = 2$ ,  $y_i = 2$ ,  $x_{i-1} = 1,5$ ,  $x_{i+1} = 2,5$ ,  $y_{i-1} = 1$ ,  $y_{i+1} = 3$ ,  $h_x = 0,5$ ,  $h_y = 1$ . Usando as Eqs. (6.59) e (6.60), as derivadas parciais  $\partial f / \partial x$  e  $\partial f / (\partial y)$  são:

$$\left. \frac{\partial u}{\partial x} \right|_{\substack{x=x_i \\ y=y_i}} = \frac{u(x_{i+1}, y_i) - u(x_{i-1}, y_i)}{2h_x} = \frac{u(2,5; 2) - u(1,5; 2)}{2 \cdot 0,5} = \frac{437 - 291}{1} = 146$$

$$\left. \frac{\partial u}{\partial y} \right|_{\substack{x=x_i \\ y=y_i}} = \frac{u(x_i, y_{i+1}) - u(x_i, y_{i-1})}{2h_y} = \frac{u(2, 3) - u(2, 1)}{2 \cdot 1} = \frac{448 - 250}{2} = 99$$

A segunda derivada parcial é calculada com a Eq. (6.64):

$$\left. \frac{\partial^2 u}{\partial y^2} \right|_{\substack{x=x_i \\ y=y_i}} = \frac{u(x_i, y_{i-1}) - 2u(x_i, y_i) + u(x_i, y_{i+1})}{h_y^2} = \frac{250 - (2 \cdot 361) + 448}{1^2} = -24$$

A segunda derivada mista  $\partial^2 u / \partial x \partial y$  é dada pela Eq. (6.65):

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial y} \Bigg|_{\substack{x = x_i \\ y = y_i}} &= \frac{[u(x_{i+1}, y_{i+1}) - u(x_{i-1}, y_{i+1})] - [u(x_{i+1}, y_{i-1}) - u(x_{i-1}, y_{i-1})]}{2h_x \cdot 2h_y} \\ &= \frac{[u(2,5; 3) - u(1,5; 3)] - [u(2,5; 1) - u(1,5; 1)]}{2 \cdot 0,5 \cdot 2 \cdot 1} = \frac{[557 - 350] - [298 - 205]}{2 \cdot 0,5 \cdot 2 \cdot 1} = 57 \end{aligned}$$

- (b) Neste caso específico,  $x_i = 2$ ,  $x_{i+1} = 2,5$ ,  $x_{i+2} = 3$ ,  $0$ ,  $y_i = 2$  e  $h_x = 0,5$ . A fórmula de cálculo da derivada parcial  $\partial u / \partial x$  usando a fórmula de diferença finita progressiva com três pontos pode ser escrita a partir da expressão para a derivada primeira apresentada na Seção 6.4.

$$\begin{aligned} \frac{\partial u}{\partial x} \Bigg|_{\substack{x = x_i \\ y = y_i}} &= \frac{-3u(x_i, y_i) + 4u(x_{i+1}, y_i) - u(x_{i+2}, y_i)}{2h_x} = \\ &= \frac{-3u(2, 2) + 4u(2,5; 2) - u(3, 0; 2)}{2 \cdot 0,5} = \frac{-3 \cdot 361 + 4 \cdot 437 - 517}{2 \cdot 0,5} = 148 \end{aligned}$$

- (c) Neste caso específico,  $y_i = 1$ ,  $y_{i+1} = 2$ ,  $y_{i+2} = 3$ ,  $x_i = 2$  e  $h_y = 1$ . A fórmula de cálculo da derivada parcial  $\partial u / \partial y$  usando a fórmula de diferença finita progressiva com três pontos pode ser escrita a partir da expressão para a derivada primeira apresentada na Seção 6.4.

$$\begin{aligned} \frac{\partial u}{\partial y} \Bigg|_{\substack{x = x_i \\ y = y_i}} &= \frac{-3u(x_i, y_i) + 4u(x_i, y_{i+1}) - u(x_i, y_{i+2})}{2h_y} = \\ &= \frac{-3u(2, 1) + 4u(2, 2) - u(2, 3)}{2 \cdot 1} = \frac{-3 \cdot 250 + 4 \cdot 361 - 448}{2 \cdot 1} = 123 \end{aligned}$$

## 6.11 PROBLEMAS

### Problemas de solução manual

*Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos. Se usar o MATLAB, não utilize funções residentes para fazer a diferenciação.*

- 6.1** A partir dos dados a seguir,

<b>x</b>	0,398	0,399	0,400	0,401	0,402
<b>f(x)</b>	0,408591	0,409671	0,410752	0,411834	0,412915

determine a derivada primeira  $f'(x)$  no ponto  $x = 0,399$ .

- (a) Use a fórmula de diferença progressiva com três pontos.  
 (b) Use a fórmula de diferença central com dois pontos.

- 6.2** Os dados a seguir mostram a população do Nepal em anos selecionados entre 1980 e 2005.

<b>Ano</b>	1980	1985	1990	1995	2000	2005
<b>População (milhões)</b>	15	17	19,3	22	24,5	27,1

Calcule a taxa de crescimento da população, em milhões por ano, para o ano de 2005.

- Use a fórmula de diferença regressiva com dois pontos.
- Use a fórmula de diferença regressiva com três pontos.
- Usando a inclinação em 2005 obtida na letra (b), aplique a fórmula de diferença central com dois pontos para extrapolar e prever a população no ano de 2010.

- 6.3** Os dados a seguir referem-se à distância necessária para um veículo parar *versus* a velocidade na qual ele começa a frenagem.

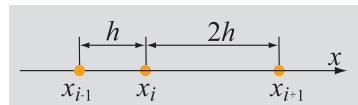
Distância de frenagem (m)	20	35	80	110	150
Velocidade (km/h)	20	30	40	50	60

- Calcule a taxa de variação da distância de frenagem a uma velocidade de 96,5 km/h usando (i) a fórmula de diferença regressiva com dois pontos, e (ii) a fórmula de diferença regressiva com três pontos.
- Estime a distância de frenagem a 110 km/h usando os resultados obtidos na letra (a) e a fórmula de diferença central com dois pontos a uma velocidade de 96,5 km/h.

- 6.4** Dados três pontos *não-uniformemente espaçados*  $(x_i, y_i)$ ,  $(x_{i+1}, y_{i+1})$  e  $(x_{i+2}, y_{i+2})$ , use a expansão em série de Taylor para desenvolver uma fórmula de diferenças finitas para avaliar a derivada primeira  $dy/dx$  no ponto  $x = x_i$ . Verifique que, quando o espaçamento entre esses pontos é igual, obtém-se a fórmula de diferença finita progressiva com três pontos. A resposta deve envolver  $y_i$ ,  $y_{i+1}$  e  $y_{i+2}$ .

- 6.5** Usando a expansão em série de Taylor com 4 termos, deduza uma fórmula de diferença regressiva com quatro pontos para avaliar a derivada primeira de uma função dada por um conjunto de pontos não-uniformemente espaçados. A fórmula deve fornecer a derivada no ponto  $x = x_i$  em termos de  $x_i$ ,  $x_{i-1}$ ,  $x_{i-2}$ ,  $x_{i-3}$ ,  $f(x_i)$ ,  $f(x_{i-1})$ ,  $f(x_{i-2})$  e  $f(x_{i-3})$ .

- 6.6** Deduza uma aproximação por diferenças finitas para  $f''(x_i)$  usando três pontos  $x_{i-1}$ ,  $x_i$  e  $x_{i+1}$ , onde o espaçamento é tal que  $x_i - x_{i-1} = h$  e  $x_{i+1} - x_i = 2h$ .



- 6.7** Uma fórmula de diferenças finitas específica empregada no cálculo da derivada primeira de uma função é:

$$f'(x_i) = \frac{-f(x_{i+3}) + 9f(x_{i+1}) - 8f(x_i)}{6h}$$

onde os pontos  $x_i$ ,  $x_{i+1}$ ,  $x_{i+2}$  e  $x_{i+3}$  estão igualmente espaçados com um passo  $h$ . Qual é a ordem do erro de truncamento ou discretização?

- 6.8** Os dados a seguir mostram o números de físicos homens e mulheres nos EUA ao longo de vários anos.

Ano	1970	1980	1990	2000	2002	2003	2004
# homens	308.627	413.395	511.227	618.182	638.182	646.493	647.347
# mulheres	25.401	54.284	104.194	195.537	215.005	225.042	235.627

- (a) Calcule a taxa de variação no número de físicos homens e mulheres em 2002 usando a fórmula de diferença regressiva com três pontos não-uniformemente espaçados (Eq. 6.37).
- (b) Use o resultado da letra (a) e a fórmula de diferença central com três pontos dada pela Eq. (6.36) para calcular (prever) o número de físicos homens e mulheres em 2010.
- 6.9** Use os dados do Problema 6.8 e a fórmula de diferença regressiva com quatro pontos deduzida no Problema 6.5:
- (a) Avalie a taxa de variação no número de físicos homens e mulheres em 2004.
- (b) Use os dados de 2003 e 2004 juntamente com a inclinação em 2004 obtida na letra (a) para estimar o ano no qual o número de físicos homens e mulheres será igual. Use a fórmula de diferença central com três pontos para calcular a derivada de uma função especificada em pontos não-uniformemente espaçados (Eq. (6.36)).
- 6.10** Use polinômios interpoladores de Lagrange para obter a fórmula de diferenças finitas para o cálculo da derivada segunda no ponto  $x = x_i$  usando pontos não-uniformemente espaçados  $x_i, x_{i+1}$  e  $x_{i+2}$ . Qual é a derivada segunda em  $x = x_{i+1}$  e em  $x = x_{i+2}$ ?
- 6.11** Dada a função  $f(x) = \frac{2x-1}{(x^4 \sin x + x + 1)^{1/4}}$ , determine o valor da derivada primeira em  $x = 2$ .
- (a) Calcule manualmente a derivada analítica da função.
- (b) Use a fórmula de diferença central com quatro pontos em  $x_{i-2} = 1,96$ ,  $x_{i-1} = 1,98$ ,  $x_{i+1} = 2,02$  e  $x_{i+2} = 2,04$  (escreva um programa no MATLAB para realizar os cálculos).
- 6.12** Para a função dada no Problema 6.11, determine o valor da derivada segunda em  $x = 2$ .
- (a) Calcule manualmente a derivada da função.
- (b) Use a fórmula de diferença central com cinco pontos em  $x_{i-2} = 1,96$ ,  $x_{i-1} = 1,98$ ,  $x_i = 2$ ,  $x_{i+1} = 2,02$  e  $x_{i+2} = 2,04$  (escreva um programa no MATLAB para realizar os cálculos).
- 6.13** Os seguintes dados representando o componente  $x$  da velocidade  $u$  são obtidos em função das coordenadas  $x$  e  $y$ .

	$x = 0$	$x = 1$	$x = 2$	$x = 3$	$x = 4$
$y = 0$	0	2	8	13	15
$y = 1$	3	7	10	15	18
$y = 2$	14	8	14	22	22
$y = 3$	7	9	12	16	17
$y = 4$	5	7	10	9	14

Use a fórmula de diferença central com quatro pontos para avaliar  $\frac{\partial^2 u}{\partial y \partial x}$  no ponto  $(2, 3)$ .

### Problemas de programação no MATLAB.

Resolva os seguintes problemas no ambiente MATLAB.

**6.14** Escreva uma função no MATLAB que determine a derivada primeira de uma função descrita por um conjunto de pontos igualmente espaçados. Chame a função de  $yd = \text{DerivPrim}(x, y)$ . Os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos, e o argumento de saída  $yd$  é um vetor com os valores da derivada em cada ponto. No primeiro e no último ponto, a função deve calcular a derivada usando as fórmulas de diferenças progressiva e regressiva com três pontos, respectivamente. Nos demais pontos,  $\text{DerivPrim}$  deve usar a fórmula de diferença central com dois pontos. Use  $\text{DerivPrim}$  para calcular a derivada da função dada no Problema 6.1.

**6.15** Escreva uma função no MATLAB que determine a derivada segunda de uma função descrita por um conjunto de pontos igualmente espaçados. Chame a função de  $ydd = \text{DerivSeg}(x, y)$ . Os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos, e o argumento de saída  $ydd$  é um vetor com os valores da derivada segunda em cada ponto. Para calcular a derivada, a função  $\text{DerivSeg}$  deve usar fórmulas de diferenças finitas que tenham um erro de truncamento de  $O(h^2)$ . Use  $\text{DerivSeg}$  para calcular a derivada da função dada pelo conjunto de pontos a seguir.

$x$	-1	-0,5	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5
$f(x)$	-3,632	-0,3935	1	0,6487	-1,282	-4,518	-8,611	-12,82	-15,91	-15,88	-9,402	9,017

**6.16** Escreva uma função no MATLAB que determine as derivadas primeira e segunda de uma função descrita por um conjunto de pontos igualmente espaçados. Chame a função de  $[yd, ydd] = \text{DerivPrimSeg}(x, y)$ . Os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos, e os argumentos de saída  $yd$  e  $ydd$  são vetores com os valores das derivadas primeira e segunda em cada ponto, respectivamente. Para calcular as derivadas, a função deve usar fórmulas de diferenças finitas que tenham um erro de truncamento de  $O(h^2)$ .

- (a) Use a função  $\text{DerivPrimSeg}$  para calcular as derivadas da função dada pelo conjunto de pontos do Problema 6.15.
- (b) Modifique a função (mude seu nome para  $\text{DerivPrimSegGraf}$ ) de forma tal que ela também crie três gráficos (em uma mesma coluna). O gráfico de cima deve mostrar a função, o gráfico do meio deve mostrar a derivada

primeira e o terceiro gráfico, a derivada segunda. Aplique a função DerivPrimSegGraf nos dados do Programa 6.15.

**6.17** Escreva uma função no MATLAB que determine a derivada de uma função dada na forma analítica. Chame a função de  $\text{dfx} = \text{DerivAnalit}(\text{'NomeFun'}, \text{xi})$ . '*NomeFun*' é uma *string* com o nome da função que calcula o valor da expressão a ser diferenciada em um dado  $x$ , e  $\text{xi}$  é o valor de  $x$  onde se deseja calcular a derivada. A função deve calcular a derivada usando a fórmula de diferença central com dois pontos. Na fórmula, os valores de  $(x_{i+1})$  e  $(x_{i-1})$  devem ser 5% maiores e menores que o valor de  $x_i$ , respectivamente.

- (a) Use a função para calcular a derivada primeira de  $f(x) = \frac{2^x}{x}$  em  $x = 2$ .
- (b) Use a função para calcular a derivada primeira da função dada no Problema 6.11 em  $x = 2$ .

**6.18** Modifique a função escrita no Problema 6.17 para incluir a extrapolação de Richardson. A função deve calcular uma primeira estimativa para a derivada conforme descrito no Problema 6.17, e uma segunda estimativa escolhendo valores  $(x_{i+1})$  e  $(x_{i-1})$  que sejam 2,5% maiores e menores do que o valor de  $x_i$ , respectivamente. As duas estimativas devem então ser usadas com a extrapolação de Richardson para calcular a derivada. Use  $\text{dfx} = \text{DifRichardson}(\text{'NomeFun'}, \text{xi})$  como nome da função e argumentos.

- (a) Use a função para calcular a derivada primeira de  $f(x) = \frac{2^x}{x}$  em  $x = 2$ .
- (b) Use a função para calcular a derivada primeira da função dada no Problema 6.11 em  $x = 2$ .

**6.19** Escreva uma função no MATLAB que calcule a derivada segunda de uma função dada na forma analítica. Chame a função de  $\text{ddfx} = \text{DDerivAnalit}(\text{'NomeFun'}, \text{xi})$ . '*NomeFun*' é uma *string* com o nome da função que calcula o valor da expressão a ser diferenciada em um dado  $x$ , e  $\text{xi}$  é o valor de  $x$  onde se deseja calcular a derivada segunda. A função deve calcular a derivada segunda usando a fórmula de diferença central com três pontos. Na fórmula, os valores de  $(x_{i+1})$  e  $(x_{i-1})$  devem ser 5% maiores e menores que o valor de  $x_i$ , respectivamente.

- (a) Use a função para calcular a derivada segunda de  $f(x) = \frac{2^x}{x}$  em  $x = 2$ .
- (b) Use a função para calcular a derivada segunda da função dada no Problema 6.11 em  $x = 2$ .

**6.20** Escreva uma função no MATLAB que determine a derivada primeira de uma função descrita por um conjunto de pontos não-uniformemente espaçados. Chame a função de  $\text{yd} = \text{DerivPrimLag}(\text{x}, \text{y})$ . Os argumentos de entrada  $\text{x}$  e  $\text{y}$  são vetores com as coordenadas dos pontos, e o argumento de saída  $\text{yd}$  é um vetor com os valores da derivada em cada ponto. No primeiro e no último ponto, a função deve calcular a derivada usando as Eqs. (6.35) e (6.37), respectivamente. Nos demais pontos, a função deve usar a Eq. (6.36). Use a função para calcular a derivada do conjunto de pontos a seguir.

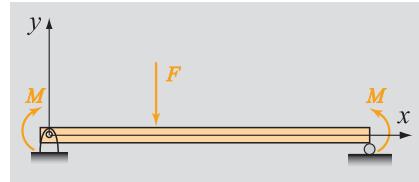
$x$	-1	-0,6	-0,3	0	0,5	0,8	1,6	2,5	2,8	3,2	3,5	4
$f(x)$	-3,632	-0,8912	0,3808	1,0	0,6487	-0,3345	-5,287	-12,82	-14,92	-16,43	-15,88	-9,402

### Problemas na matemática, na ciência e na engenharia

Resolva os seguintes problemas no ambiente MATLAB. Use programas apresentados neste capítulo, programas desenvolvidos em problemas já resolvidos ou funções residentes do MATLAB.

- 6.21** Uma viga com 1 m de comprimento é suportada em ambas as extremidades e submetida a uma carga. A deflexão da viga é dada pela equação diferencial:

$$\frac{d^2y}{dx^2} = -\frac{M(x)}{EI}$$



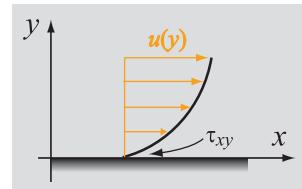
onde  $y$  é a deflexão,  $x$  é a coordenada medida ao longo do comprimento da viga,  $M(x)$  é o momento fletor e  $EI = 1,2 \times 10^7 \text{ N}\cdot\text{m}^2$  é a rigidez flexural da viga. Os dados a seguir são obtidos a partir da medição da deflexão da viga versus posição.

$x$ (m)	0	0,2	0,4	0,6	0,8	1,0
$y$ (cm)	0	7,78	10,68	8,38	3,97	0

Determine o momento fletor  $M(x)$  em cada  $x$  a partir desses dados. Use a aproximação de diferença central nos pontos internos com erro de  $O(h^2)$ . Nos pontos finais, use aproximações de diferenças progressiva e regressiva com quatro pontos e erro de  $O(h^2)$ .

- 6.22** A distribuição do componente  $x$  da velocidade  $u$  de um fluido próximo à superfície é medida em função da distância  $y$  a partir da superfície:

$y$ (m)	0	0,002	0,004	0,006	0,008
$u$ (m/s)	0	0,00618	0,011756	0,01618	0,019021



A tensão de cisalhamento  $\tau_{yx}$  no fluido é descrita pela equação de Newton:

$$\tau_{yx} = \mu \frac{\partial u}{\partial y}$$

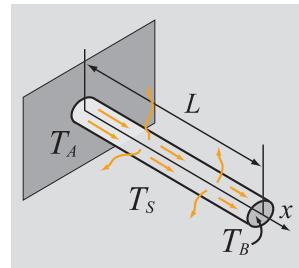
onde  $\mu$  é o coeficiente de viscosidade dinâmica. A viscosidade pode ser pensada como sendo uma medida da fricção interna no fluido. Fluidos que obedecem à equação constitutiva de Newton são chamados de fluidos Newtonianos. Calcule a tensão de cisalhamento em  $y = 0$  usando as aproximações de diferenças (i) progressiva com dois pontos e (ii) progressiva com três pontos para a derivada.

- 6.23** Um dissipador é uma superfície estendida usada para transferir calor de um material (em  $x = 0$ ) para o ambiente. O calor sai do material pela superfície externa e pela ponta do dissipador. A medição da distribuição de temperaturas ao longo de um dissipador fornece os dados a seguir:

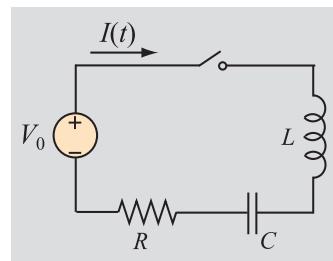
$x$ (cm)	0	1	2	3	4	5	6	7	8	9	10
$T$ (K)	473	446,3	422,6	401,2	382	364,3	348,0	332,7	318,1	304,0	290,1

O dissipador tem um comprimento  $L = 10$  cm, área da seção reta de  $1,6 \times 10^{-5}$  m<sup>2</sup> e condutividade térmica  $k = 240$  W/m/K. O fluxo de calor (W/m<sup>2</sup>) é dado por  $q_x = -k \frac{dT}{dx}$ .

- Determine o fluxo de calor em  $x = 0$ . Use a fórmula de diferença progressiva com três pontos para calcular a derivada.
- Determine o fluxo de calor em  $x = L$ . Use a fórmula de diferença regressiva com três pontos para calcular a derivada.
- Determine a quantidade de calor (em W) perdida entre  $x = 0$  e  $x = L$  (o fluxo de calor por unidade de tempo em Watts é igual ao fluxo de calor multiplicado pela área da seção reta do dissipador).



**6.24** A carga no capacitor do circuito RLC mostrado na figura ao lado em vários instantes de tempo após o fechamento da chave em  $t = 0$  é dada na tabela a seguir. A corrente  $I$  em função do tempo é dada por  $I(t) = \frac{dQ}{dt}$ . Determine a corrente fazendo a diferenciação numérica dos dados.

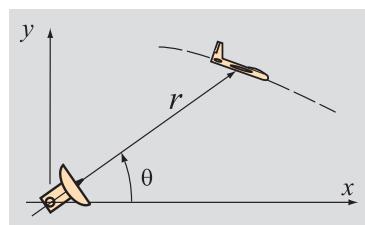


- Use a função `yd = DerivPrimLag(x, y)` escrita no Problema 6.20.
- Use a função residente do MATLAB `diff`.

Em ambas as letras, trace um gráfico de  $I$  versus  $t$ .

$Q \times 10^3$ (C)	0	6,67	16,93	23,38	27,44	29,02	29,5	29,29	27,17	24,58	21,43	18,56	16,61	16,01
$t$ (ms)	0	6,9	12,84	17,23	21,24	24,05	27,01	28,49	32,95	36,33	40,1	44,06	48,21	51,94
$Q \times 10^3$ (C)	16,87	18,06	19,51	21,42	22,99	24,05	24,12	23,35	22,5	21,53	20,33	19,44	18,98	19,36
$t$ (ms)	56,94	59,9	62,86	66,69	70,52	75,3	78,48	83,26	86,29	89,33	93,21	97,09	101,6	107,7

**6.25** Uma estação de radar faz o rastreamento do movimento de uma aeronave. A distância  $r$  da aeronave e o ângulo  $\theta$  durante um período de 60 s são apresentados na tabela a seguir. O módulo da velocidade instantânea e da aceleração da aeronave podem ser calculados com:



$$v = \sqrt{\left(\frac{dr}{dt}\right)^2 + \left(r\frac{d\theta}{dt}\right)^2} \quad a = \sqrt{\left[\frac{d^2r}{dt^2} - r\left(\frac{d\theta}{dt}\right)^2\right]^2 + \left[r\frac{d^2\theta}{dt^2} + 2\frac{dr}{dt}\frac{d\theta}{dt}\right]^2}$$

Determine os módulos da velocidade e da aceleração nos tempos dados na tabela. Trace gráficos da velocidade e da aceleração *versus* tempo (dois gráficos separados na mesma página). Resolva o problema escrevendo um programa que avalie as várias derivadas necessárias para o cálculo da velocidade e da aceleração, e então faça os gráficos. Para calcular as derivadas, use:

(a) A função `DerivPrimSeg` escrita no Problema 6.16.

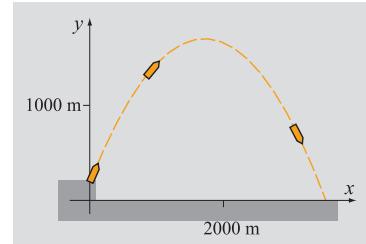
(b) A função residente do MATLAB `diff`.

<i>t</i> (s)	0	4	8	12	16	20	24	28
<i>r</i> (km)	18,803	18,861	18,946	19,042	19,148	19,260	19,376	19,495
$\theta$ (rad)	0,7854	0,7792	0,7701	0,7594	0,7477	0,7350	0,7215	0,7073
<i>t</i> (s)	32	36	40	44	48	52	56	60
<i>r</i> (km)	19,617	19,741	19,865	19,990	20,115	20,239	20,362	20,484
$\theta$ (rad)	0,6925	0,6771	0,6612	0,6448	0,6280	0,6107	0,5931	0,5750

**6.26** Um projétil é disparado de um precipício com 200 m de altura conforme mostrado. Sua posição (coordenadas *x* e *y*) em função do tempo *t* é dada por  $v = \sqrt{v_x^2 + v_y^2}$ , cujos componentes horizontal e vertical,  $v_x$  e  $v_y$ , são dados por  $v_x = \frac{\partial x}{\partial t}$  e  $v_y = \frac{\partial y}{\partial t}$ .

Escreva um programa no MATLAB que:

- Determine  $v_x$  e  $v_y$  usando a função `DerivPrim` escrita no Problema 6.14.
- Determine a velocidade *v* do projétil.
- Crie uma figura com gráficos de  $v_x$ ,  $v_y$  e *v* em função do tempo (três gráficos em uma só figura).



<i>t</i> (s)	0	2	4	6	8	10	12	14	16	18
<i>x</i> (m)	0	198	395	593	790	988	1185	1383	1580	1778
<i>y</i> (m)	200	523	806	1050	1254	1420	1546	1633	1681	1690
<i>t</i> (s)	20	22	24	26	28	30	32	34	36	
<i>x</i> (m)	1975	2173	2370	2568	2765	2963	3160	3358	3555	
<i>y</i> (m)	1659	1589	1480	1331	1144	917	651	345	1	

# Integração Numérica

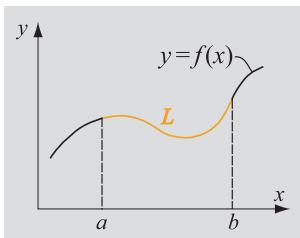
## Tópicos centrais

- Métodos do retângulo e do ponto central (7.2)
- Método trapezoidal (7.3)
- Métodos de Simpson (7.4)
- Quadratura de Gauss (7.5)
- Avaliação de integrais múltiplas (7.6)
- Uso de funções residentes do MATLAB para a integração (7.7)

## Tópicos Complementares

- Estimação de erros (7.8)
- Extrapolação de Richardson (7.9)
- Integração de Romberg (7.10)
- Integrais impróprias (7.11)

## 7.1 FUNDAMENTOS



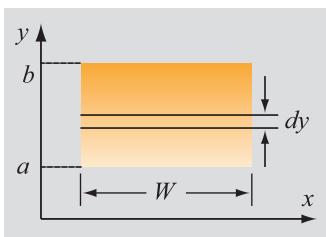
**Figura 7-1** Comprimento de uma curva.

A integração aparece com freqüência na solução de problemas e no cálculo de grandezas na engenharia e na ciência. Integrais e a operação de integração também são usadas na solução de equações diferenciais. Um dos exemplos mais simples da aplicação da integração é o cálculo do comprimento de uma curva (Fig. 7-1). Quando uma curva no plano x-y é descrita pela equação  $y = f(x)$ , o comprimento  $L$  da curva entre os pontos  $x = a$  e  $x = b$  é dada por:

$$L = \int_a^b \sqrt{1 + [f'(x)]^2} dx$$

Na engenharia, há circunstâncias que envolvem dados experimentais ou de teste, nos quais uma grandeza física a ser determinada pode ser expressa como a integral de outras grandezas medidas. Por exemplo, a taxa de fluxo total atravessando uma seção reta de largura  $W$  e altura  $(b - a)$  se relaciona ao fluxo de calor local por meio de uma integral (veja a Fig. 7-2):

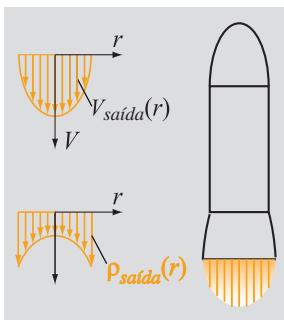
$$\dot{Q} = \int_{y=a}^{y=b} \dot{q}'' W dy$$



**Figura 7-2** Fluxo de calor atravessando uma seção reta retangular.

onde  $\dot{q}''$  é o fluxo de calor e  $\dot{Q}$  é a taxa de fluxo de calor. Medições experimentais podem fornecer valores discretos para o fluxo de calor ao longo da superfície em função de  $y$ , mas a grandeza a ser

determinada pode ser a taxa de fluxo de calor total. Neste caso, o integrando pode ser especificado como um conjunto conhecido de valores em cada valor de  $y$ .



**Figura 7-3** Sistema de impulsão de um foguete.

Em mais uma ilustração, considere o sistema de impulsão de um foguete. Conforme mostrado na Fig. 7-3, a velocidade e a densidade de fluxo saindo do motor não são uniformes ao longo da área do exaustor. Para uma área de seção reta circular, ambos variam com a coordenada radial  $r$ . A expressão resultante para o impulso pode ser obtida a partir da conservação do momento em regime permanente:

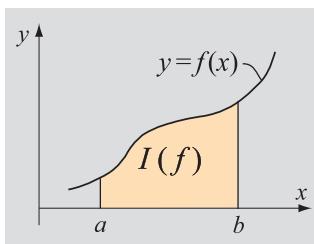
$$T = \int_0^R 2\pi\rho(r)V_{saída}^2(r)rdr$$

Onde  $T$  é o impulso,  $\rho(r)$  é a densidade de massa do fluido,  $V_{saída}(r)$  é o perfil da velocidade na saída do motor,  $r$  é a coordenada radial e  $R$  é o raio do exaustor. Cálculos computacionais podem fornecer diretamente  $\rho(r)$  e  $V_{saída}(r)$ , mas o impulso (uma grandeza que pode ser medida através de experimentos ou testes) deve ser obtido por meio da integração.

A fórmula geral de uma integral definida, também chamada de antiderivada, é:

$$I(f) = \int_a^b f(x)dx \quad (7.1)$$

onde  $f(x)$ , chamada de integrando, é função da variável independente  $x$ , enquanto  $a$  e  $b$  são os limites de integração. A integral  $I(f)$  é um número quando  $a$  e  $b$  são números. Graficamente, conforme mostrado na Fig. 7-4, o valor da integral corresponde à área sombreada sob a curva de  $f(x)$  entre  $a$  e  $b$ .

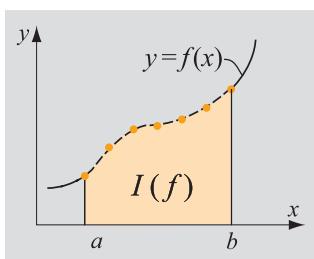


**Figura 7-4** Integral definida de  $f(x)$  entre  $a$  e  $b$ .

### A necessidade de se calcular uma integral numericamente

O integrando pode ser uma função analítica ou um conjunto de pontos discretos (dados tabulados). Quando o integrando é uma expressão matemática cuja integral pode ser facilmente calculada, pode-se obter analiticamente o valor da integral definida. A integração numérica é necessária quando a integração analítica é difícil, ou mesmo impossível, e quando o integrando é fornecido como um conjunto discreto de pontos.

#### 7.1.1 Visão geral sobre as abordagens usadas na integração numérica

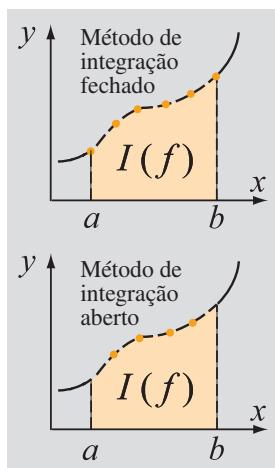


**Figura 7-5** Um número finito de pontos é usado na integração numérica.

A avaliação numérica de uma integral consiste em estimar o número  $I(f)$  correspondente à integral de uma função  $f(x)$  entre os limites  $a$  e  $b$ . Se o integrando  $f(x)$  for uma função analítica, a integração numérica é feita usando um número finito de pontos (Fig. 7-5). Uma estratégia é usar apenas os pontos finais do intervalo,  $(a, f(a))$  e  $(b, f(b))$ . Isso, contudo, pode não fornecer um resultado suficientemente preciso, especialmente se o intervalo for relativamente grande ou se o integrando variar significativamente ao longo do intervalo. Uma precisão

maior pode ser obtida com o uso de um método composto no qual o intervalo  $[a, b]$  é dividido em subintervalos menores. Neste caso, calcula-se a integral ao longo de cada subintervalo e os resultados são somados para fornecer a integral completa. Se o integrando  $f(x)$  for dado como um conjunto de pontos discretos (dados tabulados), a integração numérica é feita a partir desses pontos.

Em todos os casos, a integração numérica é feita utilizando um conjunto de pontos discretos para representar o integrando. Quando o integrando for uma função analítica, a localização dos pontos dentro do intervalo  $[a, b]$  pode ser definida pelo usuário ou pelo método de integração. Quando o integrando for um conjunto tabulado de pontos (como dados medidos em um experimento), a localização dos pontos é fixa e não pode ser mudada.



**Figura 7-6** Métodos de integração aberto e fechado.

Existem vários métodos disponíveis para o cálculo numérico de integrais. Em cada um desses métodos, uma fórmula é deduzida para calcular o valor aproximado de uma integral a partir dos pontos discretos do integrando. Esses métodos podem ser divididos em métodos abertos e métodos fechados.

### Métodos abertos e fechados

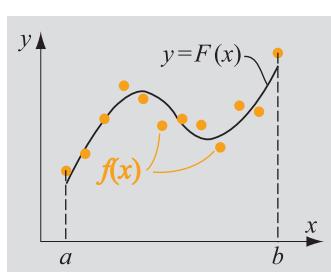
Em métodos de integração fechados, os pontos finais do intervalo (e o integrando) são usados na fórmula que estima o valor da integral. Em métodos de integração abertos, o intervalo de integração se estende além do limite especificado pelos pontos finais (Fig. 7-6). Os métodos trapezoidal (Seção 7.3) e de Simpson (Seção 7.4) são métodos fechados, enquanto os métodos do ponto central (Seção 7.2) e da quadratura de Gauss (Seção 7.5) são métodos abertos.

Há vários métodos que podem ser usados para calcular o valor de uma integral a partir dos pontos discretos do integrando. Mais comumente, isso é feito com o emprego de fórmulas de Newton-Cotes.

### Fórmulas de integração de Newton-Cotes

Nos métodos de integração numérica que usam fórmulas de Newton-Cotes, o valor do integrando é estimado entre os pontos discretos empregando-se uma função de fácil integração. Quando o integrando original é uma função analítica, a fórmula de Newton-Cotes a substitui por uma função mais simples. Quando o integrando original é dado na forma de pontos discretos, a fórmula de Newton-Cotes realiza a interpolação do integrando. Normalmente, assim como nos métodos trapezoidal (Seção 7.3) e de Simpson (Seção 7.4), as fórmulas de integração de Newton-Cotes são polinômios de diferentes graus.

Uma outra opção para a integração, caso o integrando  $f(x)$  seja especificado como um conjunto discreto de pontos, é obter a curva  $F(x)$  que melhor se ajusta a esses pontos. Em outras palavras, conforme mostrado na Fig. 7-7,  $f(x) \approx F(x)$ , onde  $F(x)$  é um polinômio ou uma função simples cuja antiderivada pode ser facilmente determinada. Nesse caso, a integral



**Figura 7-7** Cálculo da integral de  $f(x)$  usando uma função integrável  $F(x)$ .

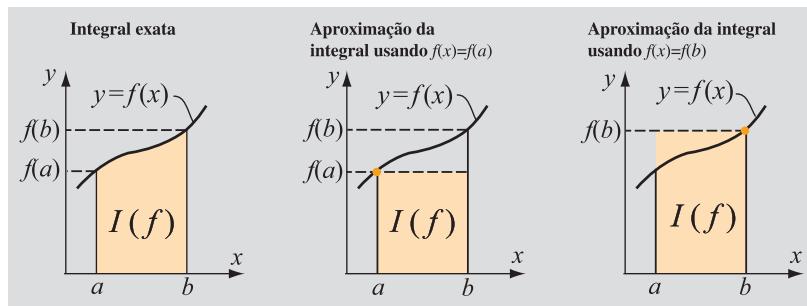
$$I(f) = \int_a^b f(x) dx \approx \int_a^b F(x) dx$$

é avaliada com métodos analíticos aprendidos na disciplina de cálculo. Esse procedimento requer o emprego de métodos numéricos para determinar  $F(x)$  (Capítulos 5 e 6), mas pode não requerer um método numérico para avaliar o integrando se  $F(x)$  for uma função integrável.

## 7.2 MÉTODOS DO RETÂNGULO E DO PONTO CENTRAL

### Método do retângulo

A aproximação mais simples para  $\int_a^b f(x)dx$  consiste em assumir que  $f(x)$  ao longo do intervalo  $x \in [a, b]$  seja uma constante igual ao valor de  $f(x)$  em qualquer um dos pontos finais desse intervalo (Fig. 7-8).



**Figura 7-8** Integração usando o método do retângulo.

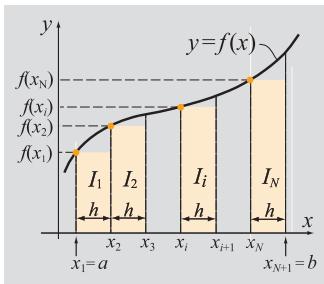
A integral pode ser então calculada de uma das seguintes maneiras:

$$I(f) = \int_a^b f(a)dx = f(a)(b-a) \quad \text{ou} \quad I(f) = \int_a^b f(b)dx = f(b)(b-a) \quad (7.2)$$

Como mostra a Fig. 7-8, a verdadeira integral é aproximada pela área de um retângulo. Obviamente, para a função monotonicamente crescente mostrada na figura, o valor da integral é subestimado quando se assume que  $f(x)$  seja igual a  $f(a)$ , e superestimado quando se assume que  $f(x)$  seja igual a  $f(b)$ . Além disso, o erro pode ser considerável. Quando o integrando for uma função analítica, o erro pode ser significativamente reduzido com o uso do método do retângulo composto.

### Método do retângulo composto

No método do retângulo composto, divide-se o domínio  $[a, b]$  em  $N$  subintervalos. A integral em cada subintervalo é calculada usando o método do retângulo, e o valor total da integral é obtido com a soma dos valores das integrais obtidos nos subintervalos. Isso é mostrado na Fig. 7-9, onde o intervalo  $[a, b]$  é dividido em  $N$  subintervalos com a definição dos pontos  $x_1, x_2, \dots, x_{N+1}$ . O primeiro ponto é  $x_1 = a$  e o último ponto é  $x_{N+1} = b$  (são necessários  $N + 1$  pontos para definir  $N$  intervalos). A Fig. 7-9 mostra subintervalos de mesma largura, mas, em geral, os



**Figura 7-9** Método do retângulo composto.

subintervalos podem ter largura arbitrária. Dessa forma, intervalos menores podem ser usados em regiões nas quais o valor do integrando varia rapidamente (inclinações maiores) e intervalos maiores podem ser usados quando o integrando varia de forma mais gradual.

Na Fig. 7-9, assume-se que o integrando em cada subintervalo tenha o valor do integrando no início do subintervalo. Usando a Eq. (7.2) em cada subintervalo, a integral em todo o domínio pode ser escrita como a soma das integrais nos subintervalos:

$$I(f) = \int_a^b f(x)dx \approx \underbrace{f(x_1)(x_2-x_1)}_{I_1} + \underbrace{f(x_2)(x_3-x_2)}_{I_2} + \dots + \underbrace{f(x_i)(x_{i+1}-x_i)}_{I_i} + \dots + \underbrace{f(x_N)(x_{N+1}-x_N)}_{I_N} = \sum_{i=1}^N [f(x_i)(x_{i+1}-x_i)] \quad (7.3)$$

Quando os subintervalos têm a mesma largura  $h$ , a Eq. (7.3) pode ser simplificada para:

$$I(f) = \int_a^b f(x)dx \approx h \sum_{i=1}^N f(x_i) \quad (7.4)$$

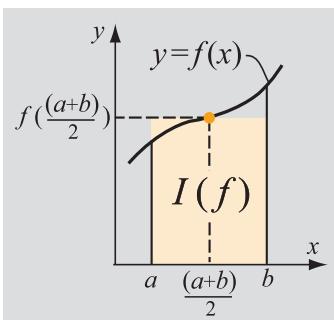
A Eq. (7.4) é a fórmula do método do retângulo composto quando os subintervalos têm uma mesma largura  $h$ .

### Método do ponto central

O método do ponto central representa uma melhoria em relação ao inocente método do retângulo. Ao invés de aproximar o integrando usando os valores da função em  $x = a$  ou  $x = b$ , utiliza-se o valor do integrando no meio do intervalo, isto é,  $f\left(\frac{(a+b)}{2}\right)$ . Substituindo na Eq. (7.1), obtém-se:

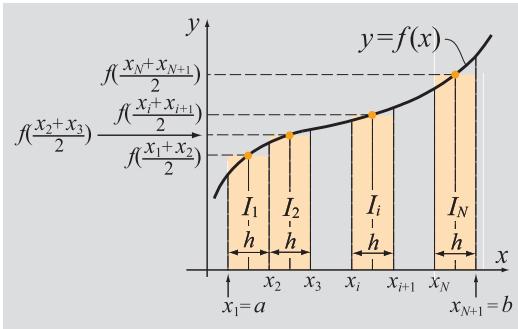
$$I(f) = \int_a^b f(x)dx \approx \int_a^b f\left(\frac{a+b}{2}\right)dx = f\left(\frac{a+b}{2}\right)(b-a) \quad (7.5)$$

Este método é mostrado graficamente na Fig. 7-10. Como se vê, o valor da integral ainda é aproximado como a área de um retângulo, mas com uma importante diferença – esta é a área de um retângulo **equivalente**. Esse método acaba sendo mais preciso do que o método do retângulo porque, para uma função monotônica como aquela mostrada na figura, as regiões da área abaixo da curva que são ignoradas podem ser compensadas de forma aproximada pelas regiões incluídas acima da curva. Entretanto, isso não é verdade em todos os casos, e com isso esse método pode ainda não ser suficientemente preciso. Como no método do retângulo, a precisão pode ser aumentada com o uso do método do ponto central composto.



**Figura 7-10** Método do ponto central.

As regiões da área abaixo da curva que são ignoradas podem ser compensadas de forma aproximada pelas regiões incluídas acima da curva. Entretanto, isso não é verdade em todos os casos, e com isso esse método pode ainda não ser suficientemente preciso. Como no método do retângulo, a precisão pode ser aumentada com o uso do método do ponto central composto.



**Figura 7-11** O método do ponto central composto.

A Fig. 7-11 mostra subintervalos de mesma largura, mas, em geral, os subintervalos podem ter largura arbitrária.

Usando a Eq. (7.5) em cada subintervalo, a integral em todo o domínio pode ser escrita como a soma das integrais nos subintervalos:

$$\begin{aligned}
 I(f) &= \int_a^b f(x) dx \approx \overbrace{f\left(\frac{x_1+x_2}{2}\right)(x_2-x_1)}^{I_1} + \overbrace{f\left(\frac{x_2+x_3}{2}\right)(x_3-x_2)}^{I_2} + \dots \\
 &\quad + \overbrace{f\left(\frac{x_i+x_{i+1}}{2}\right)(x_{i+1}-x_i)}^{I_i} + \dots + \overbrace{f\left(\frac{x_N+x_{N+1}}{2}\right)(x_{N+1}-x_N)}^{I_N} \\
 &= \sum_{i=1}^N \left[ f\left(\frac{x_i+x_{i+1}}{2}\right)(x_{i+1}-x_i) \right] \tag{7.6}
 \end{aligned}$$

Quando os subintervalos têm a mesma largura  $h$ , a Eq. (7.6) pode ser simplificada para:

$$I(f) = \int_a^b f(x) dx \approx h \sum_{i=1}^N f\left(\frac{x_i+x_{i+1}}{2}\right) \tag{7.7}$$

A Eq. (7.7) é a fórmula do método do ponto central composto quando os subintervalos têm uma mesma largura  $h$ .

### 7.3 MÉTODO TRAPEZOIDAL

Um refinamento dos métodos simples do retângulo e do ponto central consiste no uso de uma função linear para aproximar o integrando ao longo do intervalo de integração (Fig. 7-12). O uso do polinômio interpolador de Newton entre os pontos  $x = a$  e  $x = b$  resulta em:

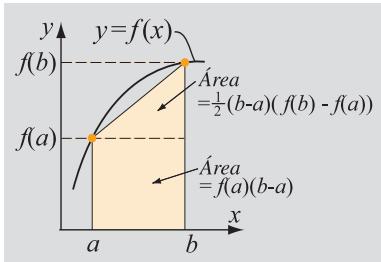
$$f(x) \approx f(a) + (x-a)f[a, b] = f(a) + (x-a)\frac{[f(b)-f(a)]}{b-a} \tag{7.8}$$

Substituindo a Eq. (7.8) na Eq. (7.1) e realizando a integração analítica, obtém-se:

### Método do ponto central composto

No método do ponto central composto, divide-se o domínio  $[a, b]$  em  $N$  subintervalos. A integral em cada subintervalo é calculada usando o método do ponto central, e o valor total da integral é obtido com a soma dos valores das integrais obtidos nos subintervalos. Isso é mostrado na Fig. 7-11, onde o intervalo  $[a, b]$  é dividido em  $N$  subintervalos com a definição dos pontos  $x_1, x_2, \dots, x_{N+1}$ . O primeiro ponto é  $x_1 = a$  e o último ponto é  $x_{N+1} = b$  (são necessários  $N+1$  pontos para definir  $N$  intervalos).

$$\begin{aligned}
 I(f) &\approx \int_a^b \left( f(a) + (x-a) \frac{[f(b)-f(a)]}{b-a} \right) dx \\
 &= f(a)(b-a) + \frac{1}{2}[f(b)-f(a)](b-a)
 \end{aligned} \tag{7.9}$$



**Figura 7-12** O método trapezoidal.

Simplificando o resultado, obtém-se uma fórmula aproximada popularmente conhecida como regra ou método trapezoidal.

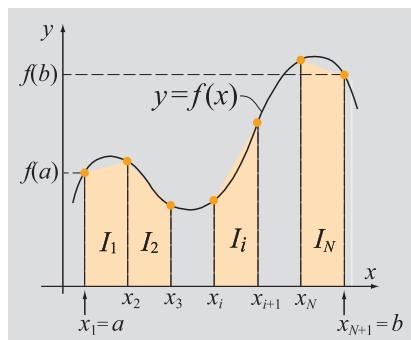
$$I(f) \approx \frac{[f(a)+f(b)]}{2}(b-a) \tag{7.10}$$

O exame do resultado antes da simplificação, isto é, o lado direito da Eq. (7.9), mostra que o primeiro termo,  $f(a)(b-a)$ , representa a área de um retângulo de altura  $f(a)$  e largura  $(b-a)$ . O segundo termo,  $\frac{1}{2}[f(b)-f(a)](b-a)$ , é a

área do triângulo cuja base é  $(b-a)$  e cuja altura é  $[f(b)-f(a)]$ . Essas áreas estão mostradas na Fig. 7-12 e servem para reforçar a noção de que, neste método, a área abaixo da curva  $f(x)$  é aproximada pela área de um trapézóide (retângulo + triângulo). Conforme mostrado na Fig. 7-12, essa abordagem é mais precisa do que o uso de um retângulo para aproximar a forma da região abaixo de  $f(x)$ .

Da mesma forma que os métodos do retângulo e do ponto central, o método trapezoidal também pode ser facilmente estendido para fornecer o nível de precisão desejado a partir da divisão do intervalo  $[a, b]$  em subintervalos.

### 7.3.1 Método trapezoidal composto



**Figura 7-13** Método trapezoidal composto.

A integral ao longo do intervalo  $[a, b]$  pode ser avaliada de forma mais precisa com a subdivisão do intervalo, a avaliação da integral em cada um dos subintervalos (com o método trapezoidal) e a soma dos resultados. Conforme mostrado na Fig. 7-13, o intervalo  $[a, b]$  é dividido em  $N$  subintervalos com a definição dos pontos  $x_1, x_2, \dots, x_{N+1}$ . O primeiro ponto é  $x_1 = a$  e o último ponto é  $x_{N+1} = b$  (são necessários  $N + 1$  pontos para definir  $N$  intervalos).

A integral ao longo do intervalo completo pode ser escrita como a soma das integrais dos subintervalos:

$$\begin{aligned}
 I(f) &= \int_a^b f(x)dx = \overbrace{\int_{x_1=a}^{x_2} f(x)dx}^{I_1} + \overbrace{\int_{x_2}^{x_3} f(x)dx}^{I_2} + \dots + \overbrace{\int_{x_i}^{x_{i+1}} f(x)dx}^{I_i} \\
 &\quad + \dots + \overbrace{\int_{x_N}^{x_{N+1}} f(x)dx}^{I_N} = \sum_{i=1}^N \int_{x_i}^{x_{i+1}} f(x)dx
 \end{aligned} \tag{7.11}$$

A aplicação do método trapezoidal em cada subintervalo  $[x_i, x_{i+1}]$  resulta em:

$$I_i(f) = \int_{x_i}^{x_{i+1}} f(x) dx \approx \frac{[f(x_i) + f(x_{i+1})]}{2} (x_{i+1} - x_i)$$

Substituindo a aproximação trapezoidal no lado direito da Eq. (7.11), obtém-se:

$$I(f) = \int_a^b f(x) dx \approx \frac{1}{2} \sum_{i=1}^N [f(x_i) + f(x_{i+1})] (x_{i+1} - x_i) \quad (7.12)$$

A Eq. (7.12) é a fórmula geral do método trapezoidal composto. Note que os subintervalos  $[x_i, x_{i+1}]$  não precisam ser idênticos (isto é, igualmente espaçados). Em outras palavras, cada um dos subintervalos pode ter uma largura diferente. Se, no entanto, todos os subintervalos tiverem o mesmo tamanho, ou seja, se

$$(x_2 - x_1) = (x_3 - x_2) = \dots = (x_{i+1} - x_i) = \dots = (x_N - x_{N-1}) = h$$

então a Eq. (7.12) pode ser simplificada para:

$$I(f) \approx \frac{h}{2} \sum_{i=1}^N [f(x_{i+1}) + f(x_i)]$$

Essa expressão pode ser reduzida a uma fórmula útil para a programação, na qual a soma é expandida:

$$I(f) \approx \frac{h}{2} [f(a) + 2f(x_2) + 2f(x_3) + \dots + 2f(x_N) + f(b)]$$

ou,

$$I(f) \approx \frac{h}{2} [f(a) + f(b)] + h \sum_{i=1}^N f(x_i) \quad (7.13)$$

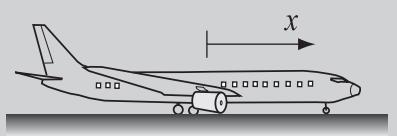
A Eq. (7.13) é a fórmula do método trapezoidal composto para o caso onde os subintervalos têm uma largura  $h$  idêntica.

O Exemplo 7-1 mostra a programação do método trapezoidal composto no MATLAB e seu uso na solução de um problema.

### Exemplo 7-1: Distância percorrida por um avião em desaceleração

Um Boeing 727-200 de massa  $m = 97000$  kg aterrissa a uma velocidade de 93 m/s (em torno de 181 nós) e liga os seus reversos em  $t = 0$ . A força  $F$  aplicada no avião à medida que ele reduz a sua velocidade é dada por  $F = -5v^2 - 570000$ , onde  $v$  é a velocidade do avião. Usando a segunda lei de Newton do movimento e da dinâmica dos fluidos, a relação entre a velocidade e a posição  $x$  do avião pode ser escrita como:

$$mv \frac{dv}{dx} = -5v^2 - 570000$$



onde  $x$  é a distância medida a partir da localização do jato em  $t = 0$ .

Determine a distância percorrida pelo avião antes que sua velocidade se reduza a 40 m/s (em torno de 78 nós) usando o método trapezoidal composto.

## SOLUÇÃO

Mesmo sabendo que a equação que descreve o problema é uma EDO, ela pode ser expressa como uma integral neste caso. Isso é feito com a separação das variáveis de tal forma que a velocidade  $v$  apareça de um lado da equação e  $x$  apareça do outro lado.

$$\frac{97000vdv}{(-5v^2 - 570000)} = dx$$

Em seguida, ambos os lados da equação são integrados. Para  $x$ , os limites de integração são 0 e uma localização arbitrária  $x$ ; para  $v$ , os limites são 93 m/s e 40 m/s.

$$\int_0^x dx = - \int_{93}^{40} \frac{97000v}{(5v^2 + 570000)} dv = \int_{40}^{93} \frac{97000v}{(5v^2 + 570000)} dv \quad (7.14)$$

O objetivo deste exemplo é mostrar como a integral definida no lado direito da equação pode ser determinada numericamente com o uso do método trapezoidal composto. Neste problema, contudo, a integração também pode ser feita analiticamente. Para comparação, a integração é feita das duas maneiras.

### Integração analítica

A integração pode ser realizada analiticamente usando substituição. Substituindo  $z = 5u^2 + 570000$ , o resultado da integral é  $x = 574,1494$  m.

### Integração numérica

Para realizar a integração numérica, a seguinte função, denominada *trapezoidal*, é criada no MATLAB:

#### Programa 7-1 Arquivo de função, método de integração trapezoidal.

```
function I = trapezoidal(integrand,a,b,N)
% A função trapezoidal calcula a integral usando o Método Trapezoidal Composto.
% Variáveis de entrada:
% integrando Função a ser integrada, digitada como uma string.
% a Limite inferior de integração.
% b Limite superior de integração.
% N Número de subintervalos.
% Variável de saída:
% I Valor da integral.

h = (b - a)/N;
func = inline(integrand);
x = a:h:b;
for i=1:N + 1
    F(i) = func(x(i));
end
I = h*(F(1) + F(N + 1))/2 + h*sum(F(2:N));
```

Calcula a largura  $h$  dos subintervalos.

Define o integrando como uma função *inline*.

Cria um vetor  $x$  com as coordenadas dos subintervalos.

Cria um vetor  $F$  com os valores do integrando em cada ponto  $x$ .

Calcula o valor da integral de acordo com a Eq. (7.13.).

A função `trapezoidal` é agora usada na janela de comandos do MATLAB para determinar o valor da integral na Eq. (7.14). Para examinar o efeito do número de subintervalos no resultado, a função é rodada três vezes, usando  $N = 10, 100$  e  $1000$ . O resultado apresentado na janela de comandos é:

```
>> format long g
>> distânci=trapezoidal('97000*v/(5*v^2+570000)',40,93,10)
distânci =
    574.085485133712
>> distânci=trapezoidal('97000*v/(5*v^2+570000)',40,93,100)
distânci =
    574.148773931409
>> distânci=trapezoidal('97000*v/(5*v^2+570000)',40,93,1000)
distânci =
    574.149406775129
```

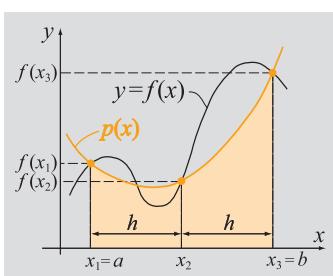
Conforme esperado, os resultados mostram que a integral é avaliada de forma mais precisa à medida que o número de subintervalos aumenta. Quando  $N = 1000$ , o resultado é igual àquele obtido analiticamente até as quatro primeiras casas decimais.

O Exemplo 7-1 revela dois pontos fundamentais:

- É importante verificar resultados obtidos por meio de cálculos numéricos (realizados manualmente ou em um computador), comparando-os com soluções analíticas conhecidas. Quando não houver uma solução analítica que possa ser usada, é necessário comparar a resposta obtida com aquela fornecida por outro método numérico.
- Em muitos problemas que envolvem a integração numérica, é possível melhorar a precisão da resposta usando mais subintervalos, isto é, reduzindo o tamanho do subintervalo.

## 7.4 MÉTODOS DE SIMPSON

O método trapezoidal descrito na última seção aproxima o integrando por uma linha reta. Em tese, uma melhor aproximação pode ser obtida com a representação do integrando como uma função não-linear de fácil integração. Uma classe de métodos com essa característica, chamados de regras ou métodos de Simpson, usa polinômios quadráticos (método de Simpson 1/3) e cúbicos (método de Simpson 3/8) para aproximar o integrando.



**Figura 7-14** Método de Simpson 1/3.

### 7.4.1 Método de Simpson 1/3

Neste método, um polinômio quadrático (de segunda ordem) é usado para aproximar o integrando (Fig. 7-14). Os coeficientes de um polinômio quadrático podem ser determinados a partir de três pontos. Para uma integral ao longo do domínio  $[a, b]$ ,

são usados os dois pontos finais  $x_1 = a$ ,  $x_3 = b$ , e o ponto central,  $x_2 = (a + b)/2$ . O polinômio pode ser escrito na forma:

$$p(x) = \alpha + \beta(x - x_1) + \gamma(x - x_1)(x - x_2) \quad (7.15)$$

onde  $\alpha$ ,  $\beta$  e  $\gamma$  são constantes desconhecidas avaliadas a partir da condição que diz que o polinômio deve passar por todos os pontos,  $p(x_1) = f(x_1)$ ,  $p(x_2) = f(x_2)$  e  $p(x_3) = f(x_3)$ . Isso resulta em:

$$\alpha = f(x_1), \quad \beta = [f(x_2) - f(x_1)]/(x_2 - x_1) \quad \text{e} \quad \gamma = \frac{f(x_3) - 2f(x_2) + f(x_1)}{2(h)^2}$$

onde  $h = (b - a)/2$ . Substituindo as constantes de volta na Eq. (7.15) e integrando  $p(x)$  ao longo do intervalo  $[a, b]$ , obtém-se:

$$\begin{aligned} I &= \int_{x_1}^{x_3} f(x)dx \approx \int_{x_1}^{x_3} p(x)dx = \frac{h}{3}[f(x_1) + 4f(x_2) + f(x_3)] \\ &= \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right] \end{aligned} \quad (7.16)$$

O valor da integral é mostrado na Fig. 7-14 como a área sombreada entre a curva  $p(x)$  e o eixo  $x$ . O nome 1/3 que aparece no método vem do fato de haver um fator de 1/3 multiplicando a expressão entre colchetes na Eq. (7.16).

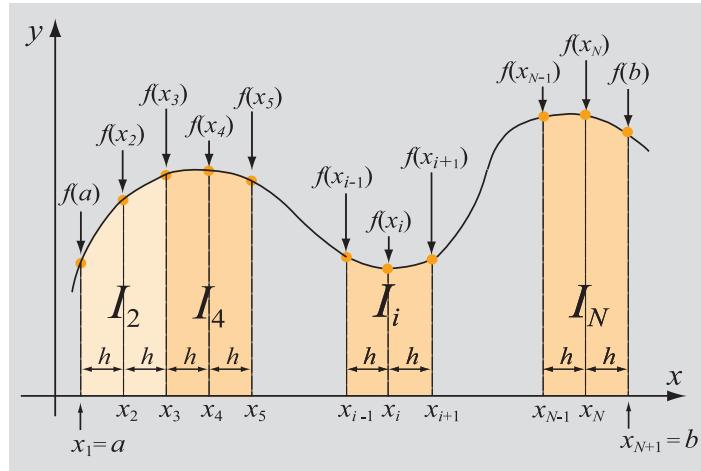
Da mesma forma que nos métodos trapezoidal e triangular, uma avaliação mais precisa da integral pode ser feita usando o método de Simpson 1/3 composto. Para isso, divide-se o intervalo em subintervalos. Em seguida, o método de Simpson 1/3 composto é usado para calcular a integral em cada subintervalo, e os valores são somados.

### Método de Simpson 1/3 composto

No método de Simpson 1/3 composto (Fig. 7-15), divide-se o intervalo  $[a, b]$  em  $N$  subintervalos. Em geral, os subintervalos podem ter largura arbitrária. A dedução aqui apresentada limita-se contudo ao caso no qual os subintervalos têm uma mesma largura  $h$ , onde  $h = (b - a)/N$ . Como são necessários três pontos para definir um polinômio quadrático, o método de Simpson 1/3 é aplicado em dois subintervalos adjacentes de cada vez (os dois primeiros, o terceiro e o quarto juntos, etc.). Conseqüentemente, o intervalo total deve ser dividido em um **número par** de subintervalos.

A integral ao longo de todo o intervalo pode ser escrita como a soma das integrais ao longo de pares de subintervalos adjacentes.

$$\begin{aligned} I(f) &= \int_a^b f(x)dx = \overbrace{\int_{x_1=a}^{x_3} f(x)dx}^{I_2} + \overbrace{\int_{x_3}^{x_5} f(x)dx}^{I_4} + \dots + \overbrace{\int_{x_{i-1}}^{x_{i+1}} f(x)dx}^{I_i} + \dots \\ &\quad + \underbrace{\int_{x_{N-1}}^{x_N} f(x)dx}_{I_N} = \sum_{i=2, 4, 6}^N \int_{x_{i-1}}^{x_{i+1}} f(x)dx \end{aligned} \quad (7.17)$$



**Figura 7-15** Método de Simpson 1/3 composto.

Usando a Eq. (7.16), pode-se escrever a integral ao longo de dois intervalos adjacentes  $[x_{i-1}, x_i]$  e  $[x_i, x_{i+1}]$  em termos do método Simpson 1/3 como:

$$I_i(f) = \int_{x_{i-1}}^{x_{i+1}} f(x) dx \approx \frac{h}{3} [f(x_{i-1}) + 4f(x_i) + f(x_{i+1})] \quad (7.18)$$

onde  $h = x_{i+1} - x_i = x_i - x_{i-1}$ . Substituindo a Eq. (7.18) em cada uma das integrais na Eq. (7.17), obtém-se:

$$I(f) \approx \frac{h}{3} [f(a) + 4f(x_2) + f(x_3) + f(x_4) + 4f(x_5) + f(x_6) + f(x_7) + \dots + f(x_{N-1}) + 4f(x_N) + f(b)]$$

Fazendo o agrupamento dos termos similares, o lado direito da equação anterior pode ser simplificado para fornecer a equação geral do método de Simpson 1/3 composto para subintervalos igualmente espaçados:

$$I(f) \approx \frac{h}{3} \left[ f(a) + 4 \sum_{i=2,4,6}^N f(x_i) + 2 \sum_{j=3,5,7}^{N-1} f(x_j) + f(b) \right] \quad (7.19)$$

onde  $h = (b - a)/N$ .

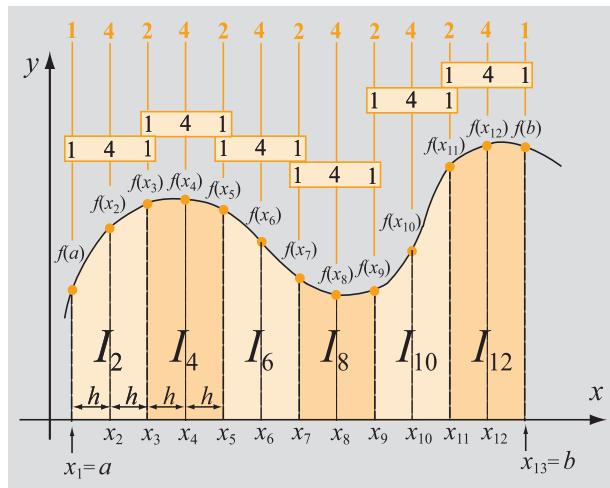
A Eq. (7.19) é a fórmula do **método de Simpson 1/3 composto** usada na integração numérica. É importante frisar que a Eq. (7.19) só pode ser usada se duas condições forem satisfeitas:

- Os subintervalos devem ser **igualmente espaçados**.
- O **número de subintervalos** no domínio  $[a, b]$  deve ser um número **par**.

A Eq. (7.19) é uma soma ponderada da função nos pontos que definem os subintervalos. O peso é 4 em todos os pontos  $x_i$  com índice par. Estes são os pontos centrais de cada par de subintervalos adjacentes (ver Eq. (7.18)). O peso é 2 em todos os pontos  $x_i$  com índices ímpares (exceto o primeiro e o último ponto).

Esses pontos estão na interface entre pares adjacentes de intervalos. Cada ponto é usado uma vez como o ponto final à direita de um par de subintervalos, e uma vez como o ponto final à esquerda do par de subintervalos seguinte. Os pontos finais são usados apenas uma vez. A Fig. 7-16 ilustra a soma ponderada de acordo com a Eq. (7.19) em um domínio  $[a, b]$  dividido em 12 subintervalos. A aplicação da Eq. (7.19) nessa ilustração fornece:

$$\begin{aligned} I(f) \approx & \frac{h}{3} \{ f(a) + 4[f(x_2) + f(x_4) + f(x_6) + f(x_8) + f(x_{10}) + f(x_{12})] \\ & + 2[f(x_3) + f(x_5) + f(x_7) + f(x_9) + f(x_{11})] + f(b) \} \end{aligned}$$



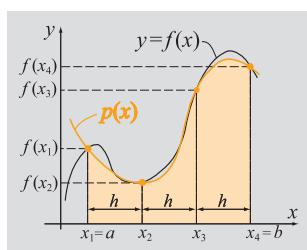
**Figura 7-16** Soma ponderada usando o método de Simpson 1/3 composto.

#### 7.4.2 Método de Simpson 3/8

Neste método, um polinômio cúbico (de terceira ordem) é usado para aproximar o integrando (Fig. 7-17). Um polinômio de terceira ordem pode ser determinado a partir de quatro pontos. Para uma integral ao longo do domínio  $[a, b]$ , são usados os pontos finais  $x_1 = a$  e  $x_4 = b$ , e dois pontos  $x_2$  e  $x_3$  que dividem o intervalo em três seções iguais. O polinômio pode ser escrito na forma:

$$p(x) = c_3x^3 + c_2x^2 + c_1x + c_0$$

onde  $c_3$ ,  $c_2$ ,  $c_1$  e  $c_0$  são constantes avaliadas a partir da condição que diz que o polinômio deve passar pelos pontos  $p(x_1) = f(x_1)$ ,  $p(x_2) = f(x_2)$ ,  $p(x_3) = f(x_3)$  e  $p(x_4) = f(x_4)$ . Uma vez determinadas as constantes, o polinômio pode ser facilmente integrado, resultando em:

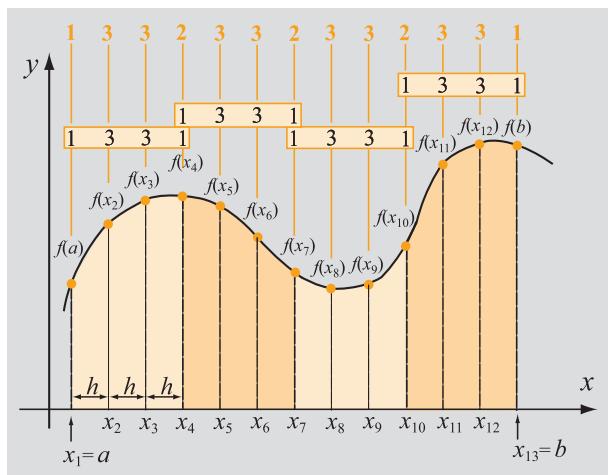


**Figura 7-17** Método de Simpson 3/8.

$$I = \int_a^b f(x)dx \approx \int_a^b p(x)dx = \frac{3}{8}h[f(a) + 3f(x_2) + 3f(x_3) + f(b)] \quad (7.20)$$

O valor da integral é mostrado na Fig. 7-17 como a área sombreada entre a curva  $p(x)$  e o eixo  $x$ . O termo  $3/8$  que aparece no nome do método vem do fator  $3/8$  presente na expressão da Eq. (7.20). Note que a Eq. (7.20) é uma soma ponderada dos valores de  $f(x)$  nos dois pontos finais  $x_1 = a$  e  $x_4 = b$ , e nos pontos  $x_2$  e  $x_3$  que dividem o intervalo em três seções iguais.

Como nos demais métodos, uma avaliação mais precisa da integral pode ser feita usando um método de Simpson 3/8 composto.



**Figura 7-18** Soma ponderada usando o método de Simpson 3/8 composto.

### Método de Simpson 3/8 composto

No método de Simpson 3/8 composto, divide-se o intervalo  $[a, b]$  em  $N$  subintervalos. Em geral, os subintervalos podem ter largura arbitrária. A dedução aqui apresentada, contudo, limita-se ao caso no qual os subintervalos têm uma mesma largura  $h$ , onde  $h = (b - a)/N$ . Como quatro pontos são necessários para definir um polinômio cúbico, o método Simpson 3/8 é aplicado em três subintervalos adjacentes de cada vez (os três primeiros, quarto, quinto e sexto juntos, etc.). Conseqüentemente, o intervalo total deve ser dividido em um número de subintervalos que seja divisível por 3.

A integral em cada grupo de três subintervalos adjacentes é avaliada usando a Eq. (7.20). A integral ao longo de todo o domínio é obtida com a soma das integrais em cada grupo de subintervalos. O processo é ilustrado na Fig. 7-18, na qual divide-se o domínio  $[a, b]$  em 12 subintervalos agrupados em quatro grupos de três. Usando a Eq. (7.20) para cada grupo e somando as quatro equações, obtém-se:

$$I(f) \approx \frac{3h}{8} \{ f(a) + 3[f(x_2) + f(x_3) + f(x_5) + f(x_6) + f(x_8) + f(x_9) \\ + f(x_{11}) + f(x_{12})] + 2[f(x_4) + f(x_7) + f(x_{10})] + f(b) \} \quad (7.21)$$

Para o caso geral que envolve a divisão do domínio  $[a, b]$  em  $N$  subintervalos (onde  $N$  é pelo menos 6 e divisível por 3), a Eq. (7.21) pode ser generalizada para:

$$I(f) \approx \frac{3h}{8} \left[ f(a) + 3 \sum_{i=2,5,8}^{N-1} [f(x_i) + f(x_{i+1})] + 2 \sum_{j=4,7,10}^{N-2} f(x_j) + f(b) \right] \quad (7.22)$$

A Eq. (7.22) corresponde ao *método de Simpson 3/8* usado na integração numérica. Este método pode ser usado se as duas condições seguintes forem satisfeitas:

- Os subintervalos devem ser *igualmente espaçados*.

- O número de subintervalos no domínio  $[a, b]$  deve ser divisível por 3.

Como o método de Simpson 1/3 só é válido para um número par de subintervalos e o método de Simpson 3/8 só é válido para um número de subintervalos divisível por 3, uma combinação de ambos pode ser usada para realizar a integração quando houver um número ímpar qualquer de subintervalos. Isso é feito usando o método de Simpson 3/8 nos três primeiros ( $[a, x_2]$ ,  $[x_2, x_3]$  e  $[x_3, x_4]$ ) ou nos três últimos subintervalos ( $[x_{N-2}, x_{N-1}]$ ,  $[x_{N-1}, x_N]$  e  $[x_N, x_b]$ ), aplicando-se o método de Simpson 1/3 no número restante (par) de subintervalos. Tal estratégia combinada funciona porque a ordem do erro numérico é a mesma em ambos os métodos (ver Seção 7.8).

## 7.5 QUADRATURA DE GAUSS

### Fundamentos

Em todos os métodos de integração apresentados até agora, a integral de  $f(x)$  ao longo do intervalo  $[a, b]$  foi avaliada representando  $f(x)$  como um polinômio de fácil integração. Dependendo do método de integração, o polinômio usado na aproximação de  $f(x)$  tem o mesmo valor em um (métodos retangular e do ponto central), dois (método trapezoidal), ou mais pontos (métodos de Simpson) dentro do intervalo. A integral é avaliada a partir do valor de  $f(x)$  nos pontos comuns usando o polinômio escolhido. Quando dois ou mais pontos são usados, o valor da integral é calculado a partir da soma ponderada dos valores de  $f(x)$  em pontos diferentes. A localização dos pontos comuns é predeterminada em cada um dos métodos de integração. Todos os métodos ilustrados até agora consideraram pontos igualmente espaçados. Esses métodos são resumidos na tabela a seguir.

Método de integração	Valores da função usada na avaliação da integral
Equação do Retângulo (7.2)	$f(a)$ ou $f(b)$ (qualquer um dos pontos finais)
Equação do Ponto Central (7.5)	$f((a + b)/2)$ (O ponto central)
Equação Trapezoidal (7.9)	$f(a)$ e $f(b)$ (Ambos os pontos finais)
Equação de Simpson 1/3 (7.16)	$f(a)$ , $f(b)$ e $f((a + b)/2)$ (Ambos os pontos finais e o ponto central)
Equação de Simpson 3/8 (7.20)	$f(a)$ , $f(b)$ , $f\left(a + \frac{1}{3}(a + b)\right)$ e $f\left(a + \frac{2}{3}(a + b)\right)$ (Ambos os pontos finais e dois pontos que dividem o intervalo em três subintervalos igualmente espaçados)

Na quadratura de Gauss, a integral também é avaliada usando a soma ponderada dos valores de  $f(x)$  em pontos distintos ao longo do intervalo  $[a, b]$  (chamados de pontos de Gauss). Os pontos de Gauss, contudo, não são igualmente

espaçados e não incluem os pontos finais. A localização dos pontos e os pesos correspondentes de  $f(x)$  são determinados de forma a minimizar o erro.

### Forma geral da quadratura de Gauss

A forma geral da quadratura de Gauss é:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n C_i f(x_i) \quad (7.23)$$

onde os coeficientes  $C_i$  são os pesos e  $x_i$  são os pontos de Gauss no intervalo  $[a, b]$ . Por exemplo, para  $n = 2$  e  $n = 3$ , a Eq. (7.23) tem a forma:

$$\int_a^b f(x)dx \approx C_1 f(x_1) + C_2 f(x_2), \quad \int_a^b f(x)dx \approx C_1 f(x_1) + C_2 f(x_2) + C_3 f(x_3)$$

O valor dos coeficientes  $C_i$  e a localização dos pontos  $x_i$  depende dos valores de  $n$ ,  $a$  e  $b$ , e são determinados de tal forma que o lado direito da Eq. (7.23) seja exatamente igual ao lado esquerdo para funções  $f(x)$  especificadas.

### Integração de $\int_{-1}^1 f(x)dx$ com a quadratura de Gauss

No domínio  $[-1, 1]$ , a forma da quadratura de Gauss é:

$$\int_{-1}^1 f(x)dx \approx \sum_{i=1}^n C_i f(x_i) \quad (7.24)$$

Os coeficientes  $C_i$  e a localização dos pontos de Gauss  $x_i$  são determinados fazendo com que a Eq. (7.24) seja exata quando  $f(x) = 1, x, x^2, x^3$ , etc. O número de casos a ser considerado depende do valor de  $n$ . Por exemplo, quando  $n = 2$ :

$$\int_{-1}^1 f(x)dx \approx C_1 f(x_1) + C_2 f(x_2) \quad (7.25)$$

As quatro constantes  $C_1$ ,  $C_2$ ,  $x_1$  e  $x_2$  são determinadas fazendo com que a Eq. (7.25) seja exata quando aplicada aos quatro casos a seguir:

$$\text{Caso 1: } f(x) = 1 \quad \int_{-1}^1 (1)dx = 2 = C_1 + C_2$$

$$\text{Caso 2: } f(x) = x \quad \int_{-1}^1 xdx = 0 = C_1 x_1 + C_2 x_2$$

$$\text{Caso 3: } f(x) = x^2 \quad \int_{-1}^1 x^2 dx = \frac{2}{3} = C_1 x_1^2 + C_2 x_2^2$$

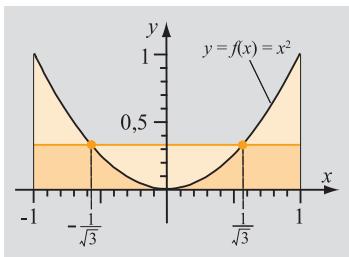
$$\text{Caso 4: } f(x) = x^3 \quad \int_{-1}^1 x^3 dx = 0 = C_1 x_1^3 + C_2 x_2^3$$

Os quatro casos fornecem um conjunto de quatro equações e quatro incógnitas. As equações são não-lineares, o que significa que podem existir múltiplas soluções. Uma solução particular pode ser obtida impondo-se um requisito adicional. Aqui, esse requisito é escolhido de tal forma que os pontos  $x_1$  e  $x_2$  estejam localizados simetricamente em relação a  $x = 0$  ( $x_1 = -x_2$ ). Da segunda equação, isso implica que  $C_1 = C_2$ . Com esses requisitos, a solução da equação resulta em:

$$C_1 = 1, \quad C_2 = 1, \quad x_1 = -\frac{1}{\sqrt{3}} = -0,57735027, \quad x_2 = \frac{1}{\sqrt{3}} = 0,57735027$$

A substituição das constantes de volta na Eq. (7.25) fornece, para  $n = 2$ :

$$\int_{-1}^1 f(x)dx \approx f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \quad (7.26)$$



**Figura 7-19** Integração de  $f(x) = x^2$  usando a quadratura de Gauss.

O lado direito da Eq. (7.26) fornece o valor exato da integral no lado esquerdo quando  $f(x) = 1$ ,  $f(x) = x$ ,  $f(x) = x^2$ ,  $f(x) = x^3$ . Isso é ilustrado na Fig. 7-19 para o caso em que  $f(x) = x^2$ . Nesse caso:

$$\int_{-1}^1 x^2 dx = \frac{2}{3} = 1\left(-\frac{1}{\sqrt{3}}\right)^2 + 1\left(\frac{1}{\sqrt{3}}\right)^2 \quad (7.27)$$

O valor da integral  $\int_{-1}^1 x^2 dx$  corresponde à área abaixo da curva  $f(x) = x^2$ . O lado direito da Eq. (7.27) é a área abaixo da linha laranja horizontal. As duas áreas são idênticas, já que a área entre  $f(x)$  e a linha laranja para  $|x| > 1/(\sqrt{3})$  é igual à área entre a linha laranja e  $f(x)$  para  $|x| < 1/(\sqrt{3})$  (as áreas sombreadas acima e abaixo da linha laranja têm a mesma área).

Quando  $f(x)$  é uma função diferente de  $f(x) = 1$ ,  $f(x) = x$ ,  $f(x) = x^2$  ou  $f(x) = x^3$ , ou de qualquer combinação linear dessas funções, a quadratura de Gauss fornece um valor aproximado para a integral. Por exemplo, se  $f(x) = \cos(x)$ , o valor exato da integral (o lado esquerdo da Eq. (7.26)) é:

$$\int_{-1}^1 \cos(x)dx = \left.\sin(x)\right|_{-1}^1 = \sin(1) - \sin(-1) = 1,68294197$$

O valor aproximado da integral, de acordo com a quadratura de Gauss (o lado direito da Eq. (7.26)), é:

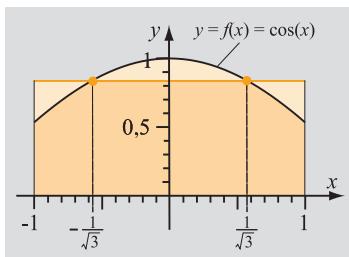
$$\cos\left(\frac{-1}{\sqrt{3}}\right) + \cos\left(\frac{1}{\sqrt{3}}\right) = 1,67582366$$

Esses resultados mostram que a quadratura de Gauss fornece uma aproximação muito boa para a integral, com um erro de 4,2%, mas não o valor exato. A última integração é ilustrada na Fig. 7-20, onde a integral exata corresponde à área abaixo da curva  $f(x) = \cos(x)$  e o valor aproximado da integral corresponde à área abaixo da linha vermelha. No caso ilustrado, as duas áreas não são idênticas. A área sombreada abaixo da linha laranja é um pouco menor (4,2%) do que a área sombreada acima da linha laranja.

A precisão da quadratura de Gauss pode ser aumentada com o uso de um  $n$  maior na Eq. (7.24). Para  $n = 3$ , a equação tem a forma:

$$\int_{-1}^1 f(x)dx \approx C_1 f(x_1) + C_2 f(x_2) + C_3 f(x_3) \quad (7.28)$$

Neste caso, há seis constantes:  $C_1$ ,  $C_2$ ,  $C_3$ ,  $x_1$ ,  $x_2$  e  $x_3$ . As constantes são determinadas fazendo com que a Eq. (7.28) seja exata quando  $f(x) = 1$ ,  $f(x) = x$ ,  $f(x) = x^2$ ,  $f(x) = x^3$ ,  $f(x) = x^4$  e  $f(x) = x^5$ . Isso fornece um conjunto de seis equações com seis



**Figura 7-20** Integração de  $f(x) = \cos(x)$  usando a quadratura de Gauss.

incógnitas (o processo de obtenção das incógnitas é igual àquele para  $n = 2$ ). As constantes determinadas são:

$$C_1 = 0,5555556, \quad C_2 = 0,8888889, \quad C_3 = 0,5555556$$

$$x_1 = -0,77459667, \quad x_2 = 0, \quad x_3 = 0,77459667$$

A equação da quadratura de Gauss quando  $n = 3$  é então:

$$\int_{-1}^1 f(x)dx \approx 0,5555556f(-0,77459667) + 0,8888889f(0) + 0,5555556f(0,77459667) \quad (7.29)$$

Como exemplo, estima-se novamente o valor da integral de  $f(x) = \cos(x)$ :

$$\int_{-1}^1 \cos(x)dx \approx 0,5555556 \cos(-0,77459667) + 0,8888889 \cos(0) + 0,5555556 \cos(0,77459667) = 1,68285982$$

Esse valor é quase idêntico ao valor exato calculado anteriormente.

A precisão da quadratura de Gauss pode ser aumentada ainda mais com o uso de valores maiores de  $n$ . A equação geral usada para estimar o valor de uma integral é:

$$\int_{-1}^1 f(x)dx \approx C_1 f(x_1) + C_2 f(x_2) + C_3 f(x_3) + \dots + C_n f(x_n) \quad (7.30)$$

A Tabela 7-1 lista os valores dos coeficientes  $C_i$  e a localização dos pontos de Gauss  $x_i$  para  $n = 2, 3, 4, 5$ , e  $6$ .

**Tabela 7-1** Coeficientes de peso e coordenadas dos pontos de Gauss

<b>n (Número de pontos)</b>	<b>Coeficientes <math>C_i</math> (pesos)</b>	<b>Pontos de Gauss <math>x_i</math></b>
2	$C_1 = 1$ $C_2 = 1$	$x_1 = -0,57735027$ $x_2 = 0,57735027$
3	$C_1 = 0,5555556$ $C_2 = 0,8888889$ $C_3 = 0,5555556$	$x_1 = -0,77459667$ $x_2 = 0$ $x_3 = 0,77459667$
4	$C_1 = 0,3478548$ $C_2 = 0,6521452$ $C_3 = 0,6521452$ $C_4 = 0,3478548$	$x_1 = -0,86113631$ $x_2 = -0,33998104$ $x_3 = 0,33998104$ $x_4 = 0,86113631$
5	$C_1 = 0,2369269$ $C_2 = 0,4786287$ $C_3 = 0,5688889$ $C_4 = 0,4786287$ $C_5 = 0,2369269$	$x_1 = -0,90617985$ $x_2 = -0,53846931$ $x_3 = 0$ $x_4 = 0,53846931$ $x_5 = 0,90617985$
6	$C_1 = 0,1713245$ $C_2 = 0,3607616$ $C_3 = 0,4679139$ $C_4 = 0,4679139$ $C_5 = 0,3607616$ $C_6 = 0,1713245$	$x_1 = -0,93246951$ $x_2 = -0,66120938$ $x_3 = -0,23861919$ $x_4 = 0,23861919$ $x_5 = 0,66120938$ $x_6 = 0,93246951$

## Integração de $\int_a^b f(x)dx$ com a quadratura de Gauss

Os coeficientes de peso e as coordenadas dos pontos de Gauss fornecidos na Tabela 7-1 são válidos apenas quando o intervalo de integração é  $[-1, 1]$ . Em geral, contudo, o intervalo pode ser qualquer domínio  $[a, b]$ . Ainda assim, pode-se usar a quadratura de Gauss com os coeficientes e pontos determinados para  $[-1, 1]$  no caso de um domínio genérico. Isso é feito com a transformação da integral  $\int_a^b f(x)dx$  em uma integral na forma  $\int_{-1}^1 f(t)dt$  por meio da mudança de variáveis:

$$x = \frac{1}{2}[t(b-a) + a + b] \quad \text{e} \quad dx = \frac{1}{2}(b-a)dt \quad (7.31)$$

A integração então tem a forma:

$$\int_a^b f(x)dx = \int_{-1}^1 f\left(\frac{(b-a)t+a+b}{2}\right) \frac{(b-a)}{2} dt$$

O Exemplo 7-2 mostra como usar essa transformação.

### Exemplo 7-2: Avaliação de uma integral definida usando a quadratura de Gauss de quarta ordem

Avalie  $\int_0^3 e^{-x^2} dx$  usando a quadratura de Gauss de quarta ordem.

#### SOLUÇÃO

**Passo 1** Como os limites de integração são  $[0, 3]$ , a integral tem que ser transformada para a forma  $\int_{-1}^1 f(t)dt$ . No presente problema,  $a = 0$  e  $b = 3$ . Substituindo esses valores na Eq. (7.31), obtém-se:  
 $x = \frac{1}{2}[t(b-a) + a + b] = \frac{1}{2}[t(3-0) + 0 + 3] = \frac{3}{2}(t+1)$  e  $dx = \frac{1}{2}(b-a)dt = \frac{1}{2}(3-0)dt = \frac{3}{2}dt$

A substituição desses valores na integral fornece:

$$I = \int_0^3 e^{-x^2} dx = \int_{-1}^1 f(t)dt = \int_{-1}^1 \frac{3}{2} e^{-\left[\frac{3}{2}(t+1)\right]^2} dt$$

**Passo 2** Use a quadratura de Gauss de quarta ordem para avaliar a integral. A partir da Eq. (7.30) e usando a Tabela 7-1:

$$I = \int_{-1}^1 f(t)dt \approx C_1 f(t_1) + C_2 f(t_2) + C_3 f(t_3) + C_4 f(t_4) = 0,3478548f(-0,86113631) + 0,6521452f(-0,33998104) + 0,6521452f(0,33998104) + 0,3478548f(0,86113631)$$

A avaliação de  $f(t) = \frac{3}{2} e^{-\left[\frac{3}{2}(t+1)\right]^2}$  fornece:

$$I = \left( 0,3478548 \frac{3}{2} e^{-\left[ \frac{3}{2}((-0,86113631) + 1) \right]^2} + 0,6521452 \frac{3}{2} e^{-\left[ \frac{3}{2}((-0,33998104) + 1) \right]^2} \right. \\ \left. + 0,6521452 \frac{3}{2} e^{-\left[ \frac{3}{2}((0,33998104) + 1) \right]^2} + 0,3478548 \frac{3}{2} e^{-\left[ \frac{3}{2}((0,86113631) + 1) \right]^2} \right) = 0,8841359$$

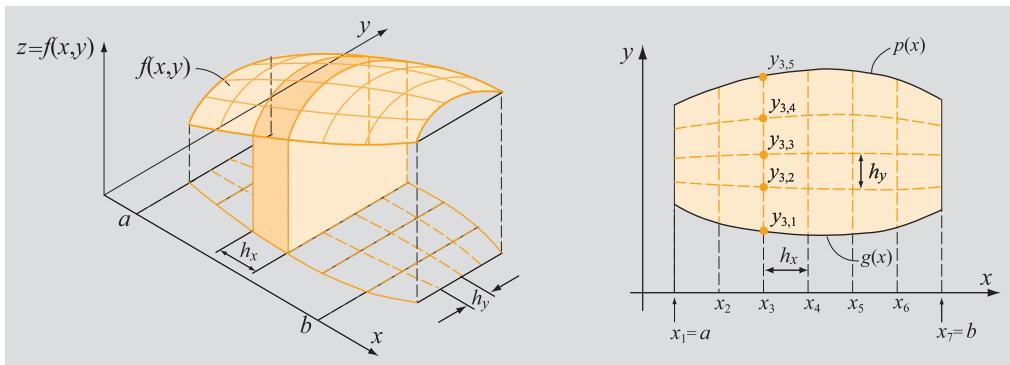
O valor exato da integral (quando realizada analiticamente) é 0,8862073. Dessa forma, o erro é pequeno, da ordem de 1%.

## 7.6 AVALIAÇÃO DE INTEGRAIS MÚLTIPLAS

Integrais duplas e triplas aparecem freqüentemente em problemas bidimensionais e tridimensionais. Uma integral dupla tem a forma:

$$I = \int_A f(x, y) dA = \int_a^b \left[ \int_{y=g(x)}^{y=p(x)} f(x, y) dy \right] dx \quad (7.32)$$

O integrando  $f(x, y)$  é função das variáveis independentes  $x$  e  $y$ . Os limites de integração da integral interna podem ser função de  $x$ , como na Eq. (7.32), ou constantes (quando eles forem constantes, a integração ocorre em uma região retangular). A Fig. 7-21 mostra esquematicamente a superfície da função  $f(x, y)$  e sua projeção no eixo  $x$ - $y$ . Nesta ilustração, o domínio  $[a, b]$  na direção  $x$  é dividido em seis subintervalos igualmente espaçados. Na direção  $y$ , o domínio  $[g(x), p(x)]$  é função de  $x$ , e em cada  $x$  a direção  $y$  é dividida em quatro subintervalos igualmente espaçados.



**Figura 7-21** Função e domínio para a integração dupla.

A integral dupla na Eq. (7.32) pode ser dividida em duas partes. A integral interna pode ser escrita como:

$$G(x) = \int_{y=g(x)}^{y=p(x)} f(x, y) dy \quad (7.33)$$

e a integral externa pode ser escrita tendo  $G(x)$  como integrando:

$$I = \int_a^b G(x) dx \quad (7.34)$$

Avalia-se a integral externa usando algum dos métodos numéricos descritos nas seções anteriores. Por exemplo, se o método de Simpson 1/3 for usado, a integral externa é avaliada com:

$$I(G) \approx \frac{h_x}{3} \{ G(a) + 4[G(x_2) + G(x_4) + G(x_6)] + 2[G(x_3) + G(x_5)] + f(b) \} \quad (7.35)$$

onde  $h_x = \frac{b-a}{6}$ . Cada um dos termos  $G$  é uma integral interna que deve ser integrada de acordo com a Eq. (7.33), usando o valor apropriado de  $x$ . Em geral, a integral  $G(x_i)$  pode ser escrita como:

$$G(x_i) = \int_{y=g(x_i)}^{y=p(x_i)} f(x_i, y) dy$$

para então ser integrada numericamente. Por exemplo, o uso do método de Simpson 1/3 na integração de  $G(x_3)$  resulta em:

$$\begin{aligned} G(x_3) &= \int_{y=g(x_3)}^{y=p(x_3)} f(x_3, y) dy \\ &= \frac{h_y}{3} \{ f(x_3, y_{3,1}) + 4[f(x_3, y_{3,2}) + f(x_3, y_{3,4})] + 2[f(x_3, y_{3,3})] + f(x_3, y_{3,5}) \} \end{aligned}$$

$$\text{onde } h_y = \frac{p(x_3) - g(x_3)}{4}.$$

Em geral, o domínio de integração pode ser dividido em qualquer número de subintervalos, e a integração pode ser feita com qualquer método numérico.

## 7.7 USO DE FUNÇÕES RESIDENTES DO MATLAB PARA A INTEGRAÇÃO

O MATLAB dispõe de várias funções residentes para realizar a integração numérica. Descreve-se a seguir o uso das funções `quad`, `quadl` e `trapz` para avaliar integrais simples, e da função `dblquad` para avaliar integrais duplas. Os comandos `quad`, `quadl` e `dblquad` são usados para integrar funções, enquanto a função `trapz` é usada para integrar dados tabulados.

### O comando `quad`

A forma do comando `quad` é:



- A função a ser integrada pode ser fornecida como uma *string*, como o nome de uma função escrita em arquivo, ou como o nome de uma função *inline*.
- A função  $f(x)$  deve ser escrita tendo como argumento um vetor  $x$  (use operações elemento por elemento), de forma a calcular o valor da função para cada elemento de  $x$ .
- O usuário deve se assegurar de que a função não apresente uma assíntota vertical (singularidade) entre  $a$  e  $b$ .
- quad calcula a integral com um erro absoluto menor que  $1,0 \times 10^{-6}$ . Esse número pode ser mudado com a inclusão do argumento opcional  $tol$ :

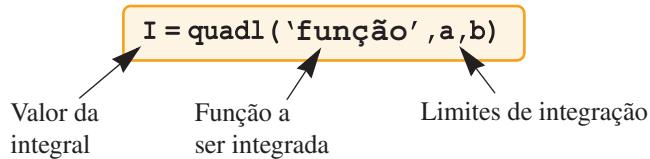
```
q = quad('função', a, b, tol)
```

$tol$  é um número que define o erro máximo. Com um maior valor de  $tol$ , a integral é calculada de forma menos precisa, porém mais rapidamente.

O comando quad usa um método de Simpson adaptativo para realizar a integração numérica. Métodos adaptativos são esquemas de integração que refinam o domínio de integração de forma seletiva, dependendo do comportamento do integrando. Se o integrando variar abruptamente na vizinhança de um ponto pertencente ao domínio de integração, então essa vizinhança é dividida em subintervalos menores.

### O comando quadl:

A forma do comando quadl (com L minúsculo na última letra) é exatamente a mesma do comando quad:



Todos os comentários feitos anteriormente para o comando quad continuam válidos para o comando quadl. A diferença entre os dois comandos está no método numérico usado no cálculo da integral. O comando quadl usa o método adaptativo de Lobatto.

### O comando trapz:

A função residente `trapz` pode ser usada para integrar uma função descrita por um conjunto de pontos discretos. Ela usa o método trapezoidal de integração numérica. A forma do comando é:

```
q = trapz(x, y)
```

onde  $x$  e  $y$  são vetores com as coordenadas  $x$  e  $y$  dos pontos, respectivamente. Os dois vetores devem ter o mesmo tamanho.

### O comando dblquad:

A função residente `dblquad` pode ser usada para avaliar a integral dupla de uma função. O formato do comando é o seguinte:

**I = dblquad('função', xmin, xmáx, ymin, ymáx)**

- A função a ser integrada pode ser fornecida como uma *string*, como o nome de uma função escrita em arquivo, ou como o nome de uma função *inline*.
- A função  $f(x)$  deve ser escrita tendo como argumento um vetor  $x$  (use operações elemento por elemento) e um escalar  $y$ .
- Os limites de integração são constantes.

No formato mostrado acima, a integração é feita usando a função quad e a tolerância padrão, que é de  $1,0 \times 10^{-6}$ . A tolerância pode ser mudada com a inclusão do argumento  $tol$ . O método de integração também pode ser mudado para quadl com a inclusão desse argumento no comando:

```
q = dblquad('função', xmin, xmáx, ymin, ymáx, tol, quadl)
```

## 7.8 ESTIMAÇÃO DE ERROS NA INTEGRAÇÃO NUMÉRICA

O erro é a diferença entre os valores obtidos com os cálculos numérico e exato da integral. Quando o integrando é um conjunto de pontos tabulados, não existe de fato um valor exato para a integral, e, como consequência, o erro não pode ser calculado ou mesmo estimado. Quando o integrando é uma função, pode-se obter o erro se o valor exato da integral puder ser determinado analiticamente. No entanto, se o valor da integral puder ser calculado analiticamente, não há por que calculá-lo numericamente. Uma situação comum ocorre quando a integral é uma expressão matemática cuja integral analítica é difícil ou impossível de resolver, sendo necessária a sua avaliação numérica. Nesse caso, o erro pode ser estimado com o emprego de algum método de integração numérica. Como ilustração, a estimativa do erro no método do retângulo é apresentada de forma mais detalhada.

No método do retângulo, a integral de  $f(x)$  ao longo do intervalo  $x \in [a, b]$  é calculada assumindo que  $f(x) = f(a)$  nesse intervalo:

$$I(f) = \int_a^b f(x)dx \approx \int_a^b f(a)dx = f(a)(b-a)$$

O erro  $E$  é então:

$$E = \int_a^b f(x)dx - f(a)(b-a) \quad (7.36)$$

Uma estimativa do erro pode ser obtida escrevendo-se a expansão em série de Taylor de  $f(x)$  com um termo e um resíduo na vizinhança do ponto  $x = a$ :

$$f(x) = f(a) + f'(\xi)(x-a) \quad (7.37)$$

onde  $\xi$  é um ponto entre  $a$  e  $b$ . Integrando ambos os lados da Eq. (7.37), obtém-se:

$$\int_a^b f(x) dx = \int_a^b [f(a) + f'(\xi)(x-a)] dx = f(a)(b-a) + \frac{1}{2} f'(\xi)(b-a)^2 \quad (7.38)$$

O erro expresso na Eq. (7.36) pode ser determinado usando a Eq. (7.38):

$$E = \int_a^b f(x) dx - f(a)(b-a) = \frac{1}{2} f'(\xi)(b-a)^2 \quad (7.39)$$

A Eq. (7.39) mostra que o erro depende de  $(b-a)$  e dos valores da derivada primeira de  $f(x)$  no intervalo  $[a, b]$ . Obviamente, o erro pode ser grande se o domínio for grande e/ou o valor das derivadas também for grande. O erro, no entanto, pode ser reduzido de forma significativa com o emprego do método do retângulo composto. Nesse caso, o domínio é dividido em subintervalos de largura  $h$ , onde  $h = x_{i+1} - x_i$ . A Eq. (7.39) pode ser usada para estimar o erro associado a um subintervalo:

$$E = \int_{x_i}^{x_{i+1}} f(x) dx - f(x_i)h = \frac{1}{2} f'(\xi_i) h^2 \quad (7.40)$$

onde  $\xi_i$  é um ponto entre  $x_i$  e  $x_{i+1}$ . Agora, a ordem de grandeza do erro pode ser controlada a partir do tamanho de  $h$ . Quando  $h$  for muito pequeno (muito menor que 1), o erro no subintervalo se torna muito pequeno. Para o intervalo completo  $[a, b]$ , uma estimativa do erro pode ser obtida a partir da soma dos erros de todos os subintervalos. Para o caso onde  $h$  é o mesmo em todos os subintervalos:

$$E = \frac{1}{2} h^2 \sum_{i=1}^N f'(\xi_i) \quad (7.41)$$

Se o valor médio da derivada  $\bar{f}'$  no intervalo  $[a, b]$  puder ser estimado usando:

$$\bar{f}' \approx \frac{\sum_{i=1}^N f'(\xi_i)}{N} \quad (7.42)$$

então a Eq. (7.41) pode ser simplificada usando a Eq. (7.42), lembrando que  $h = (b-a)/N$ :

$$E = \frac{(b-a)}{2} h \bar{f}' = O(h) \quad (7.43)$$

Essa equação fornece uma estimativa do erro no método do retângulo composto. O erro é proporcional a  $h$ , já que  $\frac{(b-a)}{2} \bar{f}'$  é uma constante. Ele é escrito como  $O(h)$ , que significa “da ordem de  $h$ ”.

Os erros nos métodos do ponto central, trapezoidal e de Simpson compostos podem ser estimados de forma similar. Os detalhes estão além do escopo deste livro, e os resultados são apresentados a seguir:

Método do ponto central composto:

$$E = \frac{(b-a)}{24} \bar{f}'' h^2 = O(h^2) \quad (7.44)$$

$$\text{Método trapezoidal composto: } E = -\frac{(b-a)}{12} \overline{f''} h^2 = O(h^2) \quad (7.45)$$

$$\text{Método de Simpson 1/3 composto: } E = -\frac{(b-a)}{180} \overline{f'''} h^4 = O(h^4) \quad (7.46)$$

$$\text{Método de Simpson 3/8 composto: } E = -\frac{(b-a)}{80} \overline{f'''} h^4 = O(h^4) \quad (7.47)$$

Note que, se o valor médio das derivadas nas Eqs. (7.43)-(7.47) puder ser limitado, então os limites para os erros também podem ser determinados. Infelizmente, a determinação de tais limites é difícil, e com isso o cálculo da ordem de grandeza exata do erro se torna difícil na prática.

## 7.9 EXTRAPOLAÇÃO DE RICHARDSON

A extrapolação de Richardson é um método usado para se obter uma estimativa mais precisa do valor de uma integral a partir de duas estimativas menos precisas dessa mesma integral. Por exemplo, duas estimativas calculadas com um erro de  $O(h^2)$  podem ser usadas no cálculo de uma estimativa com erro de  $O(h^4)$ . Esta seção começa com a dedução da fórmula de extrapolação de Richardson considerando duas estimativas iniciais calculadas com o método trapezoidal composto (erro  $O(h^2)$ ). Em seguida, deduz-se a fórmula de extrapolação de Richardson para se obter um erro de  $O(h^6)$  a partir de duas estimativas com erro de  $O(h^4)$ . Finalmente, uma fórmula geral é apresentada para a extrapolação de Richardson. Essa fórmula usa estimativas conhecidas de uma integral com um erro de ordem  $h^n$ , onde  $n$  é um número par, para calcular uma nova estimativa com erro de ordem  $h^{n+2}$ .

### Extrapolação de Richardson a partir de duas estimativas com erro de $O(h^2)$

Quando uma integral  $I(f)_h$  é avaliada numericamente com um erro de  $O(h^2)$ , o valor exato da integral  $I(f)$ , que é desconhecido, pode ser expresso como a soma de  $I(f)_h$  e o erro:

$$I(f) = I(f)_h + Ch^2 \quad (7.48)$$

onde  $C$  é uma constante. Por exemplo, se o método trapezoidal composto for usado para calcular  $I(f)_h$  (com erro dado pela Eq. (7.45)), então:

$$I(f) = I(f)_h - \frac{(b-a)}{12} \overline{f''} h^2 \quad (7.49)$$

Dois valores estimados  $I(f)_{h_1}$  e  $I(f)_{h_2}$  de uma integral podem ser calculados com o mesmo método, que nesse caso apresenta um erro de  $O(h^2)$ , usando um número diferente de subintervalos (em uma das estimativas,  $h = h_1$ ; na outra,  $h = h_2$ ). A substituição de cada uma das estimativas na Eq. (7.48) fornece:

$$I(f) = I(f)_{h_1} + Ch_1^2 \quad (7.50)$$

e

$$I(f) = I(f)_{h_2} + Ch_2^2 \quad (7.51)$$

Se for assumido que  $C$  é o mesmo em ambas as equações (o valor médio da derivada segunda  $\bar{f}''$  é independente do valor de  $h$ ), então as Eqs. (7.50) e (7.51) podem ser resolvidas para  $I(f)$  em termos de  $I(f)_{h_1}$  e  $I(f)_{h_2}$ . A solução fornece:

$$I(f) = \frac{I(f)_{h_1} - \left(\frac{h_1}{h_2}\right)^2 I(f)_{h_2}}{1 - \left(\frac{h_1}{h_2}\right)^2} \quad (7.52)$$

A Eq. (7.52) fornece uma nova estimativa para  $I(f)$ , que agora tem erro de  $O(h^4)$ , a partir de  $I(f)_{h_1}$  e  $I(f)_{h_2}$ , que têm erro de  $O(h^2)$ . A prova de que a Eq. (7.52) tem erro de  $O(h^4)$  está além do escopo deste livro<sup>1</sup>.

Um caso especial ocorre quando  $h_2 = \frac{1}{2}h_1$ . As duas estimativas para o valor da integral usadas na extrapolação são tais que a segunda estimativa tem o dobro dos subintervalos usados na primeira estimativa. Nesse caso, a Eq. (7.52) se reduz a:

$$I(f) = \frac{4I(f)_{h_2} - I(f)_{h_1}}{3} \quad (7.53)$$

### Extrapolação de Richardson a partir de duas estimativas com erro de $O(h^4)$

Quando uma integral  $I(f)_h$  é avaliada numericamente com um erro de  $O(h^4)$ , o valor exato da integral  $I(f)$ , que é desconhecido, pode ser expresso como a soma de  $I(f)_h$  e o erro:

$$I(f) = I(f)_h + Ch^4 \quad (7.54)$$

onde  $C$  é uma constante.

Dois valores estimados  $I(f)_{h_1}$  e  $I(f)_{h_2}$  de uma integral podem ser calculados com o mesmo método, que nesse caso apresenta um erro de  $O(h^4)$ , usando um número diferente de subintervalos (em uma das estimativas,  $h = h_1$ ; na outra,  $h = h_2$ ). A substituição de cada uma das estimativas na Eq. (7.54) fornece:

$$I(f) = I(f)_{h_1} + Ch_1^4 \quad (7.55)$$

e

$$I(f) = I(f)_{h_2} + Ch_2^4 \quad (7.56)$$

Se for assumido que  $C$  é o mesmo em ambas as equações (o valor médio da derivada quarta  $\bar{f}^{IV}$  é independente do valor de  $h$ ), então as Eqs. (7.55) e (7.56) podem ser resolvidas para  $I(f)$  em termos de  $I(f)_{h_1}$  e  $I(f)_{h_2}$ . A solução fornece:

$$I(f) = \frac{I(f)_{h_1} - \left(\frac{h_1}{h_2}\right)^4 I(f)_{h_2}}{1 - \left(\frac{h_1}{h_2}\right)^4} \quad (7.57)$$

<sup>1</sup> O leitor interessado deve consultar P. J. Davis e P. Rabinowitz, *Numerical Integration*, Blaisdell Publishing Company, Waltham, Massachusetts, 1967, p. 52-55.

A Eq. (7.57) fornece uma nova estimativa para  $I(f)$ , que agora tem erro de  $O(h^6)$ , a partir de  $I(f)_{h_1}$  e  $I(f)_{h_2}$ , que têm erro de  $O(h^4)$  (a prova está, novamente, além do escopo deste livro. O leitor interessado deve consultar o livro de Davis e Rabinowitz, 1967).

Um caso especial ocorre quando  $h_2 = \frac{1}{2}h_1$ . As duas estimativas do valor da integral usadas na extrapolação são tais que a segunda estimativa tem o dobro dos subintervalos usados na primeira estimativa. Nesse caso, a Eq. (7.57) se reduz a:

$$I(f) = \frac{16I(f)_{h_2} - I(f)_{h_1}}{15} \quad (7.58)$$

### Fórmula geral da extrapolação de Richardson

Uma fórmula geral de extrapolação pode ser deduzida para o caso no qual as duas estimativas iniciais do valor da integral têm um erro estimado de mesma ordem  $h^p$  e são obtidas de tal forma que o número de subintervalos de uma é o dobro do número de subintervalos da outra.

Se  $I_n$  é uma estimativa do valor da integral obtida com o uso de  $n$  subintervalos e  $I_{2n}$  é uma estimativa do valor da integral usando  $2n$  subintervalos, sendo o erro em ambas as estimativas de ordem  $h^p$ , então uma nova estimativa pode ser calculada para o valor da integral usando:

$$I = \frac{2^p I_{2n} - I_n}{2^p - 1} \quad (7.59)$$

A nova estimativa da integral tem um erro estimado de ordem  $h^{(p+2)}$ . A substituição de  $p = 2$  e  $p = 4$  na Eq. (7.59) resulta nas Eqs. (7.53) e (7.58), respectivamente. Da mesma maneira, a equação de extrapolação que fornece uma estimativa com erro  $O(h^8)$  a partir de duas estimativas com erro  $O(h^6)$  é obtida com a substituição de  $p = 6$  na Eq. (7.59):

$$I = \frac{2^6 I_{2n} - I_n}{2^6 - 1} = \frac{64}{63} I_{2n} - \frac{1}{63} I_n \quad (7.60)$$

## 7.10 INTEGRAÇÃO DE ROMBERG

A integração de Romberg é um esquema usado para melhorar a precisão do valor estimado de uma integral que emprega sucessivamente a fórmula de extrapolação de Richardson (ver Seção 7.9). O esquema usa uma série de estimativas iniciais da integral, que são calculadas com o método trapezoidal composto usando diferentes números de subintervalos. O esquema de integração de Romberg, ilustrado na Fig. 7-22, segue estes passos:

**Passo 1** O valor da integral é calculado usando o método trapezoidal composto várias vezes. Na primeira vez, o número de subintervalos é igual a  $n$ . Em cada cálculo subsequente, o número de subintervalos é dobrado. Os valores obtidos são listados na primeira coluna (esquerda) da Fig. 7-22. Na primeira linha, a integral  $I_{1,1}$  é calculada com o método trapezoidal usando  $n$  subintervalos. Na segunda linha,

$I_{2,1}$  é calculada usando  $2n$  subintervalos,  $I_{3,1}$  usando  $4n$  subintervalos, e assim por diante. O erro no cálculo das integrais apresentadas na primeira coluna é de  $O(h^2)$ .

**Passo 2** A fórmula de extrapolação de Richardson, Eq. (7.53), é usada para que melhores estimativas sejam obtidas para a integral a partir dos valores listados na primeira coluna da Fig. 7-22. Este é o primeiro nível da integração de Romberg. Os primeiros dois valores,  $I_{1,1}$  e  $I_{2,1}$ , fornecem a estimativa  $I_{1,2}$ :

$$I_{1,2} = \frac{4I_{2,1} - I_{1,1}}{3} \quad (7.61)$$

O segundo e o terceiro termos ( $I_{2,1}$  e  $I_{3,1}$ ) fornecem a estimativa  $I_{2,2}$ :

$$I_{2,2} = \frac{4I_{3,1} - I_{2,1}}{3} \quad (7.62)$$

e assim por diante. As novas estimativas são listadas na segunda coluna da Fig. 7-22. De acordo com a extrapolação de Richardson, elas apresentam um erro de  $O(h^4)$ .

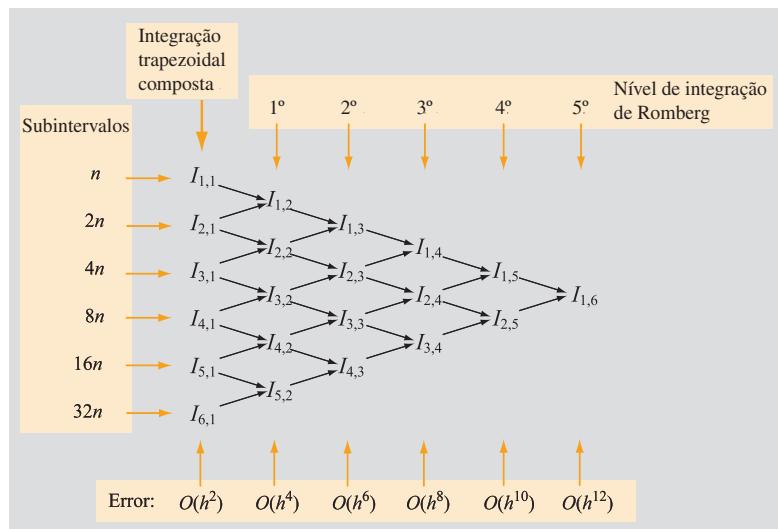
**Passo 3** A fórmula de extrapolação de Richardson, (Eq. 7.58), é usada para que melhores estimativas sejam obtidas para a integral a partir dos valores listados na segunda coluna da Fig. 7-22. Este é o segundo nível da integração de Romberg. Os primeiros dois valores,  $I_{1,2}$  e  $I_{2,2}$ , fornecem a estimativa  $I_{1,3}$ :

$$I_{1,3} = \frac{16I_{2,2} - I_{1,2}}{15} \quad (7.63)$$

O segundo e o terceiro termos ( $I_{2,2}$  e  $I_{3,2}$ ) fornecem a estimativa  $I_{2,3}$ :

$$I_{2,3} = \frac{16I_{3,2} - I_{2,2}}{15} \quad (7.64)$$

e assim por diante. As novas estimativas são listadas na terceira coluna da Fig. 7-22. De acordo com a extrapolação de Richardson, elas apresentam um erro de  $O(h^6)$ .



**Figura 7-22** Método de integração de Romberg.

**Passo 4 em diante** O processo de cálculo de estimativas mais precisas para o valor da integral continua, e cada nova coluna corresponde a um nível mais alto no método de integração de Romberg.

A equação para o cálculo dos valores extrapolados em cada nível a partir do nível anterior pode ser escrita de forma geral:

$$I_{i,j} = \frac{4^{j-1} I_{i+1,j-1} - I_{i,j-1}}{4^{j-1} - 1} \quad (7.65)$$

Os valores da primeira coluna,  $I_{1,1}$  a  $I_{k,1}$ , são calculados com o emprego do método trapezoidal composto. Em seguida, os valores extrapolados nas demais colunas são calculados usando a Eq. (7.65) para  $j = 2, 3, \dots, k$  em cada coluna  $i = 1, 2, \dots, (k-j+1)$ , onde  $k$  é o número de elementos da primeira coluna. O maior nível possível da integração de Romberg tem ordem  $k-1$ . O processo pode continuar até que haja apenas um termo na última coluna (o mais alto nível na integração de Romberg), ou quando as diferenças entre os valores estimados da integral se tornarem menores que uma tolerância predeterminada. O Exemplo 7-3 mostra a aplicação da integração de Romberg.

### Exemplo 7-3: Integração de Romberg e comparação com o método trapezoidal composto

Avalie  $\int_0^1 \frac{1}{1+x} dx$  usando três níveis da integração de Romberg. Use um passo inicial de  $h = 1$  (um subintervalo). Compare seu resultado com a resposta exata. Que número de subintervalos seria necessário para se obter o mesmo nível de precisão usando o método trapezoidal composto?

### SOLUÇÃO

**Resposta exata:** A resposta exata deste problema pode ser obtida analiticamente. A resposta é:  $\ln(2) = 0,69314718$

#### Integração de Romberg

Para realizar a integração numérica, cria-se uma função chamada Romberg, listada abaixo.

#### Programa 7-2 Arquivo de função. Integração de Romberg.

```
function IR = Romberg(integrand,a,b,Ni,Niveis)
% A função Romberg integra numericamente usando o método de integração de Romberg
% Variáveis de entrada:
% integrando Função a ser integrada, digitada como string.
% a Limite inferior de integração.
% b Limite superior de integração.
% Ni Número inicial de subintervalos.
% Niveis Número de níveis da integração de Romberg.
% Variável de saída:
% IR Matriz com os valores estimados da integral.
```

```
% Cria a primeira coluna com o método trapezoidal composto:
```

```
for i=1:Niveis+1
```

```
    Nsubinter=Ni*2^(i - 1);
```

```
    IR(i,1)=trapezoidal(integrand,a,b,Nsubinter);
```

```
end
```

```
% Calcula os valores extrapolados usando a Eq. (7.65):
```

```
for j=2:Niveis+1
```

```
    for i=1:(Niveis-j+2)
```

```
        IR(i,j)=(4^(j - 1)*IR(i + 1,j - 1) - IR(i,j - 1))/(4^(j - 1) - 1);
```

```
    end
```

```
end
```

Cria a primeira coluna da Fig. 7-22 usando a função **trapezoidal** (listada na Seção 7.3).

A função Romberg é em seguida usada na janela de comandos do MATLAB para determinar o valor da integral  $\int_0^1 \frac{1}{1+x} dx$ . O número inicial de subintervalos (para a primeira estimativa usando o método trapezoidal composto) é igual a 1.

```
>> format long
>> IntVal = Romberg('1/(1+x)',0,1,1,3)
>> format long
>> IntVal = Romberg('1/(1+x)',0,1,1,3)
IntVal =
    0.750000000000000  0.694444444444444  0.69317460317460  0.69314747764483
    0.708333333333333  0.69325396825397  0.69314790148123          0
    0.69702380952381  0.69315453065453          0          0
    0.69412185037185          0          0          0
```

↑  
Estimativas da  
integração trapezoidal  
composta.

↑  
1º nível de  
Romberg

↑  
2º nível de  
Romberg

↑  
3º nível de  
Romberg

Os resultados mostram que, com o método trapezoidal composto (primeira coluna), o valor mais preciso é obtido usando oito subintervalos, e que esse valor tem precisão de duas casas decimais. O primeiro nível da integração de Romberg (segunda coluna) aumenta a precisão para quatro casas decimais. O segundo nível da integração de Romberg (terceira coluna) aumenta a precisão para seis casas decimais. O resultado apresentado no terceiro nível da integração de Romberg (quarta coluna) também tem precisão de seis casas decimais, mas o valor é mais próximo da resposta exata.

O número de cálculos executados é igual a 10 (4 no método trapezoidal composto e 6 no procedimento de integração de Romberg). Para que uma estimativa com precisão de seis casas decimais seja obtida usando o método trapezoidal composto, é necessário aplicar este método com 276 subintervalos. Isso é mostrado a seguir, onde a função **trapezoidal** é usada na janela de comandos do MATLAB.

```
>> ITrap = trapezoidal('1/(1+x)',0,1,277)
```

```
ITrap =
```

```
0.69314799511374
```

## 7.11 INTEGRAIS IMPRÓPRIAS

Em todas as integrais  $\int_a^b f(x)dx$  consideradas até agora neste capítulo, os limites de integração  $a$  e  $b$  são finitos, e o integrando  $f(x)$  é finito e contínuo no domínio de integração. Há, no entanto, situações na ciência e na engenharia nas quais o integrando não é contínuo no intervalo de integração. Por exemplo, na estatística, a integral  $\int_{-\infty}^b \frac{1}{\sqrt{2\pi}} e^{\left(\frac{-x^2}{2}\right)} dx$  é usada para calcular a probabilidade cumulativa de uma grandeza ter um valor menor ou igual a  $b$ .

### 7.11.1 Integrais com singularidades

Uma integral  $\int_a^b f(x)dx$  tem uma singularidade quando há um ponto  $c$  no domínio  $a \leq c \leq b$  onde o valor do integrando  $f(c)$  não é definido ( $|f(x)| \rightarrow \infty$  quando  $x \rightarrow c$ ). Se a singularidade não estiver localizada em um dos pontos finais, a integral sempre pode ser escrita como a soma de duas integrais, uma ao longo de  $[a, c]$  e outra ao longo de  $[c, b]$ . Matematicamente, integrais que apresentam uma singularidade em um dos pontos finais podem ter ou não um valor finito. Por exemplo, a função  $1/(\sqrt{x})$  tem uma singularidade em  $x = 0$ , mas a integral dessa função ao longo de  $[0, 2]$  é igual a 2,  $\int_0^1 \frac{1}{\sqrt{x}} dx = 2$ . Por outro lado, a integral  $\int_0^1 \frac{1}{x} dx$  não tem um valor finito.

Numericamente, há várias maneiras de se calcular uma integral finita quando o integrando tem uma singularidade em um dos pontos finais do intervalo. Uma possibilidade é usar um método de integração aberto, no qual os pontos finais não são usados na determinação da integral. Dois métodos com essa característica, apresentados neste capítulo, são o método do ponto central composto (Seção 7.2) e a quadratura de Gauss (Seção 7.5). Outra possibilidade é usar um método numérico que considere o valor do integrando no ponto final, mas, ao invés de usar o ponto final, por exemplo,  $x = a$ , a integração começa de um ponto muito próximo ao ponto final,  $x = a + \varepsilon$ , onde  $\varepsilon \ll |a|$ .

Em alguns casos, também é possível eliminar uma singularidade analiticamente. Isso é feito usando uma mudança de variáveis ou uma transformação. Em seguida, a integral transformada pode ser integrada numericamente.

### 7.11.2 Integrais com limites infinitos

Integrais com um ou dois limites infinitos podem ter uma das formas a seguir:

$$I = \int_{-\infty}^b f(x)dx, \quad I = \int_a^{\infty} f(x)dx, \quad I = \int_{-\infty}^{\infty} f(x)dx \quad (7.66)$$

Em geral, integrais com limites infinitos podem resultar em valores finitos (elas convergem) ou infinitos (elas divergem). Quando a integral tem um valor finito, é possível realizar uma integração numérica. Tipicamente, o integrando

de uma integral como essas tem um valor finito apenas em um pequeno intervalo no domínio de integração, e um valor próximo a zero nos demais pontos. A integração numérica pode então ser realizada com a substituição do limite infinito por um limite finito no qual o valor do integrando é próximo a zero. Em seguida, qualquer um dos métodos descritos neste capítulo pode ser usado na integração numérica. A integração é feita sucessivamente, onde, em cada passo, o valor absoluto do limite é aumentado. Os cálculos param quando o valor do integrando não variar significativamente ao longo das interações sucessivas.

Em alguns casos, também é possível usar uma mudança de variáveis ou transformar a integral para uma forma que tenha limites finitos. Em seguida, a integral transformada pode ser integrada numericamente.

## 7.12 PROBLEMAS

### Problemas de solução manual

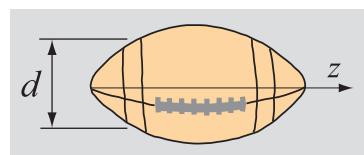
*Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos. Se usar o MATLAB, não utilize funções residentes para fazer a integração.*

**7.1** A função  $f(x)$  é dada na forma tabulada a seguir. Compare  $\int_0^1 f(x)dx$  com  $h = 0,25$  e  $h = 0,5$  usando:

- O método do retângulo composto.
- O método do ponto central composto. Use interpolação linear para determinar  $f(x)$  nos pontos centrais.
- O método trapezoidal composto.

$x$	0	0,25	0,5	0,75	1,0
$f(x)$	0,9162	0,8109	0,6931	0,5596	0,4055

**7.2** Para estimar a área superficial de uma bola de futebol americano, mede-se o seu diâmetro em diferentes pontos. A área superficial  $S$  e o volume  $V$  podem ser determinados usando:

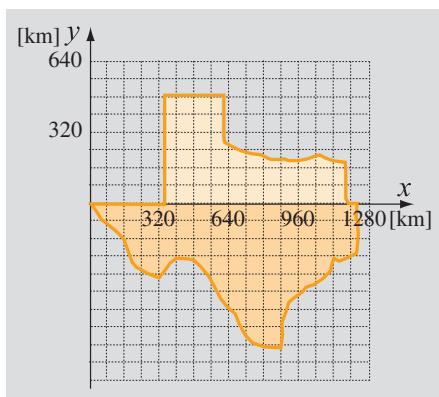


$$S = 2\pi \int_0^L r dz \quad \text{e} \quad V = \pi \int_0^L r^2 dz$$

Use o método de Simpson 1/3 composto com os dados abaixo para determinar o volume e a área superficial da bola.

$z(\text{cm})$	0	2,5	5,1	7,6	10,2	12,7	15,2	17,8	20,3	22,9	25,4	27,9	30,5
$d(\text{cm})$	0	6,6	8,1	12,2	14,2	15,2	15,7	15,2	14,2	12,2	8,1	6,6	0

**7.3** Um mapa aproximado do estado do Texas é mostrado na figura. Para determinar a área do estado, o mapa é dividido em duas partes (uma acima e outra abaixo do eixo  $x$ ). Determine a área do estado realizando a integração numérica das duas áreas. Em cada parte, faça uma lista da coordenada  $y$  na fronteira em função de  $x$ . Comece com  $x = 0$  e use incrementos de 80 km, de forma que o último ponto seja  $x = 1200$  km.



Assim que você tiver os dados tabulados, determine as integrais usando primeiro o método do ponto central composto e depois o método de Simpson 3/8 composto.

**7.4** Avalie as integrais:

$$I_1 = \int_0^{\pi} \sin^2 x dx \quad \text{e} \quad I_2 = \int_0^{2\pi} \sin^2 x dx$$

usando os métodos a seguir:

- (a) Método de Simpson 1/3. Divida o intervalo de integração em seis subintervalos.
- (b) Método de Simpson 3/8. Divida o intervalo de integração em seis subintervalos.
- (c) Quadratura de Gauss de segunda ordem.

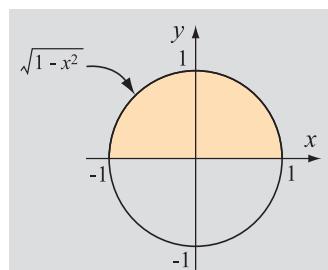
Compare os resultados e discuta o porquê das diferenças. O valor exato das integrais é  $I_1 = \pi/2$  e  $I_2 = \pi$ .

**7.5** A equação de um círculo com raio 1 (círculo unitário) é dada por  $x^2 + y^2 = 1$ , e sua área é  $A = \pi$ . Conseqüentemente:

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

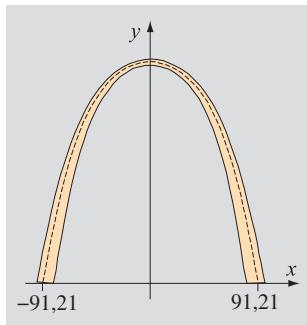
Avalie a integral usando os métodos a seguir:

- (a) Método de Simpson 1/3. Divida o intervalo de integração em oito subintervalos.
- (b) Método de Simpson 3/8. Divida o intervalo de integração em nove subintervalos.



- (c) Quadratura de Gauss de segunda ordem.

Compare os resultados e discuta o porquê das diferenças.



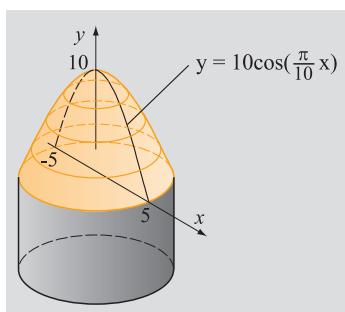
**7.6** A forma da linha centróide do Arco Gateway em Saint Louis pode ser modelada aproximadamente pela equação:

$$f(x) = 211,5 - 20,97\cosh(x/30,38)$$

para  $-91,21 \leq x \leq 91,21$

Usando a equação  $L = \int_a^b \sqrt{1 + [f'(x)]^2} dx$ , determine o comprimento do arco usando os métodos de integração a seguir:

- (a) Método de Simpson 1/3. Divida o intervalo de integração em oito subintervalos.
- (b) Método de Simpson 3/8. Divida o intervalo de integração em nove subintervalos.
- (c) Quadratura de Gauss de segunda ordem.



**7.7** O telhado de um silo é feito com a revolução da curva  $y = 10\cos\left(\frac{\pi}{10}x\right)$  de  $x = -5$  m a  $x = 5$  m em torno do eixo  $y$ , conforme mostrado na figura ao lado.

A área superficial  $S$  obtida com a revolução de uma curva  $f(x)$  no domínio de  $a$  a  $b$  em torno do eixo  $y$  pode ser calculada usando:

$$S = 2\pi \int_a^b x \sqrt{1 + [f'(x)]^2} dx$$

Calcule a área superficial do telhado usando os seguintes métodos de integração:

- (a) Método de Simpson 1/3. Divida o intervalo de integração em oito subintervalos.
- (b) Método de Simpson 3/8. Divida o intervalo de integração em nove subintervalos.
- (c) Quadratura de Gauss de segunda ordem.

**7.8** No método de Simpson 1/3 padrão (Eq. (7.16)), os pontos usados na integração são os pontos finais do intervalo,  $a$  e  $b$ , e o ponto central  $(a + b)/2$ . Deduza uma nova fórmula para o método de Simpson 1/3 no qual os pontos usados na integração sejam  $x = a$ ,  $x = b$  e  $x = (a + b)/3$ .

**7.9** Avalie a integral  $\int_{0.64}^1 \frac{1}{64}(2\cos\theta + 5\cos 3\theta)^2 \sin\theta d\theta$  usando o método de Simpson 3/8 e dividindo o intervalo de integração em nove subintervalos. Esta é a condição de ortogonalidade para o polinômio de Legendre de terceira ordem.

**7.10** Avalie a integral  $\int_0^1 225x(1-x^2)\sqrt{(1-x^2)^3} dx$  usando a integração de Romberg em dois níveis. Use  $n = 1$  como primeira estimativa para o método trapezoidal.

**7.11** Avalie a integral  $53,3904 \int_0^{10} (1 - e^{-0.18355x})$  usando a quadratura de Gauss de terceira ordem (com dois pontos).

**7.12** Avalie a integral

$$\int_0^{0.8} (0.2 + 25x - 200x^2 + 675x^3 - 900x^4 + 400x^5) dx$$

usando a quadratura de Gauss de terceira ordem (dois pontos).

**7.13** Mostre que o erro de truncamento associado ao método trapezoidal composto é de ordem  $h^2$ , onde  $h$  é o passo de integração (largura do subintervalo).

### Problemas de programação no MATLAB.

Resolva os seguintes problemas no ambiente MATLAB. Não use funções residentes para realizar as integrações.

**7.14** Escreva uma função no MATLAB que calcule a integral de uma função,  $I = \int_a^b y(x)dx$ , onde  $x$  e  $y(x)$  são dados como um conjunto de pontos tabulados, usando o método do ponto central. Use `I=IntegracaoPontoCentral(x,y)` como nome de função e argumentos. Os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos e  $I$  é o valor da integral. Use a função para resolver o Problema 7.9.

**7.15** Escreva uma função no MATLAB que calcule a integral de uma função usando o método trapezoidal composto. Use `I = Compzoidal('NomeFun', a, b)` como nome de função e argumentos. '`NomeFun`' é uma *string* contendo o nome da função que calcula o valor da expressão a ser integrada em função de  $x$ .  $a$  e  $b$  são os limites de integração, e  $I$  é o valor da integral. A função calcula o valor da integral em iterações sucessivas. Na primeira iteração, o intervalo  $[a, b]$  é dividido em dois subintervalos. Em cada iteração seguinte, o número de intervalos é dobrado. As iterações são interrompidas quando a diferença no valor da integral entre duas iterações sucessivas for menor que 1%. Use a função para resolver os Problemas 7.4 e 7.5.

**7.16** Escreva uma função no MATLAB que calcule a integral de uma função usando o método de Simpson 1/3. Use `I=Simpson13('NomeFun', a, b)` como nome de função e argumentos. '`NomeFun`' é uma *string* contendo o nome da função que calcula o valor da expressão a ser integrada em função de  $x$ .  $a$  e  $b$  são os limites de integração, e  $I$  é o valor da integral. A função calcula o valor da integral em iterações sucessivas. Na primeira iteração, o intervalo  $[a, b]$  é dividido em dois subintervalos. Em cada iteração seguinte, o número de intervalos é dobrado. As iterações são interrompidas quando a diferença no valor da integral entre duas iterações sucessivas for menor que 1%. Use a função para resolver os Problemas 7.4 e 7.5.

**7.17** Escreva uma função no MATLAB que calcule a integral de uma função usando o método de Simpson 3/8. Use  $I = \text{Simpson38}('NomeFun', a, b)$  como nome de função e argumentos. 'NomeFun' é uma *string* contendo o nome da função que calcula o valor da expressão a ser integrada em função de  $x$ .  $a$  e  $b$  são os limites de integração, e  $I$  é o valor da integral. A função calcula o valor da integral em iterações sucessivas. Na primeira iteração, o intervalo  $[a, b]$  é dividido em dois subintervalos. Em cada iteração seguinte, o número de intervalos é dobrado. As iterações são interrompidas quando a diferença no valor da integral entre duas iterações sucessivas for menor que 1%. Use a função para resolver os Problemas 7.4 e 7.5.

**7.18** Escreva uma função no MATLAB que calcule a integral de uma função  $f(x)$  no domínio  $[-1, 1]$  ( $\int_{-1}^1 f(x) dx$ ) usando a quadratura de Gauss de quarta ordem. Use  $I = \text{QuadGauss}('NomeFun')$  como nome de função e argumentos, onde 'NomeFun' é uma *string* contendo o nome da função que calcula o valor da expressão a ser integrada. Use a função para avaliar a integral do Problema 7.5.

**7.19** Escreva uma função no MATLAB que calcule a integral de uma função  $f(x)$  no domínio  $[a, b]$  ( $\int_a^b f(x) dx$ ) usando a quadratura de Gauss de quarta ordem. Use  $I = \text{QuadGauss4ab}('NomeFun', a, b)$  como nome de função e argumentos. 'NomeFun' é uma *string* contendo o nome da função que calcula o valor da expressão a ser integrada e  $a$  e  $b$  são os limites de integração. Use a função para avaliar a integral do Exemplo 7-2.

### Problemas na matemática, na ciência e na engenharia

Resolva os seguintes problemas no ambiente MATLAB. Use programas apresentados neste capítulo, programas desenvolvidos em problemas já resolvidos ou funções residentes do MATLAB.

**7.20** A densidade  $\rho$  da terra varia com o raio  $r$ . A tabela a seguir fornece a densidade aproximada em diferentes raios:

$r$ (km)	0	800	1200	1400	2000	3000	3400	3600	4000	5000	5500	6370
$\rho$ (kg/m <sup>3</sup> )	13000	12900	12700	12000	11650	10600	9900	5500	5300	4750	4500	3300

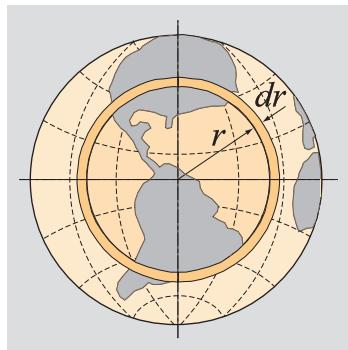
A massa da terra pode ser calculada por:

$$m = \int_0^{6370} \rho 4\pi r^2 dr$$

- (a) Escreva uma função no MATLAB que calcule a integral de dados tabulados com um espaçamento qualquer (o espaçamento entre os pontos não precisa ser o mesmo), usando o método trapezoidal composto. Use  $I = \text{TrapTabulados}(x, y)$  como nome de função e argumentos. Os argumentos de entrada  $x$  e  $y$  são vetores com as coordenadas dos pontos. Para realizar a integração, use a Eq. (7.12). Use essa função para calcular a massa da terra a partir dos dados da tabela.

- (b) Use a função `trapz` do MATLAB para calcular a massa da terra a partir dos dados da tabela.

- (c) Use a função `interp1` do MATLAB (com a opção `spline` como método de interpolação) para gerar um novo conjunto de dados interpolados a partir dos pontos da tabela. Divida o domínio  $[0, 6370]$  em 50 subintervalos iguais (use o comando `linspace`). Calcule a massa da terra integrando o conjunto de dados interpolados usando a função `trapz` do MATLAB.



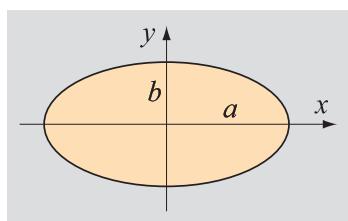
**7.21** O perímetro  $P$  de uma elipse é dado por:

$$P = 4a \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \theta} d\theta$$

$$\text{onde } k = \frac{\sqrt{a^2 - b^2}}{a}.$$

Escreva uma função no MATLAB que calcule o perímetro de uma elipse. Use `P=ElipsePer(a,b)` como nome de função e argumentos. Os argumentos de entrada  $a$  e  $b$  correspondem aos eixos maior e menor, respectivamente. Use essa função para calcular o perímetro das elipses a seguir:

$$(a) \frac{x^2}{5^2} + \frac{y^2}{2^2} = 1 \quad (b) \frac{x^2}{4^2} + \frac{y^2}{7^2} = 1$$

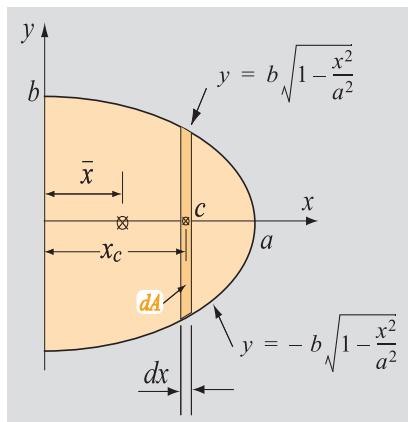


**7.22** Uma área tem a geometria de uma meia elipse, conforme mostrado na figura ao lado. A coordenada  $\bar{x}$  do centroide da área é dada por:

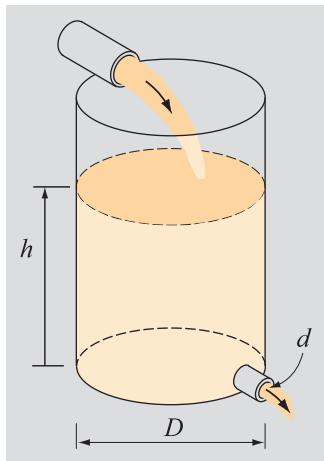
$$\bar{x} = \frac{M_y}{A}$$

onde  $A$  é a área dada por  $A = \frac{1}{2}\pi ab$  e  $M_y$  é o momento da área em torno do eixo  $y$ , dado por:

$$M_y = \int_A x_c dA = 2b \int_0^a x \sqrt{1 - \frac{x^2}{a^2}} dx$$



- (a) Escreva um programa no MATLAB que calcule  $\bar{x}$  quando  $a = 40$  mm e  $b = 15$  mm. Para realizar a integração, use a função `Compzoidal` criada no Problema 7.15.
- (b) Troque a função usada para realizar a integral na letra (a) por uma das funções residentes do MATLAB. Repita os cálculos e compare os resultados.



**7.23** Um tanque vertical cilíndrico com diâmetro  $D = 60$  cm tem um furo com diâmetro  $d = 5$  cm próximo a seu fundo. A água entra no tanque por meio de um cano em seu topo a uma taxa  $Q = 4,25 \times 10^{-3} \text{ m}^3/\text{s}$ . O tempo  $t$  necessário para encher o tanque de seu nível inicial  $h_0 = 2,7 \text{ m}$  ( $t = 0$ ) para o nível  $h$  é dado por:

$$t = \int_{h_0}^h \frac{1}{\left[ \frac{4Q}{\pi D^2} - \frac{d^2}{D^2} \sqrt{2gy} \right]} dy$$

onde  $g = 9,81 \text{ m/s}^2$ .

- (a) Use o método trapezoidal composto (função criada no Problema 7.15) para determinar quanto tempo levaria para o nível d'água atingir  $h = 1,5 \text{ m}$ .
- (b) Determine os tempos correspondentes a níveis d'água de  $h = 2,7 \text{ m}$  a  $h = 0,3 \text{ m}$  em incrementos de  $0,06 \text{ m}$  e trace o gráfico de  $h$  em função de  $t$ . Para realizar os cálculos e traçar o gráfico, escreva um programa em arquivo texto. Para realizar a integração, use funções residentes do MATLAB.

**7.24** O momento de inércia  $I_y$  em torno do eixo  $x$  da área da seção reta da meia elipse mostrada no Problema 7.22 é dado por:

$$I_y = \int_A x_c^2 dA = 2b \int_0^a x^2 \sqrt{1 - \frac{x^2}{a^2}} dx$$

Calcule  $I_y$  quando  $a = 40 \text{ mm}$  e  $b = 15 \text{ mm}$ . Para realizar a integração, use uma das funções residentes do MATLAB.

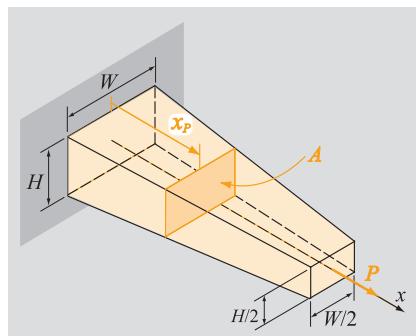
**7.25** O momento de inércia  $I_x$  em torno do eixo  $x$  da área da seção reta da meia elipse mostrada no Problema 7.22 é dado por:

$$I_x = \int_0^a \frac{1}{12} \left( 2b \sqrt{1 - \frac{x^2}{a^2}} \right)^3 dx$$

Calcule  $I_x$  quando  $a = 40 \text{ mm}$  e  $b = 15 \text{ mm}$ . Para realizar a integração, use uma das funções residentes do MATLAB.

**7.26** Uma viga de alumínio com comprimento  $L = 2 \text{ m}$  é presa em uma extremidade e carregada em sua outra extremidade por uma força axial  $P = 20 \text{ kN}$ . A área da seção reta da viga varia com seu comprimento e é dada em função de  $x$  como:

$$A = \frac{W}{2} \left( 2 - \frac{x}{L} \right) \frac{H}{2} \left( 2 - \frac{x}{L} \right)$$



onde  $W = 80$  mm e  $H = 30$  mm. A deformação normal  $\varepsilon$  ao longo da viga é dada por:

$$\varepsilon = \frac{du}{dx} = \frac{P}{EA}$$

onde  $E = 70$  GPa é o módulo elástico da viga.

O deslocamento  $\delta$  no ponto  $x = x_p$  ao longo da viga pode ser calculado por:

$$\delta = \int_0^{x_p} \frac{P}{EA} dx$$

- (a) Determine o deslocamento  $\delta$  no ponto final da viga ( $x_p = L$ ). Calcule-o duas vezes, uma vez usando a função criada no Problema 7.16 e depois usando uma das funções residentes do MATLAB.
- (b) Escreva um programa no MATLAB que calcule e trace o gráfico do deslocamento  $\delta$  em função de  $x$ . Para realizar a integração, use uma das funções residentes do MATLAB.

**7.27** No projeto de tubulações subterrâneas, é necessário estimar a temperatura do solo. A temperatura do solo pode ser estimada em várias profundidades a partir da modelagem do solo como um sólido semi-infinito com temperatura inicial constante. A temperatura em uma profundidade  $x$  no tempo  $t$  pode ser calculada pela expressão:

$$\frac{T(x, t) - T_s}{T_i - T_s} = \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right) = \frac{2}{\sqrt{\pi}} \int_0^{\frac{x}{2\sqrt{\alpha t}}} e^{-u^2} du$$

onde  $T_s$  é a temperatura na superfície,  $T_i$  é a temperatura inicial do solo e  $\alpha = 0,138 \times 10^{-6}$  m<sup>2</sup>/s é a difusividade térmica do solo. Responda as seguintes questões assumindo  $T_s = -15^\circ\text{C}$  e  $T_i = 12^\circ\text{C}$ .

- (a) Determine a temperatura em uma profundidade de  $x = 1$  m após 30 dias ( $t = 2,592 \times 10^6$  s).
- (b) Escreva um programa no MATLAB que trace um gráfico da temperatura em função do tempo em uma profundidade de  $x = 0,5$  m em 40 dias. Use incrementos de 1 dia.
- (c) Escreva um programa no MATLAB que trace um gráfico tridimensional ( $T$  versus  $x$  e  $t$ ) mostrando a temperatura em função da profundidade e do tempo para  $0 \leq x \leq 3$  m e  $0 \leq t \leq 2,592 \times 10^7$  s.

**7.28** Um termoacoplador é usado para medir a temperatura de um gás que flui em um duto. A dependência temporal da temperatura da junção esférica do termoacoplador é dada pela integral implícita:

$$t = - \int_{T_i}^T \frac{\rho V C}{A_S [h(T - T_\infty) + \varepsilon \sigma_{SB} (T^4 - T_{viz}^4)]} dT$$

onde  $T$  é a temperatura na junção do termoacoplador no tempo  $t$ ,  $\rho$  é a densidade do material na junção,  $V$  é o volume da junção esférica,  $C$  é a

capacidade térmica da junção,  $A_s$  é a área superficial da junção,  $h$  é o coeficiente de transferência de calor por convecção,  $T_\infty$  é a temperatura do gás,  $\varepsilon$  é a emissividade do material na junção,  $\sigma_{SB}$  é a constante de Stefan-Boltzmann e  $T_{viz}$  é a temperatura do duto metálico.

Para  $T_i = 298$  K,  $\varepsilon = 0,9$ ,  $\rho = 8500$  kg/m<sup>3</sup>,  $C = 400$  J/kg/K,  $T_\infty = 473$  K,  $h = 400$  W/m<sup>2</sup>/K e  $T_{viz} = 673$  K, determine o tempo  $t$  necessário para que a temperatura na junção do termoacoplador atinja 490 K usando a integração de Romberg.

# Equações Diferenciais Ordinárias: Problemas de Valor Inicial

### Tópicos centrais

- Métodos de Euler (explícito, implícito e erros) (8.2)
- Método de Euler modificado (8.3)
- Método do ponto central (8.4)
- Métodos de Runge-Kutta (8.5)
- Métodos multipasso (8.6)
- Métodos preditor-corretor (8.7)
- Sistemas de EDOs de primeira ordem (8.8)
- Solução de problemas de valor inicial de ordem mais elevada (8.9)

Uso de funções residentes do MATLAB para resolver problemas de valor inicial (8.10)

### Tópicos complementares

- Erro de truncamento local no método de Runge-Kutta de segunda ordem (8.11)
- Largura do passo de integração para se obter a precisão desejada (8.12)
- Estabilidade (8.13)
- EDOs rígidas (8.14)

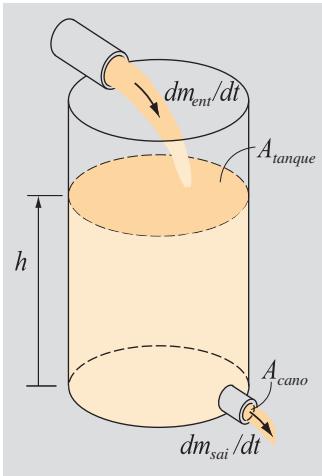
## 8.1 FUNDAMENTOS

Equação diferencial é o nome dado à equação que contém derivadas de uma função desconhecida. A solução dessa equação é a função que satisfaz à equação diferencial. Uma equação diferencial com apenas uma variável independente é chamada de **equação diferencial ordinária** (EDO). Uma EDO de primeira ordem envolve a derivada primeira da variável dependente em relação à variável independente. Por exemplo, se  $x$  é a variável independente e  $y$  é a variável dependente, a equação apresenta combinações das variáveis  $x$ ,  $y$  e  $\frac{dy}{dx}$ . Uma EDO de primeira ordem é linear se for uma função linear de  $y$  e  $\frac{dy}{dx}$  (ela pode ser uma função não-linear de  $x$ ). Exemplos de EDOs lineares e não-lineares são dadas nas Eqs. (8.1) e (8.2), respectivamente.

$$\frac{dy}{dx} + ax^2 + by = 0 \quad (\text{linear}) \quad (8.1)$$

$$\frac{dy}{dx} + ayx + b\sqrt{y} = 0 \quad (\text{não-linear}) \quad (8.2)$$

onde  $a$  e  $b$  são constantes.



**Figura 8-1** Tanque de água.

Equações diferenciais aparecem em todos os ramos da ciência e da engenharia. Elas também aparecem no ramo da economia, em aplicações financeiras e nas ciências sociais, sempre que idéias devam ser quantificadas e modelos de previsão sejam necessários. Equações diferenciais fornecem informações detalhadas a respeito das distribuições ou variações da variável dependente em função da variável independente.

Por exemplo, considere o tanque de água cilíndrico mostrado na Fig. 8-1. Enche-se o tanque por cima e a água sai por um cano conectado no fundo. O fluxo de água entrando no tanque varia com o tempo e é dado pela equação:

$$\frac{dm_{ent}}{dt} = K_1 + K_2 \cos(Ct) \quad (8.3)$$

onde  $dm_{ent}/dt$  descreve o fluxo de massa por unidade de tempo, e  $K_1$ ,  $K_2$  e  $C$  são constantes. A taxa de saída de água,  $dm_{sai}/dt$ , depende da altura  $h$  da água no interior do tanque (pressão) e é dada por:

$$\frac{dm_{sai}}{dt} = \rho A_{cano} \sqrt{2gh} \quad (8.4)$$

onde  $\rho$  é a densidade da água,  $A_{cano}$  é a área da seção reta do cano de saída e  $g$  é a aceleração da gravidade. A taxa de variação temporal da massa no interior do tanque,  $\frac{dm}{dt}$ , é a diferença entre a taxa de massa entrando no tanque e a taxa de massa saindo do tanque por unidade de tempo (conservação de massa):

$$\frac{dm}{dt} = \frac{dm_{ent}}{dt} - \frac{dm_{sai}}{dt} \quad (8.5)$$

A massa de água no interior do tanque pode ser expressa em termos da altura  $h$  e da área da seção reta do tanque,  $A_{tanque}$ , usando  $m = \rho A_{tanque} h$ , o que significa que  $\frac{dm}{dt} = \rho A_{tanque} \frac{dh}{dt}$ . Substituindo essa relação e as expressões das Eqs. (8.3) e (8.4) na Eq. (8.5), obtém-se uma equação que descreve a taxa de variação da altura  $h$ :

$$\rho A_{tanque} \frac{dh}{dt} = K_1 + K_2 \cos(Ct) - \rho A_{cano} \sqrt{2gh} \quad (8.6)$$

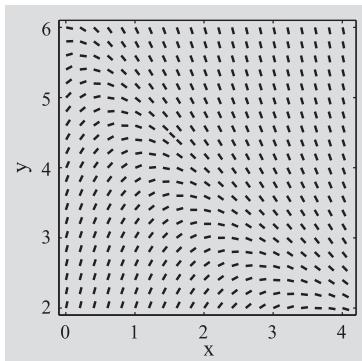
A Eq. (8.6) é uma EDO de primeira ordem, onde  $t$  é a variável independente e  $h$  é a variável dependente. A solução da Eq. (8.6) é uma função  $h(t)$  capaz de satisfazer a essa equação (em geral, um número infinito de funções pode satisfazê-la). Para que uma solução específica seja obtida, uma EDO de primeira ordem deve ter uma condição ou restrição inicial que especifique o valor da variável dependente para um valor particular da variável independente. Conforme discutido nos Capítulos 1 e 2, um problema físico apropriadamente formulado resultando em uma EDO de primeira ordem necessita de uma restrição deste tipo. Como problemas como esse são tipicamente problemas que envolvem a variação temporal

(isto é, problemas nos quais o tempo  $t$  é a variável independente), a restrição é chamada de **condição inicial** e o problema é chamado de **problema de valor inicial (PVI)**. A restrição associada à Eq. (8.6) pode ser escrita como:

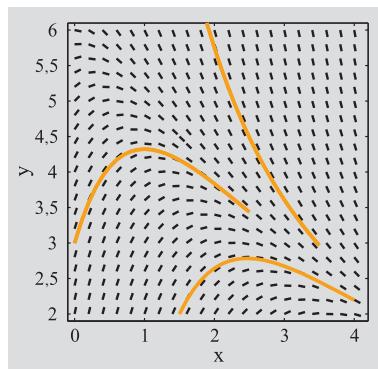
$$h = h_0 \quad \text{em} \quad t = t_0, \quad \text{ou} \quad h(t_0) = h_0$$

### Enunciado de um problema envolvendo uma EDO de primeira ordem

Uma EDO de primeira ordem tem a forma:



**Figura 8-2** Ilustração da função  $f(x, y)$ .



**Figura 8-3** Diferentes soluções para a mesma EDO.

obter uma solução analítica.

### Solução numérica de uma EDO de primeira ordem

A solução numérica de uma EDO como a Eq. (8.7) é formada por um conjunto de pontos discretos que representam a função  $y(x)$  de forma aproximada. Quando se resolve numericamente uma equação diferencial, o enunciado do problema também inclui o domínio da solução. Por exemplo, imagine que uma solução seja necessária entre  $x = a$  e  $x = b$  (o domínio é  $[a, b]$ ). Dependendo do método

$$\frac{dy}{dx} = f(x, y), \text{ com a condição inicial: } y(x_1) = y_1 \quad (8.7)$$

Na equação diferencial dada na Eq. (8.7), a função  $f(x, y)$  fornece a inclinação de  $y(x)$  em função de  $x$  e  $y$ . Por exemplo, considere a EDO a seguir:

$$\frac{dy}{dx} = f(x, y) = -1,2y + 7e^{-0,3x} \quad (8.8)$$

O valor de  $f(x, y)$  na Eq. (8.8) para  $0 \leq x \leq 4$  e  $2 \leq y \leq 6$  é ilustrado na Fig. 8-2, onde se traça a inclinação em muitos pontos no interior do domínio. As inclinações são tangentes à solução, o que significa que as inclinações se parecem com “linhas de fluxo” que mostram a direção seguida por  $y(x)$ . Fica claro a partir da Fig. 8-2 que, em geral, pode haver muitas (infinitas) soluções, já que é possível traçar diferentes linhas seguindo as “linhas de fluxo”. Três soluções possíveis são mostradas na Fig. 8-3. A solução de um problema específico é fixada pela condição inicial, que define o ponto onde  $y(x)$  começa, e pelo domínio de  $x$ , que especifica onde  $y(x)$  termina. As condições iniciais para as três soluções mostradas na Fig. 8-3 são  $y(0) = 3$ ,  $y(1,5) = 2$  e  $y(2) = 6$ .

### Solução analítica de uma EDO de primeira ordem

A solução analítica de uma EDO de primeira ordem é uma expressão matemática da função  $y(x)$  que satisfaz a equação diferencial e inclui o valor  $y(x_1) = y_1$ . Uma vez conhecida a função  $y(x)$ , pode-se calcular o seu valor em qualquer  $x$ . Conforme mencionado no Capítulo 2, existem técnicas relativamente simples que se aplicam à solução de EDOs de primeira ordem, mas, em muitas situações, não é possível

numérico usado para resolver a equação, pode-se ajustar previamente o número de pontos entre  $a$  e  $b$  nos quais deseja-se obter a solução, ou isso pode ser decidido pelo método. Por exemplo, o domínio pode ser dividido em  $N$  subintervalos de mesma largura definidos por  $N + 1$  valores da variável independente entre  $x_1 = a$  e  $x_{N+1} = b$ . A solução consiste em valores da variável dependente determinados para cada valor da variável independente. A solução é então formada por um conjunto de pontos  $(x_1, y_1), (x_2, y_2), \dots, (x_{N+1}, y_{N+1})$  que definem a função  $y(x)$ .

### Visão geral de métodos numéricos usados na solução de EDOs de primeira ordem

Chama-se solução numérica o procedimento empregado no cálculo de uma estimativa para a solução associada a um conjunto de pontos discretos. O processo de solução é incremental, o que significa que ele é determinado em passos. Ele começa no ponto no qual o valor inicial é fornecido. Em seguida, usando a solução conhecida no primeiro ponto, determina-se uma solução em um segundo ponto próximo. Depois, obtém-se uma solução em um terceiro ponto, e assim por diante. Há procedimentos que envolvem uma abordagem de **passo simples** e outros que consideram uma abordagem **multipasso**. Na abordagem de passo simples, a solução no ponto seguinte,  $x_{i+1}$ , é calculada a partir da solução conhecida no ponto atual,  $x_i$ . Na abordagem multipasso, a solução em  $x_{i+1}$  é calculada a partir das soluções conhecidas em vários pontos anteriores. A idéia é que o valor da função em vários pontos anteriores possa fornecer uma melhor estimativa para a tendência da solução. Dois tipos de métodos, **explícito** e **implícito**, também podem ser usados no cálculo da solução em cada passo. A diferença entre esses métodos está no procedimento usado na solução.

Métodos **explícitos** são aqueles que usam uma fórmula explícita para calcular o valor da variável dependente no próximo valor da variável independente. Em uma fórmula explícita, o lado direito da equação tem apenas grandezas conhecidas. Em outras palavras, o próximo valor desconhecido da variável dependente,  $y_{i+1}$ , é calculado a partir de uma expressão na forma:

$$y_{i+1} = F(x_i, x_{i+1}, y_i) \quad (8.9)$$

onde  $x_i$ ,  $y_i$  e  $x_{i+1}$  são grandezas conhecidas.

Nos métodos **implícitos**, a equação usada para calcular  $y_{i+1}$  a partir dos valores conhecidos  $x_i$ ,  $y_i$  e  $x_{i+1}$  tem a forma:

$$y_{i+1} = F(x_i, x_{i+1}, y_{i+1}) \quad (8.10)$$

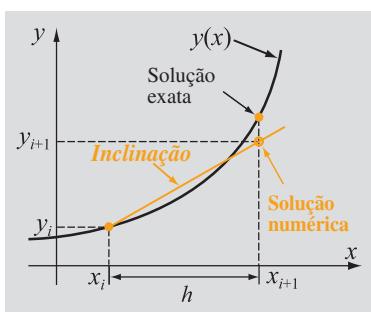
Aqui, a incógnita  $y_{i+1}$  aparece em ambos os lados da equação. Em geral, o lado direito da Eq. (8.10) é não-linear, e a equação deve ser resolvida numericamente usando os métodos descritos no Capítulo 3. Se a função no lado direito da Eq. (8.10) depender linearmente de  $y_{i+1}$ , ela constitui na realidade uma fórmula explícita como a Eq. (8.9) porque a Eq. (8.10) poderia ser rescrita e resolvida para  $y_{i+1}$ . Métodos implícitos são mais precisos que métodos explícitos, mas requerem uma maior esforço computacional em cada passo de integração.

## Erros na solução numérica de EDOs

Dois tipos de erro ocorrem quando EDOs são resolvidas numericamente: *erros de arredondamento* e *erros de truncamento*. Erros de arredondamento são causados pelo procedimento usado por computadores para realizar os cálculos (ver Capítulo 1). Erros de truncamento são causados pela natureza aproximada do método empregado na solução numérica. Como a solução numérica de uma equação diferencial é calculada em incrementos (passos), pode-se dividir o erro de truncamento em duas partes. Uma delas, chamada de *erro de truncamento local*, está associada à aplicação do método numérico em um único passo. A segunda parte, que se chama *erro de truncamento propagado, ou acumulado*, é causada pelo acúmulo dos erros de truncamento locais gerados nos passos anteriores. Juntas, as duas partes formam o *erro de truncamento global (total)* da solução. Mais detalhes a respeito de erros de truncamento são fornecidos nas seções seguintes, onde se analisa o erro associado aos vários métodos apresentados (ver Seções 8.2.2 e 8.11).

## Métodos explícitos de passo simples

Em um método explícito de passo simples, ilustrado na Figura 8-4, a solução numérica aproximada  $(x_{i+1}, y_{i+1})$  é calculada a partir da solução conhecida no ponto  $(x_i, y_i)$  usando:



**Figura 8-4** Métodos explícitos de passo simples.

$$x_{i+1} = x_i + h \quad (8.11)$$

$$y_{i+1} = y_i + \text{Inclinação} \cdot h \quad (8.12)$$

onde  $h$  é a largura do passo de integração e *Inclinação* é uma constante que estima o valor de  $\frac{dy}{dx}$  no intervalo de  $x_i$  a  $x_{i+1}$ . A solução numérica começa no ponto onde o valor inicial é conhecido. Isso corresponde a  $i = 1$  e ao ponto  $(x_1, y_1)$ . Em seguida,  $i$  se torna  $i = 2$ , e a solução no ponto seguinte,  $(x_2, y_2)$ , é calculada usando as Eqs. (8.11) e (8.12). O procedimento continua com  $i = 3$ , e assim por diante, até que os pontos cubram todo o domínio da solução.

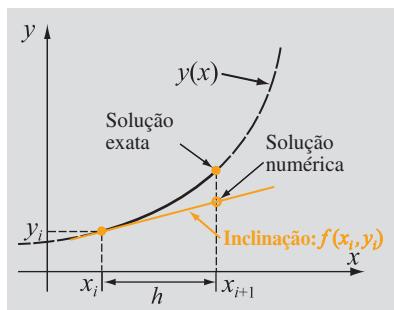
Muitos métodos explícitos de passo simples usam a forma das Eqs. (8.11) e (8.12). Vários deles são tratados nas Seções 8.2 a 8.6. A diferença entre os métodos está no valor usado na constante *Inclinação* que aparece na Eq. (8.12) e na forma pela qual ela é determinada. O método explícito mais simples é o método de Euler, descrito na Seção 8.2, no qual *Inclinação* é igual à inclinação de  $y(x)$  em  $(x_i, y_i)$ . No método de Euler modificado, descrito na Seção 8.3, o valor de *Inclinação* é uma média da inclinação de  $y(x)$  em  $(x_i, y_i)$  e do valor estimado da inclinação de  $y(x)$  no final do intervalo,  $(x_{i+1}, y_{i+1})$ . Em outro método, chamado de método do ponto central, descrito na Seção 8.4, o valor usado na constante *Inclinação* é obtido a partir de uma estimativa para a inclinação de  $y(x)$  no meio do intervalo (isto é, em  $(x_i + x_{i+1})/2$ ). Uma classe mais sofisticada de métodos, chamada de métodos de Runge-Kutta, é apresentada na Seção 8.5. Nesses métodos, o valor de *Inclinação* é calculado a partir da média ponderada de estimativas para a inclinação de  $y(x)$  em vários pontos no interior do intervalo.

## 8.2 MÉTODOS DE EULER

O método de Euler é a mais simples técnica de solução de uma EDO de primeira ordem na forma da Eq. (8.7):

$$\frac{dy}{dx} = f(x, y), \text{ com a condição inicial: } y(x_1) = y_1$$

O método pode ser formulado de forma explícita ou implícita. A formulação explícita mais comumente usada é apresentada em detalhes na seção a seguir. A formulação implícita é tratada na Seção 8.2.3.



**Figura 8-5** Método explícito de Euler.

### 8.2.1 Método explícito de Euler

O método explícito de Euler, também chamado de método de Euler progressivo, é uma técnica numérica de passo simples usada na solução de EDOs de primeira ordem. O método usa as Eqs. (8.11) e (8.12), onde o valor da constante *Inclinação* na Eq. (8.12) é a inclinação de  $y(x)$  no ponto  $(x_i, y_i)$ . Na realidade, essa inclinação é calculada a partir da própria equação diferencial:

$$\text{Inclinação} = \left. \frac{dy}{dx} \right|_{x=x_i} = f(x_i, y_i) \quad (8.13)$$

O método de Euler assume que, em uma pequena distância  $h$  na vizinhança de  $(x_i, y_i)$ , a função  $y(x)$  tem uma inclinação constante e igual à inclinação em  $(x_i, y_i)$ . A partir dessa hipótese, o próximo ponto da solução numérica  $(x_{i+1}, y_{i+1})$  é calculado usando:

$$x_{i+1} = x_i + h \quad (8.14)$$

$$y_{i+1} = y_i + f(x_i, y_i)h \quad (8.15)$$

O método explícito de Euler é ilustrado esquematicamente na Fig. 8-5. Para simplificar a ilustração, o passo de integração foi exagerado e a diferença (erro) entre as soluções numérica e exata em  $x_i$  foi ignorada. Uma ilustração mais precisa é apresentada na Seção 8.2.2. Fica óbvio, a partir da Fig. 8-5, que o erro neste método depende do valor de  $h$ ; quanto menor o valor de  $h$ , menor o erro. Uma discussão detalhada sobre o erro é apresentada mais à frente, na Seção 8.2.2.

A Eq. (8.15) do método de Euler pode ser deduzida de diferentes maneiras, começando com a equação diferencial fornecida:

$$\frac{dy}{dx} = f(x, y) \quad (8.16)$$

Uma solução aproximada para a Eq. (8.16) pode ser obtida com a integração numérica da equação ou com o uso de uma aproximação por diferenças finitas para a derivada.

### Dedução do método de Euler usando a integração numérica

A Eq. (8.16) pode ser escrita como um problema de integração após a multiplicação de ambos os seus lados por  $dx$ :

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x, y) dx \quad (8.17)$$

Realizando a integração no lado esquerdo e isolando  $y_{i+1}$ , obtém-se:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y) dx \quad (8.18)$$

O segundo termo no lado direito da equação é uma integral que precisa ser avaliada. O Capítulo 7 descreve vários métodos de integração numérica. O mais simples desses métodos é o método do retângulo (Seção 7.2.1), onde o integrando é representado de forma aproximada como um valor constante  $f(x_i, y_i)$ . Usando essa abordagem, a Eq. (8.18) se torna:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y) dx = y_i + f(x_i, y_i)(x_{i+1} - x_i) \quad (8.19)$$

que é igual à Eq. (8.15), já que  $h = (x_{i+1} - x_i)$ .

### Dedução do método de Euler usando diferenças finitas para aproximar a derivada

A fórmula de Euler apresentada na Eq. (8.15) também pode ser deduzida a partir da representação aproximada da derivada presente na equação diferencial. A derivada  $\frac{dy}{dx}$  na Eq. (8.16) pode ser aproximada pela fórmula de diferença progressiva (ver Seção 6.2) com a avaliação da EDO no ponto  $x = x_i$ :

$$\left. \frac{dy}{dx} \right|_{x_i} \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x_i, y_i) \quad (8.20)$$

A solução da Eq. (8.20) isolando  $y_{i+1}$  resulta na Eq. (8.15) do método de Euler (pelo fato de a equação poder ser deduzida dessa maneira, o método também é conhecido como o método de Euler progressivo).

O Exemplo 8-1 mostra a aplicação do método explícito de Euler na solução de EDOs de primeira ordem. A equação é resolvida numérica e analiticamente. Isso permite uma comparação entre as soluções numérica e exata, e também uma análise do efeito do passo de integração  $h$  no erro.

**Exemplo 8-1: Solução de uma EDO de primeira ordem usando o método explícito de Euler**

Use o método explícito de Euler para resolver a EDO  $\frac{dy}{dx} = -1,2y + 7e^{-0,3x}$  de  $x = 0$  a  $x = 2,5$  com a condição inicial  $y = 3$  em  $x = 0$ .

- Resolva o problema manualmente usando  $h = 0,5$ .
- Escreva um programa no MATLAB que resolva a equação usando  $h = 0,1$ .
- Use o programa da letra (b) para resolver a equação usando  $h = 0,01$ .

Em cada letra, compare os resultados com a solução exata (analítica):  $y = \frac{70}{9}e^{-0,3x} - \frac{43}{9}e^{-1,2x}$ .

## SOLUÇÃO

(a) **Solução manual:** O primeiro ponto da solução é  $(0,3)$ , que corresponde à condição inicial. No primeiro ponto,  $i = 1$ . Os valores de  $x$  e  $y$  são  $x_1 = 0$  e  $y_1 = 3$ .

O restante da solução é determinado usando as Eqs. (8.14) e (8.15). No presente problema, essas equações têm a forma:

$$x_{i+1} = x_i + h = x_i + 0,5 \quad (8.21)$$

$$y_{i+1} = y_i + f(x_i, y_i)h = y_i + (-1,2y_i + 7e^{-0,3x_i})0,5 \quad (8.22)$$

As Eqs. (8.21) e (8.22) são aplicadas cinco vezes, com  $i = 1, 2, 3, 4$ , e  $5$ .

**Primeiro passo** No primeiro passo,  $i = 1$ . As Eqs. (8.21) e (8.22) fornecem:

$$x_2 = x_1 + 0,5 = 0 + 0,5 = 0,5$$

$$y_2 = y_1 + (-1,2y_1 + 7e^{-0,3x_1})0,5 = 3 + (-1,2 \cdot 3 + 7e^{-0,3 \cdot 0})0,5 = 4,7$$

O segundo ponto é  $(0,5; 4,7)$ .

**Segundo passo** No segundo passo,  $i = 2$ . As Eqs. (8.21) e (8.22) fornecem:

$$x_3 = x_2 + 0,5 = 0,5 + 0,5 = 1,0$$

$$y_3 = y_2 + (-1,2y_2 + 7e^{-0,3x_2})0,5 = 4,7 + (-1,2 \cdot 4,7 + 7e^{-0,3 \cdot 1,0})0,5 = 4,893$$

O terceiro ponto é  $(1,0, 4,893)$ .

**Terceiro passo** No terceiro passo,  $i = 3$ . As Eqs. (8.21) e (8.22) fornecem:

$$x_4 = x_3 + 0,5 = 1,0 + 0,5 = 1,5$$

$$y_4 = y_3 + (-1,2y_3 + 7e^{-0,3x_3})0,5 = 4,893 + (-1,2 \cdot 4,893 + 7e^{-0,3 \cdot 1,5})0,5 = 4,550$$

O quarto ponto é  $(1,5, 4,550)$ .

**Quarto passo** No quarto passo,  $i = 4$ . As Eqs. (8.21) e (8.22) fornecem:

$$x_5 = x_4 + 0,5 = 1,5 + 0,5 = 2,0$$

$$y_5 = y_4 + (-1,2y_4 + 7e^{-0,3x_4})0,5 = 4,550 + (-1,2 \cdot 4,550 + 7e^{-0,3 \cdot 2,0})0,5 = 4,052$$

O quinto ponto é  $(2,0; 4,052)$ .

**Quinto passo** No quinto passo,  $i = 5$ . As Eqs. (8.21) e (8.22) fornecem:

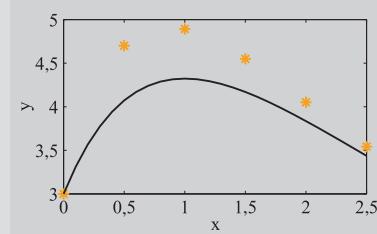
$$x_6 = x_5 + 0,5 = 2,0 + 0,5 = 2,5$$

$$y_6 = y_5 + (-1,2y_5 + 7e^{-0,3x_5})0,5 = 4,052 + (-1,2 \cdot 4,052 + 7e^{-0,3 \cdot 2,5})0,5 = 3,542$$

O sexto ponto é  $(2,5, 3,542)$ .

A figura ao lado mostra a solução numérica calculada (pontos) e a solução analítica exata (linha contínua).

Os valores das soluções exata e numérica, e o erro, que é a diferença entre as duas, são:



$i$	1	2	3	4	5	6
$x_i$	0,0	0,5	1,0	1,5	2,0	2,5
$y_i$ (numérica)	3,0	4,70	4,893	4,55	4,052	3,542
$y_i$ (exata)	3,0	4,072	4,323	4,170	3,835	3,436
Erro	0,0	-0,6277	-0,5696	0,3803	-0,2165	-0,1054

- (b) Para resolver no MATLAB a EDO dada no problema, escreve-se uma função chamada `edoEuler`. Essa função resolve problemas de valor inicial de primeira ordem usando o método explícito de Euler. Em seguida, escreve-se um programa que usa essa função para resolver a letra (b) do problema proposto. O programa também traça um gráfico que mostra as soluções numérica e exata.

**Programa 8-1 Função criada pelo usuário. Solução de uma EDO de primeira ordem usando o método explícito de Euler.**

```
function [x, y] = edoEULER(EDO,a,b,h,yINI)
% edoEULER resolve uma EDO de primeira ordem com valor inicial usando o
% método explícito de Euler.
% Variáveis de entrada:
% EDO Nome (string) de uma função em arquivo que calcula dy/dx.
% a Primeiro valor de x.
% b Último valor de x.
% h Passo de integração.
% yINI Valor da solução y no primeiro ponto (valor inicial).
% Variáveis de saída:
% x Vetor com a coordenada x dos pontos da solução.
% y Vetor com a coordenada y dos pontos da solução.
```

$x(1) = a; y(1) = yINI;$

Atribui o valor inicial a  $x(1)$  e  $y(1)$ .

$N = (b - a)/h;$

Determina o número de passos.

for  $i = 1:N$

$x(i + 1) = x(i) + h;$

Aplica a Eq. (8.14).

$y(i + 1) = y(i) + feval(EDO,x(i),y(i))*h;$

Aplica a Eq. (8.15).

end

O programa a seguir usa a função `edoEULER` para resolver a EDO usando o método explícito de Euler (letra (b) do problema). O programa também traça um gráfico com as soluções numérica e exata.

```

clear all
a = 0; b = 2.5; h = 0.1; yINI = 3;
[x, y] = edoEULER('Cap8Exmp1EDO',a,b,h,yINI);
xp = a:0.1:b;
yp=70/9*exp(-0.3*xp) - 43/9*exp(-1.2*xp);
plot(x,y,'--r',xp,yp)
xlabel('x'); ylabel('y')

```

Atribui o domínio, a largura do passo de integração e o valor inicial às variáveis.

Usa a função `edoEULER`.

Cria vetores para o traçado da solução exata.

A função `Cap8Exmp1EDO` (listada abaixo), digitada como o primeiro argumento da função `edoEULER`, calcula o valor de  $dy/dx$ .

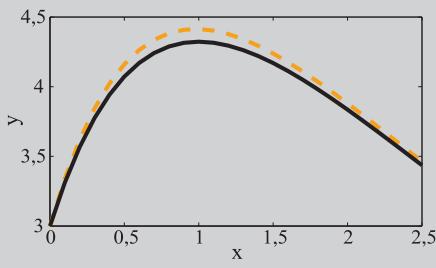
```

function dydx = Cap8Exmp1EDO(x,y)
dydx = -1.2*y + 7*exp(-0.3*x);

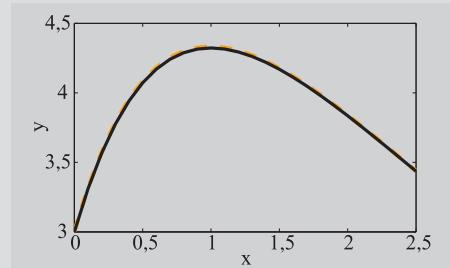
```

O gráfico criado quando o programa é executado é mostrado a seguir, com as soluções numérica (tracejada) e exata (contínua).

(c) O programa da letra (b) é executado novamente com  $h = 0,01$ . O resultado é mostrado no gráfico abaixo.



Letra (b)



Letra (c)

Conforme esperado, a solução numérica se torna mais precisa à medida que  $h$  diminui.

### 8.2.2 Análise do erro de truncamento no método explícito de Euler

Conforme mencionado na Seção 8.1, há duas fontes de erro quando EDOs são resolvidas numericamente: erros de arredondamento e erros de truncamento (ver também o Capítulo 1). Erros de arredondamento são causados pelo procedimento usado por computadores para realizar os cálculos. Erros de truncamento são

#### *Uma nota a respeito da notação usada nesta seção*

Na presente seção, são calculados diferentes valores para a variável dependente  $y$ , e esses valores são comparados em um dado valor da variável independente  $x$ . Para tornar a apresentação mais clara, identifica-se cada grandeza com um superscrito. Uma variável dependente calculada com o emprego de algum método numérico é escrita como  $y^{NS}$  (no restante do capítulo, ela é representada simplesmente pela letra  $y$ ). A solução exata da EDO é escrita como  $y^{TS}$ . Finalmente, a notação  $y^{Taylor}$  é usada para o valor calculado com uma expansão em série de Taylor.

causados pela natureza aproximada do método usado para calcular a solução em cada incremento (passo). Além disso, como a solução numérica de uma equação diferencial é calculada em incrementos (passos), o erro de truncamento é formado por um erro local e um erro propagado (ver Seção 8.1). Os erros de truncamento no método explícito de Euler são discutidos nesta seção. A discussão é dividida em duas partes. Primeiramente, analisa-se o erro de truncamento local, e em seguida os resultados são usados em uma tentativa de se estimar o erro de truncamento global.

### Erro de truncamento local

O erro de truncamento local é o erro inerente à fórmula usada na obtenção da solução numérica em um único passo (subintervalo). Ele é a diferença entre a solução numérica e a solução exata naquele subintervalo. Em um passo genérico  $i$ , a solução numérica  $y_i^{NS}$  em  $x = x_i$  é conhecida (calculada previamente), e a solução numérica  $y_{i+1}^{NS}$  em  $x = x_{i+1}$  é calculada com uma fórmula aproximada. O valor da solução exata neste passo pode ser expresso pela expansão em série de Taylor com dois termos e um resíduo (ver Capítulo 2):

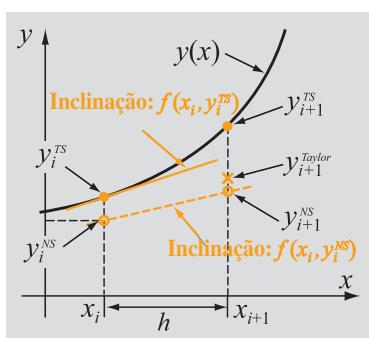
$$y_{i+1}^{Taylor} = y_i^{NS} + f(x_i, y_i^{NS})h + \frac{d^2y}{dx^2}\Big|_{x=\xi_i} \frac{h^2}{2} \quad (8.23)$$

onde  $f(x_i, y_i^{NS}) = \frac{dy}{dx}\Big|_{x=x_i}$  e  $\xi_i$  é um valor de  $x$  entre  $x_i$  e  $x_{i+1}$ . Com o método explícito de Euler, a solução numérica  $y_{i+1}^{NS}$  em  $x = x_{i+1}$  é calculada com:

$$y_{i+1}^{NS} = y_i^{NS} + f(x_i, y_i^{NS})h \quad (8.24)$$

Uma comparação entre as Eqs. (8.23) e (8.24) mostra que a fórmula explícita de Euler é uma aproximação que consiste nos dois primeiros termos da expansão na série de Taylor. A diferença entre ambas, ou seja, o erro de truncamento, ocorre porque o resíduo foi truncado. Esse erro, chamado de erro de truncamento local no passo  $i$ ,  $e_i^{TR}$ , é dado por:

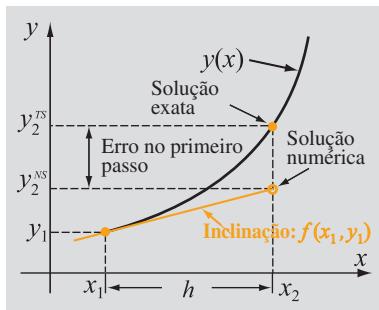
$$e_i^{TR} = y_{i+1}^{Taylor} - y_{i+1}^{NS} = \frac{d^2y}{dx^2}\Big|_{x=\xi_i} \frac{h^2}{2} = O(h^2) \quad (8.25)$$



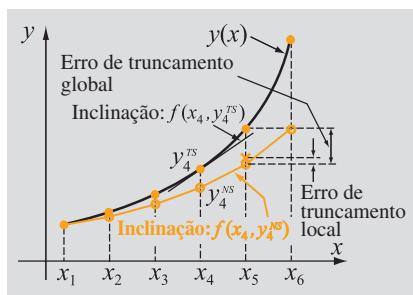
**Figura 8-6** Soluções numérica e exata.

Esse é o erro de truncamento inerente a cada passo de integração. Deve-se enfatizar, conforme ilustrado na Fig. 8-6, que o valor  $y_i^{NS}$  usado nas Eqs. (8.23) e (8.24) não corresponde à solução exata ( $y_i^{TS}$  é a solução da EDO em  $x = x_i$ ).  $y_i^{NS}$  é uma solução numérica obtida em  $x = x_i$  com a aplicação da fórmula de Euler em passos anteriores. Além disso,  $f(x_i, y_i^{NS})$  não corresponde à derivada (inclinação) da solução exata no ponto  $x = x_i$ , já que  $y_i^{NS} \neq y_i^{TS}$ . A diferença entre a solução numérica e a solução exata em  $x = x_{i+1}$  é o erro total. Parte desse erro é causada pelo erro de truncamento local, e o resto é devido ao acúmulo dos erros de truncamento associados aos passos anteriores (esta discussão exclui os erros

de arredondamento). O erro total devido somente ao erro de truncamento é chamado de **erro de truncamento global**. Uma estimativa para o erro de truncamento global associado ao método explícito de Euler é dada a seguir.



**Figura 8-7** Erro no primeiro passo.



**Figura 8-8** Erros de truncamento local e global.

### Erro de truncamento global

No primeiro subintervalo (passo),  $[x_1, x_2]$ , o erro de truncamento local é igual ao erro de truncamento global porque a solução exata no primeiro ponto é conhecida a partir da condição inicial. A partir do segundo subintervalo (passo),  $[x_2, x_3]$ , verifica-se um erro adicional em cada passo porque  $f(x_i, y_i^{\text{NS}})$  (a inclinação) na Eq. (8.24) não corresponde ao valor exato da derivada em  $x = x_i$  (o valor exato é  $f(x_i, y_i^{\text{TS}})$ ). Em outras palavras, o erro de truncamento se propaga, ou se acumula, tendo origem nos subintervalos anteriores ao considerado. O erro no primeiro passo é mostrado na Fig. 8-7, e o acúmulo do erro nos primeiros pontos é mostrado na Fig. 8-8. Em um ponto  $x_i$  qualquer, o erro de truncamento global é a diferença entre a solução exata,  $y_i^{\text{TS}}$ , e a solução numérica,  $y_i^{\text{NS}}$ :

$$E_i^{\text{TR}} = y_i^{\text{TS}} - y_i^{\text{NS}} \quad (8.26)$$

De forma similar, em  $x_{i+1}$ , o erro de truncamento global é:

$$E_{i+1}^{\text{TR}} = y_{i+1}^{\text{TS}} - y_{i+1}^{\text{NS}} \quad (8.27)$$

O valor da solução exata,  $y_{i+1}^{\text{TS}}$ , no passo  $i$  também pode ser expresso pela expansão em série de Taylor com dois termos e um resíduo:

$$y_{i+1}^{\text{TS}} = y_i^{\text{TS}} + f(x_i, y_i^{\text{TS}})h + \frac{d^2y}{dx^2}\Big|_{x=\eta_i} \frac{h^2}{2} \quad (8.28)$$

onde  $f(x_i, y_i^{\text{TS}}) = \frac{dy}{dx}\Big|_{x=x_i}$  e  $\eta_i$  é um valor de  $x$  entre  $x_i$  e  $x_{i+1}$ . Subtraindo as Eqs. (8.28) e (8.24), e usando as Eqs. (8.26) e (8.27), obtém-se:

$$E_{i+1}^{\text{TR}} = E_i^{\text{TR}} + [f(x_i, y_i^{\text{TS}}) - f(x_i, y_i^{\text{NS}})]h + \frac{d^2y}{dx^2}\Big|_{x=\eta_i} \frac{h^2}{2} \quad (8.29)$$

O teorema do valor médio para as derivadas (ver Capítulo 2) fornece:

$$[f(x_i, y_i^{\text{TS}}) - f(x_i, y_i^{\text{NS}})] = \frac{\partial f(x, y)}{\partial y}\Bigg|_{\substack{y=\gamma_i \\ x=x_i}} (y_i^{\text{TS}} - y_i^{\text{NS}}) = E_i^{\text{TR}} \frac{\partial f(x, y)}{\partial y}\Bigg|_{\substack{y=\gamma_i \\ x=x_i}} \quad (8.30)$$

onde  $\gamma_i$  é um valor de  $y$  entre  $y_i^{\text{TS}}$  e  $y_i^{\text{NS}}$ . A substituição da Eq. (8.30) na Eq. (8.29) resulta em:

$$E_{i+1}^{\text{TR}} = E_i^{\text{TR}}[1 + hf_y(x_i, \gamma_i)] + \alpha_i h^2 \quad (8.31)$$

onde  $f_y(x_i, \gamma_i) = \frac{\partial f(x, y)}{\partial y} \Big|_{\substack{y=\gamma_i \\ x=x_i}}$  e  $\alpha_i = \frac{1}{2} \frac{d^2 y}{dx^2} \Big|_{x=\eta_i}$ . Suponha agora que, no domínio

da solução, a derivada  $f_y(x_i, \gamma_i)$  seja limitada por um número positivo  $C$ . Então,  $|f_y(x_i, \gamma_i)| \leq C$  e a Eq. (8.31) pode ser escrita como:

$$E_{i+1}^{TR} \leq E_i^{TR}[1 + hC] + \alpha_i h^2 \quad (8.32)$$

A Eq. (8.32) pode ser usada para mostrar como o erro de truncamento se propaga (se acumula) à medida que a solução numérica avança. No primeiro ponto da solução, onde é dado o valor inicial,  $E_1^{TR} = 0$  (já que  $y_1^{TS} = y_1^{NS}$ ). Em seguida, a partir da Eq. (8.32), o erro de truncamento no segundo ponto é:

$$E_2^{TR} \leq \alpha_1 h^2 \quad (8.33)$$

No terceiro ponto:

$$E_3^{TR} \leq E_2^{TR}[1 + hC] + \alpha_2 h^2 = \alpha_1 h^2[1 + hC] + \alpha_2 h^2 = h^2 \{[1 + hC]\alpha_1 + \alpha_2\} \quad (8.34)$$

Da mesma forma, no quarto ponto:

$$E_4^{TR} \leq h^2 \{[1 + hC]^2 \alpha_1 + [1 + hC]\alpha_2 + \alpha_3\} \quad (8.35)$$

Em cada passo, um novo erro de truncamento local é adicionado, e os erros de truncamento dos passos anteriores são multiplicados (amplificados) por um fator  $[1 + hC]$ . No  $i$ -ésimo passo, a expressão do erro de truncamento global é:

$$E_i^{TR} \leq h^2 \{[1 + hC]^{i-2} \alpha_1 + [1 + hC]^{i-3} \alpha_2 + \dots + [1 + hC] \alpha_{i-2} + \alpha_{i-1}\} \quad (8.36)$$

Suponha agora que, no domínio da solução,  $\alpha_i$  seja limitado por um número positivo  $M$ . Então,  $|\alpha_i| \leq M$ , e a Eq. (8.36) pode ser escrita como:

$$E_i^{TR} \leq h^2 M \{1 + [1 + hC] + \dots + [1 + hC]^{i-3} + [1 + hC]^{i-2}\} \quad (8.37)$$

Se  $z$  for definido como  $z = [1 + hC]$ , então a expressão no interior das chaves {} na Eq. (8.37) se parece com a série:

$$1 + z + z^2 + \dots + z^m = \frac{z^{m+1} - 1}{z - 1} \quad (8.38)$$

Usando a Eq. (8.38), pode-se rescrever a Eq. (8.37) como:

$$E_i^{TR} \leq h^2 M \left\{ \frac{[1 + hC]^{i-1} - 1}{hC} \right\} = \frac{hM}{C} \{[1 + hC]^{i-1} - 1\} \quad (8.39)$$

É difícil determinar a ordem de grandeza de  $E^{TR}$  diretamente a partir da Eq. (8.39). É possível, contudo, determinar um limite considerando a expansão em série de Taylor da função exponencial  $e^s$  na vizinhança de  $s = 0$ :

$$e^s = 1 + s + \frac{s^2}{2} + \frac{s^3}{3!} + \dots \quad (8.40)$$

Substituindo  $s = hC$ , obtém-se:

$$e^{hC} = 1 + hC + \frac{(hC)^2}{2} + \frac{(hC)^3}{3!} + \dots \quad (8.41)$$

A Eq. (8.41) implica que  $1 + hC \leq e^{hC}$ , ou  $[1 + hC]^{i-1} \leq e^{hC(i-1)}$ . Usando esse resultado na Eq. (8.39), obtém-se:

$$E_i^{TR} \leq \frac{hM}{C} [e^{hC(i-1)} - 1] \quad (8.42)$$

que representa um limite para o erro de truncamento global no passo  $i$ . Se o processo de solução durar  $N$  passos, o limite do erro de truncamento global no último ponto  $x = x_{N+1}$  é:

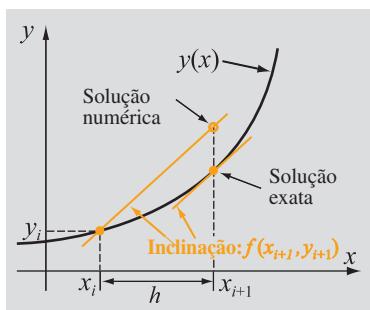
$$E_{N+1}^{TR} \leq \frac{hM}{C} [e^{hCN} - 1] = \frac{hM}{C} [e^{C(x_{N+1} - x_1)} - 1] \quad (8.43)$$

já que o número de subintervalos (passos) é  $N = (x_{N+1} - x_1)/h$ . A Eq. (8.43) mostra que  $E_{N+1}^{TR} \leq O(h)$ , e com isso o erro de truncamento global é da ordem de  $h$  mesmo que o erro de truncamento local seja da ordem de  $h^2$ .

Em resumo, o erro de truncamento local associado ao método explícito de Euler é de  $O(h^2)$ . O erro de truncamento global é de  $O(h)$ . O erro numérico total é a soma do erro de truncamento global e do erro de arredondamento. O erro de truncamento pode ser reduzido com o uso de um menor passo de integração  $h$ . Entretanto, se  $h$  se tornar pequeno ao ponto de os erros de arredondamento se tornarem significativos, o erro total pode crescer. Deve-se frisar que a Eq. (8.43) não pode ser usada para calcular um valor exato para o limite do erro de truncamento global porque os valores das constantes  $M$  e  $C$  não são conhecidos. A utilidade da Eq. (8.43) reside no fato de ela poder ser usada para comparar a precisão do método de Euler com a precisão de outros métodos.

Um exemplo numérico dos erros é apresentado na letra (a) do Exemplo 8-1. O passo de integração usado da solução tem tamanho  $h = 0,5$ , e a lista dos erros em cada passo mostra que eles são de fato da ordem de 0,5.

### 8.2.3 Método implícito de Euler



A forma do método implícito de Euler é igual à forma do método explícito, exceto pelo fato de, para uma pequena distância  $h$  na vizinhança de  $(x_i, y_i)$ , a inclinação da função  $y(x)$  ser tomada como uma constante igual à inclinação no ponto final do intervalo,  $(x_{i+1}, y_{i+1})$ . Isso é ilustrado esquematicamente na Fig. 8-9. A partir dessa hipótese, o ponto seguinte da solução numérica  $(x_{i+1}, y_{i+1})$  é calculado por:

$$x_{i+1} = x_i + h \quad (8.44)$$

$$y_{i+1} = y_i + f(x_{i+1}, y_{i+1})h \quad (8.45)$$

**Figura 8-9** Método implícito de Euler.

Agora, a incógnita  $y_{i+1}$  aparece em ambos os lados da Eq. (8.45), e, a menos que  $f(x_{i+1}, y_{i+1})$  dependa de  $y_{i+1}$  de acordo com uma relação linear ou quadrática simples, não é fácil ou mesmo possível isolar de forma explícita o termo  $y_{i+1}$ . Em geral, a Eq. (8.45) é uma equação não-linear em função da incógnita  $y_{i+1}$  que deve ser resolvida numericamente usando os métodos descritos no Capítulo 3.

A dedução da Eq. (8.45) é similar à dedução da Eq. (8.15) (ver Seção 8.2.1), exceto pelo fato de, na dedução que envolve a integração, o ponto final do intervalo ser usado no método do retângulo. Na dedução que parte da aproximação da derivada, utiliza-se a fórmula de diferença regressiva. Por esse motivo, o método implícito de Euler também é conhecido como método de Euler regressivo. Os erros de truncamento local e global são os mesmos verificados no método explícito (ver Seção 8.2.2). O Exemplo 8-3 mostra a solução de uma EDO usando o método implícito de Euler.

### Exemplo 8-2: Solução de uma EDO de primeira ordem usando o método implícito de Euler

Um componente químico apresenta um decaimento temporal quando exposto ao ar. Esse decaimento ocorre em uma taxa proporcional à sua concentração elevada a 3/2. Simultaneamente, o componente é produzido por outro processo. A equação diferencial que descreve a concentração instantânea do componente é:

$$\frac{dn(t)}{dt} = -0,8n^{3/2} + 10n_1(1 - e^{-3t}) \quad (8.46)$$

onde  $n(t)$  é a concentração instantânea e  $n_1 = 2000$  é a concentração inicial em  $t = 0$ . Resolva a equação diferencial para obter a concentração em função do tempo de  $t = 0$  a  $t = 0,5$  s usando o método implícito de Euler. Use também o método de Newton para determinar as raízes da equação não-linear resultante. Adote um passo de integração de  $h = 0,001$  s e trace um gráfico de  $n$  versus tempo.

### SOLUÇÃO

Neste problema,  $t$  é a variável independente e  $n$  é a variável dependente. A função  $f(t, n)$  é dada por:

$$f(t, n) = -0,8n^{3/2} + 10n_1(1 - e^{-3t}) \quad (8.47)$$

A solução numérica da equação diferencial é feita de forma incremental usando as Eqs. (8.44) e (8.45):

$$t_{i+1} = t_i + h \quad (8.48)$$

$$n_{i+1} = n_i + [-0,8n_{i+1}^{3/2} + 10n_1(1 - e^{-3t_{i+1}})]h \quad (8.49)$$

onde a função  $f(t, n)$  de acordo com a Eq. (8.47) foi substituída na Eq. (8.45).

Em cada passo da solução, deve-se obter  $n_{i+1}$  a partir da Eq. (8.49). Como essa equação não pode ser resolvida explicitamente, uma solução numérica é necessária. Para isso, a Eq. (8.49) é escrita na forma  $g(x) = 0$ , onde  $x = n_{i+1}$ :

$$g(x) = x + 0,8x^{3/2}h - 10n_1(1 - e^{-3t_{i+1}})h - n_i = 0 \quad (8.50)$$

A solução numérica da Eq. (8.50) usando o método de Newton (ver Seção 3.5) requer a derivada de  $g(x)$ , que é:

$$g'(x) = 1 + 0,8 \cdot \frac{3}{2} \cdot x^{1/2} \cdot h \quad (8.51)$$

A equação de iteração usada para resolver a Eq. (8.50) com o método de Newton é obtida com a substituição das Eqs. (8.50) e (8.51) na Eq. (3.14):

$$x_{j+1} = x_j - \frac{x_j + 0,8x_j^{3/2}h - 10n_1(1 - e^{-3t_{i+1}})h - n_i}{1 + 0,8 \cdot \frac{3}{2} \cdot x_j^{1/2} \cdot h} \quad (8.52)$$

O programa de MATLAB que resolve o problema é listado abaixo. O programa começa com o cálculo do número de passos  $N$  ( $N = (b - a)/h$ ) usados na solução. O programa tem um laço de repetição principal dedicado ao índice  $i$  (que varia de 1 a  $N$ ), no qual se calcula a solução no ponto  $i + 1$ . No interior desse laço de repetição, existe um outro laço dedicado ao índice  $j$ , no qual a solução da Eq. (8.50) é determinada em cada passo de integração usando o método de Newton. As iterações são realizadas usando a Eq. (8.52). O valor inicial de cada iteração é a solução (o valor de  $n$ ) no ponto anterior. As iterações são interrompidas quando o erro relativo estimado (Eq. (3.18)) é menor que 0,0001. O laço de repetição interno é limitado a 20 iterações.

**Programa 8-2 Arquivo texto. Solução de uma EDO de primeira ordem usando o método implícito de Euler.**

```
% Solução de uma EDO de primeira ordem usando o método implícito de Euler.
```

```
clear all
```

```
a = 0; b = 0.5; h = 0.002; Atribui o domínio às variáveis a e b, e a largura do passo de integração a h.
```

```
N = (b - a)/h; Calcula o número de passos.
```

```
n(1) = 2000; t(1) = a; Atribui a condição inicial ao primeiro ponto da solução.
```

```
for i=1:N
```

```
    t(i + 1) = t(i) + h; Calcula o próximo valor da variável independente, Eq. (8.48).
```

```
    x = n(i);
```

```
% Tem início o método de Newton. Inicia a solução da Eq. (8.50) usando o método de Newton.
```

```
    for j = 1:20
```

```
        num = x + 0.800*x^(3/2)*h - 10.0*n(1)*(1 - exp(-3*t(i + 1)))*h - n(i);
```

```
        denom = 1 + 0.800*1.5*x^(1/2)*h;
```

```
        xnew = x - num/denom; Eq. (8.52).
```

```
        if abs((xnew - x)/x) < 0.0001
```

```
            break
```

```
        else
```

```
            x = xnew;
```

```
        end
```

```
    end
```

```
    if j == 20
```

```
        fprintf('Não foi possível calcular a solução numérica em t = %g s', t(i))
```

```
        break
```

```
    end
```

```
% Termina o método de Newton.
```

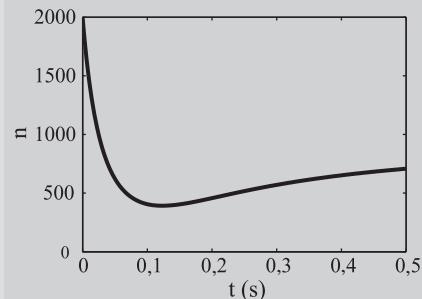
```
n(i + 1) = xnew; A solução obtida com o método de Newton é atribuída à solução da EDO no ponto seguinte.
```

```
end
```

```
plot(t,n)
```

```
axis([0 0.5 0 2000]), xlabel('t (s)'), ylabel('n')
```

O gráfico obtido após a execução do programa é mostrado ao lado.



### 8.3 MÉTODO DE EULER MODIFICADO

O método de Euler modificado é uma técnica numérica explícita de passo simples usada para resolver EDOs de primeira ordem. Ela é uma versão modificada do método explícito de Euler (esse método é às vezes chamado de método de Heun<sup>1</sup>). Conforme discutido na Seção 8.2.1, a principal hipótese adotada no método explícito de Euler assume que a derivada entre os pontos  $(x_i, y_i)$  e  $(x_{i+1}, y_{i+1})$  em cada subintervalo seja constante e igual à derivada de  $y(x)$  no ponto  $(x_i, y_i)$ . Essa hipótese é a principal fonte de erros nesse método. No método de Euler modificado, a inclinação utilizada no cálculo do valor de  $y_{i+1}$  é modificada para incluir o efeito de variação desse parâmetro ao longo do intervalo. A inclinação usada no método de Euler modificado é a média da inclinação no início do intervalo e de uma estimativa para a inclinação no final do intervalo. A inclinação no início do intervalo é:

$$\left. \frac{dy}{dx} \right|_{x=x_i} = f(x_i, y_i) \quad (8.53)$$

A estimativa para a inclinação no final do intervalo é determinada primeiramente com o cálculo de um valor aproximado para a variável  $y_{i+1}$ , escrita como  $y_{i+1}^{Eu}$ , usando o método explícito de Euler:

$$y_{i+1}^{Eu} = y_i + f(x_i, y_i)h \quad (8.54)$$

e depois com a estimativa para a inclinação no final do intervalo com a substituição do ponto  $(x_{i+1}, y_{i+1}^{Eu})$  na equação que envolve  $\frac{dy}{dx}$ :

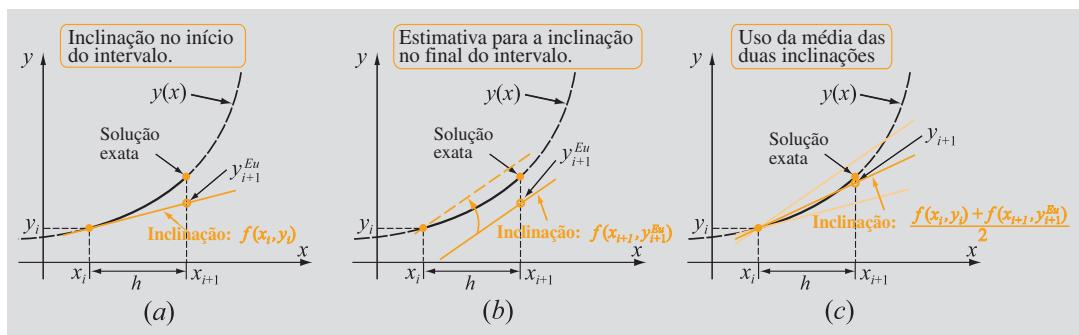
$$\left. \frac{dy}{dx} \right|_{\substack{y=y_{i+1}^{Eu} \\ x=x_{i+1}}} = f(x_{i+1}, y_{i+1}^{Eu}) \quad (8.55)$$

<sup>1</sup> Verifica-se uma inconsistência na literatura com relação ao uso do nome “método de Heun”. Às vezes, esse nome é usado com referência ao método apresentado nesta seção, mas às vezes ele se refere a outro método. Como o termo “método de Euler modificado (ou aperfeiçoado)” é freqüentemente associado ao método apresentado nesta seção, nós, assim como muitos outros autores, decidimos mantê-lo aqui. Um diferente método apresentado na Seção 8.5.1 é chamado de método de Heun.

Uma vez calculadas as duas inclinações, uma melhor estimativa para o valor de  $y_{i+1}$  é calculada a partir da média dessas inclinações:

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{Eu})}{2} h \quad (8.56)$$

O método de Euler modificado é ilustrado esquematicamente na Fig. 8-10. A Fig. 8-10a mostra a inclinação no início do intervalo, dada pela Eq. (8.53), e o valor de  $y_{i+1}^{Eu}$  calculado com a Eq. (8.54). A Fig. 8-10b mostra a inclinação estimada no final do intervalo, calculada com a Eq. (8.55), e a Fig. 8-10c mostra o valor de  $y_{i+1}$  obtido com a Eq. (8.56) a partir da média das duas inclinações. A Eq. (8.56) também pode ser deduzida com a integração da EDO ao longo do intervalo  $[x_i, x_{i+1}]$  usando o método trapezoidal.



**Figura 8-10** Método de Euler modificado.

O método de Euler modificado é resumido no seguinte algoritmo.

#### Algoritmo para o método de Euler modificado

1. A partir da solução no ponto  $(x_i, y_i)$ , calcule o próximo valor da variável independente:

$$x_{i+1} = x_i + h$$

2. Calcule  $f(x_i, y_i)$ .
3. Estime  $y_{i+1}$  usando o método de Euler:

$$y_{i+1}^{Eu} = y_i + f(x_i, y_i)h$$

4. Calcule  $f(x_{i+1}, y_{i+1}^{Eu})$ .
5. Calcule a solução numérica em  $x = x_{i+1}$ :

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{Eu})}{2} h$$

A implementação do método de Euler modificado é mostrada no Exemplo 8-3, onde o problema do Exemplo 8-1 é novamente resolvido.

### Exemplo 8-3: Solução de uma EDO de primeira ordem usando o método de Euler modificado

Use o método de Euler modificado para resolver a EDO  $\frac{dy}{dx} = -1,2y + 7e^{-0,3x}$  de  $x = 0$  a  $x = 2,5$  com a condição inicial  $y(0) = 3$ . Escreva um programa no MATLAB que resolva a equação usando  $h = 0,5$ .

Compare os resultados com a solução exata (analítica):  $y = \frac{70}{9}e^{-0,3x} - \frac{43}{9}e^{-1,2x}$ .

#### SOLUÇÃO

O programa a seguir resolve a equação e traça um gráfico que mostra as soluções numérica e exata.

#### Programa 8-3 Função criada pelo usuário. Solução de uma EDO de primeira ordem usando o método de Euler modificado.

```
function [x, y] = edoEulerMod(EDO,a,b,h,yINI)
% edoEulerMod resolve uma EDO de primeira ordem com valor inicial usando o
% método de Euler modificado.
% Variáveis de entrada:
% EDO Nome (string) de uma função em arquivo que calcula dy/dx.
% a Primeiro valor de x.
% b Último valor de x.
% h Passo de integração.
% yINI Valor da solução y no primeiro ponto (valor inicial).
% Variáveis de saída:
% x Vetor com a coordenada x dos pontos da solução.
% y Vetor com a coordenada y dos pontos da solução.

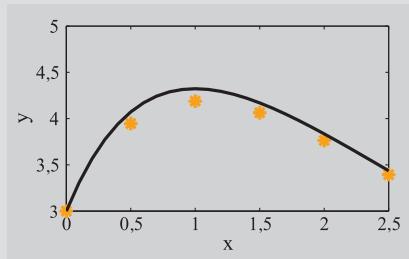
x(1) = a; y(1) = yINI; % Atribui o valor inicial a x(1) e y(1).
N = (b - a)/h; % Determina o número de passos.
for i = 1:N
    x(i + 1) = x(i) + h; % Aplica a Eq. (8.14).
    InclinEu = feval(EDO,x(i),y(i)); % Determina a inclinação no início do intervalo, Eq. (8.53).
    yEu = y(i) + SlopeEu*h; % Aplica a Eq. (8.54).
    InclinFin = feval(EDO,x(i + 1),yEu); % Determina a inclinação estimada no final do intervalo, Eq. (8.55).
    y(i + 1) = y(i) + (InclinEu + InclinFin)*h/2; % Calcule a solução numérica no passo i, Eq. (8.56).
end
```

O problema é resolvido com o seguinte programa, que usa a função edoEulerMod. A função EDO no argumento de entrada de edoEulerMod é a mesma usada no Exemplo 8-1 (Cap8Exmp1EDO). O programa também traça um gráfico que mostra as soluções exata e numérica.

```
clear all
a = 0; b = 2.5; h = 0.5; yINI = 3; % Atribui o domínio, o passo de integração e o valor inicial às variáveis.
[x, y] = edoEulerMod ('Cap8Exmp1EDO',a,b,h,yINI); % Usa a função edoEulerMod.
xp = a:0.1:b;
yp = 70/9*exp(-0.3*xp) - 43/9*exp(-1.2*xp); % Cria vetores para o traçado da solução exata.
plot(x,y,'r',xp,yp)
xlabel('x'); ylabel('y')
```

O gráfico traçado pelo programa é mostrado na figura ao lado. A figura mostra que os pontos calculados estão muito mais próximos da solução exata do que no Exemplo 8-1, letra (a), em que o método explícito de Euler foi empregado usando o mesmo passo de integração.

Os valores das soluções numérica e exata, bem como o erro, que é a diferença entre essas soluções, são:



$i$	1	2	3	4	5	6
$x_i$	0,0	0,5	1,0	1,5	2,0	2,5
$y_i$ (numérico)	3,0	3,946	4,188	4,063	3,764	3,394
$y_i$ (exato)	3,0	4,072	4,323	4,170	3,835	3,436
Erro	0,0	0,1261	0,1351	0,1063	0,0716	0,0425

A comparação dos erros aqui obtidos com aqueles obtidos no Exemplo 8-1, onde o problema foi resolvido com o método explícito de Euler usando o mesmo passo de integração, mostra que o erro referente ao método modificado é muito menor.

## 8.4 MÉTODO DO PONTO CENTRAL

O método do ponto central é mais uma modificação do método explícito de Euler. Aqui, a inclinação usada para calcular  $y_{i+1}$  é uma estimativa para a inclinação no ponto central do intervalo. Essa estimativa é calculada em dois passos. Primeiramente, o método de Euler é usado para calcular o valor aproximado de  $y$  no ponto central do intervalo,  $x_m = x_i + h/2$ , escrito como  $y_m$ :

$$y_m = y_i + f(x_i, y_i) \frac{h}{2} \quad (8.57)$$

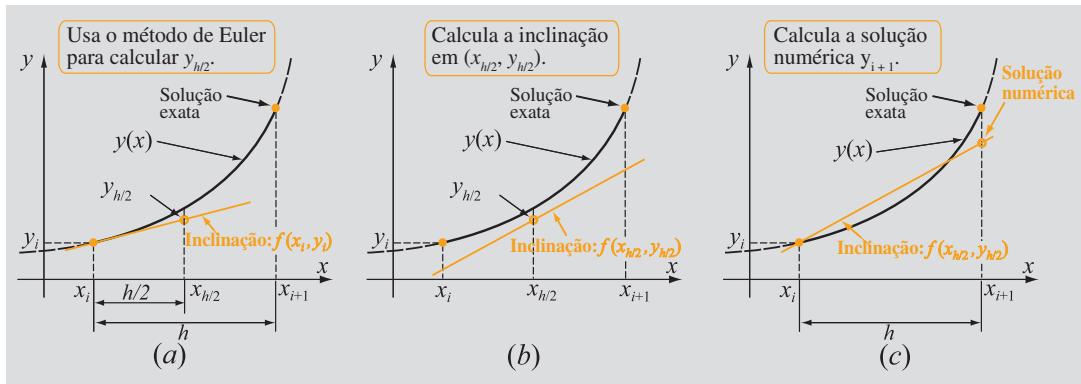
Em seguida, a inclinação estimada no ponto central é calculada com a substituição de  $(x_m, y_m)$  na equação diferencial que envolve  $\frac{dy}{dx}$ :

$$\left. \frac{dy}{dx} \right|_{x=x_m} = f(x_m, y_m) \quad (8.58)$$

A inclinação dada pela Eq. (8.58) é então usada para calcular a solução numérica,  $y_{i+1}$ :

$$y_{i+1} = y_i + f(x_m, y_m) h \quad (8.59)$$

O método do ponto central é ilustrado esquematicamente na Fig. 8.11. A Figura 8-11a mostra a determinação do ponto central com o método explícito de Euler usando a Eq. (8.57). A Fig. 8-11b mostra a inclinação estimada obtida com a Eq. (8.58) e a Fig. 8-11c mostra o valor de  $y_{i+1}$  obtido com a Eq. (8.59). A Eq. (8.59) também pode ser deduzida a partir da integração da EDO ao longo do intervalo  $[x_i, x_{i+1}]$  usando o método do retângulo aplicado no ponto central do intervalo.



**Figura 8-11** Método do ponto central.

## 8.5 MÉTODOS DE RUNGE-KUTTA

Os métodos de Runge-Kutta compõem uma família de técnicas numéricas explícitas de passo simples usadas na solução de EDOs de primeira ordem. Conforme discutido na Seção 8.1, em um subintervalo  $[x_i, x_{i+1}]$ , onde  $h = x_{i+1} - x_i$ , o valor de  $y_{i+1}$  é calculado usando:

$$y_{i+1} = y_i + \text{Inclinação} \cdot h \quad (8.60)$$

onde *Inclinação* é uma constante. O valor de *Inclinação* na Eq. (8.60) é obtido a partir do cálculo da inclinação em vários pontos no interior do subintervalo. Os diferentes tipos de métodos de Runge-Kutta são classificados de acordo com sua ordem. A ordem identifica o número de pontos usados em um subintervalo para determinar o valor de *Inclinação* na Eq. (8.60). Métodos de Runge-Kutta de segunda ordem usam a inclinação em dois pontos, métodos de terceira ordem usam três pontos, e assim por diante. A versão de quarta ordem desse método, que utiliza quatro pontos, é conhecida como método de Runge-Kutta clássico. A ordem do método também está relacionada ao erro de truncamento global. Por exemplo, o método de Runge-Kutta de segunda ordem possui um erro de truncamento global de  $O(h^3)$  e um erro de truncamento global de  $O(h^2)$ . Para cada ordem, existem vários métodos. As diferenças entre os métodos aparecem na localização dos pontos usados para determinar as inclinações e no procedimento usado para determinar a constante *Inclinação* na Eq. (8.60) a partir dessas inclinações.

Os métodos de Runge-Kutta são mais precisos do que o método explícito de Euler. A sua precisão aumenta (isto é, o erro de truncamento diminui) à medida que a ordem do método aumenta. Em cada passo, no entanto, dependendo da ordem, são necessárias várias avaliações da função para se estimar a derivada de  $f(x, y)$ .

### 8.5.1 Métodos de Runge-Kutta de segunda ordem

A forma geral dos métodos de Runge-Kutta de segunda ordem é:

$$y_{i+1} = y_i + (c_1 K_1 + c_2 K_2)h \quad (8.61)$$

com

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f(x_i + a_2 h, y_i + b_{21} K_1 h) \end{aligned} \quad (8.62)$$

onde  $c_1$ ,  $c_2$ ,  $a_2$ , e  $b_{21}$  são constantes. Os valores dessas constantes variam com o método de segunda ordem específico.

O método de Euler modificado (Seção 8.3) e o método do ponto central (Seção 8.4) são duas versões do método de Runge-Kutta de segunda ordem. Essas duas versões e uma versão adicional chamada de método de Heun são apresentadas a seguir, usando a forma geral das Eqs. (8.61) e (8.62).

#### Método de Euler modificado na forma de um método de Runge-Kutta de segunda ordem

No método de Euler modificado, as constantes nas Eqs. (8.61) e (8.62) são:

$$c_1 = \frac{1}{2}, \quad c_2 = \frac{1}{2}, \quad a_2 = 1 \quad \text{e} \quad b_{21} = 1$$

A substituição dessas constantes nas Eqs. (8.61) e (8.62) resulta em:

$$y_{i+1} = y_i + \frac{1}{2}(K_1 + K_2)h \quad (8.63)$$

onde

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f(x_i + h, y_i + K_1 h) \end{aligned} \quad (8.64)$$

As Eqs. (8.63) e (8.64) também podem ser deduzidas a partir da integração da EDO ao longo do intervalo  $[x_i, x_{i+1}]$  usando o método trapezoidal.

#### Método do ponto central na forma de um método de Runge-Kutta de segunda ordem

No método do ponto central, as constantes nas Eqs. (8.61) e (8.62) são:

$$c_1 = 0, \quad c_2 = 1, \quad a_2 = \frac{1}{2} \quad \text{e} \quad b_{21} = \frac{1}{2}$$

A substituição dessas constantes nas Eqs. (8.61) e (8.62) resulta em:

$$y_{i+1} = y_i + K_2 h \quad (8.65)$$

onde

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h\right) \end{aligned} \quad (8.66)$$

### Método de Heun

No método de Heun, as constantes nas Eqs. (8.61) e (8.62) são:

$$c_1 = \frac{1}{4}, \quad c_2 = \frac{3}{4}, \quad a_2 = \frac{2}{3} \quad \text{e} \quad b_{21} = \frac{2}{3}$$

A substituição dessas constantes nas Eqs. (8.61) e (8.62) resulta em:

$$y_{i+1} = y_i + \left( \frac{1}{4}K_1 + \frac{3}{4}K_2 \right)h \quad (8.67)$$

onde

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f\left(x_i + \frac{2}{3}h, y_i + \frac{2}{3}K_1 h\right) \end{aligned} \quad (8.68)$$

### Erro de truncamento nos métodos de Runge-Kutta de segunda ordem

O erro de truncamento local nos métodos de Runge-Kutta de segunda ordem é de  $O(h^3)$ , e o erro de truncamento global é de  $O(h^2)$ . Este erro é uma ordem menor do que os erros de truncamento associados ao método explícito de Euler. Isso significa que, para uma mesma precisão, um maior passo de integração pode ser usado nos métodos de Runge-Kutta. Em cada passo, no entanto, a função  $f(x, y)$  é calculada duas vezes nos métodos de Runge-Kutta. Uma estimativa para o erro de truncamento local presente no método de Runge-Kutta escrito como o método de Euler modificado é apresentada na Seção 8.11.

### Métodos de Runge-Kutta de segunda ordem e a expansão em série de Taylor

Métodos de Runge-Kutta de segunda ordem podem ser associados a uma expansão em série de Taylor. No intervalo  $[x_i, x_{i+1}]$ , onde  $h = x_{i+1} - x_i$  e o valor de  $y_i$  é conhecido, o valor de  $y_{i+1}$  pode ser aproximado usando três termos da expansão em série de Taylor:

$$y_{i+1} = y_i + \left. \frac{dy}{dx} \right|_{x_i} h + \left. \frac{1}{2} \frac{d^2 y}{dx^2} \right|_{x_i} h^2 + O(h^3) \quad (8.69)$$

A derivada primeira  $\left. \frac{dy}{dx} \right|_{x_i}$  corresponde ao lado direito da equação diferencial avaliada em  $x = x_i$ :

$$\left. \frac{dy}{dx} \right|_{x_i} = f(x_i, y_i) \quad (8.70)$$

e a derivada segunda  $\left. \frac{d^2 y}{dx^2} \right|_{x_i}$  pode ser determinada a partir da derivada primeira usando a regra da cadeia (ver Capítulo 2):

$$\left. \frac{d^2 y}{dx^2} \right|_{x_i} = \left. \frac{\partial f(x, y)}{\partial x} \right|_{x_i, y_i} + \left. \frac{\partial f(x, y)}{\partial y} \right|_{x_i, y_i} \left. \frac{dy}{dx} \right|_{x_i} \quad (8.71)$$

A substituição das Eqs. (8.70) e (8.71) na Eq. (8.69) resulta em:

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{1}{2} \frac{\partial f(x, y)}{\partial x} \Big|_{x_i, y_i} h^2 + \frac{1}{2} \frac{\partial f(x, y)}{\partial y} f(x_i, y_i)h^2 + O(h^3) \quad (8.72)$$

Na forma geral dos métodos de Runge-Kutta de segunda ordem (ver Eqs. (8.61) e (8.62)),  $K_2$  é definido como:

$$K_2 = f(x_i + a_2 h, y_i + b_{21} K_1 h) \quad (8.73)$$

Usando a expansão em série de Taylor de uma função de duas variáveis (ver Capítulo 2), o lado direito da última equação pode ser expandido, resultando em:

$$\begin{aligned} K_2 &= f(x_i + a_2 h, y_i + b_{21} K_1 h) = \\ &= f(x_i, y_i) + a_2 h \frac{\partial f(x, y)}{\partial x} \Big|_{x_i, y_i} + b_{21} h \frac{\partial f(x, y)}{\partial y} \Big|_{x_i, y_i} K_1 + O(h^2) \end{aligned} \quad (8.74)$$

Substituindo  $K_1 = f(x_i, y_i)$  e  $K_2$  da Eq. (8.74) na Eq. (8.61), obtém-se:

$$\begin{aligned} y_{i+1} &= y_i + c_1 f(x_i, y_i)h + c_2 f(x_i, y_i)h + c_2 a_2 h^2 \frac{\partial f(x, y)}{\partial x} \Big|_{x_i, y_i} \\ &\quad + c_2 b_{21} h^2 \frac{\partial f(x, y)}{\partial y} \Big|_{x_i, y_i} f(x_i, y_i) + O(h^3) \end{aligned} \quad (8.75)$$

As Eqs. (8.72) e (8.75) são duas equações diferentes que podem ser empregadas no cálculo do mesmo valor de  $y_{i+1}$ . Como as equações têm termos do mesmo tipo, os coeficientes de ambas devem ser iguais. As três equações a seguir são obtidas igualando-se os coeficientes de termos com a mesma potência de  $h$ :

$$c_1 + c_2 = 1, \quad c_2 a_2 = \frac{1}{2}, \quad c_2 b_{21} = \frac{1}{2} \quad (8.76)$$

As três equações têm quatro incógnitas, o que significa que não há solução única; ao invés disso, existem muitas soluções. As constantes dos métodos de Euler modificado, do ponto central e de Heun são três exemplos disso. Se os coeficientes forem escolhidos para satisfazer à Eq. (8.76), então é possível ver, a partir das Eqs. (8.72) e (8.75), que o erro de truncamento local é de  $O(h^3)$ .

O Exemplo 8-4 mostra a solução manual da EDO resolvida nos Exemplos 8-1 e 8-3 usando o método de Runge-Kutta.

**Exemplo 8-4:** Solução manual de uma EDO de primeira ordem usando o método de Runge-Kutta de segunda ordem

Use o método de Runge-Kutta de segunda ordem (versão modificada de Euler) para resolver a EDO  $\frac{dy}{dx} = -1,2y + 7e^{-0.3x}$  de  $x = 0$  a  $x = 2,0$  com a condição inicial  $y = 3$  em  $x = 0$ . Resolva manualmente usando  $h = 0,5$ .

**SOLUÇÃO**

Essa equação já foi resolvida com o método de Euler modificado no Exemplo 8-3, mas a solução foi obtida escrevendo-se um programa no MATLAB. Aqui, para ilustrar a aplicação do método de Runge-Kutta, os cálculos são realizados manualmente.

O primeiro ponto da solução é  $(0, 3)$ , que corresponde à condição inicial. Os valores de  $x$  e  $y$  no primeiro ponto são  $x_1 = 0$  e  $y_1 = 3$ .

O restante da solução é feito em passos. Em cada passo, o próximo valor da variável independente é dado por:

$$x_{i+1} = x_i + h = x_i + 0,5 \quad (8.77)$$

O valor da variável dependente  $y_{i+1}$  é obtido primeiramente com a determinação de  $K_1$  e  $K_2$  a partir da Eq. (8.64):

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f(x_i + h, y_i + K_1 h) \end{aligned} \quad (8.78)$$

e com a substituição dos valores de  $K$  na Eq. (8.63):

$$y_{i+1} = y_i + \frac{1}{2}(K_1 + K_2)h \quad (8.79)$$

**Primeiro passo** No primeiro passo,  $i = 1$ . As Eqs. (8.77)-(8.79) fornecem:

$$x_2 = x_1 + 0,5 = 0 + 0,5 = 0,5$$

$$K_1 = -1,2y_1 + 7e^{-0.3x_1} = -1,2 \cdot 3 + 7e^{-0.3 \cdot 0} = 3,4$$

$$y_1 + K_1 h = 3 + 3,4 \cdot 0,5 = 4,7$$

$$K_2 = -1,2(y_1 + K_1 h) + 7e^{-0.3(x_1 + 0,5)} = -1,2 \cdot 4,7 + 7e^{-0.3 \cdot 0,5} = 0,385$$

$$y_2 = y_1 + \frac{1}{2}(K_1 + K_2)h = 3 + \frac{1}{2}(3,4 + 0,385) \cdot 0,5 = 3,946$$

No final do primeiro passo:  $x_2 = 0,5$ ,  $y_2 = 3,946$ .

**Segundo passo** No segundo passo,  $i = 2$ . As Eqs. (8.77)-(8.79) fornecem:

$$x_3 = x_2 + 0,5 = 0,5 + 0,5 = 1,0$$

$$K_1 = -1,2y_2 + 7e^{-0.3x_2} = -1,2 \cdot 3,946 + 7e^{-0.3 \cdot 0,5} = 1,290$$

$$y_2 + K_1 h = 3,946 + 1,290 \cdot 0,5 = 4,591$$

$$K_2 = -1,2(y_2 + K_1 h) + 7e^{-0.3(x_2 + 0,5)} = -1,2 \cdot 4,591 + 7e^{-0.3 \cdot 1,0} = -0,3223$$

$$y_3 = y_2 + \frac{1}{2}(K_1 + K_2)h = 3,946 + \frac{1}{2}(1,290 + (-0,3223)) \cdot 0,5 = 4,188$$

No final do segundo passo:  $x_3 = 1,0$ ,  $y_3 = 4,188$ .

**Terceiro passo** No terceiro passo,  $i = 3$ . As Eqs. (8.77)-(8.79) fornecem:

$$x_4 = x_3 + 0,5 = 1,0 + 0,5 = 1,5$$

$$K_1 = -1,2y_3 + 7e^{-0,3x_3} = -1,2 \cdot 4,188 + 7e^{-0,3 \cdot 1,0} = 0,1601$$

$$y_3 + K_1 h = 4,188 + 0,1601 \cdot 0,5 = 4,268$$

$$K_2 = -1,2(y_3 + K_1 h) + 7e^{-0,3(x_3 + 0,5)} = -1,2 \cdot 4,268 + 7e^{-0,3 \cdot 1,5} = -0,6582$$

$$y_4 = y_3 + \frac{1}{2}(K_1 + K_2)h = 4,188 + \frac{1}{2}(0,1601 + (-0,6582)) \cdot 0,5 = 4,063$$

No final do terceiro passo:  $x_4 = 1,5$ ,  $y_4 = 4,063$ .

**Quarto passo** No quarto passo,  $i = 4$ . As Eqs. (8.77)-(8.79) fornecem:

$$x_5 = x_4 + 0,5 = 1,5 + 0,5 = 2,0$$

$$K_1 = -1,2y_4 + 7e^{-0,3x_4} = -1,2 \cdot 4,063 + 7e^{-0,3 \cdot 1,5} = -0,4122$$

$$y_4 + K_1 h = 4,063 + (-0,4122) \cdot 0,5 = 3,857$$

$$K_2 = -1,2(y_4 + K_1 h) + 7e^{-0,3(x_4 + 0,5)} = -1,2 \cdot 3,857 + 7e^{-0,3 \cdot 2,0} = -0,7867$$

$$y_5 = y_4 + \frac{1}{2}(K_1 + K_2)h = 4,063 + \frac{1}{2}(-0,4122 + (-0,7867)) \cdot 0,5 = 3,763$$

No final do quarto passo:  $x_5 = 2,0$ ,  $y_5 = 3,763$ .

A solução obtida é obviamente idêntica, exceto por erros de arredondamento, à solução apresentada no Exemplo 8-3.

### 8.5.2 Métodos de Runge-Kutta de terceira ordem

A forma geral dos métodos de Runge-Kutta de terceira ordem é:

$$y_{i+1} = y_i + (c_1 K_1 + c_2 K_2 + c_3 K_3)h \quad (8.80)$$

com

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f(x_i + a_2 h, y_i + b_{21} K_1 h) \\ K_3 &= f(x_i + a_3 h, y_i + b_{31} K_1 h + b_{32} K_2 h) \end{aligned} \quad (8.81)$$

onde  $c_1, c_2, c_3, a_2, a_3, b_{21}, b_{31}$  e  $b_{32}$  são constantes. Seis equações relacionando essas oito constantes podem ser deduzidas a partir da comparação das Eqs. (8.80) e (8.81) com a expansão em série de Taylor com quatro termos que estima o valor de  $y_{i+1}$  (a dedução está além do escopo deste livro). Isso quer dizer que, assim como nos métodos de Runge-Kutta de segunda ordem, é possível haver vários métodos de terceira ordem com diferentes conjuntos de constantes.

Um desses métodos é chamado de método de **Runge-Kutta de terceira ordem clássico**. Os valores das oito constantes neste método são:

$$c_1 = \frac{1}{6}, \quad c_2 = \frac{4}{6}, \quad c_3 = \frac{1}{6}, \quad a_2 = \frac{1}{2}, \quad a_3 = 1, \quad b_{21} = \frac{1}{2}, \quad b_{31} = -1 \quad \text{e} \quad b_{32} = 2$$

Com essas constantes, as equações que formam o método de Runge-Kutta de terceira ordem clássico são:

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 4K_2 + K_3)h \quad (8.82)$$

onde

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h\right) \\ K_3 &= f(x_i + h, y_i - K_1 h + 2K_2 h) \end{aligned} \quad (8.83)$$

As Eqs. (8.82) e (8.83) também podem ser deduzidas a partir da integração da EDO ao longo do intervalo  $[x_i, x_{i+1}]$  usando o método de Simpson 1/3.

### Erros de Truncamento nos métodos de Runge-Kutta de terceira ordem

O erro de truncamento local associado aos métodos de Runge-Kutta de terceira ordem é de  $O(h^4)$ , e o erro de truncamento global é de  $O(h^3)$ . A dedução desses erros é tediosa e está além do escopo deste livro.

Variações do método de Runge-Kutta de terceira ordem usam diferentes combinações de constantes nas Eqs. (8.80) e (8.81). As constantes desses três métodos, bem como as constantes do método de Runge-Kutta de terceira ordem clássico, são listadas na Tabela 8-1.

**Tabela 8-1** Constantes de métodos de Runge-Kutta de terceira ordem

Método	$c_1$	$c_2$	$c_3$	$a_2$	$b_{21}$	$a_3$	$b_{31}$	$b_{31}$
Clássico	1/6	4/6	1/6	1/2	1/2	1	-1	2
Nystrom	2/8	3/8	3/8	2/3	2/3	2/3	0	2/3
Quase ótimo	2/9	3/9	4/9	1/2	1/2	3/4	0	3/4
Heun de terceira ordem	1/4	0	3/4	1/3	1/3	2/3	0	2/3

### 8.5.3 Métodos de Runge-Kutta de quarta ordem

A forma geral dos métodos de Runge-Kutta de quarta ordem é:

$$y_{i+1} = y_i + (c_1 K_1 + c_2 K_2 + c_3 K_3 + c_4 K_4)h \quad (8.84)$$

com

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f(x_i + a_2 h, y_i + b_{21} K_1 h) \\ K_3 &= f(x_i + a_3 h, y_i + b_{31} K_1 h + b_{32} K_2 h) \\ K_4 &= f(x_i + a_4 h, y_i + b_{41} K_1 h + b_{42} K_2 h + b_{43} K_3 h) \end{aligned} \quad (8.85)$$

onde  $c_1, c_2, c_3, c_4, a_2, a_3, a_4, b_{21}, b_{31}, b_{32}, b_{41}, b_{42}$  e  $b_{43}$  formam um conjunto com treze constantes. Os valores dessas constantes variam com o método de quarta ordem específico.

O **método de Runge-Kutta de quarta ordem clássico** está entre os métodos mais comumente utilizados. As constantes desse método são:

$$\begin{aligned} c_1 &= c_4 = \frac{1}{6}, & c_2 &= c_3 = \frac{2}{6}, & a_2 &= a_3 = b_{21} = b_{32} = \frac{1}{2} \\ a_4 &= b_{43} = 1, & b_{31} &= b_{41} = b_{42} = 0 \end{aligned}$$

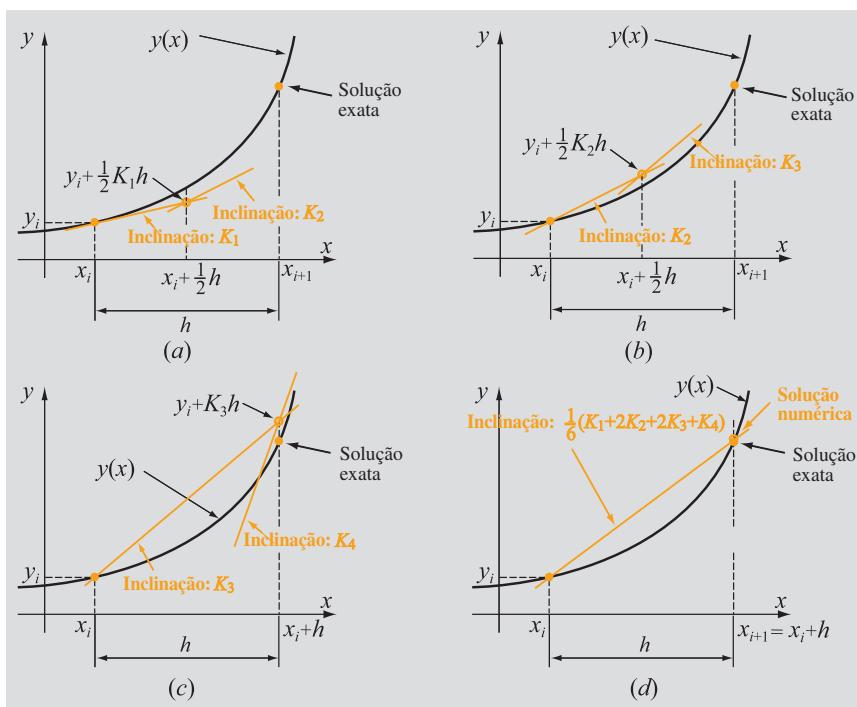
Com essas constantes, as equações do método de Runge-Kutta de quarta ordem clássico são:

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h \quad (8.86)$$

onde

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h\right) \\ K_3 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2 h\right) \\ K_4 &= f(x_i + h, y_i + K_3 h) \end{aligned} \quad (8.87)$$

O método de Runge-Kutta de quarta ordem clássico é ilustrado esquematicamente na Fig. 8-12. As letras (a) a (c) mostram a determinação das inclinações presentes na Eq. (8.87). A letra (a) mostra a inclinação  $K_1$  e seu uso na determinação da inclinação  $K_2$ ; a letra (b) mostra como a inclinação  $K_2$  é usada para determinar a inclinação  $K_3$ ; a letra (c) mostra como a inclinação  $K_3$  é usada para determinar a inclinação  $K_4$ ; finalmente, a letra (d) ilustra a aplicação da Eq. (8.86), na qual a inclinação utilizada no cálculo de  $y_{i+1}$  é uma média ponderada das inclinações  $K_1, K_2, K_3$  e  $K_4$ .



**Figura 8-12** Método de Runge-Kutta de quarta ordem clássico.

### Erros de truncamento nos métodos de Runge-Kutta de quarta ordem

O erro de truncamento local associado aos métodos de Runge-Kutta de quarta ordem é de  $O(h^5)$ , e o erro de truncamento global é de  $O(h^4)$ . A dedução desses erros é tediosa e está além do escopo deste livro.

As Eqs. (8.86) e (8.87) também podem ser deduzidas a partir da integração da EDO ao longo do intervalo  $[x_i, x_{i+1}]$  usando o método de Simpson 1/3. Entretanto, diferentemente de outros métodos de Runge-Kutta, as avaliações realizadas nos métodos de quarta ordem são combinadas de forma a fornecer um erro de truncamento local de ordem  $O(h^5)$ .

Os Exemplos 8-5 e 8-6 mostram soluções de problemas de valor inicial de primeira ordem usando o método de Runge-Kutta de quarta ordem clássico. No Exemplo 8-5, os três primeiros passos da solução são calculados manualmente, e, no Exemplo 8-6, implementa-se uma função no MATLAB.

#### Exemplo 8-5: Solução manual de uma EDO de primeira ordem usando o método de Runge-Kutta de quarta ordem

Use o método de Runge-Kutta de quarta ordem para resolver a EDO  $\frac{dy}{dx} = -1,2y + 7e^{-0,3x}$  de  $x = 0$  a  $x = 1,5$  com a condição inicial  $y = 3$  em  $x = 0$ . Resolva manualmente usando  $h = 0,5$ .

Compare os resultados com a solução exata (analítica):  $y = \frac{70}{9}e^{-0,3x} - \frac{43}{9}e^{-1,2x}$ .

#### SOLUÇÃO

O primeiro ponto da solução é  $(0, 3)$ , que corresponde à condição inicial. Os valores de  $x$  e  $y$  no primeiro ponto são  $x_1 = 0$  e  $y_1 = 3$ .

O restante da solução é feito em passos. Em cada passo, o próximo valor da variável independente é dado por:

$$x_{i+1} = x_i + h = x_i + 0,5$$

O valor da variável dependente  $y_{i+1}$  é calculado primeiramente com a avaliação de  $K_1$ ,  $K_2$ ,  $K_3$  e  $K_4$  a partir da Eq. (8.87):

$$\begin{aligned} K_1 &= f(x_i, y_i) \\ K_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1h\right) \\ K_3 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2h\right) \\ K_4 &= f(x_i + h, y_i + K_3h) \end{aligned} \tag{8.89}$$

e com a substituição dos valores de  $K$  na Eq. (8.86):

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h \tag{8.90}$$

**Primeiro passo** No primeiro passo,  $i = 1$ . As Eqs. (8.88)-(8.90) fornecem:

$$x_2 = x_1 + 0,5 = 0 + 0,5 = 0,5$$

$$K_1 = -1,2y_1 + 7e^{-0,3x_1} = -1,2 \cdot 3 + 7e^{-0,3 \cdot 0} = 3,4$$

$$x_1 + \frac{1}{2}h = 0 + \frac{1}{2} \cdot 0,5 = 0,25 \quad y_1 + \frac{1}{2}K_1h = 3 + \frac{1}{2} \cdot 3,4 \cdot 0,5 = 3,85$$

$$K_2 = -1,2\left(y_1 + \frac{1}{2}K_1h\right) + 7e^{-0,3\left(x_1 + \frac{1}{2}h\right)} = -1,2 \cdot 3,85 + 7e^{-0,3 \cdot 0,25} = 1,874$$

$$y_1 + \frac{1}{2}K_2h = 3 + \frac{1}{2} \cdot 1,874 \cdot 0,5 = 3,469$$

$$K_3 = -1,2\left(y_1 + \frac{1}{2}K_2h\right) + 7e^{-0,3\left(x_1 + \frac{1}{2}h\right)} = -1,2 \cdot 3,469 + 7e^{-0,3 \cdot 0,25} = 2,331$$

$$y_1 + K_3h = 3 + 2,331 \cdot 0,5 = 4,166$$

$$K_4 = -1,2(y_1 + K_3h) + 7e^{-0,3(x_1 + h)} = -1,2 \cdot 4,166 + 7e^{-0,3 \cdot 0,5} = 1,026$$

$$y_2 = y_1 + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h = 3 + \frac{1}{6}(3,4 + 2 \cdot 1,874 + 2 \cdot 2,331 + 1,026) \cdot 0,5 = 4,069$$

No final do primeiro passo:  $x_2 = 0,5$ ,  $y_2 = 4,069$ .

**Segundo passo** No segundo passo,  $i = 2$ . As Eqs. (8.88)-(8.90) fornecem:

$$x_3 = x_2 + 0,5 = 0,5 + 0,5 = 1,0$$

$$K_1 = -1,2y_2 + 7e^{-0,3x_2} = -1,2 \cdot 4,069 + 7e^{-0,3 \cdot 0,5} = 1,142$$

$$x_2 + \frac{1}{2}h = 0,5 + \frac{1}{2} \cdot 0,5 = 0,75 \quad y_2 + \frac{1}{2}K_1h = 4,069 + \frac{1}{2} \cdot 1,142 \cdot 0,5 = 4,355$$

$$K_2 = -1,2\left(y_2 + \frac{1}{2}K_1h\right) + 7e^{-0,3\left(x_2 + \frac{1}{2}h\right)} = -1,2 \cdot 4,355 + 7e^{-0,3 \cdot 0,75} = 0,3636$$

$$y_2 + \frac{1}{2}K_2h = 4,069 + \frac{1}{2} \cdot 0,3636 \cdot 0,5 = 4,16$$

$$K_3 = -1,2\left(y_2 + \frac{1}{2}K_2h\right) + 7e^{-0,3\left(x_2 + \frac{1}{2}h\right)} = -1,2 \cdot 4,16 + 7e^{-0,3 \cdot 0,75} = 0,5976$$

$$y_2 + K_3h = 4,069 + 0,5976 \cdot 0,5 = 4,368$$

$$K_4 = -1,2(y_2 + K_3h) + 7e^{-0,3(x_2 + h)} = -1,2 \cdot 4,368 + 7e^{-0,3 \cdot 1,0} = -0,0559$$

$$y_3 = y_2 + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h = 4,069 + \frac{1}{6}[1,142 + 2 \cdot 0,3636 + 2 \cdot 0,5976 + (-0,0559)] \cdot 0,5 = 4,32$$

No final do segundo passo:  $x_3 = 1,0$ ,  $y_3 = 4,32$ .

**Terceiro passo** No terceiro passo,  $i = 3$ . As Eqs. (8.88)-(8.90) fornecem:

$$x_4 = x_3 + 0,5 = 1,0 + 0,5 = 1,5$$

$$K_1 = -1,2y_3 + 7e^{-0,3x_3} = -1,2 \cdot 4,32 + 7e^{-0,3 \cdot 1,0} = 0,001728$$

$$x_3 + \frac{1}{2}h = 1,0 + \frac{1}{2} \cdot 0,5 = 1,25 \quad y_3 + \frac{1}{2}K_1h = 4,32 + \frac{1}{2} \cdot 0,001728 \cdot 0,5 = 4,320$$

$$K_2 = -1,2\left(y_3 + \frac{1}{2}K_1h\right) + 7e^{-0,3\left(x_3 + \frac{1}{2}h\right)} = -1,2 \cdot 4,32 + 7e^{-0,3 \cdot 1,25} = -0,373$$

$$y_3 + \frac{1}{2}K_2 h = 4,32 + \frac{1}{2} \cdot (-0,373) \cdot 0,5 = 4,227$$

$$K_3 = -1,2\left(y_3 + \frac{1}{2}K_2 h\right) + 7e^{-0.3(x_3 + \frac{1}{2}h)} = -1,2 \cdot 4,227 + 7e^{-0.3 \cdot 1,25} = -0,2614$$

$$y_3 + K_3 h = 4,32 + (-0,2614) \cdot 0,5 = 4,189$$

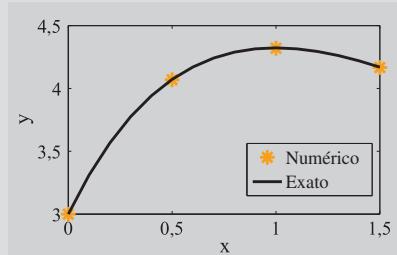
$$K_4 = -1,2(y_3 + K_3 h) + 7e^{-0.3(x_3 + h)} = -1,2 \cdot 4,189 + 7e^{-0.3 \cdot 1,5} = -0,5634$$

$$y_4 = y_3 + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h = 4,32 + \frac{1}{6}[0,001728 + 2 \cdot (-0,373) + 2 \cdot (-0,2614) + (-0,5634)] \cdot 0,5 = 4,167$$

No final do terceiro passo:  $x_4 = 1,5$ ,  $y_4 = 4,167$ .

Uma comparação entre a solução numérica e a solução exata é mostrada na tabela e na figura a seguir. O erro é de 0,003. O erro de truncamento global no método de Runge-Kutta de quarta ordem é da ordem de  $h^4$ . Neste problema,  $h^4 = 0,5^4 = 0,0625$ , que é menor que o erro real. Deve-se lembrar, no entanto, que o termo  $h^4$  deve ser multiplicado por uma constante cujo valor não é conhecido. Isso mostra que as estimativas para o erro de truncamento são suficientemente boas para permitir uma comparação entre a precisão de diferentes métodos, ainda que o valor numérico dessas estimativas não necessariamente represente o erro de forma exata.

$i$	1	2	3	4
$x_i$	0,0	0,5	1,0	1,5
$y_i$ (numérico)	3,0	4,069	4,32	4,167
$y_i$ (exato)	3,0	4,072	4,323	4,170
Erro	0,0	0,003	0,003	0,003



### Exemplo 8-6: Função criada pelo usuário para resolver uma EDO de primeira ordem usando o método de Runge-Kutta de quarta ordem

Escreva uma função no MATLAB que resolva uma EDO de primeira ordem usando o método de Runge-Kutta de quarta ordem clássico. Chame a função de  $[x, y] = \text{edoRK4}(\text{EDO}, a, b, h)$ , onde EDO é o nome da função em arquivo que calcula a derivada  $dy/dx$ .

Use a função edoRK4 para resolver  $\frac{dy}{dx} = -1,2y + 7e^{-0.3x}$  de  $x = 0$  a  $x = 2,5$  com a condição inicial  $y = 3$  em  $x = 0$ . Compare os resultados com a solução exata (analítica):  $y = \frac{70}{9}e^{-0.3x} - \frac{43}{9}e^{-1.2x}$ .

### SOLUÇÃO

Para resolver este problema, escreve-se no MATLAB uma função chamada edoRK4, que resolve uma EDO de primeira ordem. A função é em seguida usada em um programa que traça um gráfico comparando as soluções numérica e exata. A EDO é escrita em uma função à parte, que é usada pela função edoRK4.

**Programa 8-4 Função criada pelo usuário. Solução de uma EDO de primeira ordem usando o método de Runge-Kutta de quarta ordem.**

```
function [x, y] = edoRK4(EDO,a,b,h,yIni)
% edoRK4 resolve uma EDO de primeira ordem com valor inicial usando o
% método de Runge-Kutta de quarta ordem.
% Variáveis de entrada:
% EDO Nome (string) da função em arquivo que calcula dy/dx.
% a Primeiro valor de x.
% b Último valor de x.
% h Passo de integração.
% yIni Valor da solução y no primeiro ponto (valor inicial).
% Variável de saída:
% x Vetor com a coordenada x dos pontos da solução.
% y Vetor com a coordenada y dos pontos da solução.
```

x(1) = a; y(1) = yIni

Atribui o valor inicial ao primeiro ponto da solução.

n = (b - a)/h;

Calcula o número de passos.

for i = 1:n

    x(i + 1) = x(i) + h;

Calcula o novo valor da variável independente.

    K1 = feval(EDO,x(i),y(i));

Calcula  $K_1$ , Eq. (8.87).

    xhalf = x(i) + h/2;

    yK1 = y(i) + K1\*h/2;

Calcula  $K_2$ , Eq. (8.87).

    K2 = feval(ODE,xhalf,yK1);

Calcula  $K_3$ , Eq. (8.87).

    yK2 = y(i) + K2\*h/2;

    K3 = feval(EDO,xhalf,yK2);

Calcula  $K_4$ , Eq. (8.87).

    yK3 = y(i) + K3\*h;

    K4 = feval(EDO,x(i + 1),yK3);

Calcula  $K_4$ , Eq. (8.87).

    y(i + 1) = y(i) + (K1 + 2\*K2 + 2\*K3 + K4)\*h/6;

Calcula o novo valor da variável dependente,  
Eq. (8.86).

end

O programa a seguir usa a função edoRK4 para resolver a EDO.

a = 0; b = 2.5;

Usa a função edoRK4.

h = 0.5; yIni = 3;

[x,y] = edoRK4('Cap8Exmp6EDO',a,b,h,yIni)

xp = a:0.1:b;

yp = 70/9\*exp(-0.3\*xp) - 43/9\*exp(-1.2\*xp);

Cria vetores para o traçado da solução exata.

plot(x,y,'r',xp,yp)

yExato= 70/9\*exp(-0.3\*x) - 43/9\*exp(-1.2\*x) Calcula a solução exata nos pontos da solução numérica.

error = yExato - y

A função Cap8Exmp6EDO usada no argumento da função edoRK4 calcula o valor de  $dy/dx$ .

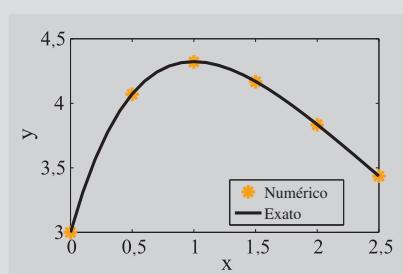
function dydx = Cap8Exmp6EDO(x,y)

dydx = -1.2\*y + 7\*exp(-0.3\*x);

Quando o programa é executado, os seguintes dados são exibidos na janela de comandos do MATLAB.

```
x =
    0   0.5000  1.0000  1.5000  2.0000  2.5000
y =
    3.0000  4.0698  4.3203  4.1676  3.8338  3.4353
yExato=
    3.0000  4.0723  4.3229  4.1696  3.8351  3.4361
erro =
    0   0.0025  0.0026  0.0020  0.0013  0.0008
```

Os resultados (o vetor `erro` e a figura) mostram que a solução numérica apresenta um erro muito pequeno, mesmo com um passo de integração bastante grande.



## 8.6 MÉTODOS MULTIPASSO

Em métodos de passo simples, a solução  $y_{i+1}$  no ponto  $x = x_{i+1}$  é obtida usando apenas o valor de  $y_i$  e  $x_i$  no ponto anterior. Em métodos multipasso, a solução  $y_{i+1}$  no ponto  $x = x_{i+1}$  é calculada a partir de dois ou mais pontos anteriores. Métodos multipasso podem ser implícitos ou explícitos. Em métodos multipasso explícitos, a solução  $y_{i+1}$  é calculada a partir de uma fórmula explícita. Por exemplo, se três pontos anteriores forem usados, o valor da variável dependente  $y_{i+1}$  é calculado com a avaliação de uma expressão na forma:

$$y_{i+1} = F(x_{i-2}, y_{i-2}, x_{i-1}, y_{i-1}, x_i, y_i, x_{i+1}) \quad (8.91)$$

Essa forma é explícita porque o lado direito da equação contém apenas grandezas conhecidas. Tal método obviamente não pode ser usado para determinar a solução no segundo ponto, pois nesse caso apenas um ponto anterior é conhecido (a condição inicial). Dependendo do número de pontos anteriores utilizados, a determinação dos primeiros pontos pode ser feita com métodos de passo simples ou com métodos multipasso que utilizem menos pontos. Em métodos multipasso implícitos, a incógnita  $y_{i+1}$  aparece em ambos os lados da equação, que é então resolvida numericamente com algum dos métodos descritos no Capítulo 3.

### 8.6.1 Método de Adams-Bashforth

O método de Adams-Bashforth é um método multipasso explícito usado na solução de EDOs de primeira ordem. Existem várias fórmulas de Adams-Bashforth disponíveis para o cálculo de  $y_{i+1}$  usando a solução conhecida em dois ou mais pontos anteriores. As fórmulas são classificadas de acordo com a sua ordem, que corresponde ao número de pontos usados e também à ordem do erro de truncamento global do esquema. A fórmula de segunda ordem usa os pontos  $(x_i, y_i)$  e  $(x_{i-1}, y_{i-1})$ . A fórmula de terceira ordem usa os pontos  $(x_i, y_i)$ ,  $(x_{i-1}, y_{i-1})$  e  $(x_{i-2}, y_{i-2})$ , e assim por diante.

As fórmulas do método de Adams-Bashforth são deduzidas a partir da integração da equação diferencial ao longo de um intervalo arbitrário  $[x_i, x_{i+1}]$ . Se a equação diferencial é dada na forma:

$$\frac{dy}{dx} = f(x, y) \quad (8.92)$$

a integração de  $x_i$  a  $x_{i+1}$  resulta em:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y) dx \quad (8.93)$$

A integração é realizada representando-se  $f(x, y)$  de forma aproximada com um polinômio que interpola essa função em  $(x_i, y_i)$  e em pontos anteriores.

#### Método de Adams-Bashforth de segunda ordem

Se a função  $f(x, y)$  na Eq. (8.93) for aproximada por um polinômio que interpola o seu valor em  $(x_i, y_i)$  e no ponto anterior,  $(x_{i-1}, y_{i-1})$ , então o polinômio é de primeira ordem e pode ser escrito na forma:

$$f(x, y) = f(x_i, y_i) + \frac{f(x_i, y_i) - f(x_{i-1}, y_{i-1})}{h}(x - x_i) \quad (8.94)$$

onde  $h = x_i - x_{i-1}$ . A Eq. (8.94) é substituída na Eq. (8.93), que é então integrada. O cálculo da integral resulta em:

$$y_{i+1} = y_i + \frac{h}{2}[3f(x_i, y_i) - f(x_{i-1}, y_{i-1})] \quad (8.95)$$

A Eq. (8.95) é o método de Adams-Bashforth de segunda ordem que aproxima a solução da equação diferencial em  $x_{i+1}$  a partir das soluções previamente calculadas em  $(x_i, y_i)$  e  $(x_{i-1}, y_{i-1})$ .

Fórmulas de ordem mais elevada que incluem mais pontos são deduzidas da mesma maneira. As fórmulas para o método de Adams-Bashforth de terceira e quarta ordem são dadas a seguir.

### Método de Adams-Bashforth de terceira ordem

A fórmula de terceira ordem fornece a solução  $y_{i+1}$  em termos dos valores conhecidos da solução nos três pontos anteriores:

$$y_{i+1} = y_i + \frac{h}{12}[23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})] \quad (8.96)$$

Quando se resolve uma EDO de primeira ordem, o primeiro ponto, correspondente à condição inicial, é conhecido. As soluções no segundo e no terceiro ponto têm que ser determinadas por outros métodos, e a fórmula na Eq. (8.96) pode ser usada a partir do quarto ponto.

### Método de Adams-Bashforth de quarta ordem

A fórmula de quarta ordem fornece a solução  $y_{i+1}$  em termos dos valores conhecidos da solução nos quatro pontos anteriores:

$$y_{i+1} = y_i + \frac{h}{24}[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}] \quad (8.97)$$

onde se usa a notação  $f_i = f(x_i, y_i)$ ,  $f_{i-1} = f(x_{i-1}, y_{i-1})$ , e assim por diante. Quando se avalia uma EDO de primeira ordem, outros métodos têm que ser usados para estimar  $y$  no segundo, no terceiro e no quarto ponto, e a fórmula na Eq. (8.97) pode ser usada a partir do quinto ponto.

#### 8.6.2 Método de Adams-Moulton

O método de Adams-Moulton é um método multipasso implícito usado na solução de EDOs de primeira ordem. Existem várias fórmulas de Adams-Moulton disponíveis para o cálculo de  $y_{i+1}$  usando a solução conhecida em dois ou mais pontos anteriores. As fórmulas são classificadas de acordo com a sua ordem, que corresponde ao número de pontos usados e também à ordem do erro de truncamento global do esquema. A fórmula de segunda ordem usa os pontos  $(x_i, y_i)$  e  $(x_{i-1}, y_{i-1})$ . A fórmula de terceira ordem usa os pontos  $(x_i, y_i)$ ,  $(x_{i-1}, y_{i-1})$  e  $(x_{i-2}, y_{i-2})$ , e assim por diante. A abordagem usada neste método é similar àquela empregada no método de Adams-Bashforth, onde a função  $f(x, y)$  na Eq. (8.93) é aproximada por um polinômio. A diferença entre os métodos está no fato de, no método de Adams-Moulton, os pontos usados na interpolação incluírem não apenas  $(x_i, y_i)$  e os pontos anteriores, mas também o ponto  $(x_{i+1}, y_{i+1})$  onde se deseja determinar a solução. Conseqüentemente, a variável  $y_{i+1}$  também aparece no lado direito da equação, o que torna o método implícito. A dedução das equações está além do escopo deste livro, mas fórmulas de segunda, terceira e quarta ordem são apresentadas a seguir.

**Método de Adams-Moulton de segunda ordem**

A fórmula de segunda ordem é:

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \quad (8.98)$$

Essa fórmula é uma forma implícita do método de Euler modificado.

**Método de Adams-Moulton de terceira ordem**

A fórmula de terceira ordem é:

$$y_{i+1} = y_i + \frac{h}{12}[5f(x_{i+1}, y_{i+1}) + 8f(x_i, y_i) - f(x_{i-1}, y_{i-1})] \quad (8.99)$$

**Método de Adams-Moulton de quarta ordem**

A fórmula de quarta ordem é:

$$y_{i+1} = y_i + \frac{h}{24}[9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}] \quad (8.100)$$

Estas fórmulas só podem ser usadas se as soluções nos pontos anteriores já forem conhecidas.

Os métodos de Adams-Moulton podem ser usados de duas maneiras. Se forem usados de forma exclusiva, eles devem ser resolvidos numericamente, já que a incógnita  $y_{i+1}$  aparece em ambos os lados das equações. Usualmente, no entanto, são usados em conjunção com outras equações nos chamados **métodos predictor-corretor**, que são apresentados na próxima seção.

## 8.7 MÉTODOS PREDITOR-CORRETOR

Os métodos predictor-corretor se referem a uma família de esquemas usados na solução de EDOs. Esses esquemas utilizam duas fórmulas, chamadas de **preditor** e **corretor**. O preditor é uma fórmula explícita usada para estimar a solução  $y_{i+1}$ . Com o preditor, o valor de  $y_{i+1}$  é calculado a partir da solução conhecida no ponto anterior ( $x_i, y_i$ ) (método de passo simples) ou em vários pontos anteriores (métodos multipasso). Uma vez encontrada uma estimativa para  $y_{i+1}$ , aplica-se o corretor. O corretor usa o valor estimado de  $y_{i+1}$  no lado direito de uma fórmula que, em circunstâncias normais, seria usada como uma fórmula implícita. Com isso, obtém-se no lado esquerdo da equação um novo valor para  $y_{i+1}$  que é mais preciso que o anterior. Portanto, a equação corretora, que é usualmente uma equação implícita, acaba sendo usada de uma maneira *explícita*, já que neste caso não é necessário resolver uma equação não-linear. Esse esquema usa os benefícios da fórmula implícita, ao mesmo tempo evitando as dificuldades associadas à solução direta de uma equação implícita. Além disso, a aplicação do corretor pode ser feita várias vezes com a substituição do novo valor de  $y_{i+1}$  no lado direito da equação corretora, o que resulta em valores de  $y_{i+1}$  cada vez mais refinados. O método predictor-corretor é resumido no algoritmo a seguir.

### Algoritmo para o método preditor-corretor

Dada uma solução nos pontos  $(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)$ .

1. Calcule  $y_{i+1}^{(1)}$  usando um método explícito.
2. Substitua a solução  $y_{i+1}^{(1)}$  obtida no primeiro passo, assim como os demais valores necessários, no lado direito de uma fórmula implícita para obter um valor refinado para  $y_{i+1}$ .
3. Repita o passo 2, substituindo novamente o valor refinado de  $y_{i+1}^{(1)}$  na fórmula implícita para obter um valor ainda mais refinado para  $y_{i+1}$ . O passo 2 pode ser repetido quantas vezes forem necessárias até que o nível desejado de precisão seja obtido, isto é, até que repetições adicionais não mudem a resposta obtida para  $y_{i+1}$  em um número específico de casas decimais.

O exemplo mais simples de um método preditor-corretor é o método de Euler modificado apresentado na Seção 8.3. Lembre que, neste método, uma primeira estimativa era calculada para  $y_{i+1}$  usando, como preditor, a fórmula explícita de Euler (Eq. (8.54)). A estimativa era então usada na obtenção de um valor mais preciso para  $y_{i+1}$  empregando a Eq. (8.56), que é o corretor. Além disso, conforme já mencionado nesta seção, esse método pode ser modificado para que a equação corretora possa ser aplicada várias vezes em cada passo. Com essa modificação, o método preditor-corretor de Euler é apresentado no algoritmo a seguir.

### Algoritmo para o método preditor-corretor de Euler modificado

Dada uma solução no ponto  $(x_i, y_i)$ .

1. Calcule uma primeira estimativa para  $y_{i+1}^{(1)}$  usando o método explícito de Euler como preditor:

$$y_{i+1}^{(1)} = y_i + f(x_i, y_i)h \quad (8.101)$$

2. Calcule melhores estimativas para  $y_{i+1}$  usando a Eq. (8.56) repetidamente como corretor:

$$y_{i+1}^{(k)} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k-1)})}{2}h \quad \text{para } k = 2, 3, \dots \quad (8.102)$$

3. Interrompa as iterações quando  $\left| \frac{y_{i+1}^{(k)} - y_{i+1}^{(k-1)}}{y_{i+1}^{(k-1)}} \right| \leq \varepsilon$

### Métodos preditor-corretor de Adams-Bashforth e Adams-Moulton

O método de Adams-Bashforth (Seção 8.6.1), que é um método explícito, e o método de Adams-Moulton (Seção 8.6.2), que é um método implícito, podem ser usados conjuntamente em um método preditor-corretor. Por exemplo, com as fórmulas de terceira ordem, a equação preditora que calcula a primeira estimativa para  $y_{i+1}^{(1)}$  é:

$$y_{i+1}^{(1)} = y_i + \frac{h}{12}[23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})] \quad (8.103)$$

A equação corretora é;

$$y_{i+1}^{(k)} = y_i + \frac{h}{12}[5f_{i+1}^{(k-1)} + 8f_i - f_{i-1}] \quad \text{para } k = 2, 3, \dots \quad (8.104)$$

onde  $f_{i+1}^{(k-1)} = f(x_{i+1}, y_{i+1}^{(k-1)})$ ,  $f_i = f(x_i, y_i)$  e  $f_{i-1} = f(x_{i-1}, y_{i-1})$ .

## 8.8 SISTEMAS DE EQUAÇÕES DIFERENCIAIS ORDINÁRIAS DE PRIMEIRA ORDEM

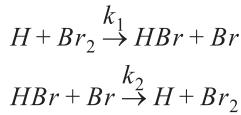
Equações diferenciais ordinárias são usadas para descrever, ou simular, processos e sistemas modelados por taxas de variação. Freqüentemente, esses processos e sistemas estão associados a muitas variáveis dependentes que afetam umas às outras. Nesses casos, é necessário resolver um sistema de EDOs de primeira ordem acopladas. Além disso, conforme discutido mais adiante na Secção 8.9, problemas de valor inicial envolvendo EDOs de segunda ordem ou de ordem superior são resolvidos a partir da conversão da equação em um sistema de equações de primeira ordem.

Um simples exemplo de aplicação de um sistema envolvendo duas EDOs de primeira ordem é a simulação do crescimento ou do decrescimento de duas populações interdependentes (um problema conhecido como predador-presa). Suponha que uma comunidade consista em  $N_L$  leões (predadores) e  $N_G$  gazelas (presas), com  $b$  e  $d$  representando as taxas de natalidade e mortalidade das respectivas espécies. Então, a taxa de variação (crescimento ou decrescimento) das populações de leões ( $L$ ) e de gazelas ( $G$ ) pode ser modelada pelas equações:

$$\begin{aligned} \frac{dN_L}{dt} &= b_L N_L N_G - d_L N_L \\ \frac{dN_G}{dt} &= b_G N_G - d_G N_G N_L \end{aligned} \quad (8.105)$$

As variáveis dependentes na Eq. (8.105) são  $N_L$  e  $N_G$ , e o tempo é a variável independente. As constantes  $b_L$  e  $b_G$  são as taxas de natalidade de leões e gazelas, respectivamente, e  $d_L$  e  $d_G$  são as suas respectivas taxas de mortalidade. A primeira equação indica que a taxa de variação da população de leões aumenta em função do produto  $N_L N_G$ , e decresce linearmente em função do número de leões. No caso das gazelas, a segunda equação diz que a taxa de variação de sua população aumenta linearmente em função do número de gazelas e decresce em função do produto  $N_L N_G$ . Uma vez especificada uma condição inicial (o número de leões e gazelas em  $t = 0$ ), a solução do sistema de equações dado na Eq. (8.105) fornece como resultado a população de ambas as espécies em função do tempo (ver Exemplo 8-11). As equações diferenciais na Eq. (8.105) são acopladas porque cada uma delas contém ambas as variáveis dependentes.

Reações químicas são freqüentemente escritas como “equações”, porém com setas no lugar do sinal de igual. Por exemplo, a reação:



representa os processos físicos associados à colisão de um átomo de hidrogênio ( $H$ ) com uma molécula de Bromo ( $Br_2$ ), resultando na formulação de uma nova molécula ( $HBr$ ) e de um átomo de Bromo ( $Br$ ). A taxa na qual ocorre essa reação é proporcional à quantidade de reagentes (neste caso, ( $H$ ) e ( $Br_2$ )). Portanto, pode-se escrever a taxa de produção de ( $HBr$ ) e ( $Br$ ) como:

$$\begin{aligned} \frac{dn_{Br}}{dt} &= k_1 n_H n_{Br_2} - k_2 n_{HBr} n_{Br} \\ \frac{dn_H}{dt} &= k_2 n_{HBr} n_{Br} - k_1 n_H n_{Br_2} \end{aligned} \quad (8.106)$$

onde  $n_i$  é o número de átomos ou moléculas de espécies  $i$  por unidade de volume, e  $k_1$  e  $k_2$  são os coeficientes das equações. Pode-se ver, a partir da Eq. (8.106), que o número de átomos por unidade de volume,  $n_{Br}$  e  $n_H$ , são as variáveis dependentes e o tempo é a variável independente.

### Fórmula geral de um sistema de EDOs de primeira ordem

Um sistema de  $n$  equações diferenciais de primeira ordem tem a forma:

$$\begin{aligned} \frac{dy_1}{dt} &= f_1(t, y_1, y_2, y_3, \dots, y_n) & y_1(t_1) &= Y_1 \\ \frac{dy_2}{dt} &= f_2(t, y_1, y_2, y_3, \dots, y_n) & y^{(2)}(t_1) &= Y_2 \\ &\dots \\ \frac{dy_n}{dt} &= f_n(t, y_1, y_2, y_3, \dots, y_n) & y^{(n)}(t_1) &= Y_3 \end{aligned} \quad (8.107)$$

onde  $t$  é a variável independente e  $y_1, y_2, y_3, \dots, y_n$  são as variáveis dependentes. Os lados direitos da Eq. (8.107) podem ser não-lineares e de complexidade arbitrária.

Alguns sistemas de EDOs de primeira ordem podem ser resolvidos com qualquer um dos métodos já discutidos. Em métodos de passo simples, a forma geral é a mesma das Eqs. (8.11) e (8.12), exceto pelo fato de a segunda equação ser aplicada em cada uma das variáveis dependentes.

$$t_{i+1} = t_i + h \quad (8.108)$$

$$\begin{aligned} y_{1,i+1} &= y_{1,i} + (Inclinação)_1 \cdot h \\ y_{2,i+1} &= y_{2,i} + (Inclinação)_2 \cdot h \\ &\dots \\ y_{n,i} &= y_{n,i} + (Inclinação)_n \cdot h \end{aligned} \quad (8.109)$$

onde  $h$  é o passo de integração e  $(Inclinação)_i$  é uma grandeza que estima o valor de  $\frac{dy_i}{dt}$  no intervalo de  $t_i$  a  $t_{i+1}$ .

As três seções a seguir apresentam os detalhes da solução de sistemas de EDOs usando o método explícito de Euler, o método de Euler modificado, e o método de Runge-Kutta de quarta ordem. Outros métodos explícitos podem ser usados da mesma maneira.

### 8.8.1 Solução de sistemas de EDOs de primeira ordem usando o método explícito de Euler

A aplicação do método de Euler na solução de sistemas de EDOs é mostrada primeiro para o caso de duas EDOs. Um sistema de duas EDOs de primeira ordem, com  $y$  e  $z$  como variáveis dependentes e  $x$  como variável independente, tem a forma:

$$\frac{dy}{dx} = f_1(x, y, z) \quad (8.110)$$

$$\frac{dz}{dx} = f_2(x, y, z) \quad (8.111)$$

para o domínio  $[a, b]$ , com condições iniciais  $y(a) = y_1$  e  $z(a) = z_1$ .

Para um sistema de duas EDOs, o método explícito de Euler é dado por:

$$x_{i+1} = x_i + h \quad (8.112)$$

$$y_{i+1} = y_i + f_1(x_i, y_i, z_i)h \quad (8.113)$$

$$z_{i+1} = z_i + f_2(x_i, y_i, z_i)h \quad (8.114)$$

O processo de solução começa com  $i = 1$  no primeiro ponto,  $x_1$ , onde os valores  $y_1$  e  $z_1$  são conhecidos. Em seguida, uma vez atribuído um valor a  $h$ , as Eqs. (8.112)-(8.114) são usadas para calcular o segundo ponto da solução (já que todas as grandezas no lado direito das equações são conhecidas). O processo continua então com  $i = 2, 3, \dots$  até o final do domínio da solução. Essa abordagem pode ser facilmente estendida para resolver um sistema de  $n$  EDOs.

### 8.8.2 Solução de sistemas de EDOs de primeira ordem usando o método de Runge-Kutta de segunda ordem (versão método de Euler modificado)

A aplicação do método de Runge-Kutta de segunda ordem na solução de um sistema de EDOs é mostrada para o caso de três EDOs. Um sistema de três EDOs de primeira ordem, com  $y$ ,  $z$  e  $w$  como variáveis dependentes e  $x$  como variável independente, tem a forma:

$$\frac{dy}{dx} = f_1(x, y, z, w) \quad (8.115)$$

$$\frac{dz}{dx} = f_2(x, y, z, w) \quad (8.116)$$

$$\frac{dw}{dx} = f_3(x, y, z, w) \quad (8.117)$$

para o domínio  $[a, b]$ , com condições iniciais  $y(a) = y_1$ ,  $z(a) = z_1$  e  $w(a) = w_1$ .

O sistema é solucionado usando o método de Runge-Kutta de segunda ordem na versão de Euler modificada; veja as Eqs. (8.63) e (8.64) na Seção 8.5.1. Quando essa formulação é usada para resolver uma equação, o valor de  $y_{i+1}$  em cada intervalo é calculado em três passos. No primeiro passo, calcula-se o valor de  $K_1$ ; no segundo passo, o valor de  $K_2$  é calculado usando  $K_1$  do primeiro passo; finalmente, no terceiro passo,  $K_1$  e  $K_2$  são usados no cálculo de  $y_{i+1}$ . Quando se resolve um sistema de equações, esse processo é aplicado paralelamente em cada uma das equações. Isso significa que, primeiramente, calcula-se o  $K_1$  de cada uma das EDOs. Em seguida, os valores de  $K_1$  são usados no cálculo dos  $K_2$ s de todas as equações. Finalmente, uma vez conhecidos os dois  $K$ s de cada equação, calcula-se o valor de cada variável independente em  $x_{i+1}$ .

Para o sistema de três EDOs dado nas Eqs. (8.115)-(8.117), o processo de solução em cada passo começa com o cálculo da variável independente no final do passo:

$$x_{i+1} = x_i + h \quad (8.118)$$

Em seguida, são calculados os valores de  $K_1$  correspondentes a cada uma das EDOs:

$$\begin{aligned} K_{y,1} &= f_1(x_i, y_i, z_i, w_i) \\ K_{z,1} &= f_2(x_i, y_i, z_i, w_i) \\ K_{w,1} &= f_3(x_i, y_i, z_i, w_i) \end{aligned} \quad (8.119)$$

onde  $K_{y,1}$ ,  $K_{z,1}$  e  $K_{w,1}$  são os termos  $K_1$  da primeira, da segunda e da terceira EDO, respectivamente. Em seguida, são calculados os termos  $K_2$  de cada uma das EDOs:

$$\begin{aligned} K_{y,2} &= f_1(x_i + h, y_i + K_{y,1}h, z_i + K_{z,1}h, w_i + K_{w,1}h) \\ K_{z,2} &= f_2(x_i + h, y_i + K_{y,1}h, z_i + K_{z,1}h, w_i + K_{w,1}h) \\ K_{w,2} &= f_3(x_i + h, y_i + K_{y,1}h, z_i + K_{z,1}h, w_i + K_{w,1}h) \end{aligned} \quad (8.120)$$

onde  $K_{y,2}$ ,  $K_{z,2}$  e  $K_{w,2}$  são os termos  $K_2$  da primeira, da segunda e da terceira EDO, respectivamente. Finalmente, os valores das três variáveis dependentes no ponto  $x_{i+1}$  são calculados usando:

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{2}(K_{y,1} + K_{y,2})h \\ z_{i+1} &= z_i + \frac{1}{2}(K_{z,1} + K_{z,2})h \\ w_{i+1} &= w_i + \frac{1}{2}(K_{w,1} + K_{w,2})h \end{aligned} \quad (8.121)$$

A generalização para um sistema de  $n$  equações é apresentada no seguinte algoritmo.

**Algoritmo para resolver um sistema de EDOs de primeira ordem com o método de Runge-Kutta de segunda ordem (versão método de Euler modificado)**

Um sistema de  $n$  EDOs com variável independente  $x$  e variáveis dependentes  $y_1, y_2, \dots, y_n$  é escrito como:

$$\begin{aligned}\frac{dy_1}{dx} &= f_1(x, y_1, \dots, y_n) \\ &\dots \\ \frac{dy_n}{dx} &= f_n(x, y_1, \dots, y_n)\end{aligned}\tag{8.122}$$

ao longo do intervalo  $[a, b]$ , com as condições iniciais:

$$\begin{aligned}y_1|_{x=a} &= Y_1 \\ &\dots \\ y_n|_{x=a} &= Y_n\end{aligned}$$

1. Escolha um passo de integração  $h = (b - a)/N$ , onde  $N$  é o número de passos.
  2. Para  $i = 1, \dots, N$ ,
- Calcule o próximo valor da variável independente:

$$x_{i+1} = x_i + h\tag{8.123}$$

- Usando  $x_i$  e as soluções  $y_{1,i}, \dots, y_{n,i}$  conhecidas, calcule  $K_1$  para cada EDO:

$$\begin{aligned}K_{1,1} &= f_1(x_i, y_{1,i}, \dots, y_{n,i}) \\ &\dots \\ K_{n,1} &= f_n(x_i, y_{1,i}, \dots, y_{n,i})\end{aligned}\tag{8.124}$$

- Usando  $x_i$ , as soluções  $y_{1,i}, \dots, y_{n,i}$  conhecidas e os valores obtidos para  $K_1$ , calcule  $K_2$  para cada EDO:

$$\begin{aligned}K_{1,2} &= f_1(x_i + h, y_{1,i} + K_{1,1}h, \dots, y_{n,i} + K_{n,1}h) \\ &\dots\end{aligned}\tag{8.125}$$

$$K_{n,2} = f_n(x_i + h, y_{1,i} + K_{1,1}h, \dots, y_{n,i} + K_{n,1}h)$$

- Usando as grandezas  $y_{1,i}, \dots, y_{n,i}$ ,  $K_{1,1}, \dots, K_{n,1}$  e  $K_{1,2}, \dots, K_{n,2}$  conhecidas, calcule a solução  $y_{1,i+1}, \dots, y_{n,i+1}$ :

$$\begin{aligned}y_{1,i+1} &= y_{1,i} + \frac{1}{2}(K_{1,1} + K_{1,2})h \\ &\dots \\ y_{n,i+1} &= y_{n,i} + \frac{1}{2}(K_{n,1} + K_{n,2})h\end{aligned}\tag{8.126}$$

O Exemplo 8-7 mostra a aplicação do método explícito de Euler e do método de Runge-Kutta de segunda ordem na solução de um sistema de equações.

**Exemplo 8-7: Solução de um sistema de duas EDOs de primeira ordem usando o método explícito de Euler e o método de Runge-Kutta de segunda ordem**

Considere o seguinte problema de valor inicial formado por duas EDOs de primeira ordem:

$$\frac{dy}{dx} = (-y + z)e^{(1-x)} + 0,5y \text{ com condição inicial } y(0) = 3 \quad (8.127)$$

$$\frac{dz}{dx} = y - z^2 \text{ com condição inicial } z(0) = 0,2 \quad (8.128)$$

no domínio  $x = 0$  a  $x = 3$ .

- (a) Resolva o sistema manualmente (os primeiros três passos) usando o método explícito de Euler, com  $h = 0,25$ .
- (b) Resolva o sistema manualmente (os primeiros dois passos) usando o método de Runge-Kutta de segunda ordem (versão método modificado de Euler), com  $h = 0,25$ .
- (c) Escreva um programa no MATLAB que resolva o sistema usando o método de Runge-Kutta de segunda ordem (versão método modificado de Euler), com  $h = 0,1$ .

Mostre em um gráfico os resultados obtidos nas três letras.

## SOLUÇÃO

- (a) **Solução manual usando o método de Euler** O primeiro ponto da solução é  $x_1 = 0$ ,  $y_1 = 3$  e  $z_1 = 0,2$ , que corresponde às condições iniciais. No primeiro ponto,  $i = 1$ .

O restante da solução é determinado usando as Eqs. (8.112)-(8.114). No presente problema, essas equações têm a forma:

$$x_{i+1} = x_i + h = x_i + 0,25 \quad (8.129)$$

$$y_{i+1} = y_i + f_1(x_i, y_i, z_i)h = y_i + [(-y_i + z_i)e^{(1-x_i)} + 0,5y_i]h \quad (8.130)$$

$$z_{i+1} = z_i + f_2(x_i, y_i, z_i)h = z_i + [y_i - z_i^2]h \quad (8.131)$$

A seguir, as Eqs. (8.129)-(8.131) são aplicadas três vezes com  $i = 1, 2, 3$ .

**Primeiro passo** No primeiro passo,  $i = 1$ . As Eqs. (8.129)-(8.131) resultam em:

$$x_2 = x_1 + 0,25 = 0 + 0,25 = 0,25$$

$$y_2 = y_1 + [(-y_1 + z_1)e^{(1-x_1)} + 0,5y_1]h = 3 + [(-3 + 0,2)e^{(1-0)} + 0,5 \cdot 3]0,25 = 1,472$$

$$z_2 = z_1 + [y_1 - z_1^2]h = 0,2 + [3 - 0,2^2]0,25 = 0,94$$

O segundo ponto da solução é  $x_2 = 0,25$ ,  $y_2 = 1,472$  e  $z_2 = 0,94$ .

**Segundo passo** No segundo passo,  $i = 2$ . As Eqs. (8.129)-(8.131) resultam em:

$$x_3 = x_2 + 0,25 = 0,25 + 0,25 = 0,5$$

$$y_3 = y_2 + [(-y_2 + z_2)e^{(1-x_2)} + 0,5y_2]h = 1,472 + [(-1,472 + 0,94)e^{(1-0,25)} + 0,5 \cdot 1,472]0,25 = 1,374$$

$$z_3 = z_2 + [y_2 - z_2^2]h = 0,94 + [1,472 - 0,94^2]0,25 = 1,0871$$

O terceiro ponto da solução é  $x_3 = 0,5$ ,  $y_3 = 1,374$  e  $z_3 = 1,087$ .

**Terceiro passo** No terceiro passo,  $i = 3$ . As Eqs. (8.129)-(8.131) resultam em:

$$x_4 = x_3 + 0,25 = 0,5 + 0,25 = 0,75$$

$$y_4 = y_3 + [(-y_3 + z_3)e^{(1-x_3)} + 0,5y_3]h = 1,374 + [(-1,374 + 1,087)e^{(1-0,5)} + 0,5 \cdot 1,374]0,25 = 1,427$$

$$z_4 = z_3 + [y_3 - z_3^2]h = 1,087 + [1,374 - 1,087^2]0,25 = 1,135$$

O quarto ponto da solução é  $x_4 = 0,75$ ,  $y_4 = 1,427$  e  $z_4 = 1,135$ .

A solução obtida com o método explícito de Euler é mostrada na figura no final da solução do problema, na letra (c).

(b) **Solução manual usando o método de Runge-Kutta de segunda ordem:** O primeiro ponto da solução é  $x_1 = 0$ ,  $y_1 = 3$  e  $z_1 = 0,2$ , que corresponde às condições iniciais. No primeiro ponto,  $i = 1$ .

O restante da solução é determinado usando as Eqs. (8.124)-(8.126). No presente problema, essas equações têm a forma:

$$x_{i+1} = x_i + h = x_i + 0,25 \quad (8.132)$$

$$\begin{aligned} K_{y,1} &= (-y_i + z_i)e^{(1-x_i)} + 0,5y_i \\ K_{z,1} &= y_i - z_i^2 \end{aligned} \quad (8.133)$$

$$\begin{aligned} K_{y,2} &= (-y_{est} + z_{est})e^{(1-x_{i+1})} + 0,5y_{est} \\ K_{z,2} &= y_{est} - z_{est}^2 \end{aligned} \quad (8.134)$$

onde:  $y_{est} = y_i + K_{y,1}h$  e  $z_{est} = z_i + K_{z,1}h$ .

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{2}(K_{y,1} + K_{y,2})h \\ z_{i+1} &= z_i + \frac{1}{2}(K_{z,1} + K_{z,2})h \end{aligned} \quad (8.135)$$

**Nota:** Para simplificar as equações, são introduzidas as grandezas  $y_{est}$  e  $z_{est}$ . Elas são calculadas após a determinação de  $K_{y,1}$  e  $K_{z,1}$ , e são usadas na avaliação de  $K_{y,2}$  e  $K_{z,2}$ .

A seguir, as Eqs. (8.132)-(8.135) são aplicadas três vezes com  $i = 1, 2, 3$ .

**Primeiro passo** No primeiro passo,  $i = 1$ . As Eqs. (8.132)-(8.135) resultam em:

$$x_2 = x_1 + 0,25 = 0 + 0,25 = 0,25$$

$$K_{y,1} = (-y_1 + z_1)e^{(1-x_1)} + 0,5y_1 = (-3 + 0,2)e^{(1-0)} + 0,5 \cdot 3 = -6,111$$

$$K_{z,1} = y_1 - z_1^2 = 3 - 0,2^2 = 2,96$$

$$y_{est} = y_1 + K_{y,1}h = 3 + (-6,111)0,25 = 1,472$$

$$z_{est} = z_1 + K_{z,1}h = 0,2 + 2,96 \cdot 0,25 = 0,94$$

$$K_{y,2} = (-y_{est} + z_{est})e^{(1-x_{i+1})} + 0,5y_{est} = (-1,472 + 0,94)e^{(1-0,25)} + 0,5 \cdot 1,472 = -0,3902$$

$$K_{z,2} = y_{est} - z_{est}^2 = 1,472 - 0,94^2 = 0,5884$$

$$y_2 = y_1 + \frac{1}{2}(K_{y,1} + K_{y,2})h = 3 + \frac{1}{2}[-6,111 + (-0,3902)]0,25 = 2,187$$

$$z_2 = z_1 + \frac{1}{2}(K_{z,1} + K_{z,2})h = 0,2 + \frac{1}{2}[2,96 + 0,5884]0,25 = 0,6436$$

O segundo ponto da solução é  $x_2 = 0,25$ ,  $y_2 = 2,187$  e  $z_2 = 0,6436$ .

**Segundo passo** No segundo passo,  $i = 2$ . As Eqs. (8.132)-(8.135) resultam em:

$$x_3 = x_2 + 0,25 = 0,25 + 0,25 = 0,5$$

$$K_{y,1} = (-y_2 + z_2)e^{(1-x_2)} + 0,5y_2 = (-2,187 + 0,6436)e^{(1-0,25)} + 0,5 \cdot 2,187 = -2,173$$

$$K_{z,1} = y_2 - z_2^2 = 2,187 - 0,6436^2 = 1,773$$

$$y_{est} = y_2 + K_{y,1}h = 2,187 + (-2,173)0,25 = 1,644$$

$$z_{est} = z_2 + K_{z,1}h = 0,6436 + 1,773 \cdot 0,25 = 1,087$$

$$K_{y,2} = (-y_{est} + z_{est})e^{(1-x_{i+1})} + 0,5y_{est} = (-1,644 + 1,087)e^{(1-0,5)} + 0,5 \cdot 1,644 = -0,09634$$

$$K_{z,2} = y_{est} - z_{est}^2 = 1,644 - 1,087^2 = 0,4624$$

$$y_3 = y_2 + \frac{1}{2}(K_{y,1} + K_{y,2})h = 2,187 + \frac{1}{2}[-2,173 + (-0,09634)]0,25 = 1,903$$

$$z_3 = z_2 + \frac{1}{2}(K_{z,1} + K_{z,2})h = 0,6436 + \frac{1}{2}[1,773 + 0,4624]0,25 = 0,9230$$

O terceiro ponto da solução é  $x_3 = 0,5$ ,  $y_3 = 1,903$  e  $z_3 = 0,9230$ .

A solução obtida com o método de Runge-Kutta de segunda ordem é mostrada na figura no final da solução do problema, na letra (c).

(c) *Solução usando um programa de computador que utiliza o método de Runge-Kutta de segunda ordem.*

Primeiramente, escreve-se uma função chamada Sis2EDOsRK2 que resolve um sistema de duas EDOs usando o método de Runge-Kutta de segunda ordem. Essa função usa duas outras funções, listadas no programa como EDO1 e EDO2, que calculam os valores de  $dy/dx$  e  $dz/dx$  em cada passo. As funções são chamadas de edoExemplo7dydx e edoExemplo7dzdx. A solução é obtida a partir de um programa escrito no MATLAB. O programa também traça um gráfico que mostra a solução da letra (c) e também as soluções calculadas nas letras (a) e (b).

**Programa 8-5 Função criada pelo usuário. Solução de um sistema de duas EDOs usando o método de Runge-Kutta de segunda ordem (versão Euler modificado).**

```
function [x, y, z] = Sis2EDOsRK2(EDO1, EDO2, a, b, h, yINI, zINI)
% Sis2EDOsRK2 resolve um sistema de duas EDOs com valores iniciais usando o
% método de Runge-Kutta de segunda ordem.
% A variável independente é x, e as variáveis dependentes são y e z.
% Variáveis de entrada:
% EDO1 Nome (string) de uma função em arquivo que calcula dy/dx.
% EDO2 Nome (string) de uma função em arquivo que calcula dz/dx.
% a Primeiro valor de x.
% b Último valor de x.
% h Tamanho do incremento.
% yINI Valor inicial de y.
% zINI Valor inicial de z.
% Variáveis de saída:
% x Vetor com a coordenada x dos pontos da solução.
% y Vetor com a coordenada y dos pontos da solução.
% z Vetor com a coordenada z dos pontos da solução.
```

$x(1) = a$ ;  $y(1) = yINI$ ;  $z(1) = zINI$ ;

Atribui o valor inicial a  $x(1)$ ,  $y(1)$  e  $z(1)$ .

$N = (b - a)/h$ ;

Determina o número de passos.

for  $i = 1:N$

$x(i + 1) = x(i) + h$ ;

Calcula o próximo valor da variável independente.

$Ky1 = feval(EDO1, x(i), y(i), z(i))$ ;

Calcula os  $K_1$ s, Eq. (8.133).

$Kz1 = feval(EDO2, x(i), y(i), z(i))$ ;

```

Ky2 = feval(EDO1,x(i + 1),y(i) + Ky1*h,z(i) + Kz1*h);
Kz2 = feval(EDO2,x(i + 1),y(i) + Ky1*h,z(i) + Kz1*h);
y(i + 1) = y(i) + (Ky1 + Ky2)*h/2;
z(i + 1) = z(i) + (Kz1 + Kz2)*h/2;
end

```

Calcula os  $K_2$ s, Eq. (8.134).

Calcula o próximo valor das variáveis dependentes, Eq. (8.135).

A seguir, são listadas as funções em arquivo que calculam os valores de  $dy/dx$ , Eq. (8.127), e  $dz/dx$ , Eq. (8.128). As funções são chamadas de `edoExemplo7dydx` e `edoExemplo7dzdx`.

```

function dydx = edoExemplo7dydx(x,y,z)
dydx = (-y + z)*exp(1 - x) + 0.5*y;

```

```

function dzdx = edoExemplo7dzdx(x,y,z)
dzdx = y - z^2;

```

O programa de MATLAB que utiliza as funções acima para resolver o problema é listado a seguir. O programa também mostra um gráfico com as soluções obtidas nas letras (a), (b) e (c).

% Solução do Exemplo 7, Capítulo 8

clear all

a = 0; b = 3; yINI = 3; zINI = 0.2; h = 0.1;

[x, y, z] = Sis2EDOsRK2 ('edoExemplo7dydx','edoExemplo7dzdx',a,b,h,yINI,zINI);

% Dados da letra (a)

xa = [0 0.25 0.5 0.75];

ya = [3 1.472 1.374 1.427];

za = [0.2 0.94 1.087 1.135];

% Dados da letra (b)

xb = [0 0.25 0.5];

yb = [3 2.187 1.903];

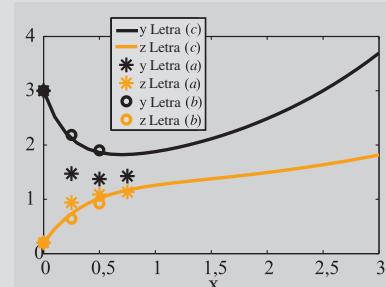
zb = [0.2 0.6436 0.9230];

plot(x,y,'-k',x,z,'-r',xa,ya,'\*k',xa,za,'\*r',xb,yb,'ok',xb,zb,'or')

axis([0 3 0 4])

Usa a função `SisEDOsRK2` para resolver o sistema.

A figura ao lado mostra a solução numérica obtida na letra (c) em todo o domínio (linhas contínuas), os primeiros quatro pontos da letra (a) (asteriscos) e os três pontos calculados na letra (b) (círculos).



### 8.8.3 Solução de sistemas de EDOs de primeira ordem usando o método de Runge-Kutta de quarta ordem clássico

A aplicação do método de Runge-Kutta de quarta ordem na solução de um sistema de EDOs é, em princípio, igual à aplicação do método de segunda ordem descrito na seção anterior. A única diferença está no fato de o método de Runge-Kutta de quarta ordem utilizar quatro grandezas para resolver cada EDO,  $K_1$ ,  $K_2$ ,  $K_3$  e  $K_4$ , que devem ser calculadas antes da avaliação das variáveis dependentes em  $x_{i+1}$  (ver Eqs. (8.86) e (8.87) na Seção 8.5.3). Novamente, como as equações estão acopladas, os  $K$ s devem ser calculados paralelamente. Isso significa que os valores de todos os  $K_1$ s devem ser calculados primeiro. Em seguida, são calculados todos os  $K_2$ s, depois todos os  $K_3$ s, e finalmente todos os  $K_4$ s. Assim que os quatro  $K$ s de cada EDO forem conhecidos, os valores das variáveis dependentes podem ser calculados.

Para que a necessidade de se escrever um grande número de equações seja evitada, a aplicação do método de Runge-Kutta de quarta ordem é apresentada aqui para o caso de um sistema com três EDOs de primeira ordem. A abordagem empregada pode ser facilmente estendida à solução de sistemas com mais EDOs.

Um sistema com três EDOs de primeira ordem, tendo  $y$ ,  $z$  e  $w$  como variáveis dependentes e  $x$  como a variável independente, tem a forma:

$$\frac{dy}{dx} = f_1(x, y, z, w) \quad (8.136)$$

$$\frac{dz}{dx} = f_2(x, y, z, w) \quad (8.137)$$

$$\frac{dw}{dx} = f_3(x, y, z, w) \quad (8.138)$$

no domínio  $[a, b]$ , com condições iniciais:  $y(a) = y_1$ ,  $z(a) = z_1$  e  $w(a) = w_1$ .

O processo de solução começa com o cálculo do valor de  $K_1$  para cada EDO:

$$\begin{aligned} K_{y,1} &= f_1(x_i, y_i, z_i, w_i) \\ K_{z,1} &= f_2(x_i, y_i, z_i, w_i) \\ K_{w,1} &= f_3(x_i, y_i, z_i, w_i) \end{aligned} \quad (8.139)$$

Em seguida, calcula-se o valor de  $K_2$  para cada uma das equações:

$$\begin{aligned} K_{y,2} &= f_1\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,1}h, z_i + \frac{1}{2}K_{z,1}h, w_i + \frac{1}{2}K_{w,1}h\right) \\ K_{z,2} &= f_2\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,1}h, z_i + \frac{1}{2}K_{z,1}h, w_i + \frac{1}{2}K_{w,1}h\right) \\ K_{w,2} &= f_3\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,1}h, z_i + \frac{1}{2}K_{z,1}h, w_i + \frac{1}{2}K_{w,1}h\right) \end{aligned} \quad (8.140)$$

Esse passo é sucedido do cálculo de  $K_3$ :

$$\begin{aligned} K_{y,3} &= f_1\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,2}h, z_i + \frac{1}{2}K_{z,2}h, w_i + \frac{1}{2}K_{w,2}h\right) \\ K_{z,3} &= f_2\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,2}h, z_i + \frac{1}{2}K_{z,2}h, w_i + \frac{1}{2}K_{w,2}h\right) \\ K_{w,3} &= f_3\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,2}h, z_i + \frac{1}{2}K_{z,2}h, w_i + \frac{1}{2}K_{w,2}h\right) \end{aligned} \quad (8.141)$$

A última grandeza a ser calculada é  $K_4$ :

$$\begin{aligned} K_{y,4} &= f_1(x_i + h, y_i + K_{y,3}h, z_i + K_{z,3}h, w_i + K_{w,3}h) \\ K_{z,4} &= f_2(x_i + h, y_i + K_{y,3}h, z_i + K_{z,3}h, w_i + K_{w,3}h) \\ K_{w,4} &= f_3(x_i + h, y_i + K_{y,3}h, z_i + K_{z,3}h, w_i + K_{w,3}h) \end{aligned} \quad (8.142)$$

Uma vez determinados os quatro  $K$ s referentes a cada uma das EDOs, o valor da variável dependente (solução) em  $x = x_{i+1}$  é calculado com:

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}(K_{y,1} + 2K_{y,2} + 2K_{y,3} + K_{y,4})h \\ z_{i+1} &= z_i + \frac{1}{6}(K_{z,1} + 2K_{z,2} + 2K_{z,3} + K_{z,4})h \\ w_{i+1} &= w_i + \frac{1}{6}(K_{w,1} + 2K_{w,2} + 2K_{w,3} + K_{w,4})h \end{aligned} \quad (8.143)$$

Uma função criada no MATLAB para resolver um sistema de duas EDOs de primeira ordem usando o método de Runge-Kutta de quarta ordem é dada no Exemplo 8-8.

## 8.9 SOLUÇÃO DE PROBLEMAS DE VALOR INICIAL DE ORDEM MAIS ELEVADA

### PVIs de Segunda Ordem

Uma EDO de segunda ordem tendo  $x$  e  $y$  como variáveis independente e dependente, respectivamente, pode ser escrita na forma:

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \quad (8.144)$$

ao longo do domínio  $[a, b]$ . Tal equação pode ser resolvida se duas restrições forem especificadas. Quando duas restrições são especificadas em um único valor de  $x$ , o problema é classificado como um problema de valor inicial (PVI). As duas condições iniciais são os valores de  $y$  e da derivada primeira de  $y$ ,  $dy/dx$ , no primeiro ponto,  $a$ , do domínio da solução. Se  $A$  e  $B$  são esses valores, as condições iniciais podem ser escritas como:

$$y(a) = A \quad \text{e} \quad \left.\frac{dy}{dx}\right|_{x=a} = B \quad (8.145)$$

Esse tipo de EDO de segunda ordem pode ser transformado em um sistema de duas EDOs de primeira ordem que podem ser resolvidas numericamente com os métodos apresentados na Seção 8.8. A solução de equações de segunda ordem (e de ordem superior) que não constituem problemas de valor inicial é apresentada no Capítulo 9.

A transformação de uma EDO de segunda ordem em um sistema com duas EDOs de primeira ordem é feita com a introdução de uma nova variável dependente,  $w$ , tal que:

$$w = \frac{dy}{dx} \quad (8.146)$$

e

$$\frac{dw}{dx} = \frac{d^2y}{dx^2} \quad (8.147)$$

Com essas definições, a EDO de segunda ordem da Eq. (8.144), com as condições iniciais descritas na Eq. (8.145), pode ser escrita como o seguinte sistema de duas EDOs de primeira ordem:

$$\frac{dy}{dx} = w \text{ com } y(a) = A \quad (8.148)$$

$$\frac{dw}{dx} = f(x, y, w) \text{ com } w(a) = B \quad (8.149)$$

Uma vez escrito dessa maneira, o sistema pode ser resolvido numericamente usando qualquer um dos métodos já apresentados.

Como um exemplo específico, considere a EDO de segunda ordem:

$$\frac{d^2y}{dt^2} = -0,2\frac{dy}{dt} - 2y + 3 \operatorname{sen}(t) \text{ com } y(0) = 3 \text{ e } \left.\frac{dy}{dt}\right|_{t=0} = 1,5 \quad (8.150)$$

Definindo  $w = \frac{dy}{dt}$ , de forma que  $\frac{dw}{dt} = \frac{d^2y}{dt^2}$ , pode-se escrever a Eq. (8.150) como o seguinte sistema de duas EDOs de primeira ordem:

$$\frac{dy}{dt} = w \text{ com } y(0) = 3 \quad (8.151)$$

$$\frac{dw}{dt} = -0,2w - 2y + 3 \operatorname{sen}(t) \text{ com } w(0) = 1,5 \quad (8.152)$$

Os métodos da Seção 8.8 podem agora ser usados para resolver numericamente o sistema formado pelas Eqs. (8.151) e (8.152).

A solução de um problema de valor inicial de segunda ordem é ilustrada no Exemplo 8-8, onde se obtém uma solução numérica para o movimento de um pêndulo. O problema é resolvido para o caso onde há amortecimento e grandes deslocamentos do pêndulo. Note que esse problema poderia ser resolvido analiticamente para pequenos deslocamentos. No entanto, o caso não-linear de grandes deslocamentos só pode ser resolvido numericamente.

### Exemplo 8-8: Movimento não-linear amortecido de um pêndulo

Um pêndulo é modelado por uma massa presa a uma haste rígida de peso desprezível. De acordo com a segunda lei de Newton, quando o pêndulo balança de um lado para o outro, a soma das forças que atuam na massa é igual à massa do pêndulo vezes a aceleração (veja o diagrama de corpo livre na figura). Escrevendo a equação de equilíbrio na direção tangencial, obtém-se:

$$\Sigma F_t = -cL \frac{d\theta}{dt} - mg \sin \theta = mL \frac{d^2\theta}{dt^2} \quad (8.153)$$

onde  $\theta$  é o ângulo do pêndulo em relação ao eixo vertical, conforme mostrado na figura,  $c = 0,16$  ( $N \cdot s$ )/m é o coeficiente de amortecimento,  $m = 0,5$  kg é a massa,  $L = 1,2$  m é o comprimento e  $g = 9,81$   $m/s^2$  é a aceleração da gravidade.

A Eq. (8.153) pode ser reescrita como a seguinte equação diferencial de segunda ordem:

$$\frac{d^2\theta}{dt^2} = -\frac{c}{m} \frac{d\theta}{dt} - \frac{g}{L} \sin \theta \quad (8.154)$$

O pêndulo é inicialmente deslocado em  $\theta = 90^\circ$ , e então, em  $t = 0$ , ele é solto, tendo velocidade inicial nula ( $\frac{d\theta}{dt} = 0$ ). Determine o ângulo do pêndulo em função do tempo,  $\theta(t)$ , nos primeiros 18 s após o instante em que ele é solto.

### SOLUÇÃO

Para resolver a Eq. (8.154), introduz-se uma nova variável dependente,  $w$ , de forma que:

$$w = \frac{d\theta}{dt} \quad \text{e} \quad \frac{dw}{dt} = \frac{d^2\theta}{dt^2}$$

Com essas definições, a Eq. (8.154) pode ser reescrita como o seguinte sistema de duas EDOs de segunda ordem:

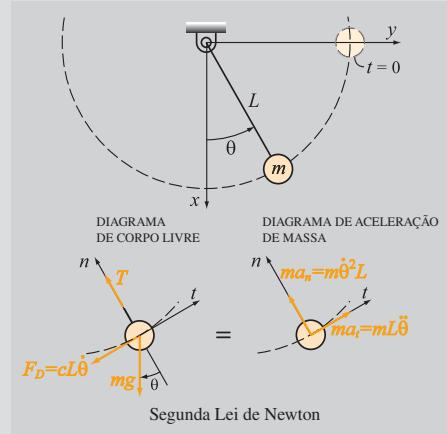
$$\frac{d\theta}{dt} = w \quad \text{com a condição inicial } \theta(0) = \frac{\pi}{2} \quad (8.155)$$

$$\frac{dw}{dt} = -\frac{c}{m}w - \frac{g}{L} \sin \theta \quad \text{com a condição inicial } w(0) = 0 \quad (8.156)$$

Esse sistema de EDOs é resolvido com uma função criada no MATLAB, denominada  $[t, x, y] = \text{Sis2EDOsRK4}(EDO1, EDO2, a, b, h, x1, y1)$  e listada a seguir. A função resolve um sistema de duas EDOs usando o método de Runge-Kutta de quarta ordem. Ela emprega duas funções adicionais que calculam o lado direito de cada EDO.

#### Programa 8-6 Função criada pelo usuário. Solução de um sistema de duas EDOs de primeira ordem usando o método de Runge-Kutta de quarta ordem.

```
function [t, x, y] = Sis2EDOsRK4(EDO1, EDO2, a, b, h, x1, y1)
% Sis2EDOsRK4 resolve um sistema de duas EDOs com valores iniciais usando o
% método de Runge-Kutta de quarta ordem.
% A variável independente é t, e as variáveis dependentes são x e y.
% Variáveis de entrada:
```



```
% EDO1 Nome (string) de uma função em arquivo que calcula dx/dt.
% EDO2 Nome (string) de uma função em arquivo que calcula dy/dt.
% a Primeiro valor de t.
% b Último valor de t.
% h Tamanho do incremento.
% x1 Valor inicial de x.
% y1 Valor inicial de y.
% Variáveis de saída:
% t Vetor com a coordenada t dos pontos da solução.
% x Vetor com a coordenada x dos pontos da solução.
% y Vetor com a coordenada y dos pontos da solução.
```

$t(1) = a; x(1) = x1; y(1) = y1;$

Atribui os valores iniciais ao primeiro ponto da solução.

$n = (b - a)/h;$

Calcula o número de passos.

for  $i = 1:n$

$t(i+1) = t(i) + h;$

$tm = t(i) + h/2;$

$Kx1 = feval(EDO1,t(i),x(i),y(i));$

$Ky1 = feval(EDO2,t(i),x(i),y(i));$

$Kx2 = feval(EDO1,tm,x(i) + Kx1*h/2,y(i) + Ky1*h/2);$

$Ky2 = feval(EDO2,tm,x(i) + Kx1*h/2,y(i) + Ky1*h/2);$

$Kx3 = feval(EDO1,tm,x(i) + Kx2*h/2,y(i) + Ky2*h/2);$

$Ky3 = feval(EDO2,tm,x(i) + Kx2*h/2,y(i) + Ky2*h/2);$

$Kx4 = feval(EDO1,t(i + 1),x(i) + Kx3*h,y(i) + Ky3*h);$

$Ky4 = feval(EDO2,t(i + 1),x(i) + Kx3*h,y(i) + Ky3*h);$

$x(i+1) = x(i) + (Kx1 + 2*Kx2 + 2*Kx3 + Kx4)*h/6;$

$y(i+1) = y(i) + (Ky1 + 2*Ky2 + 2*Ky3 + Ky4)*h/6;$

end

Para resolver o sistema de EDOs nas Eqs. (8.155) e (8.156), são escritas duas funções. Uma, chamada PenduloDtetaDt, calcula  $\frac{d\theta}{dt}$ , e a outra, chamada PenduloDwDt, calcula  $\frac{dw}{dt}$ :

function dxdt = PenduloDtetaDt(t,x,y)

$dxdt = y;$

Eq. (8.155).

function dydt = PenduloDwDt(t,x,y)

$c = 0.16; m = 0.5; g = 9.81; L = 1.2;$

$dydt = -(c/m)*y - (g/L)*\sin(x);$

Eq. (8.156).

As funções acima são usadas no programa a seguir, que resolve no MATLAB o problema do pêndulo. O programa também traça um gráfico ( $\theta$  versus tempo) com a solução.

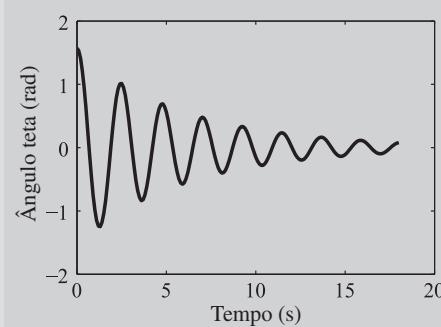
[t, x, y] = Sis2EDOsRK4('PenduloDtetaDt','PenduloDwDt',0,18,0.1,pi/2,0);

plot(t,x)

xlabel('Tempo(s)')

ylabel('Ângulo theta (rad)')

Quando o arquivo é executado, a seguinte figura é exibida:



### PVIs de Ordem Superior

EDOs de terceira e quarta ordem podem ser convertidas em sistemas de EDOs de primeira ordem de maneira similar à feita com EDOs de segunda ordem. Um PVI de  $n$ -ésima ordem tem a forma:

$$\frac{d^n y}{dx^n} = f\left(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, \dots, \frac{d^{(n-1)} y}{dx^{(n-1)}}\right) \quad \text{para } a \leq x \leq b \quad (8.157)$$

com condições iniciais:

$$y(a) = A_1, \quad \left.\frac{dy}{dx}\right|_{x=a} = A_2, \quad \left.\frac{d^2 y}{dx^2}\right|_{x=a} = A_3, \quad \dots, \quad \left.\frac{d^{(n-1)} y}{dx^{(n-1)}}\right|_{x=a} = A_{n-1} \quad (8.158)$$

Uma EDO de  $n$ -ésima ordem pode ser transformada em um sistema de  $n$  EDOs de primeira ordem com a introdução de  $n - 1$  novas variáveis dependentes,  $w_1, w_2, w_3, \dots, w_{n-1}$ , de forma que:

$$\begin{aligned} w_1 &= \frac{dy}{dx}, & w_2 &= \frac{dw_1}{dx} = \frac{d^2 y}{dx^2}, & w_3 &= \frac{dw_2}{dx} = \frac{d^3 y}{dx^3}, & \dots, \\ w_{n-1} &= \frac{dw_{n-2}}{dx} = \frac{d^{(n-1)} y}{dx^{(n-1)}} \end{aligned} \quad (8.159)$$

Com essas definições, a EDO de  $n$ -ésima ordem dada na Eq. (8.157) com as condições iniciais da Eq. (8.158) pode ser escrita como o seguinte sistema de  $n$  EDOs de primeira ordem, cada uma com sua própria condição inicial:

$$\frac{dy}{dx} = w_1 \quad \text{com} \quad y(a) = A_1 \quad (8.160)$$

$$\frac{dw_1}{dx} = w_2 \quad \text{com} \quad w_1(a) = A_2 \quad (8.161)$$

$$\frac{dw_2}{dx} = w_3 \quad \text{com} \quad w_2(a) = A_3 \quad (8.162)$$

...

$$\frac{dw_{n-2}}{dx} = w_{n-1} \quad \text{com} \quad w_{n-2}(a) = A_{n-1} \quad (8.163)$$

Por exemplo, o PVI de terceira ordem:

$$\frac{d^3y}{dx^3} = 2x - 3y + 4\frac{dy}{dx} + x\frac{d^2y}{dx^2} \quad (8.164)$$

com as condições iniciais:

$$y(0) = 3, \quad \left.\frac{dy}{dx}\right|_{x=0} = 2, \quad \left.\frac{d^2y}{dx^2}\right|_{x=0} = 7 \quad (8.165)$$

pode ser transformado nos seguinte sistema de três EDOs de primeira ordem:

$$\frac{dy}{dx} = w_1 \quad \text{com} \quad y(0) = 3 \quad (8.166)$$

$$\frac{dw_1}{dx} = w_2 \quad \text{com} \quad w_1(0) = 2 \quad (8.167)$$

$$\frac{dw_2}{dx} = 2x - 3y + 4w_1 + xw_2 \quad \text{com} \quad w_2(0) = 7 \quad (8.168)$$

### Sistemas de PVIs de ordem mais elevada

Qualquer sistema de EDOs de ordem mais elevada pode ser escrito como um sistema de EDOs de primeira ordem usando a abordagem descrita anteriormente. Por exemplo, considere um sistema de duas EDOs de segunda ordem:

$$\begin{aligned} \frac{d^2x}{dt^2} &= f(x, y, t, \frac{dx}{dt}, \frac{dy}{dt}) \\ \frac{d^2y}{dt^2} &= f(x, y, t, \frac{dx}{dt}, \frac{dy}{dt}) \end{aligned} \quad (8.169)$$

Definido  $u = dx/dt$  e  $w = dy/dt$ , o sistema da Eq. (8.169) pode ser rescrito como um sistema de quatro EDOs de primeira ordem:

$$\begin{aligned} \frac{dx}{dt} &= u & \frac{dy}{dt} &= w \\ \frac{du}{dt} &= f(x, y, t, u, w) & \frac{dw}{dt} &= f(x, y, t, u, w) \end{aligned} \quad (8.170)$$

Esse sistema pode ser resolvido com os métodos descritos na Seção 8.8, desde que condições iniciais sejam prescritas.

## 8.10 USO DE FUNÇÕES RESIDENTES DO MATLAB PARA RESOLVER PROBLEMAS DE VALOR INICIAL

O MATLAB dispõe de uma variedade de funções residentes que podem ser usadas para resolver desde uma única EDO a sistemas de EDOs de primeira ordem. Vários métodos numéricos diferentes estão disponíveis para resolver EDOs, e a maioria deles usa técnicas avançadas que otimizam o passo de integração automaticamente, minimizam o erro por passo, ou até mesmo adaptam o passo de integração ao comportamento local da solução. Esta seção introduz o uso dessas

funções residentes. A solução de uma única EDO de primeira ordem é tratada na Seção 8.10.1. A solução de sistemas de EDOs de primeira ordem com funções residentes do MATLAB é apresentada na Seção 8.10.2.

No restante desta seção,  $t$  e  $y$  são escolhidos como variáveis independente e dependente, respectivamente. Isso é feito de forma a manter-se uma consistência com as informações fornecidas no menu de ajuda do MATLAB, e também porque, em muitas aplicações, o tempo é a variável independente.

### 8.10.1 Solução de uma única EDO de primeira ordem usando o MATLAB

O MATLAB dispõe de várias funções residentes que podem ser usadas para resolver uma única EDO de primeira ordem. As diferenças entre as várias funções residentes estão nos métodos numéricos usados no processo de solução. O procedimento usado para resolver uma EDO de primeira ordem no MATLAB é resumido a seguir.

- ***Declaração do problema***

A EDO pode ser escrita na forma:

$$\frac{dy}{dt} = f(t, y) \quad \text{com} \quad y(t_1) = y_1$$

Três informações são necessárias para que uma EDO de primeira ordem seja resolvida: uma equação que forneça uma expressão para a derivada de  $y$  em relação a  $t$ , o intervalo da variável independente e o valor inicial de  $y$ . Um exemplo é:

$$\frac{dy}{dt} = \frac{t^3 - 2y}{t} \quad \text{para} \quad 1 \leq t \leq 3 \quad \text{com} \quad y = 4,2 \text{ em } t = 1 \quad (8.171)$$

- ***Função criada pelo usuário para calcular a função  $f(t, y)$***

A função  $f(t, y)$ , que fornece os valores de  $dy/dt$  para valores arbitrários de  $t$  e  $y$ , deve ser escrita pelo usuário em uma função à parte. Para a EDO dada na Eq. (8.171), a função é:

```
function dydt = EqDif (t,y)
dydt = (t^3 - 2*y)/t
```

Essa função é chamada de `EqDif` neste exemplo, mas em geral pode ter qualquer nome. Conforme mostrado mais adiante, esse nome é usado pela função residente do MATLAB responsável pela solução da EDO. O argumento de saída da função criada pelo usuário é o valor de  $dy/dt$ . Em sua forma mais simples, ela tem dois argumentos de entrada ( $t$  e  $y$ ). No entanto, ela pode ter outros argumentos de entrada além de  $t$  e  $y$ . Por exemplo, se  $f$  também for função da temperatura,  $T$  (uma constante), então a linha de definição da função também poderá incluir  $T$  como um argumento de entrada:

```
function dydt = EqDif (t, y, T)
```

Conforme explicado mais adiante, um valor numérico de  $T$  é transferido para a função quando ela é usada como argumento da função residente do MATLAB responsável pela solução da EDO.

- **Método de solução**

A Tabela 8-2 lista várias funções residentes disponíveis no MATLAB para resolver EDOs de primeira ordem. Uma breve descrição do método numérico usado em cada função residente foi incluída na tabela.

**Tabela 8-2** Funções residentes do MATLAB para resolver EDOs de primeira ordem

Nome da função	Descrição
ode45	Em problemas não-rígidos, é a melhor função a ser usada como primeira tentativa. Método de passo simples baseado em métodos de Runge-Kutta explícitos de quarta e quinta ordem.
ode23	Indicada para problemas não-rígidos. Método de passo simples baseado em métodos de Runge-Kutta explícitos de segunda e terceira ordem. Freqüentemente mais rápido, porém menos preciso que a função ode45.
ode113	Indicada para problemas não-rígidos. Método multipasso baseado nos métodos de Adams-Basforth-Moulton.
ode15s	Indicada para problemas rígidos. Método multipasso que usa um método de ordem variável. Baixa à média precisão.
ode23s	Indicada para problemas rígidos. Método de passo simples. Pode resolver alguns problemas que a função ode15s não é capaz de resolver. Baixa precisão.
ode23t	Indicada para problemas moderadamente rígidos. Baixa precisão.
ode23tb	Indicada para problemas rígidos. Usa um método de Runge-Kutta implícito. Freqüentemente mais eficiente que a função ode15s.

As várias funções têm parâmetros internos (parâmetros de controle) que especificam os detalhes da integração em cada método, tal como o passo de integração e o erro. Os valores padrão desses parâmetros são selecionados de forma a fazer com que as funções trabalhem bem na solução de problemas comuns. Conforme descrito mais adiante, o usuário pode mudar os valores dos parâmetros de controle se necessário.

- **Solução da EDO**

A forma do comando responsável pela solução da EDO é a mesma para todas as funções. O comando tem uma forma simples adequada à solução de EDOs simples e também uma forma mais avançada que inclui argumentos opcionais. A forma mais simples do comando é apresentada primeiro em um exemplo. O comando que usa argumentos opcionais é apresentado logo em seguida.

A forma simples da função residente do MATLAB utilizada na solução de EDOs de primeira ordem é:

$$[t, y] = \text{NomeDaFunção} ('EqDif', tInt, yIni)$$

onde:

NomeDaFunção	Nome da função ou do método numérico listado na Tabela 8-2 (por exemplo, ode45 ou ode23s).
EqDif	Nome da função criada pelo usuário que calcula $\frac{dy}{dt} = f(t, y)$ para valores fornecidos de $t$ e $y$ . EqDif pode ser digitada como <i>string</i> (isto é, 'EqDif'), ou com o uso de um símbolo especial (como @EqDif).
tInt	Vetor que especifica o domínio da variável independente. O vetor deve ter pelo menos dois elementos, mas pode ter mais. Se o vetor tiver dois elementos, eles devem ser [ta tb], que definem o domínio da variável independente. O vetor tInt pode, contudo, ter valores adicionais entre o primeiro e o último ponto. O número de elementos em tInt afeta a saída do comando. Ver [t, y] a seguir.
yIni	Valor inicial de $y$ (o valor de $y$ no primeiro ponto do domínio).
[t, y]	Saída, ou seja, a solução da EDO. t e y são vetores coluna. O primeiro ponto e o último correspondem ao primeiro e ao último ponto do domínio da solução. O espaçamento e o número de pontos dependem do vetor de entrada tInt. Se tInt tiver dois elementos (os pontos inicial e final), os vetores t e y conterão a solução em cada passo de integração calculado pela função. Se yIni tiver mais de dois elementos (pontos adicionais entre o primeiro e o último ponto), então os vetores t e y conterão a solução apenas nesses pontos. O número de pontos em tInt não afeta os passos de tempo utilizados pela função na solução da EDO.

Uma solução para a EDO já resolvida no Exemplo 8-6 é mostrada a seguir, no Exemplo 8-9, usando a função ode45 do MATLAB.

### Exemplo 8-9: Uso de uma função residente do MATLAB para resolver uma EDO de primeira ordem

Use a função `ode45` do MATLAB para resolver a EDO:

$$\frac{dy}{dx} = -1,2y + 7e^{-0,3x} \text{ de } x = 0 \text{ a } x = 2,5 \text{ com a condição inicial } y = 3 \text{ em } x = 0.$$

$$\text{Compare os resultados com a solução exata (analítica): } y = \frac{70}{9}e^{-0,3x} - \frac{43}{9}e^{-1,2x}.$$

#### SOLUÇÃO

O seguinte programa demonstra o uso da função `ode45`:

```
tInt = [0:0.5:2.5];
yIni = 3;
[x,y] = ode45('EqDifExp8',tInt,yIni)
yExato= 70/9*exp(-0.3*x) - 43/9*exp(-1.2*x)
erro = yExato - y
```

Domínio da variável independente.

Valor inicial da variável dependente.

Usa a função `ode45`.

Calcula a solução exata nos mesmos pontos da solução numérica.

A função criada pelo usuário `EqDifExp8` usada no argumento da função `ode45` calcula o valor de  $dy/dx$ :

```
function dydx = EqDifExp8(x,y)
dydx = -1.2*y + 7*exp(-0.3*x);
```

Como `tInt` é um vetor com seis elementos, a solução é mostrada no seis pontos especificados. Quando o programa é executado, os seguintes dados são exibidos na janela de comandos do MATLAB:

```
x =
    0
    0.5000
    1.0000
    1.5000
    2.0000
    2.5000

y =
    3.0000
    4.0723
    4.3229
    4.1696
    3.8351
    3.4361

erro =
    1.0e-005 *
        0
    -0.2400
    -0.0374
     0.0310
     0.0452
     0.0476
```

Os resultados (o vetor `erro`) mostram que a solução numérica apresenta um erro extremamente pequeno.

Conforme mencionado anteriormente, as funções residentes que o MATLAB possui para resolver EDOs de primeira ordem podem incluir argumentos de entrada opcionais que mudam os valores padrão dos parâmetros internos que controlam as propriedades de integração. Esses parâmetros podem ser usados para transferir valores de argumentos à função EqDif criada pelo usuário. Quando são utilizados argumentos opcionais, a forma do comando é:

```
[t,y]=NomeDaFunção('EqDif',tInt,yIni,options,arg1,arg2)
```

O argumento options pode ser usado para mudar os valores padrão dos parâmetros internos (propriedades) das funções que resolvem as EDOs. options é o nome de um arranjo estruturado contendo os novos valores das propriedades (o termo arranjo estruturado se refere, no MATLAB, a um arranjo que pode conter diferentes tipos de dados. Por exemplo, um campo pode conter uma variável *string*, e o campo seguinte pode conter um número). A estrutura options é criada por uma função do MATLAB denominada odeset, que tem a seguinte forma:

```
options = odeset('nome1',valor1,'nome2',valor2,...)
```

onde nome1, nome2,... são os nomes das propriedades, e valor1, valor2,... são os novos valores correspondentes às variáveis. Por exemplo, para ajustar o passo de integração para 0,0002 e o erro relativo para 0,001, o comando é escrito como:

```
options = odeset('InitialStep',0.0002,'RelTol',0.001)
```

Muitas outras propriedades podem ser encontradas na documentação de ajuda do MATLAB. No entanto, essa ferramenta deve ser utilizada por usuários mais experientes.

Os argumentos arg1, arg2,... podem ser usados, quando necessário, para transferir valores de argumentos à função EqDif, que define  $\frac{dy}{dt} = f$ . Nesse caso,  $f$  é função de  $y$ ,  $t$  e de parâmetros adicionais, e a linha de comando da função EqDif tem a forma (para o caso de dois parâmetros adicionais):

```
dydt = EqDif(t,y,arg1,arg2)
```

Quando a função que resolve a EDO é rodada, os valores dos argumentos adicionais são transferidos para a função EqDif.

### Nota importante

Na função que resolve a EDO, o argumento options deve estar presente se argumentos opcionais arg1, arg2,... forem usados para transferir valores à função EqDif. Se o usuário não desejar alterar quaisquer parâmetros internos, um vetor vazio, digitado como [], deve ser usado em options.

O uso de argumentos opcionais em funções do MATLAB que resolvem EDOs é ilustrado no Exemplo 8-10.

### Exemplo 8-10: Resfriamento de uma chapa aquecida (uso de uma função residente do MATLAB para resolver EDOs de primeira ordem)

Quando uma chapa metálica é repentinamente retirada de um forno e exposta ao ambiente, ela se resfria em função de perdas por convecção e radiação. A taxa na qual a temperatura da chapa  $T$  varia com o tempo é dada por:

$$\frac{dT}{dt} = -\frac{A_s}{\rho V C_v} [\sigma_{SB} \epsilon (T^4 - T_\infty^4) + h(T - T_\infty)] \quad (8.172)$$

onde  $A_s$  é a área superficial da chapa,  $\rho = 300 \text{ kg/m}^3$  é sua densidade de massa,  $V$  é seu volume,  $C_v = 900 \text{ J/kg/K}$  é seu calor específico na condição de volume constante e  $\epsilon = 0,8$  é sua emissividade radiativa. Além disso,  $\sigma_{SB} = 5,67 \times 10^{-8} \text{ W/m}^2/\text{K}^4$  é a constante de Stefan-Boltzmann,  $h = 30 \text{ W/m}^2/\text{K}$  é o coeficiente de transferência de calor e  $T_\infty$  é a temperatura ambiente.

Escreva uma função no MATLAB que calcule a temperatura da chapa em função do tempo nos primeiros 180 s após a sua retirada do forno e que mostre os resultados em uma figura. Chame a função de `TempChapa(T1, Vol, Area, Tamb)`, onde  $T1$  é a temperatura inicial da chapa,  $Vol$  e  $Area$  são o volume e a área da chapa, respectivamente, e  $Tamb$  é a temperatura ambiente.

Use a função para traçar um gráfico que mostre a variação da temperatura com o tempo para uma chapa com  $V = 0,003 \text{ m}^3$  e  $A_s = 0,25 \text{ m}^2$ , assumindo uma temperatura inicial de 673 K e uma temperatura ambiente é de 298 K.

### SOLUÇÃO

Para resolver o problema, duas funções são escritas. Uma é a função `TempChapa(T1, Vol, Area, Tamb)`, e a outra é uma função denominada `TaxaTemp(t, T, Vol, As, Tamb)`, que calcula o valor de  $\frac{dy}{dt}$  na

Eq. (8.172). A EDO é resolvida em `TempChapa` usando a função `ode45` do MATLAB (`ode45` usa `TaxaTemp` para resolver a equação). As constantes  $\rho$ ,  $C_v$ ,  $\epsilon$  e  $\sigma_{SB}$  são definidas no interior da `TaxaTemp`. Os argumentos de entrada de `TaxaTemp` são  $t$ ,  $T$ ,  $Vol$ ,  $As$  e  $Tamb$ . Os dois primeiros argumentos,  $t$  e  $T$ , são as variáveis independente e dependente normalmente utilizadas. Os três últimos argumentos,  $Vol$ ,  $As$  e  $Tamb$ , são argumentos adicionais que fornecem os valores do volume e da área superficial da chapa e também da temperatura ambiente. O valor desses argumentos é fornecido na função `TempChapa` e transferido para `TaxaTemp` por meio dos argumentos opcionais presentes na função `ode45`.

A listagem da função `TempChapa` é:

```
function TempChapa(T1,Vol,Area,Tamb)
% A função TempChapa calcula a temperatura de uma placa.
% Variáveis de entrada:
% T1 Temperatura inicial em graus Kelvin.
% Vol Volume da chapa em m cúbicos.
% Area Área da chapa em m quadrados.
% Tamb Temperatura ambiente em graus Kelvin.
```

```
% Saída:  
% Gráfico da temperatura versus tempo.  
  
tInt = [0 180];  
[Time Temp] = ode45(@TaxaTemp,tInt,T1,[ ],Vol,Area,Tamb);  
plot(Tempo,Temp)  
 xlabel('Tempo (s)')  
 ylabel('Temperatura (K)')
```

Argumentos opcionais de entrada.  
Domínio da variável independente.  
Usa a função `ode45`.  
Vetor vazio para o argumentos options.

A função `TaxaTemp`, que é usada como argumento da função `ode45` para calcular o valor de  $dT/dt$ , é:

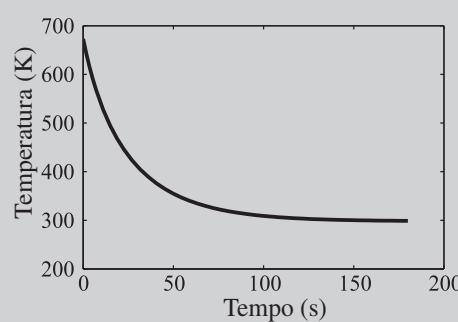
```
function dTdt = TaxaTemp(t, T, Vol, As, Tamb)  
Rho = 300; Cv = 900; h = 30; Epsi = 0.8; Sigma = 5.67e-8;  
ARVC = As/(Rho*Vol*Cv);  
SigEps = Epsi*Sigma;  
dTdt = -ARVC*(SigEps*(T^4 - Tamb^4) + h*(T - Tamb));
```

Eq. (8.172).

O problema é resolvido com o uso do seguinte comando:

```
>>TempChapa(673,0.001,0.25,298)
```

Quando se executa a função `TempChapa`, a seguinte figura é gerada no MATLAB:



### 8.10.2 Solução de sistemas de EDOs de primeira ordem usando o MATLAB

Sistemas de EDOs de primeira ordem podem ser resolvidos com as funções residentes listadas na Tabela 8-2. Os procedimentos e comandos usados na solução de sistemas são os mesmos usados no caso de uma única equação. Quando se resolve um sistema, no entanto, as equações e as condições iniciais são fornecidas como vetores, e a solução é dada pelo MATLAB na forma de uma matriz onde cada coluna contém a solução para uma variável dependente.

- **Função criada pelo usuário para calcular**  $\frac{dy_i}{dt} = f_i(t, y_1, y_2, \dots)$

Quando se resolve uma única EDO, a função `EqDif` criada pelo usuário calcula o lado direito de uma equação diferencial,  $\frac{dy}{dt} = f(t, y)$ . Sua linha de definição tem a forma `dydt=EqDif(t,y)`, onde os argumentos de entrada `t` e `y` são escalares usados para calcular o valor de  $f(t, y)$ , que é atribuído a `dydt`.

Quando se resolve um sistema de EDOs (ver Eq. (8.107)), há diversas variáveis dependentes envolvidas,  $y_1, y_2, \dots$ . Com isso, a função `EqDif` precisa calcular o lado direito de várias equações diferenciais em um instante  $t$  especificado:

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2, \dots), \quad \frac{dy_2}{dt} = f_2(t, y_1, y_2, \dots), \dots$$

A linha de definição da função tem a mesma forma daquela usada para resolver uma única EDO:  $dydt = \text{EqDif}(t, y)$ . No entanto, quando utilizada para resolver sistemas de EDOs,  $t$  é um escalar e  $y$  é um vetor contendo as variáveis dependentes  $[y_1, y_2, \dots]$ . Os componentes de  $y$  e o valor de  $t$  são usados no cálculo dos valores de  $f_1(t, y_1, y_2, \dots), f_2(t, y_1, y_2, \dots), \dots$ , que são atribuídos como elementos de um *vetor coluna* ao argumento de saída  $dydt$ .

- **Resolvendo um Sistema de EDOs**

O comando utilizado é  $[t, y] = \text{NomeDaFunção}('EqDif', tInt, yIni)$ . Quando esse comando é aplicado na solução de sistemas de EDOs, o argumento de entrada  $yIni$  é um vetor cujos componentes são os valores iniciais das variáveis dependentes,  $[Y_1, Y_2, \dots]$  (ver Eq. (8.107)). Os argumentos de saída são  $[t, y]$ . Quanto se resolve uma EDO,  $t$  e  $y$  são vetores coluna. Quando se resolve um sistema de EDOs,  $t$  é um vetor coluna e  $y$  é uma matriz, onde cada coluna de  $y$  é a solução referente a uma variável dependente. A primeira coluna de  $y$  é a solução de  $y_1$ , a segunda coluna é a solução de  $y_2$ , e assim por diante.

O Exemplo 8-11 mostra como funções residentes do MATLAB são usadas na solução de um sistema de EDOs.

### Exemplo 8-11: Problema predador-presa (uso de uma função residente do MATLAB para resolver um sistema de duas EDOs de primeira ordem)

A relação entre a população de leões (predadores),  $N_L$ , e a população de gazelas (presas),  $N_G$ , que residem em uma mesma área pode ser modelada por um sistema de duas EDOs. Suponha que uma comunidade seja formada por  $N_L$  leões (predadores) e  $N_G$  gazelas (presas), com  $b$  e  $d$  representando as taxas de natalidade e mortalidade das respectivas espécies. A taxa de variação (crescimento ou decrescimento) das populações de leões e de gazelas pode ser modelada pelas equações:

$$\begin{aligned}\frac{dN_L}{dt} &= b_L N_L N_G - d_L N_L \\ \frac{dN_G}{dt} &= b_G N_G - d_G N_G N_L\end{aligned}\tag{8.173}$$

Determine a população de leões e de gazelas em função do tempo, de  $t = 0$  a  $t = 25$  anos, se, em  $t = 0$ ,  $N_G = 3000$  e  $N_L = 500$ . Os coeficientes do modelo são:  $b_G = 1,1 \text{ ano}^{-1}$ ,  $b_L = 0,00025 \text{ ano}^{-1}$ ,  $d_G = 0,0005 \text{ ano}^{-1}$  e  $d_L = 0,7 \text{ ano}^{-1}$ .

### SOLUÇÃO

Neste problema,  $N_L$  e  $N_G$  são as variáveis dependentes, e  $t$  é a variável independente. Para resolvê-lo, escreve-se uma função chamada `TaxaPop(t, N)`. Essa função, listada a seguir, calcula os valores de  $\frac{dN_L}{dt}$  e  $\frac{dN_G}{dt}$  na Eq. (8.173). O argumento de entrada  $N$  é um vetor contendo as variáveis dependentes,

onde  $N(1) = N_L$  e  $N(2) = N_G$ . Note que esses componentes são usados no cálculo do primeiro valor das equações diferenciais. O argumento de saída  $dNdt$  é um vetor coluna cujo primeiro elemento é o valor da derivada  $\frac{dN_L}{dt}$ , e cujo o segundo elemento é o valor da derivada  $\frac{dN_G}{dt}$ .

```
function dNdt = TaxaPop(t, N)
bG = 1.1; bL = 0.00025; dG = 0.0005; dL = 0.7;
dNdt(1,1) = bL*N(1)*N(2)-dL*N(1);
dNdt(2,1) = bG*N(2)-dG*N(2)*N(1);
```

Calcula a Eq. (8.173).

O sistema de EDOs na Eq. (8.173) é resolvido com a função `ode45` do MATLAB, que usa `TaxaPop` para calcular os lados direitos das equações diferenciais. O programa a seguir mostra os detalhes. Note que `Nini` é um vetor cujo primeiro elemento é o valor inicial de  $N_L$ , e cujo segundo elemento é o valor inicial de  $N_G$ .

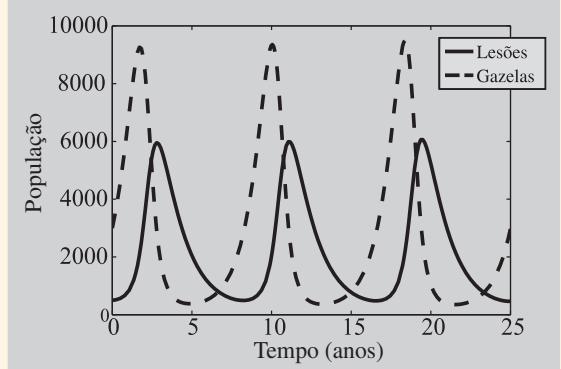
```
tInt= [0 25]; Domínio da variável independente.
Nini = [500 3000]; Valores iniciais da variável dependente (note que Nini é um vetor).
[Tempo Pop] = ode45(@TaxaPop,tInt,Nini); Resolve o sistema de EDOs.
plot(Tempo,Pop(:,1),'-',Tempo,Pop(:,2),--)
xlabel('Tempo (s)')
ylabel('População')
legend('Leões','Gazelas')
```

Nos argumentos de saída da função `ode45`, a variável `Tempo` é um vetor coluna, e a variável `Pop` é um arranjo de duas colunas contendo a solução para  $N_L$  e  $N_G$  na primeira e na segunda colunas, respectivamente. Os primeiros elementos de `[Tempo Pop]`, que são mostrados na janela de comandos do MATLAB se o ponto e vírgula for removido do final do comando, são ilustrados abaixo. O programa gera um gráfico que mostra a população de leões e de gazelas em função do tempo.

```
Tempo =
0
0.0591
0.1182
0.1773
...
Pop =
1.0e+003 *
0.5000 3.0000
0.5020 3.1545
0.5053 3.3166
...
```

Coluna de  $N_L$

Coluna de  $N_G$



## 8.11 ERRO DE TRUNCAMENTO LOCAL NO MÉTODO DE RUNGE-KUTTA DE SEGUNDA ORDEM

O método de Runge-Kutta de segunda ordem foi apresentado na Seção 8.5.1. Conforme mencionado, o erro de truncamento associado a esse método é de

$O(h^3)$ , e o erro de truncamento global é de  $O(h^2)$ . Esta seção apresenta a dedução de uma estimativa para o erro de truncamento local associado ao método de Runge-Kutta de segunda ordem na versão Euler modificado. Em geral, o erro de truncamento local em cada passo da solução é causado pelo caráter aproximado da fórmula utilizada.

### Uma nota a respeito da notação usada nesta seção

Nesta seção, são calculados diferentes valores para a variável dependente  $y$ , e esses valores são comparados em um dado valor da variável independente  $x$ . Para tornar a apresentação mais clara, identifica-se cada grandeza com um sobrescrito. Uma variável dependente calculada com o emprego de algum método numérico é escrita como  $y_i^{NS}$  (no restante do capítulo, ela é representada simplesmente pela letra  $y$ ). A solução verdadeira da EDO é escrita como  $y_i^{TS}$ . Finalmente, a notação  $y_i^{Taylor}$  é usada para o valor calculado com uma expansão em série de Taylor.

O erro de truncamento local é o erro inerente à fórmula usada na obtenção da solução numérica em um único passo (subintervalo). Ele é a diferença entre a solução numérica e a solução exata naquele subintervalo. Em um passo genérico  $i$ , a solução numérica  $y_i^{NS}$  em  $x = x_i$  é conhecida (calculada previamente), e a solução numérica  $y_{i+1}^{NS}$  em  $x = x_{i+1}$  é calculada com uma fórmula aproximada. O valor da solução exata para esse passo pode ser expresso pela expansão em série de Taylor com três termos e um resíduo (ver Capítulo 2):

$$y_{i+1}^{Taylor} = y_i^{NS} + \frac{dy}{dx} \Big|_{x_i, y_i^{NS}} h + \frac{d^2y}{dx^2} \Big|_{x_i, y_i^{NS}} \frac{h^2}{2} + \alpha_i h^3 \quad (8.174)$$

onde o quarto termo à direita corresponde ao resíduo. Como  $\frac{dy}{dx} = f(x, y)$ , a derivada segunda  $\frac{d^2y}{dx^2} \Big|_{x_i, y_i^{NS}}$  pode ser expressa em termos de  $f(x, y)$  e de suas derivadas parciais:

$$\frac{d^2y}{dx^2} = \frac{d}{dx} f(x, y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} \quad (8.175)$$

Substituindo a Eq. (8.175) na Eq. (8.174), obtém-se:

$$y_{i+1}^{Taylor} = y_i^{NS} + f(x_i, y_i^{NS}) h + \left( \frac{\partial f}{\partial x} \Big|_{x_i, y_i^{NS}} + \frac{\partial f}{\partial y} \Big|_{x_i, y_i^{NS}} f(x_i, y_i^{NS}) \right) \frac{h^2}{2} + \alpha_i h^3 \quad (8.176)$$

A solução numérica obtida com o método de Runge-Kutta de segunda ordem na versão de Euler modificado é dada pelas Eqs. (8.63)-(8.64), que são repetidas aqui:

$$y_{i+1}^{NS} = y_i^{NS} + \frac{1}{2}(K_1 + K_2)h \quad (8.177)$$

$$\begin{aligned} K_1 &= f(x_i, y_i^{NS}) \\ K_2 &= f(x_i + h, y_i^{NS} + K_1 h) \end{aligned} \quad (8.178)$$

Na equação usada no cálculo de  $K_2$ , a função  $f(x_i + h, y_i^{NS} + K_1 h)$  pode ser expandida em uma série de Taylor em torno de  $x = x_i$  e  $y = y_i^{NS}$  (para isso, utiliza-se a expansão em série de Taylor para funções de duas variáveis, já que  $f$  é função de  $x$  e  $y$  (ver Seção 2.7.2)):

$$K_2 = f(x_i + h, y_i^{NS} + K_1 h) = f(x_i, y_i^{NS}) + h \frac{\partial f}{\partial x} \Big|_{x_i, y_i^{NS}} + K_1 h \frac{\partial f}{\partial y} \Big|_{x_i, y_i^{NS}} + \beta_i h^2 \quad (8.179)$$

onde o quarto termo à direita é o resíduo. Substituindo  $K_1 = f(x_i, y_i^{NS})$  na Eq. (8.179) e então substituindo  $K_1 = f(x_i, y_i^{NS})$  e a Eq. (8.179) na Eq. (8.177), obtém-se:

$$y_{i+1}^{NS} = y_i^{NS} + f(x_i, y_i^{NS})h + \frac{1}{2}h^2 \left( \frac{\partial f}{\partial x} \Big|_{x_i, y_i^{NS}} + f(x_i, y_i^{NS}) \frac{\partial f}{\partial y} \Big|_{x_i, y_i^{NS}} \right) + \frac{1}{2}\beta_i h^3 \quad (8.180)$$

O erro de truncamento local,  $e_i^{TR}$ , é dado pela diferença entre as Eqs. (8.180) e (8.176), que é da ordem de  $h^3$ :

$$e_{i+1}^{TR} = y_{i+1}^{NS} - y_{i+1}^{Taylor} = \frac{1}{2}\beta_i h^3 - \alpha_i h^3 = O(h^3) \quad (8.181)$$

A Eq. (8.181) mostra que o erro de truncamento local é de  $O(h^3)$ . Quando o intervalo da solução  $[a, b]$  é dividido em  $N$  subintervalos de largura  $h$ , os erros de truncamento local se acumulam e o erro de truncamento global é de  $O(h^2)$ .

## 8.12 LARGURA DO PASSO DE INTEGRAÇÃO PARA SE OBTER A PRECISÃO DESEJADA

A ordem de grandeza dos erros de truncamento associados aos vários métodos numéricos empregados na solução de EDOs pode ser usada para determinar-se a largura do passo de integração  $h$  necessária para a solução de uma EDO com a precisão desejada. Lembre, do Exemplo 8-1, que, em geral, à medida que se reduz o passo de integração, mais precisa se torna a solução. Se for desejado que a solução tenha uma precisão, por exemplo, de cinco casas decimais, a informação que se tem a respeito do erro de truncamento associado a um determinado método numérico pode ajudar na determinação de um passo de integração que leve à precisão desejada. Esta seção descreve um método que pode ser aplicado na determinação desse passo de integração.

O método usado para determinar a largura do passo de integração necessário para se ter a precisão desejada na solução é ilustrado aqui para o método de Runge-Kutta de terceira ordem. Conforme mencionado na Seção 8.5.2, o método de Runge-Kutta de terceira ordem tem um erro de truncamento local da ordem de  $h^4$ . Portanto, o erro de truncamento local para um determinado valor de  $h$  pode ser escrito como:

$$e_h^{TR} = Ah^4 \quad (8.182)$$

onde  $A$  é uma constante que depende do problema analisado. Note que, se  $h$  for dividido por dois, então o erro local se torna  $e_{h/2}^{TR} = \frac{Ah^4}{16}$ . Isso significa que, ignorando-se o erro de truncamento acumulado, o erro de truncamento local decorrente do avanço da solução numérica ao longo de  $h$  para um passo de integração de  $h/2$  é aproximadamente igual a  $2e_{h/2}^{TR}$ , ou:

$$2e_{h/2}^{TR} = \frac{Ah^4}{8} \quad (8.183)$$

Subtraindo a Eq. (8.183) da Eq. (8.182), obtém-se:

$$e_h^{TR} - 2e_{h/2}^{TR} = Ah^4 - \frac{Ah^4}{8} = \frac{7}{8}Ah^4 \quad (8.184)$$

Agora, suponha que uma EDO seja resolvida duas vezes com o método de Runge-Kutta de terceira ordem: na primeira vez, a solução é calculada em um único passo de integração  $h$ , o que leva à solução numérica  $y_h$ ; na segunda vez, uma outra solução é calculada em dois passos de integração  $h/2$ , o que leva à solução numérica  $y_{h/2}$ . Usando a Eq. (8.181), as soluções numéricas podem ser expressas em termos da expansão em série de Taylor e do erro de truncamento local:

$$y_h^{NS} = y_h^{Taylor}(x_i + h) + e_h^{TR} \quad (8.185)$$

e

$$y_{h/2}^{NS} = y_h^{Taylor}(x_i + h) + 2e_{h/2}^{TR} \quad (8.186)$$

O fator 2 no lado direito da Eq. (8.186) é necessário porque, na solução que considera um passo de integração de  $h/2$ , dois passos são necessários para se obter a solução em  $x_i + h$ . Subtraindo a Eq. (8.185) da Eq. (8.186) e usando a Eq. (8.184), obtém-se:

$$y_h^{NS} - y_{h/2}^{NS} = \frac{7}{8}Ah^4 \quad (8.187)$$

A Eq. (8.187) diz que, se um código já escrito for rodado duas vezes para resolver uma EDO usando o método de Runge-Kutta de terceira ordem (considerando passos de integração  $h$  e  $h/2$ ), pode-se determinar a constante  $A$  por meio de um experimento numérico. Em outras palavras, se o código for rodado uma vez (em apenas um passo) considerando um passo de integração  $h$  arbitrariamente escolhido (embora razoável), pode-se determinar  $y_h^{NS}$ . Em seguida, se o mesmo código for rodado novamente (em dois passos) considerando um passo de integração  $h/2$ , determina-se  $y_{h/2}^{NS}$ . Como  $h$  foi escolhido, ele é uma grandeza conhecida e a Eq. (8.187) pode ser usada na determinação do parâmetro  $A$ . Uma vez determinado  $A$ , a Eq. (8.182) pode ser usada para calcular o valor de  $h$  que leva ao nível de precisão desejado. Por exemplo, se o erro desejado for  $\xi$ , então o valor de  $h$  que leva a  $e_h^{TR} \leq \xi$  é dado por:

$$h = \left( \frac{\xi}{A} \right)^{1/4} \quad (8.188)$$

Expressões similares podem ser deduzidas para métodos de Runge-Kutta de diferentes ordens, ou mesmo para outros métodos.

Critérios como a Eq. (8.188) devem ser usados apenas como uma referência, porque são inexatos. A razão para isso vem do fato de apenas o erro de truncamento (que depende do método utilizado) estar sendo considerado nesta discussão. O erro total é a soma do erro de truncamento global e do erro de arredondamento, e a Eq. (8.186) não contém nenhuma informação a respeito do erro de arredondamento, que depende da máquina utilizada (ver Capítulo 1). Conseqüentemente, o passo de integração ótimo dado pela Eq. (8.186) é uma aproximação e deve ser

tratado como tal. O Exemplo 8-12 mostra a obtenção do passo de integração ótimo para o uso do método de Runge-Kutta de quarta ordem.

### Exemplo 8-12: Determinação do passo de integração que forneça a precisão desejada

- Desenvolva uma fórmula que possa ser usada para calcular o passo de integração necessário para se obter a precisão desejada com a utilização do método de Runge-Kutta de quarta ordem.
- Considerando a EDO a seguir, aplique a fórmula obtida na letra (a) para determinar a ordem de grandeza do passo de integração necessário para que o erro seja menor que  $10^{-6}$ .

$$\frac{dy}{dx} = -\frac{y}{1+x^2}, \quad \text{com} \quad y(0) = 1, \quad \text{para} \quad 0 < x < 1 \quad (8.189)$$

### SOLUÇÃO

- Como o erro de truncamento local associado ao método de Runge-Kutta de quarta ordem é de  $O(h^5)$ , o erro de truncamento em uma solução com passo de integração  $h$  pode ser expresso por (de forma análoga à Eq. (8.182)):

$$e_h^{TR} = Ah^5 \quad (8.190)$$

Quando se divide o passo de integração por dois, o erro de truncamento local quando a variável independente avança dois passos é:

$$2e_{h/2}^{TR} = \frac{Ah^5}{16} \quad (8.191)$$

A subtração das duas últimas equações fornece uma relação análoga à Eq. (8.184):

$$e_h^{TR} - 2e_{h/2}^{TR} = Ah^5 - \frac{Ah^5}{16} = \frac{15}{16}Ah^5 \quad (8.192)$$

Usando as Eqs. (8.185) e (8.186), a diferença entre os erros de truncamento na Eq. (8.192) pode ser escrita em termos da diferença entre as soluções numéricas obtidas com passos de integração  $h$  e  $h/2$ :

$$y_h^{NS} - y_{h/2}^{NS} = \frac{15}{16}Ah^5, \quad \text{ou} \quad A = \frac{16}{15} \frac{(y_h^{NS} - y_{h/2}^{NS})}{h^5} \quad (8.193)$$

Uma vez obtidas as duas soluções numéricas, pode-se determinar a constante  $A$  na Eq. (8.193). De posse de  $A$ , a largura do passo de integração necessária para a precisão desejada é calculada usando a Eq. (8.190):

$$h = \left( \frac{\xi}{A} \right)^{1/5} \quad (8.194)$$

onde  $\xi$  é o erro desejado.

- Para determinar-se a ordem de grandeza do passo de integração de forma que o erro de truncamento local seja menor que  $10^{-6}$ , a Eq. (8.189) é resolvida duas vezes com o método de Runge-Kutta de quarta ordem: uma vez com  $h = 1$  e outra vez com  $h = 1/2$ . A solução da Eq. (8.189) é obtida usando a função edoRK4 (desenvolvida no Exemplo 8-6) na janela de comandos do MATLAB:

```
>> [x, y] = edoRK4('Cap8Exmp12EDO',0,1,1,1)
x =
0   1
y =
1.000000000000000  0.4566666666667
```

Resolve a EDO usando  $h = 1$ .

Nessa solução,  $y_h^{NS} = 0,45666666666667$ .

```
>> [x, y] = edoRK4('Cap8Exmp12EDO', 0, 1, 0.5, 1)
```

x =

0 0.500000000000000 1.000000000000000

y =

1.000000000000000 0.62900807381776 0.45599730642061

Resolve a EDO usando  $h = 1/2$ .

Nessa solução,  $y_{h/2}^{NS} = 0,45599730642061$ .

O argumento Cap8Exmp12EDO que aparece na função edoRK4 é o nome da seguinte função em arquivo que calcula o valor de  $dy/dx$ :

```
function dydx = Cap8Exmp12EDO(x,y)
dydx = -y/(1 + x^2);
```

Os resultados obtidos nas soluções numéricas são usados na Eq. (8.193) para determinar o valor da constante A:

$$A = \frac{16(0,45666667 - 0,45599730)}{15 \cdot 1^5} = 7,14 \times 10^{-4}$$

Agora que A é conhecido, a ordem de grandeza do passo de integração que leva a um erro de truncamento menor que  $10^{-6}$  pode ser calculada com a Eq. (8.194):

$$h = \left( \frac{10^{-6}}{7,14 \times 10^{-4}} \right)^{1/5} = 0,2687$$

Portanto, o passo de integração deve ser menor que  $h = 0,2687$  para que a solução da Eq. (8.189) com o método de Runge-Kutta de quarta ordem apresente um erro de truncamento menor que  $10^{-6}$ .

Para verificar esse resultado, a Eq. (8.189) é resolvida novamente usando  $h = 0,2$ , que é menor que o valor obtido acima.

```
>> [x,y]=edoRK4('Cap8Exmp12EDO',0,1,0.2,1);
>> y(6)
ans =
```

0.45594036941322

Resolve a EDO usando  $h = 0,2$ .

Mostra o último valor de y.

A solução numérica é  $y = 0,45594036941322$ .

A solução exata da Eq. (8.189) é  $y = 0,45593812776600$ .

A diferença entre as soluções é igual a  $2,24 \times 10^{-6}$ , que é maior que  $10^{-6}$  mas possui a mesma ordem de grandeza do erro desejado. Entretanto, o passo de integração  $h$  foi calculado para se obter um determinado erro de truncamento local, e a última solução também inclui um erro de truncamento global (erros de arredondamento são tipicamente desprezíveis neste caso, já que o MATLAB utiliza precisão dupla).

## 8.13 ESTABILIDADE

O erro numérico total consiste no erro de truncamento, que depende do método numérico utilizado, e do erro de arredondamento, que depende da máquina utilizada para fazer os cálculos. Quando uma EDO é resolvida numericamente, espera-se idealmente que o erro introduzido em cada passo não cresça à medida que a solução avance. Também se espera geralmente que o erro total decresça (ou pelo menos não aumente) com a redução do passo de integração  $h$ . Em outras

palavras, espera-se que a solução numérica se torne mais precisa com a redução do passo de integração. Entretanto, em algumas situações, o erro numérico cresce sem limites à medida que a solução avança em um problema de valor inicial, ou mesmo quando o passo de integração é reduzido. Essas situações são sintomas de **instabilidade**, e diz-se nesse caso que a solução se torna **instável**. A estabilidade de uma solução geralmente depende de três fatores:

- 1.** Do método de solução utilizado.
- 2.** Do passo de integração  $h$  usado na solução numérica.
- 3.** Da equação diferencial específica sendo resolvida.

Há várias maneiras de se analisar a estabilidade. Uma delas é realizar os cálculos inicialmente utilizando precisão simples e então utilizando precisão dupla (o Capítulo 1 apresenta uma discussão sobre ambas) e comparar as respostas (o MATLAB usa a precisão dupla, por *default*). Isso pode ajudar a examinar a parcela do erro total referente aos erros de arredondamento. A segunda maneira é variar o passo de integração  $h$  e observar o comportamento da solução à medida que esse parâmetro é reduzido. Isso permite o controle sobre a parcela do erro total correspondente aos erros de truncamento. Passos de integração muito pequenos também podem resultar em erros de arredondamento maiores. A terceira maneira é usar um método numérico de ordem mais elevada (com um menor erro de truncamento, como, por exemplo, o método de Runge-Kutta de quarta ordem) e comparar os resultados com as respostas obtidas com um método de ordem inferior (como, por exemplo, os métodos de Runge-Kutta de segunda ou terceira ordem). Uma outra maneira de verificar a estabilidade é testar o método utilizado em uma equação diferencial que possua solução analítica. Se o método for instável quando aplicado na solução de uma equação com solução conhecida, pode-se esperar que ele seja instável ao resolver outras equações.

O método explícito de Euler e os métodos de Runge-Kutta de passo simples e ordem mais elevada são numericamente estáveis se o passo de integração  $h$  for suficientemente pequeno. Isso é ilustrado com uma EDO de forma simples:

$$\frac{dy}{dx} = -\alpha y, \quad \text{com} \quad y(0) = 1 \quad (8.195)$$

onde  $\alpha > 0$  é uma constante. A solução exata dessa equação é  $y(x) = e^{-\alpha x}$ . Para um intervalo  $[x_i, x_{i+1}]$  (onde  $h = x_{i+1} - x_i$ ), a solução exata no ponto  $x_{i+1}$  é:

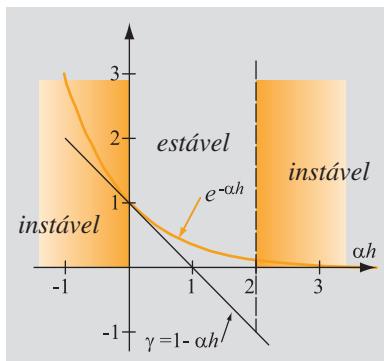
$$y^{TS}(x_{i+1}) = e^{-\alpha x_{i+1}} = e^{-\alpha(x_i + h)} = e^{-\alpha x_i} e^{-\alpha h} = y^{TS}(x_i) e^{-\alpha h} \quad (8.196)$$

A solução numérica obtida com o método explícito de Euler fornece:

$$y_{i+1} = y_i + (-\alpha y_i)h = (1 - \alpha h)y_i = \gamma y_i \quad (8.197)$$

onde  $\gamma = (1 - \alpha h)$ .

A comparação das Eqs. (8.196) e (8.197) mostra que o fator  $\gamma$  na solução numérica é apenas uma aproximação para o fator  $e^{-\alpha h}$  presente na solução exata. De fato,  $\gamma$  corresponde aos dois primeiros termos da expansão em série de Taylor de  $e^{-\alpha h}$  para  $\alpha h$  pequeno. Os dois fatores são traçados na Fig. 8-13. Esse fator  $\gamma$  é a fonte de erro e instabilidade. Pode-se ver na Eq. (8.197) que, quando  $|\gamma| < 1$ , o erro



**Figura 8-13** Critério de estabilidade.

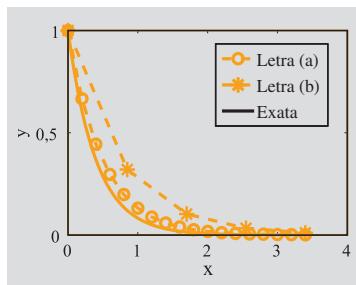
não é amplificado. Logo, para que a solução seja estável,  $|\gamma| < 1 \Rightarrow -1 < 1 - \alpha h < 1$ , o que pode ser escrito como:

$$0 < \alpha h < 2 \quad (8.198)$$

A Eq. (8.198) define um **critério de estabilidade**. Ela diz que o erro associado à solução de uma EDO na forma  $\frac{dy}{dx} = -\alpha y$  não crescerá sem limites desde que  $\alpha h < 2$ . Como  $\alpha$  depende do problema analisado, esse critério de estabilidade determina um limite superior para a largura do passo de integração  $h$  a ser utilizado em cada problema.

O Exemplo 8-13 ilustra o efeito do passo de integração na estabilidade da solução de uma EDO na forma  $\frac{dy}{dx} = -\alpha y$  quando se utiliza o método explícito de Euler.

Em contraste com o método explícito de Euler, o método implícito de Euler é incondicionalmente estável. Isso significa que os erros de truncamento não são amplificados à medida que a solução numérica avança, independentemente da largura do passo de integração. A solução do Exemplo 8-13 obtida com o método implícito de Euler (ver Problema 8.17) é mostrada na Fig. 8-14. Note que, embora a solução numérica se torne menos precisa à medida que  $h$  aumenta, ela não se torna instável. Em geral, critérios de estabilidade como a Eq. (8.198) não podem ser deduzidos para EDOs de complexidade arbitrária. Contudo, problemas de estabilidade como aqueles mostrados na figura do Exemplo 8-13 podem ser observados com a variação da largura do passo de integração.



**Figura 8-14** Solução do Exemplo 8-13 com o método implícito de Euler.

### Exemplo 8-13: Estabilidade do método explícito de Euler

Considere a solução da EDO:

$$\frac{dy}{dx} = -2,5y, \quad \text{com} \quad y(0) = 1, \quad \text{para} \quad 0 < x < 3,4 \quad (8.199)$$

- (a) Resolva a equação usando o método explícito de Euler com  $h = 0,2$ .
  - (b) Resolva a equação usando o método explícito de Euler com  $h = 0,85$ .
- A solução exata (analítica) é  $y = e^{-2,5x}$ . Mostre em um gráfico os resultados obtidos nas letras (a) e (b) juntamente com a solução exata.

### SOLUÇÃO

A Eq. (8.199) tem a forma da Eq. (8.195), com  $\alpha = 2,5$ . De acordo com a Eq. (8.198), obtém-se uma solução estável se  $h < \frac{2}{2,5} = 0,8$ . Conseqüentemente, pode-se esperar que uma solução estável seja obtida na letra (a), enquanto a solução da letra (b) tende a ser instável. As soluções são obtidas com o emprego do seguinte programa escrito no MATLAB.

```

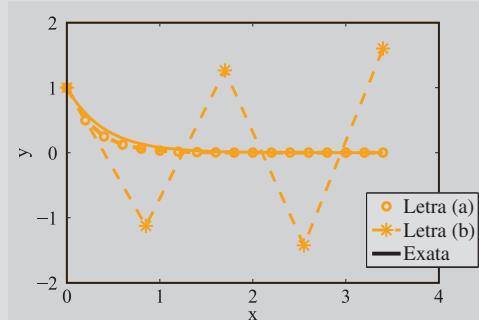
a = 0; b = 3.4; alfa = 2.5;
ha = 0.2; hb = 0.85;
xa(1) = a; xb(1) = a;
ya(1) = 1; yb(1) = 1;
Na = (b - a)/ha; Nb = (b - a)/hb;
DY = inline(' -2.5*y', 'y');
for i = 1:Na
    xa(i + 1) = xa(i) + ha;
    ya(i + 1) = ya(i) + DY(ya(i))*ha;
end
for i = 1:Nb
    xb(i + 1) = xb(i) + hb;
    yb(i + 1) = yb(i) + DY(yb(i))*hb;
end
xTrue = a:0.05:b;
yTrue = exp(-alfa*xTrue);
plot(xa,ya,'o--',xb,yb,'*r--',xTrue,yTrue,'k')

```

Solução da letra (a).

Solução da letra (b).

Solução exata (analítica).



## 8.14 EQUAÇÕES DIFERENCIAIS ORDINÁRIAS RÍGIDAS

Certas aplicações na ciência e na engenharia envolvem fenômenos físicos que variam em escalas espaciais ou temporais completamente diferentes. Exemplos de escalas temporais que apresentam variação ampla são encontrados em problemas de combustão, reações químicas, redes eletrônicas e controle. Essas aplicações freqüentemente levam a sistemas de EDOs cujas soluções incluem vários termos com amplitudes que variam com o tempo em taxas completamente diferentes. Tais sistemas de EDOs são chamados de rígidos e têm solução difícil. Por exemplo, a solução de uma EDO rígida pode apresentar somas ou diferenças de termos exponenciais como  $e^{-at}$  ou  $e^{-bt}$  ( $t$  é a variável independente), onde  $a$  e  $b$  podem ter ordens de grandeza completamente diferentes.

Como um exemplo ilustrativo de equações rígidas, considere o seguinte sistema de duas EDOs:

$$\frac{dx}{dt} = 998x - 1998y \quad (8.200)$$

$$\frac{dy}{dt} = 1000x - 2000y \quad (8.201)$$

com  $x(0) = 1$  e  $y(0) = 2$ . Uma solução analítica para o sistema pode ser obtida com a subtração das duas EDOs, tratando  $(x - y)$  como a variável dependente. Com isso:

$$y - x = e^{-2t} \quad (8.202)$$

Substituindo  $y$  da Eq. (8.202) na Eq. (8.200), obtém-se:

$$\frac{dx}{dt} = 998x - 1998(x + e^{-2t}) = -1000x - 1998e^{-2t} \quad (8.203)$$

que tem a solução:

$$x(t) = -\frac{1998}{998}e^{-2t} + \frac{2996}{998}e^{-1000t} \quad (8.204)$$

e, da Eq (8.200),  $y$  é:

$$y(t) = x(t) + e^{-2t} = -\frac{1000}{998}e^{-2t} + \frac{2996}{998}e^{-1000t} \quad (8.205)$$

Avaliando  $x$  e  $y$  em  $t = 0,1$ , as respostas exatas são:  $x(0,1) = -1,63910225$  e  $y(0,1) = -0,82037150$ . Note que, nas soluções (8.202) e (8.203), os termos contendo  $e^{-1000t}$  têm uma contribuição desprezível para a resposta. No entanto, são precisamente esses termos que forçam a escolha de passos de tempo pequenos para que a solução numérica seja precisa.

O uso do método explícito de Euler para resolver o sistema das Eqs. (8.200) e (8.201) com um passo de integração de  $h = 0,1$  fornece os seguintes resultados para  $x(0,1)$  e  $y(0,1)$ , que são os valores das variáveis dependentes após o primeiro passo:

$$x(0,1) = x_1 + (998x_1 - 1998y_1)h = 1 + (998 \cdot 1 - (1998 \cdot 2))0,1 = -298,8$$

$$y(0,1) = y_1 + (1000x_1 - 2000y_1)h = 2 + (1000 \cdot 1 - (2000 \cdot 2))0,1 = -298$$

Claramente, as respostas  $x(0,1) = -298,8$  e  $y(0,1) = -298$  estão completamente erradas. Conforme discutido na Seção 8.13, considerações de estabilidade requerem o uso de um passo de integração que seja menor que um valor crítico para que uma resposta precisa seja obtida. Por exemplo, se um passo de integração de  $h = 0,0001$  for utilizado, as soluções numéricas obtidas no primeiro passo são  $x(0,1) = -1,63906946$  e  $y(0,1) = -0,820355090$ , que se aproximam da solução exata. À medida que passos de integração cada vez menores são usados, as respostas se tornam ligeiramente mais precisas até que os erros de arredondamento começam a crescer. Assim, pode-se ver que existe um limite de precisão que pode ser atingido para equações rígidas quando se utiliza um método explícito como o de Euler.

O emprego de métodos implícitos pode levar a resultados mais precisos na solução de equações rígidas. O uso do método implícito de Euler na solução das Eqs. (8.200) e (8.201) com um passo de integração de  $h = 0,1$  leva às seguintes equações no primeiro passo de integração:

$$x_2 = x_1 + (998x_2 - 1998y_2)h = 1 + (998x_2 - 1998y_2)0,1$$

$$y_2 = y_1 + (1000x_2 - 2000y_2)h = 2 + (1000x_2 - 2000y_2)0,1$$

que resultam no seguinte sistema de duas equações simultâneas:

$$\begin{bmatrix} 98,8 & -199,8 \\ 100 & -201 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} \quad (8.206)$$

A solução do sistema da Eq. (8.206) para  $x_2$  e  $y_2$  fornece a solução das EDOs no final do primeiro passo:  $x(0,1) = -1,63861386$  e  $y(0,1) = -0,80528053$ . Esse resultado é muito mais preciso do que as respostas obtidas com o método explícito de Euler para o mesmo passo de integração. Esses resultados, resumidos na Tabela 8-3, mostram que, quando EDOs rígidas são resolvidas, o passo de integração a ser utilizado depende essencialmente da estabilidade do método numérico utilizado, e não da precisão desejada.

**Tabela 8-3** Soluções de um sistema de duas EDOs rígidas

	Solução exata	Método explícito de Euler com $h = 0,1$	Método explícito de Euler com $h = 0,0001$	Método implícito de Euler com $h = 0,1$
$x(0,1)$	-1,63910225	-298,8	-1,63906946	-1,63861386
$y(0,1)$	-0,82037150	-298	-0,82035509	-0,80528053

Como o método implícito de Euler é incondicionalmente estável, ele é muito mais eficiente na solução de EDOs rígidas do que o método explícito de Euler, que está restrito a limites de estabilidade.

O MATLAB contém várias funções residentes voltadas especificamente para a solução de sistemas rígidos. Elas são usualmente identificadas por um sufixo *s* que aparece em seu nome. Por exemplo, *ode15s* e *ode23s* são funções que podem ser usadas na solução de sistemas rígidos. Além disso, *ode23t* e *ode23tb* também podem ser usados para resolver EDOs moderadamente rígidas.

Em resumo, EDOs rígidas são EDOs que apresentam escalas temporais (ou espaciais) que variam amplamente. Na solução de um problema de valor inicial formado por um conjunto de EDOs rígidas, métodos explícitos requerem passos de integração muito pequenos, de acordo com os critérios de estabilidade pertinentes. Com isso, a solução desse sistema se torna (1) muito longa ou (2) imprecisa em função do aumento dos erros de arredondamento associado ao uso de passos de integração muito pequenos. Em contraste, métodos implícitos estão livres de restrições de estabilidade e, por isso, são preferidos para a solução de EDOs rígidas. Métodos preditor-corretor também podem ser úteis na solução de alguns problemas moderadamente rígidos.

## 8.15 PROBLEMAS

### Problemas de solução manual

Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos. Se usar o MATLAB, não utilize funções residentes para obter soluções numéricas.

**8.1** Considere a EDO de primeira ordem a seguir:

$$\frac{dy}{dx} = yx - x^3 \quad \text{de} \quad x = 0 \quad \text{a} \quad x = 1,8 \quad \text{com} \quad y(0) = 1$$

- (a) Resolva a equação manualmente usando o método explícito de Euler com  $h = 0,6$ .

- (b) Resolva a equação manualmente usando o método de Euler modificado com  $h = 0,6$ .

- (c) Resolva a equação manualmente usando o método de Runge-Kutta de quarta ordem clássico com  $h = 0,6$ .

A solução analítica da EDO é:  $y = x^2 - e^{\frac{1}{2}x^2} + 2$ . Em cada letra, calcule o erro existente entre a solução exata e a solução numérica nos pontos em que a solução numérica é determinada.

- 8.2** Considere a EDO de primeira ordem a seguir:

$$\frac{dy}{dx} = x - y \quad \text{de} \quad x = 0 \quad \text{a} \quad x = 1,5 \quad \text{com} \quad y(0) = 1$$

- (a) Resolva a equação manualmente usando o método explícito de Euler com  $h = 0,5$ .

- (b) Resolva a equação manualmente usando o método de Euler modificado com  $h = 0,5$ .

- (c) Resolva a equação manualmente usando o método de Runge-Kutta de quarta ordem clássico com  $h = 0,5$ .

A solução analítica da EDO é:  $y = x + 2e^{-x} - 1$ . Em cada letra, calcule o erro existente entre a solução exata e a solução numérica nos pontos em que a solução numérica é determinada.

- 8.3** Considere a EDO de primeira ordem a seguir:

$$\frac{dy}{dt} = y + t^3 \quad \text{de} \quad t = 0,5 \quad \text{a} \quad t = 2 \quad \text{com} \quad y(0,5) = -1$$

- (a) Resolva a equação manualmente usando o método explícito de Euler com  $h = 0,5$ .

- (b) Resolva a equação manualmente usando o método do ponto central com  $h = 0,5$ .

- (c) Resolva a equação manualmente usando o método de Runge-Kutta de quarta ordem clássico com  $h = 0,5$ .

A solução analítica da EDO é:  $y = -t^3 - 3t^2 - 6t - 6 + (71e^t)/(8e^{0.5})$ . Em cada letra, calcule o erro existente entre a solução exata e a solução numérica nos pontos em que a solução numérica é determinada.

- 8.4** Considere o seguinte sistema de duas EDOs:

$$\frac{dx}{dt} = x + y \quad \frac{dy}{dt} = y - x \quad \text{de} \quad t = 0 \quad \text{a} \quad t = 2 \quad \text{com} \quad x(0) = 1 \quad \text{e} \quad y(0) = 3.$$

- (a) Resolva a equação manualmente usando o método explícito de Euler com  $h = 0,5$ .

- (b) Resolva a equação manualmente usando o método de Euler modificado com  $h = 0,5$ .

A solução analítica do sistema é:  $x = e^t[3\sin(t) + \cos(t)]$ ,  $y = e^t[3\cos(t) - \sin(t)]$ . Em cada letra, calcule o erro existente entre a solução exata e a solução numérica nos pontos em que a solução numérica é determinada.

**8.5** Considere o seguinte sistema de duas EDOs:

$$\frac{dx}{dt} = xt - y \quad \frac{dy}{dt} = yt + x \quad \text{de } t = 0 \text{ a } t = 1,2 \text{ com } x(0) = 1 \quad \text{e} \quad y(0) = 0,5.$$

(a) Resolva a equação manualmente usando o método explícito de Euler com  $h = 0,4$ .

(b) Resolva a equação manualmente usando o método de Runge-Kutta de quarta ordem clássico com  $h = 0,4$ .

A solução analítica do sistema é:  $x = e^{\frac{1}{2}t^2} \left( \cos t - \frac{1}{2} \sin t \right)$ ,  $y = \left( -e^{\frac{1}{2}t^2} \right)$   $\left( -\sin t - \frac{1}{2} \cos t \right)$ . Em cada letra, calcule o erro existente entre a solução exata e a solução numérica nos pontos em que a solução numérica é determinada.

**8.6** Escreva as seguintes EDOs de segunda ordem como um sistema de duas EDOs de primeira ordem:

$$(a) \frac{d^2y}{dx^2} = -Py + \frac{QI}{2}x - \frac{Q}{2}x^2, \text{ onde } E, I, P, Q \text{ e } L \text{ são constantes.}$$

$$(b) \frac{d^2y}{dx^2} = M \left[ 1 + \left( \frac{dy}{dx} \right)^2 \right]^{3/2}, \text{ onde } E, I \text{ e } M \text{ são constantes.}$$

**8.7** Escreva as seguintes EDOs de segunda ordem como um sistema de duas EDOs de primeira ordem:

$$(a) \frac{1}{g} \frac{d^2h}{dt^2} = \frac{T}{w} - 1 - \frac{0,008}{w} \left( \frac{dh}{dt} \right)^2, \text{ onde } g, T \text{ e } w \text{ são constantes.}$$

$$(b) \frac{d^2Q}{dt^2} + \frac{500dQ}{15dt} + \frac{250}{15} \left( \frac{dQ}{dt} \right)^3 + \frac{Q}{15 \cdot 4,2 \times 10^{-6}} = \frac{1000}{15}.$$

**8.8** Escreva o seguinte sistema de duas EDOs de segunda ordem como um sistema de quatro EDOs de primeira ordem:

$$\frac{d^2x}{dt^2} = -\frac{\gamma}{m} \left( \frac{dx}{dt} \right) \sqrt{\left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2} \quad \frac{d^2y}{dt^2} = -g - \frac{\gamma}{m} \left( \frac{dy}{dt} \right) \sqrt{\left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2}$$

### Problemas de programação no MATLAB.

Resolva os seguintes problemas no ambiente MATLAB. Não use funções residentes para resolver as equações diferenciais.

**8.9** Use a função `edoEuler` (Programa 8-1 listado no Exemplo 8-1) para resolver o Problema 8.1. Escreva um programa no MATLAB que resolva a EDO do Problema 8.1 três vezes, primeiro usando  $h = 0,6$ , depois  $h = 0,3$  e, finalmente, uma outra vez usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.1) juntamente com as três soluções numéricas (todas na mesma figura).

**8.10** Use a função `edoEuler` (Programa 8-1 listado no Exemplo 8-1) para resolver o Problema 8.2. Escreva um programa no MATLAB que resolva a EDO do Problema 8.2 três vezes, primeiro usando  $h = 0,5$ , depois  $h = 0,25$  e, finalmente, uma outra usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.2) juntamente com as três soluções numéricas (todas na mesma figura).

**8.11** Escreva uma função no MATLAB que resolva uma EDO de primeira ordem aplicando o método do ponto central (use a forma do método de Runge-Kutta de segunda ordem, Eqs. (8.65), (8.66)). Use como nome da função e argumentos `[x, y] =edoPontoCentral(EDO, a, b, h, yINI)`, onde `EDO` é o nome (*string*) da função em arquivo que calcula  $dy/dx$ , `a` e `b` definem o domínio da solução, `h` é o passo de integração e `yINI` é o valor inicial. Os argumentos de saída, `x` e `y`, são vetores com as coordenadas  $x$  e  $y$  da função.

Use a função `edoPontoCentral` para resolver a EDO do Problema 8.2. Escreva um programa no MATLAB que a resolva duas vezes, primeiro usando  $h = 0,5$  e depois usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.2) juntamente com as duas soluções numéricas (todas na mesma figura).

**8.12** Escreva uma função no MATLAB que resolva uma EDO de primeira ordem aplicando o método de Runge-Kutta de terceira ordem clássico, Eqs. (8.82), (8.83). Use como nome da função e argumentos `[x, y] =edoRK3(EDO, a, b, h, yINI)`, onde `EDO` é o nome (*string*) da função em arquivo que calcula  $dy/dx$ , `a` e `b` definem o domínio da solução, `h` é o passo de integração e `yINI` é o valor inicial. Os argumentos de saída, `x` e `y`, são vetores com as coordenadas  $x$  e  $y$  da função.

Use a função `edoRK3` para resolver a EDO do Problema 8.2. Escreva um programa no MATLAB que a resolva duas vezes, primeiro usando  $h = 0,5$  e depois usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.2) juntamente com as duas soluções numéricas (todas na mesma figura).

**8.13** Escreva uma função no MATLAB que resolva uma EDO de primeira ordem aplicando o método de Adams-Bashforth de segunda ordem. A função deve usar o método de Euler modificado para calcular a função no segundo ponto, e a Eq. (8.95) para calcular a solução no restante dos pontos. Use como nome da função e argumentos `[x, y] =edoAdams2(EDO, a, b, h, yINI)`, onde `EDO` é o nome (*string*) da função em arquivo que calcula  $dy/dx$ , `a` e `b` definem o domínio da solução, `h` é o passo de integração e `yINI` é o valor inicial. Os argumentos de saída, `x` e `y`, são vetores com as coordenadas  $x$  e  $y$  da função.

Use a função `edoAdams2` para resolver a EDO do Problema 8.3. Escreva um programa no MATLAB que a resolva usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.3) juntamente com a solução numérica (ambas na mesma figura).

**8.14** Escreva uma função no MATLAB que resolva uma EDO de primeira ordem usando o método preditor-corretor de Euler modificado (veja o algoritmo na

Seção 8.7). Em cada passo, as iterações devem continuar até que o erro relativo

estimado seja menor que 0,001, ou seja,  $\left| \frac{y_{i+1}^{(k)} - y_{i+1}^{(k-1)}}{y_{i+1}^{(k-1)}} \right| \leq \varepsilon = 0,001$ . Use como

nome da função e argumentos  $[x, y] = \text{edoEulerPreCor}(EDO, a, b, h, yINI)$ , onde  $EDO$  é o nome (*string*) da função em arquivo que calcula  $dy/dx$ ,  $a$  e  $b$  definem o domínio da solução,  $h$  é o passo de integração e  $yINI$  é o valor inicial. Os argumentos de saída,  $x$  e  $y$ , são vetores com as coordenadas  $x$  e  $y$  da função.

Use a função `edoEulerPreCor` para resolver a EDO do Problema 8.3. Escreva um programa no MATLAB que a resolva usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.3) juntamente com a solução numérica (ambas na mesma figura).

**8.15** A função `Sis2EDOsRK2(EDO1, EDO2, a, b, h, yINI, zINI)` (Programa 8-5), listada na solução do Exemplo 8-7, resolve um sistema de duas EDOs usando o método de Runge-Kutta de segunda ordem (versão Euler modificado). Modifique a função de forma que as duas EDOs sejam fornecidas em apenas um argumento de entrada. Similarmente, o domínio e as duas condições iniciais também devem ser fornecidos como um argumento de entrada cada. Use como nome da função e argumentos  $[t, x, y] = \text{Sis2EDOsEuMod}(EDOs, ab, h, INI)$ , onde  $EDOs$  é o nome (*string*) da função em arquivo que calcula  $\frac{dx}{dt}$  e  $\frac{dy}{dt}$  (os argumentos de entrada dessa função são  $t$ ,  $x$ ,  $y$ , e seu argumento de saída é um vetor de dois elementos contendo os valores de  $\frac{dx}{dt}$  e  $\frac{dy}{dt}$ ),  $ab$  é um vetor de dois elementos que define o domínio da solução,  $h$  é o passo de integração e  $INI$  é um vetor de dois elementos que define os valores iniciais. Os argumentos de saída,  $t$ ,  $x$  e  $y$ , são os vetores da solução.

Use a função `Sis2EDOsEuMod` para resolver o sistema de EDOs do Problema 8.5. Escreva um programa no MATLAB que a resolva usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.5) juntamente com a solução numérica (ambas na mesma figura).

**8.16** A função `Sis2EDOsRK4(EDO1, EDO2, a, b, h, x1, y1)` (Programa 8-6), listada na solução do Exemplo 8-8, resolve um sistema de duas EDOs usando o método de Runge-Kutta de quarta ordem. Modifique a função de forma que as duas EDOs sejam fornecidas em apenas um argumento de entrada. Similarmente, o domínio e as duas condições iniciais também devem ser fornecidos como um argumento de entrada cada. Use como nome da função e argumentos  $[t, x, y] = \text{Sis2EDOsRKclas}(EDOs, ab, h, INI)$ , onde  $EDOs$  é o nome (*string*) da função em arquivo que calcula  $dy/dt$  e  $dy/dt$  (os argumentos de entrada dessa função são  $t$ ,  $x$ ,  $y$ , e seu argumento de saída é um vetor de dois elementos contendo os valores de  $dx/dt$  e  $dy/dt$ ),  $ab$  é um vetor de dois elementos que define o domínio da solução,  $h$  é o passo de integração e  $INI$  é um vetor com dois elementos que define os valores iniciais. Os argumentos de saída,  $t$ ,  $x$  e  $y$ , são os vetores da solução.

Use a função `Sis2EDOsRKclas` para resolver o sistema de EDOs do Problema 8.5. Escreva um programa no MATLAB que a resolva usando  $h = 0,1$ . O programa também deve traçar a solução exata (dada no Problema 8.5) juntamente com a solução numérica (ambas na mesma figura).

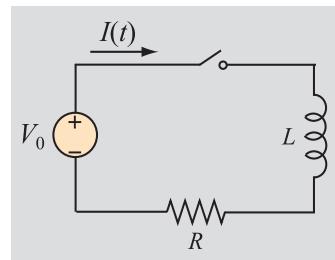
- 8.17** Escreva um programa que resolva o PVI do Exemplo 8-13 usando o método implícito de Euler com  $h = 0,1$ . Trace um gráfico com as soluções numérica e analítica.

### Problemas na matemática, na ciência e na engenharia

Resolva os seguintes problemas no ambiente MATLAB. Use programas apresentados neste capítulo, programas desenvolvidos em problemas já resolvidos ou funções residentes do MATLAB.

- 8.18** Um indutor e um resistor não-linear de resistência  $R = 500 + 250I^2 \Omega$  estão conectados em série com uma fonte de tensão CC e uma chave. A chave está inicialmente aberta, sendo então fechada no tempo  $t = 0$ . A corrente  $I$  no circuito para  $t > 0$  é determinada a partir da solução da equação:

$$\frac{dI}{dt} = \frac{V_0}{L} - \frac{R}{L}I$$



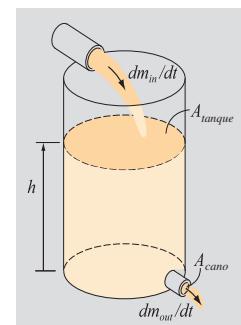
Para  $V_0 = 1000$  V e  $L = 15$  H, determine e trace a corrente em função do tempo em  $0 \geq t \geq 0,1$  s.

- (a) Resolva o problema com a função `edoPontoCentral` escrita no Problema 8.11. Use 0,005 como passo de integração.  
 (b) Resolva o problema usando uma das funções residentes do MATLAB.

- 8.19** Considere o tanque de água cilíndrico mostrado na Fig. 8-1 (repetido ao lado). Enche-se o tanque por cima e a água sai por um cano conectado no fundo. A taxa de variação da altura do nível d'água  $h$  é dada pela Eq. (8.6):

$$\rho A_{tanque} \frac{dh}{dt} = K_1 + K_2 \cos\left(\frac{\pi}{12}t\right) - \rho A_{cano} \sqrt{2gh}$$

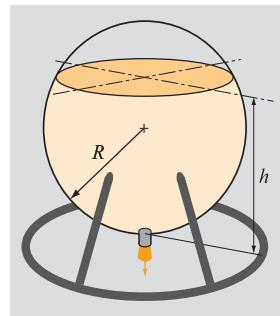
No tanque em questão,  $A_{tanque} = 3,13 \text{ m}^2$ ,  $A_{cano} = 0,06 \text{ m}^2$ ,  $K_1 = 300 \text{ kg/h}$ ,  $K_2 = 200 \text{ kg/h}$ . Além disso,  $\rho = 1000 \text{ kg/m}^3$ , e  $g = 9,81 \text{ m/s}^2$ . Determine e trace um gráfico com a altura do nível d'água em função do tempo em  $0 \leq t \leq 150$  s, se, em  $t = 0$ ,  $h = 3$  m.



- (a) Use a função `edoRK3` escrita no Problema 8.12, com um passo de integração de 0,1 s.  
 (b) Use uma das funções residentes do MATLAB.

**8.20** Um tanque esférico de raio  $R = 4$  m é esvaziado por meio de um pequeno buraco circular de raio  $r = 0,02$  m localizado no fundo. O topo do tanque está aberto. O nível d'água instantâneo no tanque,  $h$  (medido a partir do fundo do tanque, no dreno), pode ser determinado a partir da solução da seguinte EDO:

$$\frac{dh}{dt} = -\frac{r^2 \sqrt{2gh}}{2hR - h^2}$$

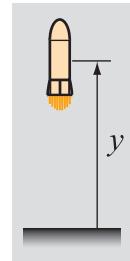


onde  $g = 9,81 \text{ m/s}^2$ . Se o nível d'água inicial em  $t = 0$  é  $h = 6 \text{ m}$ , determine o tempo necessário para drenar o tanque até um nível de  $0,5 \text{ m}$ .

- (a) Use o método de Runge-Kutta de quarta ordem (função `edORK`, listada no Programa 8-4 e desenvolvida no Exemplo 8-6).
- (b) Use uma das funções residentes do MATLAB.

**8.21** Um pequeno foguete com peso inicial de  $1360 \text{ kg}$  (incluindo  $90 \text{ kg}$  de combustível), inicialmente em repouso, é lançado verticalmente. O foguete queima o combustível em uma taxa constante de  $36 \text{ kg/s}$ , o que resulta em uma força de propulsão  $T$  constante, de  $31400 \text{ N}$ . O peso instantâneo do foguete é  $w(t) = 13500 - 360t \text{ N}$ . A força de arrasto  $D$  sentida pelo foguete é dada por  $D = 0,036g\left(\frac{dy}{dt}\right)^2 \text{ N}$ , onde  $y$  é a distância em pés, e  $g = 9,81 \text{ m/s}^2$ . Usando a lei de Newton, a equação do movimento para o foguete é dada por:

$$\frac{w}{g} \frac{d^2y}{dt^2} = T - w - D$$



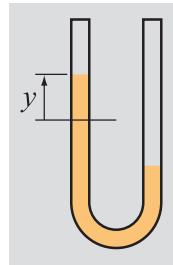
Determine e trace a posição, a velocidade e a aceleração do foguete (três figuras separadas em uma página) em função do tempo, de  $t = 0$ , quando o foguete deixa o repouso, até  $t = 3 \text{ s}$ . Reduza a EDO de segunda ordem a um sistema de duas EDOs de primeira ordem.

- (a) Use o método de Runge-Kutta de quarta ordem. Utilize a função `Sis2EDOsRK4` (Programa 8-6), desenvolvida no Exemplo 8-8, ou a função `Sis2EDOsRKclas` (escrita no Problema 8.16). Trabalhe com um passo de integração de  $0,05 \text{ s}$ .
- (b) Use uma das funções residentes do MATLAB.

**8.22** Um manômetro em formato de U (usado para medir pressão) está inicialmente cheio de água, mas é exposto a uma diferença de pressão tal que o nível d'água em seu lado esquerdo fica  $0,05 \text{ m}$  mais alto que o nível d'água em seu lado direito. Em  $t = 0$ , a diferença de pressão é repentinamente removida. Quando

se despreza o atrito, a altura do nível d'água no lado esquerdo y medida a partir do plano central entre os dois níveis d'água iniciais é dada pela solução da equação:

$$L \frac{d^2y}{dt^2} = -2gy$$



onde  $L = 0,2$  m é o comprimento total do tubo em forma de U, e  $g = 9,81$  m/s<sup>2</sup>. Resolva a EDO e trace  $y$  e  $\frac{dy}{dt}$  (duas figuras separadas em uma página) em função de  $t$  nos primeiros 10 s. Transforme a EDO de segunda ordem em duas EDOs de primeira ordem e resolva o sistema usando:

- (a) a função `Sis2EDOsEuMod` escrita no Problema 8.15. Use um passo de integração de 0,02 s.
- (b) uma função residente do MATLAB.

**8.23** Se o efeito do atrito for incluído na análise do manômetro em forma de U do Problema 8.22, a altura do nível d'água  $y$  passa a ser dada pela solução da equação:

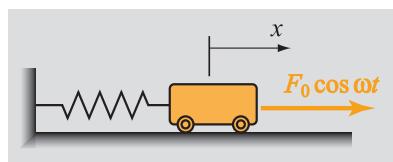
$$L \frac{d^2y}{dt^2} = -0,5 \frac{dy}{dt} - 2gy$$

Resolva a EDO e trace  $y$  e  $dy/dt$  (duas figuras separadas em uma página) em função de  $t$  nos primeiros 10 s. Transforme a EDO de segunda ordem em duas EDOs de primeira ordem e resolva o sistema usando:

- (a) a função `Sis2EDOsRKclas` escrita no Problema 8.16. Use um passo de integração de 0,02 s.
- (b) uma função residente do MATLAB.

**8.24** Considere a vibração forçada do sistema massa-mola mostrado na figura. A posição  $x$  da massa em função do tempo é dada pela solução da equação:

$$\frac{d^2x}{dt^2} = -\frac{k}{m}x + \frac{F_0}{m}\cos\omega t$$

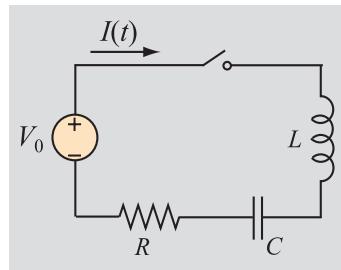


onde  $m = 2$  kg é a massa,  $k = 800$  N/m é a constante da mola,  $F_0 = 50$  N é a amplitude da força harmônica aplicada e  $\omega = 3$  rad/s é a freqüência dessa força.

As condições iniciais são  $x(0) = 0,1$  m e  $\left.\frac{dx}{dt}\right|_{t=0} = 0,1$  m/s. Resolva a EDO em  $0 \leq t \leq 10$  s e trace  $x$  e  $\frac{dx}{dt}$  em função de  $t$  (dois gráficos separados na mesma página).

- (a) Use a função `Sis2EDOsRKclas` escrita no Problema 8.16. Use um passo de integração de 0,01 s.
- (b) Use uma função residente do MATLAB.

**8.25** Um capacitor de  $C = 4,2 \mu\text{F}$  é colocado em série com o circuito do Problema 8.18. Conforme mostrado na figura, o circuito contém uma fonte de tensão CC,  $V_0 = 1000$  V, um indutor de  $L = 15 \text{ H}$  e uma resistência não-linear  $R = R_0 + R_1 I^2 \Omega$ , onde  $R_0 = 500 \Omega$  e  $R_1 = 250 \Omega/\text{A}^2$ . A chave está inicialmente aberta, sendo então fechada no tempo  $t = 0$ . A carga  $Q$  no capacitor em  $t > 0$  é determinada a partir da solução da equação:



$$\frac{d^2Q}{dt^2} + \frac{R_0 dQ}{L dt} + \frac{R_1 (dQ/dt)^3}{L} + \frac{Q}{LC} = \frac{V_0}{L}$$

Inicialmente,  $Q = 0$  e  $dQ/dt = 0$ .

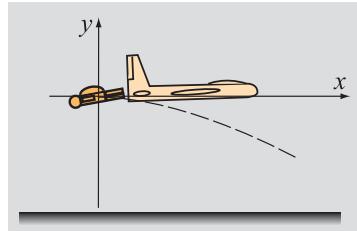
- (a) reduza a EDO de segunda ordem em um sistema de duas EDOs de primeira ordem e determine a carga  $Q$  em função do tempo em  $0 \geq t \geq 0,3\text{s}$  resolvendo o sistema com o método de Runge-Kutta de quarta ordem.

Utilize a função Sis2EDOsRK4 (Programa 8-6), desenvolvida no Exemplo 8-8, ou a função Sis2EDOsRKclas (escrita no Problema 8.16). Trabalhe com um passo de integração de  $0,002\text{s}$ .

- (b) Use os resultados obtidos na letra (a) para traçar um gráfico com a corrente no circuito. A corrente é dada pela derivada temporal da carga,  $I = \frac{dQ}{dt}$ .
- (c) Resolva o problema (letras (a) e (b)) usando funções residentes do MATLAB.

**8.26** Um pára-quedista salta de um avião em vôo retilíneo e estável. O movimento do pára-quedista é descrito aproximadamente pelo seguinte conjunto de equações:

$$\begin{aligned}\frac{d^2x}{dt^2} &= -\frac{\gamma}{m} \left( \frac{dx}{dt} \right) \sqrt{\left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2} \\ \frac{d^2y}{dt^2} &= -g - \frac{\gamma}{m} \left( \frac{dy}{dt} \right) \sqrt{\left( \frac{dx}{dt} \right)^2 + \left( \frac{dy}{dt} \right)^2}\end{aligned}$$



onde  $x$  e  $y$  descrevem a posição do pára-quedista de acordo com o sistema de coordenadas mostrado na figura. Para  $m = 80 \text{ kg}$ ,  $g = 9,81 \text{ m/s}^2$ ,  $\gamma = 5,38 \text{ Ns}^2/\text{m}^2$  e condições iniciais:

$$x(0) = 0, \quad y(0) = 0, \quad \left. \frac{dx}{dt} \right|_{t=0} = 134 \text{ m/s}, \quad \left. \frac{dy}{dt} \right|_{t=0} = 0$$

determine e trace a trajetória do pára-quedista nos primeiros 5 s. Reduza o sistema a um conjunto de quatro EDOs de primeira ordem e o resolva usando funções residentes do MATLAB.

# Equações Diferenciais Ordinárias: Problemas de Valor de Contorno

## Tópicos centrais

- O método do tiro (9.2)
- O método das diferenças finitas (9.3)
- Uso de funções residentes do MATLAB para resolver problemas de valor de contorno (9.4)

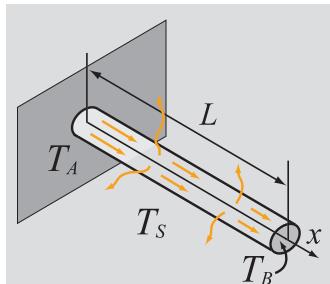
## Tópicos complementares

- Erro e estabilidade na solução numérica de problemas de valor de contorno (9.5)

## 9.1 FUNDAMENTOS

Uma solução específica pode ser determinada para uma equação diferencial se as restrições do problema forem conhecidas. Uma EDO de primeira ordem pode ser resolvida se uma restrição correspondente ao valor da variável dependente em um único ponto for conhecida (condição inicial). Para resolver uma equação de  $n$ -ésima ordem,  $n$  restrições devem ser conhecidas. As restrições podem ser o valor da variável dependente (solução) e suas derivadas em determinados valores da variável independente.

Quando todas as restrições são especificadas em um único valor da variável independente, o problema é chamado de problema de valor inicial (PVI). A solução de problemas como esse é descrita no Capítulo 8. Em muitos casos, no entanto, é necessário resolver equações diferenciais de segunda ordem e de ordem superior que têm restrições especificadas em valores diferentes da variável independente. Esses problemas são chamados de **problemas de valor de contorno** (PVC). Suas restrições são freqüentemente especificadas nos pontos finais, ou limites, do domínio da solução.



**Figura 9-1** Fluxo de calor em um dissipador em forma de pino.

Como um exemplo, considere a modelagem da distribuição de temperatura em um dissipador em forma de pino usado para resfriar um objeto (Fig. 9-1). Se convecção e radiação forem incluídas na análise, a temperatura  $T(x)$  em regime permanente ao longo do dissipador pode ser obtida com a solução de uma equação na forma:

$$\frac{d^2T}{dx^2} - \alpha_1(T - T_S) - \alpha_2(T^4 - T_S^4) = 0 \quad (9.1)$$

onde  $T_s$  é a temperatura ambiente, e  $\alpha_1$  e  $\alpha_2$  são coeficientes. A Eq. (9.1) é uma EDO de segunda ordem que pode ser resolvida com a especificação de duas condições de contorno. Tais condições podem ser as temperaturas nas extremidades do dissipador,  $T_A$  e  $T_B$ .

### Enunciado de um problema de valor de contorno de segunda ordem

Considerando uma variável independente  $x$  e uma variável dependente  $y$ , o enunciado de um problema de valor de contorno consiste na equação diferencial:

$$\frac{d^2y}{dx^2} = f(x, y, \frac{dy}{dx}) \quad (9.2)$$

no domínio da solução,  $a \leq x \leq b$ , e nas condições de contorno. As duas condições de contorno necessárias para a solução são fornecidas tipicamente nos pontos finais do domínio. Como uma condição de contorno pode ser um valor de  $y$  ou de sua derivada  $\frac{dy}{dx}$ , as condições de contorno podem ser especificadas de diferentes maneiras. Formas comuns das condições de contorno são:

- Dois valores de  $y$  são fornecidos – um em  $x = a$  e outro em  $x = b$ :

$$y(a) = Y_a \quad \text{e} \quad y(b) = Y_b \quad (9.3)$$

Essas condições são chamadas de condições de contorno de Dirichlet. Uma EDO de segunda ordem com essas condições de contorno é chamada de PVC com dois pontos.

- Dois valores de  $\frac{dy}{dx}$  são fornecidos – um em  $x = a$  e outro em  $x = b$ :

$$\left. \frac{dy}{dx} \right|_{x=a} = D_a \quad \text{e} \quad \left. \frac{dy}{dx} \right|_{x=b} = D_b \quad (9.4)$$

Essas condições são chamadas de condições de contorno de Neumann.

- A terceira possibilidade envolve condições de contorno mistas, que podem ser escritas na forma:

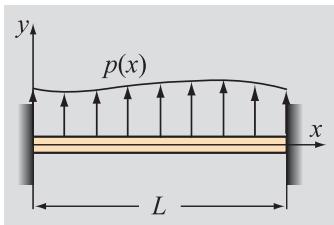
$$c_1 \left. \frac{dy}{dx} \right|_{x=a} + c_2 y(a) = C_a \quad \text{e} \quad c_3 \left. \frac{dy}{dx} \right|_{x=b} + c_4 y(b) = C_b \quad (9.5)$$

onde  $c_1$ ,  $c_2$ ,  $c_3$  e  $c_4$  são constantes. Essas condições são chamadas de condições de contorno mistas. Casos especiais ocorrem quando  $c_1 = 0$  e  $c_4 = 0$ , ou quando  $c_2 = 0$  e  $c_3 = 0$ . Nesses casos, o valor de  $y$  é dado em um ponto final e o valor da derivada é dado no outro ponto final.

Também é possível ter condições de contorno não-lineares, onde  $D_a$ ,  $D_b$ ,  $c_1$ ,  $c_2$ ,  $C_a$  ou  $C_b$  são funções não-lineares de  $y$  ou de suas derivadas.

Problemas de valor de contorno com EDOs de ordem maior que dois requerem condições de contorno adicionais, que são tipicamente valores da derivada de  $y$  de ordem mais elevada. Por exemplo, a equação diferencial que relaciona a deflexão  $y$  de uma viga associada à aplicação de uma carga distribuída  $p(x)$  é:

$$\frac{d^4y}{dx^4} = \frac{1}{EI} p(x) \quad (9.6)$$



**Figura 9-2** Viga presa submetida a uma carga distribuída.

onde  $E$  e  $I$  são o módulo elástico do material da viga e o momento de inércia da área da seção reta da viga, respectivamente. Essa EDO de quarta ordem pode ser resolvida se quatro condições de contorno, que dependem da maneira na qual a viga é suportada, forem especificadas. A viga mostrada na Fig. 9-2 está presa em ambas as extremidades, o que significa que a deflexão e a inclinação da curva de deflexão são iguais a zero nas extremidades. Isso fornece as quatro condições de contorno a seguir:

$$\begin{aligned} y(0) &= 0; & \frac{dy}{dx}\Big|_{x=0} &= 0 \\ y(L) &= 0; & \frac{dy}{dx}\Big|_{x=L} &= 0 \end{aligned} \quad (9.7)$$

### Visão geral de métodos numéricos usados na solução de problemas de valor de contorno

Este capítulo apresenta duas abordagens para a solução de problemas de valor de contorno: **métodos de tiro** e **métodos de diferenças finitas**. Métodos de tiro reduzem uma equação diferencial ordinária de segunda ordem ou de ordem superior a um problema de valor inicial. Isso é feito, conforme explicado na Seção 8.9, transformando essa equação em um sistema de EDOs de primeira ordem. O valor inicial no primeiro ponto do domínio é conhecido e utilizado como um dos valores iniciais no sistema. Os demais valores iniciais necessários são assumidos. O sistema é então resolvido, e a solução no final do intervalo é comparada com as condições de contorno ali especificadas. Se ambas apresentarem uma discordância maior que a precisão desejada, os valores iniciais assumidos são mudados e o sistema é resolvido novamente. Os cálculos são repetidos até que todas as condições de contorno específicas sejam satisfeitas. A diferença entre os vários métodos de tiro está na maneira na qual os valores das condições iniciais assumidas são modificados após cada cálculo.

Nos métodos de diferenças finitas, as derivadas que aparecem na equação diferencial são aproximadas por fórmulas de diferenças finitas (ver Capítulo 6). O domínio da solução é dividido em  $N$  subintervalos definidos por  $(N + 1)$  pontos (pontos de malha), e a equação diferencial é aproximada em cada ponto de malha pertencente ao domínio. Isso resulta em um sistema linear (ou não-linear) de equações algébricas. A solução desse sistema é a solução numérica da equação diferencial. A diferença entre os vários métodos de diferenças finitas está na fórmula de diferença finita usada na aproximação da equação diferencial.

Ambas as abordagens têm suas respectivas vantagens e desvantagens. Nos métodos de diferenças finitas, não é necessário resolver a equação diferencial várias vezes para que as condições de contorno nos pontos finais do domínio sejam satisfeitas. Por outro lado, a solução de EDOs não-lineares usando métodos de diferenças finitas resulta na necessidade de se resolver um sistema de equações não-lineares simultâneas (usualmente de forma iterativa), o que pode ser tedioso e feito com dificuldade. Métodos de tiro tem a vantagem de resolver EDOs

não-lineares de forma relativamente simples. A desvantagem desses métodos é a necessidade de se resolver a EDO várias vezes.

## 9.2 O MÉTODO DO TIRO

No método do tiro, um problema de valor de contorno (PVC) é transformado em um sistema de problemas de valor inicial (PVIs). Um PVC associado a uma EDO de segunda ordem pode ser transformado, conforme descrito na Seção 8.9, em duas EDOs de primeira ordem. Esse sistema de EDOs pode ser resolvido numericamente se a condição inicial de cada EDO for conhecida. O enunciado de um PVC associado a uma EDO de  $n$ -ésima ordem inclui  $n$  condições de contorno. Algumas dessas condições são fornecidas no primeiro ponto do domínio, e outras, no ponto final do domínio. Quando a EDO de  $n$ -ésima ordem é transformada em um sistema de  $n$  EDOs de primeira ordem, as condições de contorno fornecidas no primeiro ponto do domínio são usadas como condições iniciais para o sistema. As demais condições iniciais necessárias para a solução do problema são assumidas. O sistema é então resolvido, e a solução obtida no ponto final do domínio é comparada com as condições de contorno ali especificadas. Se a solução numérica não for suficientemente precisa, os valores iniciais assumidos são alterados, e o sistema é resolvido novamente. O processo é repetido até que a solução numérica concorde com as condições de contorno prescritas no ponto final do intervalo. Uma descrição detalhada do método do tiro para um PVC de segunda ordem é dada a seguir.

### Método do tiro para um PVC de segunda ordem

Considere o PVC associado a uma EDO de segunda ordem cujas condições de contorno são os valores da variável dependente nos pontos finais do domínio:

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \text{ para } a \leq x \leq b \quad \text{com} \quad y(a) = Y_a \quad \text{e} \quad y(b) = Y_b \quad (9.8)$$

Pode-se obter uma solução numérica com o método do tiro a partir do seguinte procedimento:

**Passo 1** A EDO é transformada em um sistema de dois PVIs de primeira ordem (ver Seção 8.9). As duas equações têm a forma;

$$\frac{dy}{dx} = w \quad \text{com a condição inicial } y(a) = Y_a \quad (9.9)$$

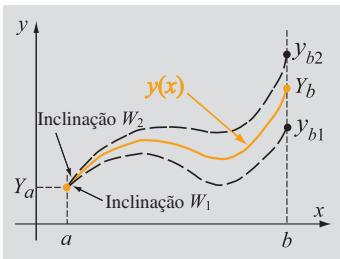
e

$$\frac{dw}{dx} = f(x, y, w) \quad (9.10)$$

O enunciado do problema não inclui uma condição inicial para a Eq. (9.10).

**Passo 2** Uma primeira estimativa (tentativa) é feita para o valor inicial da Eq. (9.10):

$$w(a) = \left. \frac{dy}{dx} \right|_{x=a} = W_1 \quad (9.11)$$



**Figura 9-3** Método do tiro.

Esta é, na realidade, uma estimativa para a inclinação em  $x = a$ . Com essa estimativa, resolve-se numericamente o sistema das Eqs. (9.9) e (9.10). A solução numérica em  $x = b$  (o final do intervalo) é  $y_{b1}$ , conforme mostrado na Fig. 9-3. Se a solução numérica estiver suficientemente próxima da condição de contorno  $Y_b$  (isto é, se o erro entre ambas for aceitável), então a solução foi obtida. Senão, o processo de solução continua no Passo 3.

**Passo 3** Uma segunda estimativa (tentativa) é feita para o valor inicial da Eq. (9.10):

$$w(a) = \left. \frac{dy}{dx} \right|_{x=a} = W_2 \quad (9.12)$$

Com essa estimativa, resolve-se novamente o sistema das Eqs. (9.9) e (9.10). A solução numérica em  $x = b$  (o final do intervalo) é  $y_{b2}$  (Fig. 9-3). Se a solução numérica estiver suficientemente próxima da condição de contorno  $Y_b$  (isto é, se o erro entre ambas for aceitável), então a solução foi obtida. Senão, o processo de solução continua no Passo 4.

**Passo 4** Uma terceira estimativa (tentativa) é feita para o valor inicial da Eq. (9.10):

$$w(a) = \left. \frac{dy}{dx} \right|_{x=a} = W_3 \quad (9.13)$$

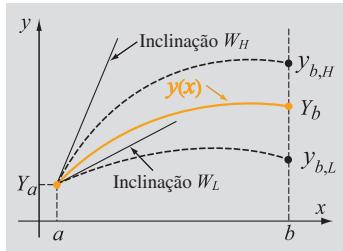
Vários métodos podem ser usados neste passo. Por exemplo, se o valor da condição inicial  $Y_b$  estiver entre  $y_{b1}$  e  $y_{b2}$  ( $y_{b1} < Y_b < y_{b2}$ ), conforme ilustrado na Fig. 9-3, o valor de  $W_3$  entre  $W_1$  e  $W_2$  pode ser obtido por meio de uma interpolação. Detalhes adicionais sobre como isso pode ser feito e sobre outros métodos são fornecidos mais adiante nesta seção.

**Passo 5** Usando  $w(a) = W_3$  como o valor inicial da Eq. (9.10), resolve-se novamente o sistema das Eqs. (9.9) e (9.10). Se a solução numérica em  $x = b$  (o final do intervalo) for igual à condição de contorno  $Y_b$  (ou se o erro entre ambas for aceitável), então a solução foi obtida. Senão, como feito no Passo 4, determina-se uma nova estimativa para o valor inicial do problema.

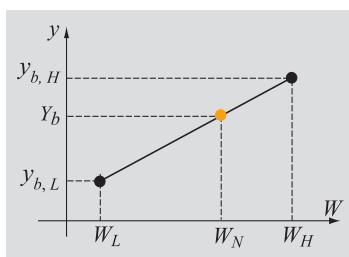
Os Passos 4 e 5 são repetidos até que a solução numérica em  $x = b$  concorde com a condição de contorno dentro uma precisão necessária. Em cada cálculo, uma nova condição de contorno é estimada para a Eq. (9.10) usando os resultados dos cálculos anteriores. Vários métodos podem ser usados com essa finalidade. Estes são apresentados a seguir.

### Estimando a inclinação (valor inicial) em $x = a$

Conforme descrito anteriormente, o processo de solução de um PVC de segunda ordem usando o método do tiro começa com a estimativa de dois valores para a inclinação de  $y(x)$  no primeiro ponto do domínio. Tentativas “inteligentes” podem ser feitas em muitas situações, especialmente quando a equação diferencial estiver associada a uma aplicação real. Depois, usando cada uma dessas tentativas, o



**Figura 9-4** Duas iterações em uma solução com o método do tiro.



**Figura 9-5** Interpolação linear para determinar  $W_N$ .

sistema de equações é resolvido para fornecer duas soluções no ponto final do domínio. Essas soluções são então usadas, no cálculo seguinte, para que se obtenha um novo valor de inclinação inicial que leve a uma solução mais precisa. O jeito mais simples de fazer isso é usar uma interpolação linear.

**Interpolação linear:** Considere duas soluções numéricas de um PVC de segunda ordem, mostradas na Fig. 9-4, obtidas assumindo-se  $W_H$  e  $W_L$  para a inclinação (condição inicial da Eq. (9.10)) em  $x = a$ . As soluções em  $x = b$  são  $y_{b,H}$  e  $y_{b,L}$ , respectivamente. Lembre que a condição de contorno neste ponto é  $y(b) = Y_b$ , e suponha que os valores  $y_{b,H}$  e  $y_{b,L}$  tenham sido obtidos de forma que  $Y_b \leq y_{b,H}$  e  $y_{b,L} \leq Y_b$ . Usando a interpolação linear ilustrada na Fig. 9-5, um novo valor pode ser determinado para a inclinação  $W_N$  correspondente a  $Y_b$  com:

$$W_N = W_L + (Y_b - y_{b,L}) \frac{W_H - W_L}{y_{b,H} - y_{b,L}} \quad (9.14)$$

O novo valor  $W_N$  é usado no cálculo seguinte, e a nova solução com  $y_{b,H}$  ou  $y_{b,L}$  (o que depende de a condição estar abaixo ou acima da condição de contorno) pode ser usada na interpolação linear para calcular a próxima estimativa para a inclinação inicial. O Exemplo 9-1 ilustra a aplicação do método do tiro com a interpolação linear. Obtém-se duas

soluções iniciais usando as duas tentativas para a inclinação no primeiro ponto do intervalo, o que permite uma solução muito mais precisa quando o problema é resolvido pela terceira vez.

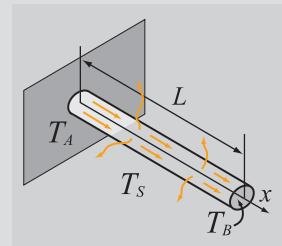
### Exemplo 9-1: Distribuição de temperaturas em um dissipador em forma de pino. Solução de uma EDO de segunda ordem (PVC) usando o método do tiro

Um dissipador em forma de pino é uma espécie de extensão fina presa a uma superfície de forma a aumentar a área superficial e permitir uma maior transferência de calor. Quando a convecção e a radiação são incluídas na análise, a temperatura  $T(x)$  em regime permanente ao longo de um dissipador em forma de pino pode ser calculada a partir da solução da equação:

$$\frac{d^2T}{dx^2} - \frac{h_c P}{kA_c}(T - T_S) - \frac{\epsilon\sigma_{SB}P}{kA_c}(T^4 - T_S^4) = 0, \quad 0 \leq x \leq L \quad (9.15)$$

com as seguintes condições de contorno:  $T(0) = T_A$  e  $T(L) = T_B$ .

Na Eq. (9.15),  $h_c$  é o coeficiente de transferência de calor por convecção,  $P$  é o perímetro da área da seção reta do dissipador,  $\epsilon$  é a emissividade radiativa da superfície do dissipador,  $k$  é a condutividade térmica do material do dissipador,  $A_c$  é a área da seção reta do dissipador,  $T_S$  é a temperatura ambiente e  $\sigma_{SB} = 5,67 \times 10^{-8} \text{ W}/(\text{m}^2\text{K}^4)$  é a constante de Stefan-Boltzmann.



Determine a distribuição de temperaturas se  $L = 0,1$  m,  $T(0) = 473$  K,  $T(0,1) = 293$  K e  $T_S = 293$  K. Use os seguintes valores para os parâmetros da Eq. (9.15):  $h_c = 40 \text{ W/m}^2/\text{K}$ ,  $P = 0,016 \text{ m}^2$ ,  $\epsilon = 0,4$ ,  $k = 240 \text{ W/m/K}$  e  $A_c = 1,6 \times 10^{-5} \text{ m}^2$ .

## SOLUÇÃO

A Eq. (9.15) é um PVC de segunda ordem não-linear. Para resolvê-la, ela é transformada em um sistema de duas EDOs de primeira ordem. A transformação é feita com a introdução de uma nova variável  $w = dT/dx$ . Com essa definição, o sistema é:

$$\frac{dT}{dx} = w \quad (9.16)$$

$$\frac{dw}{dx} = \frac{h_c P}{k A_c} (T - T_S) + \frac{\epsilon \sigma_{SB} P}{k A_c} (T^4 - T_S^4) \quad (9.17)$$

O valor inicial da Eq. (9.16) é  $T(0) = 473$ . A condição inicial da Eq. (9.17) é desconhecida.

Para ilustrar o método do tiro, o sistema formado pelas Eqs. (9.16) e (9.17) é resolvido três vezes assumindo-se três valores diferentes para a condição inicial da Eq. (9.17). Isso é feito em um programa no MATLAB que usa a função Sis2EDOsRK2 criada na Seção 8.8.2 (essa função resolve um sistema de duas EDOs de primeira ordem usando o método de Runge-Kutta).

As operações do programa, que está listado mais adiante, são:

- Resolve-se o sistema assumindo  $w(0) = -1000$ .
- Resolve-se o sistema assumindo  $w(0) = -3500$ .
- Usando a interpolação descrita pela Eq. (9.14), calcula-se um terceiro valor para  $w(0)$  a partir dos resultados obtidos nas duas primeiras soluções.
- Resolve-se o sistema usando o valor interpolado de  $w(0)$ .

Em cada solução, o programa lista a temperatura calculada no ponto final. Além disso, o programa mostra uma figura com as três soluções.

### Programa 9-1 Arquivo texto. Solução de uma EDO de segunda ordem usando o método do tiro.

```
% Solução do Exemplo 9-1
```

```
clear all
```

```
a = 0; b = 0.1; TINI = 473; wINI1 = -1000; h = 0.001;
```

Resolve o sistema assumindo  $w(0) = -1000$ .

```
[x, T1, w] = Sis2EDOsRK2('edoCap9Exmp1dTdx','edoCap9Exmp1dwdx',a,b,h,TINI,wINI1);
```

```
n = length(x);
```

```
fprintf('A temperatura em x=0.1 é %5.3f, para o valor inicial dt/dx= %4.1f\n',T1(n),wINI1)
```

```
wINI2 = -3500;
```

Resolve o sistema assumindo  $w(0) = -3500$ .

```
[x, T2, w] = Sis2EDOsRK2('edoCap9Exmp1dTdx','edoCap9Exmp1dwdx',a,b,h,TINI,wINI2);
```

```
fprintf('A temperatura em x=0.1 é %5.3f, para o valor inicial dt/dx= %4.1f\n',T2(n),wINI2)
```

```
wINI3 = wINI1 + (293 - T1(n))*(wINI2 - wINI1)/(T2(n) - T1(n));
```

Interpolação usando a Eq. (9.14).

```
[x, T3, w] = Sis2EDOsRK2('edoCap9Exmp1dTdx','edoCap9Exmp1dwdx',a,b,h,TINI,wINI3);
```

```
fprintf('A temperatura em x=0.1 é %5.3f, para o valor inicial dt/dx= %4.1f\n',T3(n),wINI3)
```

```
plot(x,T1,'-k',x,T2,'-k',x,T3,'r')
```

Resolve o sistema com o valor interpolado de  $w(0)$ .

```
xlabel('Distância (m)'); ylabel('Temperatura (K)')
```

Os primeiros dois argumentos da função `Sis2EDOsRK2` são os nomes de funções em arquivo que calculam os valores de  $dT/dx$ , Eq. (9.16), e  $dw/dx$ , Eq. (9.17). As duas funções, `edoCap9Exmp1dTdx` e `edoCap9Exmp1dwdx`, são:

```
function dTdx = edoCap9Exmp1dTdx(x,T,w)
dTdx = w;

function dwdx = edoCap9Exmp1dwdx(x,T,w)
hc = 40; P = 0.016; eps = 0.4; k = 240; Ac = 1.6E-5; Seg = 5.67E-8;
Ts = 293;
kAc = k*Ac;
A1 = hc*P/kAc; A2 = eps*Seg*P/kAc;
dwdx = A1*(T - Ts) + A2*(T^4 - Ts^4);
```

Quando o programa é executado, a seguinte mensagem é exibida na janela de comandos:

A temperatura em  $x = 0.1$  é 536.502, para o valor inicial  $dt/dx = -1000.0$

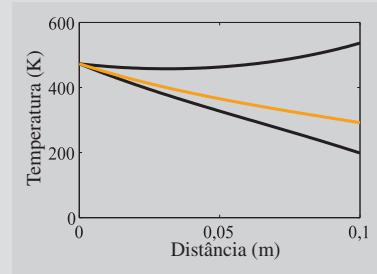
A temperatura em  $x = 0.1$  é 198.431, para o valor inicial  $dt/dx = -3500.0$

A temperatura em  $x = 0.1$  é 291.835, para o valor inicial  $dt/dx = -2800.7$

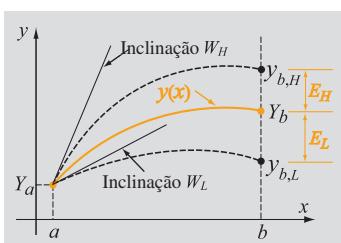
Além disso, o programa exibe uma figura com gráficos das três soluções, mostrada ao lado.

Os resultados mostram que, com o primeiro valor assumido para a inclinação,  $w(0) = -1000$ , a temperatura no ponto final é maior que a condição de contorno prescrita. A segunda hipótese,  $w(0) = -3500$ , resulta em um valor menor. A solução com o valor interpolado de  $w(0) = -2800,5$  fornece  $T(0,1) = 391,835$  K, que é um valor menor, porém mais próximo da condição de contorno de 293 K.

Pode-se esperar que uma solução mais precisa possa ser obtida com a execução de cálculos adicionais com melhores tentativas para  $w(0)$ .

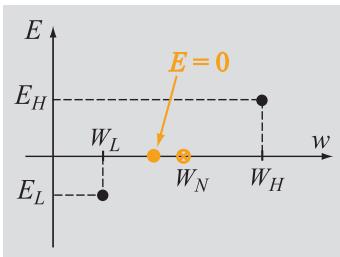


Um método mais sofisticado é considerar o erro (diferença entre a solução numérica e a condição de contorno fornecida no ponto final do domínio da solução) em função da inclinação inicial assumida, e usar os métodos do Capítulo 3 para obter a solução (o zero) dessa função. O uso dessa abordagem aplicando os métodos da bisseção e da secante é mostrado a seguir.



**Figura 9-6** Duas iterações em uma solução com o método do tiro.

**Método do tiro usando o método da bissecção:** Considere o PVC associado à Eq. (9.8), que é resolvido com a conversão do problema em um sistema de dois PVIs descrito pelas Eqs. (9.9) e (9.10). Duas soluções numéricas obtidas em cálculos anteriores são mostradas na Fig. 9-6. Em uma solução,  $y_{b,H}$ , o valor em  $x = b$ , é maior que a condição de contorno  $y(b) = Y_b$ . Na outra solução,  $y_{b,L}$ , o valor é menor. Os valores iniciais das inclinações em  $x = a$  que correspondem a essas soluções são  $W_H$  e  $W_L$  (lembre que  $W = \frac{dy}{dx} \Big|_{x=a}$ ). Os erros em



**Figura 9-7** Método da bisseção para determinar  $W_N$ .

cada solução numérica são  $E_H = y_{b,H} - Y_b$  (um número positivo) e  $E_L = y_{b,L} - Y_b$  (um número negativo). Com essa notação, o erro  $E$  pode ser considerado função da inclinação  $W$ , e o objetivo se torna determinar o valor de  $W$  em que  $E = 0$ .

Com o método da bisseção, ilustrado na Fig. 9-7, o valor inicial da inclinação  $W_N$  no passo de cálculo seguinte é:

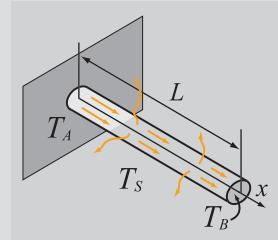
$$W_N = \frac{1}{2}(W_H + W_L) \quad (9.18)$$

Quando o método é implementado em um código de computador, os valores de  $y_{b,H}$ ,  $y_{b,L}$ ,  $W_H$  e  $W_L$  devem ser especificados. Depois, após cada iteração, o programa deve selecionar novos valores de forma que a condição de contorno esteja contida no intervalo formado por  $y_{b,H}$  e  $y_{b,L}$ . A implementação do método da bisseção é mostrada no Exemplo 9-2.

### Exemplo 9-2: Distribuição de temperaturas em um dissipador em forma de pino.

Solução de uma EDO de segunda ordem (PVC) usando o método do tiro juntamente com o método da bisseção

Escreva um programa no MATLAB que resolva o PVC do Exemplo 9-1 usando o método da bisseção e o método do tiro para determinar uma nova estimativa para a inclinação inicial  $w(0)$ . Comece resolvendo o problema duas vezes usando  $w(0) = -1000$  e  $w(0) = -3500$ , o que, conforme mostrado no Exemplo 9-1, resulta em soluções que são maiores e menores que a condição de contorno em  $x = 0,1$ , respectivamente, e que portanto podem servir de valores iniciais para o método da bisseção. Interrompa as iterações quando a diferença entre a temperatura calculada a partir da solução numérica e a condição de contorno prescrita for menor que 0,01 K.



### SOLUÇÃO

Lembre, do Exemplo 9-1, que o PVC de segunda ordem a ser resolvido é:

$$\frac{d^2T}{dx^2} - \frac{h_c P}{kA_c}(T - T_S) - \frac{\varepsilon\sigma_{SB}P}{kA_c}(T^4 - T_S^4) = 0, \quad 0 \leq x \leq 0,1 \quad (9.19)$$

com condições de contorno  $T(0) = 473$  e  $T(L) = 293$ .

A solução é obtida com a conversão da Eq. (9.19) no seguinte sistema de duas EDOs de primeira ordem.

$$\frac{dT}{dx} = w \quad (9.20)$$

$$\frac{dw}{dx} = \frac{h_c P}{kA_c}(T - T_S) + \frac{\varepsilon\sigma_{SB}P}{kA_c}(T^4 - T_S^4) \quad (9.21)$$

O valor inicial da Eq. (9.20) é  $T(0) = 473$ . A condição inicial da Eq. (9.21) é desconhecida.

O programa de MATLAB listado a seguir incorpora o método da bisseção ao método do tiro. O programa utiliza a função `Sis2EDOsRK2`, escrita na Seção 8.8.2, para resolver o sistema formado pelas Eqs. (9.20) e (9.21). A ordem das operações no programa é a seguinte:

- Resolve-se o sistema assumindo  $w(0) = -1000$ .
- Resolve-se o sistema assumindo  $w(0) = -3500$ .
- Usando as duas primeiras soluções, novos valores são calculados para  $w(0)$  usando o método da bisseção, Eq. (9.18).
- Traça-se um gráfico da solução e da solução numérica em  $x = 0,1$  onde o valor de contorno é prescrito.

**Programa 9-2 Arquivo texto. Aplicação do método do tiro juntamente com o método da bisseção.**

```
% Solução do Exemplo 9-2
```

```
clear all
```

```
a = 0; b = 0.1; TINI = 473; h = 0.001; Yb = 293;
```

```
tol = 0.01; imax = 15;
```

```
wH = -1000;
```

Resolve o sistema assumindo  $w(0) = -1000$ .

```
[x, T, w] = Sis2EDOsRK2('edoCap9Exmp1dTdx','edoCap9Exmp1dwdx',a,b,h,TINI,wH);
```

```
n = length(x);
```

```
wL = -3500;
```

Resolve o sistema assumindo  $w(0) = -3500$ .

```
[x, T, w] = Sis2EDOsRK2('edoCap9Exmp1dTdx','edoCap9Exmp1dwdx',a,b,h,TINI,wL);
```

```
for i = 1:imax + 1
```

Início das iterações.

```
    wi = (wH + wL)/2;
```

Calcula um novo valor para  $w(0)$  usando o método da bisseção, Eq. (9.18).

```
    [x, T, w] = Sis2EDOsRK2('edoCap9Exmp1dTdx','edoCap9Exmp1dwdx',a,b,h,TINI,wi);
```

```
    E = T(n) - Yb;
```

Calcula o erro entre a nova solução e a condição de contorno.

```
    if abs(E) < tol
```

```
        break
```

```
    end
```

```
    if E > 0
```

```
        wH = wi;
```

```
    else
```

```
        wL = wi;
```

```
    end
```

```
end
```

```
if i > imax
```

Final das iterações.

```
    fprintf('A solução não foi obtida em %i iterações.\n',imax)
```

```
else
```

```
    plot(x,T)
```

Atribui um novo valor a  $W_H$  ou a  $W_L$ .

```
    xlabel('Distância (m)'); ylabel('Temperatura (K)')
```

```
    fprintf('A temperatura calculada em x = 0.1 é %5.3f K.\n',T(n))
```

```
    fprintf('A solução foi obtida em %2.0f iterações.\n',i)
```

```
end
```

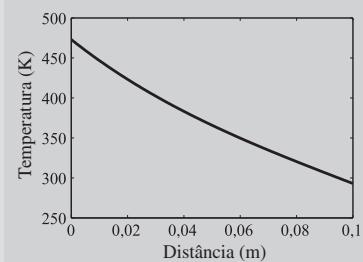
Os primeiros dois argumentos da função `Sis2EDOsRK2` são os nomes de funções em arquivo que calculam os valores de  $dT/dx$ , Eq. (9.20), e  $dw/dx$ , Eq. (9.21). As duas funções, `edoCap9Exmp1dTdx` e `edoCap9Exmp1wdx`, estão listadas no Exemplo 9-1.

Quando o programa é executado, a seguinte mensagem é exibida na janela de comandos.

A temperatura calculada em  $x = 0.1$  é 292.999 K.

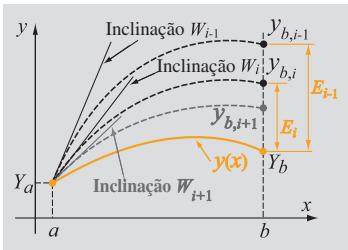
A solução foi obtida em 9 iterações.

Além disso, o programa produz uma figura com um gráfico da solução, mostrado ao lado.



**Método do tiro usando o método da secante:** O PVC associado à Eq. (9.8) é resolvido com a conversão do problema em um sistema de dois PVIs descrito

pelas Eqs. (9.9) e (9.10). Soluções numéricas obtidas em duas iterações,  $i$  e  $i - 1$ , são mostradas na Fig. 9-8. As soluções em  $x = b$  são  $y_{b,i-1}$  e  $y_{b,i}$ . Lembre que a condição de contorno neste ponto é  $y(b) = Y_b$ . Os valores iniciais das inclinações em  $x = a$  que correspondem a essa solução são  $W_{i-1}$  e  $W_i$ , respectivamente (lembre que  $W = \frac{dy}{dx} \Big|_{x=a}$ ). Os erros em cada solução numérica são  $E_{i-1} = y_{b,i-1} - Y_b$  e  $E_i = y_{b,i} - Y_b$ . Com essa notação, o erro  $E$  pode ser considerado função da inclinação  $W$ , e o objetivo fica sendo determinar o valor de  $W$  em que  $E = 0$ .



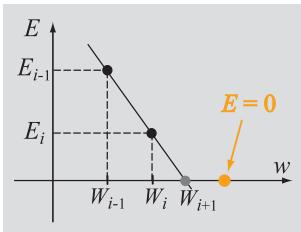
**Figura 9-8** Duas iterações em uma solução com o método do tiro.

uma estimativa para o zero de uma função a partir do valor dessa função em dois pontos próximos à solução (ver Seção 3.6). Quando usado em combinação com o método do tiro, a próxima estimativa para a inclinação  $W_{i+1}$  é calculada a partir dos pontos  $(W_i, E_i)$  e  $(W_{i-1}, E_{i-1})$ . Isso é ilustrado na Fig. 9-9. O novo valor da inclinação,  $W_{i+1}$ , é dado por (ver Eq. (3.26)):

$$W_{i+1} = W_i - \frac{(W_{i-1} - W_i)}{(E_{i-1} - E_i)} E_i \quad (9.22)$$

Neste método, os  $E$ s das duas iterações anteriores podem ser positivos, negativos ou ter sinais opostos.

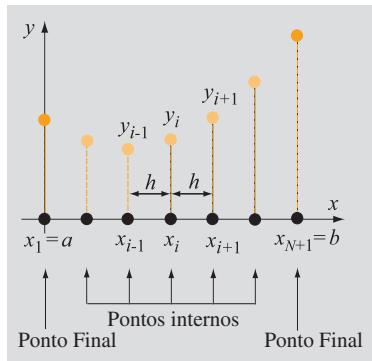
O método do tiro também pode ser usado para resolver PVCs com condições de contorno envolvendo somente derivadas ou condições de contorno mistas. A abordagem geral é a mesma usada em problemas de valor de contorno com dois pontos. A EDO de segunda ordem (ou de ordem superior) é convertida em um



**Figura 9-9** Método da secante para determinar  $W_{i+1}$ .

sistema de EDOs de primeira ordem. Em seguida, este é resolvido como um problema de valor inicial assumindo-se os valores iniciais necessários para a solução. A solução é então comparada com as condições de contorno prescritas e, se os resultados não forem suficientemente precisos, os valores iniciais assumidos são modificados e o sistema é resolvido novamente. Quando condições de contorno envolvendo derivadas forem prescritas no ponto final, o valor calculado da derivada deve ser avaliado numericamente.

### 9.3 O MÉTODO DAS DIFERENÇAS FINITAS



**Figura 9-10** Método das diferenças finitas.

a solução numérica da equação diferencial.

Em métodos de diferenças finitas, as derivadas presentes na equação diferencial são substituídas por aproximações usando diferenças finitas. Conforme mostrado na Fig. 9-10, o domínio da solução  $[a, b]$  é dividido em  $N$  subintervalos de mesma largura  $h$ , que são definidos por  $(N + 1)$  pontos chamados de **pontos de malha** (em geral, subintervalos podem ter larguras desiguais). A largura de cada subintervalo é portanto  $h = (b - a)/N$ . Os pontos  $a$  e  $b$  são os **pontos finais**, e o restante dos pontos são os **pontos internos**. A equação diferencial é então escrita em cada um dos pontos internos do domínio. Isso resulta em um sistema de equações lineares algébricas quando a equação diferencial for linear, ou em um sistema de equações não-lineares algébricas quando a equação diferencial for não-linear. A solução do sistema é

Muitas fórmulas de diferenças finitas são listadas no Capítulo 6 (ver Tabela 6-1). Freqüentemente, as fórmulas de diferença central são usadas nos métodos de diferenças finitas, pois elas fornecem uma melhor precisão. Lembre que, para uma função  $y(x)$  dada nos pontos  $(x_1, y_1), \dots, (x_i, y_i), \dots, (x_{N+1}, y_{N+1})$ , que são igualmente espaçados ( $h = x_{i+1} - x_i$  para  $i = 1 \dots N$ ), a aproximação das derivadas primeira e segunda nos pontos internos usando diferenças finitas é dada por:

$$\frac{dy}{dx} = \frac{y_{i+1} - y_{i-1}}{2h} \quad \text{e} \quad \frac{d^2y}{dx^2} = \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} \quad (9.23)$$

#### Solução de um PVC linear de segunda ordem usando o método de diferenças finitas

A aproximação por diferenças finitas de uma equação diferencial de segunda ordem na forma:

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} + g(x)y = h(x) \quad (9.24)$$

é:

$$\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2} + f(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + g(x_i)y_i = h(x_i) \quad (9.25)$$

O processo de conversão da equação diferencial dada na Eq. (9.24) para a forma algébrica mostrada na Eq. (9.25) em cada ponto desejado é chamado de **discretização**. Em um PVC com dois pontos, conhece-se o valor da solução nos pontos finais  $y_1$  e  $y_{N+1}$ . A Eq. (9.25) é escrita  $N - 1$  vezes para  $i = 2, \dots, N$ . Isso resulta em um sistema de  $N - 1$  equações lineares algébricas em função das incógnitas  $y_2, \dots, y_N$ , que podem ser resolvidas numericamente com qualquer um dos métodos numéricos prescritos no Capítulo 4.

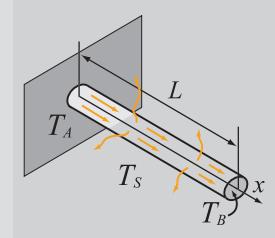
O Exemplo 9-3 mostra a solução de um PVC linear de segunda ordem usando o método de diferenças finitas. Neste exemplo, a distribuição de temperaturas em um dissipador em forma de pino é calculada para o caso em que apenas a convecção está incluída na análise.

### Exemplo 9-3: Distribuição de temperaturas em um dissipador em forma de pino.

#### Solução de uma EDO linear de segunda ordem (PVC) usando o método de diferenças finitas

Quando apenas a convecção está incluída na análise, a distribuição de temperaturas  $T(x)$  em regime permanente ao longo de um dissipador em forma de pino pode ser obtida a partir da solução da equação:

$$\frac{d^2 T}{dx^2} - \frac{h_c P}{k A_c} (T - T_s) = 0, \quad 0 \leq x \leq L \quad (9.26)$$



com condições de contorno:  $T(0) = T_A$  e  $T(L) = T_B$ .

Na Eq. (9.26),  $h_c$  é o coeficiente de transferência de calor por convecção,  $P$  é o perímetro da área da seção transversal do dissipador,  $k$  é a condutividade térmica do material do dissipador,  $A_c$  é a área da seção transversal do dissipador e  $T_s$  é a temperatura ambiente.

Determine a distribuição de temperaturas se  $L = 0,1$  m,  $T(0) = 473$  K,  $T(0,1) = 293$  K e  $T_s = 293$  K. Use os seguintes valores para os parâmetros da Eq. (9.15):  $h_c = 40$  W/m<sup>2</sup>/K,  $P = 0,016$  m<sup>2</sup>,  $k = 240$  W/m/K e  $A_c = 1,6 \times 10^{-5}$  m<sup>2</sup>.

Solucionar a EDO usando o método da diferença finita. Divida o domínio da solução em cinco subintervalos igualmente espaçados.

### SOLUÇÃO

A Eq. (9.26) é uma EDO linear de segunda ordem. Usando o método de diferenças finitas, a derivada segunda  $\frac{d^2 T}{dx^2}$  é aproximada pela fórmula de diferença central, Eq. (9.23):

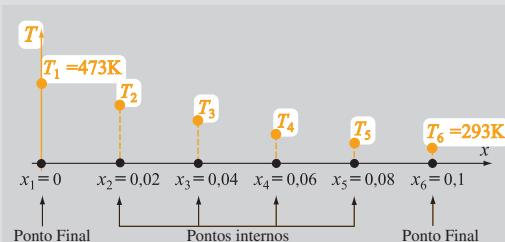
$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - \beta(T_i - T_s) = 0 \quad (9.27)$$

onde  $\beta = \frac{h_c P}{k A_c}$ . A Eq. (9.27) pode ser escrita como:

$$T_{i-1} - (2 + h^2 \beta) T_i + T_{i+1} = -h^2 \beta T_s \quad (9.28)$$

Em seguida, o domínio da solução é dividido em cinco subintervalos igualmente espaçados (definidos por seis pontos), conforme mostrado na figura.

Em seguida, a Eq. (9.28) é escrita em cada um dos pontos internos (isto é,  $i = 2, 3, 4, 5$ ):



$$\text{para } i = 2 \quad T_1 - (2 + h^2\beta)T_2 + T_3 = -h^2\beta T_S \text{ ou } -(2 + h^2\beta)T_2 + T_3 = -(h^2\beta T_S + T_1) \quad (9.29)$$

Lembre que  $T_1$  é conhecido.

$$\text{para } i = 3 \quad T_2 - (2 + h^2\beta)T_3 + T_4 = -h^2\beta T_S \quad (9.30)$$

$$\text{para } i = 4 \quad T_3 - (2 + h^2\beta)T_4 + T_5 = -h^2\beta T_S \quad (9.31)$$

$$\text{para } i = 5 \quad T_4 - (2 + h^2\beta)T_5 + T_6 = -h^2\beta T_S \text{ ou } T_4 - (2 + h^2\beta)T_5 = -(h^2\beta T_S + T_6) \quad (9.32)$$

Lembre que  $T_6$  é conhecido da condição de contorno  $T(0,1) = 293$  K.

As Eqs. (9.29)-(9.32) formam um sistema de quatro equações lineares algébricas com incógnitas  $T_2$ ,  $T_3$ ,  $T_4$  e  $T_5$ . Na forma matricial,  $[a][T] = [c]$ , o sistema pode ser escrito como:

$$\begin{bmatrix} -(2 + h^2\beta) & 1 & 0 & 0 \\ 1 & -(2 + h^2\beta) & 1 & 0 \\ 0 & 1 & -(2 + h^2\beta) & 1 \\ 0 & 0 & 1 & -(2 + h^2\beta) \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \begin{bmatrix} -(h^2\beta T_S + T_1) \\ -h^2\beta T_S \\ -h^2\beta T_S \\ -(h^2\beta T_S + T_6) \end{bmatrix} \quad (9.33)$$

O sistema de equações na Eq. (9.33) pode ser resolvido com qualquer um dos métodos descritos no Capítulo 4. A resposta é a solução da EDO dada na Eq. (9.26) nos pontos internos.

O seguinte programa de MATLAB resolve a Eq. (9.33) e traça um gráfico com os resultados.

**Programa 9-3 Arquivo texto. Solução de uma EDO linear de segunda ordem usando o método de diferenças finitas.**

```
% Solução do Exemplo 9-3
```

```
clear all
```

```
hc = 40; P = 0.016; k = 240; Ac = 1.6E-5;
```

```
h = 0.02; Ts = 293;
```

```
x = 0:0.02:0.1;
```

Cria um vetor para a coordenada  $x$  dos pontos de malha.

```
beta = hc*P/(k*Ac);
```

```
aDia = -(2 + h^2*beta);
```

```
cele = -h^2*beta*Ts;
```

```
T(1) = 473; T(6) = 293;
```

```
a = eye(4,4)*aDia;
```

```
for i = 1:3
```

```
    a(i,i + 1) = 1;
```

```
    a(i + 1,i) = 1;
```

```
end
```

```
c = [cele - T(1); cele; cele - T(6)];
```

Monta a matriz  $[a]$ .

```
T(2:5) = a\c;
```

Define o vetor  $[c]$ .

```
fprintf('As temperaturas nos pontos de malha são\n')
```

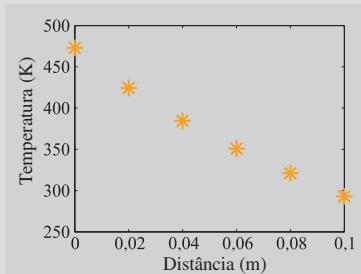
Resolve o sistema da Eq. (9.33).

```
disp(T)
```

```
plot(x,T,'*r')
```

```
xlabel('Distância (m)'); ylabel('Temperatura (K)')
```

Quando o programa é executado, mostra-se a figura ao lado e os seguintes resultados são exibidos na janela de comandos do MATLAB.



As temperaturas nos pontos de malha são  
473.0000 424.4117 384.5842 350.8623 320.9979 293.0000

### Nota adicional

A aplicação do método de diferenças finitas na solução de uma EDO nem sempre resulta em um sistema tridiagonal de equações conforme ilustrado acima. A solução numérica no Exemplo 9-3 leva a um sistema tridiagonal de equações porque a EDO é de segunda ordem e porque um esquema de diferenças centrais foi usado para aproximar a derivada segunda.

### Solução de um PVC não-linear de segunda ordem usando o método de diferenças finitas

A abordagem de solução de uma EDO não-linear usando o método de diferenças finitas é a mesma usada na solução de uma EDO linear. A única diferença está no fato de o sistema de equações resultante ser não-linear. Métodos para resolver tais sistemas são descritos no Capítulo 3. A tarefa é bem mais desafiadora do que a solução de um sistema de equações lineares. A maneira computacionalmente mais eficiente para resolver um sistema de equações não-lineares é a aplicação de algum tipo de esquema iterativo. Entretanto, conforme discutido no Capítulo 3, métodos iterativos correm o risco de divergir, a menos que os valores iniciais ou de partida utilizados nas iterações sejam suficientemente próximos da resposta final.

A aplicação do método de diferenças finitas em uma EDO não-linear resulta em um sistema de equações não-lineares. Um dos métodos que podem ser usados na solução de tal sistema é uma variação do método da iteração de ponto fixo descrito na Seção 3.2.10. Se  $[y]$  é um vetor coluna com as incógnitas, o sistema de equações não-lineares pode ser escrito na forma:

$$[a][y] + [\Phi] = [b] \quad (9.34)$$

onde  $[a]$  é uma matriz de coeficientes,  $[\Phi]$  é um vetor coluna cujos elementos são funções não-lineares das incógnitas  $y_i$  e  $[b]$  é um vetor coluna com as grandezas conhecidas.

Há muitas maneiras de se desenvolver funções iterativas para realizar o procedimento da iteração de ponto fixo. Uma forma evidente a partir da Eq. (9.34) é:

$$[a][y]^{k+1} = [b] - [\Phi]^k \quad (9.35)$$

onde  $[y]^{k+1}$  é um vetor de incógnitas e  $[\Phi]^k$  é um vetor conhecido que usa os valores da solução  $[y]^k$  obtidos na iteração anterior. A Eq. (9.35) pode agora ser resolvida para  $[y]^{k+1}$ . Note que, se o número de pontos internos for pequeno,  $[a]$  pode ser invertida. Com isso:

$$[y]^{k+1} = [a]^{-1}([b] - [\Phi]^k) \quad (9.36)$$

Essa abordagem é ilustrada no Exemplo 9-4. No caso de um grande número de pontos internos, a Eq. (9.35) pode ser resolvida com a aplicação do método de eliminação de Gauss ou, se  $[a]$  for uma matriz tridiagonal, pelo algoritmo de Thomas.

#### Exemplo 9-4 Distribuição de temperaturas em um dissipador em forma de pino. Solução de uma EDO não-linear de segunda ordem (PVC) usando o método de diferenças finitas

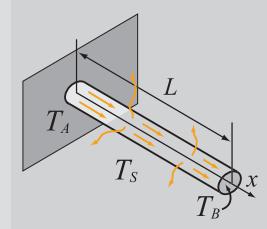
Quando a convecção e a radiação são incluídas na análise, a temperatura  $T(x)$  em regime permanente ao longo de um dissipador em forma de pino pode ser calculada a partir da solução da equação:

$$\frac{d^2T}{dx^2} - \frac{h_c P}{kA_c}(T - T_S) - \frac{\epsilon \sigma_{SB} P}{kA_c}(T^4 - T_S^4) = 0, \quad 0 \leq x \leq L \quad (9.37)$$

com as condições de contorno:  $T(0) = T_A$  e  $T(L) = T_B$ .

A definição e os valores de todas as constantes na Eq. (9.37) são dados no Exemplo 9-2.

Determine a distribuição de temperaturas se  $L = 0,1$  m,  $T(0) = 473$  K,  $T(0,1) = 293$  K e  $T_S = 293$  K. Resolva a EDO usando o método de diferenças finitas. Divida o domínio da solução em cinco subintervalos igualmente espaçados.



#### SOLUÇÃO

A Eq. (9.37) é uma EDO não-linear de segunda ordem. Usando o método de diferenças finitas, a derivada segunda  $\frac{d^2T}{dx^2}$  pode ser aproximada pela fórmula de diferença central dada na Eq. (9.23):

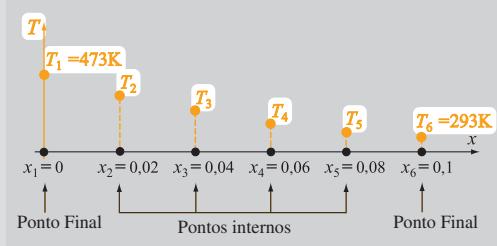
$$\frac{T_{i-1} - 2T_i + T_{i+1}}{h^2} - \beta_A(T_i - T_S) - \beta_B(T_i^4 - T_S^4) = 0 \quad (9.38)$$

onde  $\beta_A = \frac{h_c P}{kA_c}$  e  $\beta_B = \frac{\epsilon \sigma_{SB} P}{kA_c}$

A Eq. (9.38) pode ser escrita como

$$T_{i-1} - (2 + h^2 \beta_A)T_i - h^2 q \beta_B T_i^4 + T_{i+1} = -h^2 (\beta_A T_S + \beta_B T_S^4) \quad (9.39)$$

Depois, o domínio da solução é dividido em cinco subintervalos igualmente espaçados, definidos por seis pontos, conforme mostrado na figura.



Em seguida, a Eq. (9.39) é escrita em cada um dos pontos internos (isto é,  $i = 2, 3, 4, 5$ ):

$$\begin{aligned} \text{para } i = 2 \quad T_1 - (2 + h^2 \beta_A)T_2 - h^2 \beta_B T_2^4 + T_3 &= -h^2(\beta_A T_S + \beta_B T_S^4) \\ \text{ou } -(2 + h^2 \beta_A)T_2 - h^2 \beta_B T_2^4 + T_3 &= -h^2(\beta_A T_S + \beta_B T_S^4) - T_1 \end{aligned} \quad (9.40)$$

Lembre que  $T_1$  é conhecido.

$$\text{para } i = 3 \quad T_2 - (2 + h^2 \beta_A)T_3 - h^2 \beta_B T_3^4 + T_4 = -h^2(\beta_A T_S + \beta_B T_S^4) \quad (9.41)$$

$$\text{para } i = 4 \quad T_3 - (2 + h^2 \beta_A)T_4 - h^2 \beta_B T_4^4 + T_5 = -h^2(\beta_A T_S + \beta_B T_S^4) \quad (9.42)$$

$$\begin{aligned} \text{para } i = 5 \quad T_4 - (2 + h^2 \beta_A)T_5 - h^2 \beta_B T_5^4 + T_6 &= -h^2(\beta_A T_S + \beta_B T_S^4) \\ \text{ou } T_4 - (2 + h^2 \beta_A)T_5 - h^2 \beta_B T_5^4 &= -h^2(\beta_A T_S + \beta_B T_S^4) - T_6 \end{aligned} \quad (9.43)$$

Lembre que  $T_6$  é conhecido da condição de contorno  $T(0,1) = 293$  K.

As Eqs. (9.40)-(9.43) formam um sistema de quatro equações lineares algébricas com incógnitas  $T_2$ ,  $T_3$ ,  $T_4$  e  $T_5$ . Seguindo a Eq. (9.34), o sistema pode ser escrito na forma  $[a][T] + [\Phi] = [b]$ :

$$\left[ \begin{array}{cccc} -(2 + h^2 \beta_A) & 1 & 0 & 0 \\ 1 & -(2 + h^2 \beta_A) & 1 & 0 \\ 0 & 1 & -(2 + h^2 \beta_A) & 1 \\ 0 & 0 & 1 & -(2 + h^2 \beta_A) \end{array} \right] \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} + \begin{bmatrix} -h^2 \beta_B T_2^4 \\ -h^2 \beta_B T_3^4 \\ -h^2 \beta_B T_4^4 \\ -h^2 \beta_B T_5^4 \end{bmatrix} = \begin{bmatrix} -h^2(\beta_A T_S + \beta_B T_S^4) - T_1 \\ -h^2(\beta_A T_S + \beta_B T_S^4) \\ -h^2(\beta_A T_S + \beta_B T_S^4) \\ -h^2(\beta_A T_S + \beta_B T_S^4) - T_6 \end{bmatrix} \quad (9.44)$$

O sistema de equações apresentado na Eq. (9.44) pode ser resolvido usando o método da iteração de ponto fixo (ver Capítulo 3). Uma fórmula iterativa é obtida com a solução da Eq. (9.44) para  $[T]$ :

$$[T]^{k+1} = [a]^{-1}([b] - [\Phi]^k) \quad (9.45)$$

A solução iterativa começa com a escolha dos valores de  $T_2$ ,  $T_3$ ,  $T_4$  e  $T_5$ , depois com o cálculo do vetor  $[b]$  e o uso da Eq. (9.45) para calcular os novos valores de  $[T]^{k+1}$ . Os novos valores são substituídos de volta na Eq. (9.45) e assim por diante. Embora várias fórmulas iterativas possam ser escritas a partir da Eq. (9.44), nem todas convergem. As equações na Eq. (9.44) correspondem ao caso  $N = 5$ . O sistema de equações, no entanto, é tridiagonal e pode ser facilmente estendido se o domínio for dividido em um número maior de subintervalos.

O sistema pode ser resolvido com qualquer um dos métodos de solução de sistemas de equações não-lineares apresentados no Capítulo 3. Lista-se a seguir um programa escrito no MATLAB para resolver este problema numericamente. O programa foi escrito em termos do número de subintervalos  $N$ . Ele usa o método de iteração de ponto fixo e executa quatro iterações. O programa mostra a temperatura nos pontos internos após cada iteração e traça um gráfico com os resultados da última iteração.

**Programa 9-4 Arquivo texto. Solução de uma EDO não-linear de segunda ordem usando o método de diferenças finitas.**

```
% Solução do Exemplo 9-4
```

```
clear all
```

```
hc = 40; P = 0.016; k = 240; Ac = 1.6E-5; epsln = 0.4; seg = 5.67E-8;
```

```
betaA = hc*P/(k*Ac); betaB = epsln*seg*P/(k*Ac); Ts = 293;
```

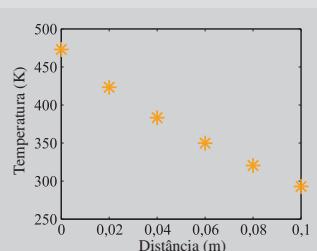
```

N = 5; h = 0.1/N;
x = 0:h:0.1; Define um vetor para as coordenadas x dos pontos de malha.
aDia = -(2 + h^2*betaA);
bele = -h^2*(betaA *Ts + betaB*Ts^4);
h2betaB = h^2*betaB;
Ti(1) = 473; Ti(N + 1) = 293;
Tnext(1) = Ti(1); Tnext(N + 1) = Ti(N + 1);
a = eye(N - 1,N - 1)*aDia;
for i = 1:N - 2
    a(i,i + 1) = 1;
    a(i + 1,i) = 1;
end
ainv = inv(a);
b(1) = bele - Ti(1); b(N-1) = bele - Ti(N+1);
c(2:N - 2) = bele; Define o vetor [c].
Ti(2:N) = 400; A tentativa inicial para os pontos internos é 400 K.
for i = 1:4 Resolve o sistema na Eq. (9.44).
    phi = -h2betaB*Ti(2:N).^4;
    Tnext(2:N) = ainv*(b' - phi);
    Ti = Tnext;
    fprintf('Após a iteração número%2.0f, as temperaturas nos pontos de malha são:\n',i )
    disp(Tnext)
end
plot(x,Tnext,'*r')
xlabel('Distância (m)'); ylabel('Temperatura (K)')

```

Quando o programa é executado, produz-se a figura à direita e os seguintes resultados são mostrados na janela de comandos do MATLAB.

Em seguida, o programa foi executado com  $N = 20$ . Os resultados dessa execução são mostrados na figura seguinte.

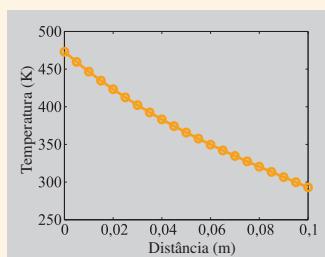


Após a iteração número 1, as temperaturas nos pontos de malha são:  
473.0000 423.2293 382.8297 349.1078 319.8155 293.0000

Após a iteração número 2, as temperaturas nos pontos de malha são:  
473.0000 423.3492 383.3225 349.8507 320.4519 293.0000

Após a iteração número 3, as temperaturas nos pontos de malha são:  
473.0000 423.3440 383.3132 349.8409 320.4456 293.0000

Após a iteração número 4, as temperaturas nos pontos de malha são:  
473.0000 423.3441 383.3134 349.8410 320.4457 293.0000



Os resultados concordam com a solução obtida usando o método do tiro no Exemplo 9-2.

### Notas adicionais

- (1) Neste exemplo, a solução iterativa de um sistema não-linear foi facilmente realizada com o emprego da função residente do MATLAB dedicada ao cálculo da inversa de uma matriz (isto foi feito por conveniência porque, neste problema, a matriz é pequena). Conforme mencionado no Capítulo 4, a inversão de uma matriz não é uma boa idéia, em geral, por ser computacionalmente ineficiente. É melhor resolver o sistema da Eq. (9.44) usando a eliminação de Gauss ou o algoritmo de Thomas.
- (2) Se for utilizada uma diferente fórmula iterativa (por exemplo,  $[y]_{i+1} = [y]_i - \{[a][y]_i + [\Phi]_i - [b]\}$ ), as iterações não convergirão a menos que as tentativas iniciais estejam próximas da solução verdadeira.

### Solução de um PVC linear de segunda ordem com condições de contorno mistas usando o método de diferenças finitas

O método de diferenças finitas também pode ser aplicado em PVCs com condições de contorno mistas. Este caso envolve a prescrição de uma restrição envolvendo uma derivada em um ou nos dois pontos finais do domínio da solução. Nesses problemas, o método de diferenças finitas é usado para discretizar a EDO nos pontos internos (como nos PVCs de dois pontos). Entretanto, o sistema de equações algébricas obtido não pode ser resolvido pelo fato de não se conhecer a solução nos pontos finais (há mais equações que incógnitas). As equações adicionais necessárias para a solução do problema são obtidas com a discretização das condições de contorno usando diferenças finitas e com a incorporação das equações resultantes nas equações algébricas escritas nos pontos internos. A solução de um PVC de segunda ordem com condições de contorno mistas é ilustrada no Exemplo 9-5.

### Nota final

A discussão e os exemplos anteriores mostram que não existe uma vantagem clara entre a escolha do método de diferenças finitas ou do método do tiro para resolver uma EDO não-linear de ordem mais elevada. O método de diferenças finitas requer a solução de um sistema de equações não-lineares, enquanto o método do tiro requer informações a respeito das derivadas de ordem mais elevada da variável dependente no contorno da esquerda. A escolha de que método usar depende portanto do problema analisado, e de quão facilmente tentativas iniciais podem ser feitas para as derivadas da variável dependente no contorno (no caso do método do tiro) ou de quão boa for a convergência do esquema de iteração de ponto fixo utilizado (no caso do método de diferenças finitas).

### Exemplo 9-5: Solução de um PVC com condições de contorno mistas

Use o método de diferenças finitas para resolver o seguinte PVC com condições de contorno mistas:

$$-2\frac{d^2y}{dx^2} + y = e^{-0.2x}, \quad \text{para } 0 \leq x \leq 1 \quad (9.46)$$

com as condições iniciais:  $y(0) = 1$  e  $\left.\frac{dy}{dx}\right|_{x=1} = -y$ .

Divida o domínio da solução em oito subintervalos e use a aproximação de diferença central para todas as derivadas. Compare a solução numérica com a solução exata:

$$y = -0,2108e^{x/(\sqrt{2})} + 0,1238e^{-x/(\sqrt{2})} + \frac{e^{-0.2x}}{0,92} \quad (9.47)$$

## SOLUÇÃO

Para usar o método de diferenças finitas, a derivada segunda  $\frac{d^2y}{dx^2}$  na Eq. (9.46) é aproximada pela fórmula de diferença central com três pontos, a Eq. (9.23):

$$-2\left(\frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}\right) + y_i = e^{-0.2x_i} \quad (9.48)$$

Combinando os termos comuns e multiplicando tudo por  $h^2$ , a Eq. (9.48) pode ser escrita como:

$$-2y_{i-1} + (4 + h^2)y_i - 2y_{i+1} = h^2 e^{-0.2x_i} \quad (9.49)$$

Em seguida, a Eq. (9.49) é escrita em cada um dos pontos internos (isto é,  $i = 2, 3, 4, 5, 6, 7, 8$ ):

$$\begin{aligned} \text{para } i = 2 & \quad -2y_1 + (4 + h^2)y_2 - 2y_3 = h^2 e^{-0.2x_2} \\ \text{ou } (4 + h^2)y_2 - 2y_3 & = 2 + h^2 e^{-0.2x_2} \end{aligned} \quad (9.50)$$

já que  $y_1 = y(0) = 1$ .

$$\text{para } i = 3: \quad -2y_2 + (4 + h^2)y_3 - 2y_4 = h^2 e^{-0.2x_3} \quad (9.51)$$

$$\text{para } i = 4: \quad -2y_3 + (4 + h^2)y_4 - 2y_5 = h^2 e^{-0.2x_4} \quad (9.52)$$

$$\text{para } i = 5: \quad -2y_4 + (4 + h^2)y_5 - 2y_6 = h^2 e^{-0.2x_5} \quad (9.53)$$

$$\text{para } i = 6: \quad -2y_5 + (4 + h^2)y_6 - 2y_7 = h^2 e^{-0.2x_6} \quad (9.54)$$

$$\text{para } i = 7: \quad -2y_6 + (4 + h^2)y_7 - 2y_8 = h^2 e^{-0.2x_7} \quad (9.55)$$

$$\text{para } i = 8: \quad -2y_7 + (4 + h^2)y_8 - 2y_9 = h^2 e^{-0.2x_8} \quad (9.56)$$

As Eqs. (9.50) a (9.56) formam um sistema de sete equações lineares com oito incógnitas,  $y_2, y_3, y_4, y_5, y_6, y_7, y_8$  e  $y_9$ . Neste problema, diferentemente de um PVC com dois pontos, o valor da solução no ponto final da direita (isto é,  $y_9$ ) não é conhecido.

Uma equação adicional pode ser deduzida a partir da discretização da condição de contorno em  $x = 1$ :

$$\left. \frac{dy}{dx} \right|_{x=1} = -y \quad (9.57)$$

A derivada presente na Eq. (9.57) é aproximada por uma fórmula de diferenças finitas. Conforme descrito no Capítulo 6, muitas fórmulas podem ser usadas com esse propósito. Como a condição da derivada é fornecida no último ponto, faz sentido usar uma fórmula unilateral regressiva que utilize os valores nos pontos anteriores. Além disso, como a derivada segunda na Eq. (9.46) é aproximada por uma fórmula com precisão de segunda ordem, faz sentido usar uma fórmula de diferença regressiva com três pontos para aproximar a derivada primeira, já que esta também é uma fórmula com precisão de segunda ordem. A forma geral da fórmula de diferença regressiva com três pontos é:

$$\frac{dy}{dx} = \frac{y_{i-2} - 4y_{i-1} + 3y_i}{2h} \quad (9.58)$$

Usando a Eq. (9.58) para aproximar a derivada no último ponto  $(x_9, y_9)$  juntamente com a condição de contorno em  $x = 1$ , obtém-se:

$$\frac{y_7 - 4y_8 + 3y_9}{2h} = -y_9 \quad (9.59)$$

Isolando  $y_9$  na Eq. (9.59):

$$y_9 = \frac{-1}{3+2h} y_7 + \frac{4}{3+2h} y_8 \quad (9.60)$$

A Eq. (9.60) fornece a relação adicional necessária para a solução de todas as incógnitas. Substituindo a Eq. (9.60) na Eq. (9.56):

$$\left( \frac{2}{3+2h} - 2 \right) y_7 + \left( 4 + h^2 - \frac{8}{3+2h} \right) y_8 = h^2 e^{-0.2x_8} \quad (9.61)$$

A Eq. (9.61) e as Eqs. (9.50)-(9.55) formam um sistema de sete equações lineares com sete incógnitas. Na forma matricial,  $[a][y] = [c]$ , o sistema pode ser expresso por:

$$\begin{bmatrix} (4+h^2) & -2 & 0 & 0 & 0 & 0 & 0 \\ -2 & (4+h^2) & -2 & 0 & 0 & 0 & 0 \\ 0 & -2 & (4+h^2) & -2 & 0 & 0 & 0 \\ 0 & 0 & -2 & (4+h^2) & -2 & 0 & 0 \\ & & -2 & (4+h^2) & -2 & 0 & 0 \\ & & & -2 & (4+h^2) & -2 & 0 \\ & & & & \left( \frac{2}{3+2h} - 2 \right) & \left( 4 + h^2 - \frac{8}{3+2h} \right) & \end{bmatrix} \begin{bmatrix} y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 2 + h^2 e^{-0.2x_2} \\ h^2 e^{-0.2x_3} \\ h^2 e^{-0.2x_4} \\ h^2 e^{-0.2x_5} \\ h^2 e^{-0.2x_6} \\ h^2 e^{-0.2x_7} \\ h^2 e^{-0.2x_8} \end{bmatrix} \quad (9.62)$$

Uma vez resolvida a Eq. (9.62), o valor de  $y_9$  pode ser calculado com a Eq. (9.60). Note que o sistema de equações na Eq. (9.62) é tridiagonal. Com isso, ele pode ser facilmente estendido ao caso em que o domínio é dividido em um número maior de subintervalos. O sistema pode ser resolvido com qualquer um dos métodos de solução de equações lineares apresentados no Capítulo 4, em especial o algoritmo de Thomas.

O seguinte programa de MATLAB apresenta a solução. O programa usa a função Tridiagonal, que foi escrita no Exemplo 4-9 para resolver um sistema tridiagonal de equações. O programa foi escrito em termos do número de subintervalos  $N$ , e com isso ele pode ser facilmente executado usando diferentes valores desse parâmetro.

**Programa 9-5 Arquivo texto. Solução de um PVC com condições de contorno mistas usando o método de diferenças finitas.**

```
% Solução do Exemplo 9-5
```

```
clear all
```

```
a = 0; b = 1;
```

Pontos finais do domínio da solução.

```
N = 8; h = (b - a)/N;
```

Define um vetor para as coordenadas  $x$  dos pontos internos.

```
x = a:h:b
```

```
hDenom = 3 + 2*h;
```

```
aDia = (4 + h^2);
```

```

y(1) = 1;
a = eye(N - 2,N - 2)*aDia;
a(N - 1,N - 1) = aDia - 8/hDenom;
for i = 1:N - 2
    a(i,i + 1) = -2;
    a(i + 1,i) = -2;
end
a(N - 1,N - 2) = 2/hDenom - 2;
c(1) = 2 + h^2*exp(-0.2*x(2));
c(2:N - 1) = h^2*exp(-0.2*x(3:N));
y(2:N) = Tridiagonal(a,c);
y(N + 1) = -1*y(N - 1)/hDenom + 4*y(N)/hDenom
yExato= -0.2108.*exp(x./sqrt(2)) + 0.1238.*exp(-x./sqrt(2)) + exp(-0.2.*x)./0.92

```

Monta a matriz [a].

Cria o vetor [c].

Resolve o sistema da Eq. (9.62).

Solução no último ponto usando a Eq. (9.60).

Quando o programa é executado com  $N = 8$ , a solução mostrada na janela de comandos do MATLAB é:

```

x =
    0   0.1250   0.2500   0.3750   0.5000   0.6250   0.7500   0.8750   1.0000
y =
    1.0000   0.9432   0.8861   0.8284   0.7701   0.7106   0.6499   0.5874   0.5230
yExato =
    1.0000   0.9432   0.8861   0.8286   0.7702   0.7109   0.6501   0.5878   0.5234

```

Os resultados mostram uma boa concordância entre as soluções numérica e exata.

## 9.4 USO DE FUNÇÕES RESIDENTES DO MATLAB PARA RESOLVER PROBLEMAS DE VALOR DE CONTORNO

Problemas de valor de contorno envolvem EDOs de segunda ordem ou de ordem superior com condições de contorno especificadas em ambos os pontos finais do intervalo. Conforme mostrado na Seção 8.9, EDOs de segunda ordem ou de ordem superior podem ser transformadas em sistemas de EDOs de primeira ordem. O MATLAB resolve problemas de valor de contorno por meio da solução de um sistema de EDOs de primeira ordem. A solução é obtida com o uso de uma função residente chamada `bvp4c`. Essa função usa um método de diferenças finitas que aplica a fórmula de Lobato de três estágios<sup>1</sup>. A função `bvp4c` tem formas diferentes, mas, para simplificar a sua apresentação, ela é introduzida aqui em sua forma mais simples e aplicada na solução de um PVC com dois pontos.

Lembre que uma EDO de segunda ordem tendo  $x$  como variável independente e  $y$  como variável dependente tem a forma:

$$\frac{d^2y}{dx^2} = f\left(x, y, \frac{dy}{dx}\right) \text{ para } a \leq x \leq b \text{ com } y(a) = Y_a \quad \text{e} \quad y(b) = Y_b \quad (9.63)$$

<sup>1</sup> A fórmula de Lobato é uma fórmula de quadratura que, diferentemente da quadratura de Gauss (ver Capítulo 7), usa, para realizar a integração exata de polinômios de grau  $2n - 1$ , os dois pontos finais do intervalo,  $(n - 1)$  pontos internos e pesos. A localização desses  $(n - 1)$  pontos internos são as raízes de um conjunto de polinômios ortogonais conhecidos como polinômios de Jacobi. Detalhes sobre esse método estão fora do escopo deste livro.

Transformando a EDO de segunda ordem em um sistema de duas EDOs de primeira ordem, obtém-se:

$$\frac{dy}{dx} = w \quad \text{e} \quad \frac{dw}{dx} = f(x, y, w) \quad (9.64)$$

onde  $w$  é uma variável dependente adicional (nova).

A forma mais simples da função `bvp4c` do MATLAB usada na solução de EDOs de primeira ordem é:

`sol = bvp4c( 'edofun' , 'ccfun' , solinit)`

onde:

`edofun`

Nome da função em arquivo que calcula  $\frac{dy}{dx} = w$  e  $\frac{dw}{dx} = f(x, y, w)$  para valores de  $x$ ,  $y$  e  $w$  fornecidos; `edofun` pode ser digitada como uma *string* (isto é, '`edofun`') ou com o uso de um caractere especial (isto é, `@edofun`).

O formato da função `edofun`, que deve ser criada pelo usuário, é:

`dydx = edofun (x, yw)`

O argumento de entrada `x` é um escalar, e o argumento de entrada `yw` é um vetor coluna com os valores das variáveis dependentes,  $\begin{bmatrix} y \\ w \end{bmatrix}$ . O argumento de saída `dydx` é um vetor coluna com os valores:  $\begin{bmatrix} w \\ f(x, y, w) \end{bmatrix}$ .

`ccfun`

Nome da função em arquivo criada pelo usuário para computar o resíduo na condição de contorno. O resíduo é a diferença entre a solução numérica e as condições de contorno prescritas (nos contornos da solução). `ccfun` pode ser digitada como uma *string* (isto é, '`ccfun`') ou com o uso de um caractere especial (isto é, `@ccfun`).

O formato da função criada pelo usuário `ccfun` é:

`res = ccfun (ya, yb)`

Os argumentos de entrada `ya` e `yb` são vetores coluna correspondentes à solução numérica em  $x = a$  e  $x = b$ . Os primeiros elementos, `ya(1)` e `yb(1)`, são os valores de  $y$  em  $x = a$  e  $x = b$ , respectivamente. Os segundos elementos `ya(2)` e `yb(2)` são os valores de  $dy/dx$  em  $x = a$  e  $x = b$ , respectivamente. O argumento de saída `res` é um vetor coluna com os valores dos resíduos. A função `ccfun` pode ser usada com quaisquer condições de contorno. Por exemplo, para condições de contorno de Dirichlet (ver Eq. (9.3)), onde

$Y_a$  e  $Y_b$  são as condições de contorno prescritas, o vetor coluna `res` é:  $\begin{bmatrix} ya(1) - Y_a \\ yb(1) - Y_b \end{bmatrix}$ . Para condições de contorno de Neumann (ver Eq. (9.4)), onde as derivadas  $D_a$  e  $D_b$  são as condições de contorno prescritas, o vetor coluna `res` é:  $\begin{bmatrix} ya(2) - D_a \\ yb(2) - D_b \end{bmatrix}$ . Condições de contorno mistas podem ter várias formas (ver Eq. (9.5)). As expressões possíveis para o vetor coluna `res` estão listadas na Fig. 9-11.

`solinit`

Condição de contorno:

$$y(a) = Y_a \quad \text{e} \quad \left. \frac{dy}{dx} \right|_{x=b} = D_b$$

o vetor `res` é:  $\begin{bmatrix} ya(1) - Y_a \\ yb(2) - D_b \end{bmatrix}$

Condição de contorno:

$$\left. \frac{dy}{dx} \right|_{x=a} = D_a \quad \text{e} \quad y(b) = Y_b$$

o vetor `res` é:  $\begin{bmatrix} ya(2) - D_a \\ yb(1) - Y_b \end{bmatrix}$

Condição de contorno (caso geral):

$$c_1 \left. \frac{dy}{dx} \right|_{x=a} + c_2 y(a) = C_a \quad \text{e}$$

$$c_3 \left. \frac{dy}{dx} \right|_{x=b} + c_4 y(b) = C_b$$

o vetor `res` é (para  $c_1, c_3 \neq 0$ ):

$$\begin{bmatrix} ya(2) - \frac{C_a}{c_1} + \frac{c_2}{c_1} ya(1) \\ yb(2) - \frac{C_b}{c_3} + \frac{c_4}{c_3} yb(1) \end{bmatrix}$$

**Figura 9-11** Resíduos para condições de contorno mistas.

`sol`

Uma estrutura contendo a tentativa inicial para a solução. `solinit` é criada por uma função residente do MATLAB chamada `bvpinit` (`solinit` é o argumento de saída de `bvpinit`).

O formato da função residente `bvpinit` é:

`solinit = bvpinit(x,yini)`

O argumento de entrada `x` é um vetor que especifica os pontos internos. Para um PVC com domínio  $[a, b]$ , o primeiro elemento de `x` é  $a$ , e o último elemento é  $b$ . Freqüentemente, um número inicial de dez pontos é adequado e pode ser criado digitando-se `x=linspace(a,b,10)`. O argumento de entrada `yini` é um vetor predefinido que tem um elemento para cada uma das variáveis dependentes associadas ao sistema de EDOs de primeira ordem a ser resolvido. No caso de duas equações, como na Eq. (9.64), o vetor `yini` tem dois elementos. O primeiro elemento é a tentativa inicial para o valor de  $y$  e o segundo elemento é a tentativa inicial para o valor de  $w$ . O MATLAB usa essas tentativas iniciais em todos os pontos internos. `yini` também pode ser fornecido como o nome de uma função em arquivo criada pelo usuário (isto é, digitada como uma variável *string* '`yini`' ou '@`yini`'). Nesse caso, a função tem a forma `y=tentativa(x)`, onde `x` é um ponto interno e `y` é o vetor contendo a tentativa inicial para a solução, conforme já explicado.

Uma estrutura contendo a solução. Três campos importantes em `sol` são:

`sol.x` Coordenada  $x$  dos pontos internos. O número de pontos internos é determinado pelo MATLAB durante o processo de solução. Ele não é, em geral, o mesmo número fornecido pelo usuário em `bvpinit`.

`sol.y` Solução numérica  $y(x)$ , ou seja, o valor de  $y$  nos pontos internos.

`sol.yp` Valor da derivada  $dy/dx$  nos pontos internos.

O uso da função residente `bvp4c` na solução de um PVC com dois pontos no MATLAB é ilustrado em detalhes no Exemplo 9-6 a seguir.

### Exemplo 9-6: Solução de um PVC com dois pontos no MATLAB usando a função residente `bvp4c`

Use a função residente `bvp4c` para resolver o seguinte PVC com dois pontos.

$$\frac{d^2y}{dx^2} + 2x \frac{dy}{dx} + 5y - \cos(3x) = 0, \quad \text{para } 0 \leq x \leq \pi \quad (9.65)$$

com as condições de contorno:  $y(0) = 1,5$  e  $y(\pi) = 0$ .

#### SOLUÇÃO

Para ser resolvida no MATLAB, a equação é escrita na forma:

$$\frac{d^2y}{dx^2} = -2x \frac{dy}{dx} - 5y + \cos(3x) \quad (9.66)$$

Em seguida, com a introdução de uma nova variável dependente  $w = \frac{dy}{dx}$ ,  $\frac{dw}{dx} = \frac{d^2y}{dx^2}$ , a EDO de segunda ordem da Eq. (9.66) é transformada no seguinte sistema de duas EDOs de primeira ordem:

$$\frac{dy}{dx} = w \quad (9.67)$$

$$\frac{dw}{dx} = -2xw - 5y + \cos(3x) \quad (9.68)$$

A função `bvp4c` do MATLAB tem a forma: `sol = bvp4c('edofun', 'ccfun', solinit)`. Antes de ser usada, duas funções, `edofun` e `ccfun`, devem ser criadas pelo usuário. No presente problema, a função `edofun` foi chamada de `edofunExemplo6`. Seu código é:

```
function dydx = edofunExemplo6(x,yw)
dydx = [yw(2)
        -2*x*yw(2) - 5*yw(1) + cos(3*x)];
```

Lado direito da Eq. (9.67).

Lado direito da Eq. (9.68).

Comentários sobre a função `edofunExemplo6`, criada pelo usuário:

- $yw$  é um vetor coluna no qual  $yw(1)$  é o valor de  $y$  e  $yw(2)$  é o valor de  $w$ .
- $dydx$  é um vetor coluna no qual  $dydx(1)$  é o valor do lado direito da Eq. (9.67) e  $dydx(2)$  é o valor do lado direito da Eq. (9.68).

No presente problema, a função `ccfun` foi chamada de `ccfunExemplo6`. O código da função é:

```
function res = bcfunExemplo6(ya,yb)
BCa = 1.5; BCb = 0;
res = [ya(1) - BCa
       yb(1) - BCb];
```

As condições de contorno são carregadas em `BCa` e `BCb`.

Resíduo em  $x = a$ .

Resíduo em  $x = b$ .

Comentários sobre a função `ccfunExemplo6`, criada pelo usuário:

- $ya$  é um vetor coluna no qual  $ya(1)$  é o valor numérico da solução  $y$  (calculado pelo MATLAB) em  $x = a$ ,  $yb(1)$  é o valor numérico da solução  $y$  (calculado pelo MATLAB) em  $x = b$ .
- $res$  é um vetor coluna no qual  $res(1)$  é o valor do resíduo em  $x = a$  e  $res(2)$  é o valor do resíduo em  $x = b$ .

Uma vez escritas as funções `edofunExemplo6` e `ccfunExemplo6`, elas são usadas em conjunto com as funções residentes do MATLAB `bvpinit` e `bvp4c` no programa a seguir, escrito em arquivo texto, para resolver o PVC da Eq. (9.65).

#### % Solução do Exemplo 9-6

```
clear all
```

Grade inicial.

```
solinit = bvpinit(linspace(0,pi,20),[0.2, 0.2]);
```

Tentativa inicial da solução.

```
sol = bvp4c('edofunExemplo6','ccfunExemplo6',solinit)
```

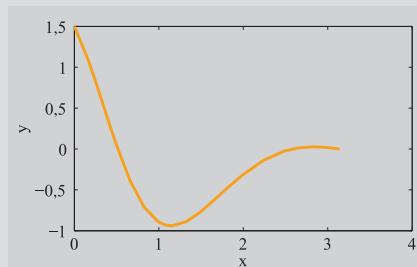
Cria a estrutura `solinit` usando a função `bvpinit`.

```
plot(sol.x, sol.y(1,:),'r')
```

Usa a função `bvp4c` para resolver a EDO.

```
xlabel('x'); ylabel('y')
```

Quando o programa é executado, exibe-se a figura ao lado, que contém o gráfico da solução. Além disso, os campos da estrutura `sol` são mostrados na janela de comandos do MATLAB (já que nenhum ponto e vírgula foi digitado no final do comando `sol=....`).



A mensagem na janela de comandos é:

```
sol =
```

```
    x: [1x24 double]
```

```
    y: [2x24 double]
```

```
    yp: [2x24 double]
```

```
solver: 'bvp4c'
```

`x` é um vetor linha com as coordenadas  $x$  dos pontos internos.

`y` é uma matriz ( $2 \times 24$ ) com os valores da solução nos pontos internos. A primeira linha é a solução para  $y$  e a segunda linha é a solução para  $w$ .

`yp` é uma matriz ( $2 \times 24$ ) com os valores das derivadas,  $\frac{dy}{dx}$  (primeira linha), e  $\frac{dw}{dx}$  (segunda linha).

O conteúdo de um campo pode ser mostrado digitando-se `sol.nome_do_campo`. Por exemplo, `sol.x` mostra os seguintes valores das coordenadas  $x$  nos pontos internos:

```
>> sol.x
```

```
ans =
```

```
Columns 1 through 12
```

```
0 0.1653 0.2480 0.3307 0.3720 0.4134 0.4960 0.6614 0.8267 0.9921 1.0748 1.1574
```

```
Columns 13 through 24
```

```
1.3228 1.4881 1.6535 1.8188 1.9842 2.2322 2.4802 2.6456 2.8109 2.8936 2.9762 3.1416
```

Isso mostra que a solução foi obtida usando 23 subintervalos.

Se `sol.y(1, :)` for digitado, exibe-se a primeira linha de `y`, que corresponde à solução nos pontos internos:

```
>> sol.y(1,:)
```

```
ans =
```

```
Columns 1 through 12
```

```
1.5000 1.0849 0.8354 0.5728 0.4404 0.3089 0.0544 -0.3909 -0.7116 -0.8908 -0.9303 -0.9404
```

```
Columns 13 through 24
```

```
-0.8901 -0.7750 -0.6271 -0.4711 -0.3242 -0.1438 -0.0276 0.0124 0.0260 0.0247 0.0192 0
```

## 9.5 ERRO E ESTABILIDADE NA SOLUÇÃO NUMÉRICA DE PROBLEMAS DE VALOR DE CONTORNO

### Erro numérico

O erro numérico associado a um problema de valor de contorno depende do método utilizado. No método do tiro, o erro numérico é o mesmo associado ao problema de valor inicial discutido na Seção 8.5. Obviamente, isso ocorre porque o método do tiro resolve o PVC rescrevendo-o em uma série de PVIs com condições de contorno à esquerda assumidas.

No caso da aplicação do método de diferenças finitas na solução de um PVC, o erro é determinado pela ordem da precisão do esquema numérico utilizado. Os erros de truncamento das diferentes aproximações utilizadas são discutidos na Seção 6.3. Conforme discutido na Seção 6.9, o erro total consiste no erro de truncamento e no erro de arredondamento. A precisão da solução obtida com o método de diferenças finitas é portanto determinada pelo maior dos dois erros de truncamento: aquele associado ao esquema de diferenças usado para representar a equação diferencial ou aquele usado para discretizar as condições de contorno. Um esforço deve ser feito para assegurar que a ordem do erro de truncamento seja a mesma para as condições de contorno e a equação diferencial.

### Estabilidade

A estabilidade da solução numérica de problemas de valor inicial foi discutida na Seção 8.13. A solução numérica de problemas de valor de contorno também é suscetível à instabilidade, embora a solução possa se tornar instável de diferentes maneiras. Em um PVI, a instabilidade foi associada ao crescimento do erro à medida que a integração progredia. Em contraste, em problemas de valor de contorno, o crescimento do erro numérico à medida que a solução avança é limitado pelas condições de contorno. Em alguns casos, pode haver múltiplas soluções válidas para o PVC. Isso faz com que, durante sua solução como um PVI, pequenas alterações na restrição inicial (isto é, na condição de contorno à esquerda) possam produzir diferentes soluções.

No caso do método do tiro (Exemplo 9-2), pode-se ver que, mesmo se duas soluções aproximadas geradas por diferentes tentativas para a incógnita  $\frac{dT}{dx}\Big|_{x=0}$  apresentarem erro, uma solução numérica precisa pode ainda assim ser obtida com o aprisionamento da condição de contorno à direita pelas duas soluções numéricas obtidas. Logo, a restrição à direita evita que os erros ilustrados no Exemplo 8-13 cresçam sem limites. Isso não significa que a solução nos pontos internos esteja livre de erros. Essa solução pode ainda assim exibir os erros de propagação mostrados no Exemplo 8-13, dependendo da escolha do método e do passo de integração.

Em alguns casos, a própria equação diferencial pode ser instável a pequenas perturbações nas condições de contorno. Neste caso, a formulação do problema deve ser examinada. Em outros casos, múltiplas soluções válidas podem existir para a EDO se diferentes condições de contorno à direita forem consideradas.

O uso do método de diferenças finitas na solução de um PVC faz com que o problema da estabilidade dependa da técnica usada para resolver o sistema de

equações. Como a solução é determinada simultaneamente em todos os pontos, a noção de avançar no tempo ou de avançar da esquerda para a direita presente em problemas de valor inicial não é relevante em problemas de valor de contorno. A estabilidade associada à solução de um PVC usando o método de diferenças finitas depende portanto da estabilidade do esquema utilizado para resolver o conjunto de equações simultâneas resultante. No Exemplo 9-3, o sistema de equações resultante associado às temperaturas desconhecidas nos pontos internos é linear. Nesse caso, são verificadas todas as potenciais dificuldades associadas à solução de sistemas de equações lineares discutidas no Capítulo 4, como o condicionamento. No Exemplo 9-4, o sistema de equações é não-linear. Nesse caso, a estabilidade é determinada pelo tipo de método usado para resolver o sistema, bem como pela proximidade da tentativa inicial em relação à solução. Como a iteração de ponto fixo foi usada para resolver o sistema de equações não-lineares no Exemplo 9-4, a estabilidade da solução numérica depende da escolha da função de iteração, bem como da tentativa inicial usada para iniciar as iterações. Isso mostra que a condição de contorno à direita não é capaz de prevenir a divergência no caso de algumas funções de iteração.

## 9.6 PROBLEMAS

### Problemas de solução manual

*Resolva manualmente os problemas a seguir. Quando necessário, use uma calculadora ou escreva um programa no MATLAB para realizar os cálculos. Se usar o MATLAB, não utilize funções residentes para obter soluções numéricas.*

**9.1** Considere a EDO de segunda ordem a seguir:

$$\frac{d^2y}{dx^2} = y + x(x - 4)$$

- (a) Usando a fórmula de diferença central para aproximar a derivada segunda, discretize a EDO (rescreva a equação em uma forma adequada para a solução usando o método de diferenças finitas).
- (b) Se  $h = 1$ , qual é o valor dos elementos diagonais da matriz de coeficientes do sistema de equações lineares resultante?

**9.2** Considere a EDO de segunda ordem a seguir:

$$\frac{d^2y}{dr^2} + \frac{y}{r} = C$$

- (a) Usando a fórmula de diferença central para aproximar a derivada segunda, discretize a EDO (rescreva a equação em uma forma adequada para a solução usando o método de diferenças finitas).
- (b) Se  $h = 1$ , qual é o valor dos elementos diagonais da matriz de coeficientes do sistema de equações lineares resultante?

**9.3** Considere a EDO de segunda ordem a seguir:

$$\frac{d^2y}{dx^2} + x \frac{dy}{dx} + y = 2x \text{ para } 0 \leq x \leq 1, \text{ com } y(0) = 1 \text{ e } y(1) = 1$$

- (a) Usando as fórmulas de diferença central para aproximar as derivadas, discretize a EDO (rescreva a equação em uma forma adequada para a solução usando o método de diferenças finitas).
- (b) Qual é a expressão para os termos acima da diagonal principal na matriz de coeficientes do sistema tridiagonal de equações lineares resultante?

**9.4** Considere o seguinte problema de valor de contorno:

$$\frac{d^2y}{dx^2} + ay + by^4 = 0 \text{ para } 0 \leq x \leq 1, \text{ com condições de contorno: } \left. \frac{dy}{dx} \right|_{x=0} = 0 \text{ e } y(1) = 1$$

onde  $a$  e  $b$  são constantes. Discretize a EDO de segunda ordem usando:

- (a) A diferença progressiva com precisão de segunda ordem.
- (b) A diferença regressiva com precisão de segunda ordem.
- (c) Discretize a condição de contorno em  $x = 0$  usando a diferença progressiva com precisão de segunda ordem.

**9.5** Considere o seguinte problema de valor de contorno:

$$\frac{d^3y}{dx^3} + \frac{dy}{dx} - y^4 = 0 \text{ para } 0 \leq x \leq 1, \text{ com condições de contorno: } y(0) = 0, \left. \frac{dy}{dx} \right|_{x=0} = 0 \text{ e } y(1) = 10.$$

Discretize a EDO de terceira ordem usando fórmulas de diferença central de segunda ordem. Quando for discretizar as condições de contorno, assegure-se de fazer com que o erro de truncamento seja compatível com aquele da EDO.

**9.6** Considere o seguinte problema de valor de contorno:

$$\frac{d^2y}{dx^2} + \frac{1}{x} \frac{dy}{dx} = -Q \text{ para } 0 \leq x \leq 1, \text{ com condições de contorno: } \left. \frac{dy}{dx} \right|_{x=0} = 0 \text{ e } \left. \frac{dy}{dx} \right|_{x=1} = ay(1) + by^4(1).$$

onde  $a$  e  $b$  são constantes. Discretize a EDO usando fórmulas de diferença central com precisão de segunda ordem. Quando for discretizar as condições de contorno, assegure-se de fazer com que o erro de truncamento seja compatível com aquele da EDO.

**9.7** Considere o seguinte problema de valor de contorno:

$$-\frac{d^2u}{dx^2} + \pi^2 u = 2\pi^2 \cos(\pi x) \text{ para } 0 \leq x \leq 1, \text{ com condições de contorno: } u(0) = 1 \text{ e } u(1) = -1$$

Quais são os elementos diagonais da matriz tridiagonal resultante quando o método de diferenças finitas usando diferenças centrais com precisão de primeira ordem é aplicado na solução do problema com um passo de integração de  $1/8$ ?

**9.8** Considere a EDO de segunda ordem na forma:

$$\frac{d^2y}{dx^2} + p \frac{dy}{dx} + qy = r(x)$$

onde  $p$  e  $q$  são constantes, e  $r(x)$  é uma função fornecida. Usando diferenças centrais com precisão de segunda ordem para representar as derivadas, discretize a EDO.

### Problemas de programação no MATLAB

*Resolva os seguintes problemas no ambiente MATLAB. Não use funções residentes para resolver as equações diferenciais.*

**9.9** Escreva uma função no MATLAB que resolva, usando o método do tiro, um problema de valor de contorno na forma:

$$\frac{d^2y}{dx^2} + f(x) \frac{dy}{dx} + g(x)y = h(x) \quad \text{para } a \leq x \leq b \quad \text{com } y(a) = Y_a \quad \text{e} \quad y(b) = Y_b$$

onde  $Y_a$  e  $Y_b$  são constantes. A função deve calcular primeiro duas soluções usando dois valores assumidos para a inclinação em  $x = a$ , especificados pelo usuário, e usar essas soluções no cálculo de uma nova inclinação inicial usando interpolação (Eq. (9.14)). Essa inclinação inicial deve ser em seguida utilizada no cálculo da solução final do problema. Chame a função de  $[x, y] = \text{PVCTir} \circ \text{Int}(\text{fOFx}, \text{gOFx}, \text{hOFx}, a, b, n, Ya, Yb, WL, WH)$ , onde  $\text{fOFx}$ ,  $\text{gOFx}$  e  $\text{hOFx}$  são os nomes (*strings*) de funções em arquivo criadas para calcular  $f(x)$ ,  $g(x)$  e  $h(x)$ , respectivamente,  $a$  e  $b$  definem o domínio da solução,  $n$  é o número de subintervalos,  $Ya$  e  $Yb$  são as condições de contorno e  $WL$  e  $WH$  são as inclinações assumidas em  $x = a$ . Uma vez calculadas as duas primeiras soluções, o programa deve confirmar que, em  $x = b$ , a condição de contorno  $Y_b$  está entre as duas soluções, e então calcular a solução final com o valor interpolado da inclinação. Se a condição de contorno em  $x = b$  não estiver entre as duas soluções, o programa deve ser interrompido e exibir uma mensagem de erro. Use a função *Sis2EDOsRK4* escrita no Exemplo 8-8 para resolver o sistema de duas EDOs de primeira ordem dentro da função *PVCTiroInt*.

Use *PVCTiroInt* para resolver o problema de valor de contorno do Exemplo 9-6. Use  $n = 100$ ,  $W_L = -5$  e  $W_H = -1,5$ .

**9.10** Escreva uma função no MATLAB que resolva, usando o método do tiro em conjunção com o método da bissecção, um problema de valor de contorno na forma:

$$\frac{d^2y}{dx^2} + f(x) \frac{dy}{dx} + g(x)y = h(x) \quad \text{para } a \leq x \leq b \quad \text{com } y(a) = Y_a \quad \text{e} \quad y(b) = Y_b$$

onde  $Y_a$  e  $Y_b$  são constantes. Chame a função de  $[x, y] = \text{PVCTiroBissecc}(\text{fOFx}, \text{gOFx}, \text{hOFx}, a, b, n, Ya, Yb, WL, WH)$ , onde  $\text{fOFx}$ ,  $\text{gOFx}$  e  $\text{hOFx}$  são os nomes (*strings*) de funções em arquivo criadas para calcular  $f(x)$ ,  $g(x)$  e  $h(x)$ , respectivamente,  $a$  e  $b$  definem o domínio da solução,  $n$  é o número de subintervalos,  $Ya$  e  $Yb$  são as condições de contorno e  $WL$  e  $WH$  são as inclinações assumidas em  $x = a$ . Uma vez calculadas as duas primeiras soluções, o programa deve confirmar que, em  $x = b$ , a condição de contorno  $Y_b$  está entre as duas soluções, e então calcular a solução final com o valor interpolado da inclinação. Se a condição de contorno em  $x = b$  não estiver entre as duas soluções, o programa deve ser interrompido e exibir uma mensagem de erro. Use a função *Sis2EDOsRK4* escrita no Exemplo 8-8 para resolver o sistema de

duas EDOs de primeira ordem dentro da função PVCTiroBissecc. Interrompa as iterações quando o valor absoluto do erro em  $x = b$  for menor que 0,001.

Use PVCTiroBissecc para resolver o problema de valor de contorno do Exemplo 9-6. Use  $n = 100$ ,  $W_L = -5$  e  $W_H = -1,5$ .

**9.11** Escreva uma função no MATLAB que resolva, usando o método do tiro em conjunção com o método da secante, um problema de valor de contorno na forma:

$$\frac{d^2y}{dx^2} + f(x)\frac{dy}{dx} + g(x)y = h(x) \text{ para } a \leq x \leq b \text{ com } y(a) = Y_a \text{ e } y(b) = Y_b$$

onde  $Y_a$  e  $Y_b$  são constantes. Chame a função de  $[x, y] = \text{PVCTiroSecante}(fOFx, gOFx, hOFx, a, b, n, Ya, Yb, WL, WH)$ , onde  $fOFx$ ,  $gOFx$  e  $hOFx$  são os nomes (*strings*) de funções em arquivo criadas para calcular  $f(x)$ ,  $g(x)$  e  $h(x)$ , respectivamente,  $a$  e  $b$  definem o domínio da solução,  $n$  é o número de subintervalos,  $Ya$  e  $Yb$  são as condições de contorno e  $WL$  e  $WH$  são as inclinações assumidas em  $x = a$ . Use a função Sis2EDOsRK4 escrita no Problema 8.8 para resolver o sistema de duas EDOs de primeira ordem dentro da função PVCTiroSecante. Interrompa as iterações quando o valor absoluto do erro em  $x = b$  for menor que 0,001.

Use PVCTiroSecante para resolver o problema de valor de contorno do Exemplo 9-6. Use  $n = 100$ ,  $W_L = -5$  e  $W_H = -1,5$ .

**9.12** Escreva uma função no MATLAB que use o método de diferenças finitas para resolver uma EDO de segunda ordem na forma:

$$\frac{d^2y}{dx^2} + p\frac{dy}{dx} + qy = r(x) \text{ para } a \leq x \leq b \text{ com } y(a) = Y_a \text{ e } y(b) = Y_b$$

onde  $p$ ,  $q$ ,  $Y_a$  e  $Y_b$  são constantes. Discretize a EDO usando fórmulas de diferença central com precisão de segunda ordem. Chame a função de  $[x, y] = \text{PVC2ordConst}(a, b, n, Ya, Yb, p, q, rOFx)$ , onde  $a$  e  $b$  definem o domínio da solução,  $n$  é o número de subintervalos e  $rOFx$  é o nome da função em arquivo criada pelo usuário para calcular  $r(x)$ . Dentro do programa, use a divisão à esquerda do MATLAB para resolver o sistema de equações lineares.

Use PVC2ordConst com 50 subintervalos para resolver a EDO do Problema 9.1 com condições iniciais  $y(0) = 1$ ,  $y(2) = 0$ . Trace um gráfico com a solução.

**9.13** Escreva uma função no MATLAB que use o método de diferenças finitas para resolver um problema de valor de contorno na forma:

$$\frac{dy^2}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = r(x) \text{ para } a \leq x \leq b \text{ com } y(a) = Y_a \text{ e } y(b) = Y_b$$

onde  $Y_a$  e  $Y_b$  são constantes. Discretize a EDO usando fórmulas de diferença central com precisão de segunda ordem. Chame a função de  $[x, y] = \text{PVC2ordVar}(a, b, Ya, Yb, n, pOFx, qOFx, rOFx)$ , onde  $a$  e  $b$  definem o domínio da solução,  $Ya$  e  $Yb$  são as condições de contorno,  $n$  é o número de subintervalos e  $pOFx$ ,  $qOFx$  e  $rOFx$  são os nomes (*strings*) das funções em arquivo criadas pelo usuário para calcular  $p(x)$ ,  $q(x)$  e  $r(x)$ , respectivamente. Dentro do programa, use a divisão à esquerda do MATLAB para resolver o sistema de equações lineares.

Use PVC2ordVar com 50 subintervalos para resolver o problema de valor de contorno do Problema 9.3. Trace um gráfico com a solução.

**9.14** Escreva uma função no MATLAB que use o método de diferenças finitas para resolver um problema de valor de contorno na forma:

$$\frac{dy^2}{dx^2} + p(x) \frac{dy}{dx} + q(x)y = r(x) \text{ para } a \leq x \leq b \quad \text{com} \quad y(a) = Y_a \quad \text{e} \quad \left. \frac{dy}{dx} \right|_{x=b} = D_b$$

onde  $Y_a$  e  $D_b$  são constantes. Discretize a EDO usando fórmulas de diferença central com precisão de segunda ordem. Chame a função de  $[x, y] = \text{PVC2ordDeriv}(a, b, Y_a, D_b, n, pOFx, qOFx, rOFx)$ , onde  $a$  e  $b$  definem o domínio da solução,  $Y_a$  e  $D_b$  são as condições de contorno,  $n$  é o número de subintervalos e  $pOFx$ ,  $qOFx$  e  $rOFx$  são os nomes (*strings*) das funções em arquivo criadas pelo usuário para calcular  $p(x)$ ,  $q(x)$  e  $r(x)$ , respectivamente. Dentro do programa, use a divisão à esquerda do MATLAB para resolver o sistema de equações lineares.

Use PVC2ordDeriv com 50 subintervalos para resolver o seguinte problema de valor de contorno:

$$\frac{d^2y}{dx^2} + \frac{1}{x} \frac{dy}{dx} = -10, \quad y(1) = 1, \quad \left. \frac{dy}{dx} \right|_{x=3} = -1,2$$

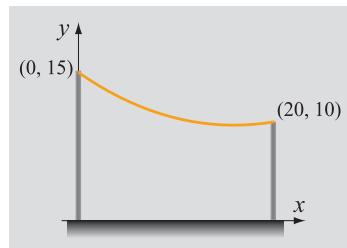
Trace um gráfico com a solução.

### Problemas na matemática, na ciência e na engenharia

Resolva os seguintes problemas no ambiente MATLAB. Use programas apresentados neste capítulo, programas desenvolvidos em problemas já resolvidos ou funções residentes do MATLAB.

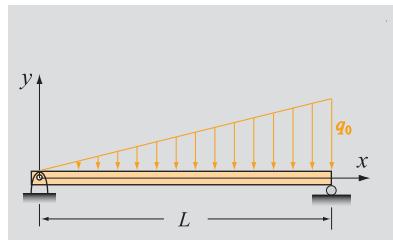
**9.15** Um cabo flexível de densidade uniforme está suspenso entre dois pontos conforme mostra a figura. A forma do cabo,  $y(x)$ , é descrita pela equação diferencial:

$$\frac{d^2y}{dx^2} = C \sqrt{1 + \left( \frac{dy}{dx} \right)^2}$$



onde  $C$  é uma constante igual à relação entre o peso por unidade de comprimento do cabo e a amplitude da componente horizontal da tensão no cabo em seu ponto mais baixo. O cabo está pendurado entre dois pontos especificados por  $y(0) = 15$  m e  $y(20) = 10$  m e  $C = 0,041 \text{ m}^{-1}$ . Use funções residentes do MATLAB para determinar e traçar a forma do cabo entre  $x = 0$  e  $x = 20$  m.

**9.16** Uma viga de comprimento  $L = 4$  m é carregada por uma carga distribuída, conforme mostrado na figura. A deflexão da viga,  $y$ , é determinada a partir da solução da seguinte EDO (quando as deflexões são pequenas):



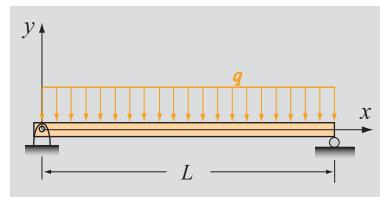
$$EI \frac{d^2y}{dx^2} = \frac{1}{6} q_0 \left( Lx - \frac{x^3}{L} \right) \quad y(0) = 0 \quad \text{e} \quad y(L) = 0$$

onde  $EI = 1,2 \times 10^7 \text{ N}\cdot\text{m}^2$  é a rigidez flexural e  $q_0 = 30 \times 10^3 \text{ N/m}$ .

Determine e trace a deflexão da viga em função de  $x$ .

- Use a função PVCTiroBissecc escrita no Problema 9.10, com  $n = 100$ ,  $W_L = 0$  e  $W_H = -0,005$ .
- Use a função PVC2ordConst escrita no Problema 9.12, com 50 subintervalos.
- Use funções residentes do MATLAB.

**9.17** Uma viga de comprimento  $L = 4 \text{ m}$  é carregada por uma carga distribuída, conforme mostrado na figura. Para grandes deflexões, a deflexão da viga,  $y$ , é determinada a partir da solução da seguinte EDO:



$$EI \frac{d^2y}{dx^2} = \left[ 1 + \left( \frac{dy}{dx} \right)^2 \right]^{3/2} \frac{1}{2} q (Lx - x^2) \quad y(0) = 0 \quad \text{e} \quad y(L) = 0$$

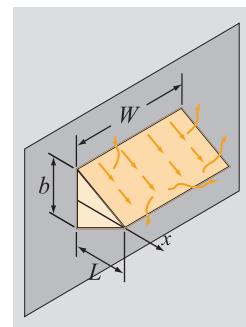
onde  $EI = 1,4 \times 10^7 \text{ N}\cdot\text{m}^2$  é a rigidez flexural e  $q = 10 \times 10^3 \text{ N/m}$ .

Use funções residentes do MATLAB para determinar e traçar a deflexão da viga em função de  $x$ .

**9.18** A distribuição de temperaturas  $T(x)$  em um dissipador com formato triangular é dada pela solução da seguinte equação:

$$\frac{d^2T}{dx^2} - \frac{1}{(L-x)} \frac{dT}{dx} - \frac{2h(LW + bL - bx)(T - T_\infty)}{kbW(L-x)} = 0$$

onde  $T_\infty = 300 \text{ K}$  é a temperatura ambiente,  $x$  é a coordenada medida ao longo do dissipador,  $k = 237 \text{ W/m/K}$  é a condutividade térmica do alumínio,  $h = 15 \text{ W/m}^2/\text{K}$  é o coeficiente de transferência de calor por convecção,  $L = 0,01 \text{ m}$  é o comprimento do dissipador,  $W = 0,1 \text{ m}$  é sua largura e  $b = 0,01 \text{ m}$  é a altura da base. As condições de contorno são:



$$T(x=0) = 673 \text{ K} \quad \text{e} \quad \left. \frac{dT}{dx} \right|_{x=L} = 0$$

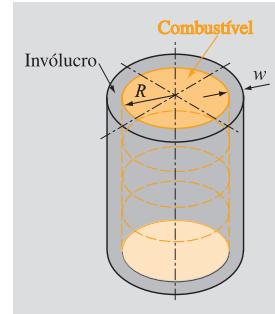
Determine e trace a distribuição de temperaturas  $T(x)$  ao longo do dissipador. Escreva um programa que resolva o problema usando o método do tiro. Note que uma derivada é fornecida como condição de contorno em  $x = L$ . Use a fórmula de diferença regressiva com três pontos para calcular o valor da derivada a partir da solução numérica em  $x = L$ . Compare a solução numérica com a condição de contorno prescrita e use o método da bisseção para calcular a nova estimativa para a inclinação em  $x = 0$ . Interrompa as iterações quando o valor absoluto do erro relativo em  $x = L$  for menor que 0,01. **Nota importante:** O ponto  $x = L$  é uma singularidade da EDO. Portanto, o problema não pode ser resolvido da maneira especificada. No entanto, uma solução aproximada pode ser obtida atribuindo-se ao dissipador um comprimento  $L = 0,0099999 \text{ m}$ .

**9.19** Resolva o Problema 9.18 usando o método de diferenças finitas com 100 subintervalos. Use fórmulas de diferença central com precisão de segunda ordem para representar as derivadas presentes na EDO e fórmulas de diferença unilaterais adequadas à representação das condições de contorno. Conforme explicado no Problema 9.18, use  $L = 0,00999999$  m como comprimento do dissipador.

**9.20** A pastilha de combustível de um reator nuclear é uma estrutura cilíndrica em que o combustível é retido no interior de um invólucro, conforme mostrado na figura. O combustível faz com que calor seja gerado no interior do cilindro e também no invólucro em decorrência de reações nucleares. A superfície externa do invólucro é resfriada com água corrente em  $T_\infty = 473$  K com um coeficiente de transferência de calor de  $h = 10^4$  W/m<sup>2</sup>/K. A condutividade térmica do material utilizado no invólucro é  $k = 16,75$  W/m/K. As dimensões da pastilha de combustível são  $R = 1,5 \times 10^{-2}$  m e  $w = 3,0 \times 10^{-3}$  m. A distribuição de temperaturas no invólucro é determinada pela solução do seguinte problema de valor de contorno:

$$\frac{1}{r} \frac{d}{dr} \left( rk \frac{dT}{dr} \right) = -10^8 \frac{e^{-r/R}}{r}, \text{ para } R \leq r \leq R + w, \text{ com } \left. \frac{dT}{dr} \right|_{r=R} = -\frac{6,32 \times 10^5}{k} \text{ e}$$

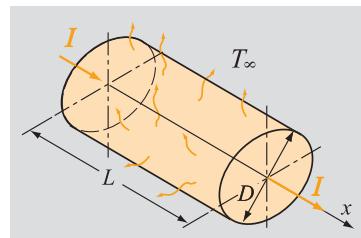
$$\left. \frac{dT}{dr} \right|_{r=R+w} = -\frac{h}{k} (T|_{r=R+w} - T_\infty)$$



Use a função residente do MATLAB bvp4c para resolver este problema de valor de contorno. Trace a distribuição de temperaturas no invólucro em função de  $r$ .

**9.21** A variação axial da temperatura em um fio desemcapado conduzindo corrente elétrica é descrita por:

$$\frac{d^2 T}{dx^2} - \frac{4h}{kD} (T - T_\infty) - \frac{4\varepsilon\sigma_{SB}(T^4 - T_\infty^4)}{kD} = -\frac{I^2 \rho_e}{k \left( \frac{1}{4}\pi D^2 \right)^2}$$



onde  $T$  é a temperatura em K,  $x$  é a coordenada ao longo do fio,  $k = 72$  W/m/K é a condutividade térmica,  $h = 2000$  W/m<sup>2</sup>/K<sup>2</sup> é o coeficiente de convecção de calor,  $\varepsilon = 0,1$  é a emissividade radiativa,  $\sigma_{SB} = 5,67 \times 10^{-8}$  W/m<sup>2</sup>/K<sup>4</sup> é a constante de Stefan-Boltzmann,  $I = 2$  A é a corrente,  $\rho_e = 32 \times 10^{-8}$  Ω.m é a resistividade elétrica,  $T_\infty = 300$  K é a temperatura ambiente,  $D = 7,62 \times 10^{-3}$  m é o diâmetro do fio e  $L = 4,0 \times 10^{-3}$  m é o comprimento do fio. As condições de contorno são:

$$\text{em } x = 0 \quad T = 300 \text{ K, em } x = \frac{L}{2} \quad \left. \frac{dT}{dx} \right|_{x=\frac{L}{2}} = 0$$

Use a função residente do MATLAB bvp4c para resolver este problema de valor de contorno para  $0 \leq x \leq \frac{L}{2}$ , já que a temperatura é simétrica em relação a  $x = \frac{L}{2}$ . Trace a distribuição de temperaturas ao longo do fio.

# Introdução ao MATLAB

### Tópicos centrais

Começando com o MATLAB (A.2)	Arquivos de programa (A.5)
Arranjos (A.3)	Arquivos de função (A.6)
Operações matemáticas com arranjos (A.4)	Programação no MATLAB (A.7)

## A.1 FUNDAMENTOS

O MATLAB é uma poderosa linguagem usada na computação técnica. O nome MATLAB vem do termo em inglês MATrix LABoratory (Laboratório de Matrizes), pois as matrizes (ou arranjos) são o seu elemento de dados básico. O MATLAB pode ser usado em cálculos matemáticos, modelagem e simulações, análise de dados e processamento, visualização e traçado de gráficos, e no desenvolvimento de algoritmos.

O MATLAB é amplamente utilizado em universidades e faculdades em cursos introdutórios e avançados em matemática, ciência e, especialmente, em engenharia. Na indústria, esse programa é usado em pesquisa, desenvolvimento e projeto. O programa padrão do MATLAB possui ferramentas (funções) que podem ser usadas para resolver problemas comuns. Além disso, o MATLAB possui caixas de ferramentas opcionais (*toolboxes*) formadas por conjuntos de programas especializados para resolver problemas específicos. Por exemplo, existem caixas de ferramentas como essas voltadas ao processamento de sinais, ao cálculo simbólico e aos sistemas de controle.

Este apêndice é uma breve introdução ao MATLAB. Ele apresenta sua sintaxe e suas características mais básicas, o que é suficiente para que o leitor possa seguir o seu uso neste livro. Para uma introdução mais completa, o leitor deve consultar o livro *MATLAB: An Introduction with Applications*, Second Edition\*, 2005, escrito por Amos Gilat.

## A.2 COMEÇANDO COM O MATLAB

Assume-se que o programa esteja instalado no computador e que o usuário possa iniciá-lo. Quando o programa está rodando, oito janelas podem ser usadas. Uma lista das várias janelas e de seu propósito é dada na Tabela A-1. Quatro janelas – a janela de comandos, a janela de figuras, a janela de edição e a janela de ajuda – são mais comumente utilizadas.

\* N. de T.: Publicado pela Bookman Editora sob o título *MATLAB com Aplicações em Engenharia*, 2.ed.

**Tabela A-1** Janelas do MATLAB

Janela	Finalidade
Janela de Comandos ( <i>Command Window</i> )	Janela principal, permite a entrada de variáveis e roda programas
Janela de Figuras ( <i>Figure Window</i> )	Contém a saída de comandos gráficos
Janela de Edição ( <i>Editor Window</i> )	Cria e procura erros em arquivos contendo programas e funções
Janela de Ajuda ( <i>Help Window</i> )	Fornece a informação de ajuda
Janela de Lançamento ( <i>Launch Pad Window</i> )	Permite o acesso a ferramentas, demonstrações e documentação
Janela Histórico de Comandos ( <i>Command History Window</i> )	Mostra os comandos utilizados na janela de comandos
Janela da Área de Trabalho ( <i>Workspace Window</i> )	Fornece informações sobre as variáveis utilizadas
Janela de Diretório Corrente ( <i>Current Directory Window</i> )	Mostra os arquivos no diretório corrente

**Janela de Comandos:** A janela de comandos é a tela principal do MATLAB, que é aberta quando o programa é iniciado.

- Comandos são digitados ao lado do cursor (>>) e executados quando a tecla **Enter** é pressionada.
- Uma vez digitado um comando e pressionada a tecla **Enter**, o comando é executado. Entretanto, executa-se apenas o último comando. Tudo o que foi executado anteriormente permanece inalterado.
- A saída gerada pelo comando é mostrada na janela de comandos, a menos que um ponto-e-vírgula (;) seja digitado no final da linha.
- Quando o símbolo % (percentual) é digitado no início de uma linha, essa se torna uma linha de comentários.
- O comando clc limpa a janela de comandos (digite clc e pressione **Enter**). Após trabalhar na janela de comandos por algum tempo, a lista de comandos utilizados pode se tornar muito longa. Uma vez executado o comando clc, a janela é limpa. O comando não muda nada que tenha sido realizado anteriormente. Por exemplo, se algumas variáveis já tiverem sido definidas, elas ainda existirão na memória e poderão ser usadas. A tecla (↑) pode ser usada para visualizar os comandos já digitados.

**Janela de Figuras:** A janela de figuras é aberta automaticamente quando comandos gráficos são executados. Ela contém gráficos criados por esses comandos.

**Janela de Edição:** A janela de edição é usada para escrever e editar programas. Essa janela é aberta a partir do menu **File** presente na janela de comandos. Mais detalhes sobre a janela de edição são fornecidos na Seção A.5, quando seu uso na criação de programas é apresentado.

**Janela de Ajuda:** Esta janela contém informações de ajuda. Ela pode ser aberta a partir do menu **Help** presente na barra de ferramentas de qualquer janela do

MATLAB. A janela de ajuda é interativa e pode ser usada na obtenção de informações sobre qualquer característica do MATLAB.

### Operações aritméticas fundamentais com escalares

A maneira mais simples de usar o MATLAB é tratá-lo como uma calculadora. Com escalares, os símbolos das operações aritméticas são:

Operação	Símbolo	Exemplo	Operação	Símbolo	Exemplo
Adição	+	$5 + 3$	Divisão à direita	/	$5 / 3$
Subtração	-	$5 - 3$	Divisão à esquerda	\	$5 \backslash 3 = 3 / 5$
Multiplicação	*	$5 * 3$	Exponenciação	^	$5 ^ 3$ (significa $5^3 = 125$ )

Expressões matemáticas podem ser digitadas na janela de comandos. Quando a tecla **Enter** é pressionada, o MATLAB calcula a expressão e fornece a resposta exibindo `ans` = e o resultado numérico da expressão na linha seguinte. Exemplos são:

```
>> 7 + 8/2
ans =
    11
>> (7+8)/2 + 27^(1/3)
ans =
    10.5000
```

Valores numéricos também podem ser atribuídos a variáveis, cujo nome pode ser formado por uma letra ou pela combinação de várias letras e números. Nomes de variáveis devem, no entanto, começar com uma letra. Assim que um valor numérico é atribuído a uma variável, esta pode ser usada em expressões matemáticas, em funções, e em quaisquer declarações e comandos do MATLAB.

```
>> a = 12
a =
    12
>> B = 4;
>> C = (a - B) + 40 - a/B*10
C =
    18
```

Como um ponto-e-vírgula é digitado no final do comando, o valor de B não é mostrado.

### Funções residentes contendo operações matemáticas elementares

Além das operações aritméticas básicas, expressões no MATLAB podem incluir funções. O MATLAB possui uma grande biblioteca de funções residentes. Uma função tem um nome e um argumento escrito entre parênteses. Por exemplo, `sqrt(x)` é a função que calcula a raiz quadrada de um número. Seu nome é `sqrt`, e o argumento é `x`. Quando a função é usada, o argumento pode ser um número, uma variável, ou uma expressão formada por números e/ou variáveis. Funções também podem ser incluídas em argumentos, bem como em expressões.

Os exemplos a seguir mostram o uso da função `sqrt(x)` quando o MATLAB trabalha com escalares.

```
>> sqrt(64)                                O argumento é um número.
ans =
    8
>> sqrt(50 + 14*3)                         O argumento é uma expressão.
ans =
    9.5917
>> sqrt(54 + 9*sqrt(100))                  O argumento inclui uma função.
ans =
    12
>> (15 + 600/4)/sqrt(121)                  A função é incluída em uma expressão.
ans =
    15
```

Uma lista de algumas funções elementares mais comumente usadas no MATLAB é fornecida na Tabela A-2. Uma lista completa de funções organizadas por nome e categoria pode ser encontrada na janela de ajuda.

**Tabela A-2** Funções matemáticas elementares

Comando	Descrição	Exemplo
<code>sqrt(x)</code>	Raiz quadrada	<code>&gt;&gt; sqrt(81)</code> ans = 9
<code>exp(x)</code>	Exponencial ( $e^x$ )	<code>&gt;&gt; exp(5)</code> ans = 148.4132
<code>abs(x)</code>	Valor absoluto	<code>&gt;&gt; abs(-24)</code> ans = 24
<code>log(x)</code>	Logaritmo natural, na base $e$ (ln)	<code>&gt;&gt; log(1000)</code> ans = 6.9078
<code>log10(x)</code>	Logaritmo na base 10	<code>&gt;&gt; log10(1000)</code> ans = 3.0000
<code>sin(x)</code>	Seno de um ângulo $x$ (em radianos)	<code>&gt;&gt; sin(pi/6)</code>
<code>sind(x)</code>	Seno de um ângulo $x$ (em graus)	<code>&gt;&gt; sind(30)</code> ans = 0.5000

As demais funções trigonométricas são escritas da mesma maneira. As funções trigonométricas inversas são escritas simplesmente acrescentando “a” na frente dos comandos, como, por exemplo, `asin(x)`.

**Tabela A-2** Funções matemáticas elementares (*continuação*)

Comando	Descrição	Exemplo
round(x)	Arredonda para o inteiro mais próximo	>> round(17/5) ans = 3
fix(x)	Arredonda em direção a zero	>> fix(13/5) ans = 2
ceil(x)	Arredonda em direção a infinito	>> ceil(11/5) ans = 3
floor(x)	Arredonda em direção a menos infinito	>> floor(-9/4) ans = -3

### Formatos de exibição

O formato no qual o MATLAB mostra a saída na tela pode ser alterado pelo usuário. O formato de saída padrão é o de ponto fixo com quatro algarismos decimais (denominado `short`). O formato pode ser alterado com o comando `format`. Uma vez digitado esse comando, todas as saídas apresentadas em seguida são mostradas no formato especificado. Vários dos formatos disponíveis no MATLAB são listados e descritos na Tabela A-3.

**Tabela A-3** Formatos de exibição

Comando	Descrição	Exemplo
<code>format short</code>	Ponto fixo com quatro algarismos decimais para: $0,001 \leq \text{número} \leq 1000$ Senão, use o formato <code>short e</code>	>> 290/7 ans = 41.4286
<code>format long</code>	Ponto fixo com quatorze algarismos decimais para: $0,001 \leq \text{número} \leq 1,00$ Senão, use o formato <code>long e</code>	>> 290/7 ans = 41.42857142857143
<code>format short e</code>	Notação científica com quatro algarismos decimais	>> 290/7 ans = 4.1429e+001
<code>format long e</code>	Notação científica com quinze algarismos decimais	>> 290/7 ans = 4.142857142857143e+001
<code>format bank</code>	Dois algarismos decimais	>> 290/7 ans = 41.43

### A.3 ARRANJOS

O arranjo é a forma fundamental que o MATLAB utiliza para armazenar e manipular dados. Um arranjo é uma lista de números organizados em linhas e/ou colunas. O arranjo mais simples (unidimensional) corresponde a uma linha ou uma coluna de números, o que é comumente chamado de vetor na ciência e na engenharia. Um arranjo mais complexo (bidimensional) é uma coleção de números organizados em linha e colunas, o que é chamado de matriz na ciência e na engenharia. Cada número presente em um vetor ou matriz é chamado de elemento. Esta seção mostra como construir vetores e matrizes. A Seção A.4 mostra como realizar operações matemáticas com arranjos.

#### Criando um vetor

No MATLAB, um vetor é criado com a atribuição dos elementos do vetor a uma variável. Isso pode ser feito de várias maneiras, dependendo da fonte de informação utilizada para os elementos do vetor. Quando um vetor contém números específicos conhecidos, o valor de cada elemento pode ser fornecido diretamente com a digitação dos valores dos elementos no interior de colchetes:

```
nome_da_variável = [número número... número]
```

Para formar um vetor linha, os números devem ser digitados utilizando espaços ou vírgulas para separar cada elemento. Para formar um vetor coluna, um ponto-e-vírgula deve ser digitado entre cada elemento. Os elementos também podem ser expressões matemáticas incluindo variáveis predefinidas, números e funções. Muitas vezes, os elementos de um vetor linha são formados por uma série de números com espaçamento constante. Nesses casos, o vetor pode ser criado digitando-se:

```
nome_da_variável = m:q:n
```

onde  $m$  é o primeiro elemento,  $q$  é o espaçamento entre os elementos e  $n$  é o último elemento. Outra opção é o comando `linspace`:

```
nome_da_variável = linspace(xi, xf, n)
```

Vários exemplos de construção de vetores são mostrados a seguir:

```
>> ano = [1984 1986 1988 1990 1992 1994 1996] Vetor linha formado com a digitação dos elementos.
ano =
    1984      1986      1988      1990      1992      1994      1996
>> pnt = [2; 4; 5] Vetor coluna formado com a digitação dos elementos.
pnt =
    2
    4
    5
>> x = [1:2:13] Vetor linha com espaçamento constante.
```

```
x =
1 3 5 7 9 11 13
>> va = linspace(0,8,6) Vetor linha com 6 elementos, primeiro elemento 0 e último elemento 8.
va =
0 1.6000 3.2000 4.8000 6.4000 8.0000
```

### Criando um arranjo bidimensional (matriz)

Um arranjo bidimensional, também chamado de matriz, tem números dispostos em linhas e colunas. Uma matriz é criada com a atribuição dos elementos da matriz a uma variável. Isso é feito com a digitação dos elementos, linha a linha, no interior de colchetes. Em cada linha, os elementos devem ser separados por espaços ou linhas.

```
nome_da_variável=[1a linha de elementos; 2a linha de elementos; ...; última linha de elementos]
```

Os elementos de uma matriz podem ser números ou expressões. Estas podem incluir números, variáveis predefinidas e funções. Todas as linhas devem ter o mesmo número de elementos. Se um elemento for nulo, ele deve ser fornecido como tal. O MATLAB mostra uma mensagem de erro caso uma matriz incompleta seja formada. Exemplos de matrizes criadas de diferentes maneiras são:

```
>> a = [5 35 43; 4 76 81; 21 32 40] Um ponto-e-vírgula é digitado entre cada linha.
a =
5 35 43
4 76 81
21 32 40
>> cd = 6; e = 3; h = 4; Definem-se as variáveis.
>> Mat = [e, cd*h, cos(pi/3); h^2, sqrt(h*h/cd), 14]
Mat = Elementos são fornecidos como expressões matemáticas.
3.0000 24.0000 0.5000
16.0000 1.6330 14.0000
```

- Todas as variáveis no MATLAB são arranjos. Um escalar é um arranjo com apenas um elemento; um vetor é um arranjo com uma linha ou uma coluna de elementos; uma matriz é um arranjo com elementos dispostos em linhas e colunas.
- A variável (escalar, vetor ou matriz) é definida ao ser carregada. Não é necessário definir o tamanho de um arranjo (um único elemento no caso de um escalar, uma linha ou coluna de elementos no caso de um vetor, ou um arranjo bidimensional de elementos no caso de uma matriz) antes que seus elementos sejam carregados.
- Se uma variável existir na forma de um escalar, de um vetor, ou de uma matriz, ela pode ser alterada para qualquer outro tamanho ou tipo de variável. Por exemplo, um escalar pode se transformar em um vetor ou em uma matriz, um vetor pode se transformar em um escalar, em um vetor de diferente tamanho ou em uma matriz, e uma matriz pode ser alterada para ter um diferente tamanho, o mesmo reduzida para se tornar um vetor ou um escalar. Essas mudanças são feitas com a adição ou com a subtração de elementos.

### Endereçamento de arranjos

Elementos de um arranjo (seja ele um vetor ou uma matriz) podem ser chamados individualmente ou em subgrupos. Isso é útil quando é necessário redefinir apenas alguns elementos, ou usar elementos específicos em cálculos, ou quando um subgrupo de elementos é utilizado na definição de uma nova variável.

O endereço de um elemento em um vetor é sua posição na linha (ou na coluna). Em um vetor  $ve$ ,  $ve(k)$  se refere ao elemento na posição  $k$ . O endereço da primeira posição é 1. Por exemplo, se o vetor  $ve$  tiver nove elementos:

$$ve = 35 \ 46 \ 78 \ 23 \ 5 \ 14 \ 81 \ 3 \ 55$$

então

$$ve(4) = 23, ve(7) = 81 \text{ e } ve(1) = 35.$$

O endereço de um elemento em uma matriz corresponde a sua posição, definida pelos números da linha e da coluna onde está localizado. Em uma matriz atribuída a uma variável  $ma$ ,  $ma(k,p)$  se refere ao elemento na linha  $k$  e na coluna  $p$ .

Por exemplo, se a matriz é:  $ma = \begin{bmatrix} 3 & 11 & 6 & 5 \\ 4 & 7 & 10 & 2 \\ 13 & 9 & 0 & 8 \end{bmatrix}$

então,  $ma(1,1) = 3$  e  $ma(2,3) = 10$ .

É possível mudar o valor de um elemento simplesmente atribuindo um novo valor ao elemento específico. Elementos também podem ser usados como variáveis em expressões matemáticas.

```
>> VCT = [35 46 78 23 5 14 81 3 55]
VCT =
    35   46   78   23   5   14   81   3   55
>> VCT(4), VCT(6) = 273
ans =
    35   46   78   23   5   273   81   3   55
>> VCT(5)^VCT(8) + sqrt(VCT(7))
ans =
    134
>> MAT = [3 11 6 5; 4 7 10 2; 13 9 0 8]
MAT =
    3   11   6   5
    4   7   10   2
    13   9   0   8
>> MAT(3,1) = 20
ans =
    3   11   6   5
    4   7   10   2
    20   9   0   8
>> MAT(2,4) - MAT(1,2)
ans =
    3   11   6   5
    4   7   10   2
    20   9   0   8
```

Define um vetor.

Atribui novos valores ao quarto e ao sexto elementos.

Usa elementos do vetor em uma expressão matemática.

Define uma matriz.

Atribui novos valores ao elemento (3,1).

Usa elementos da matriz em uma expressão matemática.

```
ans =
-9
```

### Uso de dois pontos (:) no endereçamento de arranjos

O símbolo de dois pontos pode ser usado no endereçamento de um grande número de elementos em um vetor ou em uma matriz. Em um vetor,  $va(:)$  se refere a todos os elementos do vetor  $va$  (seja ele um vetor linha ou coluna).  $va(m:n)$  se refere aos elementos  $m$  a  $n$  do vetor  $va$ .

Em uma matriz,  $A(:,n)$  se refere aos elementos em todas as linhas da coluna  $n$ .  $A(n,:)$  se refere aos elementos em todas as colunas da linha  $n$ .  $A(:,m:n)$  se refere aos elementos em todas as linhas entre as colunas  $m$  e  $n$ .  $A(m:n,:)$  se refere aos elementos em todas as colunas entre as linhas  $m$  e  $n$ .  $A(m:n,p:q)$  se refere aos elementos em todas as linhas  $m$  a  $n$  e colunas  $p$  a  $q$ .

```
>> v = [4 15 8 12 34 2 50 23 11] Define um vetor.
v =
    4   15   8   12   34   2   50   23   11
>> u = v(3:7) O vetor u é criado a partir dos elementos 3 a 7 do vetor v.
u =
    8   12   34   2   50
>> A = [1 3 5 7 9 11; 2 4 6 8 10 12; 3 6 9 12 15 18; 4 8 12 16 20 24; 5 10 15
20 25 30] Define uma matriz.
A =
    1   3   5   7   9   11
    2   4   6   8   10  12
    3   6   9   12  15  18
    4   8   12  16  20  24
    5  10  15  20  25  30
C = A(2,:)
C = O vetor C é criado a partir da segunda linha da matriz A.
    2   4   6   8   10  12
>> F = A(1:3,2:4) A matriz F é criada a partir dos elementos nas linhas 1 a 3 e colunas 2 a 4 da matriz A.
F =
    3   5   7
    4   6   8
    6   9   12
```

O MATLAB possui várias funções residentes dedicadas ao manuseio de arranjos. Várias delas estão listadas na Tabela A-4.

**Tabela A-4** Funções residentes dedicadas ao manuseio de arranjos

Comando	Descrição	Exemplo
<code>length(A)</code>	Retorna o número de elementos no vetor A	$\gg A = [5 9 2 4];$ $\gg length(A)$ $ans =$ 4

**Tabela A-4** Funções residentes dedicadas ao manuseio de arranjos (*continuação*)

Comando	Descrição	Exemplo
<code>size(A)</code>	Retorna um vetor linha $[m, n]$ , onde $m$ e $n$ são o tamanho $m \times n$ do arranjo $A$ ( $m$ é o número de linhas, $n$ é o número de colunas)	<pre>&gt;&gt; A = [6 1 4 0 12; 5 19 6 8 2] A =     6   1   4   0   12     5   19   6   8   2 &gt;&gt; size(A) ans =     2   5</pre>
<code>zeros(m, n)</code>	Cria uma matriz com $m$ linhas e $n$ colunas na qual todos os elementos são o número 0	<pre>&gt;&gt; zr = zeros(3,4) zr =     0   0   0   0     0   0   0   0     0   0   0   0</pre>
<code>ones(m, n)</code>	Cria uma matriz com $m$ linhas e $n$ colunas na qual todos os elementos são o número 1	<pre>&gt;&gt; ne = ones(4,3) ne =     1   1   1     1   1   1     1   1   1     1   1   1</pre>
<code>eye(n)</code>	Cria uma matriz quadrada com $n$ linhas e $n$ colunas na qual os elementos diagonais são iguais a 1 (matriz identidade)	<pre>&gt;&gt; idn = eye(5) idn =     1   0   0     0   1   0     0   0   1</pre>

### Strings

- *Strings* são arranjos criados com a digitação de caracteres entre aspas simples.
- *Strings* podem incluir letras, números, outros símbolos e espaços.
- Exemplos de *strings*: 'ad ef', '3%fr2', '{edcba:21!}'.
- Na criação de *strings*, a cor do texto na tela fica marrom quando a primeira aspa é digitada. Quando a aspa final é digitada, a cor muda para púrpura.

*Strings* encontram muitas aplicações diferentes no MATLAB, como por exemplo na exibição de mensagens de texto em comandos de saída, na formatação de comandos gráficos e como argumentos de entrada em algumas funções. Variáveis também podem receber *strings*, o que é feito simplesmente com a digitação do texto desejado no lado direito do operador de atribuição, conforme mostrado no exemplo a seguir.

```
>> a = 'FRty 8'
a =
FRty 8
>> B = 'Meu nome é João da Silva'
B =
Meu nome é João da Silva
```

## A.4 OPERAÇÕES MATEMÁTICAS COM ARRANJOS

Uma vez criadas no MATLAB, as variáveis podem ser usadas em uma ampla variedade de operações matemáticas. Tais operações podem ser divididas em três categorias:

1. Operações com escalares (arranjos  $(1 \times 1)$ ) e com elementos isolados de arranjos.
2. Operações com arranjos seguindo as regras da álgebra linear.
3. Operações com arranjos elemento por elemento.

Operações com escalares e elementos isolados de arranjos são feitas usando os símbolos padrão como em uma calculadora. Até o momento, todas as operações matemáticas apresentadas neste apêndice têm sido feitas dessa forma.

### Soma e subtração de arranjos

Com arranjos, as operações de soma, subtração e multiplicação seguem as regras da álgebra linear (ver Capítulo 2). As operações  $+$  (adição) e  $-$  (subtração) podem ser realizadas apenas com arranjos de mesmo tamanho (com o mesmo número de linhas e colunas). A soma ou a diferença de arranjos é obtida com a soma ou subtração de seus elementos correspondentes.

Em geral, se  $A$  e  $B$  são dois arranjos (por exemplo, matrizes  $(2 \times 3)$ ),

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}$$

então, a matriz obtida com a soma de  $A$  e  $B$  é:

$$\begin{bmatrix} (A_{11} + B_{11}) & (A_{12} + B_{12}) & (A_{13} + B_{13}) \\ (A_{21} + B_{21}) & (A_{22} + B_{22}) & (A_{23} + B_{23}) \end{bmatrix}$$

No MATLAB, quando um escalar é somado ou subtraído de um arranjo, esse número é somado ou subtraído de todos os elementos do arranjo (note que esta operação não é definida na álgebra linear). Exemplos são:

```
>> VA = [8 5 4]; VB = [10 2 7];
>> VC = VA + VB
VC =
    18    7   11
Define dois vetores VA e VB.
Define um vetor VC que é igual a VA+VB.

>> A = [5 -3 8; 9 2 10], B = [10 7 4; -11 15 1]
A =
    5   -3    8
    9     2   10
B =
    10    7    4
   -11   15    1
Define duas matrizes A e B.

>> C = A + B
Define uma matriz C que é igual a A+B.
```

```
C =
15   4   12
-2   17  11
>> C - 8
ans =
 7   -4   4
-10   9   3
```

Subtrai 8 da matriz C.

8 é subtraído de cada elemento de C.

### Multiplicação de arranjos

A operação de multiplicação \* é executada pelo MATLAB de acordo com as regras da álgebra linear (ver Seção 2.4.1). Isso significa que, se  $A$  e  $B$  são duas matrizes, a operação  $A*B$  pode ser realizada apenas se o número de colunas da matriz  $A$  for igual ao número de linhas da matriz  $B$ . O resultado é uma matriz que tem o mesmo número de linhas de  $A$  e o mesmo número de colunas de  $B$ . Por exemplo, se  $A$  é uma matriz  $(3 \times 2)$  e  $B$  é uma matriz  $(2 \times 4)$ , então a operação  $C=A*B$  tem como resultado uma matriz  $(3 \times 4)$ :

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

Define duas matrizes A e B.

```
A =
2   -1
8    3
6    7
```

```
B =
4   9   1   -3
-5   2   4   6
```

```
>> C = A*B
```

Multiplica A\*B.

```
C =
13   16   -2   -12
17   78   20   -6
-11   68   34   24
```

C é uma matriz  $(3 \times 4)$ .

Dois vetores podem ser multiplicados apenas se ambos tiverem o mesmo número de elementos e um deles for um vetor linha e o outro for um vetor coluna. A multiplicação de um vetor linha por um vetor coluna resulta em uma matriz  $(1 \times 1)$ , que é um escalar. Este é o produto escalar de dois vetores (o MATLAB também possui uma função residente, denominada `dot(a, b)`, que calcula o produto escalar de dois vetores). Quando a função `dot` é utilizada, os vetores  $a$  e  $b$  podem ser cada um deles um vetor linha ou coluna. A multiplicação de um vetor coluna por um vetor linha, ambos com  $n$  elementos, fornece como resultado uma matriz  $(n \times n)$ .

```
>> AV = [2 5 1]
```

Define o vetor linha com três elementos AV.

```
AV =
2   5   1
```

```
>> BV = [3; 1; 4]
```

Define o vetor linha com três elementos BV.

```
BV =
```

```
3
```

```
1
```

```
4
```

```
>> AV * BV
```

Multiplica AV por BV. A resposta é um escalar (produto escalar de dois vetores).

```
ans =
```

```
15
```

```
>> BV * AV
```

Multiplica BV por AV. A resposta é uma matriz (3 × 3) (Produto vetorial de dois vetores).

```
ans =
```

```
6 15 3
2 5 1
8 20 4
```

## Divisão de arranjos

A operação de divisão no MATLAB está associada à solução de um sistema de equações lineares. O MATLAB possui dois tipos de divisão de arranjos, que são a divisão à esquerda e a divisão à direita. As duas operações são explicadas na Seção 4.8.1. Note que a divisão à esquerda *não é* uma operação definida na álgebra linear (ver Seção 2.4.1). A operação de divisão no MATLAB realiza o equivalente à multiplicação de uma matriz pela inversa de outra matriz (ou vice-versa).

### Operações elemento por elemento

Operações elemento por elemento são realizadas em cada elemento do arranjo (ou dos arranjos). A soma e a subtração já são, por definição, operações elemento por elemento porque, quando dois arranjos são somados (ou subtraídos), a operação é realizada em cada elemento do arranjo. Quando dois ou mais arranjos estão envolvidos na mesma expressão, operações elemento por elemento podem ser feitas apenas com arranjos de mesmo tamanho.

A multiplicação, a divisão e a exponenciação de dois vetores ou matrizes são realizadas no MATLAB elemento por elemento quando um ponto é digitado na frente do operador aritmético.

Símbolo	Descrição	Símbolo	Descrição
$.$ *	Multiplicação	$.$ /	Divisão à direita
$.$ <sup>^</sup>	Exponenciação	$\backslash$	Divisão à esquerda

Se dois vetores  $a$  e  $b$  são  $a = [a_1 \ a_2 \ a_3 \ a_4]$  e  $b = [b_1 \ b_2 \ b_3 \ b_4]$ , então a multiplicação, a divisão e a exponenciação de dois vetores elemento por elemento são:

$$a . * b = \begin{bmatrix} a_1 b_1 & a_2 b_2 & a_3 b_3 & a_4 b_4 \end{bmatrix}$$

$$a . / b = \begin{bmatrix} a_1 / b_1 & a_2 / b_2 & a_3 / b_3 & a_4 / b_4 \end{bmatrix}$$

$$a . ^ b = \begin{bmatrix} (a_1)^{b_1} & (a_2)^{b_2} & (a_3)^{b_3} & (a_4)^{b_4} \end{bmatrix}$$

Se duas matrizes  $A$  e  $B$  são:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

então a multiplicação e a divisão dessas duas matrizes elemento por elemento resultam em:

$$A .* B = \begin{bmatrix} A_{11}B_{11} & A_{12}B_{12} & A_{13}B_{13} \\ A_{21}B_{21} & A_{22}B_{22} & A_{23}B_{23} \\ A_{31}B_{31} & A_{32}B_{32} & A_{33}B_{33} \end{bmatrix} \quad A ./ B = \begin{bmatrix} A_{11}/B_{11} & A_{12}/B_{12} & A_{13}/B_{13} \\ A_{21}/B_{21} & A_{22}/B_{22} & A_{23}/B_{23} \\ A_{31}/B_{31} & A_{32}/B_{32} & A_{33}/B_{33} \end{bmatrix}$$

Exemplos de operações elemento por elemento no MATLAB são:

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

Define uma matriz A ( $2 \times 3$ ).

A =

2	6	3
5	8	4

```
>> B = [1 4 10; 3 2 7]
```

Define uma matriz B ( $2 \times 3$ ).

B =

1	4	10
3	2	7

```
>> A .* B
```

Multiplicação dos arranjos A e B elemento por elemento.

ans =

2	24	30
15	16	28

```
>> C = A ./ B
```

Divisão dos arranjos A e B elemento por elemento.

C =

2.0000	1.5000	0.3000
1.6667	4.0000	0.5714

```
>> B .^ 3
```

Exponenciação do arranjo B elemento por elemento.

ans =

1	64	1000
27	8	343

Operações elemento por elemento são muito úteis no cálculo do valor de uma função em muitos valores de seu argumento. Isso é feito definindo-se primeiro um vetor que contenha valores da variável independente. Esse vetor é então usado em cálculos elemento por elemento para criar um vetor em que cada elemento é o valor correspondente da função. Por exemplo, o cálculo de  $y = \frac{z^3 + 5z}{4z^2 - 10}$  em oito valores de  $z$ ,  $z = 1, 3, 5, \dots, 15$ , é feito da seguinte maneira:

```
>> z = [1:2:15]                                     Define um vetor z com oito elementos.
z =
    1   3   5   7   9   11  13  15
>> y = (z.^3 + 5*z)./(4*z.^2 - 10)]           O vetor z é usado no cálculo dos elementos
y =                                              do vetor y elemento por elemento.
   -1.0000   1.6154   1.6667   2.0323   2.4650   2.9241   3.3964   3.8764
```

No último exemplo, operações elemento por elemento foram utilizadas três vezes; para calcular  $z^3$  e  $z^2$ , e para dividir o numerador pelo denominador.

O MATLAB possui muitas funções residentes dedicadas às operações com arranjos. Várias delas estão listadas na Tabela A-5.

**Tabela A-5** Funções residentes dedicadas ao manuseio de arranjos

Comando	Descrição	Exemplo
mean (A)	Se A é um vetor, a função retorna o valor médio de seus elementos	>> A = [5 9 2 4]; >> mean(A) ans = 5
sum (A)	Se A é um vetor, a função retorna a soma dos elementos do vetor	>> A = [5 9 2 4]; >> sum(A) ans = 20
sort (A)	Se A é um vetor, a função arranja os elementos do vetor em ordem ascendente	>> A = [5 9 2 4]; >> sort(A) ans = 2   4   5   9
det (A)	A função retorna o determinante da matriz quadrada A	>> A = [2 4; 3 5]; >> det(A) ans = -2

## A.5 ARQUIVOS DE PROGRAMA

Arquivos de programa (*script files*) são arquivos que contêm uma seqüência de comandos do MATLAB. Quando um arquivo de programa é rodado, o MATLAB executa os comandos na ordem em que eles são escritos, como se eles tivessem sido digitados na janela de comandos. Quando um comando gera uma saída (por exemplo, a atribuição de um valor a uma variável sem que um ponto-e-vírgula seja usado no final do comando), a saída é exibida na janela de comandos. O uso de arquivos de programa é conveniente porque eles podem ser armazenados, editados mais tarde (corrigidos e/ou alterados), e executados muitas vezes. Arquivos de programa podem ser digitados e editados em qualquer editor de texto e então copiados para o editor do MATLAB. Arquivos de programa são também chamados de arquivos M (em inglês, *M-files*) porque a extensão .m é utilizada quando eles são salvos.

### Criando e salvando um arquivo de programa

Arquivos de programa são criados e editados na janela de edição e verificação de erros (Editor/Debugger Window). Essa janela é aberta a partir da janela de comandos do MATLAB. No menu **File**, selecione *New*, e depois escolha **M-file**. Assim que a janela é aberta, os comandos podem ser digitados linha por linha. O MATLAB numera cada linha automaticamente cada vez que a tecla **Enter** é pressionada. Os comandos também podem ser digitados em qualquer editor de texto e então copiados e colados na janela de edição.

Antes de se executar um arquivo de programa, ele deve ser salvo. Isso é feito com a seleção da opção *Save As...* no menu **File**, com a determinação de uma localização (pasta) e finalmente com a escolha de um nome para o arquivo. As regras usadas na denominação de arquivos de programa seguem as regras usadas na denominação de variáveis (devem começar com uma letra, podem incluir números e o símbolo `_`, e ter comprimento máximo de 63 caracteres). Os nomes de variáveis criadas pelo usuário, de variáveis predefinidas e de comandos ou funções do MATLAB não devem ser usados para denominar arquivos de programa.

Um arquivo de programa pode ser executado digitando-se o seu nome na janela de comandos e pressionando-se a tecla **Enter**, ou diretamente a partir da janela de edição clicando-se no ícone **Run**. Antes de fazer isso, no entanto, o usuário deve se assegurar de que o MATLAB possa encontrar o arquivo (isto é, que o MATLAB saiba onde o arquivo está salvo). Para rodar um arquivo, este deve estar no diretório corrente ou no caminho de busca.

O diretório corrente é mostrado no campo “Current Directory” na barra de ferramentas da janela de comandos. O diretório corrente pode ser alterado na janela de diretório corrente (Current Directory Window).

Quando o MATLAB é solicitado para rodar um arquivo ou executar uma função, ele procura o arquivo nos diretórios listados no caminho de busca. Os diretórios incluídos no caminho de busca são mostrados na janela de determinação de caminho (*Set Path Window*), que pode ser aberta com a seleção de **Set Path** no menu **File**. Uma vez aberta a janela de determinação de caminho, novos arquivos podem ser adicionados ou removidos do caminho de busca.

### Variáveis globais

Variáveis globais são variáveis que, uma vez criadas em uma parte do MATLAB, são reconhecidas em outras partes do MATLAB. Esse é o caso das variáveis declaradas na janela de comandos e em arquivos de programa, pois ambos operam na área de trabalho. Quando uma variável é definida na janela de comandos, ela também é reconhecida e pode ser usada em um arquivo de programa. Da mesma forma, se uma variável for definida em um arquivo de programa, ela também é reconhecida e pode ser usada na janela de comandos. Em outras palavras, uma vez criada uma variável, ela existe, pode ser usada e pode receber um novo valor tanto na janela de comandos quanto em um arquivo de programa (há diferentes tipos de arquivos no MATLAB, chamados de arquivos de função, que normalmente não compartilham sua variáveis com outras partes do programa. Isso é explicado na Seção A.6).

## Entrada de um arquivo de programa

Quando um arquivo de programa é executado, as variáveis usadas nos cálculos dentro desse arquivo devem receber valores. A atribuição de um valor a uma variável pode ser feita de três maneiras, dependendo de onde e de como a variável é definida. Uma opção é defini-la e atribuir-lhe um valor dentro do próprio arquivo de programa. Neste caso, a atribuição do valor à variável faz parte do arquivo de programa. Se o usuário desejar rodar o arquivo atribuindo um diferente valor a essa variável, o arquivo deve ser editado e a atribuição da variável deve ser alterada. Em seguida, após ser salvo, o arquivo pode ser executado novamente.

Uma segunda opção é definir a variável e atribuir-lhe um valor na janela de comandos. Nesse caso, se o usuário desejar rodar o arquivo de programa considerando um diferente valor para a variável, o novo valor deve ser carregado na janela de comandos e o arquivo, executado novamente.

A terceira opção é definir a variável no arquivo de programa mas atribuir-lhe um valor específico na janela de comandos quando o programa for rodado. Isso é feito usando o comando `input` para criar a variável.

## Saída de um arquivo de programa

Conforme discutido anteriormente, o MATLAB exibe resultados automaticamente quando determinados comandos são executados. Por exemplo, quando um valor é carregado em uma variável, ou quando o nome de uma variável já definida é digitado e a tecla **Enter** é pressionada, o MATLAB exibe a variável e seu valor. O MATLAB tem vários comandos que podem ser usados para exibir resultados. Resultados podem ser mensagens informativas, dados numéricos e gráficos. `disp` e `fprintf` são dois comandos freqüentemente usados para mostrar resultados de saída. O comando `disp` mostra a saída na tela, enquanto o comando `fprintf` pode ser usado para mostrar a saída na tela ou salvá-la em arquivo.

O comando `disp` é usado para mostrar os elementos de uma variável sem mostrar o nome dessa variável, e para exibir mensagens de texto. O formato do comando `disp` é:

```
disp(nome de uma variável) ou disp('texto como string')
```

Sempre que o comando `disp` é executado, os resultados que ele mostra aparecem em uma nova linha.

O comando `fprintf` pode ser usado para exibir resultados de saída (texto ou dados) na tela ou para salvá-los em arquivo. Com esse comando, a saída pode ser formatada. Por exemplo, textos e valores numéricos de variáveis podem ser misturados e mostrados em uma mesma linha. Além disso, o formato dos números pode ser controlado. Para exibir texto e o valor de uma variável em uma mesma linha, o comando `fprintf` tem a forma:

```
fprintf('texto como string %-5.2f texto adicional', nome_da_variável)
```

O sinal % marca o ponto onde o número deve ser inserido no texto

Formata os elementos (define o formato do número)

Nome da variável cujo valor é mostrado

## A.6 ARQUIVOS DE FUNÇÃO

Um arquivo de função é um programa de MATLAB usado como uma função. Ele possui argumentos de entrada e saída que podem conter uma ou muitas variáveis, e cada uma delas pode ser um escalar, um vetor, ou um arranjo de qualquer tamanho. Esquematicamente, um arquivo de função pode ser ilustrado como:



Arquivos de função podem ser usados como funções matemáticas e sub-programas em programas maiores. Dessa forma, programas grandes podem ser construídos como a soma de “blocos construtivos” que podem ser testados de forma independente. O MATLAB possui funções residentes que realizam todas as funções matemáticas padrão e muitas operações matemáticas (por exemplo, a solução de equações não-lineares, o ajuste de curvas, etc.). O usuário pode criar funções que podem ser usadas para qualquer propósito.

Arquivos de função são criados e editados, como os arquivos de programa, na janela de edição e verificação de erros. A primeira linha executável de um arquivo de função deve ser a linha de definição da função, que tem a forma:

```
function [argumentos_de_saída] = nome_da_função(argumentos_de_entrada)
```

A palavra `function` deve ser a primeira palavra, e deve ser digitada em letras minúsculas

Lista de argumentos de saída, digitada entre colchetes

Nome da função

Lista de argumentos de entrada digitados entre parênteses e separados por vírgulas

A palavra “`function`”, digitada em letras minúsculas, deve ser a primeira palavra na linha de definição da função. Os argumentos de entrada e saída são usados para transferir dados para dentro e para fora da função. Os argumentos de entrada são listados entre parênteses após o nome da função. Usualmente, há pelo menos um argumento de entrada. Se houver mais de um, os argumentos devem ser separados por vírgulas. O código que realiza os cálculos no interior do arquivo da função é escrito em termos dos argumentos de entrada e assume que a esses argumentos sejam atribuídos valores numéricos.

Os argumentos de saída, que são listados entre colchetes no lado esquerdo da linha de definição da função, fazem a transferência da saída da função escrita em arquivo. Arquivos de função podem ter um, vários ou nenhum argumento de

saída. Se houver mais de um, os argumentos de saída são separados por vírgulas. Se houver apenas um argumento de saída, ele pode ser digitado sem colchetes. Para que o arquivo de função funcione, os argumentos de saída devem receber valores no programa de computador escrito no corpo da função.

Após a linha de definição da função, há usualmente várias linhas de comentários. Elas são opcionais, porém freqüentemente usadas para fornecer informações sobre a função. Em seguida, a função contém o programa de computador (código) que efetivamente realiza os cálculos. O código da função pode usar todas as facilidades de programação do MATLAB, incluindo cálculos, atribuição de valores, funções residentes ou criadas pelo usuário e controles de fluxo (declarações condicionais e laços de repetição; ver Seção A.7).

Todas as variáveis presentes em um arquivo de função são locais. Isso significa que as variáveis são definidas e reconhecidas apenas no interior da função. Quando um arquivo de função é executado, o MATLAB utiliza uma área de memória que é separada da área de trabalho (espaço de memória da janela de comandos e dos arquivos de programa). Em um arquivo de função, as variáveis de entrada recebem valores cada vez que a função é chamada. Essas variáveis são então usadas nos cálculos no interior do arquivo da função. Assim, o arquivo de uma função pode ter variáveis com os mesmos nomes das variáveis utilizadas na janela de comandos ou nos arquivos de programa. Uma função em arquivo não reconhece variáveis homônimas que tenham recebido valores fora da função. A atribuição de valores a essas variáveis no interior do arquivo da função não muda os seus valores fora da função.

```
function[mpag,tpag]=loan(montante,taxa,anos)
% loan calcula o pagamento mensal e total de uma hipoteca
% Argumentos de entrada:
% montante: montante total do empréstimo em R$
% taxa: taxa percentual de juros anual
% anos: número de anos
% Argumentos de saída:
% mpag: parcela mensal
% tpag: pagamento total

format bank
taxam=taxa*0.01/12;
a = 1 + taxam;
b = (anos*12)-1)/taxam;
mpag=montante*a^(anos*12)/(a*b);
tpag=mpag*anos*12;
```

Linha de definição da função.

Atribui valores às variáveis de saída.

Uma função simples, denominada `loan`, que calcula o pagamento mensal e total de uma hipoteca para um dado valor de empréstimo, uma certa taxa de juros e uma duração específica, é listada a seguir:

```
>> [mensal total] = loan(25000,7.5,4)
mensal =
    600.72
total =
28834.47
```

## A.7 PROGRAMAÇÃO NO MATLAB

Um programa de computador é formado por uma seqüência de comandos. Em um programa simples, comandos são executados um após o outro na ordem em que são digitados. Muitas situações, contudo, requerem programas mais sofisticados nos quais diferentes comandos (ou grupos de comandos) são executados quando o programa é rodado com diferentes variáveis de entrada. Em outras situações, pode ser necessário repetir muitas vezes uma seqüência de comandos no interior de um programa. Por exemplo, programas que resolvem equações numericamente repetem uma seqüência de cálculos até que o erro na resposta seja menor que um determinado valor.

O MATLAB possui várias ferramentas que podem ser utilizadas para controlar o fluxo de um programa. Declarações condicionais possibilitam saltar comandos ou executar grupos de comandos específicos em diferentes situações. Laços de repetição `for` e `while` permitem repetir várias vezes uma seqüência de comandos.

A mudança no fluxo de um programa requer algum tipo de processo de tomada de decisões. O computador deve decidir entre executar o próximo comando e saltar um ou mais comandos e continuar em uma diferente linha dentro do programa. O programa toma essas decisões comparando valores de variáveis. Isso é feito usando operadores lógicos e relacionais.

### A.7.1 Operadores lógicos e relacionais

Operadores lógicos e relacionais são usados em combinação com outros comandos para permitir a tomada de decisões que controlem o fluxo de um programa de computador. Um operador relacional compara dois números determinando se uma declaração de comparação é verdadeira ou falsa (por exemplo,  $5 < 8$ ). Se a declaração for verdadeira, ele recebe o valor 1. Se ela for falsa, ele recebe um 0. Operadores relacionais utilizados pelo MATLAB são fornecidos na tabela a seguir.

Operador relacional	Descrição	Operador relacional	Descrição
<code>&lt;</code>	Menor que	<code>&gt;=</code>	Menor ou igual a
<code>&gt;</code>	Maior que	<code>==</code>	Igual a
<code>&lt;=</code>	Menor ou igual a	<code>~=</code>	Diferente de

Note que o operador relacional “igual a” consiste em dois sinais de  $=$  (sem espaço entre eles), já que um sinal de  $=$  é o operador de atribuição de valores. Da mesma forma, em outros operadores relacionais formados por dois caracteres, não há espaço entre os caracteres ( $<=$ ,  $>=$ ,  $\sim=$ ). Dois exemplos são:

```
>> 5 > 8
ans =
0
>> 4 == 6
ans =
0
```

Um operador lógico examina declarações verdadeiro/falso e produz um resultado verdadeiro (1) ou falso (0) de acordo com o operador específico. Operadores lógicos utilizados no MATLAB são:

Operador lógico	Nome	Descrição
$\&$ Exemplo: A&B	AND	Atua em dois operandos (A e B). Se ambos forem verdadeiros, o resultado será verdadeiro (1); senão, o resultado será falso (0).
$ $ Exemplo: A B	OR	Atua em dois operandos (A e B). Se qualquer um deles ou ambos forem verdadeiros, o resultado será verdadeiro (1); senão (ambos forem falsos), o resultado será falso (0).
$\sim$ Exemplo: $\sim A$	NOT	Atua em um operando (A). Fornece o oposto do operando. Verdadeiro (1) se o operando for falso, e falso (0) se o operando for verdadeiro.

Operadores lógicos podem ter números como operandos. Um número diferente de zero é verdadeiro, e um zero é falso. Vários exemplos são:

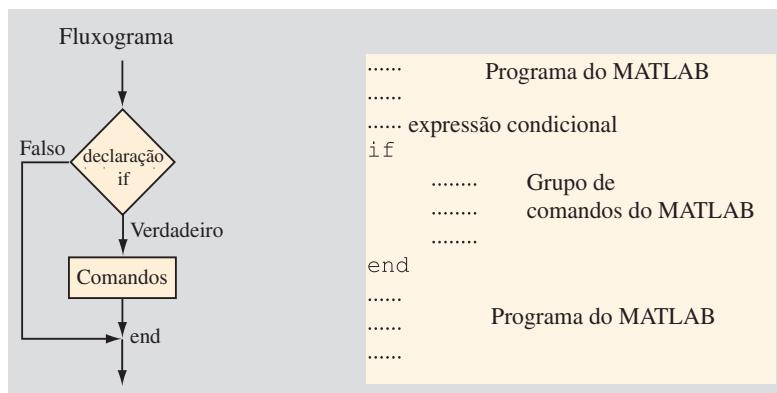
```
>> 3&7
ans =
1
3 e 7 são ambos verdadeiros (diferentes de zero), então a saída é 1.
>> a = 5|0
a =
1
a recebe 1, já que pelo menos um número é verdadeiro (diferente de zero).
>> ~25
ans =
0
```

### A.7.2 Declarações condicionais, estruturas if-else

Declaração condicional é o comando que permite ao MATLAB escolher entre a execução de um grupo de comandos logo após a declaração ou saltar esses comandos. Essa declaração envolve uma expressão condicional. Se a expressão for verdadeira, executa-se um grupo de comandos logo após a declaração. Se a expressão for falsa, o computador salta essa grupo de comandos.

#### A estrutura if-end

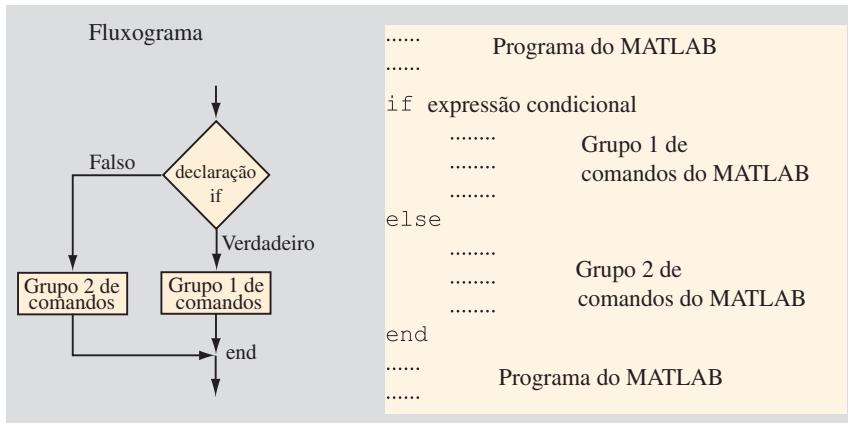
A forma mais simples de uma declaração condicional é a estrutura `if-end`, mostrada esquematicamente na Fig. A-1. A figura mostra como os comandos são digitados no programa e apresenta um fluxograma que mostra simbolicamente o fluxo, ou a sequência, na qual os comandos são executados. Quando o programa é executado, ele atinge a declaração `if`. Se a expressão condicional na declaração `if` for verdadeira (1), o programa continua a executar os comandos que sucedem essa declaração até encontrar `end`. Se a expressão condicional for falsa (0), o programa salta o grupo de comandos entre o `if` e o `end`, e continua com os comandos que sucedem a declaração `end`.



**Figura A-1** Estrutura da declaração condicional `if-end`.

#### A estrutura if-else-end

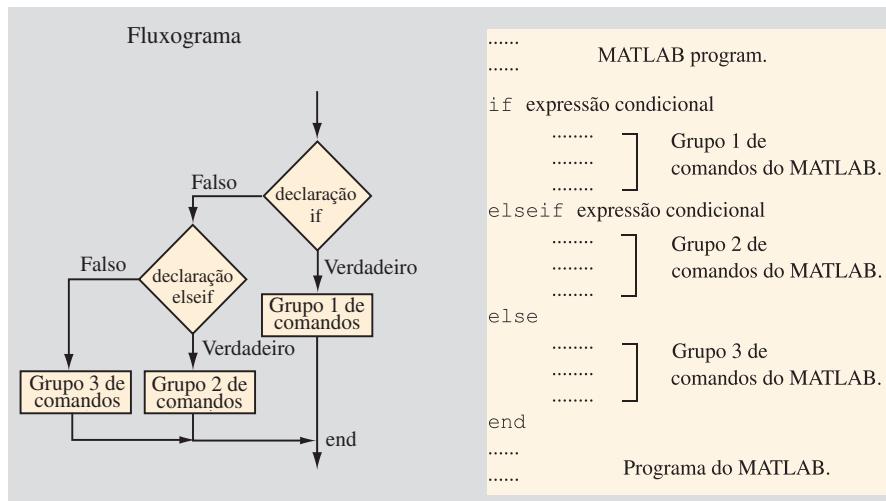
A estrutura `if-else-end` permite escolher a execução de um grupo de comandos a partir de dois grupos possíveis (ver Fig. A-2). A primeira linha é uma declaração `if` com uma expressão condicional. Se a expressão condicional for verdadeira, o programa executa o grupo 1 de comandos entre as declarações `if` e `else`, e então salta para o final. Se a expressão condicional for falsa, o programa pula para a declaração `else` e executa os comandos pertencentes ao grupo 2, entre as declarações `else` e `end`.



**Figura A-2** Estrutura da declaração condicional if-else-end.

### A estrutura if-elseif-else-end

A estrutura if-elseif-else-end é mostrada na Fig. A-3. Essa estrutura inclui duas declarações condicionais (if e elseif) que possibilitam a seleção de um entre três grupos de comandos para a execução. A primeira linha é uma declaração if com uma expressão condicional. Se a expressão condicional for verdadeira, o programa executa o grupo 1 de comandos entre as declarações if e elseif e então salta para end. Se a expressão condicional na declaração if for falsa, o programa salta para a declaração elseif. Se a expressão condicional na declaração elseif for verdadeira, o programa executa o grupo 2 de comandos entre as declarações elseif e else e então salta para end. Se a expressão condicional na declaração elseif for falsa, o programa salta para a declaração else e executa o grupo 3 de comandos entre as declarações else e end.



**Figura A-3** Estrutura da declaração condicional if-elseif-else-end.

Várias declarações elseif e grupos de comandos associados podem ser adicionados. Dessa forma, mais condições podem ser incluídas. Além disso, a declaração else é opcional. Isso significa que, no caso de várias declarações elseif e nenhuma declaração else, se qualquer uma das declarações condicionais for verdadeira, os comandos associados são executados. Do contrário, nada é executado.

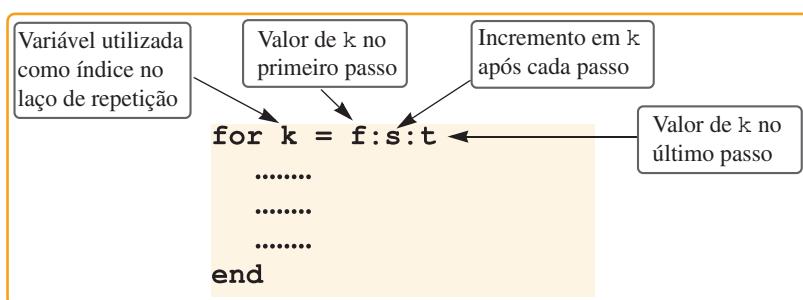
Em geral, uma mesma tarefa pode ser realizada com o uso de várias declarações elseif ou if-else-end. O uso da última é melhor por facilitar a compreensão, a modificação e a identificação de erros no programa.

### A.7.3 Laços de repetição (loops)

Um laço de repetição, ou *loop*, permite a alteração do fluxo de um programa de computador. Em um laço de repetição, a execução de um comando, ou de um grupo de comandos, é repetida várias vezes de forma consecutiva. Cada rodada de execução é chamada de passo. Em cada passo, pelo menos uma variável definida no interior do laço de repetição recebe um novo valor.

#### Laços for-end

Em laços de repetição for-end, a execução de um comando ou de um grupo de comandos é repetida um número predeterminado de vezes. A forma do laço de repetição é mostrada na Fig. A-4. No primeiro passo,  $k = f$ , e o computador executa os comandos entre for e end. Em seguida, o programa retorna para o comando for e inicia o segundo passo.  $k$  recebe um novo valor  $k = f+s$  e os comandos entre for e end são executados com esse novo valor de  $k$ . O processo se repete até o último passo, onde  $k = t$ . Após isso, o programa não volta para o for; ele continua passando pelos comandos que sucedem o comando end. Por exemplo, se  $k = 1:2:9$ , então o laço de repetição é realizado cinco vezes, e  $k$  recebe os valores 1, 3, 5, 7 e 9 em cada passo. Se o valor do incremento  $s$  for omitido, seu valor é 1 (*default*) (isto é,  $k = 3:7$  produz cinco passos com  $k = 3, 4, 5, 6, 7$ ).



**Figura A-4** Estrutura de um laço de repetição for-end.

Um programa que ilustra o uso de declarações condicionais e laços de repetição é mostrado a seguir (arquivo de programa). O programa muda os elementos de um dado vetor de tal forma que os elementos positivos e divisíveis por 3 e/ou 5

sejam multiplicados por 2. Elementos negativos e maiores que -5 são elevados à potência de 3; os demais elementos permanecem inalterados.

```
V = [5, 17, -3, 8, 0, -7, 12, 15, 20 -6, 6, 4, -2, 16];
```

```
n = length(V);
```

for k = 1:n

No  $k$ -ésimo passo do laço de repetição, o  $k$ -ésimo elemento é verificado e modificado, se necessário.

```
  if V(k) > 0 & (rem(V(k),3) == 0 | rem(V(k),5) == 0)
```

```
    V(k) = 2*V(k);
```

```
  elseif V(k) < 0 & V(k) > -5
```

```
    V(k) = V(k)^3;
```

```
  end
```

```
end
```

```
V
```

Quando o programa é executado, o vetor V a seguir é mostrado na janela de comandos do MATLAB:

```
V =
```

```
10 17 -27 8 0 -7 24 30 40 -6 12 4 -8 16
```

## A.8 TRAÇADO DE GRÁFICOS

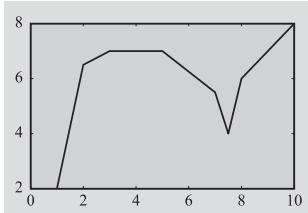
O MATLAB possui muitos comandos que podem ser usados para criar diferentes tipos de gráficos. Estes incluem gráficos padrão com eixos lineares, gráficos com eixos logarítmicos, gráficos de barras e escadas, gráficos polares, e muito mais. Os gráficos podem ser formatados para ter a aparência desejada.

Gráficos bidimensionais podem ser criados com o comando `plot`. A forma mais simples do comando é:

```
plot(x,y)
```

Os argumentos `x` e `y` são vetores (arranjos unidimensionais) que devem ter o mesmo número de elementos. Quando o comando `plot` é executado, a janela de figuras é aberta automaticamente, exibindo a figura gerada. A figura tem uma única curva com os valores de `x` na abscissa (eixo horizontal) e os valores de `y` na ordenada (eixo vertical). A curva é formada por segmentos de reta que conectam os pontos cujas coordenadas são definidas pelos elementos dos vetores `x` e `y`. Os vetores, é claro, podem ter qualquer nome. O primeiro vetor digitado no comando `plot` é usado no eixo horizontal, e o segundo vetor é usado no eixo vertical. A figura mostrada tem eixos com escala linear e intervalo de dados padrão. Por exemplo, se um vetor `x` tiver elementos 1, 2, 3, 5, 7, 7,5, 8, 10, e um vetor `y` tiver elementos 2, 6,5, 7, 7, 5,5, 4, 6, 8, um gráfico simples de `y versus x` pode ser produzido digitando-se a seguinte seqüência na janela de comandos:

```
>> x = [1 2 3 5 7 7.5 8 10];
>> y = [2 6.5 7 7 5.5 4 6 8];
>> plot(x,y)
```



**Figura A-5** Gráfico gerado a partir de pontos de dados.

Uma vez executado o comando `plot`, o gráfico mostrado na Fig. A-5 aparece na janela de figuras.

O comando `plot` possui argumentos opcionais que podem ser usados para especificar a cor e o estilo da linha, bem como a cor e o tipo dos marcadores, se algum for desejado. Com essas opções, o comando tem a forma:

```
plot(x,y, 'especificadores de linha', )
```

Especificadores de linha podem ser usados para definir o estilo e a cor da linha, e o tipo de marcadores (se marcadores forem desejados). Os especificadores de estilo de linha são:

Estilo de linha	Especificador	Estilo de linha	Especificador
sólido (padrão)	-	pontuado	:
traço	--	traço-ponto	-. .

Os especificadores de cor de linha são:

Cor de linha	Especificador						
vermelha	r	azul	b	magenta	m	preto	k
verde	g	ciano	c	amarelo	y	branco	w

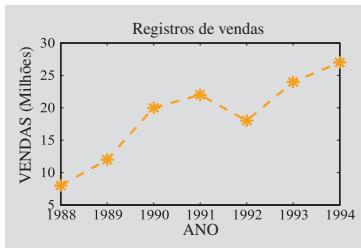
Os especificadores de tipo de marcador são:

Marcador	Especificador	Marcador	Especificador	Marcador	Especificador
sinal de mais	+	asterisco	*	quadrado	s
círculo	o	ponto	.	diamante	d

Os especificadores são digitados como *strings* no comando `plot`. Dentro da *string*, os especificadores podem ser digitados em qualquer ordem.

O comando `plot` cria gráficos simples. O gráfico pode ser modificado para incluir rótulos de eixo, um título e outras características. Gráficos podem ser formatados com o uso de comandos do MATLAB após o comando `plot`. Os comandos de formatação utilizados para adicionar rótulos e um título são:

```
xlabel('texto no formato string')
ylabel('texto no formato string')
title('texto no formato string')
```



**Figura A-6** Gráfico formatado.

```
>> ano=[1988:1:1994];
>> vendas=[8 12 20 22 18 24 27];
>> plot(ano,vendas,'--r*','linewidth',2,'markersize',12)
>> xlabel('ANO')
>> ylabel('VENDAS(Milhões)')
>> title('Registros de Vendas')
```

A formatação de gráficos também pode ser feita na janela da figura usando o Editor de Gráficos (*Plot Editor*).

## A.9 PROBLEMAS

- A.1** Defina as variáveis  $a$ ,  $b$ ,  $c$  e  $d$  como  $a = 14,75$ ,  $b = -5,92$ ,  $c = 61,4$  e  $d = 0,6$  ( $ab - c$ ). Avalie:

$$(a) \quad a + \frac{ab(a+d)^2}{c \sqrt{|ab|}} \qquad (b) \quad de^{\left(\frac{d}{2}\right)} + \frac{(ad+cd)/\left(\frac{25}{a} + \frac{35}{b}\right)}{(a+b+c+d)}$$

- A.2** Defina duas variáveis:  $\alpha = 5\pi/9$  e  $\beta = \pi/7$ . Usando essas variáveis, mostre que a seguinte identidade trigonométrica está correta calculando os valores nos lados esquerdo e direito da equação.

$$\sin \alpha \cos \beta = \frac{1}{2} [\sin(\alpha - \beta) + \sin(\alpha + \beta)]$$

- A.3** Crie um vetor linha com 16 elementos igualmente espaçados, no qual o primeiro elemento é 4 e o último elemento é 61.

- A.4** Crie um vetor coluna no qual o primeiro elemento é 31, os elementos decrescem em incrementos de  $-4$ , e o último elemento é  $-9$  (um vetor coluna pode ser criado como a transposta de um vetor linha).

- A.5** Crie a matriz mostrada usando a notação adequada para criar vetores com espaçamento constante em cada linha (isto é, não digite os elementos individualmente).

$A =$

0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000
3.0000	9.1667	15.3333	21.5000	27.6667	33.8333	40.0000
28.0000	27.7500	27.5000	27.2500	27.0000	26.7500	26.5000
6.0000	5.0000	4.0000	3.0000	2.0000	1.0000	0

**A.6** Crie a matriz  $A$  do Problema A.5 e então use pontos-e-vírgulas para endereçar um grupo de elementos de forma a criar os seguintes vetores:

- (a) Crie um vetor linha com quatro elementos chamado  $va$  que contenha do segundo ao quinto elemento da terceira linha de  $A$ .
- (b) Crie um vetor coluna com quatro elementos chamado  $vb$  que contenha os elementos da sexta coluna de  $A$ .

**A.7** Crie a matriz  $A$  do Problema A.5 e então use pontos-e-vírgulas para endereçar um grupo de elementos de forma a criar as seguintes matrizes:

- (a) Crie uma matriz  $B$  ( $3 \times 4$ ) usando a primeira, a segunda e a quarta coluna de  $A$ , juntamente com a primeira, a segunda e a sétima coluna dessa mesma matriz.
- (b) Crie uma matriz  $C$  ( $2 \times 3$ ) a partir da primeira e da segunda linha de  $A$ , juntamente com a segunda, a quinta e a sexta coluna dessa mesma matriz.

**A.8** Na função  $y = \frac{(x^3 + 1)^2}{x^2 + 2}$ , calcule o valor de  $y$  para os seguintes valores de  $x$ : -1,6, -1,2, -0,8, -0,4, 0, 0,4, 0,8 e 1,2. Resolva o problema primeiro criando um vetor  $x$ , e depois criando um vetor  $y$ , usando cálculos elemento por elemento. Trace um gráfico usando marcadores na forma de asterisco para representar os pontos e uma linha preta para uni-los. Coloque rótulos nos eixos.

**A.9** Defina  $k$  como o escalar  $k = 0,8$  e  $x$  como o vetor  $x = -3, -2,8, -2,6, \dots, 2,6, 2,8, 3$ . Depois use essas variáveis para calcular  $y = \frac{8a^2}{x^2 + 4a^2}$ . Trace um gráfico de  $y$  versus  $x$ .

**A.10** Use o MATLAB para mostrar que a soma da série infinita  $\sum_{n=0}^{\infty} (-1)^n \frac{1}{(2n+1)}$  converge para  $\pi/4$ . Faça isso calculando a soma para:

- (a)  $n = 100$
- (b)  $n = 1000$
- (c)  $n = 5000$

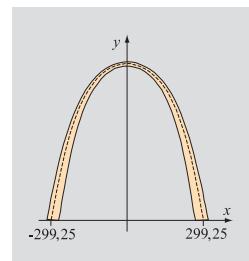
Em cada letra, crie um vetor  $n$  no qual o primeiro elemento é 0, o incremento é 1 e o último termo é 100, 1000 ou 5000. Depois, use cálculos elemento por elemento para criar um vetor nos quais os elementos são iguais a  $(-1)^n \frac{1}{(2n+1)}$ . Finalmente, use a função `sum` para somar os termos das séries. Compare os valores obtidos

nas letras  $a$ ,  $b$  e  $c$  com o valor de  $\pi/4$  (não se esqueça de digitar pontos-e-vírgulas no final dos comandos para evitar que vetores grandes sejam exibidos na tela).

- A.11** O arco Gateway em St. Louis tem sua forma descrita pela seguinte equação.

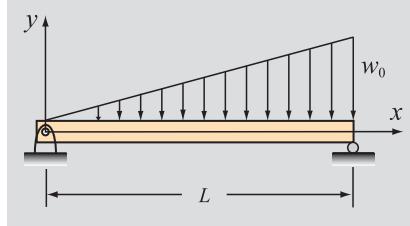
$$y = 21,147 - 20,97 \cosh\left(\frac{x}{30,39}\right) \text{ m}$$

Trace um gráfico do arco para  $-299,25 \leq x \leq 299,25$ .



- A.12** Uma viga é presa de um lado e suportada por um rolamento em sua outra extremidade. A deflexão  $y$  no ponto  $x$  de uma viga carregada da forma mostrada é dada pela equação:

$$y = \frac{-w_0 x}{360 E I L} (3x^3 - 10L^2 x^2 + 7L^4)$$

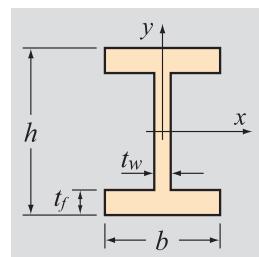


onde  $E$  é o módulo elástico,  $I$  é o momento de inércia e  $L$  é o comprimento da viga. Para a viga mostrada na figura,  $L = 6 \text{ m}$ ,  $E = 70 \times 10^9 \text{ Pa}$  (alumínio),  $I = 9,19 \times 10^{-6} \text{ m}^4$  e  $w_0 = 800 \text{ N/m}$ .

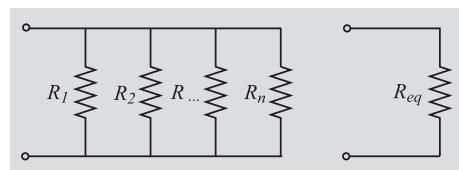
Trace a deflexão  $y$  da viga em função de  $x$ .

- A.13** Escreva uma função no MATLAB que converta a massa e a altura de uma pessoa de unidades SI (kg e cm) para unidades inglesas (libras e polegadas). Use, como nome da função e argumentos,  $[pol, lib] = \text{SIparaIng}(cm, kg)$ . Os argumentos de entrada são a altura em centímetros e a massa em kg, e os argumentos de saída são a altura em polegadas e o peso em libras. Use a função na janela de comandos para determinar, em unidades inglesas, a altura e a massa de uma pessoa de 85 kg com 178 cm de altura.

- A.14** Escreva uma função no MATLAB que determine a área da seção reta  $A$  e os momentos de inércia  $I_{xx}$  e  $I_{yy}$  de uma viga “I”. Use, como nome da função e argumentos,  $[A, I_{xx}, I_{yy}] = \text{MoIner}(b, h, t_f, t_w)$ . Os argumentos de entrada são a largura,  $b$ , a altura,  $h$ , a espessura da borda,  $t_f$  e a espessura central,  $t_w$ , da viga, em milímetros, conforme mostrado na figura. Os argumentos de saída são a área da seção reta (em  $\text{mm}^2$ ) e os momentos de inércia (em  $\text{mm}^4$ ). Use a função na janela de comandos para determinar a área da seção reta e os momentos de inércia de uma viga com  $b = 140 \text{ mm}$ ,  $h = 400 \text{ mm}$ ,  $t_f = 8,8 \text{ mm}$  e  $t_w = 6,4 \text{ mm}$ .



- A.15** Escreva uma função no MATLAB que calcule a resistência equivalente  $R_{eq}$  de  $n$  resistores  $R_1$ ,

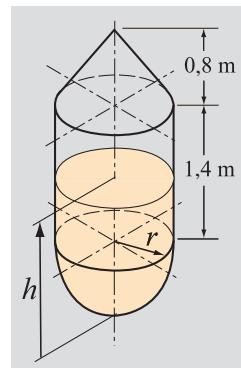


$R_2, \dots, R_n$  conectados em paralelo. Use, como nome da função e argumentos,  $\text{Req} = \text{ResistenciaEq}(R)$ . O argumento de entrada é um vetor cujos elementos são os valores dos resistores. O argumento de saída é o valor da resistência equivalente. A função deve funcionar para qualquer número de resistores. Use a função na janela de comandos para determinar a resistência equivalente dos cinco resistores a seguir, que são conectados em paralelo:  $R_1 = 200 \Omega$ ,  $R_2 = 600 \Omega$ ,  $R_3 = 1000 \Omega$ ,  $R_4 = 100 \Omega$  e  $R_5 = 500 \Omega$ .

- A.16** Um vetor é dado por:  $x = [15 85 72 59 100 80 44 60 91 38]$ . Usando declarações condicionais e laços de repetição, escreva um programa que determine a média dos elementos do vetor que são maiores que 59.
- A.17** Escreva uma função que crie um vetor cujos elementos sejam os números primos entre dois números. Como nome da função e argumentos, use  $\text{pr} = \text{Primos}(a, b)$ . A entrada da função são dois números (inteiros)  $a$  e  $b$  (tais que  $a < b$ ), e a saída  $\text{pr}$  é um vetor cujos elementos são os números primos entre  $a$  e  $b$ .
- A.18** Escreva uma função que ordene na ordem decrescente os elementos de um vetor de tamanho qualquer. Use, como nome da função e argumentos,  $y = \text{ordenadec}(x)$ . A entrada da função é um vetor  $x$  de tamanho qualquer, e a saída  $y$  é um vetor no qual os elementos de  $x$  são arranjados na ordem decrescente. Não use a função `sort` do MATLAB. Teste sua função usando-a na janela de comandos para rearranjar o vetor a seguir:  $[-2, 8, 29, 0, 3, -17, -1, 54, 15, -10, 32]$ .

- A.19** Um tanque de combustível cilíndrico tem fundo hemisférico e tampa superior cônica, conforme mostrado. O raio do cilindro e do fundo hemisférico é  $r = 60$  cm.

Escreva uma função (use como nome e argumentos  $V = \text{Vfuel}(h)$ ) que forneça o volume do combustível no tanque em função da altura  $h$ . Use a função para traçar um gráfico do volume em função de  $h$  para  $0 \leq h \leq 2,8$  m.



## Apêndice B

# Programas de MATLAB

A tabela a seguir é um resumo dos principais programas de MATLAB listados neste livro (funções e programas criados pelo usuário).

Número do programa	Nome do programa ou título	Tipo do arquivo	Descrição	Página
3-1	Método da bisseção	Programa	Resolve uma equação não-linear usando o método da bisseção	78
3-2	NewtonRaiz	Função	Determina a raiz de uma equação não-linear usando o método de Newton	84
3-3	SecanteRaiz	Função	Determina a raiz de uma equação não-linear usando o método da secante	89
4-1	Gauss	Função	Resolve um sistema de equações lineares usando o método de eliminação de Gauss	123
4-2	GaussPivot	Função	Resolve um sistema de equações lineares usando o método de eliminação de Gauss com pivotação	127
4-3	LUdecompCrout	Função	Decompõe uma matriz em matrizes triangulares inferior e superior usando o método de Crout	140
4-4	ProgressSub	Função	Aplica a substituição progressiva	140
4-5	RegressSub	Função	Aplica a substituição regressiva	141
4-6	Decomposição LU usando o método de Crout	Programa	Resolve um sistema de equações lineares com o método LU usando a decomposição de Crout	141
4-7	InversaLU	Função	Calcula a inversa de uma matriz com a decomposição LU usando o método de Crout	144

Número do programa	Nome do programa ou título	Tipo do arquivo	Descrição	Página
4-8	Iteração de Gauss-Seidel	Programa	Resolve um sistema de quatro equações lineares usando o método iterativo de Gauss-Seidel	149
4-9	Tridiagonal	Função	Resolve um sistema tridiagonal de equações lineares usando o algoritmo de Thomas	157
4-10	QRFatora	Função	Fatora uma matriz em uma matriz ortogonal e em uma matriz triangular superior	184
5-1	RegressaoLinear	Função	Ajuste de curvas usando o método de mínimos quadrados e funções lineares	205
5-2	Ajuste de curvas com funções não-lineares	Programa	Ajuste de curvas usando o método de mínimos quadrados e funções não-lineares	210
5-3	Ajuste de curvas usando regressão polinomial	Programa	Ajuste de curvas de um polinômio de quarta ordem	215
5-4	LagrangeINT	Função	Interpolação usando um polinômio de Lagrange	221
5-5	NewtonINT	Função	Interpolação usando um polinômio de Newton	227
5-6	SplineLinear	Função	Interpolação usando splines lineares	230
6-1	derivada	Função	Calcula a derivada de uma função dada por um conjunto de pontos	258
7-1	trapezoidal	Função	Integração numérica usando o método trapezoidal composto	295
7-2	Romberg	Função	Integração numérica usando o método de integração de Romberg	315
8-1	edoEuler	Função	Resolve uma EDO de primeira ordem usando o método explícito de Euler	335
8-2	Solução de uma EDO de primeira ordem	Programa	Resolve um problema de valor inicial de primeira ordem usando o método implícito de Euler	342

Número do programa	Nome do programa ou título	Tipo do arquivo	Descrição	Página
8-3	edoEulerMod	Função	Resolve uma EDO de primeira ordem usando o método de Euler modificado	345
8-4	edoRK4	Função	Resolve uma EDO de primeira ordem usando o método de Runge-Kutta de quarta ordem	358
8-5	Sis2EDOsRK2	Função	Resolve um sistema de duas EDOs de primeira ordem usando o método de Runge-Kutta de segunda ordem	371
8-6	Sis2EDOsRK4	Função	Resolve um sistema de duas EDOs de primeira ordem usando o método de Runge-Kutta de quarta ordem	376
<hr/>				
9-1	Solução de uma EDO de segunda ordem (PVC)	Programa	Resolve uma EDO de segunda ordem (PVC) usando o método do tiro	413
9-2	Solução de uma EDO de segunda ordem (PVC)	Programa	Resolve uma EDO de segunda ordem (PVC) usando o método do tiro em conjunção com o método da bisseção	416
9-3	Solução de uma EDO linear de segunda ordem (PVC)	Programa	Resolve uma EDO linear de segunda ordem usando o método de diferenças finitas	420
9-4	Solução de uma EDO não-linear de segunda ordem (PVC)	Programa	Resolve uma EDO não-linear de segunda ordem usando o método de diferenças finitas	423
9-5	Solução de uma EDO linear de segunda ordem (PVC com condições de contorno mistas)	Programa	Resolve uma EDO linear de segunda ordem com condições de contorno mistas usando o método de diferenças finitas	427

# Índice

---

## A

abs, 444  
Adams-Bashforth, método de, 360  
Adams-Moulton, método de, 361  
Ajuste de curvas, 199-200  
  combinação linear de funções não-lineares, 242-243  
  equação linear, 201  
  equação não-linear, 207-208  
  erro global, 202-203  
  polinômios, 211-212  
  regressão por mínimos quadrados, 203  
Algoritmo, 35-36  
Algoritmo de Thomas, 154-155  
Antiderivada, 44  
Arquivo de função, 458  
Arquivo de programa, 455  
Arranjo (no MATLAB), 446  
Arredondamento, 29-30  
Autovalores, 167-168  
Autovetores, 167-168

## B

Basic Fitting, interface, 242-243  
Bisseção, método da, 76-77  
Bit, 26-27  
bvp4c, 429  
bvpinit, 429-430  
Byte, 26-27

## C

Canônica, de uma EDO linear, forma, 58-59  
ccfun, 429  
ceil, 445  
clc, 442  
Coeficiente pivô, 117  
cond, 153  
Condições de contorno, 408  
Confinamento, métodos de, 74-75  
Corte, 29-30  
Crout, método de, 135  
cubic, 237-238

## D

dblquad, 308-309  
Declaração condicional, 462  
Decomposição LU, 133  
  eliminação de Gauss, 134  
  método de Crout, 135  
Decomposição LU com pivotação, 142  
Derivada  
  aproximação por diferenças finitas, 254, 255  
  diferença central, 256, 262, 264-265  
  diferença progressiva, 256, 260, 265-266  
  diferença regressiva, 256, 261, 265-266  
Derivada, 43-44  
Derivada parcial, 61-62  
Desigualdade triangular, 50-51, 161  
det, 153, 455  
Determinante, 54-55  
Diagonalmente dominante, 145-146  
Diferença central, fórmula de, 256  
Diferença progressiva, fórmula de, 256  
Diferença regressiva, fórmula de, 256  
Diferenças divididas, 223-224  
Diferenças divididas, tabela, 225  
Diferenças finitas, aproximação por, 254, 255  
Diferenças finitas, método de, 408-409, 417-418  
Diferenciação  
  derivadas parciais, 275-277  
  extrapolação de Richardson, 271-272  
  usando ajuste de curvas, 270  
  usando polinômios de Lagrange, 268-269  
Diferenciação, fórmulas de, 266-267  
Diferenciação numérica  
  erro, 274-275  
  fórmulas de diferenças finitas, 266-267  
Diferencial total, 62-63  
diff, 270-271  
Dirichlet, condição de contorno de, 408  
Discretização, 418-419  
disp, 456-457  
Divisão à direita, 149-150  
Divisão à esquerda, 149-150  
Domínio, 42  
dot, 452

**E**

EDO, 58-59  
 EDO, solução no MATLAB, 379  
 EDOs, sistemas de, solução no MATLAB, 386  
 $\text{eig}$ , 184-185  
 $\text{eps}$ , 28-29  
 Epsilon, 28-29  
 Epsilon da máquina, 28-29  
 Equação característica, 169-170  
 Equação diferencial, 58-59  
 Equação pivô, 117  
 Equações diferenciais  
     EDO rígida, 395-396  
     estabilidade, 393-394  
     forma canônica, 58-59  
     fundamentos, 58-59  
     homogêneas, 58-59  
     método de Adams-Bashforth, 360  
     método de Adams-Moulton, 361  
     método de diferenças finitas, 417-418  
     método de Euler modificado, 343  
     método de Euler progressivo, 333  
     método de Euler regressivo, 341  
     método de Heun, 343, 349, 353  
     método de Nystrom, 353  
     método do ponto central, 346  
     método do tiro, 409-410  
     método explícito de Euler, 332, 472  
     método implícito de Euler, 340  
     métodos de Runge-Kutta, 347  
     métodos explícitos, 330  
     métodos explícitos de passo simples, 331  
     métodos implícitos, 330  
     métodos multipasso, 359  
     métodos preditor-corretor, 362  
     multipasso, 330  
     não-homogêneas, 58-59  
     não-lineares, 59-60  
     ordem, 58-59  
     passo de integração, 389-390  
     passo simples, 330  
     primeira ordem, 329  
     problema de valor de contorno, 407  
     problema de valor inicial, 329  
     PVI de segunda ordem, 374  
     Runge-Kutta de quarta ordem, 353  
     Runge-Kutta de segunda ordem, 348  
     Runge-Kutta de terceira ordem, 352  
     sistema de primeira ordem, 364  
     solução analítica, 59-60  
 Equações lineares, sistema de, 55-56, 91-92, 113  
 Equações não-lineares, sistemas de, 96-97

**E**

Erro, 159  
     de arredondamento, 29-30  
     diferenciação numérica, 274-275  
     integração numérica, 309  
     na solução de uma EDO, 331  
     relativo, 75-76, 163  
     relativo estimado, 75-76  
     relativo verdadeiro, 33-34, 75-76  
     resolvendo um sistema de equações, 162  
     resolvendo uma equação, 74-75  
     total, 33-34  
     verdadeiro, 33-34, 75-76  
 Erro de truncamento, 32-33  
     global, 331, 338  
     local, 331, 337-338  
     local no método de Runge-Kutta de segunda ordem, 388-389  
     método explícito de Euler, 336-337  
     propagado, 331  
     Runge-Kutta de quarta ordem, 355  
     Runge-Kutta de segunda ordem, 349  
     Runge-Kutta de terceira ordem, 353  
 Escalar, 46-47  
 Estabilidade, solução de EDOs, 393-394  
 Euler progressivo, método de, 333  
 Euler regressivo, método de, 341  
 $\text{exp}$ , 444  
 Expoente, 25-26  
 Extrapolação de Richardson  
     diferenciação, 271-272  
     integração, 311  
 $\text{eye}$ , 450

**F**

Fatoração QR, método de, 175-176  
 $\text{fix}$ , 445  
 $\text{floor}$ , 445  
 $\text{for-end}$ , 463-464  
 $\text{format}$ , 445  
 Formatos de exibição (MATLAB), 445  
 Fórmula de diferença  
     central, 256  
     progressiva, 256  
     regressiva, 256  
 Fórmula de Newton-Cotes, 289  
 $\text{fprintf}$ , 456-457  
 Função de iteração, 88-90  
 $\text{fzero}$ , 93-94

**G**

Gauss, método de eliminação de, 116  
 Gauss com pivotação, método de eliminação de, 126

Gauss-Jordan, método de eliminação de, 129  
 Gauss-Seidel, método iterativo de, 146-147  
 Gráficos, traçado de (no MATLAB), 464-465

**H**

Heun, método de, 343, 349, 353  
 Householder, matriz de, 177

**I**

`if-else-end`, 462, 463-464  
`if-elseif-end`, 463  
`if-end`, 462  
**Integração**  
 extrapolação de Richardson, 311  
 fórmula de Newton-Cotes, 289  
 integração de Romberg, 313-314  
 integrais múltiplas, 306-307  
 método de Simpson 1/3, 296  
 método de Simpson 3/8, 299  
 método do ponto central, 291  
 método do retângulo, 290  
 método trapezoidal, 292  
 métodos abertos, 289  
 métodos fechados, 289  
 quadratura de Gauss, 300-301

**Integração composta**  
 método de Simpson 1/3, 297  
 método de Simpson 3/8, 300  
 método do ponto central, 292  
 método do retângulo, 290  
 método trapezoidal, 293

**Integração de Romberg**, 313-314

**Integrais impróprias**, 316-317

**Integrais múltiplas**, 306-307

**Integral**

com limites ilimitados, 317-318  
 com singularidades, 316-317  
 definida, 45-46  
 indefinida, 44

`interp1`, 240-241

**Interpolação**, 199-200

linear, 240-241  
 mais próxima, 240-241  
 polinômios de Lagrange, 217-218  
 polinômios de Newton, 221-222  
 por partes (spline), 228  
 único polinômio, 215-216

**Interpolação spline**

cúbica, 233-234  
 linear, 228  
 quadrática, 230

**Intervalo**, 42

`inv`, 152-153  
 Inversa de uma matriz, 53-54, 142, 150-151

**J**

Jacobi, método iterativo de, 146-147  
 Jacobiano, 63-64, 100  
 Janela de comandos, 442

**L**

Laço de repetição, 463-464  
`length`, 449  
`linear`, 240-241  
 Linearmente dependente, 49  
 Linearmente independente, 49  
`log`, 444  
`log10`, 444  
`lu`, 151-152

**M**

Mantissa, 25-26  
**Matriz**, 50-51  
 determinante de, 54-55  
 diagonal, 52-53  
 identidade, 52-53  
 inversa, 53-54, 142  
 multiplicação, 52  
 multiplicação por escalar, 51  
 quadrada, 52-53  
 simétrica, 53-54  
 soma, 51  
 subtração, 51  
 tamanho, 50-51  
 transposta, 51  
 triangular inferior, 52-53, 115  
 triangular superior, 52-53, 113-114  
 zero, 53-54  
**Matrizes similares**, 176-177  
`mean`, 455  
**Método da falsa posição**, 79-80  
**Método da iteração do ponto fixo**, 88-90, 101  
**Método da potência**, 170-171  
 básico, 170-171  
 deslocado, 175-176  
 inverso, 174-175  
**Método de Euler modificado**, 343  
**Método de Simpson 1/3 (integração)**, 296  
**Método de Simpson 3/8 (integração)**, 299  
**Método do ponto central (integração)**, 291  
**Método do ponto central (solução de uma EDO)**, 346  
**Método do retângulo (integração)**, 290  
**Método do tiro**, 409-410

- Método explícito de Euler, 332  
 Método implícito de Euler, 340  
 Método trapezoidal (integração), 292  
 Métodos abertos, 74-75  
 Métodos de Runge-Kutta, 347  
   de quarta ordem, 353  
   de segunda ordem, 348  
   de terceira ordem, 352  
   método de Heun, 349  
 Métodos explícitos (resolvendo uma EDO), 330  
 Métodos implícitos (resolvendo uma EDO), 330  
 Métodos iterativos, 145-146  
 Métodos multipasso, equações diferenciais, 359  
 Métodos preditor-corretor, 362
- N**
- nearest, 240-241  
 Neumann, condição de contorno de, 408  
 Newton, método de, 81-82, 97-98  
 norm, 153  
 Norma, 56-58, 161  
 Norma 1, 161  
 Norma 2, 162  
 Norma 2 Euclidiana, 161  
 Norma infinita, 161, 162  
 Normas de matrizes, 162  
 Normas de vetores, 161  
 Nós, 228  
 Número binário, 24-25  
 Número condição, 163-164  
 Número decimal, 23-24  
 Nystrom, método de, 353
- O**
- ode113, 381  
 ode15s, 381  
 ode23, 381  
 ode23s, 381  
 ode23t, 381  
 ode23tb, 381  
 ode45, 381  
 ones, 450  
 Operações elemento por elemento, 453  
 Operador lógico, 460-461  
 Operador relacional, 459-460  
 Overflow, 27-28
- P**
- Padrão IEEE, 26-27  
 Passo de integração na solução de EDOs, 389-390
- pchip, 237-238, 241-242  
 plot, 464-465  
 Polinômios de Lagrange  
   ajuste de curvas, 217-218  
   fórmulas de diferenciação, 268-269  
 Polinômios de Newton, 221-222  
 polyder, 271-272  
 polyfit, 240-241  
 Ponto fixo, representação de números em, 25-26  
 Pontos de malha, 417-418  
 Potência básico, método da, 171  
 Potência deslocada, método da, 175-176  
 Precisão dupla, 26-27  
 Precisão simples, 26-27  
 Problema do autovalor, 167-168  
 Produto escalar, 49  
 Produto vetorial, 49  
 PVC de dois pontos, 408
- Q**
- qr, 186-187  
 quad, 307-308  
 quadl, 308-309  
 Quadratura de Gauss, 300-301
- R**
- Regra da cadeia, 62-63  
 Regra de Cramer, 55-56  
 Regressão  
   equação linear, 203  
   polinomial, 213  
 Regressão por mínimos quadrados, 203  
 Regula falsi, método, 79-80  
 Representação de números  
   binária, 24-25  
   decimal, 23-24  
   ponto flutuante, 25-26  
 Resíduo, 159  
 Resíduo relativo, 163  
 roots, 94-95  
 round, 445
- S**
- Secante, método da, 86-87  
 Série de Taylor  
   duas variáveis independentes, 65-67  
   fórmula de diferenças finitas, 260  
   Runge-Kutta de segunda ordem, 349  
     uma variável independente, 64-65  
 sin, 444

`sind`, 444  
Sistema de EDOs de primeira ordem, 364  
Sistema triagonal de equações, 154-155  
Sistemas mal condicionados, 165-166  
`size`, 450  
`solinit`, 429-430  
Soluções múltiplas, 94-95  
Soma de Riemann, 45-46  
`sort`, 455  
`spline`, 237-238, 241-242  
Splines cúbicas, 233-234  
Splines lineares, 228  
Splines quadráticas, 230  
`sqrt`, 444  
Strings, 450  
Substituição progressiva, 115  
Substituição regressiva, 115  
`sum`, 455

## T

Teorema do valor intermediário, 43-44  
Teorema do valor médio para derivadas, 44  
Teorema do valor médio para integrais, 45-46  
Teorema fundamental do cálculo, 45-46  
`title`, 467  
Tolerância, 75-76

Transposta, 48-49, 51  
`trapz`, 308-309

## U

Underflow, 27-28

## V

Valor médio de uma função, 46-47  
Variáveis globais, 455-456  
Variável dependente, 42  
Variável independente, 42  
Vetor (no MATLAB), 446  
Vetor, 46-47  
Vetor unitário, 47-48

## X

`xlabel`, 467

## Y

`ylabel`, 467

## Z

`zeros`, 450