

## Gerais

5.36. Seja a função  $f(x) = \sqrt{x} + 1$ .

a) Achar o polinômio de Lagrange de grau 1 que passa pelos pontos de abscissas 0 e 1.

b) Integrar, analiticamente, o polinômio obtido no item (a) no intervalo  $[0, 1]$ .

c) Calcular  $\int_0^1 \sqrt{x} + 1 \, dx$  utilizando a regra do trapézio.

d) Justificar por que os resultados dos itens (b) e (c) são iguais.

5.37. Deduzir a fórmula de integração numérica que utiliza um polinômio interpolador de grau 4 e comparar seus coeficientes com aqueles compilados na Tabela 5.1.

5.38. Propor a fórmula composta de Newton-Cotes para a regra de integração baseada em um polinômio interpolador de grau 4.

5.39. Seja a integral  $\int_0^2 (4x^3 + 2x + 1) \, dx$ .

a) Avaliar pela regra do 1/3 de Simpson com  $m = 2$ .

b) O resultado é exato? Por quê?

5.40. Considere a integral  $\int_0^\pi (x^2 + \sin(x)) \, dx$ .

a) Determinar o número de subintervalos para que a regra dos 3/8 de Simpson calcule a integral com  $E < 10^{-3}$ .

b) Calcular utilizando o programa implementado no Exercício 5.16.

c) Comparar os resultados acima com o valor exato.

5.41. Deduzir a fórmula do erro de integração para a regra do 1/3 de Simpson composta.

5.42. Deduzir a fórmula do erro de integração para a regra dos 3/8 de Simpson composta.

5.43. Considere a integral

$$\int_1^3 \int_{-1}^1 (4x^3y^3 + 2x^2y^2 + 1) \, dy \, dx.$$

a) Calcular pela fórmula de Newton-Cotes com  $n_x = n_y = m_x = m_y = 1$ .

b) Calcular por Gauss-Legendre com  $n_x = n_y = 2$ .

c) Verificar a exatidão dos dois resultados acima.

d) Por que um dos resultados é exato e o outro não?

5.44. Gerar uma tabela com os valores da distribuição de probabilidade normal padrão no intervalo  $[0, 5; 0, 6]$ , com incremento 0,005 e exibindo 5 decimais.

5.45. Calcular a integral  $\int_0^\infty \sqrt{x} e^{-x} \, dx$ .

## Capítulo 6

## Raízes de equações

A necessidade de encontrar valores de  $x = \xi$  que satisfaçam à equação  $f(x) = 0$  aparece frequentemente em uma ampla variedade de problemas provenientes das Ciências e das Engenharias. Esses valores especiais são chamados de raízes da equação  $f(x) = 0$  ou zeros da função  $f(x)$ , os quais podem ser vistos na Figura 6.1. Para equações algébricas de grau até quatro, suas

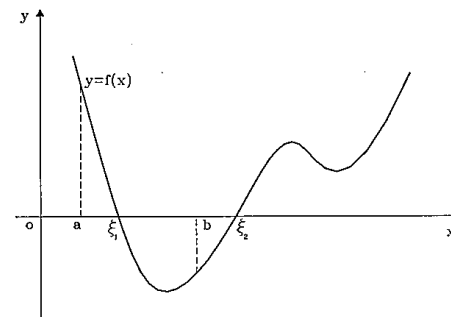


Figura 6.1 Raízes da equação  $f(x) = 0$ .

raízes podem ser calculadas por meio de uma expressão, tal como  $x = (-b \pm \sqrt{b^2 - 4ac})/2a$  para determinar as duas raízes de  $f(x) = ax^2 + bx + c = 0$ . No entanto, para equações algébricas de grau superior a quatro e para a grande maioria das equações transcendentais, as raízes não podem ser calculadas analiticamente. Nesses casos, têm que ser usados métodos que encontrem uma solução aproximada para essas raízes.

## 6.1 Isolamento de raízes

O problema de calcular uma raiz pode ser dividido em duas fases

1. Isolamento da raiz, isto é, encontrar um intervalo  $[a, b]$  que contenha uma, e somente uma, raiz de  $f(x) = 0$  (ver Figura 6.1).
2. Refinamento da raiz, ou seja, a partir de um valor inicial  $x_0 \in [a, b]$ , gerar uma sequência  $\{x_0, x_1, x_2, \dots, x_k, \dots\}$  que convirja para uma raiz exata  $\xi$  de  $f(x) = 0$ .

Existem alguns métodos para cálculo das raízes de equações polinomiais que não requerem que haja um prévio isolamento de cada raiz. No entanto, a maioria dos métodos para cálculo de raízes necessita que a mesma esteja confinada em um dado intervalo e, além do mais, essa raiz deve ser única em tal intervalo. Devido à existência de alguns teoremas da Álgebra que fornecem importantes informações sobre as equações polinomiais, o isolamento de raízes das equações algébricas e das transcendentais será visto separadamente.

### 6.1.1 Equações algébricas

Seja uma equação algébrica de grau  $n$ ,  $n \geq 1$ , escrita na forma de potências

$$P(x) = c_n x^n + c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \dots + c_2 x^2 + c_1 x + c_0 = 0 \quad (6.1)$$

com os coeficientes  $c_i$  sendo reais e  $c_n \neq 0$ . Antes de serem abordados métodos para determinar o número e os limites das raízes reais de uma equação polinomial, serão vistos modos de avaliar um polinômio e algumas propriedades importantes desse tipo de função.

#### Avaliação de polinômio

Para obter o valor de um polinômio  $P(x) = c_n x^n + c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \dots + c_2 x^2 + c_1 x + c_0$  em um ponto  $x = a$ , usualmente, se faz

$$P(a) = c_n a^n + c_{n-1} a^{n-1} + c_{n-2} a^{n-2} + \dots + c_2 a^2 + c_1 a + c_0.$$

Dessa maneira, para avaliar  $P(x)$  de grau  $n$ , em  $x = a$ , são necessárias  $n(n+1)/2$  multiplicações e  $n$  adições.

Exemplo 6.1 Avaliar  $P(x) = 3x^5 - 2x^4 + 5x^3 + 7x^2 - 3x + 1$  em  $x = 2$ .

$$P(2) = 3 \times 2^5 - 2 \times 2^4 + 5 \times 2^3 + 7 \times 2^2 - 3 \times 2 + 1 = 127,$$

sendo requeridas 15 multiplicações e 5 adições.

No entanto, uma maneira mais eficiente de avaliar um polinômio é o método de Horner, que consiste em reescrever o polinômio de forma a evitar potências.

Deste modo,

$$\begin{aligned} P(x) &= c_n x^n + c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \dots + c_2 x^2 + c_1 x + c_0, \\ &= (c_n x^{n-1} + c_{n-1} x^{n-2} + c_{n-2} x^{n-3} + \dots + c_2 x + c_1)x + c_0, \\ &= ((c_n x^{n-2} + c_{n-1} x^{n-3} + c_{n-2} x^{n-4} + \dots + c_2)x + c_1)x + c_0, \\ &\dots \\ P(x) &= \underbrace{(\dots (c_n x + c_{n-1})x + c_{n-2})x + \dots + c_2)x + c_1)x + c_0}_{n-1}. \end{aligned}$$

© processo de Horner requer apenas  $n$  multiplicações e  $n$  adições para avaliar um polinômio de grau  $n$ .

Exemplo 6.2 Avaliar  $P(x) = 3x^5 - 2x^4 + 5x^3 + 7x^2 - 3x + 1$  em  $x = 2$  usando o processo de Horner.

$$\begin{aligned} P(x) &= (((((3x - 2)x + 5)x + 7)x - 3)x + 1, \\ P(2) &= (((((3 \times 2 - 2) \times 2 + 5) \times 2 + 7) \times 2 - 3) \times 2 + 1 = 127. \end{aligned}$$

A Figura 6.2 mostra o algoritmo de Horner para avaliar um polinômio de grau  $n$  no ponto  $x = a$ . Os parâmetros de entrada são o grau  $n$  do polinômio, o vetor  $c$  contendo os coeficientes, sendo  $P(x) = c(1)x^n + c(2)x^{n-1} + \dots + c(n)x + c(n+1)$  e o ponto  $a$  onde  $P(x)$  deve ser avaliado. O parâmetro de saída é a ordenada  $y = P(a)$ .

```

Algoritmo Horner
{ Objetivo: Avaliar um polinômio de grau  $n$  no ponto  $a$  }
parâmetros de entrada  $n, c, a$ 
{ grau, coeficientes e ponto a ser avaliado, onde  $c$  é tal que }
{  $P(x) = c(1)x^n + c(2)x^{n-1} + \dots + c(n)x + c(n+1)$  }
parâmetros de saída  $y$  { ordenada  $P(a)$  }
 $y \leftarrow c(1)$ 
para  $i \leftarrow 2$  até  $n+1$  faça
     $y \leftarrow y * a + c(i)$ 
fimpara
fimalgoritmo
  
```

Figura 6.2 Algoritmo de Horner para avaliar polinômio.

Exemplo 6.3 Avaliar o polinômio  $P(x) = 3x^5 - 2x^4 + 5x^3 + 7x^2 - 3x + 1$  do Exemplo 6.2, em  $x = 2$  usando o algoritmo da Figura 6.2.

```
% Os parametros de entrada
n = 5
c = 3   -2   5   7   -3   1
a = 2
% produzem o resultado
y = 127
```

### Propriedades gerais

**Teorema 6.1** Uma equação algébrica de grau  $n$  tem exatamente  $n$  raízes, reais ou complexas, contando cada raiz de acordo com a sua multiplicidade.

A demonstração deste teorema é dada por Uspensky [38]. Uma raiz  $\xi$  de (6.1) tem multiplicidade  $m$  se

$$P(\xi) = P'(\xi) = P''(\xi) = \dots = P^{m-1}(\xi) = 0 \text{ e}$$

$$P^m(\xi) \neq 0$$

sendo

$$P^i(\xi) = \frac{d^i P(x)}{dx^i} \Big|_{x=\xi}, \quad i = 1, 2, \dots, m.$$

Exemplo 6.4 Seja

$$P(x) = x^4 + 2x^3 - 12x^2 + 14x - 5 \rightarrow P(1) = 0,$$

$$P'(x) = 4x^3 + 6x^2 - 24x + 14 \rightarrow P'(1) = 0,$$

$$P''(x) = 12x^2 + 12x - 24 \rightarrow P''(1) = 0 \text{ e}$$

$$P'''(x) = 24x + 12 \rightarrow P'''(1) = 36.$$

Assim,  $\xi = 1$  é uma raiz de multiplicidade  $m = 3$ . Considerando também que  $P(-5) = 0$ , o polinômio de grau 4 acima pode ser escrito na forma fatorada  $P(x) = (x-1)^3(x+5)$ . ■

**Teorema 6.2** Se os coeficientes de uma equação algébrica forem reais, então suas raízes complexas serão complexos conjugados em pares, ou seja, se  $\xi_1 = a + bi$  for uma raiz de multiplicidade  $m$ , então  $\xi_2 = a - bi$  também será uma raiz e com a mesma multiplicidade.

Exemplo 6.5 As raízes de  $P(x) = x^2 - 4x + 13 = 0$  são

$$\xi = \frac{4 \pm \sqrt{(-4)^2 - 4 \times 1 \times 13}}{2} \rightarrow \begin{cases} \xi_1 = 2 + 3i \\ \xi_2 = 2 - 3i. \end{cases}$$

**Corolário 6.1** Uma equação algébrica de grau ímpar com coeficientes reais tem, no mínimo, uma raiz real.

Exemplo 6.6 As raízes da equação  $P(x) = x^3 - 9x^2 + 33x - 65 = 0$  são  $\xi_1 = 5$ ,  $\xi_2 = 2 + 3i$  e  $\xi_3 = 2 - 3i$ . Portanto, esta equação de grau 3 tem uma raiz real. ■

### Relação entre raízes e coeficientes

Se  $\xi_i$ ,  $i = 1, 2, \dots, n$  forem as raízes de  $P(x) = 0$ , então ela pode ser escrita na forma fatorada

$$P(x) = c_n(x - \xi_1)(x - \xi_2) \dots (x - \xi_n) = 0.$$

Multiplicando os fatores,

$$\begin{aligned} P(x) = & c_n x^n - c_n(\xi_1 + \xi_2 + \dots + \xi_n)x^{n-1} \\ & + c_n(\xi_1\xi_2 + \xi_1\xi_3 + \dots + \xi_1\xi_n + \xi_2\xi_3 + \dots + \xi_2\xi_n + \dots + \xi_{n-1}\xi_n)x^{n-2} \\ & - c_n(\xi_1\xi_2\xi_3 + \xi_1\xi_2\xi_4 + \dots + \xi_1\xi_2\xi_n + \xi_1\xi_3\xi_4 + \dots + \xi_{n-2}\xi_{n-1}\xi_n)x^{n-3} \\ & + \dots (-1)^n c_n(\xi_1\xi_2\xi_3 \dots \xi_n) = 0. \end{aligned}$$

Comparando a expressão acima com  $P(x) = 0$  escrita na forma de potências (6.1) e aplicando a condição de igualdade das equações algébricas, tem-se que

$$\xi_1 + \xi_2 + \dots + \xi_n = -\frac{c_{n-1}}{c_n},$$

$$\xi_1\xi_2 + \xi_1\xi_3 + \dots + \xi_1\xi_n + \xi_2\xi_3 + \dots + \xi_2\xi_n + \dots + \xi_{n-1}\xi_n = \frac{c_{n-2}}{c_n},$$

$$\xi_1\xi_2\xi_3 + \xi_1\xi_2\xi_4 + \dots + \xi_1\xi_2\xi_n + \xi_1\xi_3\xi_4 + \dots + \xi_{n-2}\xi_{n-1}\xi_n = -\frac{c_{n-3}}{c_n},$$

...

$$\xi_1\xi_2\xi_3 \dots \xi_n = (-1)^n \frac{c_0}{c_n}.$$

As expressões acima relacionando os coeficientes de uma equação algébrica com as suas raízes são conhecidas como relações de Girard. Essas relações são válidas também para as raízes complexas.

Exemplo 6.7 As raízes da equação do Exemplo 6.6,  $P(x) = x^3 - 9x^2 + 33x - 65 = 0$ , são  $\xi_1 = 5$ ,  $\xi_2 = 2 + 3i$  e  $\xi_3 = 2 - 3i$ , assim as relações de Girard são

$$5 + (2 + 3i) + (2 - 3i) = 9 = -\frac{-9}{1},$$

$$5(2 + 3i) + 5(2 - 3i) + (2 + 3i)(2 - 3i) = 33 = \frac{33}{1} \text{ e}$$

$$5(2 + 3i)(2 - 3i) = 65 = (-1)^3 \frac{-65}{1}. \quad \blacksquare$$

Exemplo 6.8 Sejam as equações algébricas de Lagrange definidas pela fórmula de recorrência (5.18) a partir de  $L_0(x) = 1$  e  $L_1(x) = x$

$$L_2(x) = \frac{3x^2 - 1}{2} = 0,$$

$$L_3(x) = \frac{5x^3 - 3x}{2} = 0,$$

$$L_4(x) = \frac{35x^4 - 30x^2 + 3}{8} = 0,$$

$$L_5(x) = \frac{63x^5 - 70x^3 + 15x}{8} = 0.$$

Todas as equações possuem  $c_{n-1} = 0$ , implicando, pelas relações de Girard, que a soma das raízes é nula, ou seja, as raízes são simétricas em relação à origem. Também, as equações de grau ímpar possuem  $c_0 = 0$ ; como consequência, elas têm uma raiz nula. Essas propriedades podem ser verificadas na Figura 5.11, na página 235.

#### Limites das raízes reais

Uma equação algébrica na forma (6.1) pode ter suas raízes reais delimitadas usando o Teorema de Lagrange, cuja demonstração é dada por Demidovich e Maron [10].

**Teorema 6.3 (Lagrange)** Dada a equação  $P(x) = c_n x^n + c_{n-1} x^{n-1} + c_{n-2} x^{n-2} + \dots + c_2 x^2 + c_1 x + c_0 = 0$ , se  $c_n > 0$  e  $k$  ( $0 \leq k \leq n-1$ ) for o maior índice de coeficiente escolhido dentre os coeficientes negativos, então o limite superior das raízes positivas de  $P(x) = 0$  pode ser dado por

$$L = 1 + \sqrt[n-k]{\frac{B}{c_n}},$$

onde  $B$  é o valor absoluto do maior coeficiente negativo em módulo.

Desse modo, se  $\xi_p$  for a maior das raízes positivas de  $P(x) = 0$ , então  $\xi_p \leq L$ .

Exemplo 6.9 Seja  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$ . Os coeficientes negativos são  $c_2 = -13$  e  $c_1 = -14$ , portanto,  $k = 2$ , pois  $2 > 1$ ,  $B = |-14|$  e

$$L = 1 + \sqrt[4-2]{\frac{14}{1}} \rightarrow L = 4,74.$$

O Teorema de Lagrange garante que  $P(x) = 0$  não tem nenhuma raiz maior que 4,74.

Se  $c_i > 0$  ( $i = 0, 1, \dots, n$ ), então  $P(x) = 0$  não tem raízes positivas, pois  $P(x) = \sum_{i=0}^n c_i x^i > 0$  para  $c_i > 0$  e  $x > 0$ . Para determinar os limites superiores e inferiores das raízes positivas e negativas, são necessárias três equações auxiliares

$$P_1(x) = x^n P(1/x) = 0,$$

$$P_2(x) = P(-x) = 0 \text{ e}$$

$$P_3(x) = x^n P(-1/x) = 0.$$

Se  $\xi_i$ ,  $i = 1, 2, \dots, n$ , as raízes de  $P(x) = 0$ , então  $P(x)$  na forma fatorada é

$$P(x) = c_n (x - \xi_1)(x - \xi_2) \dots (x - \xi_n).$$

Desse modo,

$$P_1(x) = c_n x^n (1/x - \xi_1)(1/x - \xi_2) \dots (1/x - \xi_n),$$

$$P_1(x) = c_n (1 - x\xi_1)(1 - x\xi_2) \dots (1 - x\xi_n),$$

cuja raízes são  $1/\xi_1, 1/\xi_2, \dots, 1/\xi_n$ . Similarmente,

$$P_2(x) = c_n (-x - \xi_1)(-x - \xi_2) \dots (-x - \xi_n),$$

com raízes  $-\xi_1, -\xi_2, \dots, -\xi_n$ , e

$$P_3(x) = c_n x^n (-1/x - \xi_1)(-1/x - \xi_2) \dots (-1/x - \xi_n),$$

$$P_3(x) = c_n (-1 - x\xi_1)(-1 - x\xi_2) \dots (-1 - x\xi_n),$$

sendo as raízes  $-1/\xi_1, -1/\xi_2, \dots, -1/\xi_n$ .

Exemplo 6.10 Seja  $P(x) = x^4 - 6x^3 - 5x^2 + 42x + 40 = 0$ , com raízes  $\xi_1 = -2$ ,  $\xi_2 = -1$ ,  $\xi_3 = 4$ ,  $\xi_4 = 5$ , então as equações auxiliares e suas respectivas raízes são

$$P_1(x) = x^4 P(1/x) = 40x^4 + 42x^3 - 5x^2 - 6x + 1,$$

$$(\xi_1 = -0,5; \xi_2 = -1, \xi_3 = 0,25; \xi_4 = 0,2),$$

$$P_2(x) = P(-x) = x^4 + 6x^3 - 5x^2 - 42x + 40,$$

$$(\xi_1 = 2, \xi_2 = 1, \xi_3 = -4, \xi_4 = -5),$$

$$P_3(x) = x^4 P(-1/x) = 40x^4 - 42x^3 - 5x^2 + 6x + 1,$$

$$(\xi_1 = 0,5; \xi_2 = 1, \xi_3 = -0,25; \xi_4 = -0,2).$$

Se  $1/\xi_q$  for a maior das raízes positivas de  $P_1(x) = 0$ , então  $\xi_q$  será a menor das raízes positivas de  $P(x) = 0$  (ver Exemplo 6.10). Sendo  $L_1$  o limite superior das raízes positivas de  $P_1(x) = 0$ , calculado pelo Teorema 6.3, tem-se que

$$\frac{1}{\xi_q} \leq L_1 \rightarrow \xi_q \geq \frac{1}{L_1},$$

consequentemente, o limite inferior das raízes positivas de  $P(x) = 0$  é  $1/L_1$ . Desse modo, se  $P(x) = 0$  possuir raízes positivas  $\xi^+$ , elas estarão no intervalo

$$\boxed{\frac{1}{L_1} \leq \xi^+ \leq L_1}.$$

Por outro lado, se  $-\xi_r$  for a maior das raízes positivas de  $P_2(x) = 0$ , então  $\xi_r$  será a menor das raízes negativas de  $P(x) = 0$  (ver Exemplo 6.10). Sendo  $L_2$  o limite superior das raízes positivas de  $P_2(x) = 0$ , dado pelo Teorema 6.3

$$-\xi_r \leq L_2 \rightarrow \xi_r \geq -L_2.$$

Se  $-1/\xi_s$  for a maior das raízes positivas de  $P_3(x) = 0$ , então  $\xi_s$  será a menor das raízes negativas de  $P(x) = 0$  (ver Exemplo 6.10). Sendo  $L_3$  o limite superior das raízes positivas de  $P_3(x) = 0$ , dado pelo Teorema 6.3

$$-\frac{1}{\xi_s} \leq L_3 \rightarrow \xi_s \leq -\frac{1}{L_3}.$$

Então, se  $P(x) = 0$  tiver raízes negativas  $\xi^-$ , elas estarão no intervalo

$$\boxed{-L_2 \leq \xi^- \leq -\frac{1}{L_3}}.$$

A Figura 6.3 mostra os limites das raízes reais de uma equação algébrica. É importante notar que esses limites não garantem a existência das raízes reais, mas tão-somente informam onde as raízes reais estarão caso existam.

**Exemplo 6.11** Calcular os limites das raízes reais de  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$  do Exemplo 6.9.

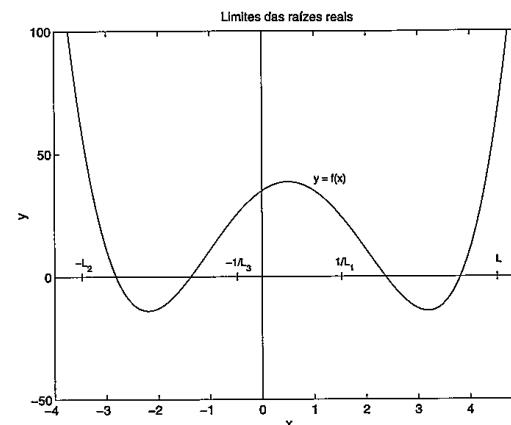
As equações auxiliares são

$$P_1(x) = x^4 P\left(\frac{1}{x}\right) = x^4 \left(\frac{1}{x^4} + \frac{2}{x^3} - \frac{13}{x^2} - \frac{14}{x} + 24\right) = 0 \rightarrow$$

$$P_1(x) = 24x^4 - 14x^3 - 13x^2 + 2x + 1 = 0,$$

$$L_1 = 1 + \sqrt[4]{\frac{14}{24}} \rightsquigarrow \frac{1}{L_1} = 0,63,$$

$$P_2(x) = P(-x) = (-x)^4 + 2(-x)^3 - 13(-x)^2 - 14(-x) + 24 = 0 \rightarrow$$



**Figura 6.3** Limites das raízes reais de uma equação algébrica.

$$P_2(x) = x^4 - 2x^3 - 13x^2 + 14x + 24 = 0,$$

$$L_2 = 1 + \sqrt[4]{\frac{13}{1}} \rightsquigarrow -L_2 = -14 \text{ e}$$

$$P_3(x) = x^4 P\left(\frac{1}{-x}\right) = x^4 \left(\frac{1}{(-x)^4} + \frac{2}{(-x)^3} - \frac{13}{(-x)^2} - \frac{14}{(-x)} + 24\right) = 0 \rightarrow$$

$$P_3(x) = 24x^4 + 14x^3 - 13x^2 - 2x + 1 = 0,$$

$$L_3 = 1 + \sqrt[4]{\frac{13}{24}} \rightsquigarrow -\frac{1}{L_3} = -0,58.$$

Considerando que  $L = 4,74$ , conforme o Exemplo 6.9, então os limites das raízes reais são

$$0,63 \leq \xi^+ \leq 4,74 \text{ e } -14 \leq \xi^- \leq -0,58.$$

Pode ser construído um dispositivo prático para facilitar a determinação dos limites das raízes reais. O dispositivo é constituído de dois blocos. No primeiro bloco, são definidos os coeficientes de  $P(x) = 0$  e de suas três equações auxiliares  $P_1(x) = 0$ ,  $P_2(x) = 0$  e  $P_3(x) = 0$ . Para tal,

1. colocar os coeficientes de  $P(x) = 0$  na coluna  $P(x)$ , com  $c_n$  no topo,
2. inverter a ordem dos coeficientes da coluna  $P(x)$  e colocá-los em  $P_1(x)$ ,
3. trocar o sinal dos coeficientes de  $P(x)$ , cujos índices sejam ímpares e atribuí-los a  $P_2(x)$ ,

4. inverter a ordem dos coeficientes da coluna  $P_2(x)$  e colocá-los em  $P_3(x)$  e
5. se algum  $c_n < 0$ , então trocar o sinal de todos os coeficientes da coluna para garantir que  $c_n > 0$ , conforme exigência do Teorema 6.3.

No segundo bloco, são atribuídos os parâmetros necessários para aplicar o Teorema 6.3 a cada uma das quatro equações. Assim,

- $k$  é o índice do primeiro coeficiente negativo,
- $n$  é o grau do polinômio,
- $B$  é o valor absoluto do maior coeficiente negativo em módulo,
- $L_i$  é o limite superior das raízes positivas de  $P_i(x) = 0$  dado pelo Teorema 6.3 e
- $L_\xi$  são os limites superiores e inferiores das raízes positivas e negativas de  $P(x) = 0$ , sendo que  $L_\xi(P) = L$ ,  $L_\xi(P_1) = 1/L_1$ ,  $L_\xi(P_2) = -L_2$  e  $L_\xi(P_3) = -1/L_3$ .

Exemplo 6.12 Calcular os limites das raízes de  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$  do Exemplo 6.11 usando o dispositivo prático.

$n=4$	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$
$c_4$	1	24	1	24
$c_3$	2	-14	-2	14
$c_2$	-13	-13	-13	-13
$c_1$	-14	2	14	-2
$c_0$	24	1	24	1
$k$	2	3	3	2
$n-k$	2	1	1	2
$B$	-14	-14	-13	-13
$L_i$	4,74	1,58	14	1,74
$L_\xi$	4,74	0,63	-14	-0,58

O algoritmo mostrado na Figura 6.4, baseado no dispositivo prático, determina os limites das raízes reais de uma equação polinomial a partir dos seus coeficientes. Os parâmetros de entrada são o grau  $n$  do polinômio e o vetor  $c$  dos coeficientes, e  $P(x) = c(1)x^n + c(2)x^{n-1} + \dots + c(n)x + c(n+1)$ . O parâmetro de saída é o vetor  $L$  contendo os limites das raízes reais, de forma que  $L(1) = 1/L_1$ ,  $L(2) = L$ ,  $L(3) = -L_2$  e  $L(4) = -1/L_3$ .

Exemplo 6.13 Calcular os limites das raízes reais da equação polinomial  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$  do Exemplo 6.11 usando o algoritmo da Figura 6.4.

```

Algoritmo LimitesRaízes
{ Objetivos Achar os limites das raízes reais de uma equação polinomial }
parâmetros de entrada  $n, c$  { grau do polinômio e coeficientes, sendo }
{  $P(x) = c(1)x^n + c(2)x^{n-1} + \dots + c(n)x + c(n+1)$  }
parâmetros de saída  $L$ 
{ limites inferior e superior das raízes positivas e negativas, respectivamente }
se  $c(1) = 0$  então escreva "coeficiente  $c(1)$  nulo", abandone, fimse
 $t \leftarrow n + 1$ ;  $c(t+1) \leftarrow 0$ 
repita { se  $c(n+1)$  for nulo, então o polinômio é deflacionado }
  se  $c(t) \neq 0$  então interrompa, fimse;  $t \leftarrow t - 1$ 
fimrepita
{ cálculo dos quatro limites das raízes reais }
para  $i \leftarrow 1$  até 4 faça
  se  $i = 2$  ou  $i = 4$  então { inversão da ordem dos coeficientes }
    para  $j \leftarrow 1$  até  $t/2$  faça
       $Aux \leftarrow c(j)$ ;  $c(j) \leftarrow c(t-j+1)$ ;  $c(t-j+1) \leftarrow Aux$ 
    fimpara
  senão
    se  $i = 3$  então
      { reinversão da ordem e troca de sinais dos coeficientes }
      para  $j \leftarrow 1$  até  $t/2$  faça
         $Aux \leftarrow c(j)$ ;  $c(j) \leftarrow c(t-j+1)$ ;  $c(t-j+1) \leftarrow Aux$ 
      fimpara
      para  $j \leftarrow t-1$  até 1 passo -2 faça  $c(j) \leftarrow -c(j)$ , fimpara
    fimse
  fimse
  { se  $c(1)$  for negativo, então é trocado o sinal de todos os coeficientes }
  se  $c(1) < 0$  então
    para  $j \leftarrow 1$  até  $t$  faça  $c(j) \leftarrow -c(j)$ , fimpara
  fimse
   $k \leftarrow 2$  { cálculo de  $k$ , o maior índice dos coeficientes negativos }
  repita
    se  $c(k) < 0$  ou  $k > t$  então interrompa, fimse
     $k \leftarrow k + 1$ 
  fimrepita { cálculo de  $B$ , o maior coeficiente negativo em módulo }
  se  $k \leq t$  então
     $B \leftarrow 0$ 
    para  $j \leftarrow 2$  até  $t$  faça
      se  $c(j) < 0$  e  $\text{abs}(c(j)) > B$  então  $B \leftarrow \text{abs}(c(j))$ , fimse
    fimpara
    { limite das raízes positivas de  $P(x) = 0$  e das equações auxiliares }
     $L(j) \leftarrow 1 + \sqrt[k]{B/c(1)}$ 
  senão
     $L(j) \leftarrow 10^{100}$ 
  fimse
  fimpara { limites das raízes positivas e negativas de  $P(x) = 0$  }
   $Aux \leftarrow L(1)$ ;  $L(1) \leftarrow 1/L(2)$ ;  $L(2) \leftarrow Aux$ ;  $L(3) \leftarrow -L(3)$ ;  $L(4) \leftarrow -1/L(4)$ 
fimalgoritmo

```

Figura 6.4 Algoritmo para achar os limites das raízes de equação polinomial.

(Ver significado da função  $\text{abs}$  na Tabela 1.1, na página 6.)

```
% Os parametros de entrada
n = 4
c = 1    2   -13  -14   24
% produzem os resultados
L = 0.6316  4.7417 -14.0000 -0.5760
```

Pelo gráfico mostrado na Figura 6.5(a), observa-se que o intervalo original pode ser reduzido a  $[-5, 4]$ . A Figura 6.5(b) revela claramente as quatro raízes isoladas nos intervalos  $[-5, -3]$ ,  $[-3, -1]$ ,  $[0, 2]$  e  $[2, 4]$ .

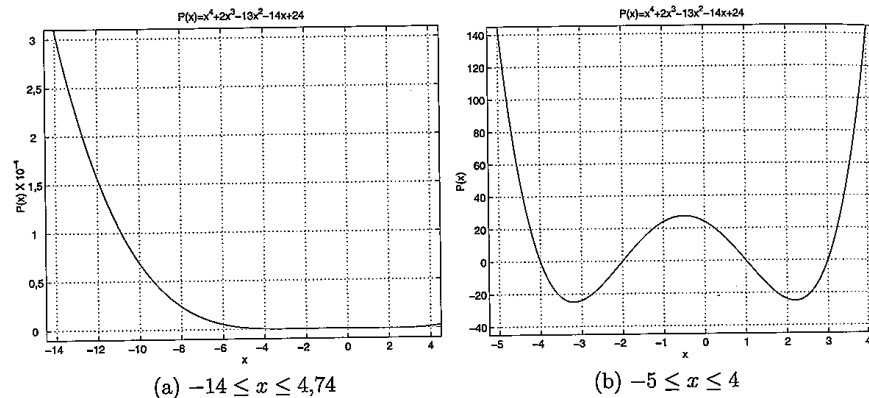


Figura 6.5 Gráficos do polinômio  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24$ .

### Número de raízes reais

A Regra de sinais de Descartes [10] é uma maneira simples para determinar o número de raízes reais de uma equação algébrica. Sabendo-se os limites e o número das raízes reais, a tarefa de isolar essas raízes ficará grandemente facilitada.

**Teorema 6.4 (Regra de sinais de Descartes)** O número de raízes reais positivas  $n^+$  de  $P(x) = 0$  é igual ao número de variações de sinais na sequência dos coeficientes ou é menor que este número por um inteiro par, sendo as raízes contadas de acordo com a sua multiplicidade e não sendo considerados os coeficientes nulos.

**Corolário 6.2** Se  $P(x) = 0$  não possuir coeficientes nulos, então o número de raízes reais negativas  $n^-$  (contando multiplicidades) é igual ao número de permanências de sinais na sequência dos coeficientes ou é menor que este número por um inteiro par.

A Regra de sinais de Descartes consegue discernir entre as raízes positivas e negativas; no entanto, não consegue separar as raízes reais das complexas, as quais aparecem em pares conjugados, conforme o Teorema 6.2. Por exemplo, se o número de variações de sinais for 5, então  $n^+ = 5$  ou 3 ou 1.

Exemplo 6.14 Para  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$ , tem-se que

$$n^+ = 2 \text{ ou } 0, \text{ e } n^- = 2 \text{ ou } 0.$$

Para essa equação, se existirem duas raízes positivas, elas satisfarão a  $0,63 \leq \xi^+ \leq 4,74$  e, se houver duas negativas, elas estarão no intervalo  $-14 \leq \xi^- \leq -0,58$  (ver Exemplo 6.11 e Figura 6.5(b)).

Portanto, combinando a Regra de sinais de Descartes e o Teorema de Lagrange, conseguem-se importantes informações para o pleno isolamento das raízes, como será visto mais adiante.

Exemplo 6.15 Calcular os limites e o número de raízes reais de  $P(x) = x^3 - 3x^2 - 6x + 8 = 0$ .

Para aplicar o Teorema 6.3 e obter os  $L_i$ 's, o coeficiente  $c_n = c_3$  deve ser positivo em cada um dos quatro polinômios. Como isso não é verdadeiro para  $P_2(x)$ , então todos os coeficientes desse polinômio devem ter os seus sinais trocados. Os zeros do novo polinômio serão iguais aos do polinômio original e, por conseguinte, os dois polinômios terão o mesmo  $L_2$ .

$n=3$	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$
$c_3$	1	8	-1	8
$c_2$	-3	-6	-3	6
$c_1$	-6	-3	6	-3
$c_0$	8	1	8	-1
$k$				
$n-k$				
$B$				
$L_i$				
$L_\xi$				

Trocar sinal  
de  $P_2(x)$

$n=3$	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$
$c_3$	1	8	1	8
$c_2$	-3	-6	3	6
$c_1$	-6	-3	-6	-3
$c_0$	8	1	-8	-1
$k$	2	2	1	1
$n-k$	1	1	2	2
$B$	-6	-6	-8	-3
$L_i$	7	1,75	3,83	1,61
$L_\xi$	7	0,57	-3,83	-0,62

Os limites das raízes são  $0,57 \leq \xi^+ \leq 7$  e  $-3,83 \leq \xi^- \leq -0,62$  e o número de raízes reais são  $n^+ = 2$  ou 0 e  $n^- = 1$ , significando que existe uma raiz real negativa e as outras duas serão reais positivas ou complexas.

Exemplo 6.16 Achar os limites e o número de raízes reais de  $P(x) = x^6 - 5x^5 + 7x^4 + 19x^3 - 98x^2 - 104x = 0$ .

Como  $c_0 = 0$ , tem-se, pelas relações de Girard, que existe pelo menos uma raiz nula. Para obter as equações auxiliares,  $P(x)$  deve ser deflacionado pela divisão por  $(x - 0)$ , resultando em  $P(x) = x^5 - 5x^4 + 7x^3 + 19x^2 - 98x - 104 = 0$ . Este novo  $P(x)$  terá todas as raízes do antigo, com exceção da raiz nula. Neste exemplo, como  $c_5 < 0$  para  $P_i(x)$ ,  $i = 1, 2, 3$ , os sinais de todos os coeficientes de  $P_1(x)$ ,  $P_2(x)$  e  $P_3(x)$  devem ser trocados para que se possa aplicar o Teorema 6.3.

$n=5$	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$
$c_5$	1	-104	-1	-104
$c_4$	-5	-98	-5	98
$c_3$	7	19	-7	19
$c_2$	19	7	19	-7
$c_1$	-98	-5	98	-5
$c_0$	-104	1	-104	-1
$k$				
$n-k$				
$B$				
$L_i$				
$L_\xi$				

Trocar sinal de  
 $P_1(x), P_2(x), P_3(x)$

$n=5$	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$
$c_5$	1	104	1	104
$c_4$	-5	98	5	-98
$c_3$	7	-19	7	-19
$c_2$	19	-7	-19	7
$c_1$	-98	5	-98	5
$c_0$	-104	-1	104	1
$k$	4	3	2	4
$n-k$	1	2	3	1
$B$	-104	-19	-98	-98
$L_i$	105	1,43	5,61	1,94
$L_\xi$	105	0,70	-5,61	-0,51

Assim, os limites das raízes são  $0,70 \leq \xi^+ \leq 105$  e  $-5,61 \leq \xi^- \leq -0,51$  e o número de raízes reais são  $n^+ = 3$  ou 1 e  $n^- = 2$  ou 0, garantindo pelo menos uma raiz real positiva. ■

### 6.1.2 Equações transcendentais

Infelizmente, as equações transcendentais não dispõem de teoremas, como visto para as algébricas, que forneçam informações sobre os limites e o número de raízes reais. Se as algébricas têm um número finito de raízes, o mesmo, às vezes, não ocorre com uma transcendente que pode ter um número infinito de raízes, como, por exemplo,  $f(x) = \sin(x) = 0$  ou mesmo não ter raízes como  $f(x) = \sin(x) - 2 = 0$ .

O método gráfico é a maneira mais simples para achar um intervalo que contenha uma única raiz. Ele consiste em fazer um esboço da função no intervalo de interesse. No entanto, a maior dificuldade com as equações transcendentais consiste, justamente, em determinar este intervalo, já que não existe um "Teorema de Lagrange" para elas. O que se faz na prática é usar a intuição, o conhecimento a respeito da função e, como ocorre na maioria das vezes, o método da tentativa e erro, ou seja, a partir de um intervalo inicial, tenta-se um outro que isole a raiz observando-se a forma da curva.

O algoritmo descrito na Figura 6.6 fornece um intervalo  $[a, b]$ , no qual uma função  $f(x)$  troca de sinal, ou seja,  $f(a)f(b) < 0$ . Este intervalo é expandido a partir de um dado valor  $z$ . Apesar de a raiz não estar necessariamente isolada, pois pode haver um número ímpar de raízes no intervalo, o algoritmo fornecerá um intervalo de partida para o isolamento de uma raiz.

O parâmetro de entrada é o ponto  $z$  a partir do qual o intervalo será gerado. Os parâmetros de saída são o limite inferior  $a$  e superior  $b$  do intervalo procurado e a condição de erro  $CondErro$ . Se  $CondErro = 0$  significa que  $f(a)f(b) \leq 0$ , ou seja, existe um número ímpar de raízes no intervalo  $[a, b]$ . Mas, se  $CondErro = 1$ , então  $f(a)f(b) > 0$ , significando que o intervalo  $[a, b]$  não tem raízes ou tem um número par de raízes. A função  $f(x)$  deve ser especificada em cada caso. No entanto, a maneira de fazer essa especificação vai depender da linguagem de programação utilizada para implementar o algoritmo **TrocaSinal**.

```

Algoritmo TrocaSinal
{ Objetivo: Achar um intervalo  $[a, b]$  onde uma função troca de sinal }
parâmetros de entrada z
  { ponto a partir do qual o intervalo será gerado }
parâmetros de saída a, b, CondErro
  { limite inferior e superior do intervalo e condição de erro, sendo }
  { CondErro = 0 se  $f(a)f(b) \leq 0$  e CondErro = 1 se  $f(a)f(b) > 0$ . }
se z = 0 então
  a ← -0,05; b ← 0,05
senão
  a ← 0,95 * z; b ← 1,05 * z
fimse
iter ← 0; Aurco ← 2 / (raiz2(5) - 1)
Fa ← f(a); Fb ← f(b) { avaliar a função em a e b }
escreva iter, a, b, Fa, Fb
repita
  se Fa * Fb ≤ 0 ou iter ≥ 20 então interrompa, fimse
  iter ← iter + 1
  se abs(Fa) < abs(Fb) então
    a ← a + Aurco * (b - a)
    Fa ← f(a) { avaliar a função em a }
  senão
    b ← b + Aurco * (b - a)
    Fb ← f(b) { avaliar a função em b }
  fimse
  escreva iter, a, b, Fa, Fb
fimrepita
se Fa * Fb ≤ 0 então
  CondErro ← 0
senão
  CondErro ← 1
fimse
finalgoritmo

```

Figura 6.6 Algoritmo para achar um intervalo onde uma função troca de sinal.

(Ver significado das funções abs e raiz2 na Tabela 1.1, na página 6.)

Exemplo 6.17 Achar um intervalo, a partir de  $z = 5$ , onde  $f(x) = 2x^3 - \cos(x+1) - 3$  troca de sinal, utilizando o algoritmo da Figura 6.6.

```

% 0 parametro de entrada
z = 5
% produz os resultados

```



Determinação de intervalo onde ocorre troca de sinal

iter	a	b	Fa	Fb
0	4.7500	5.2500	210.4826	285.4068
1	3.9410	5.2500	119.1909	285.4068
2	1.8229	5.2500	10.0655	285.4068
3	-3.7221	5.2500	-105.2218	285.4068

a = -3.7221

b = 5.2500

CondErro = 0

De fato, a função muda de sinal no intervalo  $[-3,7221; 5,2500]$ , pois  $f(-3,7221) = -105,2218$  e  $f(5,2500) = 285,4068$ . Pelo gráfico mostrado na Figura 6.7(a), percebe-se que o intervalo  $[-1, 2]$  contém uma única raiz. A curva pode ser esboçada neste intervalo menor, Figura 6.7(b), para examinar o gráfico com uma melhor definição.

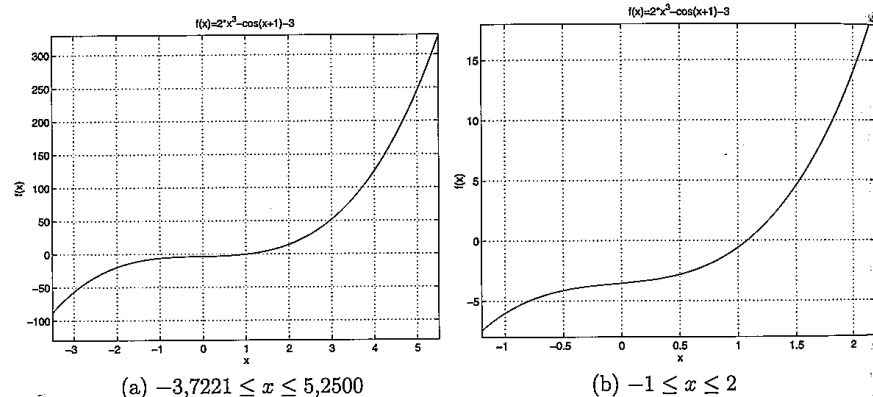


Figura 6.7 Esboços da função  $f(x) = 2x^3 - \cos(x+1) - 3$  em dois intervalos.

Exemplo 6.18 Isolar, graficamente, os zeros da função  $f(x) = 0,05x^3 - 0,4x^2 + 3\sin(x)x$ .

Inicialmente, pode-se definir um intervalo amplo para ter uma visão do comportamento da função. A Figura 6.8(a) apresenta um esboço para  $-20 \leq x \leq 20$ . Um melhor detalhamento pode ser obtido usando  $-4 \leq x \leq 12$ , o qual é mostrado na Figura 6.8(b). Podem ser observadas seis raízes no intervalo  $[-4, 12]$  e cada uma destas raízes estará isolada nos seguintes intervalos:  $[-4, -2]$ ,  $[-1, 1]$ ,  $[2, 4]$ ,  $[6, 8]$ ,  $[8, 10]$  e  $[10, 12]$ .

O método gráfico também pode ser usado para isolar as raízes de equações algébricas. Conhecendo os limites e o número de raízes reais pelo Teorema de Lagrange e pela Regra de sinais de Descartes, o isolamento por gráfico fica enormemente facilitado.

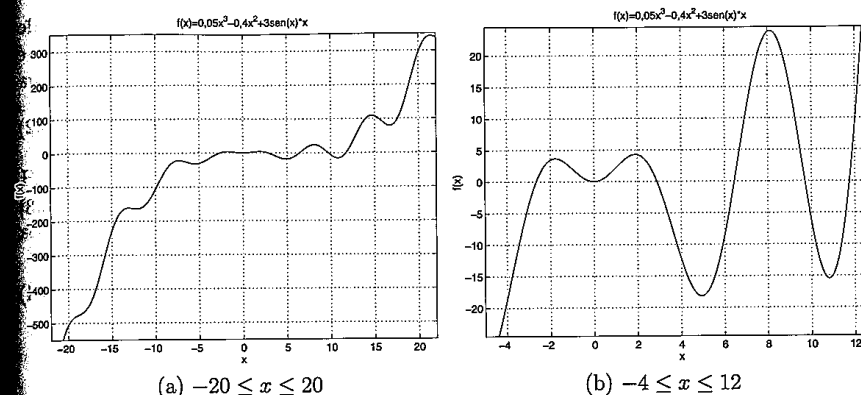


Figura 6.8 Esboços da função  $f(x) = 0,05x^3 - 0,4x^2 + 3\sin(x)x$ .

### 6.1.3 Convergência da raiz

Se uma raiz  $\xi$  já estiver isolada em um dado intervalo  $[a, b]$ , então a próxima etapa consiste em gerar uma sequência  $\{x_0, x_1, x_2, \dots, x_k, \dots, \xi\} \in [a, b]$  que convirja para esta raiz exata  $\xi$  de  $f(x) = 0$ .

#### Critério de parada

Antes de serem abordados os métodos para produzirem esta sequência, é necessário definir um critério de parada, ou seja, quando se deve interromper a geração da sequência. A demonstração do teorema abaixo é dada por Demidovich e Maron [10].

**Teorema 6.5** Sejam  $\xi$  uma raiz exata e  $x_k$  uma raiz aproximada de  $f(x) = 0$ , sendo  $\xi$  e  $x_k \in [a, b]$  e  $|f'(x)| \geq m > 0$  para  $a \leq x \leq b$ , com

$$m = \min_{a \leq x \leq b} |f'(x)|.$$

Então, o erro absoluto satisfaz

$$|x_k - \xi| \leq \frac{|f(x_k)|}{m}.$$

Exemplo 6.19 Avaliar o erro absoluto cometido ao considerar  $x_k = 2,23$  como aproximação da raiz positiva de  $f(x) = x^2 - 5 = 0$  no intervalo  $[2, 3]$ .

$$m = \min_{2 \leq x \leq 3} |2x| = 4.$$

Assim,

$$|2,23 - \xi| \leq \frac{0,0271}{4} = 0,0068 \rightarrow$$

$$2,23 - 0,0068 \leq \xi \leq 2,23 + 0,0068 \quad (\xi = \sqrt{5} \approx 2,2361).$$

O Teorema 6.5 é de aplicação muito restrita, haja vista que ele requer que seja avaliado o mínimo da derivada primeira da função  $f(x)$  em questão. Na prática, a sequência é interrompida quando seus valores satisfizerem a pelo menos um dos critérios

$$|x_k - x_{k-1}| \leq \varepsilon, \quad (6.2)$$

$$\left| \frac{x_k - x_{k-1}}{x_k} \right| \leq \varepsilon, \quad (6.3)$$

$$|f(x_k)| \leq \varepsilon \quad (6.4)$$

onde  $\varepsilon$  é a tolerância fornecida.

### Ordem de convergência

Uma questão fundamental é definir a rapidez com que a sequência gerada por um dado método,  $\{x_0, x_1, x_2, \dots, x_k, \dots\}$ , converge para a raiz exata  $\xi$ . Para tal, seja o erro da  $k$ -ésima iteração definido por

$$\varepsilon_k = x_k - \xi \quad (6.5)$$

ou seja, a diferença entre a raiz  $\xi$  e a sua estimativa  $x_k$ . Um critério para avaliar a convergência é

$$\lim_{k \rightarrow \infty} |\varepsilon_{k+1}| = K|\varepsilon_k|^\gamma \quad (6.6)$$

onde  $K$  é a constante de erro assintótico e  $\gamma$  é a ordem de convergência do método gerador da sequência. Por causa disso, quanto maior for o valor de  $\gamma$ , mais rápida a sequência convergirá para a raiz da equação.

## 6.2 Método da bisseção

Seja uma função  $f(x)$  contínua no intervalo  $[a, b]$ , sendo  $\xi$  a única raiz de  $f(x) = 0$  neste intervalo. O método da bisseção consiste, simplesmente, em subdividir o intervalo ao meio a cada iteração e manter o subintervalo que contenha a raiz, ou seja, aquele em que  $f(x)$  tenha sinais opostos nos extremos. Deste modo, obtém-se uma sequência de intervalos encaixados  $\{[a_1, b_1], [a_2, b_2], [a_3, b_3], \dots, [a_k, b_k]\}$  nos quais

$$f(a_i)f(b_i) < 0, \quad i = 1, 2, \dots, k. \quad (6.7)$$

Na  $k$ -ésima iteração, tem-se

$$b_k - a_k = \frac{b - a}{2^k}. \quad (6.8)$$

A sequência  $\{a_1, a_2, a_3, \dots, a_k\}$  é monotônica não decrescente limitada, enquanto  $\{b_1, b_2, b_3, \dots, b_k\}$  é monotônica não crescente limitada. Então, por (6.8), essas duas sequências possuem um limite comum  $\xi$

$$\lim_{k \rightarrow \infty} a_k = \lim_{k \rightarrow \infty} b_k = \xi.$$

Passando ao limite da desigualdade (6.7) com  $k \rightarrow \infty$  e considerando que  $f(x)$  é contínua, então  $[f(\xi)]^2 \leq 0 \rightarrow f(\xi) = 0$ , ou seja,  $\xi$  é uma raiz de  $f(x) = 0$ . Os primeiros pontos da sequência  $\{x_1, x_2, x_3, \dots, x_k, \dots\}$  são mostrados na Figura 6.9.

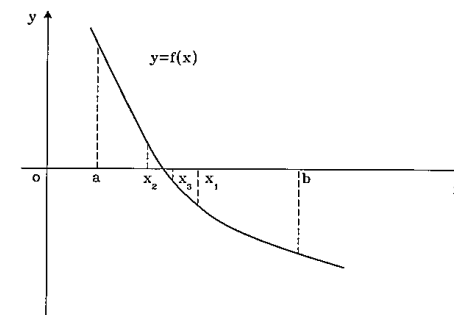


Figura 6.9 Interpretação gráfica do método da bisseção.

Uma das grandes vantagens do método da bisseção é que ele tem convergência garantida se  $f(x)$  for contínua em  $[a, b]$  e se  $\xi \in [a, b]$ . Ele é também um dos poucos métodos nos quais é possível determinar *a priori* o número de iterações necessárias para calcular a raiz com uma tolerância  $\varepsilon$  a partir de um intervalo  $[a, b]$ . Para tanto, substituindo  $x_k = (b_k - a_k)/2$  em (6.8), tem-se que

$$|x_k - x_{k-1}| = \frac{b - a}{2^{k+1}}.$$

Utilizando o critério (6.2),

$$\frac{b - a}{2^{k+1}} \leq \varepsilon,$$

então o número de iterações para calcular uma raiz no intervalo  $[a, b]$  com tolerância  $\varepsilon$  é

$$k \geq \log_2 \left( \frac{b - a}{\varepsilon} \right) - 1. \quad (6.9)$$

O algoritmo mostrado na Figura 6.10 calcula, pelo método da bisseção, a raiz de uma equação  $f(x) = 0$  contida no intervalo  $[a, b]$ , com tolerância  $\varepsilon$  usando os critérios (6.2) e

(6.4). Os parâmetros de entrada são o limite inferior  $a$  e o superior  $b$  do intervalo que contém a raiz, a tolerância  $Toler$  para o cálculo da raiz e o número máximo de iterações  $IterMax$ . A função  $f(x)$  deve ser especificada de acordo com a linguagem de programação utilizada. Os parâmetros de saída são a raiz da equação  $Raiz$ , o número de iterações gastas  $Iter$  e a condição de erro  $CondErro$ . Se  $CondErro = 0$  significa que a raiz foi encontrada e  $CondErro = 1$  indica que a raiz não foi encontrada com a tolerância e o número máximo de iterações fornecidos.

Exemplo 6.20 Calcular a raiz de  $f(x) = 2x^3 - \cos(x+1) - 3 = 0$  do Exemplo 6.17, que está no intervalo  $[-1, 2]$ , com  $\varepsilon \leq 0,01$  pelo algoritmo da Figura 6.10.

% Os parametros de entrada

a = -1

b = 2

Toler = 0.0100

IterMax = 100

% produzem os resultados

Calculo de raiz de equacao pelo metodo da bissecao							
iter	a	Fa	b	Fb	x	Fx	Delta_x
0	-1.00000	-6.00000	2.00000	13.98999	0.50000	-2.8207e+00	1.50000
1	0.50000	-2.82074	2.00000	13.98999	1.25000	1.5344e+00	0.75000
2	0.50000	-2.82074	1.25000	13.98999	0.87500	-1.3606e+00	0.37500
3	0.87500	-1.36062	1.25000	13.98999	1.06250	-1.2895e-01	0.18750
4	1.06250	-0.12895	1.25000	13.98999	1.15625	6.4419e-01	0.09375
5	1.06250	-0.12895	1.15625	13.98999	1.10938	2.4356e-01	0.04688
6	1.06250	-0.12895	1.10938	13.98999	1.08594	5.3864e-02	0.02344
7	1.06250	-0.12895	1.08594	13.98999	1.07422	-3.8393e-02	0.01172
8	1.07422	-0.03839	1.08594	13.98999	1.08008	7.5211e-03	0.00586

Raiz = 1.08008

Iter = 8

CondErro = 0

A raiz é  $\xi \approx x_8 = 1,08008$ . Por (6.9), o número de iterações  $k \geq \log((2 - (-1))/0,01) - 1 \approx 7,23$ , ou seja, para alcançar o critério de parada (6.2)  $|x_k - x_{k-1}| \leq \varepsilon$  (ver coluna Delta\_x) foram necessárias 8 iterações. Poderiam ser gastas mais iterações para atender ao outro critério de parada (6.4)  $|f(x_k)| \leq \varepsilon$ .

Exemplo 6.21 Determinar a maior raiz de  $f(x) = 0,05x^3 - 0,4x^2 + 3 \sin(x)x = 0$  com  $\varepsilon \leq 0,005$ , usando o algoritmo da Figura 6.10.

Pela Figura 6.8(b) do Exemplo 6.18, tem-se que  $\xi \in [10, 12]$ .

% Os parametros de entrada

a = 10

b = 12

Toler = 0.0050

IterMax = 100

% produzem os resultados

#### Algoritmo Bisseção

{ Objetivo: Calcular a raiz de uma equação pelo método da bisseção }

parâmetros de entrada  $a, b, Toler, IterMax$

{ limite inferior, limite superior, tolerância e número máximo de iterações }

parâmetros de saída  $Raiz, Iter, CondErro$

{ raiz, número de iterações gastas e condição de erro, sendo }

{  $CondErro = 0$  se a raiz foi encontrada e  $CondErro = 1$  em caso contrário. }

$Fa \leftarrow f(a); Fb \leftarrow f(b)$  { avaliar a função em  $a$  e  $b$  }

se  $Fa * Fb > 0$  então

escreva "função não muda de sinal nos extremos do intervalo dado"

abandone

fimse

$\Delta x \leftarrow \text{abs}(b - a)/2; Iter \leftarrow 0$

repita

$x \leftarrow (a + b)/2; Fx \leftarrow f(x)$  { avaliar a função em  $x$  }

escreva  $Iter, a, Fa, b, Fb, x, Fx, \Delta x$

se  $(\Delta x \leq Toler \text{ e } \text{abs}(Fx) \leq Toler)$  ou  $Iter \geq IterMax$  então

interrompa

fimse

se  $Fa * Fx > 0$  então

$a \leftarrow x; Fa \leftarrow Fx$

senão

$b \leftarrow x$

fimse

$\Delta x \leftarrow \Delta x/2; Iter \leftarrow Iter + 1$

fimrepita

$Raiz \leftarrow x$

{ teste de convergência }

se  $\Delta x \leq Toler \text{ e } \text{abs}(Fx) \leq Toler$  então

$CondErro \leftarrow 0$

senão

$CondErro \leftarrow 1$

fimse

finalgoritmo

Figura 6.10 Algoritmo do método da bisseção.

(Ver significado da função abs na Tabela 1.1, na página 6.)

Calculo de raiz de equacao pelo metodo da bissecao							
iter	a	Fa	b	Fb	x	Fx	Delta_x
0	10.00000	-6.32063	12.00000	9.48337	11.00000	-1.4850e+01	1.00000
1	11.00000	-14.84968	12.00000	9.48337	11.50000	-7.0594e+00	0.50000
2	11.50000	-7.05935	12.00000	9.48337	11.75000	2.0128e-01	0.25000
3	11.50000	-7.05935	11.75000	9.48337	11.62500	-3.6975e+00	0.12500

4	11.62500	-3.69752	11.75000	9.48337	11.68750	-1.8136e+00	0.06250
5	11.68750	-1.81359	11.75000	9.48337	11.71875	-8.2229e-01	0.03125
6	11.71875	-0.82229	11.75000	9.48337	11.73438	-3.1451e-01	0.01562
7	11.73438	-0.31451	11.75000	9.48337	11.74219	-5.7611e-02	0.00781
8	11.74219	-0.05761	11.75000	9.48337	11.74609	7.1585e-02	0.00391
9	11.74219	-0.05761	11.74609	9.48337	11.74414	6.9247e-03	0.00195
10	11.74219	-0.05761	11.74414	9.48337	11.74316	-2.5359e-02	0.00098
11	11.74316	-0.02536	11.74414	9.48337	11.74365	-9.2209e-03	0.00049
12	11.74365	-0.00922	11.74414	9.48337	11.74390	-1.1491e-03	0.00024

Raiz = 11.74390  
 Iter = 12  
 CondErro = 0

A raiz da equação é  $\xi \approx x_{12} = 11,74390$ .

Apesar de o método da bisseção ser robusto, ele não é eficiente devido à sua convergência lenta. Pode ser observado que  $f(x)$  não decresce, monotonicamente, como no último exemplo, no qual  $|f(x_3)| > |f(x_2)|$ . Isso ocorre porque somente o sinal de  $f(x_{k-1})$  é usado para o cálculo do próximo  $x_k$ , sem levar em consideração o seu valor.

O método da bisseção é mais usado para reduzir o intervalo antes de usar um outro método de convergência mais rápida, dos quais alguns serão vistos a seguir.

### 6.3 Métodos baseados em aproximação linear

A velocidade de convergência da sequência  $\{x_i\}$  para a raiz  $\xi$  de uma equação  $f(x) = 0$  pode ser aumentada usando-se um esquema diferente da bisseção. Um esquema consiste em aproximar  $f(x)$  por um polinômio linear no intervalo de interesse  $[x_0, x_1]$ . Se o intervalo for pequeno, essa aproximação é válida para a maioria das funções. Deste modo, uma estimativa da raiz  $\xi$  é tomada como o valor onde a reta cruza o eixo das abscissas.

A equação do polinômio de grau 1 que passa pelos pontos de coordenadas  $[x_0, f(x_0)]$  e  $[x_1, f(x_1)]$  é dada por (3.1), conforme mostrado no Capítulo 3,

$$y = f(x_1) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_1).$$

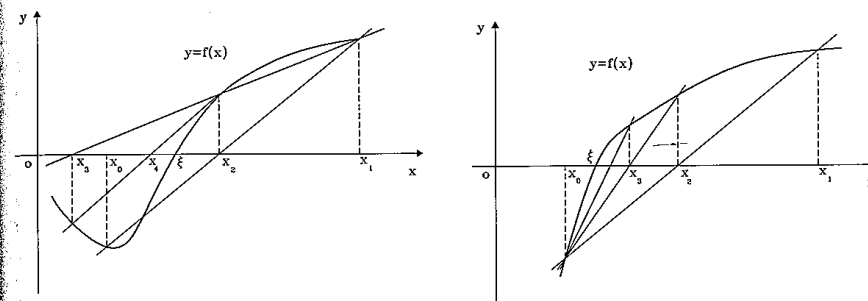
Então o valor da abscissa  $x_2$ , para o qual  $y = 0$ , é tomado como uma aproximação da raiz

$$x_2 = x_1 - \frac{f(x_1)}{f(x_1) - f(x_0)}(x_1 - x_0).$$

Na próxima iteração, um dos pontos extremos do intervalo  $[x_0, x_1]$  será substituído por  $x_2$ , que é uma melhor estimativa da raiz  $\xi$ . O processo se repete, gerando, desta forma, uma sequência  $\{x_i\}$  até que o critério de parada seja satisfeito. Existe uma família de métodos baseados nesta aproximação linear, dentre os quais o método da secante, o da *regula falsi* (posição falsa) e o pégas.

#### 6.3.1 Método da secante

O método da secante usa os pontos obtidos nas duas últimas iterações como pontos-base por onde passará o polinômio linear, conforme mostra a Figura 6.11(a).



(a) Método da secante

(b) Método da *regula falsi*

Figura 6.11 Métodos para cálculo da raiz baseados em aproximação linear.

O algoritmo da Figura 6.12 determina, pelo método da secante, a raiz de uma equação  $f(x) = 0$  contida em um intervalo  $[a, b]$ , com tolerância  $\epsilon$  usando os critérios (6.2) e (6.4). Os parâmetros de entrada são o limite inferior  $a$  e o superior  $b$  do intervalo que isola a raiz, a tolerância *Toler* para o cálculo da raiz e o número máximo de iterações *IterMax*. A função  $f(x)$  deve ser especificada de acordo com a linguagem de programação utilizada. Os parâmetros de saída são a raiz da equação *Raiz*, o número de iterações gastas *Iter* e a condição de erro *CondErro*, em que *CondErro* = 0 significa que a raiz foi calculada com sucesso e *CondErro* = 1 indica que a raiz não foi encontrada com a tolerância e o número máximo de iterações dados.

Durante as iterações, o ponto  $(a, Fa)$  é substituído por  $(b, Fb)$  e  $(b, Fb)$  é trocado pelo recém-calculado  $(x, Fx)$ . Por essa razão, antes das iterações, é verificado se  $|Fa| < |Fb|$  de modo a ser abandonado, na primeira iteração, o ponto mais distante da raiz.

Exemplo 6.22 Determinar a raiz de  $f(x) = 2x^3 - \cos(x + 1) - 3 = 0$  do Exemplo 6.17 pelo método da secante mostrado na Figura 6.12, com  $\epsilon \leq 0,01$ , sabendo-se que  $\xi \in [-1, 2]$ .

```
% Os parametros de entrada
a = -1
b = 2
Toler = 0.0100
IterMax = 100
% produzem os resultados
```

```

Algoritmo Secante
{ Objetivo: Calcular a raiz de uma equação pelo método da secante }
parâmetros de entrada a, b, Toler, IterMax
{ limite inferior, limite superior, tolerância e número máximo de iterações }
parâmetros de saída Raiz, Iter, CondErro
{ raiz, número de iterações gastas e condição de erro, sendo }
{ CondErro = 0 se a raiz foi encontrada e CondErro = 1 em caso contrário. }
Fa ← f(a); Fb ← f(b) { avaliar a função em a e b }
se abs(Fa) < abs(Fb) então
    t ← a; a ← b; b ← t; t ← Fa; Fa ← Fb; Fb ← t
fimse
Iter ← 0; x ← b; Fx ← Fb
repita
    DeltaX ← -Fx/(Fb - Fa) * (b - a)
    x ← x + DeltaX; Fx ← f(x) { avaliar a função em x }
    escreva Iter, a, Fa, b, Fb, x, Fx, DeltaX
    se (abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler) ou Iter ≥ IterMax então
        interrompa
    fimse
    a ← b; Fa ← Fb; b ← x; Fb ← Fx; Iter ← Iter + 1
fimrepita
Raiz ← x
{ teste de convergência }
se abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler então
    CondErro ← 0
senão
    CondErro ← 1
fimse
fim algoritmo

```

Figura 6.12 Algoritmo do método da secante.

(Ver significado da função *abs* na Tabela 1.1, na página 6.)

iter	a	Fa	b	Fb	x	Fx	Delta_x
0	2.00000	13.98999	-1.00000	-6.00000	-0.09955	-3.623e+00	9.005e-01
1	-1.00000	-6.00000	-0.09955	-3.62323	1.27313	1.773e+00	1.373e+00
2	-0.09955	-3.62323	1.27313	1.77312	0.82210	-1.640e+00	-4.510e-01
3	1.27313	1.77312	0.82210	-1.64011	1.03883	-3.068e-01	2.167e-01
4	0.82210	-1.64011	1.03883	-0.30676	1.08869	7.576e-02	4.986e-02
5	1.03883	-0.30676	1.08869	0.07576	1.07881	-2.438e-03	-9.875e-03

*Raiz* = 1.07881  
*Iter* = 5  
*CondErro* = 0

Assim, a raiz da equação é  $\xi \approx x_5 = 1,07881$ .

O método da secante pode apresentar alguns problemas. Se a função não for, aproximadamente, linear no intervalo que contém a raiz, uma aproximação sucessiva pode sair deste intervalo como mostrado na Figura 6.11(a).

### 6.3.2 Método da *regula falsi*

Uma maneira de evitar problemas é garantir que a raiz esteja isolada no intervalo inicial e continue dentro dos novos intervalos gerados. O método da *regula falsi* retém o ponto no qual o valor da função tem sinal oposto ao valor da função no ponto mais recente, garantindo, desta forma, que a raiz continue isolada entre dois pontos, como mostrado na Figura 6.11(b).

O algoritmo do método da *regula falsi*, apresentado na Figura 6.13, encontra a raiz de uma equação  $f(x) = 0$  contida em um intervalo  $[a, b]$ , com tolerância  $\epsilon$  usando os critérios (6.2) e (6.4). Os parâmetros de entrada são o limite inferior  $a$  e o superior  $b$  do intervalo que contém a raiz, a tolerância *Toler* para o cálculo da raiz e o número máximo de iterações *IterMax*. A função  $f(x)$  deve ser especificada de acordo com a linguagem de programação adotada. Os parâmetros de saída são a raiz de  $f(x) = 0$  *Raiz*, o número de iterações gastas *Iter* e a condição de erro *CondErro*, em que *CondErro* = 0 mostra que a raiz foi calculada com êxito e *CondErro* = 1 indica que a raiz não foi encontrada com a tolerância e o número máximo de iterações fornecidos.

Exemplo 6.23 Achar a raiz de  $f(x) = 2x^3 - \cos(x + 1) - 3 = 0$  do Exemplo 6.17 usando o método da *regula falsi*, dado na Figura 6.13, com  $\epsilon \leq 0,01$ , sabendo-se que  $\xi \in [-1, 2]$ .

```

% Os parametros de entrada
a = -1
b = 2
Toler = 0.0100
IterMax = 100
% produzem os resultados
Calculo de raiz de equacao pelo metodo da regula falsi

```

iter	a	Fa	b	Fb	x	Fx	Delta_x
0	-1.00000	-6.00000	2.00000	13.98999	-0.09955	-3.623e+00	-2.100e+00
1	-0.09955	-3.62323	2.00000	13.98999	0.33235	-3.163e+00	4.319e-01
2	0.33235	-3.16277	2.00000	13.98999	0.63985	-2.407e+00	3.075e-01
3	0.63985	-2.40710	2.00000	13.98999	0.83952	-1.551e+00	1.997e-01
4	0.83952	-1.55114	2.00000	13.98999	0.95534	-8.810e-01	1.158e-01
5	0.95534	-0.88102	2.00000	13.98999	1.01723	-4.631e-01	6.189e-02
6	1.01723	-0.46306	2.00000	13.98999	1.04872	-2.333e-01	3.149e-02
7	1.04872	-0.23328	2.00000	13.98999	1.06432	-1.150e-01	1.560e-02
8	1.06432	-0.11498	2.00000	13.98999	1.07195	-5.607e-02	7.628e-03
9	1.07195	-0.05607	2.00000	13.98999	1.07565	-2.719e-02	3.704e-03
10	1.07565	-0.02719	2.00000	13.98999	1.07745	-1.315e-02	1.793e-03
11	1.07745	-0.01315	2.00000	13.98999	1.07831	-6.355e-03	8.666e-04

```

Raiz = 1.07831
Iter = 11
CondErro = 0

```

```

Algoritmo RegulaFalsi
{ Objetivo: Calcular a raiz de uma equação pelo método da regula falsi }
parâmetros de entrada a, b, Toler, IterMax
  { limite inferior, limite superior, tolerância e número máximo de iterações }
parâmetros de saída Raiz, Iter, CondErro
  { raiz, número de iterações gastas e condição de erro, sendo }
  { CondErro = 0 se a raiz foi encontrada e CondErro = 1 em caso contrário. }
  Fa ← f(a); Fb ← f(b); { avaliar a função em a e b }
  se Fa * Fb > 0 então
    escreva "função não muda de sinal nos extremos do intervalo dado"
    abandone
  fimse
  se Fa > 0 então
    t ← a; a ← b; b ← t; t ← Fa; Fa ← Fb; Fb ← t
  fimse
  Iter ← 0; x ← b; Fx ← Fb
  repita
    DeltaX ← -Fx / ((Fb - Fa) * (b - a))
    x ← x + DeltaX; Fx ← f(x); { avaliar a função em x }
    escreva Iter, a, Fa, b, Fb, x, Fx, DeltaX
    se (abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler) ou Iter ≥ IterMax então
      interrompa
    fimse
    se Fx < 0 então
      a ← x; Fa ← Fx
    senão
      b ← x; Fb ← Fx
    fimse; Iter ← Iter + 1
  fimrepita
  Raiz ← x
  { teste de convergência }
  se abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler então
    CondErro ← 0
  senão
    CondErro ← 1
  fimse
finalgoritmo

```

Figura 6.13 Algoritmo do método da *regula falsi*.

(Ver significado da função abs na Tabela 1.1, na página 6.)

A raiz da equação é  $\xi \approx x_{11} = 1,07831$ . Neste exemplo, a convergência para a raiz só se fez de um lado do intervalo, tornando este método mais lento que o da secante, embora mais robusto. Quanto mais longe o ponto fixo for da raiz, mais lenta será a convergência. ■

Conforme será visto mais adiante, o método da *regula falsi* tem uma ordem de convergência menor que o da secante porque o ponto mantido fixo não é geralmente um dos mais recentes.

### 6.3.3 Método pégaso

De modo similar aos outros métodos baseados em aproximação linear, no método pégaso [12] a sequência  $\{x_i\}$  é obtida pela fórmula de recorrência

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1}), \quad k = 1, 2, 3, \dots$$

Os pontos  $[x_{k-1}, f(x_{k-1})]$  e  $[x_k, f(x_k)]$  pelos quais será traçada a reta para obter  $x_{k+1}$  são escolhidos de modo que  $f(x_{k-1})$  e  $f(x_k)$  tenham sempre sinais opostos, garantindo assim que  $\xi \in [x_{k-1}, x_k]$ . Além do mais, o valor de  $f(x_{k-1})$  é reduzido por um fator igual a  $f(x_k)/(f(x_k) + f(x_{k+1}))$  de modo a evitar a retenção de um ponto, como ocorre na *regula falsi*. Deste modo, a reta pode ser traçada por um ponto não pertencente à curva de  $f(x)$ .

A Figura 6.14 ilustra graficamente o método pégaso. Deve ser observado que a estimativa  $x_4$  da raiz é obtida usando os pontos de coordenadas  $[x_3, f(x_3)]$  e  $[x_1, p]$ , sendo  $p = f(x_1) \times f(x_2)/(f(x_2) + f(x_3))$ , o qual não pertence à função  $f(x)$ . É claro, pela figura, que  $x_4$  é uma melhor aproximação da raiz do que  $x'_4$  que seria obtido pelo método da *regula falsi* (linha tracejada).

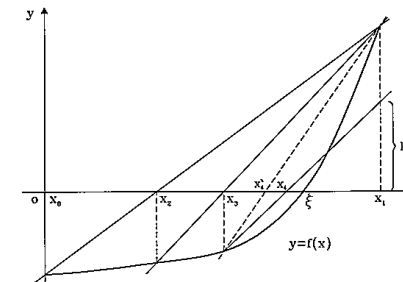


Figura 6.14 Interpretação gráfica do método pégaso.

O algoritmo da Figura 6.15 calcula, pelo método pégaso, uma raiz da equação  $f(x) = 0$  pertencente ao intervalo  $[a, b]$ , com tolerância  $\varepsilon$  usando os critérios (6.2) e (6.4). Os parâmetros de entrada são o limite inferior  $a$  e o superior  $b$  do intervalo que isola a raiz, a tolerância *Toler* para o cálculo da raiz e o número máximo de iterações *IterMax*. A função  $f(x)$  deve ser especificada de acordo com a linguagem de programação utilizada. Os parâmetros de saída são a raiz da equação *Raiz*, o número de iterações gastas *Iter* e a

condição de erro *CondErro*, em que *CondErro* = 0 indica que a raiz foi calculada com sucesso e *CondErro* = 1 avisa que a raiz não foi encontrada com a tolerância e o número máximo de iterações dados.

#### Algoritmo Pégaso

```
{ Objetivo: Calcular a raiz de uma equação pelo método pégaso }
parâmetros de entrada a, b, Toler, IterMax
{ limite inferior, limite superior, tolerância e número máximo de iterações }
parâmetros de saída Raiz, Iter, CondErro
{ raiz, número de iterações gastas e condição de erro, sendo }
{ CondErro = 0 se a raiz foi encontrada e CondErro = 1 em caso contrário. }
Fa ← f(a); Fb ← f(b); { avaliar a função em a e b }
x ← b; Fx ← Fb; Iter ← 0
repita
    DeltaX ← -Fx/(Fb - Fa) * (b - a)
    x ← x + DeltaX; Fx ← f(x); { avaliar a função em x }
    escreva Iter, a, Fa, b, Fb, x, Fx, DeltaX
    se (abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler) ou Iter ≥ IterMax então
        interrompa
    fimse
    se Fx * Fb < 0 então
        a ← b; Fa ← Fb
    senão
        Fa ← Fa + Fb/(Fb + Fx)
    fimse
    b ← x; Fb ← Fx
    Iter ← Iter + 1
fimrepita
Raiz ← x
{ teste de convergência }
se abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler então
    CondErro ← 0
senão
    CondErro ← 1
fimse
finalgoritmo
```

Figura 6.15 Algoritmo do método pégaso.

(Ver significado da função abs na Tabela 1.1, na página 6.)

Exemplo 6.24 Calcular com  $\varepsilon \leq 0,01$ , a raiz de  $f(x) = 2x^3 - \cos(x+1) - 3 = 0$  do Exemplo 6.17, pelo método pégaso mostrado na Figura 6.15, sabendo-se que  $\xi \in [-1, 2]$ .

% Os parametros de entrada

a = -1

b = 2

Toler = 0.0100

IterMax = 100

% produzem os resultados

Calculo de raiz de equacao pelo metodo pegaso

Iter	a	Fa	b	Fb	x	Fx	Delta_x
0	-1.00000	-6.00000	2.00000	13.98999	-0.09955	-3.623e+00	-2.100e+00
1	2.00000	13.98999	-0.09955	-3.62323	0.33235	-3.163e+00	4.319e-01
2	2.00000	7.46964	0.33235	-3.16277	0.82842	-1.608e+00	4.961e-01
3	2.00000	4.95180	0.82842	-1.60817	1.11563	2.954e-01	2.872e-01
4	0.82842	-1.60817	1.11563	0.29537	1.07106	-6.294e-02	-4.457e-02
5	1.11563	0.29537	1.07106	-0.06294	1.07889	-1.807e-03	7.828e-03

Raiz = 1.07889

Iter = 5

CondErro = 0

Portanto, a raiz da equação é  $\xi \approx x_5 = 1,07889$ .

Exemplo 6.25 Achar o ponto de máximo  $\mu$  do polinômio  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24$ , pelo método pégaso, com  $\varepsilon \leq 10^{-5}$ , sabendo-se que  $\mu \in [-1, 1]$ , de acordo com a Figura 6.5(b).

A condição de máximo de uma função é que a derivada primeira se anule e que a derivada segunda seja negativa. Assim, o problema é equivalente a calcular uma raiz de  $P'(x) = 0$

$$P'(x) = 4x^3 + 6x^2 - 26x - 14 = 0.$$

% Os parametros de entrada

a = -1

b = 1

Toler = 1.0000e-05

IterMax = 100

% produzem os resultados

Calculo de raiz de equacao pelo metodo pegaso

Iter	a	Fa	b	Fb	x	Fx	Delta_x
0	-1.00000	14.00000	1.00000	-30.00000	-0.36364	-3.944e+00	-1.364e+00
1	-1.00000	12.37317	-0.36364	-3.94440	-0.51746	5.064e-01	-1.538e-01
2	-0.36364	-3.94440	-0.51746	0.50640	-0.49996	-1.135e-03	1.750e-02
3	-0.51746	0.50640	-0.49996	-0.00114	-0.50000	4.764e-08	-3.914e-05
4	-0.49996	-0.00114	-0.50000	0.00000	-0.50000	0.000e+00	1.643e-09

Raiz = -0.50000

Iter = 4

CondErro = 0

Como  $P''(x) = 12x^2 + 12x - 26$ , então  $P''(-0,5) = -29 < 0$ , confirmando que  $\mu \approx x_4 = -0,5$  é um ponto de máximo.

### 6.3.4 Ordem de convergência

Para determinar as ordens de convergência dos métodos baseados em aproximação linear, considere uma estimativa  $x_2$  da raiz  $\xi$  obtida por uma reta passando pelos pontos de coordenadas  $[x_0, f(x_0)]$  e  $[x_1, f(x_1)]$

$$x_2 = x_1 - \frac{f(x_1)}{f(x_1) - f(x_0)}(x_1 - x_0) = \frac{x_1 f(x_0) - x_0 f(x_1)}{f(x_0) - f(x_1)}.$$

Expandindo  $f(x_k)$  em série de Taylor com relação à raiz  $\xi$  e considerando o erro da  $k$ -ésima iteração dado por (6.5), tem-se

$$\epsilon_2 + \xi = \frac{(\epsilon_1 + \xi) \left( \epsilon_0 f'(\xi) + \epsilon_0^2 \frac{f''(\xi)}{2} + \dots \right) - (\epsilon_0 + \xi) \left( \epsilon_1 f'(\xi) + \epsilon_1^2 \frac{f''(\xi)}{2} + \dots \right)}{(\epsilon_0 - \epsilon_1) f'(\xi) + (\epsilon_0^2 - \epsilon_1^2) \frac{f''(\xi)}{2} + \dots}.$$

Simplificando

$$\epsilon_2 = \frac{\frac{f''(\xi)}{2} \epsilon_0 \epsilon_1 (\epsilon_0 - \epsilon_1) + \dots}{f'(\xi) (\epsilon_0 - \epsilon_1) + \frac{f''(\xi)}{2} (\epsilon_0 - \epsilon_1) (\epsilon_0 + \epsilon_1) + \dots}.$$

Dividindo por  $f'(\xi)(\epsilon_0 - \epsilon_1)$

$$\epsilon_2 = \frac{\frac{f''(\xi)}{2f'(\xi)} \epsilon_0 \epsilon_1 + \dots}{1 + \frac{f''(\xi)}{2f'(\xi)} (\epsilon_0 + \epsilon_1) + \dots},$$

$$\epsilon_2 \approx \frac{f''(\xi)}{2f'(\xi)} \epsilon_0 \epsilon_1. \quad (6.10)$$

No caso do método da *regula falsi*, como a raiz deve ficar sempre isolada em um intervalo, então  $x_0$  será geralmente fixo durante várias iterações. Consequentemente, o erro  $\epsilon_0$  também será fixo, resultando que o erro da  $k$ -ésima iteração será da forma

$$\epsilon_{k+1} = K_r \epsilon_k.$$

Assim, por (6.6) verifica-se que o método da *regula falsi* apresenta convergência de primeira ordem, conforme Acton [2]. Para o método da secante, como os valores de  $x_k$  e  $x_{k-1}$  são sempre atualizados, (6.10) pode ser generalizada por

$$\epsilon_{k+1} = C \epsilon_{k-1} \epsilon_k.$$

Por (6.6)

$$|\epsilon_{k+1}| = K |\epsilon_k|^\gamma.$$

Usando esta equação na equação anterior, tem-se

$$K |\epsilon_k|^\gamma = |C| |\epsilon_k| \left( \frac{|\epsilon_k|}{K} \right)^{\frac{1}{\gamma}}.$$

Rearranjando,

$$K^{1+\frac{1}{\gamma}} |\epsilon_k|^\gamma = |C| |\epsilon_k|^{1+\frac{1}{\gamma}}.$$

Como a ordem de convergência  $\gamma$  deve ser positiva e pela equação acima

$$\gamma = 1 + \frac{1}{\gamma} \rightarrow \gamma^2 - \gamma - 1 = 0 \leadsto \gamma = \frac{1 + \sqrt{5}}{2} \approx 1,61803.$$

Consequentemente, o método da secante tem ordem de convergência igual à relação áurea, como mostrado por Hildebrand [22]. Além disso,

$$K^{1+\frac{1}{\gamma}} = |C|.$$

Como  $1 + \frac{1}{\gamma} = \gamma$ , então

$$K^\gamma = |C| \rightarrow K = |C|^{\frac{1}{\gamma}} = |C|^{\gamma-1}.$$

Portanto, por (6.6) e (6.10), o método da secante apresenta

$$|\epsilon_{k+1}| \approx \left| \frac{f''(\xi)}{2f'(\xi)} \right|^{\gamma-1} |\epsilon_k|^\gamma.$$

Segundo Dowell e Jarratt [12], o método pégaso tem ordem de convergência 1,642, mostrando que ele é superior ao método da secante e ao da *regula falsi*.

## 6.4 Métodos baseados em aproximação quadrática

Na Seção 6.3 foram vistos métodos para cálculo de raízes baseados na aproximação da função  $f(x)$  por um polinômio interpolador de grau 1. Naqueles métodos, uma estimativa da raiz é o ponto onde a reta intercepta o eixo das abscissas. A estimativa da raiz de  $f(x) = 0$  pode ser ainda melhor se em vez de um polinômio interpolador de grau 1 for utilizado um polinômio de grau 2.

### 6.4.1 Método de Muller

O método de Muller [33] consiste em aproximar a função  $f(x)$ , na vizinhança da raiz  $\xi \in [x_0, x_2]$ , por um polinômio quadrático. Este polinômio é construído de modo a passar pelos três pontos de coordenadas  $[x_0, f(x_0)]$ ,  $[x_1, f(x_1)]$  e  $[x_2, f(x_2)]$ . Deste modo, o zero do polinômio é usado como uma estimativa da raiz  $\xi$  de  $f(x) = 0$ . O processo é, então, repetido usando sempre os três pontos mais próximos da raiz.

Um polinômio de segundo grau que passa pelos três pontos de coordenadas  $[x_{i-2}, f(x_{i-2})]$ ,  $[x_{i-1}, f(x_{i-1})]$  e  $[x_i, f(x_i)]$  na forma

$$P_2(v) = av^2 + bv + c \quad (6.11)$$



onde  $v = x - x_{i-1}$  pode ser construído de acordo com a Figura 6.16. Para cada um dos três pontos, tem-se que

$$\begin{aligned} P_2(x_{i-2}) &= f(x_{i-2}) \rightarrow a(x_{i-2} - x_{i-1})^2 + b(x_{i-2} - x_{i-1}) + c = f(x_{i-2}), \\ P_2(x_{i-1}) &= f(x_{i-1}) \rightarrow a(0)^2 + b(0) + c = f(x_{i-1}) \rightarrow c = f(x_{i-1}) \text{ e} \\ P_2(x_i) &= f(x_i) \rightarrow a(x_i - x_{i-1})^2 + b(x_i - x_{i-1}) + c = f(x_i). \end{aligned} \quad (6.12)$$

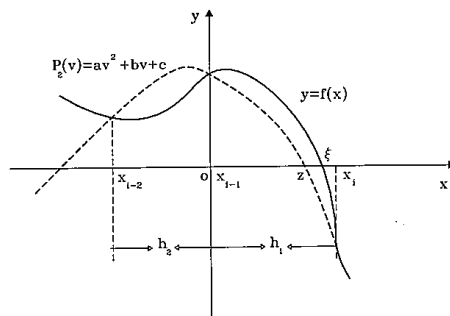


Figura 6.16 Interpretação gráfica do método de Muller.

Definindo

$$h_1 = x_i - x_{i-1} \text{ e}$$

$$h_2 = x_{i-1} - x_{i-2}$$

e, em vista de (6.12), é obtido o seguinte sistema linear em termos das incógnitas  $a$  e  $b$

$$h_2^2 a - h_2 b = f(x_{i-2}) - f(x_{i-1}),$$

$$* \quad h_1^2 a + h_1 b = f(x_i) - f(x_{i-1}),$$

cujas soluções é

$$a = \frac{1}{h_1(h_1 + h_2)} (f(x_i) - (r+1)f(x_{i-1}) + rf(x_{i-2})) \quad (6.13)$$

sendo  $r = h_1/h_2$  e

$$b = \frac{1}{h_1} (f(x_i) - f(x_{i-1})) - ah_1, \quad (6.14)$$

onde  $a$  é dado por (6.13). Os dois zeros do polinômio de grau 2 em  $v$  (6.11) são

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

Para ser obtida a raiz mais próxima de  $x_{i-1}$ , o sinal na expressão acima deve ser escolhido de modo a tornar o numerador o menor possível. Assim, em vista da transformação  $v = x - x_{i-1}$ , a próxima estimativa da raiz  $\xi$  de  $f(x) = 0$  é

$$x_{i+1} = x_{i-1} + \frac{-b + \text{sign}(b)\sqrt{b^2 - 4ac}}{2a},$$

onde  $a$ ,  $b$  e  $c$  são dados por (6.13), (6.14) e (6.12), respectivamente. Na próxima iteração, devem ser utilizados os três pontos mais próximos de  $\xi$ .

O algoritmo mostrado na Figura 6.17 determina, pelo método de Muller, uma raiz da equação  $f(x) = 0$  que está no intervalo  $[a, c]$ , com tolerância  $\varepsilon$  usando os critérios (6.2) e (6.4). Os parâmetros de entrada são o limite inferior  $a$  e o superior  $c$  do intervalo que contém a raiz, a tolerância *Toler* para o cálculo da raiz e o número máximo de iterações *IterMax*. A função  $f(x)$  deve ser especificada de acordo com a linguagem de programação escolhida. Os parâmetros de saída são a raiz de  $f(x) = 0$  *Raiz*, o número de iterações gastas *Iter* e a condição de erro *CondErro*, em que *CondErro* = 0 significa que a raiz foi determinada com sucesso e *CondErro* = 1 mostra que a raiz não foi encontrada com a tolerância e o número máximo de iterações fornecidos.

Exemplo 6.26 Calcular com  $\varepsilon \leq 0,01$ , a raiz de  $f(x) = 2x^3 - \cos(x+1) - 3 = 0$  do Exemplo 6.17, pelo método de Muller apresentado na Figura 6.17, sabendo-se que  $\xi \in [-1, 2]$ .

% Os parametros de entrada

a = -1

b = 2

Toler = 0.0100

IterMax = 100

% produzem os resultados

Calculo de raiz de equacao pelo metodo de Muller

iter	a	b	c	x	Fx	Delta_x
0	-1.00000	0.50000	2.00000	0.86331	-1.42476e+00	3.63315e-01
1	0.50000	0.86331	2.00000	1.05488	-1.86933e-01	1.91564e-01
2	0.86331	1.05488	2.00000	1.07803	-8.58214e-03	2.31508e-02
3	1.05488	1.07803	2.00000	1.07912	-4.55606e-05	1.08694e-03

Raiz = 1.07912

Iter = 3

CondErro = 0

Assim, a raiz da equação é  $\xi \approx x_3 = 1,07912$ . ■

Exemplo 6.27 Achar a raiz de  $f(x) = 0,05x^3 - 0,4x^2 + 3 \sin(x)x = 0$  do Exemplo 6.18, com  $\varepsilon \leq 10^{-10}$ , que se encontra no intervalo  $[10, 12]$ , usando o método de Muller.

```

Algoritmo Muller
{ Objetivo: Calcular a raiz de uma equação pelo método de Muller }
parâmetros de entrada a, c, Toler, IterMax
{ limite inferior, limite superior, tolerância e número máximo de iterações }
parâmetros de saída Raiz, Iter, CondErro
{ raiz, número de iterações gastas e condição de erro, sendo }
{ CondErro = 0 se a raiz foi encontrada e CondErro = 1 em caso contrário. }
{ avaliar a função em a, c e b }
Fa ← f(a); Fc ← f(c); b ← (a + c)/2; Fb ← f(b)
x ← b; Fx ← Fb; DeltaX ← c - a; Iter ← 0
repara
  h1 ← c - b; h2 ← b - a; r ← h1/h2; t ← x
  A ← ((Fc - (r + 1) * Fb + r * Fa) / (h1 * (h1 + h2)))
  B ← ((Fc - Fb) / h1 - A * h2)
  C = Fb; z ← (-B + sinal(B) * raiz2(B^2 - 4 * A * C)) / (2 * A)
  x ← b + z; DeltaX ← x - t; Fx ← f(x); { avaliar a função em x }
  escreva Iter, a, b, c, x, Fx, DeltaX
  se (abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler) ou Iter ≥ IterMax então
    interrompa
fimse
se x > b então
  a ← b; Fa ← Fb
senão
  c ← b; Fc ← Fb
fimse
b ← x; Fb ← Fx; Iter ← Iter + 1
fimrepara
Raiz ← x
{ teste de convergência }
se abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler então
  CondErro ← 0
senão
  CondErro ← 1
fimse
finalgoritmo

```

Figura 6.17 Algoritmo do método de Muller.

(Ver significado das funções abs, raiz2 e sinal na Tabela 1.1, na página 6.)

```

% Os parametros de entrada
a = 10
b = 12
Toler = 1.0000e-10
IterMax = 100

```

```

% produzem os resultados
Calculo de raiz de equacao pelo metodo de Muller
iter   a         b         c         x         Fx         Delta_x
0      10.00000   11.00000   12.00000   11.74014  -1.25090e-01  7.40141e-01
1      11.00000   11.74014   12.00000   11.74398   1.54925e-03  3.83681e-03
2      11.74014   11.74398   12.00000   11.74393  -1.45315e-07  -4.68547e-05
3      11.74014   11.74393   11.74398   11.74393   1.06581e-14   4.39453e-09
4      11.74393   11.74393   11.74398   11.74393   1.06581e-14   0.00000e+00

Raiz    = 11.74393
Iter     = 4
CondErro = 0

```

A raiz da equação é  $\xi \approx x_4 = 11,74393$ .

Hildebrand [22] mostrou que, para o método de Muller, a expressão (6.6) apresenta a forma

$$|\epsilon_{k+1}| \approx \left| \frac{f'''(\xi)}{6f'(\xi)} \right|^{\frac{\gamma-1}{2}} |\epsilon_k|^\gamma$$

onde  $\gamma$  é a raiz positiva da equação

$$\gamma = 1 + \frac{1}{\gamma} + \frac{1}{\gamma^2} \rightarrow \gamma^3 - \gamma^2 - \gamma - 1 = 0,$$

indicando que o método de Muller tem ordem de convergência  $\gamma \approx 1,8393$ .

#### 6.4.2 Método de van Wijngaarden-Dekker-Brent

Este método é o resultado da combinação da interpolação inversa quadrática e da bissecção implementados de forma a garantir que a raiz continue sempre isolada. Ele foi proposto inicialmente por van Wijngaarden, Dekker e outros e melhorado posteriormente por Brent [6]. Na interpolação quadrática, a forma analítica de um polinômio  $P_2(x) \approx f(x) = y$  é determinada a partir de três pontos de coordenadas  $[x_{i-2}, f(x_{i-2})]$ ,  $[x_{i-1}, f(x_{i-1})]$  e  $[x_i, f(x_i)]$ ; assim, para obter um valor aproximado de  $f(t)$ , basta avaliar  $P_2(t)$ . Por outro lado, na interpolação inversa quadrática, o polinômio interpolador de grau 2

$$\Pi_2(y) \approx f^{-1}(y) = x \quad (6.15)$$

é construído a partir dos pontos de coordenadas  $[f(x_{i-2}), x_{i-2}]$ ,  $[f(x_{i-1}), x_{i-1}]$  e  $[f(x_i), x_i]$  (ver Seção 3.13.2). Neste caso, para ter um valor aproximado de  $f^{-1}(z)$  é necessário avaliar  $\Pi_2(z)$ . Este polinômio  $\Pi_2(y)$  pode ser obtido por interpolação de Lagrange

$$\begin{aligned} \Pi_2(y) = & x_{i-2} \frac{(y - f(x_{i-1}))(y - f(x_i))}{(f(x_{i-2}) - f(x_{i-1}))(f(x_{i-2}) - f(x_i))} \\ & + x_{i-1} \frac{(y - f(x_{i-2}))(y - f(x_i))}{(f(x_{i-1}) - f(x_{i-2}))(f(x_{i-1}) - f(x_i))} \\ & + x_i \frac{(y - f(x_{i-2}))(y - f(x_{i-1}))}{(f(x_i) - f(x_{i-2}))(f(x_i) - f(x_{i-1}))}, \end{aligned}$$

conforme mostrado na Seção 3.2. Como  $y = f(x) \rightarrow x = f^{-1}(y)$ , então uma aproximação da raiz  $\xi$  de  $f(x) = 0$  é o ponto de abscissa correspondente à  $f^{-1}(0)$ . Em vista de (6.15), esta aproximação é dada por  $x = \Pi_2(0)$ .

Na Figura 6.18, é mostrado um algoritmo baseado nos programas das referências [32, 34] para calcular, pelo método de van Wijngaarden-Dekker-Brent, uma raiz da equação  $f(x) = 0$  pertencente ao intervalo  $[a, b]$ , com tolerância  $\epsilon$  usando critérios semelhantes a (6.2) e (6.4). Os parâmetros de entrada são o limite inferior  $a$  e o superior  $b$  do intervalo que isola a raiz, a tolerância  $Toler$  para o cálculo da raiz e o número máximo de iterações  $IterMax$ . A função  $f(x)$  deve ser especificada de acordo com a linguagem de programação usada. Os parâmetros de saída são a raiz da equação  $Raiz$ , o número de iterações gastas  $Iter$  e a condição de erro  $CondErro$ , em que  $CondErro = 0$  indica que a raiz foi calculada com sucesso e  $CondErro = 1$  avisa que a raiz não foi encontrada com a tolerância e o número máximo de iterações dados.

Exemplo 6.28 Calcular pelo método de van Wijngaarden-Dekker-Brent, descrito na Figura 6.18, a menor raiz de  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$  do Exemplo 6.11, com  $\epsilon \leq 10^{-10}$ , sabendo-se que  $\xi \in [-5, -3]$  de acordo com a Figura 6.5(b).

```
% Os parametros de entrada
a = -5
b = -3
Toler = 1.0000e-10
IterMax = 100
% produzem os resultados
Calculo de raiz pelo metodo de van Wijngaarden-Dekker-Brent
iter    a          c          b          Fb          z
0      -5.00000    -5.00000    -3.00000    -2.40000e+01    -1.00000e+00
1      -3.00000    -5.00000    -3.28571    -2.47397e+01    -8.57143e-01
2      -3.28571    -3.28571    -4.14286    1.12453e+01     4.28571e-01
3      -4.14286    -4.14286    -3.87500    -7.85522e+00    -1.33929e-01
4      -3.87500    -4.14286    -3.98516    -1.02599e+00    -7.88495e-02
5      -3.98516    -3.98516    -4.00032    2.26777e-02     7.58292e-03
6      -4.00032    -4.00032    -4.00000    -2.86125e-04    -1.63983e-04
7      -4.00000    -4.00032    -4.00000    -7.80927e-08    -1.61940e-04
8      -4.00000    -4.00032    -4.00000    0.00000e+00    -1.61940e-04

Raiz    = -4.00000
Iter     = 8
CondErro = 0
```

ou seja, a menor raiz de  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$  é  $\xi = x_8 = -4$ .

Exemplo 6.29 Calcular a raiz de  $f(x) = 0,05x^3 - 0,4x^2 + 3\text{sen}(x)x = 0$  do Exemplo 6.18, com  $\epsilon \leq 10^{-10}$ , que se encontra no intervalo  $[10, 12]$ , utilizando o método de van Wijngaarden-Dekker-Brent.

```
Algoritmo van Wijngaarden-Dekker-Brent
{ Objetivo: Calcular a raiz pelo método de van Wijngaarden-Dekker-Brent }
parâmetros de entrada a, b, Toler, IterMax
{ limite inferior, limite superior, tolerância e número máximo de iterações }
parâmetros de saída Raiz, Iter, CondErro
{ raiz, número de iterações gastas e condição de erro, sendo }
{ CondErro = 0 se a raiz foi encontrada e CondErro = 1 em caso contrário. }
Fa ← f(a); Fb ← f(b); { avaliar a função em a e b }
se Fa * Fb > 0 então
    escreva "a função não muda de sinal nos extremos do intervalo dado"
    abandone
fimse
c ← b; Fc ← Fb; Iter ← 0
repita
    { altera a, b e c para que b seja a melhor estimativa da raiz }
    se Fb * Fc > 0 então c ← a, Fc ← Fa, d ← b - a, e ← d, fimse
    se abs(Fc) < abs(Fb) então
        a ← b; b ← c; c ← a; Fa ← Fb; Fb ← Fc; Fc ← Fa
    fimse
    Tol ← 2 * Toler * max(abs(b), 1); z ← (c - b) / 2
    escreva Iter, a, c, b, Fb, z
    { teste de convergência }
    se abs(z) ≤ Tol ou Fb = 0 ou Iter ≥ IterMax então interrompa, fimse
    { escolha entre interpolação e bisseção }
    se abs(c) ≥ Tol e abs(Fa) > abs(Fb) então
        s ← Fb / Fa
        se s = c então { interpolação linear }
            p ← 2 * z * s; q ← 1 - s
        senão { interpolação inversa quadrática }
            q ← Fa / Fc; r ← Fb / Fc; p ← s * (2 * z * q * (q - r) - ((b - a) * ((r - 1)))
            q ← ((q - 1) * (r - 1) * (s - 1))
        fimse
        se p > 0 então q ← -q, senão p ← -p, fimse
        se 2 * p < min(3 * z * q - abs(Tol * q), abs(c * q)) então { aceita interpolação }
            e ← d; d ← p / q
        senão { usa bisseção devido à falha na interpolação }
            d ← z; e ← z
        fimse
    senão { bisseção }
        d ← z; e ← z
    fimse
    a ← b; Fa ← Fb
    se abs(d) ≥ Tol então b ← b + d, senão b ← b + sinal(z) * Tol, fimse
    Iter ← Iter + 1; Fb ← f(b) { avaliar a função em b }
fimrepita
Raiz ← b
se abs(z) ≤ Tol ou Fb = 0 então CondErro ← 0, senão CondErro ← 1, fimse
finalgoritmo
```

Figura 6.18 Algoritmo do método de van Wijngaarden-Dekker-Brent.  
(Ver significado das funções abs, max, min e sinal na Tabela 1.1, na página 6.)

% Os parametros de entrada

a = 10

b = 12

Toler = 1.0000e-10

IterMax = 100

% produzem os resultados

Calculo de raiz pelo metodo de van Wijngaarden-Dekker-Brent

iter	a	c	b	Fb	z
0	12.00000	12.00000	10.00000	-6.32063e+00	1.00000e+00
1	10.79988	10.79988	12.00000	9.48337e+00	-6.00061e-01
2	12.00000	12.00000	11.54358	-5.94963e+00	2.28208e-01
3	11.54358	12.00000	11.71954	-7.96853e-01	1.40231e-01
4	11.71954	11.71954	11.74464	2.34449e-02	-1.25507e-02
5	11.74464	11.74464	11.74392	-2.86520e-04	3.58711e-04
6	11.74392	11.74464	11.74393	-1.00128e-07	3.54380e-04
7	11.74393	11.74393	11.74393	1.06581e-14	-1.51400e-09

Raiz = 11.74393

Iter = 7

CondErro = 0

A raiz procurada é  $\xi \approx x_7 = 11,74393$ .

Segundo Brent [6], a convergência pelo método é garantida desde que haja uma raiz no intervalo. A combinação da certeza de convergência do método da bisseção com a rapidez de um método de ordem de convergência maior como o da interpolação inversa quadrática resulta em um esquema robusto e eficiente.

O método de van Wijngaarden-Dekker-Brent é recomendado como sendo o mais adequado para o cálculo do zero de uma função de uma variável quando a sua derivada não estiver disponível [32, 34].

## 6.5 Métodos baseados em tangente

Com exceção do método da bisseção, os demais métodos descritos nas seções anteriores são baseados na aproximação de um arco da curva de  $f(x)$  por polinômios lineares e quadráticos. A seguir, serão apresentados dois métodos baseados no cálculo da tangente à curva de  $f(x)$ .

### 6.5.1 Método de Newton

Sejam  $\xi$  a única raiz de  $f(x) = 0$  no intervalo  $[a, b]$  e  $x_k$  uma aproximação desta raiz, sendo  $x_0 \in [a, b]$ . Além disso, as derivadas  $f'(x)$  e  $f''(x)$  devem existir, ser contínuas e com sinal constante neste intervalo. Geometricamente, o método de Newton é equivalente a aproximar um arco da curva por uma reta tangente traçada a partir de um ponto da curva, o que faz com que ele seja conhecido também como o método das tangentes. Considere a Figura 6.19, na qual

$$\tan(\alpha) = \frac{f(x_0)}{x_0 - x_1} = f'(x_0) \rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad e$$

$$\tan(\beta) = \frac{f(x_1)}{x_1 - x_2} = f'(x_1) \rightarrow x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

A generalização das expressões acima fornece a fórmula de recorrência do método de Newton

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \quad (6.16)$$

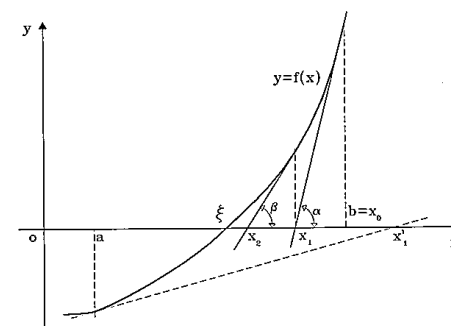


Figura 6.19 Interpretação gráfica do método de Newton.

A fórmula do método de Newton pode também ser deduzida analiticamente. Para isso seja

$$\xi = x_k + \delta_k \quad (6.17)$$

tal que  $\delta_k$  tenha um valor pequeno. Fazendo uma expansão em série de Taylor

$$f(\xi) = f(x_k + \delta_k) \approx f(x_k) + f'(x_k)\delta_k = 0 \rightarrow \delta_k = -\frac{f(x_k)}{f'(x_k)}.$$

Substituindo essa correção em (6.17), obtém-se (6.16). Pela Figura 6.19, a sequência produzida por (6.16) convergirá para a raiz  $\xi$  se o valor inicial for  $x_0 = b$ . No entanto, para aquela figura, o processo pode não convergir se  $x_0 = a$ , pois ter-se-á  $x'_1 \notin [a, b]$ . A questão da escolha do valor inicial de modo a garantir a convergência para a raiz é resolvida pelo teorema, a seguir, apresentado por Demidovich e Maron [10].

**Teorema 6.6** Se  $f(a)f(b) < 0$ , e  $f'(x)$  e  $f''(x)$  forem não nulas e preservarem o sinal em  $[a, b]$ , então partindo-se da aproximação inicial  $x_0 \in [a, b]$  tal que  $f(x_0)f''(x_0) > 0$  é possível construir, pelo método de Newton (6.16), uma sequência  $\{x_i\}$  que convirja para a raiz  $\xi$  de  $f(x) = 0$ .

Por este teorema, o valor inicial  $x_0$  deve ser um ponto no qual a função tenha o mesmo sinal de sua derivada segunda, isto é, se  $f''(x_0) > 0$ , então  $x_0$  é tal que  $f(x_0) > 0$ , e por outro lado, se  $f''(x_0) < 0$ , então  $f(x_0) < 0$ .

Um algoritmo do método de Newton é mostrado na Figura 6.20, no qual a raiz de uma equação  $f(x) = 0$  é calculada com tolerância  $\varepsilon$  usando os critérios (6.2) e (6.4). Os parâmetros de entrada são o valor inicial  $x_0$ , a tolerância *Toler* para o cálculo da raiz e o número máximo de iterações *IterMax*. A função  $f(x)$  e sua derivada  $f'(x)$  devem ser especificadas de acordo com a linguagem de programação utilizada. Os parâmetros de saída são a raiz de  $f(x) = 0$  *Raiz*, o número de iterações gastas *Iter* e a condição de erro *CondErro*, em que *CondErro* = 0 mostra que a raiz foi calculada com sucesso e *CondErro* = 1 avisa que a raiz não foi encontrada com a tolerância e o número máximo de iterações fornecidos.

#### Algoritmo Newton

```
{ Objetivo: Calcular a raiz de uma equação pelo método de Newton }
parâmetros de entrada x0, Toler, IterMax
{ valor inicial, tolerância e número máximo de iterações }
parâmetros de saída Raiz, Iter, CondErro
{ raiz, número de iterações gastas e condição de erro, sendo }
{ CondErro = 0 se a raiz foi encontrada e CondErro = 1 em caso contrário. }
{ avaliar a função e sua derivada em x0 }
Fx ← f(x0); DFx ← f'(x0); x ← x0; Iter ← 0
escreva Iter, x, DFx, Fx
repita
    DeltaX ← -Fx/DFx; x ← x + DeltaX
    Fx ← f(x); DFx ← f'(x); { avaliar a função e sua derivada em x }
    Iter ← Iter + 1
    escreva Iter, x, DFx, Fx, DeltaX
    se (abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler) ou DFx = 0 ou Iter ≥ IterMax
        então interrompa
fimse
fimrepita
Raiz ← x
{ teste de convergência }
se abs(DeltaX) ≤ Toler e abs(Fx) ≤ Toler então
    CondErro ← 0
senão
    CondErro ← 1
fimse
finalgoritmo
```

Figura 6.20 Algoritmo do método de Newton.

(Ver significado da função abs na Tabela 1.1, na página 6.)

Exemplo 6.30 Determinar a maior raiz de  $P(x) = x^4 + 2x^3 - 13x^2 - 14x + 24 = 0$  com  $\varepsilon \leq 10^{-5}$ , utilizando o método de Newton.

Pelo Exemplo 6.11 e Figura 6.5(b), sabe-se que  $\xi \in [2, 4]$ ,  $f(2) < 0$  e  $f(4) > 0$ . As derivadas são  $P'(x) = 4x^3 + 6x^2 - 26x - 14$  e  $P''(x) = 12x^2 + 12x - 26 > 0$ ,  $2 \leq x \leq 4$ . Assim,  $x_0 = 4$ , pois  $P(4)P''(4) > 0$ .

```
% Os parametros de entrada
x0 = 4
Toler = 1.0000e-05
IterMax = 100
% produzem os resultados
Calculo de raiz de equacao pelo metodo de Newton
iter    x          DFx          Fx          Delta_x
0       4.00000    2.34000e+02    1.44000e+02
1       3.38462    1.21825e+02    3.64693e+01    -6.15385e-01
2       3.08526    8.03682e+01    6.40563e+00    -2.99358e-01
3       3.00555    7.06567e+01    3.90611e-01    -7.97036e-02
4       3.00003    7.00030e+01    1.80793e-03    -5.52830e-03
5       3.00000    7.00000e+01    3.93537e-08    -2.58264e-05
6       3.00000    7.00000e+01    1.42109e-14    -5.62196e-10
```

```
Raiz = 3.00000
Iter = 6
CondErro = 0
```

A raiz da equação é  $\xi = x_6 = 3$ .

Exemplo 6.31 Calcular o ponto de inflexão  $\iota$  da função  $f(x) = 2x^3 + 3x^2 - e^x + 3$  com  $\varepsilon \leq 10^{-5}$  pelo método de Newton, usando o algoritmo mostrado na Figura 6.20.

A condição de ponto de inflexão de uma função é que a derivada segunda se anule. Deste modo, deve-se achar uma raiz de  $f''(x) = 0$

$$g(x) = f''(x) = 12x - e^x + 6 = 0.$$

Pela Figura 6.21(a) verifica-se que  $\iota \in [-2, 1]$ . As derivadas são  $g'(x) = 12 - e^x$  e  $g''(x) = -e^x < 0 \forall x$ . Como  $g(-2) \approx -1,1353 < 0$  e  $g(1) \approx 5,2817 > 0$ , então  $x_0 = -2$  porque  $g(-2)g''(-2) > 0$ .

```
% Os parametros de entrada
x0 = -2
Toler = 1.0000e-05
IterMax = 100
% produzem os resultados
Calculo de raiz de equacao pelo metodo de Newton
iter    x          DFx          Fx          Delta_x
0       -2.00000    1.18647e+01    -1.81353e+01
1       -0.47148    1.13759e+01    -2.81878e-01    1.52852e+00
2       -0.44671    1.13603e+01    -1.93175e-04    2.47785e-02
3       -0.44669    1.13603e+01    -9.24905e-11    1.70045e-05
4       -0.44669    1.13603e+01    0.00000e+00    8.14158e-12
```

```
Raiz = -0.44669
Iter = 4
CondErro = 0
```

© ponto de inflexão é  $\iota \approx x_4 = -0,44669$ .

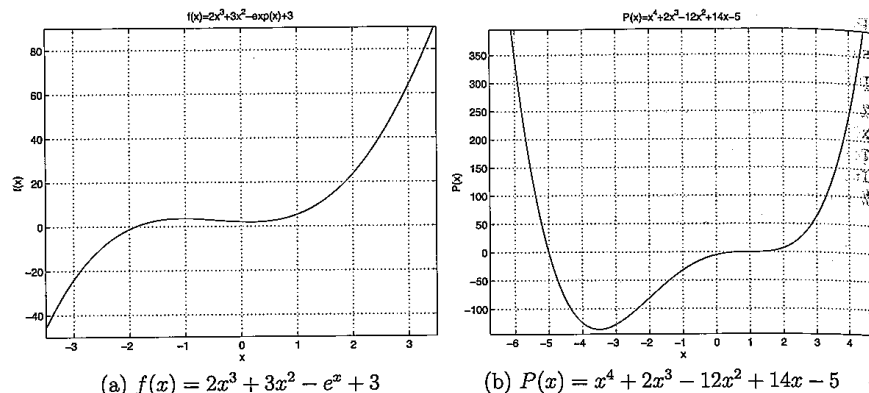


Figura 6.21 Esboços para isolamento de raízes.

Para determinar a ordem de convergência do método de Newton, considere (6.16)

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

que em vista do erro da  $k$ -ésima iteração (6.5) torna-se

$$\epsilon_{k+1} = \epsilon_k - \frac{f(x_k)}{f'(x_k)}.$$

Expandindo  $f(x_k)$  em série de Taylor em torno da raiz  $\xi$ , tem-se

$$f(x_k) = f(\xi) + \epsilon_k f'(\xi) + \frac{\epsilon_k^2 f''(\xi)}{2} + \dots,$$

$$f'(x_k) = f'(\xi) + \epsilon_k f''(\xi) + \dots$$

Substituindo as duas expressões acima em (6.18)

$$\epsilon_{k+1} = \epsilon_k - \frac{\epsilon_k f'(\xi) + \frac{\epsilon_k^2 f''(\xi)}{2} + \dots}{f'(\xi) + \epsilon_k f''(\xi) + \dots}$$

$$\epsilon_{k+1} = \frac{\epsilon_k f'(\xi) + \frac{\epsilon_k^2 f''(\xi)}{2} + \dots - \epsilon_k f'(\xi) - \frac{\epsilon_k^2 f''(\xi)}{2} - \dots}{f'(\xi) + \epsilon_k f''(\xi) + \dots}$$

$$|\epsilon_{k+1}| \approx \frac{f''(\xi)}{2f'(\xi)} |\epsilon_k|^2.$$

Conseqüentemente, em vista de (6.6), o método de Newton tem convergência quadrática. Isso significa que, nas proximidades da raiz, o número de dígitos corretos da estimativa da raiz praticamente dobra a cada iteração.

### 6.5.2 Método de Schröder

O método de Newton apresenta uma convergência apenas linear quando uma raiz tem multiplicidade  $m > 1$ , isto porque a medida que  $f(x_k) \rightarrow 0$ , o denominador  $f'(x_k) \rightarrow 0$  (ver Exemplo 6.4). Uma modificação simples proposta por Schröder [35] permite o cálculo de uma raiz de multiplicidade  $m$ , mantendo a convergência quadrática, utilizando a fórmula de recorrência

$$x_{k+1} = x_k - m \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots \quad (6.19)$$

O algoritmo do método de Schröder é basicamente igual ao de Newton mostrado na Figura 6.20, mas com um parâmetro extra  $m$  para definir a multiplicidade, o qual é usado no comando

$$\text{DeltaX} \leftarrow -m * Fx / DFx$$

substituto de

$$\text{DeltaX} \leftarrow -Fx / DFx.$$

Exemplo 6.32 Calcular a raiz de  $P(x) = x^4 + 2x^3 - 12x^2 + 14x - 5 = 0$  de multiplicidade  $m = 3$ , com tolerância  $\varepsilon \leq 10^{-5}$  (ver Exemplo 6.4), pelo método de Schröder usando um algoritmo adaptado da Figura 6.20.

Pela Figura 6.21(b), verifica-se que  $\xi \in [0,5 \ 1,5]$ . As derivadas são  $P'(x) = 4x^3 + 6x^2 - 24x + 14$  e  $P''(x) = 12x^2 + 12x - 24 > 0 \ \forall x > 1$ . Conseqüentemente, para satisfazer ao Teorema 6.6,  $x_0 = 1,5$  porque  $P(1,5)P''(1,5) > 0$ .

% Os parametros de entrada

m = 3

x0 = 1.5

Toler = 1.0000e-05

IterMax = 100

% produzem os resultados

Calculo de raiz de equacao pelo metodo de Schroder

iter	x	DFx	Fx	Delta_x
0	1.50000	5.00000e+00	8.12500e-01	
1	1.01250	2.82031e-03	1.17432e-05	-4.87500e-01
2	1.00001	1.34883e-09	4.44089e-15	-1.24913e-02
3	1.00000	2.68212e-11	0.00000e+00	-9.87718e-06

Raiz = 1.00000

Iter = 3

CondErro = 0

Portanto,  $\xi = x_3 = 1$ . O método de Newton gasta 26 iterações para calcular esta raiz com a mesma tolerância  $\varepsilon \leq 10^{-5}$ .

## 6.6 Comparação dos métodos para cálculo de raízes

Um estudo comparativo do desempenho de métodos utilizando uma série de equações está longe de ser perfeito, justamente pela dependência do resultado na escolha dessas equações. Por isso, a determinação da ordem de convergência é mais adequada, pois não é baseada em nenhum empirismo.

Mesmo assim, é interessante verificar o desempenho dos métodos estudados neste capítulo. Para tal foram escolhidas cinco equações, das quais duas com raiz de multiplicidade  $m > 1$ , e o intervalo que isola a raiz desejada

$$f_1(x) = 2x^4 + 4x^3 + 3x^2 - 10x - 15 = 0, \quad \xi \in [0, 3].$$

$$f_2(x) = x^5 - 2x^4 - 9x^3 + 22x^2 + 4x - 24 = 0, \quad \xi \in [0, 5], \quad \text{com } m = 3.$$

$$f_3(x) = 5x^3 + x^2 - e^{1-2x} + \cos(x) + 20 = 0, \quad \xi \in [-5, 5].$$

$$f_4(x) = \sin(x)x + 4 = 0, \quad \xi \in [1, 5].$$

$$f_5(x) = (x - 3)^5 \log_e(x) = 0, \quad \xi \in [2, 5], \quad \text{com } m = 5.$$

Para todos os métodos foram utilizados o mesmo número máximo de iterações (500), tolerância ( $\varepsilon = 10^{-10}$ ) e critério de parada, com exceção do método de van Wijngaarden-Dekker-Brent que usa um critério ligeiramente diferente. Para o método de Newton,  $x_0$  foi escolhido como o ponto médio do intervalo dado, sem considerar o Teorema 6.6.

Os resultados estão apresentados nas Tabelas 6.1–6.5, nas quais Iter é o número de iterações gastas, Erro é a condição de erro (se *CondErro* = 1 ou Raiz está fora do intervalo dado, então Erro é sim) e  $t_{rel}$  é o tempo relativo ao tempo gasto pelo método da bisseção.

**Tabela 6.1** Comparação de métodos usando  $f_1(x) = 2x^4 + 4x^3 + 3x^2 - 10x - 15$ .

Método	Raiz	Iter	Erro	$t_{rel}$
bisseção	1,49288	37		1,00
secante	-1,30038	8	sim	0,28
regula falsi	1,49288	77		2,06
pégaso	1,49288	10		0,35
Muller	1,49288	4		0,25
W-D-Brent	1,49288	9		0,63
Newton	1,49288	4		0,20

O método da bisseção mostrou sua robustez, pois não falhou apesar de não ser o mais eficiente. A secante, embora seja rápida, encontrou uma raiz fora do intervalo dado (Tabela 6.1). A *regula falsi* apresentou uma convergência muito lenta e falhou três vezes. O pégaso, além de

**Tabela 6.2** Comparação de métodos usando  $f_2(x) = x^5 - 2x^4 - 9x^3 + 22x^2 + 4x - 24$ .

Método	Raiz	Iter	Erro	$t_{rel}$
bisseção	1,99999	35		1,00
secante	2,00000	47		1,36
regula falsi	1,82374	500	sim	13,42
pégaso	1,99999	60		1,76
Muller	2,00001	500	sim	17,85
W-D-Brent	2,00001	57		3,59
Newton	2,00001	37		1,55
Schröder	2,00000	4	sim	0,23

**Tabela 6.3** Comparação de métodos usando  $f_3(x) = 5x^3 + x^2 - e^{1-2x} + \cos(x) + 20$ .

Método	Raiz	Iter	Erro	$t_{rel}$
bisseção	-0,92956	41		1,00
secante	-0,92956	21		0,56
regula falsi	0,69661	500	sim	12,33
pégaso	-0,92956	19		0,57
Muller	-0,92956	32		1,16
W-D-Brent	-0,92956	8		0,51
Newton	-0,92956	11		0,48

**Tabela 6.4** Comparação de métodos usando  $f_4(x) = \sin(x)x + 4$ .

Método	Raiz	Iter	Erro	$t_{rel}$
bisseção	4,32324	36		1,00
secante	4,32324	7		0,27
regula falsi	4,32324	9		0,32
pégaso	4,32324	7		0,28
Muller	4,32324	6		0,35
W-D-Brent	4,32324	7		0,57
Newton	4,32324	6		0,30

ser robusto, foi competitivo com relação ao sofisticado método de van Wijngaarden-Dekker-Brent.

O método de Muller não foi robusto, embora eficiente, pois falhou nos casos onde a raiz possui multiplicidade (Tabelas 6.2 e 6.5). O método de van Wijngaarden-Dekker-Brent foi robusto, mas também foi menos eficiente na presença de multiplicidade.

O método de Schröder é uma efetiva modificação do método de Newton para evitar problemas com raízes de multiplicidade, conforme mostram os resultados das Tabelas 6.2 e 6.5, pelas quais ele foi, sem dúvida, o mais eficiente.

Tabela 6.5 Comparação de métodos usando  $f_5(x) = (x-3)^5 \log_e(x)$ .

Método	Raiz	Iter	Erro	$t_{rel}$
bisseção	3,00000	34		1,00
secante	3,00000	137		3,90
regula falsi	2,67570	500	sim	13,89
pégaso	3,00000	187		5,47
Muller	3,01289	500	sim	18,82
W-D-Brent	3,00000	80		5,45
Newton	3,00000	95		4,45
Schröder	3,00000	4		0,26

## 6.7 Exemplos de aplicação

Será mostrado como o cálculo de raiz de equação pode ser aplicado para determinar a taxa de juros de um financiamento, estudo de grande valia para o consumidor, e achar o comprimento de um cabo suspenso que é um problema de interesse para as Engenharias Civil e Elétrica.

### 6.7.1 Juros de financiamento

#### Definição do problema

O preço à vista de uma mercadoria é R\$ 1.100,00. No entanto, ela pode ser financiada por dois planos

Plano 1: entrada de R\$ 100,00 e mais 6 prestações de R\$ 224,58.

Plano 2: sem entrada e 10 prestações de R\$ 163,19.

Qual dos dois planos de financiamento é melhor para o consumidor?

#### Modelagem matemática

O melhor plano será aquele que tiver a menor taxa de juros. Da Matemática Financeira tem-se que

$$\frac{1 - (1+j)^{-p}}{j} = \frac{v - e}{m},$$

sendo  $j$  a taxa de juros,  $p$  o prazo,  $v$  o preço à vista,  $e$  a entrada e  $m$  a mensalidade. Rearranjando a equação acima

$$f(j) = \frac{1 - (1+j)^{-p}}{j} - \frac{v - e}{m} = 0.$$

Assim, o problema de determinar a taxa de juros do financiamento recai em calcular a raiz de uma equação transcendente.

#### Solução numérica

Inicialmente, cada raiz é isolada para depois ser refinada por um dos métodos vistos nas seções anteriores. As equações para cada plano são

$$f_1(j) = \frac{1 - (1+j)^{-6}}{j} - \frac{1.100 - 100}{224,58} = 0 \text{ e}$$

$$f_2(j) = \frac{1 - (1+j)^{-10}}{j} - \frac{1.100 - 0}{163,19} = 0.$$

Os esboços das duas funções são apresentados na Figura 6.22, indicando que cada raiz está isolada no intervalo  $[0,05; 0,1]$ .

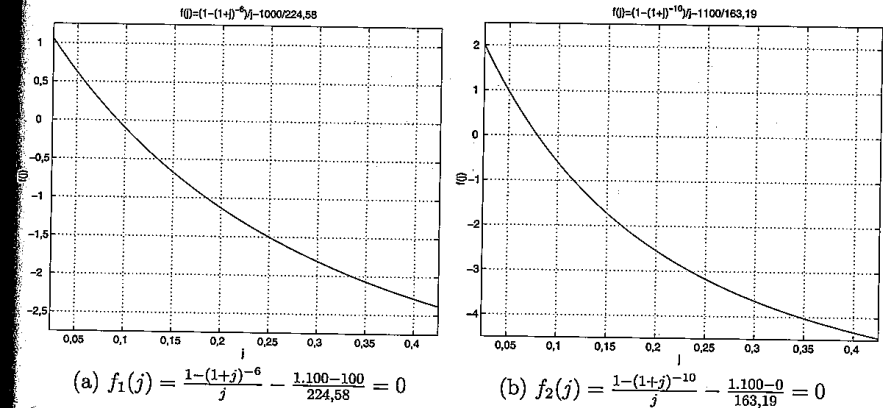


Figura 6.22 Isolamento de raízes para cálculo de juros.

Cada raiz pode agora ser refinada usando, por exemplo, o método pégaso implementado a partir do algoritmo da Figura 6.15, com  $\epsilon \leq 10^{-5}$ .

$$\text{Para o plano 1: } f_1(j) = \frac{1 - (1+j)^{-6}}{j} - \frac{1.100 - 100}{224,58} = 0$$

% Os parametros de entrada

a = 0.0500

b = 0.1000

Toler = 1.0000e-05

IterMax = 100



% produzem os resultados

Calculo de raiz de equacao pelo metodo pegaso						
iter	a	Fa	b	Fb	x	Fx
0	0.05000	0.62294	0.10000	-0.09750	0.09323	-9.758e-03
1	0.05000	0.56626	0.09323	-0.00976	0.09250	-9.373e-05
2	0.05000	0.56088	0.09250	-0.00009	0.09249	1.384e-07

Raiz = 0.09249  
Iter = 2  
CondErro = 0

A taxa de juros do financiamento pelo plano 1 é  $j_1 = 9,25\%$ .

$$\text{Para o plano 2: } f_2(j) = \frac{1 - (1+j)^{-10}}{j} - \frac{1.100 - 0}{163,19} = 0$$

% Os parametros de entrada

a = 0.0500  
b = 0.1000  
Toler = 1.0000e-05  
IterMax = 100

% produzem os resultados

Calculo de raiz de equacao pelo metodo pegaso						
iter	a	Fa	b	Fb	x	Fx
0	0.05000	0.98113	0.10000	-0.59604	0.08110	-6.381e-02
1	0.05000	0.88624	0.08110	-0.06381	0.07901	-6.079e-04
2	0.05000	0.87788	0.07901	-0.00061	0.07899	3.996e-06
3	0.07901	-0.00061	0.07899	0.00000	0.07899	-2.778e-10

Raiz = 0.07899  
Iter = 3  
CondErro = 0

Para este plano, a taxa de juros é  $j_2 = 7,90\%$ .

Análise dos resultados

O total pago por cada plano é

Plano 1: R\$ 100,00 + 6 × R\$ 224,58 = R\$ 1.447,48.

Plano 2: 10 × R\$ 163,19 = R\$ 1.631,90.

Em princípio, o plano 1 parece melhor, pois é desembolsada uma quantia menor. No entanto, isto é um engano porque a taxa de juros do plano 1 é maior. Assim, o plano 2 é melhor por possuir uma menor taxa de juros.

Sempre que possível, o consumidor deve optar pelo pagamento à vista, a menos que ele consiga um investimento que remunere o seu capital acima das taxas de juros praticadas pelas lojas.

## 6.7.2 Cabo suspenso

### Definição do problema

Determinar o comprimento de um cabo suspenso entre dois pontos distantes 50 m que estão no mesmo nível e fazendo uma flecha de 2 m, conforme Figura 6.23(a).

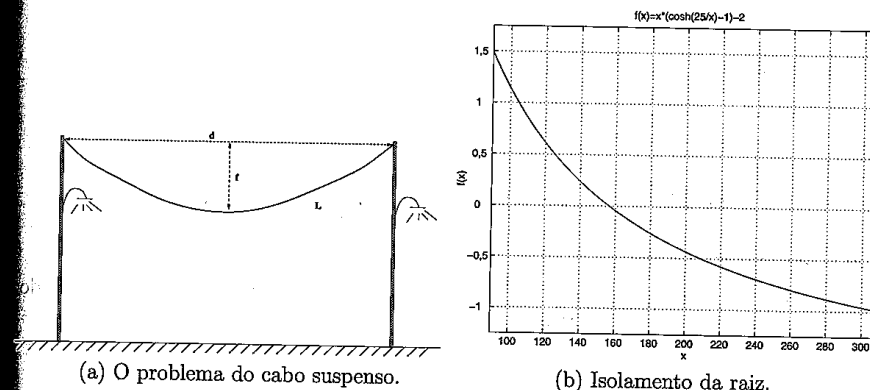


Figura 6.23 Comprimento de um cabo suspenso.

### Modelagem matemática

O comprimento  $L$  é dado pela equação

$$L = 2\alpha \sinh\left(\frac{d}{2\alpha}\right),$$

sendo  $d$  a distância entre os dois pontos e  $\alpha$  a raiz da equação

$$g(x) = x \left( \cosh\left(\frac{d}{2x}\right) - 1 \right) - f = 0,$$

onde  $f$  é a flecha.

### Solução numérica

Pela Figura 6.23(b), nota-se que a raiz  $\alpha$  está isolada no intervalo  $[140, 160]$ . Utilizando o método pegaso, implementado a partir do algoritmo da Figura 6.15, obtêm-se

% Os parametros de entrada  
a = 140  
b = 160  
Toler = 1.0000e-05  
IterMax = 100

% produzem os resultados

Calculo de raiz de equacao pelo metodo pegaso							
iter	a	Fa	b	Fb	x	Fx	Delta_x
0	140.00000	0.23808	160.00000	-0.04290	156.94652	-4.662e-03	-3.053e+00
1	140.00000	0.21474	156.94652	-0.00466	156.58642	-5.405e-05	-3.601e-01
2	140.00000	0.21228	156.58642	-0.00005	156.58219	1.055e-07	-4.222e-03
3	156.58642	-0.00005	156.58219	0.00000	156.58220	-2.828e-12	8.224e-06

Raiz = 156.58220  
Iter = 3  
CondErro = 0

Sendo  $\alpha = 156,58220$ , então o comprimento do cabo é

$$L = 2\alpha \sinh\left(\frac{d}{2\alpha}\right) = 2 \times 156,58220 \times \sinh\left(\frac{50}{2 \times 156,58220}\right) \leadsto L = 50,21270.$$

#### Análise dos resultados

O comprimento do cabo suspenso entre dois pontos no mesmo nível distantes 50 m e fazendo uma flecha de 2 m é igual a 50,21 m.

## 6.8 Exercícios

### Seção 6.1

6.1. Calcular os limites e o número de raízes reais de  $P(x) = 2x^3 - 3x^2 - 6x + 5 = 0$ .

6.2. Implementar, em qualquer linguagem de programação, o algoritmo para determinar os limites das raízes reais de uma equação algébrica mostrado na Figura 6.4.

6.3. Calcular os limites das raízes reais da equação polinomial do Exercício 6.1 utilizando o programa do Exercício 6.2.

6.4. Implementar o algoritmo para encontrar um intervalo onde uma função troca de sinal, descrito na Figura 6.6, utilizando qualquer linguagem de programação.

6.5. Isolar as duas raízes da equação transcendente  $f(x) = e^{-x} + x^2 - 10 = 0$  usando o programa do Exercício 6.4.

### Seção 6.2

Usando o método da bissecção com  $\varepsilon \leq 0,001$ , calcular pelo menos uma raiz de cada equação abaixo.

6.6.  $f(x) = e^{2x} - 2x^3 - 5 = 0$ .

6.7.  $f(x) = 2x^3 - 5x^2 - x + 3 = 0$ .

6.8.  $f(x) = 5x^2 + \log_{10}(x+1) - 2 = 0$ .

6.9. Implementar o algoritmo da Figura 6.10 em qualquer linguagem de programação.

6.10. Resolver os Exercícios 6.6–6.8 usando o programa do Exercício 6.9.

### Seção 6.3

Achar pelo menos uma raiz de cada equação abaixo com  $\varepsilon \leq 10^{-4}$  usando os três métodos: secante, *regula falsi* e pégaso. Fazer a compa-

ração do número de iterações gasto por cada método.

6.11.  $f(x) = 2x^3 - 5x^2 - 10x + 20 = 0$ .

6.12.  $f(x) = 5\log_{10}(x) + 3x^4 - 7 = 0$ .

6.13.  $f(x) = 2^x + \cos(x)x^2 = 0$ .

6.14. Implementar, em qualquer linguagem de programação, os algoritmos descritos nas Figuras 6.12, 6.13 e 6.15.

6.15. Resolver os Exercícios 6.11–6.13 utilizando os programas escritos no Exercício 6.14.

### Seção 6.4

Achar pelo menos uma raiz de cada equação abaixo com  $\varepsilon \leq 10^{-5}$  usando os métodos de Muller e de van Wijngaarden-Dekker-Brent e comparar o número de iterações gasto por cada método.

6.16.  $f(x) = e^{-\cos(x)} + 2x^4 - x^2 - 5 = 0$ .

6.17.  $f(x) = 3x^x - \log_{10}(2 + \sqrt{x}) + 3 \tan(x) - 4 = 0$ .

6.18.  $f(x) = 3x^4 + 2x^3 + 4x^2 + 25x - 30 = 0$ .

6.19. Implementar, usando qualquer linguagem de programação, os algoritmos das Figuras 6.17 e 6.18.

6.20. Resolver os Exercícios 6.16–6.18 utilizando os programas do Exercício 6.19.

### Seção 6.5

Achar pelo menos uma raiz positiva de cada equação abaixo com  $\varepsilon \leq 10^{-5}$  pelos métodos de Newton e de Schröder e comparar o número de iterações gastas por cada método.

6.21.  $f(x) = 4x^3 + x + \cos(x) - 10 = 0$ .