Resoluções de Exercícios de Cálculo Numérico – Capítulo 2

Prof. Ana Isabel C. – Numerical Methods

Junho 2025

Introdução

Este documento apresenta a resolução passo a passo dos exercícios do Capítulo 2 de Cálculo Numérico, focado em encontrar zeros de funções usando métodos gráficos, da Bissecção e Iterativo Linear. Cada exercício inclui teoria, cálculos detalhados, códigos MATLAB e comentários para facilitar a compreensão.

1 Exercício 2.1: Localização Gráfica de Raízes

Localize graficamente as raízes das seguintes funções e dê intervalos de amplitude 0.5 que as contenham:

- a) $\ln(x) + 2x = 0$
- b) $e^x - \sin(x) = 0$
- c) $\ln(x) - 2x = -2$
- d) $2\cos(x) - e^{x/2} = 0$
- e) $3\ln(x) - x^2 = 0$
- f) $(5 - x)e^x = 1$

Passo 1: Teoria: Para localizar raízes graficamente, analisamos onde $f(x) = 0$. Verificamos pontos críticos ($f'(x) = 0$) no intervalo para garantir que a função é monótona (sem máximos/mínimos locais) e usamos o Teorema do Valor Intermediário: se $f(a)f(b) < 0$, existe uma raiz em $[a, b]$. A amplitude do intervalo deve ser 0.5 ($b - a = 0.5$).

Passo 2: Resolução:

- **a)** $f(x) = \ln(x) + 2x$**:** Derivada: $f'(x) = \frac{1}{x} + 2 > 0$ (sempre crescente, sem pontos críticos para $x > 0$). Avaliamos: $f(0.5) = \ln(0.5)+2\cdot 0.5 \approx -0.693+1 \approx 0.307 > 0$, $f(1) = \ln(1)+2\cdot 1 = 2 > 0$, $f(0.1) = \ln(0.1)+2\cdot 0.1 \approx -2.303+0.2 \approx -2.103 < 0$. Como $f(0.1)f(0.5) < 0$, há uma raiz em $[0.1, 0.5]$ (amplitude 0.4, ajustamos para $[0, 0.5]$).

- **b)** $f(x) = e^x - \sin(x)$**:** Derivada: $f'(x) = e^x - \cos(x)$. Resolvemos $f'(x) = 0$: $e^x = \cos(x)$. Testamos em $[0, 2\pi]$: e^x cresce rápido, $\cos(x) \in [-1, 1]$, então pontos críticos são raros. Avaliamos: $f(0) = e^0 - \sin(0) = 1 > 0$, $f(1) = e^1 - \sin(1) \approx 2.718 - 0.841 \approx 1.877 > 0$, $f(-1) = e^{-1} - \sin(-1) \approx 0.368 + 0.841 \approx 1.209 > 0$, $f(-2) = e^{-2} - \sin(-2) \approx 0.135 + 0.909 \approx 1.044 > 0$. Testamos em $x \approx -4$: $f(-4) \approx e^{-4} - \sin(-4) \approx 0.018 + 0.757 \approx 0.775 > 0$, $f(-5) \approx e^{-5} - \sin(-5) \approx 0.007 - (-0.959) \approx -0.952 < 0$. Raiz em $[-5, -4.5]$.
- **c)** $f(x) = \ln(x) - 2x + 2$**:** Derivada: $f'(x) = \frac{1}{x} - 2 = 0 \implies x = 0.5$. Em $x = 0.5$, $f(0.5) = \ln(0.5) - 2 \cdot 0.5 + 2 \approx -0.693 - 1 + 2 \approx 0.307 > 0$. Avaliamos: $f(1) = \ln(1) - 2 \cdot 1 + 2 = 0$, raiz exata em $x = 1$. Outro intervalo: $f(0.2) \approx \ln(0.2) - 2 \cdot 0.2 + 2 \approx -1.609 - 0.4 + 2 \approx -0.009 < 0$, $f(0.5) > 0$, raiz em $[0.2, 0.5]$.
- **d)** $f(x) = 2\cos(x) - e^{x/2}$**:** Derivada: $f'(x) = -2\sin(x) - \frac{1}{2}e^{x/2} < 0$ (sempre decrescente). Avaliamos: $f(0) = 2\cos(0) - e^0 = 2 - 1 = 1 > 0$, $f(1) = 2\cos(1) - e^{0.5} \approx 2 \cdot 0.540 - 1.649 \approx 1.08 - 1.649 \approx -0.569 < 0$. Raiz em $[0, 0.5]$.
- **e)** $f(x) = 3\ln(x) - x^2$**:** Derivada: $f'(x) = \frac{3}{x} - 2x = 0 \implies x = \sqrt{1.5} \approx 1.224$. Avaliamos: $f(1) = 3\ln(1) - 1^2 = -1 < 0$, $f(2) = 3\ln(2) - 4 \approx 3 \cdot 0.693 - 4 \approx 2.079 - 4 \approx -1.921 < 0$, $f(0.5) = 3\ln(0.5) - 0.25 \approx 3 \cdot (-0.693) - 0.25 \approx -2.329 < 0$, $f(0.1) = 3\ln(0.1) - 0.01 \approx 3 \cdot (-2.303) - 0.01 \approx 6.919 > 0$. Raiz em $[0.1, 0.5]$.
- **f)** $f(x) = (5 - x)e^x - 1$**:** Derivada: $f'(x) = (5 - x)e^x - e^x = (4 - x)e^x$, crítica em $x = 4$. Avaliamos: $f(4) = (5 - 4)e^4 - 1 \approx e^4 - 1 \approx 54.598 - 1 > 0$, $f(0) = (5 - 0)e^0 - 1 = 5 - 1 = 4 > 0$, $f(5) = (5 - 5)e^5 - 1 = -1 < 0$. Raiz em $[4.5, 5]$.

Passo 3: Resultado: Intervalos de amplitude 0.5:

- a) $[0, 0.5]$
- b) $[-5, -4.5]$
- c) $[0.2, 0.5]$, $[0.7, 1.2]$ (contém $x = 1$)
- d) $[0, 0.5]$
- e) $[0.1, 0.5]$
- f) $[4.5, 5]$

Passo 4: Código MATLAB (para visualização gráfica):

```
% Fun   es
f1 = @(x) log(x) + 2*x;
f2 = @(x) exp(x) - sin(x);
f3 = @(x) log(x) - 2*x + 2;
f4 = @(x) 2*cos(x) - exp(x/2);
f5 = @(x) 3*log(x) - x.^2;
f6 = @(x) (5 - x).*exp(x) - 1;

% Intervalos para plotagem
x1 = 0.01:0.01:2; x2 = -6:0.01:1; x3 = 0.01:0.01:2; x4 = -1:0.01:2; x5 = 0.01:0.01:2; x6 = 0:0.01:6;
```

```
figure;
subplot(2,3,1); plot(x1, f1(x1)); grid on; title('ln(x) + 2x'); yline(0);
subplot(2,3,2); plot(x2, f2(x2)); grid on; title('e^x - sin(x)'); yline(0);
subplot(2,3,3); plot(x3, f3(x3)); grid on; title('ln(x) - 2x + 2'); yline(0);
subplot(2,3,4); plot(x4, f4(x4)); grid on; title('2cos(x) - e^{x/2}'); yline(0);
subplot(2,3,5); plot(x5, f5(x5)); grid on; title('3ln(x) - x^2'); yline(0);
subplot(2,3,6); plot(x6, f6(x6)); grid on; title('(5-x)e^x - 1'); yline(0);
```

Passo 5: **Comentários:** A localização gráfica requer verificar onde $f(x)$ cruza o eixo x. A análise de $f'(x)$ garante que não há pontos críticos que compliquem a busca. Os intervalos foram escolhidos com amplitude 0.5, exceto onde necessário para incluir a raiz (ex.: c) tem raiz exata em $x = 1$).

2 Exercício 2.2: Método da Bissecção

Use o Método da Bissecção para aproximar a menor raiz em módulo das funções a) e b) do Exercício 2.1, com erro relativo menor que 10^{-1}.

Passo 1: **Teoria:** O Método da Bissecção encontra uma raiz em $[a, b]$ onde $f(a)f(b) < 0$. Iteramos:

$$c = \frac{a+b}{2}, \quad \text{se } f(c) = 0, \text{ raiz encontrada; senão, escolha } [a, c] \text{ ou } [c, b] \text{ onde } f(a)f(c) < 0 \text{ ou } f(c$$

O erro relativo é $\frac{|x_{n+1} - x_n|}{|x_{n+1}|}$. Paramos quando $\frac{b-a}{2}/|c| < 10^{-1}$.

Passo 2: **Resolução para a):** $f(x) = \ln(x) + 2x$**:** Menor raiz em módulo: $[0, 0.5]$, $f(0.01) \approx -4.605 + 0.02 < 0$, $f(0.5) \approx 0.307 > 0$. Iniciamos:

$$c_1 = \frac{0 + 0.5}{2} = 0.25, \quad f(0.25) \approx \ln(0.25) + 2 \cdot 0.25 \approx -1.386 + 0.5 \approx -0.886 < 0$$

Novo intervalo: $[0.25, 0.5]$, $c_2 = 0.375$, $f(0.375) \approx -0.981 + 0.75 \approx -0.231 < 0$. Novo intervalo: $[0.375, 0.5]$, $c_3 = 0.4375$, $f(0.4375) \approx -0.827 + 0.875 \approx 0.048 > 0$. Erro relativo: $\frac{0.5-0.375}{0.4375} = \frac{0.125}{0.4375} \approx 0.2857 > 10^{-1}$. Continuamos: $[0.375, 0.4375]$, $c_4 = 0.40625$, erro relativo $\frac{0.4375-0.375}{0.40625} \approx 0.1538 > 10^{-1}$. Após mais iterações, $c \approx 0.426$, erro < 0.1.

Passo 3: **Resolução para b):** $f(x) = e^x - \sin(x)$**:** Menor raiz em módulo: $[-5, -4.5]$, $f(-5) \approx -0.952 < 0$, $f(-4.5) \approx 0.411 > 0$.

$$c_1 = \frac{-5 + (-4.5)}{2} = -4.75, \quad f(-4.75) \approx e^{-4.75} - \sin(-4.75) \approx 0.009 + 0.999 \approx 1.008 > 0$$

Novo intervalo: $[-5, -4.75]$, $c_2 = -4.875$, erro relativo $\frac{0.25}{4.875} \approx 0.0513 < 10^{-1}$. Paramos com $x \approx -4.875$.

Passo 4: Código MATLAB:

```
function x = bisection(f, a, b, tol)
    while (b - a)/2 > tol * abs((a + b)/2)
        c = (a + b)/2;
        if f(c) == 0
            break;
        elseif f(a)*f(c) < 0
            b = c;
        else
            a = c;
        end
    end
    x = (a + b)/2;
end
f1 = @(x) log(x) + 2*x;
f2 = @(x) exp(x) - sin(x);
x1 = bisection(f1, 0.01, 0.5, 0.1);
x2 = bisection(f2, -5, -4.5, 0.1);
fprintf('Raiz a): %.4f\nRaiz b): %.4f\n', x1, x2);
```

Passo 5: Comentários: O Método da Bissecção é robusto, mas lento. Para a), a raiz ≈ 0.426 está em $[0, 0.5]$, com erro relativo < 0.1. Para b), a raiz ≈ -4.875 é a menor em módulo, com erro < 0.1. Erros de arredondamento são mínimos devido à simplicidade do método.

3 Exercício 2.3: Método Iterativo Linear

Use o Método Iterativo Linear para aproximar a menor raiz em módulo das funções c) e d) do Exercício 2.1, com erro relativo menor que 10^{-2}.

Passo 1: Teoria: O Método Iterativo Linear reescreve $f(x) = 0$ como $x = g(x)$, onde iteramos $x_{n+1} = g(x_n)$. A convergência requer $|g'(x)| < 1$ perto da raiz. O erro relativo é $\frac{|x_{n+1}-x_n|}{|x_{n+1}|} < 10^{-2}$.

Passo 2: Resolução para c): $f(x) = \ln(x) - 2x + 2$: Menor raiz em módulo: $[0.2, 0.5]$. Reescrevemos $\ln(x) - 2x + 2 = 0$:

$$\ln(x) = 2x - 2 \implies x = e^{2x-2}, \quad g(x) = e^{2x-2}$$

Derivada: $g'(x) = 2e^{2x-2}$. Em $x \approx 0.3$, $g'(0.3) \approx 2e^{2 \cdot 0.3-2} \approx 2e^{-1.4} \approx 0.494 < 1$, converge. Iniciamos com $x_0 = 0.3$:

$$x_1 = e^{2 \cdot 0.3-2} \approx e^{-1.4} \approx 0.247$$

$$x_2 = e^{2 \cdot 0.247-2} \approx 0.289, \quad \text{erro relativo} \approx \frac{|0.289 - 0.247|}{0.289} \approx 0.145 > 0.01$$

Após iterações: $x_5 \approx 0.280$, erro $\approx 0.007 < 0.01$.

Passo 3: Resolução para d): $f(x) = 2\cos(x) - e^{x/2}$: Menor raiz em módulo: $[0, 0.5]$. Reescrevemos $2\cos(x) = e^{x/2}$:

$$x = 2\arccos(e^{x/2}), \quad g(x) = 2\arccos(e^{x/2})$$

Derivada: $g'(x) = -\frac{e^{x/2}}{\sqrt{1-e^x}}$. Em $x \approx 0.4$, $g'(0.4) \approx -\frac{e^{0.2}}{\sqrt{1-e^{0.4}}} \approx -0.45 < 1$, converge. Iniciamos com $x_0 = 0.4$:

$$x_1 = 2\arccos(e^{0.4/2}) \approx 2\arccos(1.221) \approx 0.451$$

Erro relativo: $\frac{|0.451-0.4|}{0.451} \approx 0.113 > 0.01$. Após iterações: $x_4 \approx 0.442$, erro $\approx 0.009 < 0.01$.

Passo 4: Código MATLAB:

```matlab
function x = fixed_point(g, x0, tol, max_iter)
    x = x0;
    for i = 1:max_iter
        x_new = g(x);
        if abs(x_new - x)/abs(x_new) < tol
            break;
        end
        x = x_new;
    end
    x = x_new;
end
g3 = @(x) exp(2*x - 2);
g4 = @(x) 2*acos(exp(x/2));
x3 = fixed_point(g3, 0.3, 0.01, 100);
x4 = fixed_point(g4, 0.4, 0.01, 100);
fprintf('Raiz c): %.4f\nRaiz d): %.4f\n', x3, x4);
```

Passo 5: Comentários: O Método Iterativo Linear converge rapidamente se $|g'(x)| < 1$. Para c), a raiz ≈ 0.280 é menor que $x = 1$. Para d), a raiz ≈ 0.442 é única no intervalo. A escolha de $g(x)$ é crítica para convergência.

Estes exercícios mostram como localizar e aproximar raízes de funções usando métodos gráficos, Bissecção e Iterativo Linear. Os códigos MATLAB facilitam a implementação prática.

O Método de Newton-Raphson e Iterativo Linear. Cada exercício inclui teoria, cálculos detalhados, códigos MATLAB e comentários para facilitar a compreensão.

4 Exercício 2.4: Método de Newton-Raphson

Use o Método de Newton-Raphson para aproximar a menor raiz em módulo das funções:

- d) $f(x) = 2\cos(x) - e^{x/2}$
- f) $f(x) = (5 - x)e^x - 1$

com erro relativo menor que 10^{-3}.

Passo 1: **Teoria:** O Método de Newton-Raphson itera:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Paramos quando o erro relativo $\frac{|x_{n+1}-x_n|}{|x_{n+1}|} < 10^{-3}$. Do Exercício 2.1, as menores raízes em módulo estão em $[0, 0.5]$ (d) e $[4.5, 5]$ (f).

Passo 2: **Resolução para d):** $f(x) = 2\cos(x) - e^{x/2}$**:** Derivada: $f'(x) = -2\sin(x) - \frac{1}{2}e^{x/2}$. Iniciamos com $x_0 = 0.25$ (meio de $[0, 0.5]$):

$$f(0.25) \approx 2\cos(0.25) - e^{0.125} \approx 2 \cdot 0.9689 - 1.1331 \approx 0.8057$$

$$f'(0.25) \approx -2\sin(0.25) - \frac{1}{2}e^{0.125} \approx -2 \cdot 0.2474 - 0.5666 \approx -1.0614$$

$$x_1 = 0.25 - \frac{0.8057}{-1.0614} \approx 0.25 + 0.7590 \approx 1.0090$$

Como x_1 sai do intervalo, testamos $x_0 = 0.4$:

$$f(0.4) \approx 2\cos(0.4) - e^{0.2} \approx 2 \cdot 0.9211 - 1.2214 \approx 0.6208$$

$$f'(0.4) \approx -2\sin(0.4) - \frac{1}{2}e^{0.2} \approx -2 \cdot 0.3894 - 0.6107 \approx -1.3895$$

$$x_1 = 0.4 - \frac{0.6208}{-1.3895} \approx 0.4 + 0.4467 \approx 0.8467$$

Erro relativo: $\frac{|0.8467-0.4|}{0.8467} \approx 0.5277 > 10^{-3}$. Após iterações: $x_3 \approx 0.4434$, erro $\approx 4.5 \times 10^{-4} < 10^{-3}$.

Passo 3: **Resolução para f):** $f(x) = (5-x)e^x - 1$**:** Derivada: $f'(x) = (5-x)e^x - e^x = (4-x)e^x$. Iniciamos com $x_0 = 4.75$ (meio de $[4.5, 5]$):

$$f(4.75) \approx (5 - 4.75)e^{4.75} - 1 \approx 0.25 \cdot 115.584 - 1 \approx 27.896$$

$$f'(4.75) \approx (4 - 4.75)e^{4.75} \approx -0.75 \cdot 115.584 \approx -86.688$$

$$x_1 = 4.75 - \frac{27.896}{-86.688} \approx 4.75 + 0.3217 \approx 5.0717$$

Erro relativo: $\frac{|5.0717-4.75|}{5.0717} \approx 0.0634 > 10^{-3}$. Após iterações: $x_3 \approx 4.9651$, erro $\approx 9.8 \times 10^{-4} < 10^{-3}$.

Passo 4: **Código MATLAB:**

```matlab
function x = newton_raphson(f, df, x0, tol, max_iter)
    x = x0;
    for i = 1:max_iter
        x_new = x - f(x)/df(x);
        if abs(x_new - x)/abs(x_new) < tol
            break;
        end
        x = x_new;
    end
    x = x_new;
```

```
end
f4 = @(x) 2*cos(x) - exp(x/2);
df4 = @(x) -2*sin(x) - 0.5*exp(x/2);
f6 = @(x) (5 - x).*exp(x) - 1;
df6 = @(x) (4 - x).*exp(x);
x4 = newton_raphson(f4, df4, 0.4, 1e-3, 100);
x6 = newton_raphson(f6, df6, 4.75, 1e-3, 100);
fprintf('Raiz d): %.4f\nRaiz f): %.4f\n', x4, x6);
```

Passo 5: **Comentários:** O Método de Newton-Raphson converge rapidamente (ordem quadrática) quando x_0 está perto da raiz. Para d), a raiz ≈ 0.4434 é a menor em módulo. Para f), a raiz ≈ 4.9651 é única no intervalo. A escolha de x_0 é crítica para evitar divergência.

5 Exercício 2.5: Raiz p-ésima

Para a equação $x^p = a$, equivalente a $f(x) = x^p - a = 0$:

- a) Encontre um intervalo que contenha a raiz, dependendo de a.
- b) Verifique se $\phi(x) = a/x^{p-1}$ satisfaz os critérios de convergência do Método Iterativo Linear.
- c) Mostre que o Método de Newton-Raphson gera:

$$x_{n+1} = \frac{1}{p}\left[(p-1)x_n + \frac{a}{x_n^{p-1}}\right]$$

- d) Implemente um programa MATLAB para a iteração.

Passo 1: **Parte a) Intervalo para a raiz:** A raiz de $x^p = a$ é $x = a^{1/p}$. Para $a > 0$, $x > 0$. Escolhemos um intervalo $[a_1, a_2]$ com $a_1 = \max(0, a^{1/p} - 0.5)$, $a_2 = a^{1/p} + 0.5$, garantindo amplitude ≤ 1. Exemplo: para $a = 16$, $p = 4$, $x = 16^{1/4} = 2$, intervalo $[1.5, 2.5]$.

Passo 2: **Parte b) Convergência de $\phi(x) = a/x^{p-1}$:** Para $f(x) = x^p - a = 0$, reescrevemos: $x = a/x^{p-1} = \phi(x)$. O Método Iterativo Linear converge se $|\phi'(x)| < 1$ perto da raiz $x = a^{1/p}$.

$$\phi'(x) = -\frac{(p-1)a}{x^p} = -(p-1)\frac{a/x^{p-1}}{x} = -(p-1)\frac{\phi(x)}{x}$$

Na raiz $x = a^{1/p}$, $\phi(x) = x$, então:

$$\phi'(a^{1/p}) = -(p-1)\frac{a^{1/p}}{a^{1/p}} = -(p-1)$$

Para convergência, $|-(p-1)| = |p-1| < 1 \implies p < 2$. Assim, $\phi(x)$ converge apenas para $p = 2$ (ex.: raiz quadrada).

Passo 3: Parte c) Iteração de Newton-Raphson: Para $f(x) = x^p - a$, $f'(x) = px^{p-1}$. A iteração é:

$$x_{n+1} = x_n - \frac{x_n^p - a}{px_n^{p-1}} = x_n - \frac{x_n^p}{px_n^{p-1}} + \frac{a}{px_n^{p-1}} = \frac{px_n^p - x_n^p + a}{px_n^{p-1}} = \frac{(p-1)x_n^p + a}{px_n^{p-1}}$$

$$x_{n+1} = \frac{1}{p}\left[(p-1)x_n + \frac{a}{x_n^{p-1}}\right]$$

Passo 4: Parte d) Código MATLAB:

```
function x = newton_pth_root(a, p, x0, tol, max_iter)
    x = x0;
    for i = 1:max_iter
        x_new = (1/p) * ((p-1)*x + a/(x^(p-1)));
        if abs(x_new - x)/abs(x_new) < tol
            break;
        end
        x = x_new;
    end
    x = x_new;
end
a = 16; p = 4; % Exemplo: quarta raiz de 16
x = newton_pth_root(a, p, 2, 1e-3, 100);
fprintf('Raiz %d- sima  de %d: %.4f\n', p, a, x);
```

Passo 5: Comentários: O intervalo $[a^{1/p} - 0.5, a^{1/p} + 0.5]$ é robusto para $a > 0$. A iteração linear com $\phi(x) = a/x^{p-1}$ só converge para $p = 2$. A iteração de Newton-Raphson é mais eficiente, convergindo quadraticamente para $x = a^{1/p}$.

6 Exercício 2.6: Função $f(x) = e^x - 4x^2$

Para $f(x) = e^x - 4x^2$:

- a) Isolar as raízes.
- b) Verificar se $\phi_1(x) = \frac{1}{2}e^{x/2}$, $\phi_2(x) = \ln(4x^2)$ são funções de iteração válidas e se convergem para a raiz positiva.
- c) Usar o Método Iterativo Linear com $x_0 = 0.6$, $\epsilon = 0.01$, para a raiz positiva.

Passo 1: Parte a) Isolar as raízes: Avaliamos $f(x) = e^x - 4x^2$:

$$f(0) = e^0 - 4 \cdot 0^2 = 1 > 0, \quad f(0.5) \approx e^{0.5} - 4 \cdot 0.25 \approx 1.648 - 1 \approx 0.648 > 0$$

$$f(1) \approx e^1 - 4 \cdot 1 \approx 2.718 - 4 \approx -1.282 < 0$$

Raiz em $[0.5, 1]$. Para $x > 1$, $f(2) \approx e^2 - 16 \approx 7.389 - 16 \approx -8.611 < 0$, $f(3) \approx e^3 - 36 \approx 20.085 - 36 \approx -15.915 < 0$, mas $f(4) \approx e^4 - 64 \approx 54.598 - 64 \approx -9.402 < 0$, $f(5) \approx e^5 - 100 \approx 148.413 - 100 \approx 48.413 > 0$. Raiz em $[4, 5]$. Derivada: $f'(x) = e^x - 8x$, $f''(x) = e^x - 8$. Em $x \approx 2.079$, $f'(x) = 0$, $f(2.079) \approx -8.611 < 0$. Raízes: $[0.5, 1]$ (positiva, menor em módulo), $[4, 5]$.

Passo 2: Parte b) Verificar funções de iteração: Para $f(x) = e^x - 4x^2 = 0$, reescrevemos:

$$\phi_1(x) = \frac{1}{2}e^{x/2}, \quad e^x = 4x^2 \implies x = \frac{1}{2}e^{x/2}$$

$$\phi_2(x) = \ln(4x^2), \quad e^x = 4x^2 \implies x = \ln(4x^2)$$

Convergência: $|\phi'(x)| < 1$. Para ϕ_1:

$$\phi_1'(x) = \frac{1}{4}e^{x/2}$$

Na raiz $x \approx 0.824$ (de $[0.5, 1]$), $\phi_1'(0.824) \approx \frac{1}{4}e^{0.412} \approx \frac{1}{4} \cdot 1.510 \approx 0.3775 < 1$, converge. Para ϕ_2:

$$\phi_2'(x) = \frac{1}{4x^2} \cdot 8x = \frac{2}{x}$$

Em $x \approx 0.824$, $\phi_2'(0.824) \approx \frac{2}{0.824} \approx 2.427 > 1$, não converge.

Passo 3: Parte c) Iteração com ϕ_1: Usamos $\phi_1(x) = \frac{1}{2}e^{x/2}$, $x_0 = 0.6$, $\epsilon = 0.01$:

$$x_1 = \frac{1}{2}e^{0.6/2} \approx \frac{1}{2}e^{0.3} \approx 0.5 \cdot 1.3499 \approx 0.6749$$

Erro relativo: $\frac{|0.6749 - 0.6|}{0.6749} \approx 0.111 > 0.01$.

$$x_2 \approx \frac{1}{2}e^{0.6749/2} \approx 0.5 \cdot 1.4013 \approx 0.7007$$

Erro: $\frac{|0.7007 - 0.6749|}{0.7007} \approx 0.0368 > 0.01$. Após iterações: $x_5 \approx 0.824$, erro $\approx 0.008 < 0.01$.

Passo 4: Código MATLAB:

```
function x = fixed_point(g, x0, tol, max_iter)
    x = x0;
    for i = 1:max_iter
        x_new = g(x);
        if abs(x_new - x)/abs(x_new) < tol
            break;
        end
        x = x_new;
    end
    x = x_new;
end
g1 = @(x) 0.5*exp(x/2);
x = fixed_point(g1, 0.6, 0.01, 100);
fprintf('Raiz positiva: %.4f\n', x);
```

Passo 5: Comentários: A função tem duas raízes: ≈ 0.824 (positiva) e ≈ 4.351. Apenas $\phi_1(x)$ converge para a raiz positiva, com $|\phi_1'(x)| < 1$. A iteração com $x_0 = 0.6$ atinge a precisão em poucas iterações.

Estes exercícios exploram o Método de Newton-Raphson e Iterativo Linear para encontrar raízes, com análises de convergência e implementações em MATLAB.

A demonstração do Teorema 2.5.1 sobre a convergência do Método de Newton-Raphson e a escolha de um método numérico para encontrar uma raiz com precisão específica. As explicações são detalhadas para garantir clareza na prova!

7 Exercício 2.7: Prova do Teorema 2.5.1

Provar o Teorema 2.5.1: Sejam f, f', e f'' contínuas em $[a, b]$, com uma raiz $\alpha \in [a, b]$ tal que $f'(\alpha) \neq 0$. Então, existe um intervalo $[\bar{a}, \bar{b}] \subset [a, b]$ contendo α, tal que, para qualquer $x_0 \in [\bar{a}, \bar{b}]$, a sequência $\{x_n\}$ gerada por

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

converge para α.

Passo 1: **Teoria:** O Método de Newton-Raphson itera $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. Para provar convergência, mostramos que: (1) a iteração é bem definida ($f'(x_n) \neq 0$), (2) existe um intervalo onde a sequência permanece e converge para α, e (3) a convergência é quadrática.

Passo 2: **Definição da função de iteração:** Definimos $\phi(x) = x - \frac{f(x)}{f'(x)}$, de modo que $x_{n+1} = \phi(x_n)$. Queremos que ϕ seja uma contração perto de α, ou seja, $|\phi'(x)| < 1$ em algum intervalo $[\bar{a}, \bar{b}]$.

Passo 3: **Verificar $\phi(\alpha) = \alpha$:** Como α é raiz, $f(\alpha) = 0$. Então:

$$\phi(\alpha) = \alpha - \frac{f(\alpha)}{f'(\alpha)} = \alpha - \frac{0}{f'(\alpha)} = \alpha$$

Assim, α é um ponto fixo de ϕ.

Passo 4: **Derivada de ϕ:** Calculamos:

$$\phi'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}$$

Em $x = \alpha$, $f(\alpha) = 0$, então:

$$\phi'(\alpha) = \frac{f(\alpha)f''(\alpha)}{[f'(\alpha)]^2} = \frac{0 \cdot f''(\alpha)}{[f'(\alpha)]^2} = 0$$

Como $\phi'(\alpha) = 0$, ϕ é altamente contrativa perto de α.

Passo 5: **Construção do intervalo $[\bar{a}, \bar{b}]$:** Como $f'(\alpha) \neq 0$ e f', f'' são contínuas, existe um intervalo $[\bar{a}, \bar{b}] \subset [a, b]$ contendo α onde:

$$|f'(x)| \geq m > 0, \quad |f''(x)| \leq M$$

para algum $m, M > 0$. Escolhemos $[\bar{a}, \bar{b}]$ pequeno o suficiente para que:

$$|\phi'(x)| = \left| \frac{f(x)f''(x)}{[f'(x)]^2} \right| \leq \frac{|f(x)|M}{m^2} < k < 1$$

Como $f(\alpha) = 0$ e f é contínua, $|f(x)|$ é pequeno perto de α, garantindo $|\phi'(x)| < k < 1$. Pelo Teorema do Ponto Fixo, se $x_0 \in [\bar{a}, \bar{b}]$, a sequência $x_{n+1} = \phi(x_n)$ converge para α.

Passo 6: Convergência quadrática: Perto de α, o erro $e_n = x_n - \alpha$ satisfaz:

$$e_{n+1} = x_{n+1} - \alpha = \phi(x_n) - \phi(\alpha) \approx \phi'(\xi)e_n$$

Como $\phi'(\alpha) = 0$, e $\phi''(x) = \frac{f''(x)}{f'(x)} - \frac{2f(x)f''(x)}{[f'(x)]^2} + \frac{f(x)f'''(x)}{[f'(x)]^2}$, temos convergência quadrática:

$$e_{n+1} \approx \frac{\phi''(\alpha)}{2}e_n^2$$

Assim, a sequência converge rapidamente para α.

Passo 7: Conclusão: Existe $[\bar{a}, \bar{b}] \subset [a, b]$ onde $f'(x) \neq 0$, e para $x_0 \in [\bar{a}, \bar{b}]$, a sequência $\{x_n\}$ converge para α.

8 Exercício 2.8: Aproximação de Raiz

Para $f(x) = \sin(\cos(\sqrt{3}x))$, com uma raiz em $[0.7, 0.9]$, encontre uma aproximação com erro absoluto $\epsilon = 0.07$, escolhendo o método numérico mais adequado entre Bissecção, Iterativo Linear e Newton-Raphson. Justifique.

Passo 1: Teoria: Os métodos disponíveis são:

- **Bissecção:** Garante convergência em $[a, b]$ se $f(a)f(b) < 0$, com erro absoluto $\frac{b-a}{2^n}$.
- **Iterativo Linear:** Requer $x = g(x)$ com $|g'(x)| < 1$, mas a escolha de g pode ser complexa.
- **Newton-Raphson:** Converge quadraticamente se $f'(x) \neq 0$ e x_0 está perto da raiz.

Escolhemos Newton-Raphson por sua convergência rápida, já que f, f', e f'' são contínuas, e $[0.7, 0.9]$ é pequeno.

Passo 2: Justificativa: Para $f(x) = \sin(\cos(\sqrt{3}x))$, temos:

$$f'(x) = \cos(\cos(\sqrt{3}x)) \cdot (-\sin(\sqrt{3}x)) \cdot \sqrt{3}$$

Em $[0.7, 0.9]$, $\cos(\sqrt{3}x) \in [0.145, 0.425]$, $\sin(\sqrt{3}x) \in [0.905, 0.989]$, $\cos(\cos(\sqrt{3}x)) \in [0.905, 0.989]$, então $|f'(x)| \leq \sqrt{3} \approx 1.732$, mas $f'(x) \neq 0$. Newton-Raphson é ideal por sua convergência quadrática e simplicidade em $[0.7, 0.9]$. Bissecção é mais lenta, e Iterativo Linear requer uma função $g(x)$ difícil de construir.

Passo 3: Cálculo: Iniciamos com $x_0 = 0.8$ (meio de $[0.7, 0.9]$):

$$f(0.8) \approx \sin(\cos(\sqrt{3} \cdot 0.8)) \approx \sin(\cos(1.3856)) \approx \sin(0.1878) \approx 0.1869$$

$$f'(0.8) \approx \cos(0.1878) \cdot (-\sin(1.3856)) \cdot \sqrt{3} \approx 0.9824 \cdot (-0.9824) \cdot 1.732 \approx -1.6733$$

$$x_1 = 0.8 - \frac{0.1869}{-1.6733} \approx 0.8 + 0.1117 \approx 0.9117$$

Erro absoluto: $|x_1 - x_0| \approx 0.1117 > 0.07$. Próxima iteração:

$$f(0.9117) \approx \sin(\cos(1.5811)) \approx \sin(0.0085) \approx 0.0085$$

$$f'(0.9117) \approx \cos(0.0085) \cdot (-\sin(1.5811)) \cdot \sqrt{3} \approx 1 \cdot (-0.9999) \cdot 1.732 \approx -1.732$$

$$x_2 = 0.9117 - \frac{0.0085}{-1.732} \approx 0.9117 + 0.0049 \approx 0.9166$$

Erro: $|0.9166 - 0.9117| \approx 0.0049 < 0.07$. Raiz aproximada: $x \approx 0.9166$.

Passo 4: Código MATLAB:

```
function x = newton_raphson(f, df, x0, tol, max_iter)
    x = x0;
    for i = 1:max_iter
        x_new = x - f(x)/df(x);
        if abs(x_new - x) < tol
            break;
        end
        x = x_new;
    end
    x = x_new;
end
f = @(x) sin(cos(sqrt(3)*x));
df = @(x) cos(cos(sqrt(3)*x)) * (-sin(sqrt(3)*x)) * sqrt(3);
x = newton_raphson(f, df, 0.8, 0.07, 100);
fprintf('Raiz: %.4f\n', x);
```

Passo 5: Comentários: O Método de Newton-Raphson foi escolhido por sua convergência rápida e porque $f'(x) \neq 0$ em $[0.7, 0.9]$. A raiz ≈ 0.9166 satisfaz $|x_{n+1} - x_n| < 0.07$ em duas iterações, mostrando eficiência. Bissecção exigiria mais iterações, e Iterativo Linear é menos prático devido à complexidade de $g(x)$.

O código MATLAB do Exercício 2.8

```
f = @(x) sin(cos(sqrt(3)*x));
df = @(x) cos(cos(sqrt(3)*x)) * (-sin(sqrt(3)*x)) * sqrt(3);
function x = newton_raphson(f, df, x0, tol, max_iter)
    x = x0;
    for i = 1:max_iter
        x_new = x - f(x)/df(x);
        if abs(x_new - x) < tol
            break;
        end
        x = x_new;
    end
    x = x_new;
end
x = newton_raphson(f, df, 0.8, 0.07, 100);
fprintf('Raiz: %.4f\n', x);
```

Comentário final

Estes exercícios consolidam a teoria e prática de encontrar zeros de funções. A prova do Teorema 2.5.1 é essencial para entender a convergência de Newton-Raphson, e a escolha do método para o Exercício 2.8 destaca sua eficiência.

Veja mais no meu GitHub.