

Resoluções de Exercícios de Cálculo Numérico – Capítulo 1

Prof. Ana Isabel C. – Numerical Methods

Junho 2025

Introdução

Este documento apresenta a resolução passo a passo dos exercícios do Capítulo 1 de Cálculo Numérico, focado em métodos numéricos básicos e sua implementação em MATLAB. Cada exercício inclui a teoria, cálculos detalhados, códigos MATLAB e comentários sobre os resultados.

1 Exercício 1.1: Aproximação da Derivada

Aproximar a derivada $f'(x)$ de $f(x) = \sin(x)$ em $x = \pi$ e $x = 0$ usando a diferença progressiva:

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

com $h = 0.1, 0.01, 0.001, \dots, 10^{-10}$. Comentar os resultados.

Passo 1: Teoria: A derivada $f'(x)$ pode ser aproximada pela diferença progressiva:

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

Para $f(x) = \sin(x)$, sabemos que $f'(x) = \cos(x)$. Assim, $f'(\pi) = \cos(\pi) = -1$ e $f'(0) = \cos(0) = 1$. Vamos calcular a aproximação e o erro para diferentes h .

Passo 2: Cálculo para $x = \pi$: Sabemos que $\sin(\pi) = 0$, $\cos(\pi) = -1$. Para $h = 0.1$:

$$f(\pi + 0.1) = \sin(\pi + 0.1) \approx \sin(3.2415926535) \approx -0.099833$$

$$f'(\pi) \approx \frac{\sin(\pi + 0.1) - \sin(\pi)}{0.1} = \frac{-0.099833 - 0}{0.1} \approx -0.99833$$

Erro: $|-1 - (-0.99833)| = 0.00167$. Para $h = 0.01$:

$$f(\pi + 0.01) \approx \sin(3.1515926535) \approx -0.00999983$$

$$f'(\pi) \approx \frac{-0.00999983 - 0}{0.01} \approx -0.999983$$

Erro: $|-1 - (-0.999983)| \approx 1.7 \times 10^{-5}$. Para $h = 10^{-10}$, o erro diminui inicialmente, mas pode aumentar devido a erros de arredondamento (ver comentários).

Passo 3: Cálculo para $x = 0$: Sabemos que $\sin(0) = 0$, $\cos(0) = 1$. Para $h = 0.1$:

$$f(0 + 0.1) = \sin(0.1) \approx 0.099833$$

$$f'(0) \approx \frac{0.099833 - 0}{0.1} \approx 0.99833$$

Erro: $|1 - 0.99833| = 0.00167$. Para $h = 0.01$:

$$f'(0) \approx \frac{\sin(0.01) - 0}{0.01} \approx \frac{0.00999983}{0.01} \approx 0.999983$$

Erro: $|1 - 0.999983| \approx 1.7 \times 10^{-5}$.

Passo 4: Código MATLAB:

```
1 x = [pi, 0]; % Pontos de avalia o
2 h = 10.^(-1:-1:-10); % Valores de h
3 f = @(x) sin(x);
4 df_exact = @(x) cos(x); % Derivada exata
5 for i = 1:length(x)
6     fprintf('x = %.2f, f''(x) exato = %.4f\n', x(i), df_exact
7         (x(i)));
8     for j = 1:length(h)
9         df_approx = (f(x(i) + h(j)) - f(x(i))) / h(j);
10        error = abs(df_exact(x(i)) - df_approx);
11        fprintf('h = %.0e, f''(x) aprox = %.6f, erro = %.2e\n
12            ', h(j), df_approx, error);
13    end
14 end
```

Passo 5: Comentários: Para h grande (0.1), o erro de truncamento é dominante, mas pequeno (≈ 0.00167). Conforme h diminui, o erro reduz (ex.: 1.7×10^{-5} para $h = 0.01$), mas para $h \leq 10^{-8}$, erros de arredondamento crescem devido à precisão finita do MATLAB (double precision, 10^{-16}). Isso é mais evidente em $x = \pi$, onde a subtração $\sin(\pi + h) - \sin(\pi)$ amplifica erros numéricos.

2 Exercício 1.2: Soma Numérica

Calcular:

$$A + \sum_{k=1}^{10^7} 10^{-7}$$

para $A = 10, 10^2, 10^3, \dots, 10^{15}$. Comentar os resultados.

Passo 1: Teoria: A soma $\sum_{k=1}^{10^7} 10^{-7} = 10^7 \cdot 10^{-7} = 1$. Assim, o resultado exato é $A + 1$. Vamos calcular numericamente e verificar a precisão.

Passo 2: Cálculo para $A = 10$: Somamos 10^{-7} 10^7 vezes:

$$S = 10 + \sum_{k=1}^{10^7} 10^{-7} \approx 10 + 1 = 11$$

Para $A = 10^{15}$:

$$S = 10^{15} + \sum_{k=1}^{10^7} 10^{-7} \approx 10^{15} + 1$$

Numericamente, somar 10^{-7} muitas vezes pode acumular erros de arredondamento.

Passo 3: Código MATLAB:

```
1 A = 10.^(1:15);
2 for i = 1:length(A)
3     sum = A(i);
4     for k = 1:1e7
5         sum = sum + 1e-7;
6     end
7     fprintf('A = %.0e, Soma = %.16f, Erro = %.2e\n', A(i),
8           sum, abs(sum - (A(i) + 1)));
9 end
```

Passo 4: Comentários: Para $A = 10$, a soma é precisa ($S \approx 11$). Para $A = 10^{15}$, o erro de arredondamento cresce porque 10^{-7} é muito pequeno comparado a 10^{15} , e a soma iterativa acumula erros de precisão (double precision, 10^{-16}). O resultado pode ser $S \approx 10^{15}$, sem captar o +1, devido à perda de dígitos significativos.

3 Exercício 1.3: Epsilon da Máquina

Calcular o *epsilon da máquina* usando o algoritmo:

- Input: A , número que representa a grandeza.
- $Ep \leftarrow 1$
- Enquanto $A + Ep > A$, faça: $Ep \leftarrow Ep/2$
- Output: $2 \cdot Ep$

para $A = 1, 10, 100, 1000$. Comentar os resultados.

Passo 1: Teoria: O *epsilon da máquina* (ϵ) é o menor número tal que $1 + \epsilon > 1$ na aritmética do computador. O algoritmo reduz Ep até que $A + Ep = A$ devido à precisão finita.

Passo 2: Cálculo para $A = 1$: Inicializamos $Ep = 1$. Iteramos:

$$1 + 1 = 2 > 1, \quad Ep \leftarrow 0.5$$

$$1 + 0.5 = 1.5 > 1, \quad Ep \leftarrow 0.25$$

Continuamos até $Ep \approx 2.22 \times 10^{-16}$, quando $1 + Ep = 1$. Então, $2 \cdot Ep \approx 4.44 \times 10^{-16}$.

Passo 3: Cálculo para outros A : Para $A = 10$, o limite é $Ep \approx 10 \cdot 2.22 \times 10^{-16}$, pois $A + Ep$ deve ser distinguível de A . Resultado: $2 \cdot Ep \approx 4.44 \times 10^{-15}$.

Passo 4: Código MATLAB:

```

1 A = [1, 10, 100, 1000];
2 for i = 1:length(A)
3     Ep = 1;
4     while A(i) + Ep > A(i)
5         Ep = Ep / 2;
6     end
7     fprintf('A = %.0f, Epsilon = %.2e\n', A(i), 2*Ep);
8 end

```

Passo 5: Comentários: O *epsilon da máquina* para $A = 1$ é $\approx 2.22 \times 10^{-16}$ (padrão IEEE 754, double precision). Para $A = 10, 100, 1000$, o epsilon efetivo escala com A , pois a precisão relativa é constante ($\epsilon_{\text{mach}} \approx 2.22 \times 10^{-16}$). Assim, $2 \cdot Ep \approx 2 \cdot A \cdot 2.22 \times 10^{-16}$.

Conclusão

Estes exercícios introduzem conceitos fundamentais de Cálculo Numérico, como aproximação de derivadas, somas numéricas e precisão da máquina, com implementações em MATLAB. Veja mais no meu GitHub.