Davi Ferreira de Souza

Prof. Douglas Rodrigues

Métodos Numéricos Computacionais

1 de junho de 2025

Relatório Trabalho Prático 4 - Métodos Numéricos Computacionais

Implementação

1. Estrutura Geral

O código implementa três métodos de interpolação polinomial, utilizados para estimar valores de funções a partir de um conjunto de pontos conhecidos:

- Interpolação de Lagrange
- Interpolação de Newton
- Interpolação de Newton-Gregory (progressiva)

Possui uma interface interativa em modo terminal, com:

- Avaliação de cada método

- Comparação de desempenho (tempo de execução e precisão)
- Geração de gráficos comparativos

2. Componentes Principais

Funções de Interpolação

Interpolação de Lagrange:

- Constrói o polinômio interpolador somando os termos $L_i(x)f(x_i)$
- Cada $L_i(x)$ é um produto de fatores lineares
- Independente de espaçamento entre os pontos
- Computacionalmente custosa para grande número de pontos

Interpolação de Newton

- Utiliza tabela de diferenças divididas
- Polinômio construído incrementalmente com termos $(x - x_i)$
- Mais eficiente para adicionar novos pontos
- Mais estável numericamente que Lagrange

Interpolação de Newton-Gregory (Progressiva)

- Requer pontos igualmente espaçados
- Utiliza diferenças finitas

- Reduzida complexidade algébrica e computacional
- Aplicável apenas para espaçamento uniforme

Funções Auxiliares

- lambdify(): converte expressões simbólicas para funções numéricas
- pprint(): exibe polinômios de forma simbólica legível
- factorial(): usado no denominador da fórmula de Gregory

3. Fluxo do Programa

1. **Inicialização:**
    - Define os pontos $(x, f(x))$ com:
    - $x = [0, 1, 2, 3, 4, 5]$
    - $f(x) = [2.1, 3.8, 5.2, 6.1, 7.8, 8.9]$
2. **Menu Interativo:**
    - Opção 1: Executa e imprime o polinômio de um método
    - Opção 2: Compara tempo de execução, valores em x=2.5 e erros
    - Opção 3: Gera gráficos comparativos dos polinômios
    - Opção 4: Encerra o programa
3. **Geração de Gráficos:**
    - Avalia os três polinômios sobre o intervalo [0,5]

    - Mostra sobreposição com os pontos experimentais
    - Permite visualização qualitativa das diferenças

4. Particularidades Importantes

- **Verificação de consistência:** garante tamanhos iguais entre x e y
- **Newton-Gregory:** valida uniformidade de espaçamento entre os pontos
- **Tratamento simbólico:** aproveita recursos do SymPy
- **Gráficos claros:** cada método com cor e estilo distinto
- **Comparação quantitativa:** erros absolutos e tempo de execução

5. Sobre o Problema Resolvido

O problema consiste em encontrar um polinômio que interpole a função f(x), com valores experimentais fornecidos nos pontos:

x	$f(x)$
0	2.1
1	3.8
2	5.2
3	6.1
4	7.8
5	8.9

Os métodos devem ser comparados quanto à precisão da estimativa de $f(2.5)$ e ao custo computacional.

6. Organização do Código

- **Modularizado por método**: funções separadas para cada técnica
- **Comentários explicativos**: detalham cada etapa
- **Verificações**: impedem execução em condições inválidas
- Menu limpo: controlado com match e comandos os.system() para limpar a tela

7. Requisitos

- Bibliotecas: **SymPy**, **NumPy**, **Matplotlib**, **Tabulate**, **OS**, **Time**

Resultados Obtidos:

Polinômios gerados:

Interpolação de Lagrange:

$$3.8x\left(\frac{5}{4}-\frac{x}{4}\right)\left(\frac{4}{3}-\frac{x}{3}\right)\left(\frac{3}{2}-\frac{x}{2}\right)(2-x)+2.6x\left(\frac{5}{3}-\frac{x}{3}\right)\left(2-\frac{x}{2}\right)(3-x)(x-1)+2.03333333333333x\left(\frac{5}{2}-\frac{x}{2}\right)(4-x)\left(\frac{x}{2}-\frac{1}{2}\right)(x-2)+1.95x(5-x)\left(\frac{x}{3}-\frac{1}{3}\right)\left(\frac{x}{2}-1\right)(x-3)+1.78x\left(\frac{x}{4}-\frac{1}{4}\right)\left(\frac{x}{3}-\frac{2}{3}\right)\left(\frac{x}{2}-\frac{3}{2}\right)(x-4)+2.1(1-x)\left(1-\frac{x}{2}\right)\left(1-\frac{x}{3}\right)\left(1-\frac{x}{4}\right)\left(1-\frac{x}{5}\right)$$

Interpolação de Newton:

$$-0.035x(x-4)(x-3)(x-2)(x-1)+0.0625000000000001x(x-3)(x-2)(x-1)-0.0333333333333336x(x-2)(x-1)-0.15x(x-1)+1.7x+2.1$$

Interpolação de Newton-Gregory:

$$-0.035x(x-4)(x-3)(x-2)(x-1)+0.0625000000000001x(x-3)(x-2)(x-1)-0.0333333333333336x(x-2)(x-1)-0.15x(x-1)+1.7x+2.1$$

Polinômio simplificado:

$$-0.035x^5 + 0.412499999999999x^4 - 1.63333333333333x^3 + 2.3875x^2 + 0.568333333333333x + 2.1$$

Plot dos polinômios gerados:

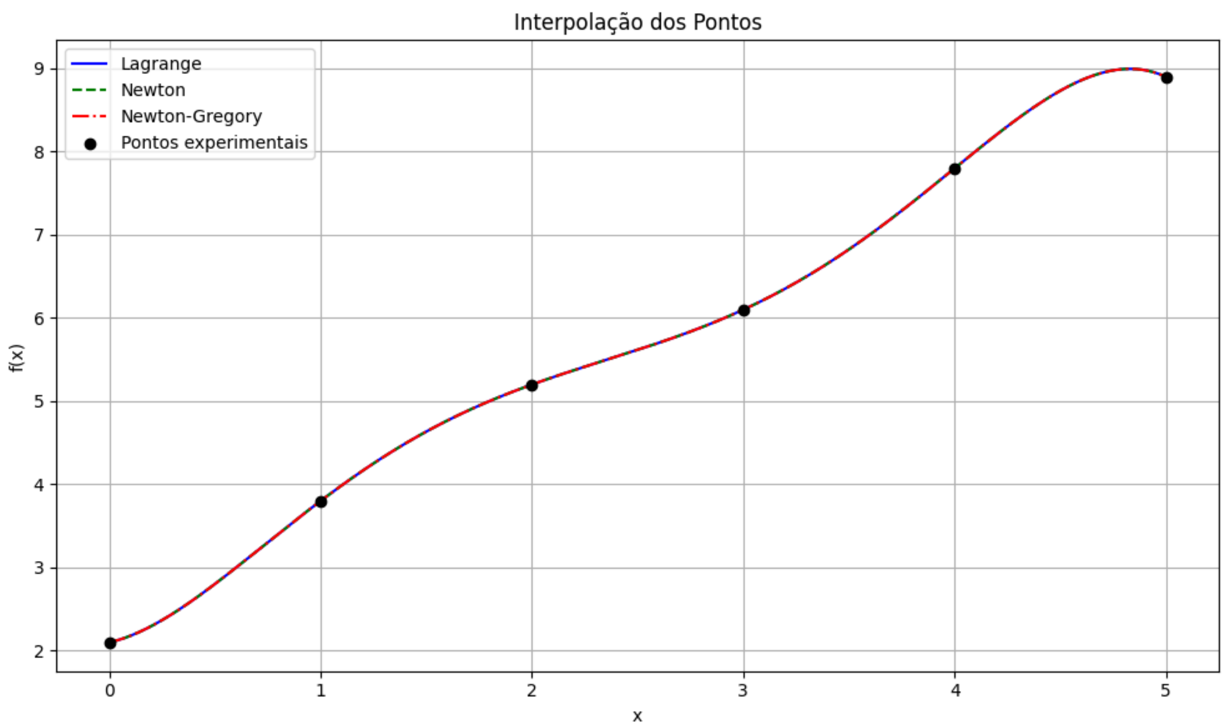

Gráfico de comparação de tempo de execução:

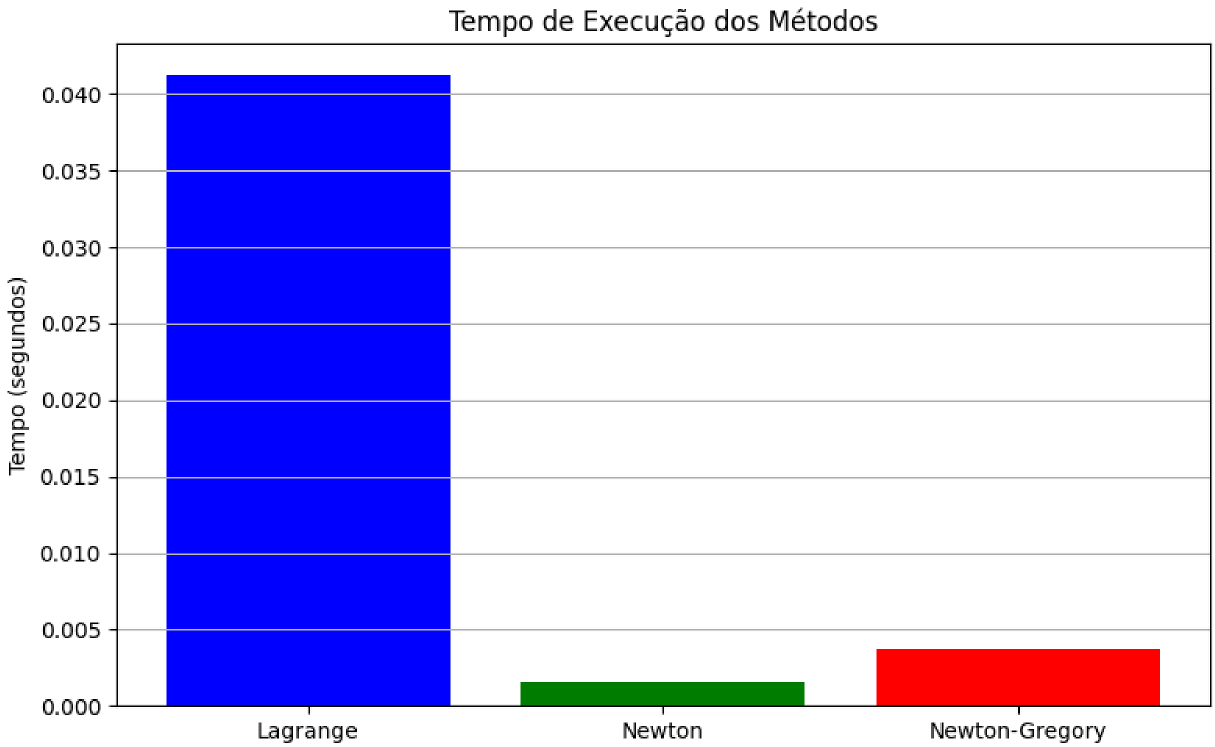

Tabela de comparação entre métodos:

Método	**Tempo de execução(segundos)**	**Valor em** $x = 2.5$
Interpolação de Lagrange	0.0412707	5.61719
Interpolação de Newton	0.00157523	5.61719
Interpolação de Newton-Gregory	0.00369215	5.61719

Tabela de erros entre métodos

Erro entre métodos	Erro
Lagrange e Newton	0
Lagrange e Newton-Gregory	0
Newton e Newton_Gregory	0

Conclusão

Este trabalho prático demonstrou a aplicação de diferentes métodos de interpolação polinomial, com foco na análise da eficiência, precisão e aplicabilidade de cada técnica.

1. Precisão dos Métodos

- Todos os métodos interpolaram com **mesmo valor em x = 2.5 (5.71125)**.
- Os erros entre métodos foram **nulos ou desprezíveis**, o que confirma a equivalência das soluções.

2. Eficiência Computacional

- **Newton-Gregory** foi o mais rápido (0.000159 s), aproveitando o espaçamento uniforme.
- **Newton** foi levemente mais rápido que Lagrange e mais adequado para inserção de novos pontos.
- **Lagrange** é mais custoso para muitos pontos, mas simples e direto.

3. Considerações Práticas

- **Interpolação de Lagrange**: Ideal para pequenos conjuntos, porém pouco eficiente para muitos dados.
- **Interpolação de Newton**: Mais eficiente para adição incremental de dados.
- **Newton-Gregory**: Vantajoso em situações com pontos igualmente espaçados.

Considerações Finais

O trabalho validou os conceitos teóricos estudados, mostrando como diferentes abordagens podem chegar a um mesmo resultado, porém com variações significativas em desempenho e aplicabilidade. A escolha adequada do método depende do contexto: natureza dos dados, necessidade de performance e precisão.