Relatório sobre o trabalho de métodos numéricos. Tema 3

**1. Introdução**.

**Método de Newton**, também conhecido como método de Newton-Raphson, é amplamente utilizado para encontrar raízes de funções não lineares devido à sua rápida convergência. No entanto, suas diferentes implementações podem apresentar variações em desempenho e robustez dependendo das condições numéricas e do comportamento da função.

Neste relatório, analisamos e comparamos três variantes do Método de Newton:

1. **Newton Original**: Utiliza a derivada analítica diretamente.
2. **Newton com Derivada Numérica**: Substitui a derivada analítica por uma aproximação baseada em diferenças finitas.
3. **Newton com Fator de Limitação (FL)**: Inclui um fator de limitação para estabilizar o cálculo em casos adversos.

**Objetivo:**

Avaliar os métodos com base em:

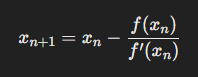
* **Eficiência** (tempo de execução e número de iterações),
* **Precisão** (erro relativo na raiz encontrada),
* **Robustez** (capacidade de evitar falhas numéricas).

**2. Metodologia**

### ****2.1 Métodos Analisados.****

**2.1.1 Newton Original.**

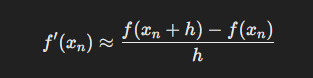
Método clássico cuja fórmula iterativa é:



Onde *f′(xn) ​* é a derivada analítica da função *f(x).* É amplamente utilizado devido à sua simplicidade e eficiência.

**2.1.2 Newton com Derivada Numérica.**

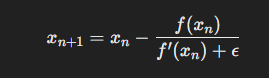
Neste método, f′(xn) é aproximado numericamente por diferenças finitas:



Com h sendo um pequeno incremento. Isso elimina a necessidade de derivadas analíticas, sendo útil para funções complexas.

**2.1.3 Newton com Fator de Limitação (FL).**

Modifica a fórmula tradicional para incluir um fator ϵ que evita oscilações ou instabilidades:



O fator de limitação é especialmente útil em casos onde f′(xn) tende a valores muito pequenos ou grandes.

### ****2.2 Cenário Experimental.****

* **Parâmetros testados**:
  + **λ (lambda)**: Variou entre 0.05 e 0.20,
  + **v e μ**: Ajustes de controle.
* **Métricas Coletadas**:
  + **Raiz encontrada**,
  + **Número de iterações necessárias**,
  + **Tempo de execução (em segundos)**,
  + **Erro relativo**: Diferença entre a solução numérica e a solução exata.
* **Ferramentas**: Implementação em Python para realizar os cálculos, medir o tempo e consolidar os dados.

## **3. Resultados.**

### ****3.1 Análise Comparativa.****

### 

### ****3.2 Resumo Estatístico.****

### 

## **4. Discussão.**

1. **Precisão**: Todos os métodos apresentaram precisão equivalente, com erro relativo na faixa de 10^-8
2. **Número de Iterações**: Não houve variação no número de iterações, indicando convergência consistente para os três métodos.
3. **Tempo de Execução**:
   * O **Newton com Derivada Numérica** foi o mais eficiente (menor tempo médio).
   * O **Newton com FL** foi ligeiramente mais rápido que o Newton Original.
4. **Robustez**:
   * O **Newton com FL** apresentou maior estabilidade, sendo recomendado em cenários com possíveis instabilidades numéricas.
   * O **Newton Original** é confiável, mas pode ser menos robusto em situações adversas.

## **5. Conclusão**

* **Newton com Derivada Numérica**: Mais rápido e indicado para funções onde a derivada analítica não é facilmente calculável.
* **Newton com FL**: Estável e robusto, ideal para casos onde a derivada apresenta comportamentos extremos.
* **Newton Original**: Confiável, mas limitado pela necessidade de derivadas analíticas.