



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
DCA0304 - METODOS COMPUTACIONAIS EM ENGENHARIA

MICHAEL SOARES DE FRANÇA

RELATÓRIO - ATIVIDADE EXTRA

NATAL
2023

Método utilizado para encontrar a raiz

A função fornecida foi:

$$f(x) = x^3 - 10 \quad (1)$$

O método utilizado utiliza 3 equações consecutivas de forma iterativa para aproximar o próximo valor de X, fazendo com que a função convirja para o valor de $f(x) = 0$.

As equações fornecidas foram as seguintes:

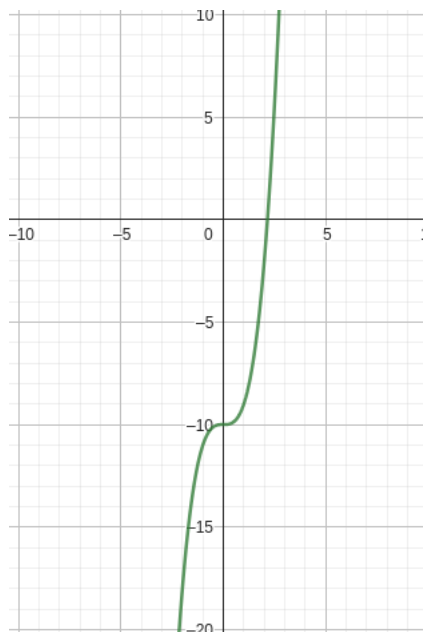
$$y_n = x_n - \frac{2f(x_n)^2}{f(x_n + f(x_n)) - f(x_n - f(x_n))} \quad (2)$$

$$z_n = y_n - \frac{y_n - x_n}{2f(y_n) - f(x_n)} f(y_n) \quad (3)$$

$$x_{n+1} = z_n - \frac{y_n - x_n}{2f(y_n) - f(x_n)} f(z_n) \quad (4)$$

Esse método se apresenta como uma melhoria substancial em relação ao método de Newton e o da secante, tendo em vista uma maior ausência de necessidade da derivada e da secante nos pontos. Por outro lado, analiticamente ele possui um grau de divergência relevante quando o valor inicial fornecido é muito distante da raiz do polinômio.

Figura 1: $f(x) = x^3 - 10$



Observando o comportamento da função, é perceptível que a raiz é tal que:

$$x = \sqrt[3]{10} \cong 2,1544346900319 \quad (5)$$

Para resolução desse problema, foram utilizadas 3 linguagens que são normalmente utilizadas em computação numérica:

1. Python v3.11.4

2. C

3. Julia v1.9.3

Com o objetivo de comparar o grau de dificuldade na escrita do código-fonte e na velocidade de obtenção da resposta. Para as três linguagens, a resposta do valor de x foi 2.1544346900319, padronizando a precisão de 13 casas decimais. Para uma precisão com mais casas decimais, talvez o valor divirja um pouco nas últimas, mas nada substancial para prejudicar a análise.

Comparativo entre linguagens

Para avaliar o tempo de execução do código em Python foi utilizado a biblioteca `time`. Para o código em C, foi utilizada a biblioteca `time.h`. Já para Júlia, foi utilizada uma função interna que executa uma parte do código e verifica o tempo. Após rodar os códigos em cada linguagem, encontrou-se os seguintes resultados:

1. C: 2 μ s

2. Julia: 100 μ s

3. Python: 104 μ s