

## Métodos Numéricos

### Isolamento de Raízes – Método da Dicotomia ou Bissecção

Criar um programa que descobre os zeros de equações através do método da dicotomia ou bissecção, dada uma equação de qualquer grau, um intervalo e um erro. Deve-se utilizar funções para cada parte importante do programa, e as variáveis devem sempre ser locais (passá-las como parâmetro quando necessário).

1. **Receber os dados:** criar um método para receber e armazenar equações de qualquer grau. Embora possam ter qualquer grau, é necessário tratar apenas equações do tipo em que X é a base de cada elemento, por exemplo  $X^3+4X^2-10$ . O usuário deve ser instruído de como informar a equação.
2. **Análise teórica:** verificar se  $f(a).f(b)<0$  para garantir que existe raiz no intervalo recebido (pode-se solicitar um novo intervalo até obter um nessas condições ou encerrar o programa).
3. **Implementação do Método da Dicotomia ou Bissecção:** calcular a raiz da equação. A tabela formatada mostrando todos os cálculos realizados deve ser exibida, seguida da informação do resultado.

#### Importante:

- Evitar repetição de código. Por exemplo, crie uma função que calcula o valor numérico de equações e utilize a mesma função na análise teórica e na implementação do método da dicotomia, apenas enviando para ela os parâmetros pertinentes em cada caso.
- O código fonte deverá estar comentado e conter cabeçalho com RA, nome e descrição do programa.
- A entrega será feita via Canvas: fazer o upload do arquivo .c ou .cpp contendo código fonte com identificação (Nome e RA).

**Dica:** a biblioteca **math.h** contém várias funções que implementam operações matemáticas a serem utilizadas.

## Algoritmo para Implementação em Linguagem C

1. Utilizar o programa disponibilizado no **CANVAS** e realizar as implementações necessárias.
2. Alterar a forma de armazenamento dos fatores multiplicadores do polinômio (variável *multi*) de **vetor** para **ponteiro**.
  - 2.1. **Importante:** Todas as referências devem ser corrigidas.
3. **Receber** as seguintes informações do usuário:
  - 3.1. Grau da função polinomial, que deve estar entre 2 e 6.
  - 3.2. Receber o intervalo [A; B], ou seja, receber o valor de A e o valor de B.
  - 3.3. Receber o valor do erro  $\epsilon$
4. Implementar o Método da Dicotomia
  - 4.1. Verificar se no intervalo [A; B] indicado existe zero de função:  $f(a)*f(b) < 0$

**Se**  $f(a)*f(b) < 0$

Inicializar uma variável **I** (iteração) com o valor igual a 1: **I=1**

Calcular o número de iterações  $K = \frac{\log(b_0 - a_0) - \log(\epsilon)}{\log(2)}$

Enquanto **K**  $\geq$  **I**

Achar o ponto médio:  $m = \frac{a + b}{2}$

**Se**  $f(a)*f(m) < 0$

**b = m**

**Senão**  $//f(m)*f(b) < 0$

**a = m**

Exibir os valores referente a linha da tabela

**I = I + 1**

Fim Enquanto

**Senão**

Exibir uma mensagem indicando que a dicotomia não é válida.

### 5. Dicas:

- 5.1. Utilizar a **função** em Linguagem C chamada **float funcaoX(float ponto)** para calcular os valores da Função  $f(x)$ . A função deve receber como parâmetro o valor do ponto a ser calculado, por exemplo **funcaoX(5)**.
- 5.2. Utilizar a **função** em Linguagem C chamada **void ImprimirTabela (...)** para exibir os valores referentes a cada uma das iterações.