Universidade de Brasília

Relatório de atividade do módulo 1

Métodos Numéricos para Engenharia

# Zeros de Funções

*Aluno:*
Wilton Rodrigues

*Matrícula:*
13/0049212

31 de agosto de 2016

# Sumário

# 1 Introdução

O objetivo deste relatório é exercitar os conceitos aprendidos em aula, com relação ao tópico: Zeros de funções. Que tem como objetivo prover métodos matemáticos capazes de determinar o ponto, ou pontos, nos quais a equação cruza ou toca o eixo X, ou seja, um valor numérico que satisfaça à equação. Tarefa que pode dispender um enorme esforço, ou em alguns casos é até mesmo impossível, em equações que não possuem solução analítica. Como é o caso da Equação de Kepler que é dada por:

$$M = x - Esin(x) \tag{1}$$

Dado que $E = 0.2$ e $M = 0.5$, o objetivo deste trabalho é obter a raíz da equação (1) com precisão de 10 casas decimais.

Fazendo as devidas substituições e manipulações matemáticas, obtemos a seguinte equação:

$$\begin{aligned} -Esin(x) + x - M &= 0 \\ -0.2sin(x) + x - 0.5 &= 0 \end{aligned} \tag{2}$$

A partir da equação (2), utilizaremos dois passos para encontrar a raíz: O passo 1 tem como objetivo obter um intervalo $[a, b]$ aproximado no qual $x \in [a, b]$. O passo 2 é onde aplicaremos o método numérico da Bissecção para refinar a solução.

# 2 Metodologia

Neste primeiro passo, será feita uma análise da equação, baseando-se no gráfico da mesma e nos conceitos teóricos e teoremas do método escolhido. O sucesso ou falha do próximo passo está diretamente ligado aos resultados obtidos nesta fase de análise. Baseando-se no teorema:

**Teorema 1 (Função Contínua)** *Seja f(x) uma função contínua num intervalo [a,b]. Se f(a)f(b) < 0, então existe pelo menos um ponto $x = \alpha$ entre a e b que é zero de f(x).*

E analisando o gráfico da figura 1 é possível perceber que a equação (2) intercepta o eixo X no intervalo $[0, 1]$

Ao substituirmos estes pontos na equação (2), obtemos os seguintes resultados.

$$
\begin{aligned}
a = 0, b = 1 \\
f(a) = -0.2sin(0) + 0 - 0.5 \\
f(a) = -0.5 \\
f(b) = -0.2sin(1) + 1 - 0.5 \\
f(b) = 0.3317058030384207 \\
f(a) * f(b) = -0.16585290151921034
\end{aligned} \tag{3}
$$

Então, de acordo com o teorema 1, como o resultado de $f(a) * f(b) < 0$ de fato há uma raíz entre o intervalo $[0, 1]$.

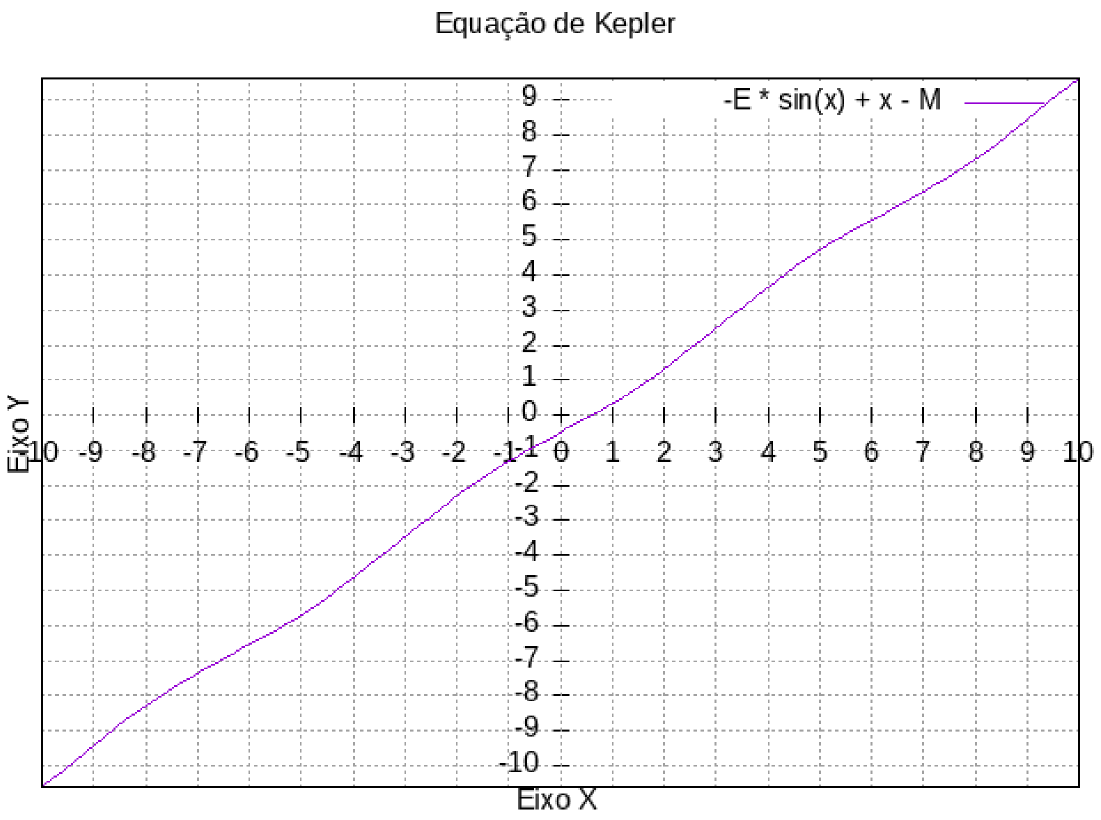


Figura 1: Plotagem da função no intervalo $[-10, 10]$

# 3 Diagrama esquemático de execução

Nesta seção, encontra-se o fluxo de execução da solução da equação (1) utilizando a linguagem C. Que é apresentada na próxima sessão.

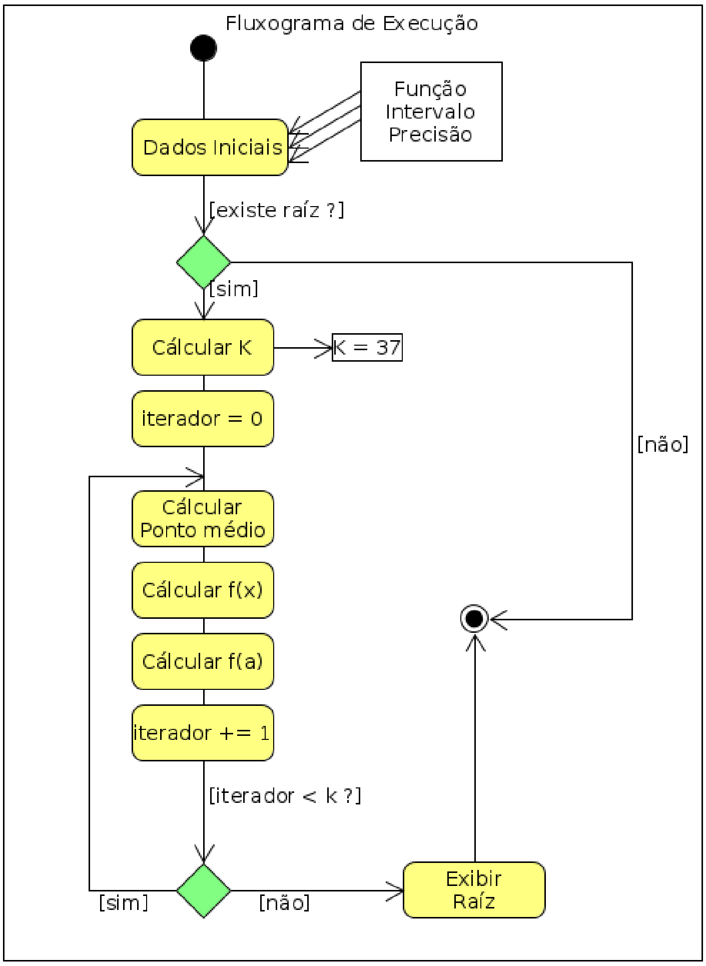


Figura 2: Fluxo de execução da solução

A solução elaborada neste relatório funciona da seguinte maneira. Tanto a função, quanto a precisão desejada são inseridas diretamente no código fonte. Apenas o intervalo no qual se deseja verificar a existencia da raíz é solicitado ao usuário em tempo de execução. Caso o intervalo informado não possua uma raíz, de acordo com teorema 1, uma mensagem é apresentada ao usuário e a execução encerra. Caso o intervalo seja válido a raíz correspondente é apresentada e então o programa se encerra.

# 4 Código Fonte

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#define precision 0.00000000001

int calc_k(double a, double b){
  double k;
  k = (log(b - a) - log(precision)) / log(2);
  return (int)ceil(k);
}

double calc_ponto_medio(double a, double b){
  double x;
  x = (a + b) / 2;
  return x;
}

double calc_fx(double x){
  double fx;
  fx = -0.2 * sin(x) + x - 0.5;
  return fx;
}

double calc_fa(double a){
  double fa;
  fa = -0.2 * sin(a) + a - 0.5;
  return fa;
}

int existence(double a, double b){
  double fa = calc_fx(a);
  double fb = calc_fx(b);
  if (fa * fb < 0)
    return 1;
  else
    return 0;

}

int main(){
  double a, b;
  printf("Digite o primeiro valor do intervalo: \n");
  scanf("%lf", &a);
  printf("Digite o segundo valor do intervalo: \n");
  scanf("%lf", &b);

```

```c
  if(existence(a, b) == 1){
    int k = calc_k(a, b);
    double x = 0.0;
    double fx = 0.0;
    double fa = 0.0;

    for(int i = 0; i < k; i++){
      x = calc_ponto_medio(a, b);
      fx = calc_fx(x);
      fa = calc_fa(a);
      if (fx * fa > 0){
        a = x;
      }
      else{
        b = x;
      }
    }

    printf("%.11lf \n", x);
  }
  else{
    printf("Nao existe raiz neste Ponto\n");
  }
  return 0;
}
```

# 5 Resultados e discussões