Universidade de Brasília - FGA
Aluno: Pedro de Lyra Pereira - 11/0135725
Professor: Rafael Morgado
Disciplina: Métodos Numéricos

# Aplicação do método da bissecção no cálculo da raíz da equação de Kepler

## Introdução

A análise numérica consiste no estudo de métodos que calculem soluções numéricas de problemas que muitas vezes não possuem solução analítica, isto é, não possuem uma solução exata que possa ser escrita em termos de funções conhecidas (e.g.: Bhaskara). O uso de ferramentas computacionais se faz necessário em muitos casos para executar o algoritmo descrito por um método numérico.

O objetivo deste documento é explicar a aplicação de um método numérico na computação da raíz da equação de Kepler. A equação de Kepler dá a relação entre as coordenadas polares de um corpo celeste (como um planeta) e o tempo decorrido a partir de um determinado ponto inicial. Ela é de importância fundamental na mecânica celeste, mas não pode ser explicitamente expressa em termos de funções simples.

Seja *M* a anomalia média (uma parametrização do tempo) e *E* a anomalia excêntrica (uma parametrização do ângulo polar) de um corpo em órbita em uma elipse com excentricidade $e$, temos que:

$M = E - e \cdot sin(E)$ é a **equação de Kepler**

O método escolhido para calcular a raíz desta equação é o método da **bissecção**, que se baseia em uma busca binária sobre um intervalo pré-determinado. O algoritmo proposto pelo método será implementado em linguagem C e o código será disponibilizado no fim deste documento.

## Método da Bissecção

O método da bissecção é um método numérico destinado à busca de raízes de funções que bissecta um intervalo repetidamente e seleciona um dos subintervalos resultantes para continuar a procura pela raíz. Este algoritmo implementa um paradigma conhecido na ciência da computação: **dividir para conquistar**. A proposta é simples: a cada iteração, o espaço de busca é reduzido pela metade e este tipo de pesquisa específica é denominada **busca binária**.

Este método numérico faz uso da seguinte informação: dada uma função contínua $f : [a,b] \rightarrow \mathbb{R},\ y = f(x)$, tendo $f(a)$ e $f(b)$ sinais opostos, o teorema do valor intermediário garante a existência de uma raíz no intervalo $[a, b]$. Em outras palavras, $f(a) \cdot f(b) < 0$ implica a existência de uma raíz no intervalo $[a, b]$.

Portanto, dado um intervalo inicial $[a, b]$ onde as condições descritas acima sejam satisfeitas, o método da bissecção propõe que seja verificado se o ponto médio deste intervalo $c = \frac{a+b}{2}$ é a raíz da função. Caso o ponto médio não seja raíz da função, verifica-se então qual dos subintervalos resultantes da bissecção do intervalo original contém a raíz, para determinar isso, basta verificar qual dos subintervalos satisfaz a condição descrita anteriormente ( $f(a) \cdot f(b) < 0$, lembrando que os subintervalos gerados são $[a, c]$ e $[c, b]$ ).

Este algoritmo é executado até que um dos pontos médios resultantes de uma bissecção produza um erro considerado desprezível dentro de um determinado contexto, ou seja, um dos valores encontrados satisfaz os critérios de precisão estabelecidos. No caso deste trabalho, a raíz calculada possuirá uma precisão de dez casas decimais e, portanto, um valor $x$ que resulte em $-\varepsilon < f(x) < \varepsilon$, onde $\varepsilon = 10^{-10}$ será considerado raíz da equação.

Em resumo, temos que a equação de Kepler é dada por:

$$M = E - e \cdot sin(E) \Rightarrow f(E) = E - e \cdot sin(E) - M$$

E desejamos calcular um valor $E$ que resulte em $-\varepsilon < f(E) < \varepsilon$. Para chegar a este resultado, utilizaremos o método da bissecção descrito anteriormente. Neste trabalho, será considerado que $e = 0.2$ e $M = 0.5$.

Para garantir a obtenção da raíz durante a execução do método, é coerente definir um intervalo inicial quão grande quanto possível (dentro das limitações da capacidade de armazenamento do tipo de dado das variáveis utilizadas, no caso, *double*), pois desta forma, as chances da raíz pertencer a este intervalo serão maiores.

## Análise do algoritmo

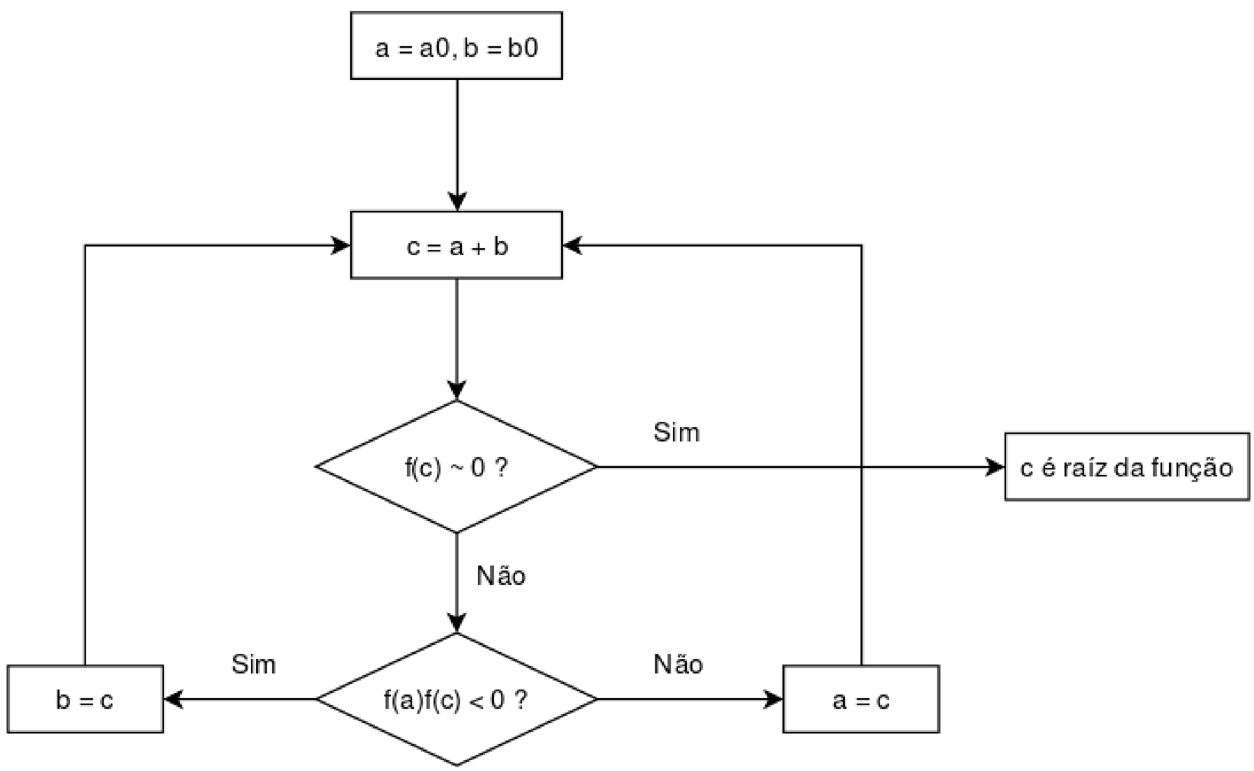
A cada etapa do algoritmo, o erro absoluto é reduzido pela metade. Seja $c_n = \frac{a+b}{2}$ o ponto médio do intervalo da ***n***-ésima iteração, então a diferença entre $c_n$ e uma solução $c$ é limitada por $|c_n - c| \leq \frac{|b-a|}{2^n}$, portanto, se $\varepsilon_n$ for o erro absoluto da ***n***-ésima iteração $\varepsilon_{n+1} \leq \frac{\varepsilon_n}{2}$, o que implica que o método possui uma taxa de convergência linear.

A complexidade assintótica da busca binária é logarítmica, ou seja, $O(log(n))$, onde $n$ é o número de elementos pertencentes ao intervalo inicial (espaço de busca). No caso do método da bissecção, o número de elementos dos intervalos depende da precisão exigida pelo algoritmo (o valor de $\varepsilon$). É sabido que alguns números reais não possuem representação binária no interior de um computador e, portanto, um valor aproximado é usado no lugar destes números. Porém, para efeito de cálculos, suponhamos que todos os números reais com uma precisão de até dez casas decimais possuam uma representação binária.

Temos então que um intervalo de $n$ números inteiros possuirá $\frac{n}{\varepsilon}$ ($\simeq n \cdot 10^{10}$) números reais, que é o tamanho do conjunto de elementos pertencentes ao espaço de busca do problema. Como $\varepsilon$ é um valor constante, a complexidade da busca permanece logarítmica e se torna $O(log(\frac{n}{\varepsilon})) = O(log(n))$. Este resultado é condizente com a execução do algoritmo: a cada iteração o espaço de busca é dividido pela metade e, portanto, até a obtenção do número pesquisado, aproximadamente $log(n)$ iterações serão realizadas.

## Fluxograma do algoritmo

Um fluxograma fornece uma descrição visual das ações e eventos ocorridos durante a execução do algoritmo. Segue abaixo um fluxograma que descreve o método da bissecção:




Na implementação do problema, os valores iniciais de $a$ e $b$ (intervalo de busca inicial) são $-\infty$ e $+\infty$, onde $\infty$ é um valor suficientemente grande definido no início do programa. O fluxograma acima descreve as etapas do algoritmo descrito anteriormente.

## Programa em linguagem C

```
bisection_method.c+
#include <stdio.h>
#include <math.h>

#define INF 1e4         // Infinito
#define EPS 1e-10       // Epsilon (grau de precisão do cálculo)
#define e   0.2         // Excentricidade orbital
#define M   0.5         // Anomalia média

// Equação de Kepler
double kepler(double E) {
        return E - e * sin(E) - M;
}

// Método da bissecção. Dado um intervalo [a, b], verifica-se se o ponto médio
// 'c' deste intervalo é raíz da equação de kepler, ou seja, satisfaz a condição
// de que f(c) é aproximadamente zero (|f(c)| < EPS). Caso o ponto médio não seja
// raíz, verifica-se qual dos dois subintervalos resultantes da bissecção contém
// a raíz (f(a)f(c) < 0 ou f(c)f(b) < 0) e inicia-se o processo novamente em busca
// da raíz sobre o novo intervalo.
double bisection(double a, double b) {
        double c = (a + b) / 2;
        if(fabs(kepler(c)) < EPS) {
                return c;
        }

        if(kepler(a) * kepler(c) < 0) {
                return bisection(a, c);
        } else {
                return bisection(c, b);
        }
}

int main(void) {
        printf("Kepler's equation root: x = %.10lf\n", bisection(-INF, INF));
        return 0;
}
```

A saída exibida pelo programa após sua execução:

```
pedro@kali:~/UnB/numerical_calculus/1st_module$ ./prog
Kepler's equation root: x = 0.6154681694
```

## Conclusão

O resultado obtido é condizente, pois a equação de Kepler tende à zero quando a anomalia excêntrica $E$ se aproxima de $0.6154681694$. A execução do programa leva cerca de $1ms$ para ser concluída e é coerente com a complexidade do algoritmo.