CÁLCULO NUMÉRICO

Jackson F. Magnabosco

Ciência da Computação – Universidade Regional Integrada do Alto Uruguai e das Missões
Erechim (URI)

Caixa Postal 743 – 99709-910 –Erechim – RS – Brasil

jacksonmagnabosco@hotmail.com

Resumo: Em diversos problemas de caráter científico, é comum depararmo-nos com a necessidade de obter uma solução aproximada para encontrar raízes de funções. Nesse ponto, este trabalho objetiva realizar um estudo acerca do método da bisseção para a obtenção de uma solução aproximada das raízes de funções. Com o intuito de ilustrar o funcionamento e aplicação deste método, foram realizados testes numéricos de problemas extraídos da lista de exercícios do portal da universidade, por meio da implementação deste método no ambiente integral de desenvolvimento NetBeans utilizando a linguagem de programação Java.

LISTA DE TABELAS

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. Introdução

Habitualmente, nota-se nos diversos problemas provenientes da ciência e engenharia a necessidade de encontrar soluções para o qual uma função f(x) obtenha valor igual a zero.

Estas soluções são denominadas de raízes da equação ou zeros da função (BARROSO et al., 1987).

A ocorrência de problemas envolvendo essa questão torna importante o estudo de artifícios que permitam encontrar estas raízes, visto que a determinação das mesmas será fundamental no estudo de problemas reais que estejam relacionados à equação f(x) = 0.

O registro mais antigo relacionado à problemática de encontrar uma raiz de uma equação data o ano de 1700 a.C. em uma tábua cuneiforme da Babilônia (BURDEN, 2008).

Este mesmo autor relata que, "... um dos problemas básicos mais importantes da aproximação numérica é o problema de determinação de uma raiz (…) ou solução de uma equação da forma f(x)=0" (BURDEN, 2008, p. 46).

No entanto, sempre irá ocorrer à necessidade de uma análise da função, bem como das técnicas numéricas para aproximações de raízes, tendo em vista que é indispensável o conhecimento de algumas questões que influenciarão na escolha do método mais apropriado para determinada situação, como também na sua eficiência, pois cada técnica de aproximação estabelece condições que irão indicar quando haverá convergência, como será essa convergência e com que rapidez ela acontecerá. Para Sperandio et al. (2003), identificar as circunstância em que o método converge é o primeiro indicativo para escolhê-lo.

Neste trabalho foi pesquisado e desenvolvido um software sobre o Método da Bissecção que é uma técnica iterativa que incide em encurtar a amplitude do intervalo que contém a raiz até que se atinja a exatidão necessária, empregando a divisão sucessiva de [a,b] ao meio.

1.2 Objetivos

O objetivo da realização do trabalho foi o de proporcionar aos acadêmicos situações de contato com desenvolvimento de um software para resolver problemas teóricos utilizando uma implementação do método de bisseção no ambiente integral de desenvolvimento NetBeans na linguagem de programação Java.

1.2.1 Objetivos específicos

Criar funções para encontrar as raízes de uma função qualquer utilizando o método de bissecção.

Implementar um sistema para gerenciar os dados importantes para o método da bisseção, possibilitando a manipulação destes dados através de inserções, alterações e exclusões.

1.3 Justificativa

O conteúdo trata-se de um método simples e robusto, ele é usado frequentemente para obter uma primeira aproximação de uma solução, a qual é então utilizada como ponto inicial para métodos que convergem mais rapidamente.

2 Desenvolvimento

Primeiramente foi definido a técnica iterativa que incide em encurtar a amplitude do intervalo que contém a raiz até que se atinja a exatidão necessária.

Após isto foi escolhida a linguagem de programação para o sistema e o ambiente integral de desenvolvimento.

Com base no algoritmo do método da bissecção pode ser descrito da seguinte forma, como sugere Ruggiero; Lopes (1996).

Tabela 1 – Algoritmo do Método da Bissecção

Algoritmo do Método da Bissecção
Seja a função $f(x)$ contínua no intervalo $[a,b]$ de modo que $f(a) \cdot f(b) < 0$, e que o intervalo $[a,b]$ contenha raiz única.
1) Dados Iniciais:
a) Intervalo Inicial $[a, b]$
b) Precisão ϵ
2) Se $(b-a) < \epsilon$, então escolha para x_0 qualquer $x \in [a,b]$. FIM.
3) $k = 1$
4) $M = f(a)$
5) $x = \frac{a+b}{2}$
6) Se $M = f(x) > 0$, faça $a = x$. Vá para o passo 8.
7) $b = x$
8) Se $(b-a) < \epsilon$, escolha para x_0 qualquer $x \in [a,b]$. FIM.
9) $k = k + 1$. Volte para o passo 5.

Fonte: Adaptado de Ruggiero; Lopes (1996).

Dessa forma definimos as classes e as funções que foram utilizadas para desenvolver o sistema.

Figura 1 – Classes e metodos utilizados

```java
public class CalculoRaiz {
    public static void main(String[] args) {
        Bisseccao.calcularRaiz(0, 1, 0.01);
     //  Bisseccao.calcularRaiz(0.1, 1, 0.01);
    // Bisseccao.calcularRaiz(1, 2, 0.01);
    }
}
```

```java
public class Bisseccao {

    public static double a, b, xn, e, erro, fafnx;
    public static int n;

    public static void calcularRaiz(double a, double b, double erro) {
        Bisseccao.a = a;
        Bisseccao.b = b;
        Bisseccao.erro = erro;
        Bisseccao.n = 0;
        do {

            Bisseccao.xn = Bisseccao.calcularXn(Bisseccao.a, Bisseccao.b);
            Bisseccao.fafnx = Bisseccao.funcao(Bisseccao.a) * Bisseccao.funcao(Bisseccao.xn);
            Bisseccao.e = Bisseccao.calcularErro(Bisseccao.a, Bisseccao.b);
            Bisseccao.mostrarResultados();
            if (Bisseccao.fafnx > 0) {
                Bisseccao.a = Bisseccao.xn;
            } else {
                Bisseccao.b = Bisseccao.xn;
            }
            Bisseccao.n++;
        } while (Bisseccao.e > Bisseccao.erro);

    }
```

```java
public static double funcao(double x) {
    return 3*x - Math.cos(x);
// return x + Math.log10(x);
 // return Math.pow(x, 2) - 2;
}

public static void mostrarFuncao(double x) {
    System.out.println("f(" + x + "): " + Bisseccao.funcao(x));
}

private static double calcularErro(double a, double b) {
    return Math.abs((b - a) / 2);
}

private static double calcularXn(double a, double b) {
    return (a + b) / 2;
}

private static void mostrarResultados() {

    System.out.println("\n");
    System.out.println("n: " + Bisseccao.n);
    System.out.println("a: " + Bisseccao.a);
    System.out.println("b: " + Bisseccao.b);
    System.out.println("xn: " + Bisseccao.xn);
    System.out.println("f(xn):" + Bisseccao.funcao(Bisseccao.xn));
    System.out.println("e: " + Bisseccao.e);
}
```

Após isto foram realizados testes numéricos de três problemas extraídos da lista de exercícios do portal da universidade.

1. Calcular a raiz da função f(x) = x + log(x) , sabendo que a raiz pertence ao intervalo [0,1 ; 1], com erro de 0,01

2. Calcular a raiz da função f(x) = 3x - cos(x) , sabendo que a raiz pertence ao intervalo [0 ; 1], com erro de 0,01

3. Calcular a raiz da função f(x) = x ^ 2 - 2, sabendo que a raiz pertence ao intervalo [1 ; 2], com erro de 0,01

3 Resultados

Questão 1

```
n: 0
a: 0.1
b: 1.0
xn: 0.55
f(xn):0.29036268949424393
e: 0.45
```

```
n: 1
a: 0.1
b: 0.55
xn: 0.325
f(xn):-0.1631166390211256
e: 0.22500000000000003
```

```
n: 2
a: 0.325
b: 0.55
xn: 0.4375
f(xn):0.07847805735833208
e: 0.11250000000000002
```

```
n: 3
a: 0.325
b: 0.4375
xn: 0.38125
f(xn):-0.037540147645157784
e: 0.056249999999999994
```

```
n: 4
a: 0.38125
b: 0.4375
xn: 0.409375
f(xn):0.021496317335858273
e: 0.02812500000000001
```

```
n: 5
a: 0.38125
b: 0.409375
xn: 0.39531249999999996
f(xn):-0.007746952808069363
e: 0.014062500000000006
```

```
n: 6
a: 0.39531249999999996
b: 0.409375
xn: 0.40234375
f(xn):0.00694100939332265
e: 0.007031250000000017
```

Questão 2

```
n: 0
a: 0.0
b: 1.0
xn: 0.5
f(xn):0.6224174381096272
e: 0.5
```

```
n: 1
a: 0.0
b: 0.5
xn: 0.25
f(xn):-0.21891242171064473
e: 0.25
```

```
n: 2
a: 0.25
b: 0.5
xn: 0.375
f(xn):0.1944923780876857
e: 0.125
```

```
n: 3
a: 0.25
b: 0.375
xn: 0.3125
f(xn):-0.01406794804817224
e: 0.0625
```

```
n: 4
a: 0.3125
b: 0.375
xn: 0.34375
f(xn):0.08975253687211893
e: 0.03125
```

```
n: 5
a: 0.3125
b: 0.34375
xn: 0.328125
f(xn):0.03772673911394664
e: 0.015625
```

```
n: 6
a: 0.3125
b: 0.328125
xn: 0.3203125
f(xn):0.011800430315537014
e: 0.0078125
```

Questão 3

```
n: 0
a: 1.0
b: 2.0
xn: 1.5
f(xn):0.25
e: 0.5
```

```
n: 1
a: 1.0
b: 1.5
xn: 1.25
f(xn):-0.4375
e: 0.25
```

```
n: 2
a: 1.25
b: 1.5
xn: 1.375
f(xn):-0.109375
e: 0.125
```

```
n: 3
a: 1.375
b: 1.5
xn: 1.4375
f(xn):0.06640625
e: 0.0625
```

```
n: 4
a: 1.375
b: 1.4375
xn: 1.40625
f(xn):-0.0224609375
e: 0.03125
```

```
n: 5
a: 1.40625
b: 1.4375
xn: 1.421875
f(xn):0.021728515625
e: 0.015625
```

```
n: 6
a: 1.40625
b: 1.421875
xn: 1.4140625
f(xn): -0.0012
e: 0.0078125
```

4 Conclusão

O intuito da realização do trabalho foi o de proporcionar aos acadêmicos situações de contato com a importância do método da bisseção para a obtenção de uma solução aproximada das raízes de funções, de modo que, estes adquiram experiência e conhecimento na área de Cálculo Numérico.

Por intermédio das pesquisas realizadas, e a partir delas foi possível implementar um sistema que atenda a solução.

Para efetuar a modelagem foram utilizadas ferramentas eficientes que forneceram uma interação fácil do usuário com o sistema como NetBeans.

No processo de desenvolvimento deste trabalho ocorreu um grande enriquecimento em relação à modelagem orientada a objetos e programação matemática, de um modo geral a realização deste trabalho, permitiu ampliar conhecimentos adquiridos durante o decorrer do semestre.

O fato de desenvolver um trabalho é sempre empolgante assim aproveitamos toda a criatividade e inteligência, rendendo um comprometimento extraordinário.

Com certeza que, as aulas feitas antes e depois do trabalho, foram essenciais para a evolução dos alunos e compreensão da disciplina.

Por isso, acredito que, o aproveitamento das aulas foi total devido à um grande conjunto de fatores, os quais ocorreram durante o semestre inteiro.

5 Referência

ARENALES, Selma; DAREZZO, Arthur. Cálculo Numérico: Aprendizagem com Apoio de Software. 2 ed. São Paulo: Thompson Learning, 2008

BARROSO, Leônidas Conceição et al.Cálculo Numérico . São Paulo: Harbra Ltda, 1987. Boyer, Carl B. História da Matemática. São Paulo. Edgard Blucher, 1974.

BURDEN, Richard L. E Faires, Douglas. Análise Numérica. São Paulo: CENGAGE 2008.

CAMPOS FILHO, Frederico Ferreira. Algoritmos Numéricos. 2.ed. Rio de Janeiro: LTC, 2010.

FLEMMING, Diva Marília; GONÇALVES, Mirian Buss. Cálculo A: funções, limite, derivação e integração. 6. ed. rev. e ampl. São Paulo: Pearson Prentice Hall, 2009.

FRANCO, Neide Bertold. Cálculo Numérico. São Paulo: Prentice Hall, 2006.