

Universidade Federal da Fronteira Sul – UFFS

Curso: Ciência da Computação (2ª fase)

Disciplina: Cálculo 1

Professor: Milton Kist

Discentes: Daniele Karoline C.

TRABALHO APLICADO 1

Problema (primeira parte): Use o Teorema do Valor Intermediário para mostrar que cada equação abaixo possui pelo menos uma solução, indicando um intervalo onde tenha cada equação tenha solução. Justifique sua afirmação.

Escolhemos a equação $e^{-x} = 2x^2 - 4$.

$$e^{-x} = 2x^2 - 4$$

$$e^{-x} - 2x^2 + 4 = 0$$

$$\underline{f(x) = e^{-x} - 2x^2 + 4 = 0}$$

$$f(0) = e^0 - 2 \cdot 0^2 + 4$$

$$f(0) = 1 - 0 + 4$$

$$\underline{f(0) = 5}$$

$$f(4) = e^{-4} - 2 \cdot 4^2 + 4$$

$$f(4) = 0.018 - 32 + 4$$

$$\underline{f(4) = -27,98}$$

Pode-se concluir que, pelo TVI, existe um $c \in [0, 4]$ de modo que $f(c) = 0$. Portanto, para $e^{-x} = 2x^2 - 4$ existe solução real sendo f continua em $[0, 4]$, $f(0) > 0$ e $f(4) < 0$.

Problema (segunda parte): Nesta parte os estudantes deverão fazer uma implementação. Deverão criar um algoritmo que aproxime a solução da referida equação (escolhida para fazer a primeira parte) com intervalo de comprimento igual ou inferior a $1/10$, que contenha uma solução, isto é, o erro de aproximação da referida solução será menor que 10^{-1} . Quando o erro for o desejado, deverá ser indicado o comando de parada. Dados de entrada: Dois números reais Dados de saída: O algoritmo deve retornar com a seguinte mensagem:

(a) “não é possível afirmar que existe solução neste intervalo, tente outros dois números”;

(b) Ou “a equação tem, pelo menos, uma solução neste intervalo” e também deve retornar um intervalo de comprimento menor que 10^{-1} .

→ O trabalho foi desenvolvido na linguagem C, utilizou-se as propriedades de funções (structs) e bibliotecas.

→ // indicam comentários no código, não aparecem durante a execução.

→ Para facilitar, é possível utilizar um compilador online para poder executar o código: https://www.onlinegdb.com/online_c_compiler

→ Basta clicar no link, colar o código na área escura (clica na área, CTRL+A , CTRL+V). Após, clique em “run” para executar.

CÓDIGO:

```
#include <stdio.h>
```

```
#include <stdbool.h>
```

```
#include <math.h>
```

```
//Importamos as bibliotecas necessárias para a realização dos calculos e  
verificação
```

```
int main(){
```

```
//declaração das variaveis que iremos usar
```

```
float x, y, e = 2.718282, k, dist, centro, t;
```

```
//input para pedir os valores do intervalo
```

```
printf("Digite o primeiro valor do intervalo:\n");
```

```
scanf("%f", &x);
```

```
printf("Digite o segundo valor do intervalo:\n");
```

```
scanf("%f", &y);
```

```
if(x > y){
```

```
    t = x;
```

```
    x = y;
```

```
    y = t;
```

```
}
```

```
//função para aplicar os valores na fórmula e obter o valor intermediario
```

```
float valori(float j){
```

```
    k = pow(e, -j) - 2 * pow(j, 2) + 4;
```

```
    return k;
```

```
}
```

//função que verifica se há raízes. caso os valores sejam menores que 0 e opostos, então há um momento em que o eixo x é cortado

```
bool oposicao(float a, float b) {  
    if(a <= 0 && b >= 0){return true;}  
    if(a >= 0 && b <= 0){return true;}  
    else {return false;}  
}
```

//verifica se há oposição entre o vi de x, y e valor do centro. calcula a distancia entre eles

```
if(oposicao(valori(x), valori(y))){  
    dist = y - x;  
    while (dist > 0.1) {  
        centro = x + (dist/2);  
        if (oposicao(valori(x), valori(centro))) {  
            y = centro;  
        }  
        else if (oposicao(valori(centro), valori(y))) {  
            x = centro;  
        }  
    }  
    dist = y - x;  
}
```

//printa a mensagem ao usuario

```
printf("A equacao tem pelo menos uma solucao neste intervalo: [%f , %f]\n", x, y);  
}
```

```
else {  
    printf("Nao e possivel afirmar que existe solucao neste intervalo, tente outros dois  
    numeros.\n");  
}  
return 0;  
}
```