



UNIVERSIDADE DE BRASÍLIA

RELATÓRIO DE ATIVIDADE DO MÓDULO 5

MÉTODOS NUMÉRICOS PARA ENGENHARIA

Integração Numérica

Aluno:
Wilton Rodrigues

Matrícula:
13/0049212

20 de novembro de 2016

1 Introdução

O objetivo deste relatório é exercitar os conceitos aprendidos em aula, com relação ao tópico: Integração Numérica.

Em determinadas situações, integrais são difíceis, ou mesmo impossíveis de se resolver analiticamente. Para contornar esse obstáculo é possível, através da substituição da função $f(x)$ por um polinômio que a aproxime razoavelmente num dado intervalo $[a, b]$, *encontrar o valor da integral*.

O problema a ser solucionado é o que trata da função seno integral, como é mostrado a seguir:

$$Si(x) = \int_0^x \frac{\sin(t)}{t} dt \quad (1)$$

Onde o objetivo deste relatório será, através do método da regra do trapézio composta, plotar o gráfico da equação (1) com 100 valores de x no intervalo $0 < x < 10$, além de calcular os valores de $Si(0, 5)$ e $Si(5, 5)$ com precisão de 6 casas decimais.

2 Metodologia

Baseando-se no método da Regra do Trapézio composta, como dito anteriormente, tem-se a seguinte relação:

$$\int_{x_0}^{x_n} f(x) dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x) dx = \sum_{i=0}^{n-1} \frac{h}{2} [f(x_i) + f(x_{i+1})] \quad (2)$$

Onde podemos escrever a equação (2) como:

$$\int_{x_0}^{x_n} f(x) dx = \frac{h}{2} [f(x_0) + f(x_1) + \dots + f(x_{n-1}) + f(x_n)] \quad (3)$$

Como o problema especifica uma quantidade mínima de pontos a serem apresentados, a qual vamos chamar de N , o nosso h será calculado de acordo com a seguinte relação:

$$h = (x_n - x_0)/N \quad (4)$$

3 Diagrama esquemático de execução

Nesta seção, encontra-se o fluxo de execução utilizando a linguagem C. Que é apresentada na próxima sessão.

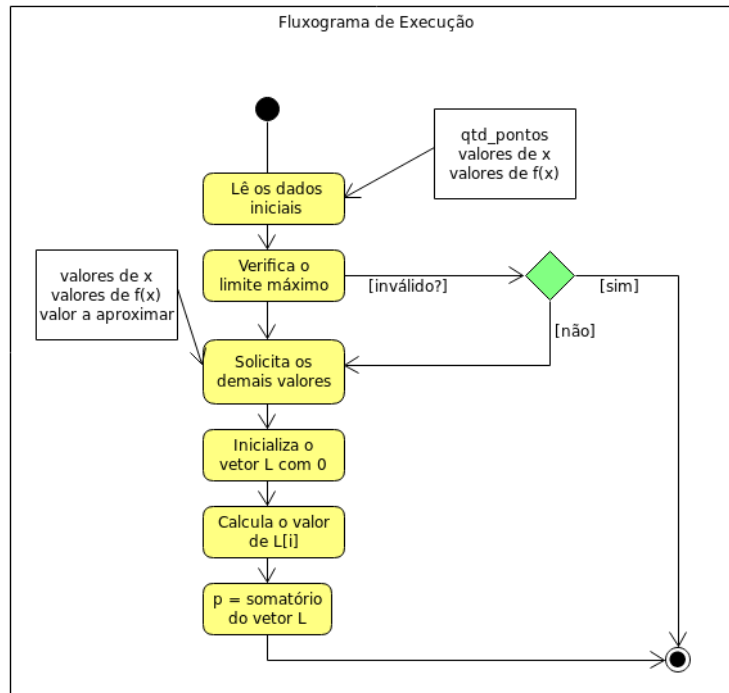


Figura 1: Fluxo de execução da solução

A solução elaborada neste relatório funciona da seguinte maneira. É necessário inserir a quantidade de pontos conhecidos. A quantidade máxima permitida é de 10 elementos, ou seja, funções de grau máximo 9. Após isso o programa solicitará a inserção do valor de cada ponto e em seguida o valor da função. Por fim, também é solicitado o valor de x que deseja aproximar. Então, o programa inicia o vetor L com zero em todas as suas posições e em seguida entra em um loop calculando o somatório do método de legrange, baseado no princípio de que os elementos i e j não podem ser iguais, para que haja o cancelamento de todos os elementos. O resultado da divisão é então alocado na respectiva célula do vetor L . Por fim, há um loop para somar todos os valores de lagrange encontrados na variável p , que é então exibida ao usuário, estando preenchida com o valor em função do ponto x que o usuário inseriu.

4 Código Fonte

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <assert.h>
4 #include <math.h>
5 #define QTD_POINTS 100
6
7 int main(int argc, char *argv[]) {
8
9     double upper_limit, below_limit = 0.000000000001;;
10    double x[QTD_POINTS], f[QTD_POINTS];
11    double I;
12    char file_name[] = "in.dat";
13    FILE *file;
14
15    sscanf(argv[1], "%lf", &upper_limit);
16    double h = upper_limit / QTD_POINTS;
17
18    file = fopen(file_name, "w");
19    if(file == NULL)
20        printf("Erro, nao foi possivel abrir o arquivo\n");
21    else{
22        for(int i = 1; i <= QTD_POINTS + 1; i++){
23            x[i] = below_limit;
24            below_limit += h;
25            f[i] = sin(x[i])/x[i];
26            fprintf(file, "%lf %lf\n", x[i], f[i]);
27        }
28        fclose(file);
29    }
30
31    double even_sum = 0.0;
32    for(int i = 2; i < QTD_POINTS + 1; i += 2){
33        even_sum += f[i];
34    }
35
36    double odd_sum = 0.0;
37    for(int i = 3; i < QTD_POINTS + 1; i += 2){
38        odd_sum += f[i];
39    }
40
41    I = (h/2)*(f[1] + (2*(odd_sum + even_sum) + f[QTD_POINTS + 1]));
42    printf("I=%lf\n", I);
43    return EXIT_SUCCESS;
44 }
```

5 Resultados e discussões

Nesta seção discutiremos os resultados obtidos após a execução do programa apresentado na seção anterior.

Primeiramente, será apresentada a plotagem do gráfico da equação (1) para o intervalo $[0, 10]$, como havia sido estabelecido no início do relatório. Para gerar estes pontos, basta executar o programa com o limite superior = 10. Um arquivo será gerado com todos os pontos e os respectivos valores. Então, através da ferramenta Gnuplot, foi feita a plotagem do gráfico.

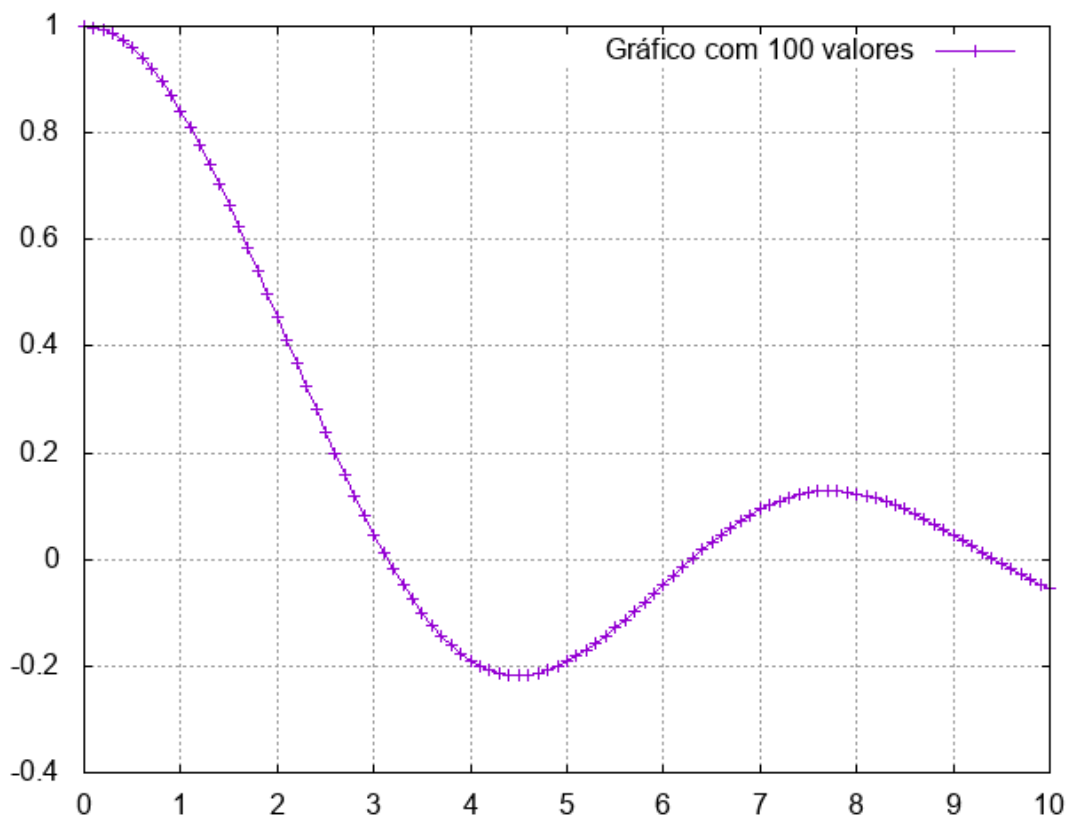


Figura 2: Plotagem de 100 valores no intervalo $[0, 10]$

Após isso, executou-se o programa para os valores $Si(0, 5)$ e $Si(5, 5)$:

```
[0][wilton@asus]~/Dropbox/UnB/Métodos Numéricos/solutions/m5/solution
$ ./a.out 0.5
I=0.493107
```

Figura 3: Resultado da execução do programa para $Si(0, 5)$

```
[0][wilton@asus]~/Dropbox/UnB/Métodos Numéricos/solutions/m5/solution
$ ./a.out 5.5
I=1.468762
```

Figura 4: Resultado da execução do programa para $Si(5, 5)$

Após a execução do programa foram obtidos os seguintes valores:

$$Si(0, 5) = 0.493107 \text{ e } Si(5, 5) = 1.468762$$

Devido ao fato de o método utilizado ter sido a Regra do Trapézio Composto, para não cairmos em uma indeterminação ao aplicarmos o método ao primeiro elemento que resultaria em uma divisão por 0, substituiu-se este valor por 1×10^{-8} .

O resultado encontrado a partir da solução proposta é condizente. Pois ao fazermos a comparação dos valores obtidos com os resultados obtidos no site WolframAlpha (<https://www.wolframalpha.com>), chegamos ao mesmo resultado:

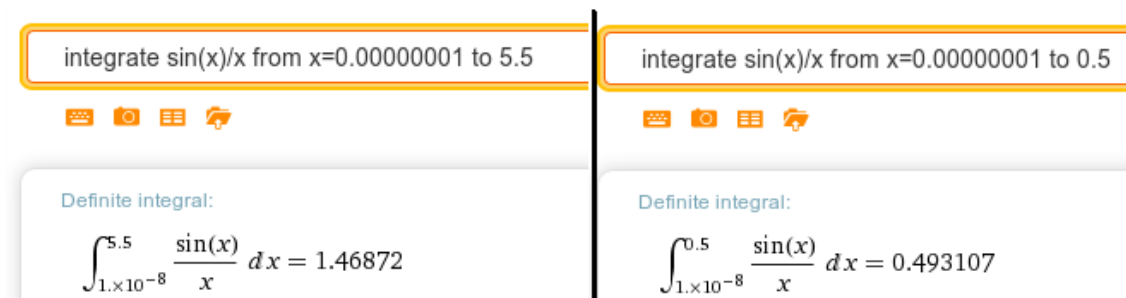


Figura 5: Resultados do WolframAlpha

Sendo assim, o objetivo proposto no início do relatório foi satisfatoriamente alcançado.

6 Ferramentas

Todas as ferramentas utilizadas neste relatório são ferramentas open source (software livre). Permitindo assim que qualquer um possa reproduzir e contestar as afirmações presentes neste documento.

1. Arch Linux (<https://www.archlinux.org>)
 - Sistema operacional utilizado.
2. GCC (<https://gcc.gnu.org>)
 - Compilador de C utilizado para compilar a solução.
3. Python (<https://www.python.org>)
 - Linguagem de programação utilizada para conferir os valores da solução.
4. vim (<http://www.vim.org>)
 - Editor de texto.
5. L^AT_EX (<https://www.latex-project.org>)
 - Sistema tipográfico de alta qualidade (utilizado para elaborar o relatório).
6. Gnuplot (<http://www.gnuplot.info>)
 - Utilitário de representação gráfica (utilizado para plotagem do gráfico).
7. UMLet (<http://www.umlet.com>)
 - Ferramenta de UML (utilizado para criar o fluxo de execução).
8. Shutter (<http://shutter-project.org>)
 - Programa de captura de tela (utilizado para capturar os resultados).