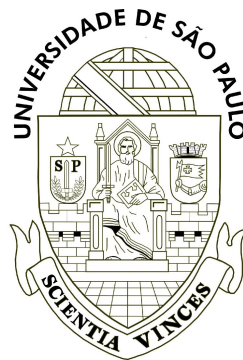


Universidade de São Paulo
Instituto de Matemática e Estatística

MAC0210 - Laboratório de Métodos Numéricos
2023



Exercício-programa 3: Integração Numérica

Beatriz Viana Costa

Conteúdo

1. Parte 1 - Computando trabalho	3
1.1. Decisões de implementação	3
1.2. Resultados	4
2. Parte 2 - Integração por Monte Carlo	5
2.1. Integrais unidimensionais	5
2.2. Integrais multidimensionais	5
2.3. Resultados dos testes	6

1. Parte 1 - Computando trabalho

1.1. Decisões de implementação.

Para a implementação deste método foi utilizada a interpolação de *Lagrange* dos pontos dados. Sabemos que um polinômio de grau n de *Lagrange* pode ser escrito da seguinte forma.

$$p_n(x) = \sum_{j=0}^n f(x_j) \times L_j(x)$$
$$L_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^n \frac{(x - x_i)}{(x_j - x_i)}$$

Onde denotamos $f(x) = F(x)\cos(\theta(x))$.

Para isso foi escrita a função *evalP7x(long double x)* que utiliza $\psi(x)$ e também w_j para a construção do polinômio de *Lagrange* de grau 7 a partir dos pontos fornecidos, a função utiliza as seguintes fórmulas para os cálculos, para $n = 7$:

$$\psi(x) = \prod_{i=0}^{n-1} (x - x_i)$$
$$w_j = \prod_{\substack{i=0 \\ i \neq j}}^{n-1} \frac{1}{(x_j - x_i)}$$
$$p_7(x) = \psi(x) \times \sum_{j=0}^{n-1} \frac{w_j \times f(x_j)}{(x - x_j)}$$

Os pontos dados no enunciado foram guardados em vetores globais no programa, dessa forma podem ser acessados pelas funções a partir de seu índice i , $0 \leq i \leq 6$, totalizando assim 7 pontos.

Já para a valoração de cada um dos métodos de integração compostos, foram utilizadas as seguintes relações. Para a regra integração numérica do trapézio composto:

$$\int_a^b f(x) dx \approx \frac{h}{2} \left[f(a) + 2 \sum_{i=1}^{r-1} f(a + ih) + f(b) \right]$$

E para a regra de integração numérica de Simpson composta:

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[f(a) + 2 \sum_{k=1}^{r/2-1} f(t_{2k}) + 4 \sum_{k=1}^{r/2} (t_{2k-1}) + f(b) \right]$$

Onde $a = 0.00$, $b = 30.00$, r a quantidade de subintervalos, que é um inteiro escolhido pelo usuário, e h o espaçamento entre cada um dos pontos, $h = \frac{b-a}{r}$. Uma ressalva é que o método de Simpson composto deve ser aplicado para uma quantidade par de subintervalos, dessa forma se o usuário fornecer uma quantidade r ímpar de subintervalos, a função de Simpson arredondará esse valor para o próximo número par superior.

Para a compilação do programa é necessário digitar os seguintes comandos no terminal:

```
$ gcc -ansi -Wall -pedantic parte1.c
```

O que irá chegar um executável chamado *a.out* Para executá-lo basta digitar:

```
./a.out
```

O programa pedirá a quantidade r de intervalos que se deseja usar na integração numérica e após isso mostrará os resultados tanto da integração do trapézio composto quanto de Simpson composto.

1.2. Resultados.

O programa foi executado para diferentes valores de r para os testes. Segue tabela com os resultados:

Regra do trapézio composto	
Quant. r	Área calculada
1	5.305500
2	133.190250
5	119.958702
10	117.832290
15	117.442341
20	101.603581
25	109.496835
30	117.209191
35	117.188604
40	117.175245
45	117.166087
50	117.159538

Regra de Simpson composto	
Quant. r	Área calculada
1	32.690981
2	32.690981
5	134.443337
10	122.226986
15	79.670421
20	101.875761
25	110.510760
30	117.131474
35	117.062107
40	117.048387
45	117.050146
50	117.058983

É possível notar que para certos valores de r a área calculada pelos dois métodos se aproximam entre si, mais especificamente para $r \geq 20$. Para os valores de r anteriores os resultados variam, mostrando que quanto menor os intervalos, melhor a aproximação.

Além disso, a partir de $r = 30$, os resultados apresentados são muito parecidos, havendo diferença apenas nas casas decimais. Isto é, quando o espaçamento h entre cada ponto utilizado nos cálculos da integral é menor do que 1, nossa precisão nestes cálculos aumenta.

2. Parte 2 - Integração por Monte Carlo

Para a integração de Monte Carlo é uma integração numérica utilizando a valoração de números randômicos para a aproximação de integrais definidas.

Para cada função dada no enunciado foi encontrada uma maneira diferente de encontrar uma aproximação adequada, sendo que todas seguem a ideia de tirar a média de n variáveis aleatórias independentes e uniformemente distribuídas, com n tendendo a um número tão grande quanto se queira (e que seja representável por um *long unsigned int*).

Para a compilação do programa é necessário digitar os seguintes comando no terminal:

```
$ gcc -ansi -Wall -pedantic parte2.c
$ ./a.out
```

O programa irá solicitar um número n que será utilizado para o cálculo da média para as 4 aproximações pedidas no enunciado, e após isso irá mostrar na tela, a aproximação de $\int_0^1 \sin(x) dx$, $\int_3^7 x^3 dx$, $\int_0^\infty e^{-x} dx$ e uma aproximação para π , respectivamente.

2.1. Integrais unidimensionais.

(1) $\int_0^1 \sin(x) dx$:

Sabemos por meio do cálculo analítico dessa integral que $\int_0^1 \sin(x) = -\cos(x) \Big|_0^1 = -\cos(1) + \cos(0) \approx 0.459698$.

Para a realização da integração numérica foi utilizada a relação de integração unidimensional apresentada no enunciado, onde n variáveis aleatórias ξ_i tem seus valores atribuídos aleatoriamente, sendo que $\xi_i \in [0; 1]$.

A função seno foi valorada em $\sin(a + \xi \times (b - a)) = \sin(0 + \xi \times (1 - 0)) = \sin(\xi)$, $\xi \in [0; 1]$. Posteriormente é retirada a média por n .

(2) $\int_3^7 x^3 dx = \frac{x^4}{4} \Big|_3^7 = 580$

De maneira similar ao anterior, a função dada foi valorada em $a + \xi_i \times (b - a)$, onde $\xi_i \in [0; 1]$, ξ_i é uma variável aleatória uniformemente distribuída, $0 \leq i \leq n$, e depois é retirada a média destas n variáveis.

(3) $\int_0^\infty e^{-x} dx = -e^{-x} \Big|_0^\infty = 1$

A integração numérica foi feita de maneira similar às duas anteriores, contudo, foi necessária a realização de uma troca de variáveis, uma vez que a integral está sendo também analisada no infinito, um valor que não é computável.

Escrevendo a mesma integral observada em valores calculáveis, encontramos que $\int_0^\infty e^{-x} dx = \int_0^1 e^{-x} + e^{-1} dx$. Como feitos nos demais testes, $x = \xi_i$, $0 \leq i \leq n$ e $x_i \in [0; 1]$ e é retirada a média entre as n variáveis aleatórias.

2.2. Integrais multidimensionais.

(1) Aproximação de π

Para esta aproximação foi utilizada a integral multidimensional, uma vez que para aproximar π dependemos de dois eixos, x e y .

O procedimento foi similar aos anteriores, n variáveis aleatórias x_i e y_i tiveram seus valores atribuídos de forma aleatória, tal que $x_i, y_i \in [0; 1]$. Caso $x^2 + y^2 \leq 1$, ou seja, estamos

verificando se o ponto encontrado pertence ao pedaço da circunferência de raio 1 que está no primeiro quadrante, incrementamos a variável que guarda a aproximação de π .

Ao final, tiramos a média, como feito nos outros casos, e também multiplicamos por 4, uma vez, que segundo o próprio enunciado, estamos verificando se o ponto pertence ao primeiro quadrante, é necessário contar se pertence também aos outros três quadrantes.

2.3. Resultados dos testes.

Foram realizados testes para $n \geq 10^5$ e as aproximações se mostraram sólidas, com o erro absoluto diminuindo à medida que n cresce.

Por Monte Carlo se tratar de um método que usa variáveis aleatórias (no caso do programa escrito, pseudoaleatórias), o mesmo valor de n pode resultar em duas aproximações diferentes ao executar o programa mais de uma vez.