Relatório do EP3 - Integração Numérica

Leonardo Heidi Almeida Murakami / Número USP: 11260186

18 de junho de 2025

1 Introdução

Este relatório detalha a implementação e os resultados do EP3. O objetivo principal deste EP foi explorar diferentes métodos de integração numérica, incluindo a Regra do Trapézio Composta, a Regra de Simpson Composta e o Método de Monte Carlo (unidimensional e multidimensional). A primeira parte do exercício focou no cálculo de trabalho utilizando interpolação e regras compostas através de dados fornecidos, enquanto a segunda parte abordou a integração de funções diversas e a aproximação de π usando Monte Carlo.

2 Parte 1: Computando Trabalho

2.1 Fundamentação Teórica

O trabalho realizado por uma força variável $F(x)$ ao longo de um deslocamento de x_0 a x_n, considerando um ângulo $\theta(x)$ entre a força e a direção do movimento que também varia com a posição, é dado pela integral:

$$\tau = \int_{x_0}^{x_n} F(x)cos(\theta(x))dx \quad (3)$$

Em situações reais, $F(x)$ pode ser difícil de integrar analiticamente, ou a força pode estar disponível apenas na forma de dados tabulados. Nesses casos, a integração numérica se torna a única opção viável.

Logo, o processo foi dividido em duas etapas:

1. Interpolar a função $F(x)cos(\theta(x))$ usando os dados fornecidos na Tabela 1. A interpolação de Lagrange ou Newton foi sugerida. A implementação utilizou a interpolação de Lagrange.
2. Aproximar a integral (3) utilizando as regras do trapézio composto e de Simpson composto.

2.2 Decisões de Projeto e Implementação (Parte 1)

2.2.1 Interpolação

A interpolação de Lagrange foi escolhida para estimar os valores de $F(x)cos(\theta(x))$ em pontos não presentes na Tabela 1. A função 'lagrange_interp' foi implementada para este propósito, recebendo o ponto alvo x_{target}, os arrays de dados x e y (onde $y = F(x)cos(\theta(x))$) e o número de pontos. Uma função 'g_interpd' foi criada como um "wrapper"para a função interpolada, facilitando sua passagem para as rotinas de integração numérica.

2.2.2 Regra do Trapézio Composta

A Regra do Trapézio Composta aproxima a integral dividindo o intervalo de integração em n subintervalos e aplicando a regra do trapézio em cada um. A fórmula geral é:

$$\int_a^b f(x)dx \approx \frac{h}{2}[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)]$$

onde $h = (b-a)/n$. A função 'trapezoidal_rule_interp' implementa esta regra, utilizando a função interpolada 'g_interpd'.

2.2.3 Regra de Simpson Composta

A Regra de Simpson Composta oferece uma aproximação mais precisa, utilizando polinômios de segundo grau. Requer um número par de subintervalos. A fórmula geral é:

$$\int_a^b f(x)dx \approx \frac{h}{3}[f(x_0) + 4\sum_{i=1,3,\ldots}^{n-1} f(x_i) + 2\sum_{i=2,4,\ldots}^{n-2} f(x_i) + f(x_n)]$$

onde $h = (b-a)/n$. A função 'simpson_rule_interp' foi implementada, incluindo uma verificação para garantir que o número de intervalos seja par.

2.3 Resultados (Parte 1)

Os cálculos foram realizados no intervalo de $x = 0.0$ a $x = 30.0$ (os limites da Tabela 1). A função interpolada 'g_interpd' foi utilizada.

2.3.1 Regra do Trapézio Composta

```
--- Resultados da Regra do Trapezio Composta ---
  Intervalos | Trabalho (J)
  -----------|--------------
           6  |      119.0893
          66  |      117.1476
         126  |      117.1360
         186  |      117.1336
         246  |      117.1328
         306  |      117.1324
         366  |      117.1321
         426  |      117.1320
         486  |      117.1319
         546  |      117.1319
```

Com o aumento do número de intervalos, a aproximação do trabalho converge para aproximadamente 117.1319 J.

2.3.2 Regra de Simpson Composta

```
--- Resultados da Regra de Simpson Composta ---
  Intervalos | Trabalho (J)
  -----------|--------------
           6  |      117.1272
          12  |      117.1271
          18  |      117.1306
          24  |      117.1313
          30  |      117.1315
          36  |      117.1315
          42  |      117.1316
          48  |      117.1316
          54  |      117.1316
          60  |      117.1316
```

A Regra de Simpson também converge rapidamente para um valor próximo de 117.1316 J. Observa-se que a Regra de Simpson tende a convergir mais rapidamente e para um valor ligeiramente diferente da Regra do Trapézio, mas bem próximo.

3 Parte 2: Integração por Monte Carlo

3.1 Fundamentação Teórica

O método de Monte Carlo é uma técnica numérica que utiliza amostragem aleatória para estimar integrais. Para uma integral unidimensional $I = \int_a^b g(x)dx$, se U é uma variável aleatória uniformemente

distribuída em $[0,1]$, então $\mathbb{E}(g(U)) = \int_0^1 g(x)dx$. Pela Lei Forte dos Grandes Números, a média das amostras de $g(U_i)$ converge para o valor esperado:

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} g(U_i) \to \mathbb{E}[g(U)] = I \quad (8)$$

Para integrais no formato $\int_a^b g(x)dx$, uma troca de variável é necessária para mapear o intervalo $[a,b]$ para $[0,1]$. A transformação $x = a + (b-a)u$ (onde $u \in [0,1]$) resulta em $dx = (b-a)du$. Assim, $\int_a^b g(x)dx = (b-a)\int_0^1 g(a+(b-a)u)du$.

Para integrais multidimensionais, como $I = \int_0^1 \cdots \int_0^1 g(x_1, ..., x_d)dx_1...dx_d$, a ideia é análoga:

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} g(U_1^i, U_2^i, ..., U_d^i) \quad (12)$$

onde $(U_1^i, ..., U_d^i)$ são n conjuntos independentes de d variáveis aleatórias uniformemente distribuídas em $[0,1]$.

3.2 Decisões de Projeto e Implementação (Parte 2)

3.2.1 Geração de Números Aleatórios

A função ‘random_uniform‘ foi implementada para gerar números pseudoaleatórios no intervalo $[0,1]$ usando ‘rand()‘ e ‘RAND_MAX‘ da biblioteca ‘stdlib.h‘. O gerador foi semeado com ‘srand(time(NULL))‘ na função ‘main‘ para garantir diferentes sequências de números aleatórios a cada execução.

3.2.2 Integração Monte Carlo 1D

A função ‘monte_carlo_1d‘ implementa o método para integrais unidimensionais. Recebe a função a ser integrada $g(x)$, os limites inferior e superior (a e b) e o número de amostras $n_{samples}$. A transformação linear para mapear de $[0,1]$ para $[a,b]$ é aplicada dentro do loop de amostragem.

3.2.3 Integração Monte Carlo Multidimensional

A função ‘monte_carlo_md‘ é uma generalização para múltiplas dimensões. Recebe a função g, a dimensão d e o número de amostras $n_{samples}$. Para cada amostra, um vetor de d coordenadas aleatórias uniformes é gerado e passado para a função g.

3.2.4 Funções de Teste

Foram implementadas funções específicas para cada integral a ser calculada:

- ‘g_sin_x(double x)‘ para $sin(x)$.
- ‘g_x_cubed(double x)‘ para x^3.
- ‘g_exp_neg_x(double x)‘ para e^{-x}. Para a integral de 0 a ∞, um limite superior prático (100.0) foi utilizado para fins de computação, dado que e^{-x} se aproxima de zero rapidamente.
- ‘g_pi_approx(double coords[], int d)‘ para a aproximação de π. Esta função retorna 1 se um ponto (x,y) (gerado aleatoriamente em $[0,1]\times[0,1]$) está dentro do quarto de círculo unitário ($x^2+y^2 \leq 1$) e 0 caso contrário. A integral resultante representa a área do quarto de círculo ($\pi/4$), e, portanto, o valor de π é obtido multiplicando o resultado por 4.

3.3 Resultados (Parte 2)

Os resultados do método de Monte Carlo foram comparados com os valores analíticos, e o erro relativo percentual foi calculado para diferentes números de amostras.

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

Valor analítico: $1 - cos(1) \approx 0.459698$.

```
--- Integral 1: integral de 0 a 1 de sin(x) dx ---
  Amostras |  Aproximacao  | Erro Rel.
  ---------|---------------|----------
      1000 |      0.459079 |    0.135%
     10000 |      0.460817 |    0.244%
    100000 |      0.458462 |    0.269%
   1000000 |      0.459451 |    0.054%
  ---------|---------------|----------
  Analitico:    0.459698
```

3.3.2 Integral 2: $\int_3^7 x^3 dx$

Valor analítico: $\frac{7^4}{4} - \frac{3^4}{4} = \frac{2401-81}{4} = \frac{2320}{4} = 580$.

```
--- Integral 2: integral de 3 a 7 de x^3 dx ---
  Amostras |  Aproximacao  | Erro Rel.
  ---------|---------------|----------
      1000 |    569.505912 |    1.809%
     10000 |    577.484120 |    0.434%
    100000 |    580.260766 |    0.045%
   1000000 |    580.090541 |    0.016%
  ---------|---------------|----------
  Analitico: 580.000000
```

3.3.3 Integral 3: $\int_0^\infty e^{-x} dx$

Valor analítico: $\lim_{b\to\infty} \int_0^b e^{-x}dx = \lim_{b\to\infty}[-e^{-x}]_0^b = \lim_{b\to\infty}(-e^{-b} + e^0) = 0 + 1 = 1$.
(Para computação, usamos o limite superior de 100.0, pois e^{-x} é desprezível para $x > 100$).

```
--- Integral 3: integral de 0 a infinito de e^(-x) dx ---
  Amostras |  Aproximacao  | Erro Rel.
  ---------|---------------|----------
      1000 |      0.890897 |   10.910%
     10000 |      1.031842 |    3.184%
    100000 |      1.018391 |    1.839%
   1000000 |      1.004121 |    0.412%
  10000000 |      0.999929 |    0.007%
  ---------|---------------|----------
  Analitico:    1.000000
  Nota: Usando limite superior = 100.0 para "aproximar" infinito
```

3.3.4 Integral 4: Aproximação de π

Valor analítico: $\pi \approx 3.141593$.

```
--- Integral 4: Aproximacao do pi Usando o Metodo do Quarto de Circulo ---
  Amostras |  Aproximacao  | Erro Rel.
  ---------|---------------|----------
      1000 |      3.096000 |    1.451%
     10000 |      3.130400 |    0.356%
    100000 |      3.134240 |    0.234%
   1000000 |      3.140156 |    0.046%
  10000000 |      3.142102 |    0.016%
  ---------|---------------|----------
  Analitico:    3.141593
```

3.4 Observações (Parte 2)

Para todas as integrais testadas com o método de Monte Carlo, a precisão da aproximação geralmente melhora com o aumento do número de amostras, conforme esperado pela Lei Forte dos Grandes Números. O erro relativo diminui à medida que n (número de amostras) aumenta. No entanto, a taxa de convergência do método de Monte Carlo é mais lenta (proporcional a $1/\sqrt{n}$) em comparação com os métodos de Newton-Cotes (Trapézio e Simpson), que têm ordens de convergência maiores (proporcionais a h^2 e h^4, respectivamente). Isso significa que, para atingir a mesma precisão, Monte Carlo geralmente requer um número muito maior de avaliações da função, mas sua eficácia brilha em integrais de alta dimensão, onde os outros métodos se tornam computacionalmente inviáveis.