



INTEGRAÇÃO NUMÉRICA USANDO FÓRMULAS DE GAUSS

Exercício Programa 02

Relatório Final

Natanael Magalhães Cardoso, 8914122

Valber Marcelino Filho, 11353165

Professora: Cláudia Peixoto

Turma: 03



Universidade de São Paulo

Escola Politécnica

MAP3121 - Métodos Numéricos e Aplicações



Integração Numérica usando Fórmulas de Gauss

Natanael Magalhães Cardoso^{*}, Valber Marcelino Filho[†]

1. INTRODUÇÃO

Existem muitas técnicas de integração numérica que consistem em aproximar o integrando por um polinômio em uma região e, então, integrá-lo exatamente. Porém, muitas vezes, um integrando mais complicado pode ser fatorado em uma função de peso e outra função melhor aproximada por um polinômio (Golub and Welsch, 1969), como mostra a eq. (1)

$$\int_a^b g(x)dx = \int_a^b w(x)f(x)dx \approx \sum_{j=1}^n w_j f(x_j) \quad (1)$$

A quadratura gaussiana é uma família de métodos de integração numérica baseados em uma escolha determinística de pontos ponderados (ou nós) (Stoer and Bulirsch, 2013). Uma regra de quadratura gaussiana de n -pontos é construída para produzir um resultado exato para polinômios de grau $2n - 1$ ou menos por uma escolha adequada dos nós x_j e pesos w_j .

Este trabalho mostra a implementação de um algoritmo em Python (Van Rossum and Drake, 2009) para o cálculo de integrais duplas usando quadratura gaussiana com nós e pesos (raízes e coeficientes do polinômio de Legendre) calculados no intervalo $[-1, 1]$ com a devida transformação de variáveis para obter uma integral em um domínio de integração genérico (Seção 2). Bem como os resultados para oito casos de teste e a comparação dos resultados obtidos por esta implementação com os resultados obtidos quando a integral é calculada usando a biblioteca SciPy (Virtanen et al., 2020) (Seção 3).

^{*}nUSP: 8914122, Turma: 03

[†]nUSP: 11353165, Turma: 03

2. MÉTODOS

2.1. QUADRATURA GAUSSIANA

2.1.1 Fórmula exata para integração de polinômios \mathcal{P}_5 no intervalo $[-1, 1]$

Queremos conhecer os valores dos pesos w_0, w_1 e w_2 e dos nós x_0, x_1 e x_2 para que a igualdade da eq. (2) seja válida para qualquer polinômio $f(x) \in \mathcal{P}_5$.

$$\int_{-1}^1 f(x)dx = w_0f(x_0) + w_1f(x_1) + w_2f(x_2), \quad \forall f \in \mathcal{P}_5 \quad (2)$$

Pela propriedade da simetria da quadratura gaussiana:

$$x_1 = 0, \quad x_0 = -x_2, \quad w_0 = w_2 \quad (3)$$

Aplicando (3) em (2):

$$\int_{-1}^1 f(x)dx = w_0f(x_0) + w_1f(0) + w_0f(-x_0), \quad \forall f \in \mathcal{P}_5 \quad (4)$$

Então, basta testar para os seguintes polinômios $f_0(x) = 1, f_1(x) = x, f_2(x) = x^2, f_3(x) = x^3, f_4(x) = x^4$ e $f_5(x) = x^5$, usando a propriedade da eq. (5).

$$\int_{-1}^1 x^k dx = \begin{cases} 0, & k \text{ ímpar} \\ \frac{2}{k+1}, & k \text{ par} \end{cases} \quad (5)$$

Para f_1, f_3 e f_5 :

$$\int_{-1}^1 f(x)dx = w_0f(x_0) + 0 - w_0f(x_0) = 0 \quad (6)$$

Para f_0, f_2 e f_4 , temos o seguinte sistema não-linear:

$$\int_{-1}^1 f_0(x)dx = \int_{-1}^1 dx = w_0 + w_1 + w_0 = 2w_0 + w_1 = 2 \quad (7)$$

$$\int_{-1}^1 f_2(x)dx = \int_{-1}^1 x^2 dx = w_0x_0^2 + 0 + w_0(-x_0)^2 = 2w_0x_0^2 = 2/3 \quad (8)$$

$$\int_{-1}^1 f_4(x)dx = \int_{-1}^1 x^4 dx = w_0x_0^4 + 0 + w_0(-x_0)^4 = 2w_0x_0^4 = 2/5 \quad (9)$$

Fazendo (9) \div (8) para calcular x_0 , substituindo x_0 de (10) em (8) para calcular w_0 , substituindo

w_0 de (11) em (7) para calcular w_1 e considerando as relações em (3) para calcular w_2 e x_2 , chega-se à solução do sistema:

$$x_0^2 = \frac{3}{5} \Rightarrow x_0 = -\sqrt{\frac{3}{5}} \quad \text{e} \quad x_2 = -x_0 = \sqrt{\frac{3}{5}} \quad (10)$$

$$w_0 = w_2 = \frac{5}{9} \quad \text{e} \quad w_1 = \frac{8}{9} \quad (11)$$

Assim, substituindo os valores de (10) e (11) em (2) chega-se à fórmula exata da integral no intervalo $[-1, 1]$ de polinômios de grau ≤ 5 na eq. (12)

$$\int_{-1}^1 f(x)dx = \frac{5}{9}f(-\sqrt{3/5}) + \frac{8}{9}f(0) + \frac{5}{9}f(\sqrt{3/5}), \quad \forall f \in \mathcal{P}_5 \quad (12)$$

2.1.2 Fórmula exata para integração de polinômios \mathcal{P}_5 no intervalo $[a, b]$

Para que a eq. (12) seja válida em um intervalo de integração $[a, b]$, $a, b \in \mathbb{R}$, os nós são linearmente transportados do intervalo $[-1, 1]$ para o intervalo $[a, b]$ e os pesos são multiplicados por um fator de escala.

A eq. (13) mostra a fórmula do escalamento min-max

$$t = (b - a) \left[\frac{x - \min(x)}{\max(x) - \min(x)} \right] + a \quad (13)$$

Substituindo os valores de $\min(x) = -1$ e $\max(x) = 1$, calculamos t e dt nas eqs. (14) e (15)

$$t = \frac{(b - a)x}{2} + \frac{b + a}{2} \quad (14)$$

$$dt = \frac{b - a}{2} dx \quad (15)$$

Aplicando a substituição da variável em uma função $g(\cdot)$:

$$\int_a^b g(t)dt = \frac{b - a}{2} \int_{-1}^1 g\left(\frac{b - a}{2}x + \frac{b + a}{2}\right) dx \quad (16)$$

Finalmente, considerando a igualdade (12) dada pela quadratura gaussiana e a substituição de variável (16), a fórmula exata para o cálculo de um polinômio de ordem ≤ 5 em um intervalo $[a, b]$ é dada pela eq. (17)

$$\int_a^b f(x)dx = \frac{5}{9}\gamma f(-\gamma\sqrt{3/5} + \lambda) + \frac{8}{9}\gamma f(\lambda) + \frac{5}{9}\gamma f(\gamma\sqrt{3/5} + \lambda), \quad \forall f \in \mathcal{P}_5 \quad (17)$$

$$\text{com } \gamma = \frac{b - a}{2} \text{ e } \lambda = \frac{b + a}{2}$$

2.1.3 Aproximação da integral dupla em uma região qualquer

Agora, considerando $f(x)$ uma função qualquer (não necessariamente pertencente à \mathcal{P}_{2n-1}), o valor da integral no intervalo $[-1, 1]$ pode ser aproximado como mostra a eq. (18)

$$\int_{-1}^1 f(x)dx \approx \sum_{j=0}^n w_j f(x_j) \quad (18)$$

Aplicando a substituição de variável (16) em (18):

$$\int_a^b f(x)dx \approx \sum_{j=0}^n \frac{b-a}{2} w_j f\left(\frac{b-a}{2}x_j + \frac{b+a}{2}\right) \quad (19)$$

Para uma função de duas variáveis $G(x, y)$, sua integral dupla em uma região $R = \{(x, y) \mid a \leq x \leq b, c(x) \leq y \leq d(x)\}$ ou $R = \{(x, y) \mid a \leq y \leq b, c(y) \leq x \leq d(y)\}$ é aproximada da seguinte forma:

$$I = \int_a^b \int_{c(x)}^{d(x)} g(x, y) dy dx \approx \int_a^b G_i(x) dx \quad (20)$$

onde

$$G_i(x) = \sum_{j=0}^n \frac{d(x_{ij}) - c(x_{ij})}{2} u_{ij} g\left(x_i, \frac{d(x_{ij}) - c(x_{ij})}{2} y_{ij} + \frac{d(x_{ij}) + c(x_{ij})}{2}\right) \quad (21)$$

Assim, a fórmula da integral aproximada de uma função $G(x, y)$ em uma região R é dada pela eq. (22)

$$I \approx \frac{b-a}{2} \sum_{i=0}^n v_i \sum_{j=0}^n \frac{d(x_{ij}) - c(x_{ij})}{2} u_{ij} g\left(\frac{b-a}{2}x_i + \frac{b+a}{2}, \frac{d(x_{ij}) - c(x_{ij})}{2} y_{ij} + \frac{d(x_{ij}) + c(x_{ij})}{2}\right) \quad (22)$$

2.2. IMPLEMENTAÇÃO

A implementação do software para o cálculo de integrais duplas em regiões quaisquer usando quadratura gaussiana é baseada no cálculo de I da eq. (22). A Tabela (1) mostra uma breve descrição das funções implementadas no programa.

■ **Tabela 1:** Descrição das funções

Função	Descrição
<code>get_pairs</code>	Recebe o parâmetro n , acessa o dicionário de dados, calcula os valores negativos e retorna uma quantidade n de pesos e nós (raízes e coeficientes do polinômio de Legendre).
<code>double_gauss_quadrature</code>	Recebe os parâmetros f, a, b, c, d e n e calcula a aproximação da integral dupla em uma região qualquer da função f , sendo a e b os limites da integral mais externa e c e d os limites da integral mais interna. O cálculo é feito usando quadratura gaussiana com n nós.

3. RESULTADOS E DISCUSSÃO

3.1. MÁXIMA PRECISÃO DOS TESTES

A Tabela 2 foi obtida a partir do atributo `float_info` do módulo `sys` do Python (Kong et al.; Van Rossum et al.), que mostra informações sobre cálculos com pontos flutuantes na máquina. Dois parâmetros importantes são o número máximo de dígitos decimais que podem ser representados fielmente em um ponto flutuante e o erro epsilon, que é a diferença entre 1.0 e o menor valor maior que 1.0 que é representável como um ponto flutuante.

■ **Tabela 2:** Informações sobre a precisão do sistema com cálculo usando pontos flutuantes

Parâmetro	Valor
Número máximo de dígitos	15
Epsilon	$2.220446049250313 \cdot 10^{-16}$

3.2. CASOS DE TESTE

O programa foi avaliado a partir de oito casos de teste listados a seguir

Caso de Teste 1.a: Cálculo do volume de um cubo cujas arestas tem comprimento 1.

$$\int_0^1 \int_0^1 1 dx dy \quad (23)$$

Caso de Teste 1.b: Cálculo do volume de um tetraedro com vértices $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$ e $(0, 0, 1)$.

$$\int_0^1 \int_0^{1-x} (1-x-y) dy dx \quad (24)$$

Caso de Teste 2.a: Cálculo da área da região no primeiro quadrante limitada pelos eixos e pela curva $y = 1 - x^2$ (sequência de integração: $dydx$).

$$\int_0^1 \left[\int_0^{1-x^2} dy \right] dx \quad (25)$$

Caso de Teste 2.b: Cálculo da área da região no primeiro quadrante limitada pelos eixos e pela curva $y = 1 - x^2$ (sequência de integração: $dx dy$).

$$\int_0^1 \left[\int_0^{\sqrt{1-y}} dx \right] dy \quad (26)$$

Caso de Teste 3.a: Cálculo da área da superfície descrita por $z = e^{\frac{y}{x}}$, $0.1 \leq x \leq 0.5$, $x^3 \leq y \leq x^2$.

$$\int_{0.1}^{0.5} \int_{x^3}^{x^2} \sqrt{\left(-\frac{e^{\frac{y}{x}} y}{x^2}\right)^2 + \left(\frac{e^{\frac{y}{x}}}{x}\right)^2 + 1} dy dx \quad (27)$$

Caso de Teste 3.b: Cálculo do volume abaixo da superfície descrita por $z = e^{\frac{y}{x}}$, $0.1 \leq x \leq 0.5$, $x^3 \leq y \leq x^2$.

$$\int_{0.1}^{0.5} \int_{x^3}^{x^2} e^{\frac{y}{x}} dy dx \quad (28)$$

Caso de Teste 4.a: Cálculo do volume da calota esférica de altura $\frac{1}{4}$ da esfera de raio 1.

$$\int_0^{\frac{1}{4}} \int_0^{\sqrt{\frac{1}{4}-x^2}} 2\pi y dy dx \quad (29)$$

Caso de Teste 4.b: Cálculo do volume do sólido de revolução obtido da rotação da região R , em torno do eixo y , delimitada por $x = 0$, $x = e^{-y^2}$, $y = -1$ e $y = 1$.

$$\int_{-1}^1 \int_0^{e^{-y^2}} 2\pi x dx dy \quad (30)$$

3.3. VALIDAÇÃO DOS RESULTADOS

■ **Tabela 3:** Comparação dos resultados obtidos pelo algoritmo de quadratura gaussiana (QG) implementado neste trabalho com os resultados obtidos pelo uso da biblioteca Scipy com 22 casas decimais.

Teste	# Nós	Resultado (QG)	Resultado (Scipy)	Erro
1.a	6	1.0		0.0
	8	1.0	1.0	0.0
	10	1.0		0.0
1.b	6	0.1666666666666666296592		$5.551115 \cdot 10^{-17}$
	8	0.1666666666666666574148	0.1666666666666666851704	$2.775558 \cdot 10^{-17}$
	10	0.1666666666666666851704		0.0
2.a	6	0.66666666666666667406815		$1.110223 \cdot 10^{-16}$
	8	0.6666666666666666296592	0.6666666666666666296592	0.0
	10	0.66666666666666667406815		$1.110223 \cdot 10^{-16}$
2.b	6	0.6670464379156135770188		$3.797712 \cdot 10^{-4}$
	8	0.6668355801001766280933	0.66666666666666668517038	$1.689134 \cdot 10^{-4}$
	10	0.6667560429365088081610		$8.937627 \cdot 10^{-5}$
3.a	6	0.1054978824004978721351		$2.206568 \cdot 10^{-14}$
	8	0.1054978824005199378178	0.1054978824005199378178	0.0
	10	0.1054978824005199378178		0.0
3.b	6	0.0333055661162371882678		$5.107026 \cdot 10^{-15}$
	8	0.0333055661162320743029	0.0333055661162320812418	$6.938894 \cdot 10^{-18}$
	10	0.0333055661162320743029		$6.938894 \cdot 10^{-18}$
4.a	6	0.1799870791119152213522		$2.775558 \cdot 10^{-17}$
	8	0.1799870791119152491078	0.1799870791119152491078	0.0
	10	0.1799870791119152491078		0.0
4.b	6	0.00000000000000002220446		$2.220446 \cdot 10^{-16}$
	8	0.0000000000000000555112	0.0	$5.551115 \cdot 10^{-17}$
	10	0.00000000000000001110223		$1.110223 \cdot 10^{-16}$

Para validação, comparamos os valores das integrais calculadas pelo programa implementado com os valores das integrais calculadas pela biblioteca SciPy do Python. Estes resultados estão dispostos na Tabela 3, que sumariza todos os casos de testes calculados de ambas as formas para diferentes valores de nós.

É notado que praticamente todos os erros são da ordem do máximo número de casas decimais fielmente representáveis por um ponto flutuante na máquina onde se ocorreu o teste (como visto na Tabela 2). Com exceção do Caso de Teste 2.b que apresenta um erro da ordem de 10^{-4} .

A discrepância de erros entre os Casos de Teste 2.a e 2.b mostra o efeito da propagação de erros de arredondamentos causados em operação com pontos flutuantes. Isto é, o limite superior de integração do Caso de Teste 2.b causa maior erro por conta da raiz quadrada.

4. CONCLUSÃO

Foi possível implementar um algoritmo para o cálculo de integrais duplas numericamente usando quadratura gaussiana em uma região de integração qualquer a partir da transformação do intervalo $[-1, 1]$, onde as raízes e coeficientes do polinômio de Legendre foram calculadas, em um intervalo $[a, b]$ com erros comparáveis aos resultados obtidos pela biblioteca SciPy.

REFERÊNCIAS

- Gene H. Golub and John H. Welsch. Calculation of gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969. doi: 10.1090/s0025-5718-69-99647-1. URL <https://doi.org/10.1090/s0025-5718-69-99647-1>.
- Qingkai Kong, Timmy Siau, and Alexandre M Bayen. Python programming and numerical methods : a guide for engineers and scientists. <https://pythonnumericalmethods.berkeley.edu/notebooks/chapter09.02-Floating-Point-Numbers.html>. Accessed: 2022-06-01.
- J Stoer and R Bulirsch. *Introduction to numerical analysis*. Texts in Applied Mathematics. Springer, New York, NY, 2 edition, March 2013.
- Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1441412697.
- Guido Van Rossum et al. The python standard library documentation: Sys module. https://docs.python.org/3/library/sys.html#sys.float_info. Accessed: 2022-06-01.
- Pauli Virtanen et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.



Universidade de São Paulo

Escola Politécnica

MAP3121 - Métodos Numéricos e Aplicações