Giovanna Girotto - 11803416
Vittor Braide Costa - 11806920

Relatório EP2 MAP3121 - Fórmulas de Integração Numérica de Gauss

São Paulo, SP

Junho, 2022

Giovanna Girotto - 11803416
Vittor Braide Costa - 11806920

Relatório EP2 MAP3121 - Fórmulas de Integração Numérica de Gauss

Relatório da Tarefa 2 - *Fórmulas de Integração Numérica de Gauss* proposto pela disciplina MAP3121 - Métodos Numéricos e Aplicações

Escola Politécnica da Universidade de São Paulo

São Paulo, SP

Junho, 2022

Lista de ilustrações

Sumário

1 Introdução

O presente relatório tem como objetivo implementar fórmulas de integração numérica conhecidas como Fórmulas de Integração de Gauss e aplicá-las ao cálculo de algumas integrais duplas. Para isso, primeiro apresenta-se o método de integração numérica utilizando as fórmulas de Gauss.

Feito isso, deduz-se as fórmulas para funções polinomiais de forma que, para esses casos, a solução seja exata para um polinômio de grau n. Além disso, parte-se da fórmula bem conhecida na literatura pra grau menor ou igual a 5 em um intervalo fixo $[-1, 1]$, chega-se em fórmulas para intervalos arbitrários do tipo $[a, b]$. Para tanto, é descrita a técnica de mudança de variável necessária para a aplicação da fórmula em outros intervalos.

Finalmente, apresenta-se o método para calcular integrais duplas através da integração de Gauss, o que será testado no código em linguagem *Python* elaborado neste Exercício Programa. Assim, testa-se o código e apresenta-se o resultado de quatro exercícios propostos no enunciado da tarefa para três valores de n: 6, 8 e 10.

2 Fórmulas de Gauss

Nos diversos problemas de engenharia, deseja-se criar um modelo matemático que represente um sistema idealizado. Nesse sentido, alguns processos de integração precisam ser realizados de maneira numérica, permitindo uma simulação fidedigna aos resultados reais.

Por conseguinte, alguns métodos de integração numérica foram idealizados para viabilizar a realização de simulações ou resolução de problemas matemáticos. Entre eles, as Fórmulas de Gauss são constantemente utilizadas, já que apresentam boa precisão, em especial quando trata-se de polinômios.

Tal método pode ser introduzido pela forma de integração de uma função f de a até b, em que utiliza-se um somatório de pesos ω_j multiplicados por n nós x_j. Vale ressaltar que os nós pertencem ao intervalo $[a, b]$ e que existe um erro $E_n(f)$ nesse processo.

$$\int_a^b f(x)dx = \sum_{j=1}^{n} f(x_j) + E_n(f) \tag{2.1}$$

Dessa forma, as Fórmulas de Gauss são constantemente aplicadas para integração de funções polinomiais, em que determina-se os pesos e os nós para uma aproximação exata.

2.1 Aplicação em funções polinomiais

Assim como comentado anteriormente, a integração pelas Fórmulas de Gauss são muito eficientes em funções polinomiais, já que a escolha correta de pesos e nós possibilita a identificação de uma solução exata. Isso acontece, visto que é possível definir um intervalo simétrico da integração dada pela Equação 2.1, como $[-1, 1]$. Assim, aplicando o somatório da multiplicação de nós e pesos para $n = 2$ obtêm-se os seguintes resultados para as funções indicadas:

$$\begin{cases} f_1(x) = 1 \Rightarrow \int_{-1}^{1} 1dx = 2 = w_1 + w_2 \\ f_2(x) = x \Rightarrow \int_{-1}^{1} xdx = 0 = w_1x_1 + w_2x_2 \\ f_3(x) = x^2 \Rightarrow \int_{-1}^{1} x^2dx = \frac{2}{3} = w_1x_1^2 + w_2x_2^2 \\ f_4(x) = x^3 \Rightarrow \int_{-1}^{1} x^3dx = 0 = w_1x_1^3 + w_2x_2^3 \end{cases} \tag{2.2}$$

De imediato, é possível observar que o resultado final corresponde a um sistema de 4 equações e 4 incógnitas. Além disso, sabendo que o intervalo utilizado é simétrico, espera-se que $x_1 = -x_2$ e, consequentemente, $w_1 = w_2$. Assim, por meio de tais simplificações, é fácil encontrar os valores dos pesos w_1 e w_2 e dos pontos x_1 e x_2. Analogamente, é possível replicar esse processo para um número n de operações e obter valores tabelados de x_j e w_j.

Nesse processo, percebe-se que para o caso geral, o número de equações é igual a $2n$ e a última função integrada apresenta grau $2n-1$ (no caso específico para $n = 2$, tem-se 4 equações e $f_3(x)$ tem grau 3). Dessa forma, é possível concluir que para um polinômio de grau k deve-se respeitar a relação $0 \leq k \leq 2n-1$, já que para $k \geq 2n$ tem-se um sistema sobredeterminado.

Essa restrição pode ser obtida também quando aplica-se a equação 2.1 para a integral de $f(x) = (x-x_1)^2(x-x_2)^2...(x-x_n)^2$ que apresenta grau $2n$ e resultado nulo, sendo que na verdade a integral de tal função em $[a, b]$ é positiva. Assim sendo, se a relação 2.1 é válida para um polinômio, ou seja $E_n(f) = 0$, é possível provar a seguinte relação:

$$\int_a^b (x-x_1)(x-x_2)...(x-x_n)q(x)dx = 0 \tag{2.3}$$

Dado que o polinômio fatorado apresenta grau igual a n e ao final o grau da função completa deve ser menor que $2n$, $q(x)$ deve ser uma função polinomial de grau menor ou igual a $n-1$. Assim, aplicando a relação obtida pelas Fórmulas de Gauss, tem-se:

Seja $F(x) = (x-x_1)(x-x_2)...(x-x_n)q(x)$

$$\int_a^b F(x)dx = \sum w_j F(x_j) = \sum w_j (x_j - x_1)(x_j - x_2)...(x_j - x_n)q(x_j) \tag{2.4}$$

Desse modo, independentemente do valor que $q(x_j)$ assume, o somatório sempre se anula para $x_1, x_2, ..., x_n$ igual aos nós da fórmula de integração e n arbitrário.

Dando sequência, para o presente trabalho, deseja elaborar um algoritmo que calcule a integral de determinadas funções para $n = 6, n = 8$ e $n = 10$. Dessa forma, os valores encontrados pela forma supracitada podem ser encontrados abaixo. Vale ressaltar que caso a função f que deseja-se integrar seja polinomial de grau $1 \leq k \leq 2n-1$, o erro final $E_n(f)$ é nulo.

x_j	w_j
-0,9324695142031520278123016	0,1713244923791703450402961
-0,6612093864662645136613996	0,3607615730481386075698335
-0,2386191860831969086305017	0,4679139345726910473898703
0,2386191860831969086305017	0,4679139345726910473898703
0,6612093864662645136613996	0,3607615730481386075698335
0,9324695142031520278123016	0,1713244923791703450402961

Tabela 1 – Pesos e nós para $n = 6$

x_j	w_j
-0,9602898564975362316835609	0,1012285362903762591525314
-0,7966664774136267395915539	0,2223810344533744705443560
-0,5255324099163289858177390	0,3137066458778872873379622
-0,1834346424956498049394761	0,3626837833783619829651504
0,1834346424956498049394761	0,3626837833783619829651504
0,5255324099163289858177390	0,3137066458778872873379622
0,7966664774136267395915539	0,2223810344533744705443560
0,9602898564975362316835609	0,1012285362903762591525314

Tabela 2 – Pesos e nós para $n = 8$

x_j	w_j
-0,9739065285171717200779640	0,0666713443086881375935688
-0,8650633666889845107320967	0,1494513491505805931457763
-0,6794095682990244062343274	0,2190863625159820439955349
-0,4333953941292471907992659	0,2692667193099963550912269
-0,1488743389816312108848260	0,2955242247147528701738930
0,1488743389816312108848260	0,2955242247147528701738930
0,4333953941292471907992659	0,2692667193099963550912269
0,6794095682990244062343274	0,2190863625159820439955349
0,8650633666889845107320967	0,1494513491505805931457763
0,9739065285171717200779640	0,0666713443086881375935688

Tabela 3 – Pesos e nós para $n = 10$

2.2 Mudança de Variável

Para intervalos fixos, em especial $[-1, 1]$, os pesos e nós para aproximar integrais são bem conhecidos na literatura e podem ser encontrados em tabelas. Usando a lógica apresentada na equação 2.2, a fórmula

$$\int_{-1}^{1} f(x)dx \approx \frac{5}{9}f(-\sqrt{6}) + \frac{8}{9}f(0) + \frac{5}{9}f(\sqrt{6}) \tag{2.5}$$

é exata para a integral de -1 a 1 de polinômios de grau menor ou igual a 5.

A fim de se obter uma fórmula para a integral de uma função em um intervalo fixo $[a, b]$ genérico, faz-se uma mudança de variável $(t \to x)$ de tal forma que:

$$\begin{cases} t = -1 \Rightarrow x = a \\ t = -1 \Rightarrow x = b \end{cases}$$

Assim, sejam C_1 e C_2 duas constantes, faz-se a transformação linear:

$$x(t) = C_1 + C_2 t \tag{2.6}$$

$$a = C_1 - C_2 \tag{2.7}$$

$$b = C_1 + C_2 \tag{2.8}$$

$$\text{Resolve-se o sistema: } C_1 = \frac{a+b}{2};\ C_2 = \frac{b-a}{2} \tag{2.9}$$

$$\Rightarrow \boxed{x(t) = \frac{b-a}{2}t + \frac{a+b}{2}} \tag{2.10}$$

Os pesos também devem ser multiplicados por um fator de escala igual ao coeficiente angular da transformação linear das variáveis, isto é, multiplica-se cada peso por $\frac{b-a}{2}$.

$$\int_a^b f(x)dx = \int_{-1}^1 \frac{b-a}{2} f(\frac{b-a}{2}t + \frac{a+b}{2})dt \tag{2.11}$$

Dessa forma, chega-se na fórmula de Gauss exata para polinômios de grau menor ou igual a 5 em um intervalo $[a, b]$:

$$\int_a^b f(x)dx = \frac{b-a}{2} \int_{-1}^1 \frac{5}{9} f(\frac{b-a}{2}(-\sqrt{6}) + \frac{a+b}{2}) + \frac{8}{9} f(\frac{a+b}{2}) + \frac{5}{9} f(\frac{b-a}{2}(\sqrt{6}) + \frac{a+b}{2}) \tag{2.12}$$

2.3 Integrais duplas

A partir da técnica de mudança de variável, é possível utilizar a fórmula de Gauss exata em $[-1, 1]$ para polinômios de grau menor ou igual a 5 a fim de calcular o valor de uma integral dupla.

Assim, supõe-se uma região R do espaço descrita por:

$$(x, y)|a \leq x \leq b, c(x) \leq y \leq d(x) \tag{2.13}$$

E deseja-se calcular a integral:

$$I = \int\int_R f(x, y)dxdy \tag{2.14}$$

$$\Rightarrow I = \int_a^b \int_{c(x)}^{d(x)} f(x, y)dydx \tag{2.15}$$

Para calcular essa integral, separa-se em duas etapas. Primeiro, calcula-se o valor de $F(x, y) = \int_{c(x)}^{d(x)} f(x, y)dy$ aplicando a mudança de variável do intervalo $[c(x), d(x)] \rightarrow [-1, 1]$. Nota-se que não é um intervalo fixo em x, isto é, os extremos variam para cada valor de x.

Dessa forma, para calcular d e c, usam-se os nós (p) da fórmula de Gauss para calcular os diferentes valores de x a partir da mudança de variáveis:

$$x_i = \frac{b-a}{2} p_i + \frac{a+b}{2} \tag{2.16}$$

E, para cada valor de x, obtém-se o intervalo $[c(x), d(x)]$ correspondente. A partir disso, pode-se calcular y:

$$y_i = \frac{d(x_i) - c(x_i)}{2} p_i + \frac{d(x)_i + c(x)_i}{2} \tag{2.17}$$

E, assim, calcula-se facilmente o valor da primeira integral fazendo a multiplicando os pesos pelo fator de correção $\frac{d(x)-c(x)}{2}$.

$$F(x,y) = \sum_{j=1}^{n} w_{ij} \frac{d(x_i) - c(x_i)}{2} f(x_i, y_i j) \tag{2.18}$$

Feito isso, basta utilizar este resultado como a função a ser integrada no intervalo $[a,b]$, desta vez fixo. Com a mesma lógica da sessão anterior e corrigindo os pesos por um fator de $\frac{b-a}{2}$, chega-se na integral I desejada.

$$I = \sum_{i=1}^{n} w_i \frac{b-a}{2} F(x_i) \tag{2.19}$$

A mesma lógica desenvolvida pode ser aplicada para regiões descritas por:

$$(x,y)|c \leq y \leq d, a(y) \leq x \leq b(y) \tag{2.20}$$

3 Tarefa

Todos os exercícios desenvolvidos nessa parte foram propostos no enunciado do trabalho. Para execução do código, elaborou-se uma função genérica denominada 'integracao_gaussiana', a qual pode ser visualizada abaixo.

```
def integracao_gaussiana(W,p,n,a,b,c,d,f):
    I1=0
    for i in range(n):
        x=((b-a)*p[i]+(a+b))/2
        u=(b-a)*W[i]/2
        for j in range(n):
            v=(d(x)-c(x))*W[j]/2
            y=((d(x)-c(x))*p[j]+(c(x)+d(x)))/2
            I1+=u*v*f(x,y)
    return I1
```

Os parâmetros de entrada de tal função consistem nos pesos w_j e nos pontos x_j para j variando até n, valor que também deve ser fornecido ao chamar a função. Esses valores são tabelados e podem ser encontrados nas tabelas 1, 2 e 3.

Em sequência, a e b também são variáveis solicitadas para aplicar a função e consistem nos intervalos constantes da integração mais externa. Por último, solicita-se c, d e f, que representam variáveis do tipo *lambda* no *Python*, já que c e d são utilizados para descrever os intervalos de integração com dependência em uma variável da integral dupla e f diz respeito à função em que deseja integrar propriamente. Em suma, as variáveis de entrada podem ser esquematizadas numa integral genérica do tipo:

$$\int_a^b \int_{c(x)}^{d(x)} f(x,y)dydx \tag{3.1}$$

Assim sendo, em cada exercício, inseriu-se todas as 8 variáveis consideradas *inputs* da função para cada exercício em particular. Ao final, um código *main* foi produzido para possibilitar uma interface com o usuário e demonstrar os resultados de acordo com os interesses dele. Todos os resultados estão apresentados com precisão dupla, já o código completo pode ser encontrado no Apêndice A.

3.1 Exercício 1

O primeiro exercício é divido em duas partes, a primeira solicita o cálculo do volume de um cubo de aresta com 1 unidade de comprimento. Dessa forma, a integração fica representada pela equação abaixo:

$$V = \int_0^1 \int_0^1 1dydx \tag{3.2}$$

Resolvendo tal integral de forma analítica, encontra-se 1 unidade de volume. Já para a solução numérica, aplica-se $a = b = c = d = f = 1$ como *input* da função e obtêm-se o resultado abaixo:

n	**Solução**
6	1.
8	1.
10	1.

Tabela 4 – Solução numérica volume do cubo

De imediato, é possível analisar que a solução encontrada é exata independentemente de n e é coerente com o valor analítico encontrado, reforçando a hipótese de grande efetividade das Fórmulas de Gauss para funções polinomias.

Em sequência, deseja-se calcular o volume do tetraedro formado pelos pontos $(0, 0, 0), (1, 0, 0), (0, 1, 0)$ e $(0, 0, 1)$. Para tanto, inicialmente é necessário formular como fica a integral do problema e, retomando conceitos de álgebra linear, sabe-se que a equação do plano é dada por:

$$ax + by + cz + d = 0 \tag{3.3}$$

Lembrando que o vetor normal ao plano é dado por $\vec{n} = (a, b, c)$, é possível encontrar $\vec{n}$ pelo produto vetorial de dois vetores que compõe o plano. Desse modo, fazendo:

$$\begin{cases} \vec{AB} = (0, 0, 1) - (1, 0, 0) = (-1, 0, 1) \\ \vec{AC} = (0, 1, 0) - (1, 0, 0) = (-1, 1, 0) \end{cases} \tag{3.4}$$

O $\vec{n}$ é:

$$\vec{n} = \vec{AB} \times \vec{AC} = (-1, -1, -1) \tag{3.5}$$

Tem-se, portanto, que $a = b = c = -1$. Substituindo as coordenadas $(1, 0, 0)$ do ponto que pertence ao plano na equação 3.3, encontra-se que $d = 1$. Nesse sentido, rearranjando a equação do plano, chega-se ao final em uma equação que relaciona z em função de x e y

$$z = f(x, y) = -x - y + 1 \tag{3.6}$$

Considerando a projeção do tetraedro no plano xy, sabe-se que os intervalos de integração são tais que $a = 0$, $b = 1$, $c = 0$ e $d = -x + 1$. Finalmente, o volume será dado por:

$$\int_0^1 \int_0^{-x+1} (-x - y + 1)dydx \tag{3.7}$$

Resolvendo tal integral analiticamente, encontra-se $1,\overline{6}$ e para encontrar a solução numérica aplicou-se novamente a função genérica criada em *Python* com os *inputs* supracitados e obteve-se os seguintes resultados:

n	**Solução**
6	0.16666667
8	0.16666667
10	0.16666667

Tabela 5 – Solução numérica volume do tetraedro

Novamente, a solução é exata e os mesmo comentários feitos para o volume do cubo podem ser aplicados.

3.2 Exercício 2

O exercício 2 corresponde ao cálculo de uma área A da região delimitada pelos eixos de coordenadas e pela curva $y = 1 - x^2$. Nesse sentido propõe-se aplicar a integral de duas diferentes maneiras.

$$A = \int_0^1 \int_0^{1-x^2} dydx = \int_0^1 \int_0^{\sqrt{1-y}} dxdy \tag{3.8}$$

Prontamente, é possível identificar que a integral é a mesma, mas os limites da primeira são representados por funções polinomiais. Além disso, resolvendo a integral de forma analítica o resulto é igual a $\frac{2}{3}$.

Sendo $a = 0$, $b = 1$, $c = 1$, $d = 1 - x^2$ e $f = 1$, aplica-se a primeira forma de calcular a área, por meio da função elaborada em *Python*, Assim sendo, os resultados estão na forma exata e coerentes com a solução analítica.

n	**Solução**
6	0.66666667
8	0.66666667
10	0.66666667

Tabela 6 – Solução numérica da integração com intervalos em função de x

Posteriormente, alguns valores das variáveis de entrada são alterados para calcular a área da segunda maneira. Desse modo, usando a função com $a = 0$, $b = 1$, $c = 0$ e $d = \sqrt{1-y}$, obtêm-se os valores abaixo.

n	**Solução**
6	0.66704644
8	0.66683558
10	0.66675604

Tabela 7 – Solução numérica da integração com intervalos em função de y

É possível concluir que, diferentemente da integração feita pela primeira forma, os resultados obtidos não são mais exatos, dado que a função $d(y) = \sqrt{1-y}$ não é polinomial. Dessa forma, o erro das Fórmulas de Gauss $E_n(f)$ não é mais nulo e deve ser considerado. De maneira mais precisa, os erros obtidos para os diferentes valores de n foram calculados e disponibilizados abaixo, pelos resultados percebe-se que ele é decrescente conforme aumenta-se o numéro de n.

n	**Erro**
6	0.06%
8	0.03%
10	0.01%

Tabela 8 – Erro das Fórmulas de Gauss

3.3 Exercício 3

Dada a superfície descrita por $z = e^{y/x}$, com $0.1 \leq x \leq 0.5$, $x^3 \leq y \leq x^2$, neste exercício pede-se a área desta superfície (A) e o volume abaixo dela (V).

Primeiro, analisa-se o volume de uma função $z = f(x, y)$, dado por:

$$V = \int\int_A f(x,y)dA \tag{3.9}$$

Substituindo os valores e parâmetros para o exercício em questão:

$$V = \int_{0.1}^{0.5}\int_{x^3}^{x^2} e^{y/x}dydx \tag{3.10}$$

Resolvendo analiticamente, tem-se:

$$V = \int_{0.1}^{0.5}(xe^{y/x})|_{x^3}^{x^2}dx = \int_{0.1}^{0.5} x(e^x - e^{x^2})dx = 0.0333055661162... \tag{3.11}$$

Esse resultado representa o valor exato do volume, conferido por meio do *Software Mathematica*.

Calcula-se, agora, a mesma integral, mas usando a fórmula de Gauss. Para tanto, usa-se o código "integração_gaussiana"desenvolvido no Exercício Programa. Feito isso, obtém-se a solução numérica da integração para diferentes n's:

n	**Solução**
6	0.033305567
8	0.033305567
10	0.033305567

Tabela 9 – Solução numérica da integração utilizando a Quadratura Gaussiana

Passa-se, finalmente, para a segunda parte do exercício, em que analisa-se a área da superfície descrita por $z = e^{y/x}$. Pra tanto, faz-se uso da relação para cálculo da área de uma superfície do tipo $z = f(x, y)$, dada por:

$$A = \int\int_A \sqrt{f_x(x,y)^2 + f_y(x,y)^2 + 1} dA \tag{3.12}$$

Substituindo os valores e parâmetros para o exercício em questão:

$$A = \int_{0.1}^{0.5}\int_{x^3}^{x^2} \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial x}\right)^2 + 1}\ dxdy = \int_{0.1}^{0.5}\int_{x^3}^{x^2} \sqrt{\left(\frac{-ye^{y/x}}{x^2}\right)^2 + \left(\frac{e^{y/x}}{x}\right)^2 + 1}\ dxdy \tag{3.13}$$

Resolvendo tal integral analiticamente, obtém-se que o resultado é $A = 0.105497882401$, resultado também conferido pelo *Software Mathematica*.

Por fim, de forma análoga ao cálculo do volume, usa-se o código desenvolvido para se obter o resultado da integração numérica através da fórmula de Gauss. Feito isso, expressa-se os resultados para diferentes n's na tabela abaixo:

n	**Solução**
6	0.10549788
8	0.10549788
10	0.10549788

Tabela 10 – Solução numérica da integração utilizando a Quadratura Gaussiana

De imediato, verifica-se que a solução é tanto para o cálculo do volume quanto para o da superfície, ao menos dentro da precisão analisada. Contudo, a função da superfície integrada não se trata de um polinômio, reforçando a hipótese de que a Fórmula de Gauss é capaz de expandir sua efetividade para diferentes funções que não as de polinômio, para as quais oferece solução exata pra grau menor ou igual a 5.

3.4 Exercício 4

Neste exercício pedem-se as soluções numéricas de dois volumes: o de uma calota esférica de altura $\frac{1}{4}$ de uma esfera de raio 1 e o do sólido de revolução obtido na rotação da região, em torno do eixo y, delimitada por $x = 0$, $x = e^{-y^2}$, $y = -1$ e $y = 1$.

Vale ressaltar que o volume do sólido de revolução obtido da rotação de uma região R em torno de y é dado pela fórmula:

$$V = 2\pi \int\int_R d_y(x, y)dxdy \tag{3.14}$$

Onde $d_y(x, y)$ é a distância do ponto (x, y) ao eixo y.

Assim sendo, analisa-se primeiro o volume da calota esférica. Para isso, considera-se a função da circunferência mostrada na imagem abaixo rotacionada, a qual será rotacionada em torno no eixo y para se obter o volume desejado:

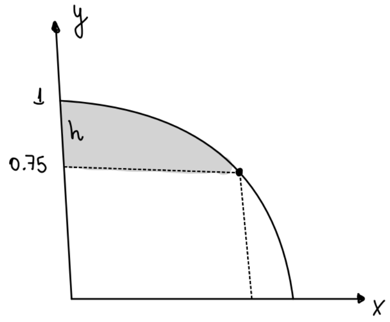

Figura 1 – Representação gráfica do problema da calota esférica

A partir disso, faz-se:

$$x^2 + y^2 = 1 \Rightarrow x = \sqrt{1 - y^2}$$
$$V = 2\pi \int_{0.75}^{1} \int_{0}^{\sqrt{1-y^2}} xdxdy$$

Resolvendo a integral analiticamente, obtém-se o resultado de $V = 0.179987079112$, também conferido pelo *Software Mathematica*.

Aplicando esta fórmula ao código desenvolvido no exercício programa, tem-se os resultados da integração numérica pela fórmula de Gauss para diferentes n's:

n	**Solução**
6	0.17998708
8	0.17998708
10	0.17998708

Tabela 11 – Solução numérica da integração utilizando a Quadratura Gaussiana

Passa-se, então, para a segunda parte do exercício. Nele, analisa-se a região delimitada por x e y como mostra a imagem:

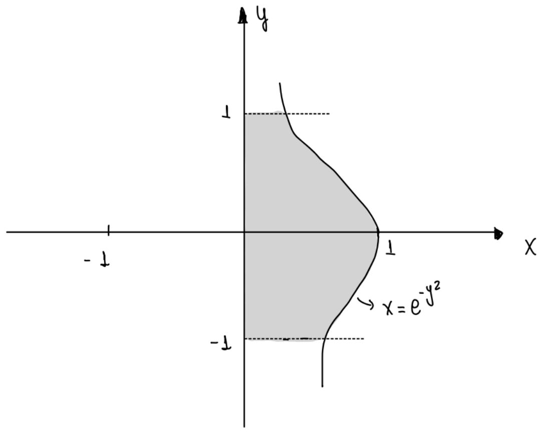

Figura 2 – Representação gráfica da área a ser rotacionada

Nesse caso, a distância $d_y(x, y)$ é a própria coordenada x e o volume fica:

$$V = 2\pi \int_{-1}^{1} \int_{0}^{e^{-y^2}} x dx dy \tag{3.15}$$

Resolvendo a integral analiticamente, obtém-se o resultado de $V = 3.75824963423$, também conferido pelo *Software Mathematica*.

Novamente, aplicando esta fórmula ao código desenvolvido no exercício programa, tem-se os resultados da integração numérica pela fórmula de Gauss para diferentes n's:

n	**Solução**
6	3.75816503
8	3.75824926
10	3.75824963

Tabela 12 – Solução numérica da integração utilizando a Quadratura Gaussiana

Percebe-se que neste último caso, o resultado da integração numérica pela fórmula de Gauss gera diferenças em relação à solução exata, em que o erros estão representados na Tabela abaixo. Tal resultado está de acordo com o fato de que a fórmula de Gauss oferece a solução exata para a integração de funções polinomiais de grau menor ou igual a 5, o que não é o caso da função $y = e^{-y^2}$. Ainda assim, os valores fornecidos possuem erros pequenos em relação ao valor exato, o que mostra as diferentes aplicações possíveis para a fórmula estudada.

n	**Erro**
6	0.008%
8	0.00004%
10	4×10^{-7}%

Tabela 13 – Erro das Fórmulas de Gauss

4 Conclusão

A elaboração de modelos matemáticos para representar modelos físicos é de grande importância para engenharia. Por meio das simulações computacionais, atualmente é possível desenvolver projetos extremamente ousados e complexos que desafiam constantemente a humanidade. Nesse sentido, justifica-se a grande importância dos métodos numéricos.

Assim sendo, para elaboração de tais simulações, é necessário conhecer métodos como as Fórmulas de Gauss que possibilitam a execução de operações matemáticas numericamente. Desse modo, ao longo do presente trabalho, foi possível identificar como encontra-se os pesos e pontos para cálculo de uma integral simétrica, além da expansão de tal caso mais simples para integração em um intervalo genérico $[a, b]$, com a mudança de variável.

Por último, aplicou-se, ainda, o caso para integral dupla que possibilita uma amplitude de aplicação do método muito grande. Assim, em conclusão, foi possível aplicar as Fórmulas de Gauss em diferentes intervalos e funções, as quais demonstram algumas falhas do método ou validaram a efetividade da técnica que encontra inclusive soluções exatas.

Apêndices

APÊNDICE A – Código completo

Código desenvolvido no *Python*

```python
import numpy as np
def main():
    p6=([-0.9324695142031520278123016,-0.6612093864662645136613996,
     -0.2386191860831969086305017,0.2386191860831969086305017,
     0.6612093864662645136613996,0.9324695142031520278123016])
    #Pesos
    W6=([0.1713244923791703450402961,0.3607615730481386075698335,
     0.4679139345726910473898703,0.4679139345726910473898703,
     0.3607615730481386075698335,0.1713244923791703450402961])

    p8=([-0.9602898564975362316835609,-0.7966664774136267395915539,
     -0.5255324099163289858177390,-0.1834346424956498049394761,
     0.1834346424956498049394761,0.5255324099163289858177390,
     0.7966664774136267395915539,0.9602898564975362316835609])
    W8=([0.1012285362903762591525314,0.2223810344533744705443560,
     0.3137066458778872873379622,0.3626837833783619829651504,
     0.3626837833783619829651504,0.3137066458778872873379622,
     0.2223810344533744705443560,0.1012285362903762591525314])

    p10=([-0.9739065285171717200779640,-0.8650633666889845107320967,
     -0.6794095682990244062343274,-0.4333953941292471907992659,
     -0.1488743389816312108848260,0.1488743389816312108848260,
     0.4333953941292471907992659,0.6794095682990244062343274,
     0.8650633666889845107320967,0.9739065285171717200779640])
    W10=([0.0666713443086881375935688,0.1494513491505805931457763,
     0.2190863625159820439955349,0.2692667193099963550912269,
     0.2955242247147528701738930,0.2955242247147528701738930,
     0.2692667193099963550912269,0.2190863625159820439955349,
     0.1494513491505805931457763,0.0666713443086881375935688])

    print("Trabalho da disciplina MAP3121 - Métodos Numéricos e Aplicações")
    print("Desenvolvido por: Giovanna Girotto - 11803416 & Vittor Braide Costa - 11806920")
    print("Para resolução do EP, foi construída uma função genérica denominada 'integracao_gaussiana' e, para demonstração dos resultados, a função foi chamada para cada exercício proposto no enunciado. Os resultados são demonstrados em precisão dupla")
    exercicio = int(input("Assim sendo, digite o número do exercício (1,2,3 ou 4) que deseja visualizar o resultado. Caso queira visualizar todos os resultados simultaneamente, digite 0"))
```

```
    if exercicio==1:
        print("Exercicio 1")
        print("Volume do Cubo - Solução exata = 1.")
        ex11=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex11)
        print("Volume do tetraedro - Solução exata = 1/6")
        ex12=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex12)
        continua=int(input("Digite 1 para visualizar a solução dos outros exercícios ou digite 0 para encerrar o código"))
        if continua==1:
            print("Exercicio 2")
            print("Integral com extremos 0 e 1-x^2 - Solução exata = 2/3")
            ex21=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex21)
            print("Integral com extremos 0 e sqrt(1-y) - Solução exata = 2/3")
            ex22=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex22)
            print("Exercicio 3")
            print("Volume abaixo da região - Solução 'exata' (precisão dupla) = 0.03330557")
            ex31=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x))])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex31)
            print(" rea da superfície - Solução 'exata' (precisão dupla) = 0.10549788")
```

```
            ex32=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex32)
            print("Exercicio 4")
            print("Volume da calota esférica - Solução 'exata' (precisão dupla) = 0.17998708")
            ex41=np.double([integracao_gaussiana(W6,p6,6,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5 ,lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex41)
            print("Sólido obtido por revolução - Solução 'exata' (precisão dupla) = 3.75824963")
            ex42=np.double([integracao_gaussiana(W6,p6,6,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex42)
        else:
            exit()
    if exercicio==2:
        print("Exercicio 2")
        print("Integral com extremos 0 e 1-x^2 - Solução exata = 2/3")
        ex21=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex21)
        print("Integral com extremos 0 e sqrt(1-y) - Solução exata = 2/3")
        ex22=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex22)
```

```
        continua=int(input("Digite 1 para visualizar a solução dos outros exercícios ou digite 0 para encerrar o código"))
        if continua==1:
            print("Exercicio 1")
            print("Volume do Cubo - Solução exata = 1.")
            ex11=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1,lambda x,y:1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex11)
            print("Volume do tetraedro - Solução exata = 1/6")
            ex12=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex12)
            print("Exercicio 3")
            print("Volume abaixo da região - Solução 'exata' (precisão dupla) = 0.03330557")
            ex31=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x))])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex31)
            print(" rea  da superfície - Solução 'exata' (precisão dupla) = 0.10549788")
            ex32=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex32)
            print("Exercicio 4")
            print("Volume da calota esférica - Solução 'exata' (precisão dupla) = 0.17998708")
            ex41=np.double([integracao_gaussiana(W6,p6,6,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5 ,lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y)])
```

```
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex41)
            print("Sólido obtido por revolução - Solução 'exata' (precisão dupla) = 3.75824963")
            ex42=np.double([integracao_gaussiana(W6,p6,6,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex42)
        else:
            exit()

    if exercicio==3:
        print("Exercicio 3")
        print("Volume abaixo da região - Solução 'exata' (precisão dupla) = 0.03330557")
        ex31=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x))])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex31)
        print(" rea  da superfície - Solução 'exata' (precisão dupla) = 0.10549788")
        ex32=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex32)
        continua=int(input("Digite 1 para visualizar a solução dos outros exercícios ou digite 0 para encerrar o código"))
        if continua==1:
            print("Exercicio 1")
            print("Volume do Cubo - Solução exata = 1.")
            ex11=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1,lambda x,y:1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex11)
```

```
            print("Volume do tetraedro - Solução exata = 1/6")
            ex12=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex12)
            print("Exercicio 2")
            print("Integral com extremos 0 e 1-x^2 - Solução exata = 2/3")
            ex21=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex21)
            print("Integral com extremos 0 e sqrt(1-y) - Solução exata = 2/3")
            ex22=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex22)
            print("Exercicio 4")
            print("Volume da calota esférica - Solução 'exata' (precisão dupla) = 0.17998708")
            ex41=np.double([integracao_gaussiana(W6,p6,6,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5 ,lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex41)
            print("Sólido obtido por revolução - Solução 'exata' (precisão dupla) = 3.75824963")
            ex42=np.double([integracao_gaussiana(W6,p6,6,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex42)
        else:
            exit()
    if exercicio==4:
        print("Exercicio 4")
```

```
    print("Volume da calota esférica - Solução 'exata' (precisão dupla) = 0.17998708")
    ex41=np.double([integracao_gaussiana(W6,p6,6,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5 ,lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y)])
    print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
    print(ex41)
    print("Sólido obtido por revolução - Solução 'exata' (precisão dupla) = 3.75824963")
    ex42=np.double([integracao_gaussiana(W6,p6,6,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y)])
    print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
    print(ex42)
    continua=int(input("Digite 1 para visualizar a solução dos outros exercícios ou digite 0 para encerrar o código"))
    if continua==1:
        print("Exercicio 1")
        print("Volume do Cubo - Solução exata = 1.")
        ex11=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex11)
        print("Volume do tetraedro - Solução exata = 1/6")
        ex12=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex12)
        print("Exercicio 2")
        print("Integral com extremos 0 e 1-x^2 - Solução exata = 2/3")
        ex21=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex21)
        print("Integral com extremos 0 e sqrt(1-y) - Solução exata = 2/3")
```

```
            ex22=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex22)
            print("Exercicio 3")
            print("Volume abaixo da região - Solução 'exata' (precisão dupla) = 0.03330557")
            ex31=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x))])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex31)
            print(" rea  da superfície - Solução 'exata' (precisão dupla) = 0.10549788")
            ex32=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5)])
            print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
            print(ex32)
        else:
            exit()
    else:
        print("Exercicio 1")
        print("Volume do Cubo - Solução exata = 1.")
        ex11=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex11)
        print("Volume do tetraedro - Solução exata = 1/6")
        ex12=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:-x+1,lambda x,y:-x-y+1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex12)
        print("Exercicio 2")
        print("Integral com extremos 0 e 1-x^2 - Solução exata = 2/3")
```

```
        ex21=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:1-x**2,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex21)
        print("Integral com extremos 0 e sqrt(1-y) - Solução exata = 2/3")
        ex22=np.double([integracao_gaussiana(W6,p6,6,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W8,p8,8,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1),integracao_gaussiana(W10,p10,10,0,1,lambda x:0,lambda x:(1-x)**0.5,lambda x,y:1)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex22)
        print("Exercicio 3")
        print("Volume abaixo da região - Solução 'exata' (precisão dupla) = 0.03330557")
        ex31=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x)),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:np.exp(y/x))])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex31)
        print(" rea da superfície - Solução 'exata' (precisão dupla) = 0.10549788")
        ex32=np.double([integracao_gaussiana(W6,p6,6,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W8,p8,8,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5),integracao_gaussiana(W10,p10,10,0.1,0.5,lambda x:x**3,lambda x:x**2,lambda x,y:((-y/(x**2)*np.exp(y/x))**2+(1/x*np.exp(y/x))**2+1)**0.5)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex32)
        print("Exercicio 4")
        print("Volume da calota esférica - Solução 'exata' (precisão dupla) = 0.17998708")
        ex41=np.double([integracao_gaussiana(W6,p6,6,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8,8,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5 ,lambda x,y:2*np.pi*y),integracao_gaussiana(W10,p10,10,0.75,1,lambda x:0,lambda x:(1-x**2)**0.5,lambda x,y:2*np.pi*y)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex41)
        print("Sólido obtido por revolução - Solução exata = 3.75824963")
        ex42=np.double([integracao_gaussiana(W6,p6,6,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),integracao_gaussiana(W8,p8
```

```
    ,8,-1,1,lambda x:0,lambda x:np.exp(-(x**2)),lambda x,y:2*np.pi*y),
    integracao_gaussiana(W10,p10,10,-1,1,lambda x:0,lambda x:np.exp(-(x**2))
    ,lambda x,y:2*np.pi*y)])
        print("Solução numérica p/ n=6, n=8 e n=10 respectivamente:")
        print(ex42)

def integracao_gaussiana(W,p,n,a,b,c,d,f):
    I1=0
    for i in range(n):
        x=((b-a)*p[i]+(a+b))/2
        u=(b-a)*W[i]/2
        for j in range(n):
            v=(d(x)-c(x))*W[j]/2
            y=((d(x)-c(x))*p[j]+(c(x)+d(x)))/2
            I1+=u*v*f(x,y)
    return I1
main()
```