MAP-2220 - BMAC - 2021 Primeiro exercício programa
Fórmulas de Integração Numérica de Gauss e Integrais Duplas

Instruções gerais -

Os exercícios computacionais nesta disciplina têm como objetivo fundamental complementar temas vistos em classe através de problemas práticos que empreguem técnicas numéricas em sua solução. Neste exercício programa sua tarefa será implementar uma técnica de integração numérica (fórmulas de Gauss) e aplicá-la ao cálculo de integrais duplas.

Este programa deve ser entregue no sistema moodle (e-disciplinas) no site da disciplina MAP2220 do BMAC até as 23 horas 59 minutos do dia **21 de dezembro de 2021**.

O programa deve ser escrito em Python, sem utilizar SciPy ou outras bibliotecas de álgebra linear computacional. Podem ser usados Matplotlib, NumPy (apenas para trabalhar com aritmética de vetores, matrizes, leitura ou escrita de dados), bibliotecas básicas auxiliares: sys, time, datetime, os, math.

Ao desenvolver os projetos vocês possivelmente trocarão ideias entre si. Esta interação é mais do que natural, é algo saudável e desejável, vocês estarão aprendendo mais, e ciência hoje é, cada vez mais, fruto de colaboração e de trabalho em equipe. As tarefas propostas devem, no entanto, ser redigidas em grupos de no máximo 3 estudantes. Haverá controle de cópias e caso estas sejam detectadas, todos os envolvidos terão nota zero no programa.

Além de obedecer as condições acima os exercícios programas devem:

- Incluir um arquivo texto LEIAME.txt com instruções de compilação e execução, indicando as bibliotecas usadas.
- Além do arquivo LEIAME.txt, o trabalho entregue deve conter um relatório (em um arquivo .pdf), contendo a análise do problema estudado (veja detalhes no item seguinte), e o código usado para as simulações computacionais (arquivos fonte) feitas. A entrega deve ser feita em um arquivo compactado único.
- O relatório anexado deve apresentar os resultados e a análise das tarefas expostas no enunciado.
- O código entregue deve estar bem documentado, de modo a facilitar a correção. "Rodar"os testes também deve ser simples para um usuário do seu programa, sem que este tenha que editar seu código. Ou seja, o programa deve pedir como entrada qual teste o usuário quer rodar, qual método e quais os parâmetros para essa execução.
- Cada grupo entregará apenas um exercício programa, a(o) estudante do grupo que entregar o arquivo deve destacar no relatório e no código o nome dos membros do grupo.

Bom trabalho!

Fórmulas de Gauss

Uma maneira de obter uma fórmula geral de integração numérica para aproximar $\int_a^b f(x)dx$ de uma função $f : [a,b] \longrightarrow \mathbb{R}$ é escolher pontos $x_j \in [a,b]$ (chamados nós) e pesos $w_j \in \mathbb{R}$ e considerar

$$\int_a^b f(x)dx \approx \sum_{j=1}^{n} \omega_j f(x_j). \tag{1}$$

Os nós e os pesos são escolhidos de acordo com algum critério pré-determinado que leva em conta o erro cometido nessa aproximação.

No caso das chamadas fórmulas de Gauss esse critério será *tornar essa aproximação exata para todos os polinômios de grau menor ou igual a k, com o* ***maior valor de k possível***.

Comece por perceber que com esse objetivo, por certo $k \leq 2n-1$, uma vez que, escolhidos x_j e w_j, a aproximação (1) daria resultado 0 para o polinômio de grau $2n$, $f(x) = (x-x_1)^2(x-x_2)^2 \dots (x-x_n)^2$, cuja integral é estritamente positiva.

Assim, o melhor que pode-se esperar nesse contexto é que (1) *seja exata para todos os polinômios de grau menor ou igual a* $2n-1$.

Para analisar se isso é possível, denote por $E_n(f)$ o erro cometido pela aproximação (1), i.e.

$$\int_a^b f(x)\,dx = \sum_{j=1}^{n} \omega_j f(x_j) + E_n(f) \tag{2}$$

e perceba que o objetivo pretendido é que $E_n(f) = 0$ para todo polinômio de grau menor ou igual a $2n-1$.

Caso seja possível escolher nós e pesos assim, então $E_n(f_k) = 0$, para toda $f_k(x) = x^k$, com $0 \leq k \leq 2n-1$.

Ao impor em (2) que $E_n(f_k) = 0$, se $0 \leq k \leq 2n-1$, obtém–se um sistema de $2n$ equações (algébricas, mas não lineares) nas incógnitas x_j e w_j, $1 \leq j \leq n$. Com "um pouco de sorte"esse sistema terá solução, com mais sorte ainda os valores x_j obtidos estão em $[a,b]$ e nesse caso... o mundo continuará redondo:

> **Exercício** Prove que, se $E_n(f_k) = 0$ para todo $k \in \{0, \dots, 2n-1\}$, então $E_n(p) = 0$ para qualquer polinômio de grau menor ou igual a $2n-1$.

A pergunta que resta é... a "sorte"supramencionada acontece ou não?

Isso será analisado em etapas, a primeira é a questão da existência dos nós.

O ponto de partida aqui é uma observação simples, se os nós e os pesos foram escolhidos de forma a $E_n(p) = 0$ para todo polinômio $p(x)$ de grau menor ou igual a $2n-1$, então

$$\int_a^b (x-x_1)(x-x_2)\dots(x-x_n)q(x)\,dx = 0, \tag{3}$$

se $q(x)$ é um polinômio de grau menor ou igual a $n-1$ (basta aplicar (2) ao polinômio $s(x) = (x-x_1)(x-x_2)\dots(x-x_n)q(x)$ que tem grau menor ou igual a $2n-1$).

Ou seja, $x(x) = (x - x_1)(x - x_2) \dots (x - x_n)$ tem de ser ortogonal a qualquer polinômio $q(x)$ de grau menor ou igual a $n - 1$ em relação ao produto interno

$$< u, v >= \int_a^b u(x)v(x)\, dx. \tag{4}$$

Uma pausa para relembrar algo sobre ortogonalidade…

Existe uma *única* família $\mathfrak{F} = \{\widetilde{p}_j(x),\, 0 \le j \le n\}$ de polinômios mônicos, em que o grau de $\widetilde{p}_j(x)$ é j e que são dois a dois ortogonais em relação a (4).

Note que a família $\mathfrak{F}$ depende apenas do intervalo $[a, b]$ e do produto interno considerado. Assim, uma vez fixado o intervalo $[a, b]$, as raízes de cada $\widetilde{p}_j$ de $\mathfrak{F}$ ficam determinadas.

Por outro lado, o polinômio $s(x) = (x - x_1)(x - x_2) \dots (x - x_n)$ considerado antes é mônico, portanto, conclui–se de (3) e da discussão anterior que este polinômio tem de ser igual a $\widetilde{p}_n(x)$.

Música maestro! Acabou-se de provar que

> **Fato** *Se os nós e os pesos usados em* (2) *são tais que* $E_n(f) = 0$ *para todos os polinômios* $f(x)$ *de grau menor ou igual a* $2n - 1$ *então os nós têm de ser as raízes do polinômio* $\widetilde{p}_n(x)$ *da família* $\mathfrak{F}$ *descrita acima.*

Ainda não se conhece a resposta sobre a existência ou não de nós que façam a mágica pretendida, mas uma coisa foi estabelecida, *se existirem tais nós eles têm de ser as raízes de* $\widetilde{p}_n$.

Antes de passar á análise da existência dos pesos, vai–se ver que esses "candidatos a nós"são pontos de $[a, b]$. Essa parte fica para você provar!

> **Exercício** Seja $\{p_k\}$ uma família de polinômios ortogonais em relação ao produto interno (4), com grau de p_k igual a k. Prove que para todo $n \ge 1$, p_n tem n raízes reais distintas contidas no intervalo aberto (a, b). Se $\{q_k\}$ for outra família de polinômios ortogonais em relação ao mesmo produto interno, então q_n tem as mesmas raízes que p_n.

Agora estuda-se a questão da *existência dos pesos*, para isso sejam $x_1, \dots, x_n$ as raízes de $\widetilde{p}_n(x)$ e considere os polinômios de Lagrange

$$L_i(x) = \prod_{\substack{k=1 \\ k \ne i}}^{n} \frac{x - x_k}{x_i - x_k}, \quad 1 \le i \le n.$$

Quer-se que $E_n(L_i) = 0$, se $1 \le i \le n$, então, de (2) resulta que, necessariamente,

$$\omega_i = \int_a^b L_i(x)\, dx,\ 1 \le i \le n. \tag{5}$$

Qual o estado da arte neste ponto?

Bem… ainda não é conhecido se existem pesos e nós para os quais $E_n(p) = 0$ sempre que $p(x)$ for um polinômio de grau menor ou igual a $2n - 1$, mas já se sabe que *se existirem pesos e nós assim então os nós têm de ser as raízes de* $\widetilde{p}_n(x)$ *e os pesos têm de ser os determinados por* (5).

Em outras palavras, conhecem-se condições necessárias para a existência de pesos e nós, serão elas suficientes?

Como foi visto em interpolação, os polinômios de Lagrange $L_i(x),\ 1 \leq i \leq n$ formam uma base do espaço vetorial de todos os polinômios de grau menor ou igual a n, então os nós e os pesos foram escolhidos para $E_n(L_i) = 0$, segue-se de pronto que $E_n(g) = 0$, *para todo polinômuio* $g(x)$ *de grau menor os igual a* n. (por que?)

Com isso pode-se pedir outra vez ao maestro que comande a banda...

> **Proposição** *Se* $x_1, \ldots, x_n$ *são as raízes de* $\widetilde{p}_n(x)$ *e os pesos* $w_1, \ldots, w_n$ *são os definidos em* (5) *então* $E_n(f) = 0$ *para todos os polinômios de grau menor ou igual a* $2n - 1$.

> **Demonstração** Seja $f(x)$ um polinômio de grau menor ou igual a $2n - 1$ e considere a divisão de $f(x)$ por $\widetilde{p}_n(x) = (x - x_1)(x - x_2) \ldots (x - x_n)$ para obter polinômios $q(x)$ e $r(x)$ tais que $f(x) = \widetilde{p}_n(x)q(x) + r(x)$ e com o grau de $r(x)$ estritamente menor do que n.
> Além disso, claro que o grau de $q(x)$ é também menor ou igual a $n - 1$, portanto $\widetilde{p}_n(x)$ e $q(x)$ são ortogonais em relação a (4).
> Então $\int_a^b f(x)dx = \int_a^b r(x)dx$ e assim $E_n(f) = E_n(r)$. Como o grau de $r(x)$ é menor do que n, já foi visto que $E_n(r) = 0$. $\square$

Resumo da ópera, *existe uma única fórmula de integração numérica do tipo (2) que é exata para polinômios de grau menor ou igual a* $2n - 1$.

O que se espera com esse tipo de fórmula é obter aproximações "boas"para integrais de funções suaves usando uma quantidade pequena de nós, há boas razões para isso, mas esse ponto não será abordado aqui.

Polinômios de Legendre e Mudança de Variável

Algumas famílias de polinômios ortogonais são bem conhecidas e as raízes e pesos podem ser encontradas em tabelas[1]. Em particular, a partir dos polinômios de Legendre podemos obter fómulas de integração numérica no intervalo $[-1, 1]$. Por exemplo, $p_3(x) = \frac{1}{2}(5x^3 - 3x)$ é o polinômio de Legendre de grau 3. As raízes são $x_1 = -\sqrt{0.6}$, $x_2 = 0$ e $x_3 = \sqrt{0.6}$, com os respectivos pesos $\omega_1 = 5/9$, $\omega_2 = 8/9$ e $\omega_3 = 5/9$. Portanto, a fórmula de integração

$$\int_{-1}^{1} f(x)\, dx \approx \frac{5}{9} f(-\sqrt{0.6}) + \frac{8}{9} f(0) + \frac{5}{9} f(\sqrt{0.6})$$

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

Outros valores de n e seus respectivos nós e pesos podem ser usados. Devido à paridade, as raízes dos polinômios de Legendre são simétricas em torno da

[1]Veja por exemplo a página https://dlmf.nist.gov/

origem, e cada par $-x_j$ e x_j de raízes simétricas tem o mesmo peso ω_j. Note que conhecendo as fórmulas de Gauss para o intervalo $[-1, 1]$, podemos obter as fórmulas de Gauss em qualquer intervalo $[a, b]$ usando mudança de variável. Os nós são linearmente transportados do intervalo $[-1, 1]$ para o intervalo $[a, b]$ e os pesos são multiplicados por um fator de escala (qual?).

Integrais Duplas

Deseja-se calcular a integral

$$I = \iint_R f(x, y)\, dx\, dy$$

em uma região R do plano. Vamos supor que R é da forma

$$\{(x, y) \,|\, a \leq x \leq b, c(x) \leq y \leq d(x)\} \text{ ou } \{(x, y) \,|\, c \leq y \leq d, a(y) \leq x \leq b(y)\}$$

o que permite o cálculo da integral por fórmulas iteradas para uma ampla classe de funções f (por exemplo, para funções contínuas).

Assim, no primeiro caso,

$$I = \int_a^b \left[\int_{c(x)}^{d(x)} f(x, y)\, dy \right] dx$$

Neste ponto serão usadas as fórmulas de Gauss com n nós vistas antes.

Para isso considere $F(x) = \int_{c(x)}^{d(x)} f(x, y) dy,\ x \in [a, b]$, então $I = \int_a^b F(x) dx$ e aproxime esta integral pela fómula de Gauss com n nós, ou seja

$$I \approx \sum_{i=1}^{n} u_i F(x_i).$$

Note que os nós x_i e os pesos u_i são conhecidos, eles dependem apenas do intervalo $[a, b]$, e para obter a aproximação desejada para a integral dupla basta conhecer $F(x_i) = \int_{c(x_i)}^{d(x_i)} f(x_i, y) dy$.

Aqui será usada, mais una vez..., a parafernália das fórmulas de Gauss para obter uma aproximação desses valores $F(x_i)$, de forma mais precisa,

$$F(x_i) \approx \sum_{j=1}^{n} v_{ij} f(x_i, y_{ij}),$$

onde y_{ij} e v_{ij} são os nós e os pesos nos intervalos $[c(x_i), d(x_i)]$.

O segundo caso pode ser deduzido de maneira análoga.

Tarefa

Implemente um programa em Python para o cálculo de integrais duplas em regiões R do plano, como as descritas acima, por fórmulas iteradas. Você deverá usar fórmulas de Gauss com n nós para as integrações numéricas. Os nós e os pesos são fornecidos para o intervalo $[-1, 1]$ e o programa deverá fazer os ajustes necessários para outros intervalos. *Use precisão dupla.*

Teste o seu programa nos exemplos abaixo usando fórmulas de Gauss com $n = 6$, 8 e 10. Os nós e os pesos estão listados ao final deste enunciado (devido à simetria já mencionada, são dados apenas os nós maiores ou iguais a zero).

Exemplo 1 Calcule os volumes do cubo cujas arestas têm comprimento 1 e do tetraedro com vértices $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$ e $(0, 0, 1)$. Você deve obter resultados exatos, exceto por erros de arredondamento (por que?).

Exemplo 2 A área A da região no primeiro quadrante limitada pelos eixos e pela curva $y = 1 - x^2$ pode ser obtida por

$$A = \int_0^1 \left[\int_0^{1-x^2} dy \right] dx = \int_0^1 \left[\int_0^{\sqrt{1-y}} dx \right] dy = \frac{2}{3}$$

Calcule numericamente as duas integrais duplas acima e observe os resultados.

Exemplo 3 Considere a superfície descrita por $z = e^{y/x}$, $0.1 \leq x \leq 0.5$, $x^3 \leq y \leq x^2$. Calcule a sua área e o volume da região abaixo dela.
Obs. A área de uma superfície de equação $z = f(x, y)$, $(x, y) \in R$ é

$$V = \iint_R \sqrt{f_x(x, y)^2 + f_y(x, y)^2 + 1}\, dx\, dy.$$

Exemplo 4 Considere uma região fechada R do plano xy e seja γ uma reta no mesmo plano, que não intercepta o interior de R.
O volume V do sólido de revolução obtido pela rotação da região R em torno de γ é

$$V = 2\pi \iint_R d_\gamma(x, y)\, dx\, dy$$

onde $d_\gamma(x, y)$ é a distância do ponto (x, y) à reta γ. Use esta expressão para calcular o volume da calota esférica de altura $\frac{1}{4}$ da esfera de raio 1, e o volume do sólido de revolução obtido da rotação da região delimitada por $x = 0$, $x = e^{-y^2}$, $y = -1$ e $y = 1$, em torno do eixo y.

Seu programa deve exibir, para cada n, este valor de n e os valores calculados das integrais. Quando for o caso, o programa deve mostrar também, em cada exemplo, os valores exatos.

DADOS PARA FÓRMULAS DE GAUSS

$n = 6$

x_j	w_j
0.238619186083196908630501 7	0.467913934572691047389870 3
0.661209386466264513661399 6	0.360761573048138607569833 5
0.932469514203152027812301 6	0.171324492379170345040296 1

$n = 8$

x_j	w_j
0.183434642495649804939476 1	0.362683783378361982965150 4
0.525532409916328985817739 0	0.313706645877887287337962 2
0.796666477413626739591553 9	0.222381034453374470544356 0
0.960289856497536231683560 9	0.101228536290376259152531 4

$n = 10$

x_j	w_j
0.148874338981631210884826 0	0.295524224714752870173893 0
0.433395394129247190799265 9	0.269266719309996355091226 9
0.679409568299024406234327 4	0.219086362515982043995534 9
0.865063366688984510732096 7	0.149451349150580593145776 3
0.973906528517171720077964 0	0.066671344308688137593568 8