Cálculo Numérico 1

Roberto F. Ausas

rfausas@icmc.usp.br
`www.lmacc.icmc.usp.br/~ausas/`

- Antes de começar a realizar o tarefa se recomenda estudar com as Jupyter Notebook apresentadas pelo professor.
- A tarefa e o relatório serão feitos em grupo (máximo 3 integrantes).
- O relatório será feito na própria Jupyter Notebook desenvolvida com algumas explicações e os resultados obtidos ao rodar.
- NÃO ENTREGRAR ARQUIVOS .zip OU QUALQUER OUTRO FORMATO QUE NÃO SEJA O DA JUPYTER NOTEBOOK .ipynb, POIS SERÃO DESCONSIDERADOS.
- **Todos os exercícios devem estar no mesmo arquivo e as células devem ter sido executadas para que o professor possa ver os resultados.**
- Cada aluno deverá colocar o relatório no escaninho.
- Na jupyter notebook deverá constar o nome de todos os participantes.
- A data de entrega será até às 6am do dia 12/07/2023 no escaninho do Tidia.

Parte 1: Problemas não lineares

Exo. A. Considere o método de Newton para achar o zero $\bar{x}$ de uma função.

(a) Fazer uma função geral de python que implemente o método da Newton, a qual recebe entre os seus argumentos o nome da função cujo zero será determinado, uma semente inicial x_0, a tolerância e o máximo número de iterações.

(b) Implementar como opções que a derivada da função possa ser fornecida de duas formas:

- Calculando na mão a derivada analítica que é programada pelo usuario numa outra função a ser fornecida.
- Calculando uma aproximação da derivada dada por diferenciação numérica, i.e.

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

em que h é um parâmetro algoritmico a ser fornecido pelo usuário. Testar diferentes valores para h (p.e., $h = 10^{-3},\ 10^{-4},\ 10^{-5}$).

(c) Pensar um exemplo e testar a implementação.

(d) Comparar os resultados com a função `fsolve` de python. Notar que neste caso, a derivada da função também pode ser fornecida pelo usuario ou deixar que a função `fsolve` a calcule.

Exo. B. Considerar as redes hidráulicas estudadas na lista anterior, para o qual foi desenvolvida a função `ResolveRede()` e outras funções que calculavam a vazão nos canos e a potência consumida pela bomba. Considerar uma rede com os seguintes parâmetros:

- `n = 8`

- `m = 9`
- `QB = 3`
- `natm = n*m`
- `nB = 0`
- As condutâncias dependem de um parâmetro x o qual pretende-se ajustar. Considerar os casos:

$$\begin{aligned} CH &= 2.3 + 0.1\,(x-1)^2, \\ CV &= 1.8 + 0.2\,(x-1)^2 \end{aligned}$$

ou

$$\begin{aligned} CH &= 2.3 + 10\,e^{-(x-5)^2}, \\ CV &= 1.8 + 10\,e^{-(x-5)^2} \end{aligned}$$

Por claridade e eficiência, recomenda-se implementar o código da seguinte forma:

- Criar a rede com a função `GeraRede()`
- Fazer uma função `ResolveRede()` que recebe todos os argumentos necessários: (`conec, QB, nB, natm`, etc) e também o parâmetro `x` no qual pretende-se avaliar as condutâncias.
- Avaliar `CH` e `CV` como função de `x` e definir o vetor de condutâncias `C` que irá ser finalmente usado na função `Assembly` e no resto dos cálculos.
- Plotar a potência dissipada como função do parâmetro x e determinar visualmente para que valor de x a potencia é igual a 6.
- Aplicar as funções desenvolvidas nos exercícios anteriores e a função `fsolve`.

Parte 2: Sistemas sobredeterminados

1. (Teórico) Na prova do teorema das equações normais, abrir todas as contas para provar que:

$$\begin{aligned} \|Ax - b\|_G^2 &= (Ax - b)^T G (Ax - b) \\ &= (Ax^* + Ad - b)^T G (Ax^* + Ad - b) \\ &= \|Ax^* - b\|_G^2 + 2d^T (A^T G A x^* - A^T G b) + \|Ad\|_G^2 \quad \square \end{aligned}$$

   sendo $x = x^* + d$ com $d \in \mathbb{R}^n$ arbitrário. Pode supor que $G = I_{m\times m}$ para simplificar.

2. (Teórico) Dada uma base ortogonal de col(A) formada pelos vetores $\{\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \ldots, \mathbf{q}^{(r)}\}$, provar que, na projeção ortogonal de b sobre col(A), definida como:

$$b^{\perp} = \mathbb{P}b = \sum_{i=1}^{n} (\mathbf{q}^{(i)\intercal} b)\, \mathbf{q}^{(i)}$$

   o operador $\mathbb{P}$ é dado por $Q_1 Q_1^\intercal$, em que a matriz Q_1 é formada pelos vetores $\mathbf{q}^{(i)}$.

3. Considerar a fatoração QR de uma matriz A, e Q_1 a submatriz de Q formada pelas primeras n colunas. Provar que $Q_1^\intercal Q_1 = I_{n\times n}$.

4. Considerar o código fornecido na jupyter notebook `Grads.ipynb` que implementa o método da máxima descida.

- Considar o caso em que `sampling_stoch = 1`. Neste caso, é usada a matriz completa A do sistema sobredeterminado. Calcular a solução do problema e comparar com a solução das equações normais no caso do produto escalar usual, i.e.,

$$A^{\intercal} A x = A^{\intercal} b$$

  Fazer um gráfico da "descida", i.e., do processo iterativo convergindo na solução até atingir a tolerânci `TOLr = 1e-8`.

- Considerar o caso estocástico, tomando `sampling_stoch > 1`. Fazer gráficos mostrando a "descida" até converger na solução variando o valor de `sampling_stoch` e tomando uma tolerância `TOLr = 1e-2`. Tomar valores `100, 1000, 10000, 100000`. Fazer um gráfico que mostre quantas iterações o processo demora como função de `sampling_stoch`.

Parte 3: Interpolação, Integração e diferenciação numérica

Considerar as redes hidráulicas estudadas na lista anterior, para o qual foi desenvolvida a função `ResolveRede()` e a função que calcula a potência consumida pela bomba. Considerar uma rede com os seguintes parâmetros:

- `n = 8`
- `m = 9`
- `QB = 3`
- `natm = n*m`
- `nB = 0`

- As condutâncias dependem de um parâmetro x:

$$
\begin{aligned}
CH &= 2.3 + 10\, e^{-(x-5)^2}, \\
CV &= 1.8 + 10\, e^{-(x-5)^2}
\end{aligned}
$$

(a) **Interpolação global:** Fazer um script que avalia a potência em 10 valores de x em $[1, 10]$ e calcula o polinômio interpolador de grau 9. Plotar no mesmo gráfico o polinômio avaliado em 100 pontos no intervalo $[1, 10]$, a potência avaliada nesses mesmos pontos e os 10 pontos usados na interpolação. Fazer os cálculos usando a matriz de Vandermonde.

(b) **Interpolação por partes: Lineares e Cúbicas (splines):** Ler o Cap. 3, pag. 100 do livro de Quarteroni sobre interpolação de splines por partes. Pesquisar o uso da função `interp1d` de `scipy.interpolate` e aplicá-la para resolver o problema (ver exemplo fornecido na jupyter notebook disponibilizada). Comparar com a interpolação polinomial (global) do item anterior realizando gráficos nos quais se mostram os pontos considerados e as aproximações obtidas.

(b) **Integração:** Integrar a função da potencia no intervalo $[1, 10]$ utilizando os métodos utilizados considerandos diferentes partições e/ou número de pontos nas regras de quadratura.

(c) **Diferenciação:** Diferenciar a função da potencia no intervalo $[1, 10]$ e mostrar os resultados em gráficos, considerando as fórmulas estudadas e diferentes valores para o parâmetro h.

Parte 4: Autovaloes-Autovetores e Resolução de EDOs

Exercício A: Provar que $W(t)$ é solução do sistema de EDOs,

$$
M\, \frac{d^2W}{dt^2} + \, K\, W = 0
$$

inserindo

$$W(t) = \sum_{k=1}^{d} \Phi^{(k)}\, c_k\, \sin(\omega_k t + \phi_k)$$

no lado esquerdo da equação, i.e.,

$$M\,\frac{d^2W}{dt^2} + \,K\,W$$

e provar que de fato isso é zero, devido a que os vetores $\Phi^{(k)}$ e as frequências ω_k, são solução do problema de autovalores generalizado

$$K\,\Phi^{(k)} = \omega_k^2\, M\,\Phi^{(k)}$$

Finalmente, determinar os valores das constantes c_k e ϕ_k a partir das condições iniciais U_0 e V_0.

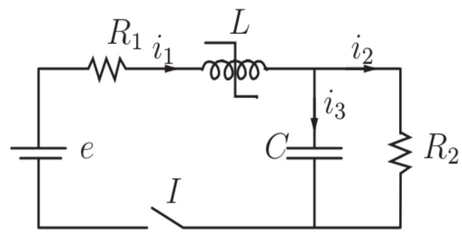
Lembrar que os autovetores $\Phi^{(i)}$ cumprem $(\Phi^{(j)})^T M\,\Phi^{(i)} = \delta_{ij}$ (i.e., é igual a 0 se $i \neq j$ e é igual a 1 em caso contrário.

Exercício B: Mudar a função `BuildMatrizes()` para calcular o modos de oscilação de uma membrana de forma triangular. A densidade ρ do material da membrana e a espessura podem ser tomadas igual a 1. Calcular os valores das primeiras 3 frequências considerado diferentes discretizações (p.e, $21 \times 21, 41 \times 41$ e 61×61). Comparar os resultados numa tabela.

Exercício C: EDOs

1. Considerar a equação de segunda ordem para o circuito da figura (apresentado no livro de Quarteroni & Salieri):

$$L\,C\,\frac{d^2v}{dt^2} + \left(\frac{L}{R_2} + R_1\,C\right)\,\frac{dv}{dt} + \left(\frac{R_1}{R_2} + 1\right)\,v = e$$



em que $v(t)$ é a diferença de potencial no capacitor. No tempo $t = 0$ o interruptor é fechado e consideramos a condição inicial $v(0) = 0$ e $v'(0) = 0$. Os parâmetros são $L = 0.1$Hy, $C = 10^{-3}$F , $R_1 = R_2 = 10$ Ohm $e = 5$ V.

(a) Transformar a equação de segunda ordem em duas equações de primeira ordem.

(b) Escrever o sistema com notação vetorial.

(c) Escrever detalhadamente, componente por componente, como resulta o esquema numérico para o caso de Euler explícito.

(d) Resolver usando um paso de tempo $h = 0.001$ e plotar $v(t)$ e a sua derivada $v'(t)$.

2. Considerar o sistema de n equações de primeira ordem:

$$\begin{cases} \mathbf{y}' = \mathbf{f}(\mathbf{y}) \\ \mathbf{y}(0) = \mathbf{y}_0 \end{cases}$$

em que $\mathbf{f}(\mathbf{y})$ é uma função linear, i.e.

$$\mathbf{f}(\mathbf{y}) = A\,\mathbf{y}, \quad A \in \mathbb{R}^{n \times n}$$

Aplicar o método α:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\, f\left(\alpha \mathbf{y}_{n+1} + (1-\alpha)\mathbf{y}_n\right), \quad \alpha \in [0,1]$$

e demostrar que o esquema numérico pode ser escrito como:

$$\mathbf{y}_{n+1} = B\, \mathbf{y}_n$$

em que $B \in \mathbb{R}^{n\times n}$ é a matriz

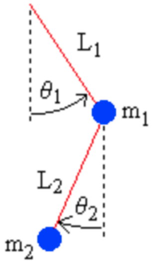
$$B = \left[\mathbb{I} - h\, \alpha\, A\right]^{-1} \left[\mathbb{I} + h\,(1-\alpha)\, A\right]$$

sendo $\mathbb{I}$ a matriz identidade de $n \times n$.
Demostrar que:

$$\mathbf{y}_{n+1} = B^{n+1}\, \mathbf{y}_0$$

3. Resolver numericamente o problema do pêndulo duplo ilustrado na figura usando a função `scipy.integrate.solve_ivp`. O método padrão ou *default* que é usado é o RK45, que é um método de Runge-Kutta que adapta o passo de tempo. As equações são:

   Resolver considerando diferentes condições iniciais e valores de m_1 e m_2 (p.e., $m_1 \approx m_2$, $m_1 \gg m_2$, $m_2 \gg m_1$). Fazer animações para ilustrar os resultados.



$$\theta_1' = \omega_1$$

$$\theta_2' = \omega_2$$

$$\omega_1' = \frac{-g\,(2\,m_1 + m_2)\sin\theta_1 - m_2\,g\sin(\theta_1 - 2\,\theta_2) - 2\sin(\theta_1 - \theta_2)\,m_2\,(\omega_2{}^2\,L_2 + \omega_1{}^2\,L_1\cos(\theta_1 - \theta_2))}{L_1\,(2\,m_1 + m_2 - m_2\cos(2\,\theta_1 - 2\,\theta_2))}$$

$$\omega_2' = \frac{2\sin(\theta_1 - \theta_2)\,(\omega_1{}^2\,L_1\,(m_1 + m_2) + g(m_1 + m_2)\cos\theta_1 + \omega_2{}^2\,L_2\,m_2\cos(\theta_1 - \theta_2))}{L_2\,(2\,m_1 + m_2 - m_2\cos(2\,\theta_1 - 2\,\theta_2))}$$