Trabalho 1

Cálculo Numérico (SME0104)
Professora Cynthia Lage Ferreira

18 de abril de 2024

Orientações Gerais

- Esta avaliação é **individual ou em dupla** e deverá ser desenvolvida na plataforma Colab (https://colab.research.google.com/).
- Cada aluno/dupla deverá produzir um **arquivo .ipynb** contendo tanto a parte escrita (teórica) quanto a parte prática (códigos em Python) de cada um dos exercícios.
- Os arquivos deverão estar identificados da seguinte forma: **NOMEDOALUNO1+NOMEDOALUNO2.ipynb** a fim de facilitar a organização das atividades pela professora. Enviem apenas um arquivo por dupla.
- Os arquivos deverão ser **enviados até às 20h do dia 21/04/2024** através da plataforma e-disciplinas da USP (https://edisciplinas.usp.br/). **Os arquivos recebidos fora do prazo ou por e-mail não serão corrigidos.**
- Apenas os alunos que estiverem com a **situação regularizada no Sistema Jupiter** terão suas avaliações corrigidas.
- Todos os exercícios deverão conter justificativas teóricas e todos os códigos utilizados para resolver os problemas deverão ser apresentados, executados e minimamente comentados.
  Questões com respostas sem justificativas não serão consideradas.
- Os alunos que quiserem poderão apresentar um resumo teórico referente aos conteúdos de cada questão. A realização desta tarefa poderá gerar uma bonificação ao aluno, a critério da professora.
- As funções prontas do Python dos métodos estudados poderão ser utilizadas para validar os resultados obtidos, mas não as utilize como ÚNICA forma de solução dos exercícios.

1 Sistemas Lineares - métodos diretos

Discuta, detalhadamente, as diferenças entre as funções *func1* e *func2* apresentadas abaixo. Comente os códigos, os resultados obtidos e apresente as suas conclusões a partir da aplicação destas duas funções no exemplo abaixo.

```
import numpy as np
import time

def func1( A ):

    n = A.shape[ 0 ]
    U = A.copy()
    L = np.eye( n )

    for j in range( n - 1 ):
        for i in range( j + 1, n ):
            L[ i, j ] = U[ i, j ] / U[ j, j ]
            U[ i, j : n ] = U[ i, j : n ] - L[ i, j ] * U[ j, j : n ]
    return ( L, U )


def func2( A, p ):

    n = A.shape[ 0 ]
    U = A.copy()
    L = np.eye( n )

    for j in range( n - 1 ):
        v = min( n, j + p + 1 )
        for i in range( j + 1, v ):
            L[ i, j ] = U[ i, j ] / U[ j, j ]
            U[ i, j : v ] = U[ i, j : v ] - L[ i, j ] * U[ j, j : v ]
    return ( L, U )


#Exemplo

n = 2000
p = 2
A = np.zeros( ( n, n ) )
for i in range( n ):
    for j in range( max( 0, i - p ), min( n, i + p + 1 ) ):
        A[ i, j ] = np.random.normal()


start_time = time.time()
( L, U ) = func1( A )
end_time = time.time()
print( end_time - start_time )


start_time = time.time()
( L_, U_ ) = func2( A, p )
```

```
end_time = time.time()
print( end_time - start_time )


print( np.linalg.norm( L @ U - A ) )
print( np.linalg.norm( L_ @ U_ - A ) )
```

2 Sistems Lineares - métodos iterativos

Dada a matriz esparsa

$$A = \begin{bmatrix} 4 & -2 & 1 & 0 & 0 & 0 \\ -2 & 7 & -2 & 1 & 0 & 0 \\ 1 & -2 & 7 & -2 & 1 & 0 \\ 0 & 1 & -2 & 7 & -2 & 1 \\ 0 & 0 & 1 & -2 & 7 & -2 \\ 0 & 0 & 0 & 1 & -2 & 4 \end{bmatrix},$$

considere o sistema $Ax = b$, em que $\mathbf{b} = [-1, -2, 1, 1, -2, -1]^T$.

a) Um método iterativo pode ser escrito na forma

$$x^{(k+1)} = Cx^{(k)} + g,\ k \geq 0.$$

Escreva as matrizes de iteração C_J e C_{GS} e os vetores g_J e g_{GS} dos métodos de Gauss-Jacobi e Gauss-Seidel, respectivamente.

b) Verifique se os métodos de Gauss-Jacobi e Gauss-Seidel obtidos no item anterior convergem.

c) Resolva numericamente o sistema $Ax = b$ em questão usando os métodos de Gauss-Jacobi e Gauss-Seidel utilizando erro absoluto inferior a $1e^{-8}$ e chute inicial $x0 = (1, 0, 0, 0, 0, 0)$. Discuta os resultados obtidos. Faça um gráfico erro X iterações.

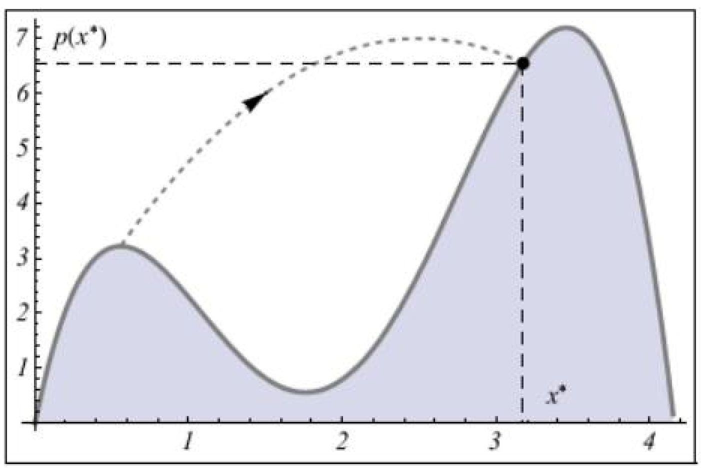
3 Zeros de funções e sistemas não lineares

A região sombreada do gráfico apresentado a seguir representa o perfil de duas elevações dado pela função $p(x) = -x^4 + 7.7x^3 – 18x^2 + 13.6x$. Um projétil é lançado a partir da menor elevação e descreve uma curva dada por $q(x) = -x^2 + 5x + 0.75$. Pede-se determinar a altura na qual ocorre o impacto com a maior elevação.

a) Formule o problema de modo que sua solução seja uma raiz de uma função não linear $f : \Re \to \Re$. Use o método da bisseção com precisão 0.001 e até 5 iterações para aproximar esta raiz e, consequentemente, a altura na qual ocorre o impacto.

b) Formule este problema de modo que sua solução seja uma raiz de função não linear $F : \Re^2 \to \Re^2$. Use o método de Newton para sistemas com precisão 0.001 para aproximar esta raiz e, consequentemente, a altura na qual ocorre o impacto.

Comente as soluções detalhadamente, apresentando os códigos utilizados, os critérios de parada e comparações entre os dois resultados.



$p(x^*)$
x^*
1
2
3
4
5
6
7