	PROFESSOR: Paulo César Linhares da Silva		
	CURSO: CIÊNCIAS DA COMPUTAÇÃO		
	DISCIPLINA: CÁLCULO NUMÉRICO		
	PERÍODO: MANHÃ. HORÁRIO: 07:00 ÀS 08:40		DATA: 05/12/2024
	NOME:		PONTOS OBTIDOS
	SEMESTRE SUPLEMENTAR: 2024		
2ªAVALIAÇÃO DE CÁLCULO NUMÉRICO	Sistemas Lineares	NOTA OBTIDA:	

Cálculo Numérico

1ª Questão: Implemente a decomposição LU de uma matriz A de ordem n por n baseando-se no método de Crount. Utilize esta decomposição para resolver um sistema linear do tipo $A_{nxn}x_{nx1} = b_{nx1}$.

Observação: No método de Crount, os elementos da diagonal principal da matriz U são todos iguais a 1.

2ª Questão: Implemente usando a linguagem de programação Python o algoritmo a seguir.

ALGORITMO 1: Resolução de um Sistema Triangular Superior

Dado um sistema triangular superior n × n com elementos da diagonal da matriz A não nulos, as variáveis $x_n, x_{n-1}, x_{n-2}, \dots x_2, x_1$ são assim obtidas:

$x_n = b_n / a_{nn}$

Para k = (n − 1),..., 1

$$\left[\begin{array}{l} s = 0 \\ \text{Para } j = (k+1), \dots, n \\ s = s + a_{kj}x_j \\ x_k = (b_k - s) / a_{kk} \end{array}\right.$$

3ª Questão: Implemente usando a linguagem de programação Python o algoritmo a seguir.

ALGORITMO 2: Resolução de Ax = b através da Eliminação de Gauss.

Seja o sistema linear $Ax = b$, $A: n \times n$, $x: n \times 1$, $b: n \times 1$.

Supor que o elemento que está na posição a_{kk} é diferente de zero no início da etapa k.

Eliminação

$$
\left[\begin{array}{l}
\text{Para } k = 1, \dots, n-1 \\
\quad \left[\begin{array}{l}
\text{Para } i = k+1, \dots, n \\
\qquad m = \dfrac{a_{ik}}{a_{kk}} \\
\qquad a_{ik} = 0 \\
\qquad \text{Para } j = k+1, \dots n \\
\qquad a_{ij} = a_{ij} - m a_{kj} \\
\qquad b_i = b_i - m b_k
\end{array}\right.
\end{array}\right.
$$

Resolução do sistema:

$$
\left[\begin{array}{l}
x_n = b_n / a_{nn} \\
\text{Para } k = (n-1), \dots\ 2,1 \\
\quad \left[\begin{array}{l}
s = 0 \\
\text{Para } j = (k+1), \dots, n \\
[s = s + a_{kj} x_j \\
x_k = (b_k - s) / a_{kk}
\end{array}\right.
\end{array}\right.
$$

4ª Questão: Implemente o método de Gauss- Jacobi a partir do algoritmo (código em Python) sugerido no SIGAA. Verifique os critérios de convergência e se a solução é apropriada para o sistema.

5ª Questão: Implemente o método de Gauss- Seidel a partir do algoritmo (código em Python) sugerido no SIGAA. Verifique os critérios de convergência e se a solução é apropriada para o sistema.

6ª Questão: Resolva o projeto proposto a seguir.

PROJETO

a) Compare as soluções dos sistemas lineares

$$\begin{cases} x - \, y = 1 \\ x - 1.00001\, y = 0 \end{cases} \quad \text{e} \quad \begin{cases} x - \, y = 1 \\ x - 0.99999\, y = 0 \end{cases}$$

Fatos como este ocorrem quando a matriz A do sistema está próxima de uma matriz singular e então o sistema é *mal condicionado.*

Dizemos que um sistema linear é *bem condicionado* se pequenas mudanças nos coeficientes e/ou nos termos independentes acarretarem pequenas mudanças na solução do sistema. Caso contrário, o sistema é dito mal condicionado.

Embora saibamos que uma matriz A pertence ao conjunto das matrizes não inversíveis se, e somente se, $\det(A) = 0$, o fato de uma matriz A ter $\det(A) \approx 0$ não implica necessariamente que o sistema linear que tem A por matriz de coeficientes seja mal-condicionado.

O *número de condição de A*, $\text{cond}(A) = \| A \| \, \| A^{-1} \|$, onde $\| \, . \, \|$ é uma norma de matrizes [14], é uma medida precisa do bom ou mau condicionamento do sistema que tem A por matriz de coeficientes, pois demonstra-se que

$$\frac{1}{\text{cond}(A)} = \min \left\{ \frac{\| A - B \|}{\| A \|} \text{ tais que B é não inversível} \right\}.$$

b) As matrizes de Hilbert, H_n, onde

$h_{ij} = \dfrac{1}{i + j - 1}$, $1 \leq i, j \leq n$ são exemplos clássicos de matrizes mal condicionadas.

(*b.1*) – Use pacotes computacionais, que estimam ou calculam cond(A), para verificar que quanto maior for n, mais mal condicionada é H_n.

(*b.2*) – Resolva os sistemas $H_n x = b_n$, $n = 3, 4, 5, \ldots, 10$, onde b_n é o vetor cuja i-ésima componente é

$$\sum_{j=1}^{n} \frac{1}{i + j - 1}$$ Desta forma a solução exata será: $x^* = (1 \; 1 \; . \; . \; . \; 1)^T$.

(*b.3*) – Analise seus resultados.

7ª Questão: Implementar um algoritmo em Python para o polinômio de Lagrange. Este algoritmo deve receber um arquivo e partir deste extrair os valores de x e y para serem utilizados na construção do polinômio de Lagrange.

8ª Questão: Resolva o problema a seguir.

Dada a tabela abaixo,

a) **Calcule $e^{3.1}$ usando um polinômio de interpolação sobre três pontos.**

b) **Dê um limitante para o erro cometido.**

x	2.4	2.6	2.8	3.0	3.2	3.4	3.6	3.8
e^x	11.02	13.46	16.44	20.08	24.53	29.96	36.59	44.70

9ª Questão: Resolva o problema a seguir.

8. A tabela abaixo representa a inflação **bimestral** medida pelo INPC no ano de 2000.

bimestre	*janeiro*	*fevereiro*	*marco*	*maio*	*junho*
inflação(%)	0,75	0,64	0,24	2,94	0,37

(a) Estime qual foi a inflação em abril , utilizando um polinômio interpolador de grau $n \leq 2$.

(b) Calcule o erro da estimativa anterior.

(c) Podemos garantir,usando o resultado do item anterior, que a inflação semestral foi menor que 6%?.

(d) Determine a inflação do mês de julho, usando um polinômio de grau $n \leq 2$.

10ª Questão: Resolva o problema a seguir.

Encontrar uma aproximação para $f(0.25)$ por spline cúbica natural na tabela:

x	0	0.5	1.0	1.5	2.0
f(x)	3	1.8616	−0.5571	−4.1987	−9.0536