Universidade Federal Rural do Rio de Janeiro
Instituto Multidisciplinar
Prof Erito Marques
Aluno: **Matricula:**

Primeira Prova de Implementação de Álgebra Linear Computacional

1) Construa um algoritmo que receba uma matriz A de ordem maior ou igual a 2 e avalie se essa matriz é uma matriz de Vandermonde. Matrizes de Vandermonde de ordem n tem a seguinte característica.

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix}.$$

b) Calcule a norma de Frobenius, norma linha, norma coluna de A e o produto interno entre a primeira linha e a segunda coluna, bem como o ângulo entre esses dois vetores.

c) Calcule o determinante da matriz A usando a seguinte equação:

$$\Delta(x_1, \ldots, x_n) \equiv \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{vmatrix} = \prod_{\substack{i,j \\ i>j}} (x_i - x_j)$$

2) a) Construa um algoritmo em C que faça a multiplicação de uma matriz de tamanho mxn por um vetor de tamanho nx1. Insira no seu código o cálculo do número total de flops.
b) Exiba o resultado da multiplicação entre uma matriz e um vetor no Matlab.
b) Construa um algoritmo em C que faça a multiplicação de uma matriz A(mxn) por uma matriz B(nxk) e que calcule o número total de flops.
c) Construa um algoritmo que calcule o determinante de uma matriz de ordem menor ou igual a 4.

4) Construa um algoritmo que receba uma matriz A com número de linhas e de colunas menores que 4. Use o método SOR para resolver o sistema Bx = b, onde b é digitado pelo usuário. Calcule o número condição da matriz A e um limitante inferior para ele (equação 2.2.28).

5) Construa um algoritmo que receba um uma matriz B e um vetor b inseridos pelo usuário e a solução do sistema Bx = b. Construa um algoritmo que compare a solução real com aquela obtida pelos métodos de Jacobi e de Gauss-Seidel.
b) Avalie se o critério das linhas e das colunas é satisfeito para a matriz B*B, B*B*B e B+B.
c) Faça perturbações de 0,02 em componentes do vetor b e veja o impacto na solução original. Repita essa análise fazendo perturbações nos coeficientes da matriz B.
d) Encontre o fator de Cholesky da matriz B ou retorne não ser possível realizar tal tarefa.
e) Encontre as matrizes L e U e resolva o sistema Bx = b com essas informações.

6) Construa um algoritmo que receba n vetores e avalie se eles são linearmente independentes.
b) Calcule vetores unitários na direção e sentido dos vetores informados pelo usuário.

7) Construa um algoritmo que calcule o w ótima e resolva o sistema Ax = b usando o método SOR (definido pelo usuário), onde A e uma matriz positiva definida de ordem 3 e tridiagonal.
b) Resolva o sistema Ax = b usando a fatoração de Cholesky.

8) Considere um sistema Ax = b, onde A e b são inseridos pelo usuário. Resolva o problema pela fatoração LU. Exiba as matrizes L e U. Compare a solução obtida com o método de Gauss-Seidel. Avalie se A é uma matriz banda.

9) Construa um algoritmo que receba uma matriz A associada a um sistema linear e retorne ao usuário quais métodos iterativos poderiam ser utilizados para resolver esse sistema. Indique também se a matriz atende o critério das linhas, das colunas, de Sassenfeld e da norma.

10) Considere um sistema Ax = b, onde A e b são inseridos pelo usuário.
a) Encontre o fator de Cholesky ou retorne não ser possível encontrá-lo.
b) Resolva o sistema Ax = b, caso tenha obtido o fator de Cholesky no item anterior.
c) Peça ao usuário para inserir um valor para w entre zero e dois e execute o método SOR.
d) Compare a solução obtida pelo método de Cholesky com a obtida com o metodo SOR.

11) a) Construa um algoritmo em C que receba uma matriz A com m linhas e n colunas. Em seguida calcule o traço da matriz A elevada ao quadrado. Exiba na tela o valor do traço e número de flops necessários para elevar a matriz ao quadrado bem como a ordem desse algoritmo.
b) Avalie se a matriz A é ortogonal.
c) Escolha de maneira aleatória duas colunas de A e calcule seu produto interno. Exiba o número de flops

12) Construa um algoritmo que avalie se uma matriz A de ordem k é triangular superior ou inferior. Caso seja triangular peça ao usuário que insira o vetor b. Resolva o sistema Ax = b. Sendo x o vetor solução do sistema. Retorne também um vetor unitário de mesma direção e sentido de x.

13) Construa um algoritmo que gere aleatoriamente uma matriz A quadrada de ordem k. Peça que o usuário insira um vetor b. Resolva Ax = b pelo método de Cholesky e pelo método de Gauss-Seidel. Compare as duas soluções substituindo-as no sistema original. Exiba o valor da norma de Frobenius do fator de Cholesky.

14) a) Construa um algoritmo avalie os critérios da norma, das linhas, das colunas e de Sassenfeld para um sistema Ax = b. O algoritmo deve indicar o que se concluir após aplicação de cada critério.

b) Resolva o sistema Ax=b usando o método de Jacobi.
c) Resolva o sistema Ax=b usando o método de Gauss-Seidel.
d) Faça perturbações no vetor b e avalie o impacto na solução.
e) Faça perturbações nos coeficientes da matriz A e avalie o impacto na solução.

15) Construa um algoritmo que receba uma matriz A e avalie se ela é tridiagonal, ortogonal ou positiva definida.

16) Construa um algoritmo que calcule o número condição de uma matriz A de ordem 3 e que retorne o valor da norma 1, infinito e de Frobenius da matriz inversa de A.

17) Construa um algoritmo que resolva o sistema Ax = b usando a fatoração QR. Considere A uma matriz de ordem 2. Seu algoritmo deve exibir ao final as matrizes Q e R.
b) Resolva o sistema usando o método SOR. Compare os resultados.

18) Construa um algoritmo que receba matrizes L e U triangulares inferior e superior. Calcule um limitante inferior do número condição. Use a equação (2.2.28). Avalie a influência do parâmetro w na sua resposta.

19) Construa um algoritmo receba um sistema Ax = b e o valor do vetor solução. Em seguida resolva o sistema pelos métodos de Jacobi, Gauss-Seidel e SOR. Utilize o que foi desenvolvido no exemplo 2.4.2 para fazer comparações entre as soluções.

20) Construa um algoritmo que receba p vetores de tamanho p. Compare as normas de cada vetor. Calcule a norma 1, norma 2 e norma infinito. Calcule a distância entre quaisquer desses dois vetores usando a norma 1, 2 e infinito. Construa um algoritmo que calcule o produto interno e o do ângulo entre quais quer desses dois vetores x e y. Construa um algoritmo que calcule a norma 1, norma infinito e norma de Frobenius de matrizes na qual cada vetor representa vetor representa uma linha. Calcule a calcule a distância entre quaisquer dessas matrizes.

21) Construa um algoritmo que calcule um limitante inferior do número condição. Use a equação (2.2.28). Dê um exemplo de aplicação do seu algoritmo usando uma matriz de Hilbert.

<u>Recomendações</u>
a) A prova deve ser feita em grupo. Cada questão deve conter uma explanação sucinta e adequada sobre o tema a ser implementado. Plágios de qualquer natureza ou citações sem a respectiva fonte não serão admitidas.
b) Os códigos devem ser implementados obrigatoriamente em linguagem C. O compilador a ser utlizado é o DEV-C.
c) As provas devem ser entregues impressas. Após a entrega, haverá uma arguição com os componentes do grupo a respeito do trabalho. Não serão admitidas entregas fora da data limite estabelecida.
d) As situações não contempladas nessa orientação serão avaliadas caso a caso.