

”

E-fólio B | Folha de resolução para E-fólio



UNIDADE CURRICULAR: Computação Numérica

CÓDIGO: 21180

DOCENTE: Paulo Shirley

A preencher pelo estudante

NOME: Sérgio Paulo Alves da Rosa

N.º DE ESTUDANTE: 1100477

CURSO: LEI

DATA DE ENTREGA: 29 de dezembro de 2024

TRABALHO / RESOLUÇÃO:

1. Introdução

O código desenvolvido implementa e demonstra a resolução de sistemas de equações lineares utilizando a decomposição LU (sem recorrer a pivot), calculando erros com base na norma infinita. Este relatório analisa os ficheiros “efb.m”, “sol_lu_mat.m” e “norma_inf_mat.m”.

2. Resumo das Funções Implementadas

2.1. efb.m

2.1.1. Objetivo Principal

O script (“efb.m”) demonstra, de forma prática, a resolução de um sistema de equações lineares $AX = B$, utilizando factoração LU (sem recorrer a pivot). Este cria duas matrizes aleatórias, A e X_0 , calcula $B = A * X_0$, e utiliza a função “sol_lu_mat(A, B, tol)” para obter a solução aproximada. Em seguida, avalia o erro via norma infinita.

2.1.2. Etapas Principais

- **Geração das Matrizes**
 - A e X_0 são inicializadas com dimensões 8×8 , com valores aleatórios no intervalo $[-5, 5]$, arredondados a duas casas decimais.
 - B é calculada pela multiplicação $A \times X_0$.
- **Verificação de Dimensões**
 - O script confirma a compatibilidade de dimensões entre A e X_0 . Caso sejam incompatíveis, o processamento é interrompido.
- **Resolução do Sistema**
 - “sol_lu_mat(A, B, tol)” faz a factoração de A em L (matriz triangular inferior) e U (matriz triangular superior) e realiza substituições diretas e regressivas para encontrar a solução X .
 - Se a matriz A é considerada singular (com base na tolerância “tol”), não há solução e a função retorna vazio.
- **Cálculo do Erro**

- Após obter a matriz X , o script utiliza (em outro ponto do código) a função “norma_inf_mat” para calcular a diferença entre X e X_0 .
Essa norma infinita fornece uma métrica da exatidão da solução.

2.1.3. Observações Finais

- O código é modular, em que separa a lógica de geração e de resolução do sistema na função “sol_lu_mat”.
- O uso de tolerância, “tol”, permite controlar a sensibilidade na detecção de pivôs muito pequenos, evitando a divisão por valores nulos ou próximos de zero.
- A formatação do output (“format bank”) ajuda na visualização de resultados com duas casas decimais.

2.2. sol_lu_mat.m

2.2.1. Objetivo

A função “sol_lu_mat(A, B, tol)” resolve o sistema de equações lineares $AX = B$ por meio da decomposição LU (sem recorrer a uso de pivot), verificando a singularidade de A com base em uma tolerância “tol”.

2.2.2. Estrutura Principal

- **Validação dos Parâmetros (linhas iniciais)**
 - Garante que a função é chamada com três argumentos (A, B, tol).
 - Verifica se a matriz A é quadrada e se a matriz B possui a dimensão compatível de forma que sejam efetuados os cálculos necessários.
- **Construção de L e U (linhas 51 a 71)**
 - Inicializa L como matriz identidade e U como matriz nula.
 - Preenche as linhas de U e as colunas de L .
 - Se o pivô $|U_{(k,k)}|$ fica abaixo de “tol”, a matriz é considerada singular, e a matriz X vazia.
- **Resolução do Sistema (linhas 72 a 84)**
 - Após obter L e U , efetua duas substituições:
 1. Forward-substitution ($L_y = B$).
 2. Backward-substitution ($U_x = Y$).
 - O resultado é armazenado na matriz X .

2.2.3. Fluxo de Execução

- Recebe A , B e tol .
 - Verifica a autenticidade das dimensões.
 - Realiza a factoração LU sem recorrer a pivot, monitorizando os pivôs pequenos.
 - Caso não encontre singularidade, resolve o sistema para cada coluna de B .
 - Retorna a matriz X como a solução do sistema ou vazio se a matriz A for considerada singular.

2.2.4. Conclusão

O ficheiro implementa uma rotina de factoração LU passo a passo, analisando a possível singularidade de forma controlada pela tolerância (“tol”). Isto permite resolver sistemas de equações lineares $AX = B$ de maneira modular e eficiente.

2.3. norma_inf_mat.m

2.3.1. Objetivo

A função “norma_inf_mat(A)” calcula a norma infinita de uma matriz A , que consiste no maior valor absoluto de soma de cada linha.

2.3.2. Estrutura do Código

- Recebe como parâmetro uma matriz A de dimensão $(m \times n)$.
- Itera por cada linha i de A , somando os valores absolutos dos elementos dessa linha (“rowSum”).
- Compara com o valor atual de N e atualiza se o “rowSum” for maior.

2.3.3. Resultado

Retorna a maior soma de valores absolutos entre todas as linhas da matriz, representando a norma infinita

3. Testes e Resultados

3.1. Parâmetros Utilizados

Tamanho da matriz: 8×8 .

Tolerância: 10^{-6}

3.2. Resultados Obtidos

Matrizes Geradas: Valores variam entre $[-5, 5]$.

Norma Infinita do Erro: Valores típicos, compatíveis com a precisão numérica esperada.

3.3. Tratamento de Singularidades

O script retorna corretamente uma mensagem de erro e interrompe o processamento quando detecta singularidades na matriz A .

4. Avaliação

Eficiência: As implementações são otimizadas para matrizes de tamanho médio, utilizando substituições diretas e regressivas, sem funções pré-definidas.

Clareza: Comentários e validações ajudam na compreensão e evitam erros de uso.

Limitação: Não suporta recurso a pivot, o que pode comprometer a precisão e estabilidade numérica em algumas matrizes mal condicionadas.

5. Conclusão

O trabalho demonstra o uso eficaz de métodos numéricos para resolver sistemas de equações lineares, oferecendo uma base sólida para extensões futuras, como a inclusão de pivot ou aplicação a problemas maiores.

6. Referências

Livro base: Análise Numérica.

Notas da disciplina: Computação Numérica.