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

UNIDADE CURRICULAR: Computação numérica

CÓDIGO: 21180

DOCENTE: Paulo Shirley

A preencher pelo estudante

NOME: Hugo Miguel Lopes Silva

N.º DE ESTUDANTE: 2100455

CURSO: Licenciatura em Engenharia Informática

DATA DE ENTREGA: 8 de Janeiro de 2024

TRABALHO / RESOLUÇÃO:

1.1)

A função condm foi implementada da seguinte forma:

```
function c = condm(A, p)
    % Verifica se p é igual a 1 ou infinito
    if ~(p == 1 || p == inf)
        % Se p não for 1 ou infinito, retorna uma matriz vazia
        c = [];
        return;
    end

    % Calcula a norma de A com base em p usando a função my_norm
    normA = my_norm(A, p);

    % Calcula a inversa de A
    A_inv = inv(A);

    % Calcula a norma da inversa de A com base em p usando a função my_norm
    normA_inv = my_norm(A_inv, p);

    % Calcula a matriz de condicionamento como o produto das duas normas
    c = normA * normA_inv;
end
```

Funções Auxiliares

Para calcular as normas ℓ_1 e $\ell\infty$, foram criadas duas funções auxiliares: my_abs e my_sum.

A função my_abs recebe uma matriz A como entrada e retorna uma matriz B com os valores absolutos de A. Isso é feito percorrendo cada elemento de A e trocando o sinal dos elementos negativos.

A função my_sum calcula a soma das colunas (quando dim é 1) ou das linhas (quando dim é 2) de uma matriz A. Ela inicializa um vetor ou matriz de zeros, dependendo da dimensão desejada, e realiza a soma dos elementos conforme a dimensão especificada.

A função my_norm calcula a norma ℓ_1 ou $\ell\infty$ de uma matriz A, dependendo do valor de p. Para ℓ_1, ela utiliza a função my_sum para calcular a norma de máximo de coluna dos valores absolutos da matriz A. Para $\ell\infty$, ela calcula a norma de máximo de linha dos valores absolutos da matriz A.

De seguida 3 casos de teste para verificar a execução da função pedida:

Para p =1:

```
>> A = [6, 1 ,4 ; 4,8,4; 6,3,5]
A =

   6   1   4
   4   8   4
   6   3   5

>> p=1
p = 1
>> c = condm(A, p);
>> display (c)
c = 45.714
>>
```

Para p = inf:

```
>> A = [6, 1 ,4 ; 4,8,4; 6,3,5]
A =

   6   1   4
   4   8   4
   6   3   5

>> p=inf
p = Inf
>> c = condm(A, p);
>> display (c)
c = 52.571
>>
```

E agora para p = 0.5:

```
>> A = [6, 1 ,4 ; 4,8,4; 6,3,5]
A =

   6   1   4
   4   8   4
   6   3   5

>> p=0.5
p = 0.5000
>> c = condm(A, p);
>> display (c)
c = [](0x0)
>>
```

1.2)

1. Inicialização e Pré-Algoritmo:

A função começa por inicializar algumas variáveis. n é a dimensão da matriz A (número de linhas ou colunas), e x é o vetor que armazenará a solução do sistema linear.

2. Escolha Parcial de Pivot:

A implementação segue o método de escolha parcial de pivot para garantir a estabilidade numérica do algoritmo. Isso significa que a cada iteração do loop externo (controlado pela variável k), a função procura o elemento com o maior valor absoluto na coluna atual (k) e troca as linhas se necessário. Isso é feito para evitar divisões por valores muito pequenos e potencialmente instáveis.

```
[max_val, i_max] = max_custom(abs_custom(A(k:n, k)));
i_max = i_max + k - 1;
```

O código acima calcula o índice i_max do elemento com o maior valor absoluto na coluna atual a partir da linha k até o final. De seguida, ele verifica se o valor absoluto desse elemento é menor que a tolerância (tol). Se for, a matriz A é considerada singular, e a função retorna um vetor vazio x com uma mensagem de erro.

3. Troca de Linhas:

Se o elemento com o maior valor absoluto não estiver na linha k, a função realiza uma troca de linhas entre a linha k e a linha i_max na matriz A e também no vetor b. Isso garante que o elemento com o maior valor absoluto se torne o elemento pivô da iteração atual.

```
if i_max ~= k
    A([k, i_max], :) = A([i_max, k], :);
    b([k, i_max]) = b([i_max, k]);
end
```

4. Eliminação:

A eliminação é realizada nas linhas abaixo da linha atual (k). O objetivo é fazer com que todos os elementos abaixo do elemento pivô (A(k, k)) se tornem zero. Isso é feito subtraindo uma múltipla adequada da linha pivô da linha atual.

```
for i = k+1:n
    factor = A(i, k) / A(k, k);
    A(i, k:n) = A(i, k:n) - factor * A(k, k:n);
    b(i) = b(i) - factor * b(k);
end
```

5. Verificação de Singularidade:

No final do loop externo, a função verifica novamente se o elemento pivô da última iteração é menor que a tolerância (tol). Se for, a matriz A é considerada singular, e a função retorna um vetor vazio x com uma mensagem de erro.

6. Substituição Inversa:

Após a eliminação, a matriz A torna-se triangular superior, e a função realiza a substituição inversa para encontrar o vetor solução x. Ela começa pela última linha e vai subindo até a primeira, resolvendo as equações lineares.

7. Funções Auxiliares:

As funções auxiliares abs_custom e max_custom são usadas para calcular o valor absoluto de um número e encontrar o índice do valor máximo em um vetor, respectivamente.

De seguida o exemplo de 2 casos de teste:

```
>> A = [2, 3, -1; 4, 4, -3; 2, -3, 2];
>> b = [8; 3; 1];
>> tol = 1e-6;
>> x = elim_gausspt(A, b, tol);
>> disp(x)
   0.9167
   3.8333
   5.3333
>>
```

```
>> A = [1, 2, 3; 4, 5, 6; 7, 8, 9];
>> b = [1; 2; 3];
>> tol = 1e-6;
>> x = elim_gausspt(A, b, tol);
Matriz A singular!
>> disp(x)
[](0x0)
>>
```

1.3)

1-Criação de Matriz e Vetor Aleatórios e definição de tolerância :

Neste trecho, uma matriz A (15x15) e um vetor b (15x1) são criados com elementos aleatórios no intervalo [-1, 1].

A é uma matriz de coeficientes do sistema linear.

b é o vetor do lado direito das equações.

tol é a tolerância que pretendemos utilizar para a resolução do exercicio.

```
A = rand(15) * 2 - 1;  % Elementos em [-1, 1]
b = rand(15, 1) * 2 - 1;  % Elementos em [-1, 1]
tol = 1e-13;
```

2-Inicialização de vetores para armazenar dados para gráfico:

Aqui, dois vetores erros e num_conds são inicializados com zeros para armazenar os dados que serão usados para criar o gráfico no final.

```
erros = zeros(1, 50);
num_conds = zeros(1, 50);
```

3-Função para calcular a Norma Euclidiana:

Uma função chamada normaEuclidiana é definida para calcular a norma euclidiana de um vetor v.

```
function norma = normaEuclidiana(v)
    norma = sqrt(sum(v.^2));
end
```

4-Ciclo de 50 Iterações:

Um loop é executado 50 vezes (de 1 a 50) para realizar uma série de cálculos em cada iteração.

A variável k é calculada como (i - 1) * (0.9 / 49) para variar gradualmente de 0.0 a 0.9 ao longo das iterações.

```
for i = 1:50
    k = (i - 1) * (0.9 / 49);
    A(1, :) = (1 - k) * A(1, :) + k * A(2, :);
```

5-Definiçao de p e cálculo do Número de Condição:

O valor de p é definido como 1.

É possível ajustá-lo para Inf (infinito) de acordo com o que pretendemos.

Em cada iteração, o número de condição da matriz A é calculado usando a função condm(A, p).

```
p = 1;  % ou p = Inf;
num_cond = condm(A, p);
```

6-Resolução do Sistema Ax = b e cálculo da Norma do Erro:

O sistema linear Ax = b é resolvido usando a função elim_gausspt(A, b, tol).

A norma do erro, que é a norma euclidiana do vetor de resíduo Ax - b, é calculada. Se o vetor de solução x for vazio, o erro é considerado infinito.

```
x = elim_gausspt(A, b, tol);
if isempty(x)
    error_norm = Inf;
else
    error_norm = normaEuclidiana(A * x - b);
end
```

7-Armazenamento de dados para o Gráfico e exibição de resultados:

Os valores do número de condição (num_cond) e do erro (error_norm) são armazenados nos vetores num_conds e erros para posterior plotagem.

Os resultados são exibidos no ecrã com informações sobre a iteração atual.

```
    erros(i) = error_norm;
    num_conds(i) = num_cond;
    fprintf('Iteração %d: Número de Condição = %f, Norma do Erro = %f\n', i, num_cond, error_norm);
end
```

8-Filtro de valores Infinitos e criação do gráfico:

Valores infinitos no vetor de erros são filtrados para evitar problemas de plotagem.

Um gráfico log-log é criado com os valores dos erros em relação aos números de condição, com títulos e etiquetas de eixo adequadas.

```
indices_validos = isfinite(erros) & isfinite(num_conds);
erros_filtrados = erros(indices_validos);
num_conds_filtrados = num_conds(indices_validos);

loglog(num_conds_filtrados, erros_filtrados, 'b-');
title('norma do erro vs num. condição');
xlabel('log10(Número de Condição)');
ylabel('log10(Erro)');
grid on;
```

Exemplo de execução:

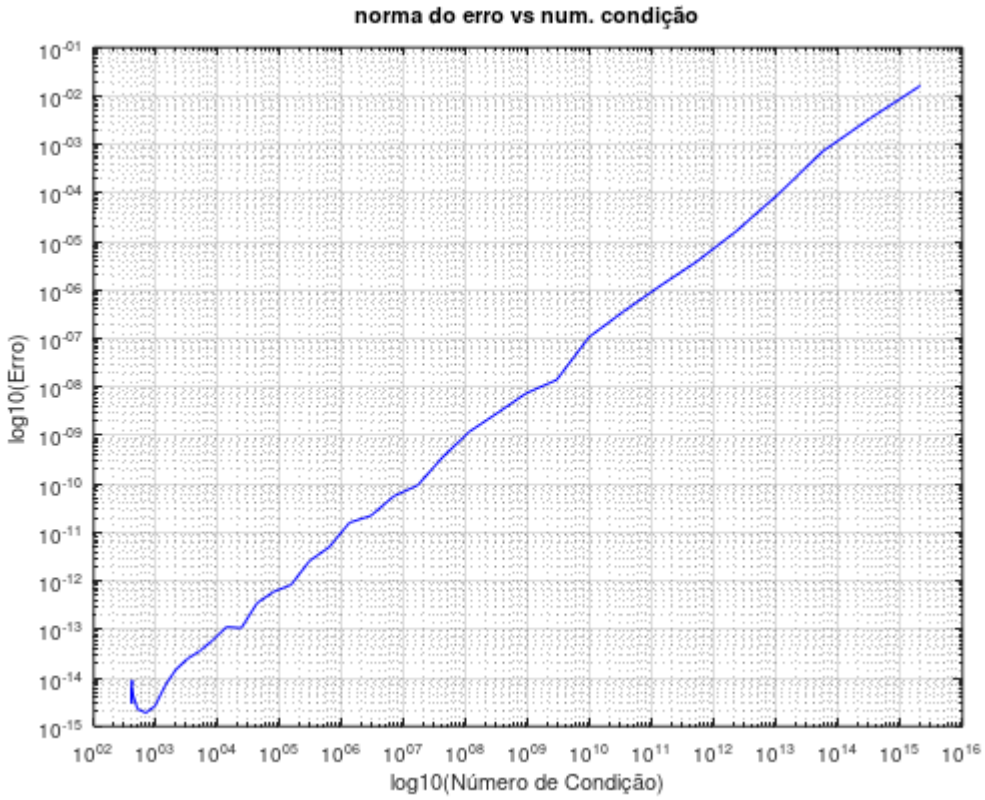
```
>> efb23
Iteração 1: Número de Condição = 414.242431, Norma do Erro = 0.000000
Iteração 2: Número de Condição = 413.980871, Norma do Erro = 0.000000
Iteração 3: Número de Condição = 413.467358, Norma do Erro = 0.000000
Iteração 4: Número de Condição = 412.725384, Norma do Erro = 0.000000
Iteração 5: Número de Condição = 411.790598, Norma do Erro = 0.000000
Iteração 6: Número de Condição = 410.707963, Norma do Erro = 0.000000
Iteração 7: Número de Condição = 409.528113, Norma do Erro = 0.000000
Iteração 8: Número de Condição = 408.303315, Norma do Erro = 0.000000
Iteração 9: Número de Condição = 407.083517, Norma do Erro = 0.000000
Iteração 10: Número de Condição = 405.912884, Norma do Erro = 0.000000
Iteração 11: Número de Condição = 404.827195, Norma do Erro = 0.000000
Iteração 12: Número de Condição = 403.852290, Norma do Erro = 0.000000
Iteração 13: Número de Condição = 403.003635, Norma do Erro = 0.000000
Iteração 14: Número de Condição = 403.735093, Norma do Erro = 0.000000
Iteração 15: Número de Condição = 508.601025, Norma do Erro = 0.000000
Iteração 16: Número de Condição = 695.070844, Norma do Erro = 0.000000
Iteração 17: Número de Condição = 976.687199, Norma do Erro = 0.000000
Iteração 18: Número de Condição = 1411.697398, Norma do Erro = 0.000000
Iteração 19: Número de Condição = 2099.744186, Norma do Erro = 0.000000
Iteração 20: Número de Condição = 3215.300066, Norma do Erro = 0.000000
Iteração 21: Número de Condição = 5071.377726, Norma do Erro = 0.000000
Iteração 22: Número de Condição = 8243.956924, Norma do Erro = 0.000000
Iteração 23: Número de Condição = 13821.308961, Norma do Erro = 0.000000
Iteração 24: Número de Condição = 23917.149151, Norma do Erro = 0.000000
Iteração 25: Número de Condição = 42756.730675, Norma do Erro = 0.000000
Iteração 26: Número de Condição = 79043.656645, Norma do Erro = 0.000000
Iteração 27: Número de Condição = 151277.316271, Norma do Erro = 0.000000
Iteração 28: Número de Condição = 300086.302210, Norma do Erro = 0.000000
Iteração 29: Número de Condição = 617804.833458, Norma do Erro = 0.000000
Iteração 30: Número de Condição = 1321919.045273, Norma do Erro = 0.000000
Iteração 31: Número de Condição = 2944251.163140, Norma do Erro = 0.000000
Iteração 32: Número de Condição = 6837335.660640, Norma do Erro = 0.000000
Iteração 33: Número de Condição = 16585589.400677, Norma do Erro = 0.000000
Iteração 34: Número de Condição = 42108462.288148, Norma do Erro = 0.000000
Iteração 35: Número de Condição = 112136634.609341, Norma do Erro = 0.000000
Iteração 36: Número de Condição = 313982542.859934, Norma do Erro = 0.000000
Iteração 37: Número de Condição = 926815905.129525, Norma do Erro = 0.000000
Iteração 38: Número de Condição = 2892610077.417536, Norma do Erro = 0.000000
Iteração 39: Número de Condição = 9576884455.833014, Norma do Erro = 0.000000
Iteração 40: Número de Condição = 33760242408.776222, Norma do Erro = 0.000000
Iteração 41: Número de Condição = 127250062642.689301, Norma do Erro = 0.000001
Iteração 42: Número de Condição = 515310171343.783875, Norma do Erro = 0.000004
Iteração 43: Número de Condição = 2254456050488.990723, Norma do Erro = 0.000016
Iteração 44: Número de Condição = 10725196817791.535156, Norma do Erro = 0.000094
Iteração 45: Número de Condição = 55904921951761.726562, Norma do Erro = 0.000720
```

```
Iteração 46: Número de Condição = 3221373991294 32.687500, Norma do Erro = 0.003424
Iteração 47: Número de Condição = 2079499798056551.000000, Norma do Erro = 0.016179
warning: matrix singular to machine precision, rcond = 6.60974e-17
warning: called from
    condm at line 13 column 11
    efb23 at line 24 column 14

Matriz A singular!
Iteração 48: Número de Condição = 15129189795111002.000000, Norma do Erro = Inf
warning: matrix singular to machine precision, rcond = 8.37844e-18
warning: called from
    condm at line 13 column 11
    efb23 at line 24 column 14

Matriz A singular!
Iteração 49: Número de Condição = 119354021835438000.000000, Norma do Erro = Inf
warning: matrix singular to machine precision, rcond = 1.47016e-18
warning: called from
    condm at line 13 column 11
    efb23 at line 24 column 14

Matriz A singular!
Iteração 50: Número de Condição = 680198332540650624.000000, Norma do Erro = Inf
>>
```



O gráfico representa a relação entre o número de condição de uma matriz e a norma do erro ao resolver um sistema linear Ax=b através do método numérico de eliminação de Gauss com escolha parcial de pivot.

Analisando o gráfico, observamos um aumento exponencial da norma do erro à medida que o número de condição aumenta. Um número de condição elevado (em escala logarítmica) indica que a matriz A está perto de ser singular ou mal condicionada, resultando em grande

sensibilidade a pequenas perturbações nos dados de entrada, o que se reflete na precisão da solução encontrada para x.

```
Iteração 48: Número de Condição = 15129189795111002.000000, Norma do Erro = Inf
warning: matrix singular to machine precision, rcond = 8.37844e-18
warning: called from
    condm at line 13 column 11
    efb23 at line 24 column 14

Matriz A singular!
Iteração 49: Número de Condição = 119354021835438000.000000, Norma do Erro = Inf
warning: matrix singular to machine precision, rcond = 1.47016e-18
warning: called from
    condm at line 13 column 11
    efb23 at line 24 column 14
```

A norma do erro passa a infinito a partir da iteração 48 porque a matriz A tornou-se singular ou quase singular. Isso é indicado pelas mensagens de aviso que mencionam que a matriz é "singular to machine precision" e o valor de "rcond" que é muito próximo de zero.

A tendência exibida no gráfico é consistente com a teoria: sistemas com números de condição altos são mais propensos a ter soluções numéricas com erros significativos devido à amplificação de erros de arredondamento inerentes aos cálculos numéricos.

Além disso, o gráfico log-log permite uma visualização clara da relação proporcional entre o logaritmo do número de condição e o logaritmo do erro, evidenciando a relação exponencial entre essas duas quantidades.