Universidade do Minho
Departamento de Matemática
Lic. em Ciências da Computação

Ano Lectivo de 2022/2023
1^o Semestre

Análise Numérica

Folha 6 - Sistemas de equações lineares

---

1. Dada uma matriz A, de ordem n, não-singular, e um vector b, a execução de $\gg x = A\backslash b$, no Matlab, dá a solução x do sistema $Ax = b$. No caso de A não ser quadrada (isto é, o número de incógnitas x_i diferente do número de equações), a solução do sistema pode não existir (sistema impossível) ou o sistema pode ser indeterminado (isto é, a solução não ser única). Mesmo nestes casos, o Matlab apresentará sempre uma e uma só "solução"que é preciso saber interpretar em cada caso. Execute, $\gg x = A\backslash b$ em cada um dos seguintes casos e comente os resultados obtidos:

   a) $A = [1\ 2\ 3; 4\ 5\ 6]$, $b = [1; 1]$

   b) $A = [3\ 1; 2\ 1; 1\ 0]; b = [5\ 4\ 1]'$

   c) $A = [3\ 1; 2\ 1; 1\ 0]; b = [5\ 4\ 0]'$

   d) $A = [3\ 1\ 2; 2\ 1\ 6; 5\ 2\ 8]; b = [5\ 4\ 1]'$

2. **a)** Apresente na forma de um algoritmo o método de substituição inversa para sistemas $Ax = b$, sendo A uma matriz triangular superior;

   b) Verifique que o número de operações aritméticas do algoritmo é exactamente igual a n^2, sendo n a ordem da matriz A;

   c) Desenvolva no Matlab uma função, $x =$ `STriangular` (A, b), que implementa o algoritmo;

   d) Para gerar a matriz que é a parte triangular superior da matriz de Hilbert de ordem 5, execute $\gg A = triu(hilb(5))$; use a função da alínea anterior para resolver o sistema com a matriz A e b o vector de unidades.

   e) Use a mesma função o número de vezes que for necessário para calcular a inversa A^{-1}. Compare o resultado obtido com a matriz dada por $inv(A)$.

3. **a)** Apresente na forma de um algoritmo o método de eliminação de Gauss (sem pivotação) para resolver o sistema $Ax = b$, sendo A uma matriz de ordem n;

   b) Verifique que o número de operações aritméticas do algoritmo é aproximadamente igual a $\frac{2}{3}n^3$;

   c) Desenvolva no Matlab uma função, $x =$ `GaussElim`(A, b), que implementa o algoritmo; reduzida a matriz do sistema à forma triangular superior, por transformações de equivalência, a função `GaussElim` usa a função `STriangular` para a fase da substituição inversa;

   d) Teste a sua função `GaussElim` com $A = rand(4)$ e $b = ones(4, 1)$. Calcule o resíduo $r := b - Ax$.

4. Considere a seguinte matriz

$$A = \begin{bmatrix} 2 & 4 & 1 \\ 0.5 & 1 & 1.25 \\ 2 & 3 & 4 \end{bmatrix}$$

   a) Tente resolver o sistema $Ax = b$, com $b = ones(3,1)$, usando a função `GaussElim` que desenvolveu antes. O que acontece? Qual é o problema?

   b) Altere o valor na posição $(2,2)$ da matriz fazendo $\gg A(2,2) = 1 + 2\hat{\ } - 52$. Resolva o sistema com $b = ones(3,1)$, usando a função `GaussElim`. Compare a solução obtida com o resultado dado por $\gg A \backslash b$.

5. **a)** A partir do código da função `GaussElim` desenvolva uma implementação do método de eliminação de Gauss com pivotação parcial (chame-lhe `GaussElimPP).`

   b) Use a nova função para resolver o sistema do exercício anterior e compare a solução obtida com o resultado de $\gg A \backslash b$.

6. **a)** No Matlab execute $A = hilb(10)$ para definir a matriz de Hilbert de ordem 10 e $x = ones(10,1)$. Calcule o vector b, tal que x é a solução do sistema $Ax = b$.

   b) Verifique que a solução $xtil$ do sistema $Ax = b$ dada por $xtil = A \backslash b$ tem um erro elevado. Qual é o problema?

7. Considere as matrizes seguintes

$$A = \begin{bmatrix} \epsilon & 1 \\ 1 & 1 \end{bmatrix}, L = \begin{bmatrix} 1 & 0 \\ \epsilon^{-1} & 1 \end{bmatrix} \text{ e } U = \begin{bmatrix} \epsilon & 1 \\ 0 & 1 - \epsilon^{-1} \end{bmatrix}.$$

   Uma vez que se tem $A = L \cdot U$, para determinar a solução do sistema $Ax = b$, podemos resolver $Ly = b$ e $Ux = y$.

   a) Para resolver o sistema com $\epsilon = 2^{-52}$ e $b = [10\ 1]^t$, defina no Matlab as matrizes L e U e resolva os sistemas $Ly = b$ e $Ux = y$.

   b) Tendo em conta que a solução exacta é $x = \left(\frac{9}{\epsilon-1}, \frac{\epsilon-10}{\epsilon-1}\right)$, use a teoria do condicionamento de sistemas de equações lineares para explicar o erro cometido.

   c) Uma vez que podemos escrever $A = P \cdot L \cdot U$, com

$$L = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}, U = \begin{bmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{bmatrix} \text{ e } P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

   podemos usar esta decomposição para resolver o mesmo sistema. Será de esperar uma solução melhor do que a obtida em a)? Porquê?

RESOLUÇÃO

1. **a)** O sistema tem duas equações e três incógnitas e todos os vetores da forma

$$\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix},$$

com α qualquer, são solução (o sistema é indeterminado). Neste caso o Matlab apresenta apenas uma das soluções

```
>> A=[1 2 3; 4 5 6]; b=[1;1]; x=A\b

x =

   -0.5000
         0
    0.5000
```

b) O sistema tem três equações e duas incógnitas mas, apesar disso, tem solução

```
>> A=[3 1; 2 1; 1 0]; b=[5 4 1]'; x=A\b
x =

    1.0000
    2.0000
```

c) O sistema tem três equações e duas incógnitas e não tem solução (é impossível); neste caso o Matlab devolve a solução do sistema

$$(A^T A)x = A^T b.$$

Este vetor x (conhecido como a aproximação dos quadrados mínimos) minimiza o erro $||Ax - b||_2$.

```
>> A=[3 1; 2 1; 1 0]; b=[5 4 0]'; x=A\b

x =

    0.3333
    3.6667

>> norm(A*x-b)

ans =

    0.5774
```

Para qualquer outro vetor $y \in \mathbb{R}^3$, é $||Ay - b||_2 > ||Ax - b||_2$.

d) Embora a matriz seja quadrada (o sistema tem tantas equações quantas as incógnitas), ela é singular e o sistema é impossível.

```
A=[3 1 2; 2 1 6; 5 2 8]; b=[5 4 1]'; x=A\b
```

```
Warning: Matrix is close to singular or badly scaled. Results may
be inaccurate. RCOND =  7.304099e-18.


ans =

   1.0e+16 *

    1.4412
   -5.0440
    0.3603
```

2. **d)** `>> A=triu(hilb(5));  x=STriangular(A,ones(5,1))`

```
x =

   -0.1229
   -0.4504
   -0.6994
   -0.8750
    9.0000
```

e) A k-ésima coluna da matriz A^{-1} é a solução do sistema $Ax = e_k$ onde e_k representa a k-ésima coluna da matriz identidade. No seguinte denotaremos A^{-1} por X.

```
>> for k=1:5; b=zeros(5,1); b(k)=1; X(:,k)=A\b; end
>> X, inv(A)

X =

    1.0000   -1.5000    0.2083    0.1069    0.0618
         0    3.0000   -3.7500    0.1750    0.1246
         0         0    5.0000   -5.8333    0.1339
         0         0         0    7.0000   -7.8750
         0         0         0         0    9.0000


ans =

    1.0000   -1.5000    0.2083    0.1069    0.0618
         0    3.0000   -3.7500    0.1750    0.1246
         0         0    5.0000   -5.8333    0.1339
         0         0         0    7.0000   -7.8750
         0         0         0         0    9.0000
```

3. **a)** `>> A=[2 4 1; 0.5 1 1.25; 2 3 4]; b=ones(3,1); x=GaussElim(A,b)`

```
ans =

    2.0000    4.0000    1.0000    1.0000
         0         0    1.0000    0.7500
         0   -1.0000    3.0000         0
```

```
ans =

    2.0000    4.0000    1.0000    1.0000
         0         0    1.0000    0.7500
         0       NaN       Inf       Inf


x =

   NaN
   NaN
   NaN
```

O 1º passo de redução produz um zero na posição (2,2) da matriz ampliada. Como a função GaussElim não faz troca de linhas, o multiplicador calculado é

$$m = -1/0 = -Inf$$

o que conduz a uma solução NaN (not-a-number). Em resumo, sem troca de linhas o método de eliminação de Gauss falha neste caso.

b)
```
>> A(2,2)=1+2^-52; x=GaussElim(A,b)
x =

   -3.8750
    2.0000
    0.7500

>> A\b

ans =

   -4.3750
    2.2500
    0.7500
```

Com a alteração efetuada na entrada $A(2,2)$, o pivot no segundo passo de redução já não é nulo mas é muito mais pequeno do que o elemento A(3,2) a eliminar. Isto produz um multiplicador muito grande (em valor absoluto)

$$m = -1/2^{-52} = -2^{52}$$

que vai causar erros grandes na solução.

4. **a)**
```
>> A=hilb(10); x=ones(10,1); b=A*x

b =

   2.9290e+00
   2.0199e+00
   1.6032e+00
   1.3468e+00
```

```
   1.1682e+00
   1.0349e+00
   9.3073e-01
   8.4670e-01
   7.7725e-01
   7.1877e-01
```

b) `>> xtil=A\b; norm(xtil-x)`

```
ans =

   8.3663e-04
```

O número de condição é muito grande, logo o sistema é mal-condicionado. Com efeito, tem-se

```
>> cond(A)

ans =

   1.6025e+13
```

o que faz com que pequenos erros devidos aos arredondamentos sejam muito ampliados afetando a precisão da solução calculada.