

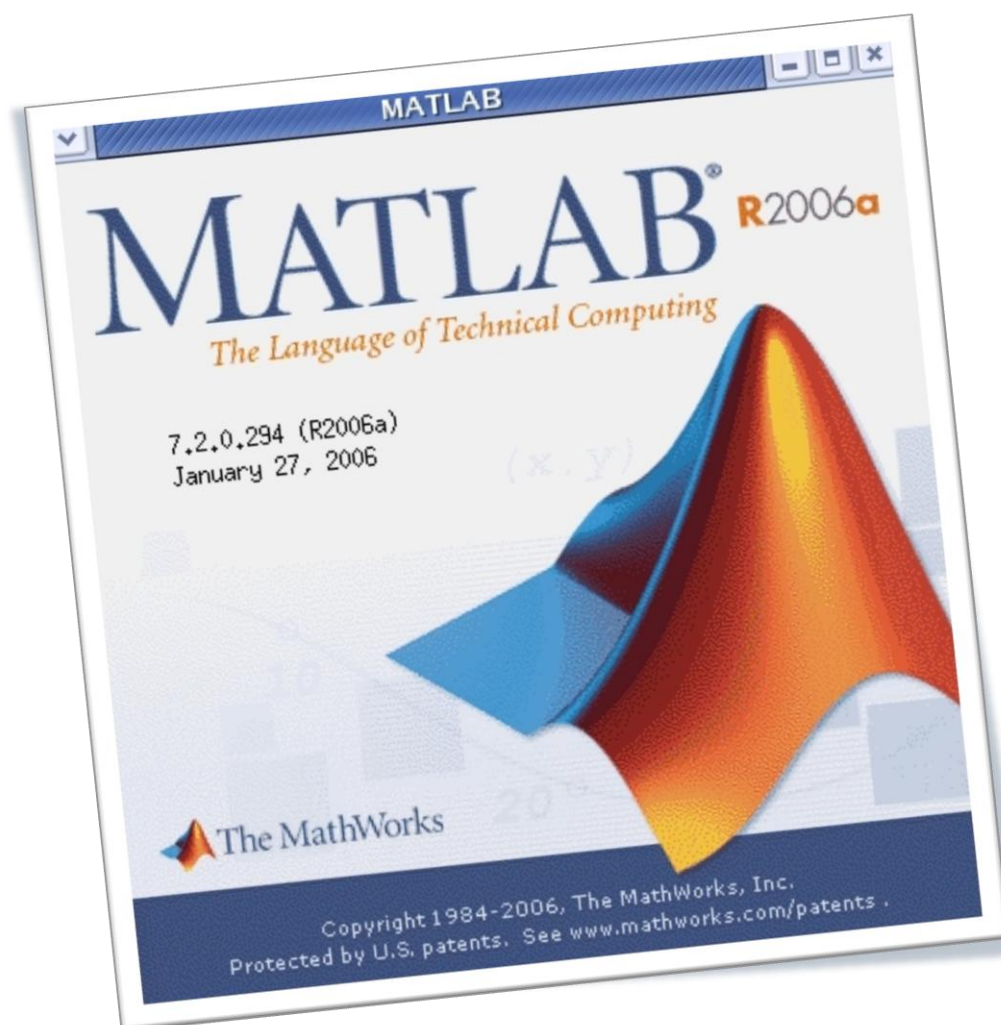
---

# ANÁLISE NUMÉRICA

---

## Sistemas de Equações Lineares

---



## 1. Inversa

O sistema de equações lineares

$$\begin{cases} 0.125x + 0.2y + 0.4z = 2.3 \\ 0.375x + 0.5y + 0.6z = 4.8 \\ 0.5x + 0.3y = 2.9 \end{cases}$$

Pode ser escrito na forma matricial por  $AX=B$

Onde

$$A = \begin{bmatrix} 0.125 & 0.2 & 0.4 \\ 0.375 & 0.5 & 0.6 \\ 0.5 & 0.3 & 0 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 2.3 \\ 4.8 \\ 2.9 \end{bmatrix}$$

O sistema terá solução única caso

$$\det(A) \neq 0$$

e a sua solução pode ser obtida por

$$X = A^{-1}B.$$

Em Matlab:

```
det(A)
X=inv(A)*B
A*X % verificação
```

## 2. Método de Eliminação de Gauss

```
function x = Gauss(A, b)
[n, n] = size(A);      % Descobre a dimensão de A
[n, k] = size(b);      % Descobre a dimensão de b
x = zeros(n,k);        % Cria uma matriz x
for i = 1:n-1
    m = -A(i+1:n,i)/A(i,i); % multiplicadores
    A(i+1:n,:) = A(i+1:n,:) + m*A(i,:);
    b(i+1:n,:) = b(i+1:n,:) + m*b(i,:);
end;
x(n,:) = b(n,+)/A(n,n); % Retrosustituição
for i = n-1:-1:1
    x(i,:) = (b(i,:) - A(i,i+1:n)*x(i+1:n,:))/A(i,i);
end
```

O método de eliminação de Gauss consiste em escrever a matriz ampliada  $A|B$  em escada e depois fazer-se a retrossubstituição. Para resolver o sistema dado na secção 1 basta guardar um m-file (script) com o nome Gauss e depois correr a instrução

```
Gauss(A,B)
```

## 3. Factorização LU

Neste método usa-se o comando do Matlab lu:

```
[L,U,P]=lu(A)
```

A matriz L que se obtém é triangular inferior, a matriz U é triangular superior e a matriz P é uma matriz de permutações que indica se é necessário fazer troca de linhas para obter a factorização LU. Se P não for a matriz identidade, então é necessário trocar linhas nas matrizes A e B. Tais trocas deduzem-se da transformação de P na matriz identidade.

No exemplo da secção 1, obtém-se o seguinte resultado para a matriz P

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

que mostra que se teve de trocar a linha 3 com a linha 1. Assim para resolvermos por este método poder-se-á trocar logo no início as matrizes A e B pela seguintes matrizes

$$A_2 = \begin{bmatrix} 0.5 & 0.3 & 0 \\ 0.375 & 0.5 & 0.6 \\ 0.125 & 0.2 & 0.4 \end{bmatrix} \quad \text{e} \quad B_2 = \begin{bmatrix} 2.9 \\ 4.8 \\ 2.3 \end{bmatrix}$$

```
[L,U,P]=lu(A2)
```

Recomeça-se a factorização no Matlab

Note-se que o sistema escreve-se agora por

$$AX = B \Leftrightarrow LUX = B$$

E portanto tem-se que realizar os seguintes passos para resolver o sistema:

1.  $LY=B$  para descobrirmos a matriz Y

2.  $UX=Y$  para descobrirmos a solução  $X$ .

Ou seja, no Matlab

```
Y=subs(L,B2)
X=retro(U,Y)
```

onde as funções subs e retro são definidas por

```
function x = subs(A, b)
[n, n] = size(A);      % Descobre a dimensão de A
[n, k] = size(b);      % Descobre a dimensão de b
x = zeros(n,k);        % Cria uma matriz x
x(1,:) = b(1,:)/A(1,1); % substituição
for i = 2:1:n
    x(i,:) = (b(i,:) - A(i,1:i-1)*x(1:i-1,:))/A(i,i);
end
```

```
function x = retro(A, b)
[n, n] = size(A);      % Descobre a dimensão de A
[n, k] = size(b);      % Descobre a dimensão de b
x = zeros(n,k);        % Cria uma matriz x
x(n,:) = b(n,:)/A(n,n); % Retrosustituição
for i = n-1:-1:1
    x(i,:) = (b(i,:) - A(i,i+1:n)*x(i+1:n,:))/A(i,i);
end
```

e

## 4. Pesquisa Parcial de Redutor

O Matlab tem por defeito uma função que factoriza a matriz  $A$  em LU e resolve o sistema com escolha parcial de redutor:

## 5. Exemplo

O sistema  $AX=B$  onde as matrizes  $A$  e  $B$  são

$$A = \begin{bmatrix} 1^9 & 1^8 & \dots & 1^0 \\ 2^9 & 2^8 & \dots & 2^0 \\ 3^9 & 3^8 & \dots & 3^0 \\ \vdots & \vdots & \ddots & \vdots \\ 10^9 & 10^8 & \dots & 10^0 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 2 \\ 3 \\ 4 \\ \vdots \\ 11 \end{bmatrix}$$

Tem como solução

$$X = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 1 \end{bmatrix}.$$

Resolva-se no Matlab pelos diversos métodos:

Inversa	Gauss	PPR
-0.0000000000000000	0.00000000000000879	0.0000000000000000
0.0000000000000000	-0.00000000000043309	-0.0000000000000000
-0.0000000000000000	0.0000000000911107	0.0000000000000000
0.0000000000000003	-0.000000010687355	-0.0000000000000001
0.0000000000000002	0.000000076621056	0.0000000000000004
0.0000000000000112	-0.000000346091310	-0.0000000000000012
-0.0000000000000057	0.000000977862633	0.0000000000000018
0.0000000000000405	-0.000001651213377	-0.0000000000000000
1.0000000000000284	1.000001492906150	0.999999999999974
1.0000000000000076	1.0000000000000076	1.0000000000000018

## 6. Factorização de Cholesky

Dada uma matriz  $A$  simétrica definida positiva, a factorização de Cholesky consiste em obter uma matriz  $L$  triangular inferior tal que

$$A = LL^T.$$

Em Matlab, o comando

```
chol(A,'lower')
```

fornece a matriz L. Se se só fizer chol(A), obtém-se a matriz  $L^T$ .

Para resolver o sistema matricial  $AX=B$ , onde

$$A = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 1 & 2 \\ 2 & 2 & 10 \end{bmatrix} \quad X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 6 \\ 14 \\ 42 \end{bmatrix}$$

Ter-se-á que usar os seguintes comandos:

```
L= chol(A,'lower')
Y=subs(L,B)
X=retro(L',Y)
```

## 7. Método de Jacobi

Um método iterativo é um conjunto sequencial de procedimentos que permite obter uma solução aproximada e melhorada do sistema a partir da solução aproximada anterior .

```
function X=jacobi(A,B,P,delta, max1)
% P é a aproximação inicial
% delta é o majorante do erro da última solução aproximada obtida
% max1 é o número máximo de iterações
% Output - X é uma matriz N x 1 que corresponde à solução aproximada do
% do sistema AX=B pelo método de Jacobi.
N = length(B);
for k=1:max1
    for j=1:N
        X(j)=(B(j)-A(j,[1:j-1,j+1:N])*P([1:j-1,j+1:N]))/A(j,j);
    end
    err=abs(norm(X'-P));
    relerr=err/(norm(X)+eps);
    P=X';
    if (err<delta)|(relerr<delta)
        break
    end
end
X=X';
```

Os métodos iterativos que se irão estudar são o método de Jacobi e o método de Gauss-Seidel.

O código para estes métodos foi retirado do file Exchange do Matlab e que se encontra no livro “Numerical Methods Using Matlab” de J.H.Mathews e de K.D.Fink.

## 8. Método de Gauss-Seidel

```
function X=gseid(A,B,P,delta, max1)
N = length(B);
for k=1:max1
    for j=1:N
        if j==1
            X(1)=(B(1)-A(1,2:N)*P(2:N))/A(1,1);
        elseif j==N
            X(N)=(B(N)-A(N,1:N-1)*(X(1:N-1)))/A(N,N);
        else
            X(j)=(B(j)-A(j,1:j-1)*X(1:j-1)-A(j,j+1:N)*P(j+1:N))/A(j,j);
        end
    end
    err=abs(norm(X'-P));
    relerr=err/(norm(X)+eps);
    P=X';
    if (err<delta)|(relerr<delta)
        break
    end
end
X=X';
```

## 9. Exemplo

Resolva-se pelos métodos iterativos o sistema  $AX=B$  onde

$$A = \begin{bmatrix} 4 & -1 & -1 & 0 & 0 & 0 \\ -1 & 4 & 0 & -1 & 0 & 0 \\ -1 & 0 & 4 & -1 & -1 & 0 \\ 0 & -1 & -1 & 4 & 0 & -1 \\ 0 & 0 & -1 & 0 & 4 & -1 \\ 0 & 0 & 0 & -1 & -1 & 4 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 1 \\ 5 \\ 0 \\ 3 \\ 1 \\ 5 \end{bmatrix}$$

A solução deste sistema é

$$X = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}.$$

Use-se a seguinte aproximação inicial

$$X^{(0)} = \begin{bmatrix} 0.25 \\ 1.25 \\ 0 \\ 0.75 \\ 0.25 \\ 1.25 \end{bmatrix}.$$

Jacobi						
k	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
0	0,25	1,25	0	0,75	0,25	1,25
1	0,5625	1,5	0,3125	1,375	0,5625	1,5
2	0,703125	1,734375	0,625	1,578125	0,703125	1,734375
3	0,839844	1,820313	0,746094	1,773438	0,839844	1,820313
4	0,891602	1,90332	0,863281	1,84668	0,891602	1,90332
5	0,94165	1,93457	0,907471	1,91748	0,94165	1,93457
6	0,96051	1,964783	0,950195	1,944153	0,96051	1,964783
7	0,978745	1,976166	0,966293	1,96994	0,978745	1,976166
8	0,985615	1,987171	0,981857	1,979656	0,985615	1,987171
9	0,992257	1,991318	0,987721	1,98905	0,992257	1,991318
10	0,99476	1,995327	0,993391	1,992589	0,99476	1,995327
11	0,997179	1,996837	0,995527	1,996011	0,997179	1,996837
12	0,998091	1,998298	0,997593	1,9973	0,998091	1,998298
13	0,998091	1,998298	0,997593	1,9973	0,998091	1,998298
14	0,998973	1,998848	0,998371	1,998547	0,998973	1,998848
15	0,999305	1,99938	0,999123	1,999017	0,999305	1,99938
16	0,999626	1,99958	0,999406	1,999471	0,999626	1,99958



Gauss-Seidel						
k	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
0	0,25	1,25	0	0,75	0,25	1,25
1	0,5625	1,578125	0,390625	1,554688	0,660156	1,803711
2	0,742188	1,824219	0,739258	1,841797	0,885742	1,931885
3	0,890869	1,933167	0,904602	1,942413	0,959122	1,975384
4	0,959442	1,975464	0,965244	1,979023	0,985157	1,991045
5	0,985177	1,99105	0,987339	1,992359	0,994596	1,996739
6	0,994597	1,996739	0,995388	1,997216	0,998032	1,998812
7	0,998032	1,998812	0,99832	1,998986	0,999283	1,999567
8	0,998032	1,998812	0,99832	1,998986	0,999283	1,999567
9	0,999739	1,999842	0,999777	1,999865	0,999905	1,999943
10	0,999905	1,999943	0,999919	1,999951	0,999965	1,999979
11	0,999965	1,999979	0,99997	1,999982	0,999987	1,999992
12	0,999987	1,999992	0,999989	1,999993	0,999995	1,999997
13	0,999995	1,999997	0,999996	1,999998	0,999998	1,999999
14	0,999998	1,999999	0,999999	1,999999	0,999999	2
15	0,999999	2	0,999999	2	1	2
16	1	2	1	2	1	2