INSTITUTO TECNOLÓGICO DE AERONÁUTICA

CCI-22 - Professor Victor Curtis

Pedro Luchiari de Carvalho - Turma 4

Relatório 3 - Sistemas Lineares

```
format long
A=[3 -0.1 -0.2; 0.1 7 -0.3; 0.3 -0.2 10];
b=[7.85; -19.3; 71.4];

n=size(A,1);
EliminacaodeGauss(A, b, n);
```

```
solução pelo método de Gauss:
x = 3×1
    3.000000000000000
   -2.500000000000000
    7.000000000000002
```

```
Ab=[3 -0.1 -0.2 7.85; 0.1 7 -0.3 -19.3; 0.3 -0.2 10 71.4];
Gauss_Jordan(Ab);
```

```
solução pelo método de Gauss-Jordan:
x = 3×1
    3.000000000000000
   -2.500000000000000
    7.000000000000002
```

```
[L, U] = lu_sempivoteamento(A);
y=L\b;
x=U\y;
disp ('solução pelo método de decomposição LDU:'), x
```

```
solução pelo método de decomposição LDU:
x = 3×1
    3.000000000000000
   -2.500000000000000
    7.000000000000002
```

```
prec=0.0000001;
maxiteracoes=1000;
[X1, dr, k] = Gauss_Jacobi(A,b,prec,maxiteracoes);
fprintf('O número de iterações pelo método de Gauss-Jacobi é: %d\n',k);
```

```
O número de iterações pelo método de Gauss-Jacobi é: 6
```

```
disp('solução pelo método de Gauss-Jacobi:'), X1
```

```
solução pelo método de Gauss-Jacobi:
X1 = 3×1
    3.000000000797412
   -2.499999997999336
    7.000000002352559
```

```
fprintf('O critério de parada foi atendido com: %d\n',dr);
```

```
O critério de parada foi atendido com: 1.244819e-08
```

```
[X1, dr, k]=Gauss_Siedel(A,b,prec,maxiteracoes);
fprintf('O número de iterações pelo método de Gauss-Siedel é: %d\n',k);
```

```
O número de iterações pelo método de Gauss-Siedel é: 5
```

```
disp('solução pelo método de Gauss-Siedel:'); X1
```

```
solução pelo método de Gauss-Siedel:
X1 = 3×1
    2.999999998055569
   -2.500000000456044
    7.000000000049212
```

```
fprintf('O critério de parada foi atendido com: %d\n',dr);
```

```
O critério de parada foi atendido com: 5.063053e-08
```

a) Método de Gauss

```
function x = EliminacaodeGauss(A, b, n)
    for k=1:n-1
        for i=k+1:n
            M = A(i,k)/A(k,k);
            A(i,k) = 0;
        for j=k+1:n
            A(i,j) = A(i,j) - M*A(k,j);
        end
        b(i,1) = b(i,1) - M*b(k,1);
    end
end
x(n,1)=b(n,1)/A(n,n);
for k=n-1:-1:1
    S = 0;
    for j=k+1:n
        S = S + A(k,j)*x(j,1);
    end
    x(k,1) = (b(k,1)-S)/A(k,k);
end
disp ('solução pelo método de Gauss:'), x
end
```

b) Método de Gauss-Jordan

```
function x =Gauss_Jordan(Ab)
[m,n]=size(Ab);
for j=1:m
    for z=2:m
        if Ab(j,j)==0
            t=Ab(1,:);Ab(1,:)=Ab(z,:);
            Ab(z,:)=t;
        end
    end
    for i=j+1:m
        Ab(i,:)=Ab(i,:)-Ab(j,:)*Ab(i,j)/Ab(j,j);
    end
end
for j=m:-1:2
    for i=j-1:-1:1
        Ab(i,:)=Ab(i,:)-Ab(j,:)*Ab(i,j)/Ab(j,j);
    end
end
for s=1:m
    Ab(s,:)=Ab(s,:)/Ab(s,s);
    x(s,1)=Ab(s,n);
end
disp ('solução pelo método de Gauss-Jordan:'), x
end
```

c) Método de Decomposição LU

```
function [L, U] = lu_sempivoteamento(A)
n = size(A, 1);
L = eye(n);
for k = 1 : n
    L(k + 1 : n, k) = A(k + 1 : n, k) / A(k, k);
    for l = k + 1 : n
        A(l, :) = A(l, :) - L(l, k) * A(k, :);
    end
end
U = A;
end
```

d) Método de Gauss Jacobi

```
function[X1, dr, k]= Gauss_Jacobi(A,b,prec,maxiteracoes);
    ord=size(A);
    for i=1:size(b)
```

```
        X(i,1)=b(i,1)/A(i,i);
    end
    k = 1;
    for i = 1:ord
        for j = 1:ord
            if i == j
                C(i,j) = 0;
                g(i,1) = b(i,1)/A(i,i);
            else
                C(i,j) =-A(i,j)/A(i,i);
            end
        end
    end
    while k<=maxiteracoes
        X1=C*X+g;
        for i = 1:ord
            deltax(i,1)=abs(X1(i,1)-X(i,1));
        end
        dr=max(abs(deltax))/max(abs(X1));
        if (dr<prec)
            break;
        end
        X = X1; k = k+1;
    end
end
```

e) Método de Gauss-Siedel

```
function[X1, dr, k]= Gauss_Siedel(A,b,prec,maxiteracoes);
    ord=size(A);
    for i=1:size(b)
        X(i,1)=0;
    end
    k = 1;
    for i = 1:ord
        for j = 1:ord
            if i == j
                C(i,j) = 0;
                g(i,1) = b(i,1)/A(i,i);
            else
                C(i,j) =-A(i,j)/A(i,i);
            end
        end
    end
    while k<=maxiteracoes
        X1 = X;
```

```
        for i = 1:ord
            X1(i,1) = C(i,:)*X1 + g(i,1);
            deltax(i,1)=abs(X1(i,1)-X(i,1));
        end
        dr=max(abs(deltax))/max(abs(X1));
        if (dr<prec)
            break;
        end
        X = X1; k = k+1;
    end
```

Discussão:

Nota-se que, como esperado, os primeiros métodos levam a um cálculo exato da solução, sujeito apenas a erros de processamento da máquina. Os métodos de Gauss-Jacobi e Gauss-Siedel levaram a uma solução aproximada com acurácia de ordem extremamente elevada com um número baixo de iterações, tendo a solução por Gauss-Siedel uma iteração a menos que a solução por Gauss-Jacobi.