MA-202 Numerical Techniques (2022)

B. Tech. II year CSE

Name - Rajan Kumar

Roll No - 202051152

1.)

```
clc;
clear;
close all;
A = [1 1 1; 2 1 3; 3 4 -2];
b =[4;7;9];
% 1.1) Using Gauss Elemination
x = solofLinearEq(A,b);
disp("Solution using Gauss Elemination:");
```

```
Solution using Gauss Elemination:
```

```
disp(x);
```

```
    1
    2
    1
```

```
% 1.2) Using LU Decomposition
x = solofLinearEq(A,b,2);
disp("Solution using LU Decomposition:");
```

```
Solution using LU Decomposition:
```

```
disp(x);
```

```
    1
    2
    1
```

```
% 1.3) Using Gauss elimination + partial pivoting
x = solofLinearEq(A,b,3);
disp("Solution using Gauss elimination + partial pivoting:");
```

```
Solution using Gauss elimination + partial pivoting:
```

```
disp(x);
```

```
    1.0000
    2.0000
    1.0000
```

2.)

```
% 2.1) Using Jacobi method of iteration
A = [1 2 2 1; 2 2 4 2;1 3 2 5;2 6 5 8];
b =[1;0;2;4];
x = Iterative_method_of_LS(A,b,1,1e-3,100);
disp("Solution using Gauss Seidel Method :");
```

```
Solution using Gauss Seidel Method :
```

```
disp(x);
```

```
   1.0e+30 *

   -5.0703
   -1.2677
    7.6056
   -2.5352
```

```
% 2.2) Using Gauss-Siedel method
x = Iterative_method_of_LS(A,b,2,1e-3,100);
disp("Solution using Jacobi Method :");
```

```
Solution using Jacobi Method :
```

```
disp(x);
```

```
   1.0e+54 *

   -1.1841
   -0.9701
   -0.8970
   -0.4485
```

Function Define

```
% 1.
```

```
function fval = solofLinearEq(a,b,choice)
if ~exist('choice','var')
choice = 1;
% By Default method is Gauss Elemination
end
switch choice
case 1
fval = GaussElimination(a, b);
case 2
fval = LUdecomposition(a,b);
case 3
fval = partialpivoting(a, b);
end
end
```

```
% Gauss Elimination
function fval = GaussElimination(A,b)
Ab =[A,b];
n = length(A);
for i =2:n
k = Ab(i,1)/Ab(1,1);
Ab(i,:)= Ab(i,:)-k*Ab(1,:);
end
i = n;
k = Ab(i,2)/Ab(2,2);
Ab(i,:)= Ab(i,:)-k*Ab(2,:);
fval = zeros(n,1);
for i = n :-1:1
fval(i)= (Ab(i,end)-Ab(i,i+1:n)*fval(i+1:n))/Ab(i,i);
end
end

%LU decompostion
function fval = LUdecomposition(A,b)
Ab =[A,b];
n = length(A);
L =eye(n);
for i =2:n
k = Ab(i,1)/Ab(1,1);
L(i, 1)=k;
Ab(i,:)= Ab(i,:)-k*Ab(1,:);
end
i =n;
k = Ab(i,2)/Ab(2,2);
L(i,2)=k;
Ab(i,:)= Ab(i,:)-k*Ab(2,:);
U = Ab(1:n,1:n);
y = inv(L)*b;
fval = inv(U)*y ;
end

%Guass elimination + partial pivoting
function fval = partialpivoting(A,b)
Ab =[A,b];
n = length(A);
col1 = Ab(:,1);
[~,idx]= max(col1);
d =Ab(1,:);
Ab(1,:)=Ab(idx,:);
Ab(idx,:)= d;
for i =2:n
k = Ab(i,1)/Ab(1,1);
Ab(i,:)= Ab(i,:)-k*Ab(1,:);
col2 = Ab(2:end,2);
[~,idx]= max(col2);
d =Ab(2,:);
Ab(2,:)=Ab(idx,:);
Ab(idx,:)= d;
i =3;
```

```
k = Ab(i,2)/Ab(2,2);
Ab(i,:)= Ab(i,:)-k*Ab(2,:);
fval = zeros(n,1);
for i =n :-1:1
fval(i)= (Ab(i,end)-Ab(i,i+1:n)*fval(i+1:n))/Ab(i,i);
end
end
end

% 2
function fval= Iterative_method_of_LS(a,b,choice,tol,maxItr)
switch choice
case 1
fval = gaussSeidel(a, b,tol,maxItr);
case 2
fval = Jacobi(a,b,tol,maxItr);
end
end

%gauss seidel method
function sol= gaussSeidel(A,b,tol,maxitr)
n=length(A);
Xnext=zeros(n,1);
for loop=1:maxitr
Xcurr=Xnext;
for i=1:n
temp_val=0;
for j=1:n
if(i~=j)
temp_val=temp_val+(A(i,j)*Xnext(j));
end
end
Xnext(i)=(b(i)-temp_val)/A(i,i);
end
error=Xnext - Xcurr;
err=norm(error);
if err<=tol
sol=Xnext;
break;
end
end
sol=Xnext;
end

%jacobi method
function f_val= Jacobi(A,b,tol,maxitr)
n=length(A);
Xcurr_val =zeros(n,1);
Xnext_Val=zeros(n,1);
for loop=1:maxitr
for i=1:n
temp=0;
for j=1:n
if(i~=j)
```

```
temp=temp+(A(i,j)*Xcurr_val(j));
end
end
Xnext_Val(i)=(b(i)-temp)/A(i,i);
end
error=Xnext_Val-Xcurr_val;
err=norm(error);
if err<=tol
f_val = Xnext_Val;
break;
end
Xcurr_val=Xnext_Val;
end
f_val=Xnext_Val;
end
```