MA-202 Numericals Techniques Lab-4 Solution of Linear Equations

Hritik Kumar

202051088

Question 1

Gauss elimination

```
A = [1 1 1; 2 1 3; 3 4 -2]; b =[4;7;9]; x
= GaussElemination(A,b); disp("Solution
using Gauss Elemination:");
```

```
Solution using Gauss Elemination:
```

```
disp(x);
```

```
     1
     2
     1
```

LU Decomposition

```
A = [1 1 1; 2 1 3; 3 4 -2]; b =[4;7;9]; x
= LUdecomposition(A,b); disp("Solution
using LU Decomposition:");
```

```
Solution using LU Decomposition:
```

```
disp(x);
```

```
     1
     2
     1
```

Partial Pivoting

```
A = [1 1 1; 2 1 3; 3 4 -2]; b =[4;7;9]; x
= partialpivoting(A,b); disp("Solution
using Partial Pivoting:");
```

```
Solution using Partial Pivoting:
```

```
disp(x);
```

```
    1.0000
    2.0000
    1.0000
```

Question 2

Using Gauss Seidel Method

```
A = [1 2 2 1; 2 2 4 2;1 3 2 5;2 6 5 8]; b
=[1;0;2;4]; x = gaussSeidel(A,b,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
```

Using Jacobi Method

```
A = [1 2 2 1; 2 2 4 2;1 3 2 5;2 6 5 8];
b =[1;0;2;4]; x = Jacobi(A,b,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 for Gauss Elimination

```
function fval =
GaussElemination(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
```

Function for LU Decomposition

```
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
```

Function for PartialPivoting

```
function fval =
partialpivoting(A,b) Ab =[A,b]; n =
length(A); col1 = Ab(:,1);
[dummy,idx]= max(col1); dummy
=Ab(1,:); Ab(1,:)=Ab(idx,:);
Ab(idx,:)= dummy; for i =2:n
 k = Ab(i,1)/Ab(1,1);
Ab(i,:)= Ab(i,:)-k*Ab(1,:);
end col2 = Ab(2:end,2);
[dummy,idx]= max(col2);
dummy =Ab(2,:);
Ab(2,:)=Ab(idx,:);
Ab(idx,:)= dummy; 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
```

Function for Gauss Seidel

```
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=0;  for j=1:n  if(i~=j)
temp=temp+(A(i,j)*Xnext(j));  end  end
Xnext(i)=(b(i)-temp)/A(i,i);  end
```

```
error=Xnext - Xcurr;  err=norm(error);
if err<=tol  sol=Xnext;  break;  end  end
sol=Xnext; end
```

Function for Jacobi Method

```
function fval= Jacobi(A,b,tol,maxitr)
n=length(A);  Xcurr =zeros(n,1);
Xnext=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(j));  end
end  Xnext(i)=(b(i)-temp)/A(i,i);
end  error=Xnext-Xcurr;
err=norm(error);  if err<=tol
fval=Xnext;  break;  end
Xcurr=Xnext;  end  fval=Xnext; end
```