Indian Institute of Information Technology Vadodara

MA202: Numerical Techniques Lab
Semester: IV
Lab 4

Name : Abhiyank Raj Tiwari

Student Id : 201951011

Section : 2A

Course Instructor : Dr Vivek Vyas

Note: I have made PDF from next page using matlab only. They are in parts.

I have merged them all.

Table of Contents

Question-1

```
A = [1 1 1; 2 1 3; 3 4 -2];
b =[4;7;9];
x = GaussElimination(A,b);
disp('Solution of linear system using Gauss Elimination:');
disp(x);
x = LUdecomposition(A,b);
disp('Solution of linear system using LU Decomposition:');
disp(x);
x = partialpivoting(A,b);
disp('Solution of linear system using Gauss Elimination + partial
 pivoting:');
disp(x);
```

1(a): {Gauss Elimination Function}

```
%Using Gauss Elimination
function fval = GaussElimination(A,b)
    %get augumented matrix
    Ab =[A,b];
    %Row Operation
    % Rj =Rj-k(i,j)*Ri where ki,j) = A(j,i)/A(i,i)
    n = length(A);
    %A(1,1) as pivot element
    for i = 2:n
        k = Ab(i,1)/Ab(1,1);
        Ab(i,:)= Ab(i, :)-k*Ab(1,:);
    end
    %A(2, 2) as pivot element
    i =n;
    k = Ab(i,2)/Ab(2,2);
    Ab(i,:)= Ab(i,:)-k*Ab(2,:);
    %A(3,3 ) as pivot element
    %Back-Subsituation
    fval = zeros (n, 1);
    %x(3)=Ab(3,4)/Ab(3,3);
    for i =n :-1:1
        %x(2)= (Ab(2,4)-Ab(2,3)*x(3))/Ab(2,2);
        fval(i)= (Ab(i,end)-Ab(i,i+1:n)*fval(i+1:n))/Ab(i,i);
        %x(1) = (Ab(1,4)-(Ab(1,3)*x(3) +Ab(1,2)*x(2)))/Ab(1.1);
        %x1) = (Ab(1,4)-(Ab(1,1+1:n)*x(1+1:n))/Ab(1,1);
    end
```

```
end
```

Solution of linear system using Gauss Elimination:
1
2
1

1(b): {LU Decompostion Function}

```
%Using LU Decomposition
function fval = LUdecomposition(A,b)
    %get augumented matrix
    Ab =[A,b];
    n= length(A);
    L =eye(n);
    %Row Operation
    % Rj =Rj-k(i,j)*Ri where k(i,j) = A(j,i)/A(i,i)
    %A(1,1) as pivot element
    for i = 2:n
        k = Ab(i,1)/Ab(1,1);
        L(i, 1)=k;
        Ab(i,:)=Ab(i,:)-k*Ab(1,:);
    end
    %A(2,2) as pivot element
    i =n;
    k = Ab(i,2)/Ab(2,2);
    L(i,2)=k;
    Ab(i,:)= Ab(i, :)-k*Ab(2,:);
    %A(3,3 ) as pivot element
    U = Ab (1:n, 1:n);
    y = inv(L)*b;
    fval = inv(U)*y;
end
```

Solution of linear system using LU Decomposition:
1
2
1

1(c): {Gauss Elimination + Partial Pivoting Function}

```
%Gauss elimination + partial pivoting function
function fval = partialpivoting(A,b)
    %get augumented matrix
    Ab =[A,b];
    n = length(A);
    %A(1;1) as pivot element
    % Ensure A(1,1) is largest element in column-1
    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
    %A(2,2) as pivot element
    % Ensure A(2,2) is largest element in column-2
    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,:);
    %A(3,3) as pivot element
    % By Back-Subtituation
    fval = zeros (n, 1); %x(3)=Ab(3,4)/Ab(3,3);
    for i =n :-1:1
fval(i)= (Ab(i, end)-Ab(i,i+1:n)*fval(i+1:n))/Ab(i,i);
end
end
```

```
Solution of linear system using Gauss Elimination + partial pivoting:
    1.0000
    2.0000
    1.0000
```

Published with MATLAB® R2020b

Table of Contents

Question-2

```
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 of linear system using Jacobi Method of iteration:')
disp(x);
x = gaussSeidel(A,b, 1e-3, 100);
disp('Solution of linear system using Gauss Seidel Method :');
disp(x);
```

2(a): {Jacobi Function}

```
%Using Jacobi Method of iteration
function fval= Jacobi(A, b, tol,maxitr)
    % Here co-efficient matrix A must be strictly diagonally dominant
 matrix
    %tol is maximum bearable tolerance in answer
    % maxitr is limit of iterations
    n=length(A);
    Xcurr =zeros(n,1); % assuming initial approximation as zero vector
    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));%This loop calculates #k#j
 a(k.])*x(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
```

Solution of linear system using Jacobi Method of iteration:

```
   1.0e+54 *

   -1.1841
   -0.9701
   -0.8970
   -0.4485
```

2(b): {Gauss Seidel Function}

```
%Using Gauss Seidel Method
function sol= gaussSeidel(A,b, tol,maxitr)
    % Here co-efficient matrix A must be strictly diagonally dominant
 matrix
    % tol is maximum bearable tolerance in answer
    % maxitr is limit of iterations
    n=length(A);
    Xnext=zeros(n,1); % assuming initial approximation as zero vector
    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
```

```
Solution of linear system using Gauss Seidel Method :
   1.0e+30 *

   -5.0703
   -1.2677
    7.6056
   -2.5352
```

Published with MATLAB® R2020b