



**MATLAB Code**

**Matrix Inversion**

B = zeros (12,1);

B (5) =100;

A = [1 -1 -1 -1 -1 0 0 0 0 0 0 0;

0 1 0 0 0 0 0 0 -1 -1 -1 0;

0 0 0 1 0 0 -1 0 0 0 -1 -1;

0 0 0 0 1 -1 -1 -1 0 0 0 0;

1 0 0 0 0 0 0 0 0 0 0 0;

0 0 0 0 1 0 0 -5 0 0 0 0;

0 0 0 1 0 0 1 0 0 0 0 -0.84;

0.7 -1 -1 0 0 0 0 0 0 0 0 0;

0.55 0 0 0 0 0 0 0 -1 0 0 -1;

0 0 0 0 0 0 0 0 0.2 -1 0 0;

0 0.85 0 0 0 0 0 0 -1 0 -1 0;

0 0 0 0 0 3.2 -1 -1 0 0 0 0];

X = A\B;

**Gauss Elimination**

B=zeros(12,1);

B(5,1)=100;

C=[1 -1 -1 -1 -1 0 0 0 0 0 0 0;

0 1 0 0 0 0 0 0 -1 -1 -1 0;

0 0 0 1 0 0 -1 0 0 0 -1 -1;

0 0 0 0 1 -1 -1 -1 0 0 0 0;

1 0 0 0 0 0 0 0 0 0 0 0;

0 0 0 0 1 0 0 -5 0 0 0 0;

0 0 0 1 0 0 1 0 0 0 0 -0.84;

0.7 -1 -1 0 0 0 0 0 0 0 0 0;

0.55 0 0 0 0 0 0 0 -1 0 0 -1;

0 0 0 0 0 0 0 0 0.2 -1 0 0;

0 0.85 0 0 0 0 0 0 -1 0 -1 0;

0 0 0 0 0 3.2 -1 -1 0 0 0 0];

A = [C B];

[r,c]=size(A);

% sin -> whether the matrix is singular(sin=1) or non-singular(sin=0)

singular=false;

for i=1:r

% finding the i-th pivot:

% partial pivoting:

if(i<r)% do partial pivoting only if there are any row below the current row

imax=i; %index of the element with maximum value

max=A(i,i); %value of that element

for k=i+1:r

% finding the max

if abs(A(k,i))>abs(max)

max=A(k,i);

imax=k;

end

end

%swap the rows

A([i,imax],:)=A([imax,i],:);

end

if A(i,i)==0

% matrix is singular

singular=true;

end

% do for all remaining elements in current row

for j=i+1:r

A(j,:)=A(j,:)-A(i,:)\*A(j,i)/A(i,i);

A(j,i)=0; % fill lower triangular matrix with zeros

end

end

% if matrix is non-singular

if singular==false

sol=zeros(r,1);% solution array

% backward susbstitution

for i=r:-1:1

s=A(i,c);% s-> it will become the value of x(i)

for j=r:-1:i+1

s=s-A(i,j)\*sol(j,1);% this value needs to be removed from s

end

sol(i,1)=s/A(i,i);% divide by coeff of x(i)

end

else

disp('Matrix is Singular.');

end



