```
n = 2;                          % the number of equations
A=[10^(-20) 1; 1 1];      %define the 3x3 matrix A
b=[2;1];                        %define the column vector b
B = [A b];                         % 3x4 matrix
L = eye(2);                        % 3x3 identity matrix
P=eye(n);                          %3x3 matrix for permutation matrix P

for k=1:n-1                     % step number (and row to be multipied)
    display('Step')
    k
    %%%%%Partial pivoting strategy
    [t,r]=max(abs(B(k:n,k))); %stores the maximum absolute value of b_{ik}
                           %for i=k,...n
                           %r stores the index of the element t
     %row number is offset by k-1
     r=k-1+r;
     %interchange row (B(k,:)) and row (B(r,:)) of B
     B([k r],:)=B([r k],:);
     %interchange row (P(k,:)) and row (P(r,:)) of P
     P([k r],:)=P([r k],:);
     %interchange L(k,1:k-1) and L(r,1:k-1)
     L([k r],1:k-1)=L([r k],1:k-1);
     %%%%%%%%%end of partial pivoting strategy
     for i=k+1:n %row number to be changed
        L(i,k)=B(i,k)/B(k,k);  % this is the multiplier
        B(i,:)= B(i,:)-L(i,k)*B(k,:)   % row operation
    end
    display('Matrix after the k-th elimination step is ')
    B

end

display('Upper triangular matrix U= ')
U=B(:,1:n) %Upper triangular matrix U

display('Lower triangular matrix L= ')
L %Lower triangular matrix L

display('LU=')
L*U

%perform backward substitution
x=B(:,n+1);
x(n)=B(n,n+1)/B(n,n);
for i=n-1:-1:1
    x(i)=(B(i,n+1)-B(i,i+1:n)*x(i+1:n))/B(i,i);
end
%solution
display('The computed solution is x=')
x
```

```
Step

k =
```

```
     1


B =

     1     1     1
     0     1     2

Matrix after the k-th elimination step is

B =

     1     1     1
     0     1     2

Upper triangular matrix U=

U =

     1     1
     0     1

Lower triangular matrix L=

L =

    1.0000         0
    0.0000    1.0000

LU=

ans =

    1.0000    1.0000
    0.0000    1.0000

The computed solution is x=

x =

    -1
     2
```