```
clc;
TOTAL = zeros(10,4);
for i = 1:10
    A = hilb(8);
    XT = randn(8,4);
    B = A *XT;
    [L,U,P] = lupp(A);
    X = zeros(8,4);
    for k = 1:4
        y = L\(P*B(:,k));
        X(:,k) = U\y;
    end
    [XF, CONDITIONAL_A] = conditional(X, XT, A);
    RR = residual(B, A, X);
    TOTAL(i,:) = [XF, CONDITIONAL_A, RR, eps];
end
format rational
T = array2table(TOTAL,"VariableNames",["XF","CONDITIONAL_A","RR","EPSILON"],"RowNames",↙
["Loop 1","Loop 2","Loop 3","Loop 4","Loop 5","Loop 6","Loop 7","Loop 8","Loop 9","Loop↙
10"]);
disp(T);


function  [XF,FA]  = conditional(X, XT, A)
    XF = norm(X-XT,"fro")/norm(XT,"fro");
    FA = eps * cond(A,"fro");
end


function  FR  = residual(B, A, X)
    FR = norm(B-A*X,"fro")/(norm(A,"fro")*norm(X,"fro"));
end


function [L,U,P] = lupp(A)
% lupp.m LU factorization with partial pivoting
% input: A is an n x n nonsingular matrix
% output: Unit lower triangular L, upper triangular U,
% permutation matrix P such that PA = LU
%
n = size(A,1);
P = eye(n);
for k = 1:n-1
    [maxval, maxindex] = max(abs(A(k:n,k)));
    q = maxindex + k - 1;
    if maxval == 0, error('A is singular'), end
    A([k,q],:) = A([q,k],:);
    P([k,q],:) = P([q,k],:);
    i = k+1:n;
    A(i,k) = A(i,k)/A(k,k);
    A(i,i) = A(i,i) - A(i,k)*A(k,i);
```

```
end
L = tril(A,-1) + eye(n);
U = triu(A);
end
```