KUSHAL AGARWAL
2201mc22_kushal@iitp.ac.in
6287173664
28-01-24

LAB 2 (TASK 1)

```
%We have already created the function luSelfnP ...

%Now, we are using that function and trying to evaluate the solution of the

% give equations

function[X]=Q1(n)

A=ones(n,n);  % Creating a matrix with all ones

[m ,n]=size(A);

% A and B are as given in the Question

B=[1;0];

A(1,1)=power(10,-15);       % Changing the values of A(1,1) as mentioned in

question


[L,U]=luSelfnP(A)     % Here, we are using the luselfnP function and

evaluating L and U

 Y=FdSubs(L, B);

 X=BdSubs(U, Y);

 disp('Using A\B');

 display(A\B);

 disp('Using LU');

end

% conclusion is that MATLAB can work well up to the 16th place of decimal
```

LAB 2 (TASK 2)

```
% Task 2

clc , clearvars;

%We have already created the function luSelfwP …

%Now, we are using that function and trying to evaluate the solution of the

given equations

function[X]=Q2(A)

[n,n]=size(A);

B=[1;0];

[P,L,U]=luSelfwP(A);    % evaluating P L U using the function luSelfwP function

B=P*B;

Y=FdSubs(L, B);

X=BdSubs(U, Y);


P, L, U

% This is run in the terminal by giving an input
```

LAB 2 (TASK 3)

```
% Task 3

clc , clearvars;

% Running a for so that the value of n can be varied from 20 to 100

for n=20:20:100

A=rand(n,n);   % Generating a square matrix using rand

A(1,1)=power(10,-20);

%Here, I am calculating L and U using the luSelfnP function, which we have

%created previously

[L,U]=luSelfnP(A);

% After getting L and U now, we are calculating the norm of A-LU

% N = norm(V) returns the 2-norm or Euclidean norm of the vector V and is

 % the same as norm(V,2).

val1=norm(A - L*U);

val1

% Here I am trying to calculate P,L, and U using the luSelfwp function

[P,L,U]=luSelfwP(A);

val2=norm(P*A - L*U);

val2

% Here I am trying to calculate P, L U using the inbuilt Matlab function lu

[P,L,U]=lu(A);

val3=norm(P*A - L*U);

val3

end

% This is run in the terminal by giving an input
```

LAB 2 (TASK 4)

```
% Task - 4

clc,clearvars;

%I have created a function named rref2, and now I am trying to use it as per

%instruction given in the question

% Define initial values for m and n

m = 4;

n = 3;

% Loop to iterate through different matrix sizes

for i = 1:5

   % Generate a random matrix A

   A = rand(m,n);

   % Use the in-built MATLAB function to calculate reduced row echelon form

(rref)

   A_rref = rref(A);

   % Display the result

   disp('Matrix A in Reduced Row Echelon Form (Using MATLAB rref):');

   disp(A_rref);

   % Increment the matrix size for the next iteration

   m=m+1;

   n=n+1;

end
```