**1.**

**LU**

function [L,U] = mylu(A)

n = size(A,1);

L = zeros(n);

for j = 1:n-1

    for i = j+1:n

        mult = A(i,j)/A(j,j);

        A(i,j+1:n) = A(i,j+1:n) - mult\*A(j,j+1:n);

        A(i,j) = 0;

        L(i,j) = mult;

    end

end

for i = 1:n

    L(i,i) = 1

End

U = A;

**Usolve**

function x = usolve(U, y)

n = length(y);

x = zeros(n,1);

for i = n:-1:1

  x(i) = y(i);

  for j = i+1:n

    x(i) = x(i) - U(i,j) \* x(j);

  end

  x(i) = x(i) / U(i,i);

end

**Lsolve**

function y = lsolve(L, b)

n = length(b);

y = zeros(n,1);

for i = 1:n

  y(i) = b(i);

  for j = 1:i-1

    y(i) = y(i) - L(i,j) \* y(j);

  end

end

OUTPUT:

A =

1 2 3

4 5 6

7 8 0

B = [1 0 2]’

>> [L U] = mylu(A)

L =

1 0 0

4 1 0

7 2 1

U =

1 2 3

0 -3 -6

0 0 -9

>> y = lsolve(L,B)

y =

1

-4

3

>> x = usolve(U,y)

x =

-2.0000

2.0000

-0.3333

>> A\B

ans =

-2.0000

2.0000

-0.3333

**3.c**

function [a, b] = gauss(A,B)

n = length(B);

a = A;  b = B;

for col = 1:n

  for row = (col+1):n

    mult = a(row,col) / a(col,col);

    a(row,:) = a(row,:) - mult \* a(col,:);

    b(row) = b(row) - mult \* b(col);

  end

end

**output:**

>> vpa(A)

ans =

[ 0.0000000000000001, 1.0]

[ 1.0, 1.0]

>> [mya myb] = gauss(A,b)

mya =

1.0e+16 \*

0.0000 0.0000

0 -1.0000

myb =

1.0e+16 \*

0.0000

-2.0000

>> vpa(mya)

[ 0.0000000000000001, 1.0]

[ 0, -10000000000000000.0]

>> vpa(myb)

2.0

-19999999999999996.0

since ma is an upper triangle matrix, we can use usolve

>> usolve(ma,mb)

>> x = usolve(mya,myb)

x =

4.4409

2.0000

>> A\b

ans =

1.0000

2.0000

Obviously without partial pivoting, the result is not so accurate. In addition, the relative error is really large because the floating point error.

**Lu with partial pivoting:**

function [L,U,piv] = mylu(A,b)

n = size(A,1);

L = zeros(n);

piv =[1:n];

for j = 1:n-1

      max\_abs = 0;

  pivot = j;

  for i = j:n

    if abs(A(i,j)) > max\_abs

      max\_abs = abs(A(i,j));

      pivot = i;

    end

  end

  %swap

  temp = A(j,:);

  A(j,:) = A(pivot,:);

  A(pivot,:) = temp;

  btemp = b(j);

  b(j) = b(pivot);

  b(pivot) = btemp;

  temppiv = piv(j);

  piv(j) = piv(pivot);

  piv(pivot) = temppiv;

    for i = j+1:n

        mult = A(i,j)/A(j,j);

        A(i,j+1:n) = A(i,j+1:n) - mult\*A(j,j+1:n);

        A(i,j) = 0;

        L(i,j) = mult;

    end

end

for i = 1:n

    L(i,i) = 1;

end

U = A;

>> [l u piv b] = mylu(A,b)

l =

1.0000 0 0

0.5714 1.0000 0

0.1429 0.5000 1.0000

u =

7.0000 8.0000 0

0 0.8571 3.0000

0 0 4.5000

piv =

3 1 2

b =

3

2

>> y = lsolve(l,b)

y =

3.0000

2.0000

>> x = usolve(u,y)

x =

1

2

**while using the built-in**

>> [l u p] = lu(A)

l =

1.0000 0 0

0.1429 1.0000 0

0.5714 0.5000 1.0000

u =

7.0000 8.0000 0

0 0.8571 3.0000

0 0 4.5000

p =

0 0 1

1 0 0

0 1 0

So my implementation is correct.