Labwork 3

V. Gauss elimination

Exercise 1:

Solve the system of linear equation:

$$\begin{cases} 2x + y - z = 8 \\ -3x - y + 2z = -11 \\ -2x + y + 2z = -3 \end{cases}$$

a. using *inv* or *mldivide* in Matlab

b. Use the function *GaussPivot.m*

```
function x = GaussPivot(A,b)
% GaussPivot: Gauss elimination pivoting
% x = GaussPivot(A,b): Gauss elimination with pivoting.
% input:
% A = coefficient matrix
% b = right hand side vector
% output:
% x = solution vector
[m,n]=size(A);
if m~=n, error('Matrix A must be square'); end
nb=n+1;
Aug=[A b];
% forward elimination
for k = 1:n-1
    % partial pivoting
    [big,i]=max(abs(Aug(k:n,k)));
    ipr=i+k-1;
    if ipr~=k
        Aug([k,ipr],:)=Aug([ipr,k],:);
    end
    for i = k+1:n
        factor=Aug(i,k)/Aug(k,k);
        Aug(i,k:nb)=Aug(i,k:nb)-factor*Aug(k,k:nb);
    end
end
% back substitution
x=zeros(n,1);
x(n)=Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
    x(i)=(Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
```

c. Modify the code above to Gauss Elimination without Pivoting. Explain the code

VI. LU Decomposition

Exercise 2: LU decomposition

Solve the following set of equations with LU factorization with pivoting

$$\begin{cases} 2x + y - z = 8 \\ -3x - y + 2z = -11 \\ -2x + y + 2z = -3 \end{cases}$$

a. Using function *lu* in Matlab
b. Use the function *LU_pivot.m* code

```
function [L, U, P] = LU_pivot(A)
[n, n1] = size(A); L=eye(n); P=eye(n); U=A;
for j = 1:n
  [pivot m] = max(abs(U(j:n, j)));
  m = m+j-1;
  if m ~= j
    U([m,j],:) =  U([j,m], :);   % interchange rows m and j in U
    P([m,j],:) =  P([j,m], :);   % interchange rows m and j in P
    if j >= 2;    % very_important_point
      L([m,j],1:j-1) =  L([j,m], 1:j-1);   % interchange rows m
and j in columns 1:j-1 of L
    end;
  end
  for i = j+1:n
    L(i, j) = U(i, j) / U(j, j);
    U(i, :) =  U(i, :) - L(i, j)*U(j, :);
  end
end
```

c. Change the code of LU with pivoting in *LU_pivot.m* to LU non pivoting code
d. Comparison and explain the difference between *LU_pivot.m* and *mylup.m* code:

```
function [ L, U, P ] = mylup( A )
%% LU DECOMPOSITION WITH PIVOTING
[n,m] = size(A);
C = zeros(n, n+1);

% initialize
C(1:n, 1:n) = A;
C(1:n, n+1) = (1:n)';

% main algorithm
for j=1:n-1
    if (sum(C(1:n,j)) == 0)
        error('mylup:input', 'The solution is not determined.');
    end
    [~, i] = max(abs(C(1:n, j)));

    if (i ~= j)
        tt = C(i, 1:n+1);
        C(i, 1:n+1) = C(j, 1:n+1);
        C(j, 1:n+1) = tt;
    end

    for i = j+1:n
        t = C(i, j) / C(j, j);
        C(i, j) = t;
        C(i, j+1:n) = C(i, j+1:n) - t*C(j, j+1:n);
    end
end
```

```
% refine the results
PP = C(1:n, n+1);
C = C(1:n, 1:n);
L = tril(C, -1);
U = triu(C);
P = zeros(n, n);

for i=1:n
    L(i,i)=1;
    P(i, PP(i)) = 1;
end
end
```