Numerical Computation - Assignment 5

薛劭杰　　1930026143

Q1.

Let $A = \begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 3 & 4 & 4 \\ 0 & 2 & 1 & -1 \end{bmatrix}$, and decompose A into LU.

$\Rightarrow \begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 0 & 4 & 3 & 2 \\ 0 & 2 & 1 & -1 \end{bmatrix}$ by $R_3 - R_1$. $E_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$\Rightarrow \begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$ by $R_3 - 2R_1$ and $R_4 - R_1$ $E_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -2 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$

$U = \begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$ and $U = E_2 * E_1 * A \Rightarrow A = (E_1)^{-1} * (E_2)^{-1} * U$

So, $L = (E_1)^{-1} * (E_2)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$.

Q2.

Let $A = \begin{bmatrix} 4 & 2 & 0 \\ 4 & 4 & 2 \\ 2 & 2 & 3 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$, and decompose A into LU.

$\Rightarrow \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 1 & 3 \end{bmatrix}$ by $R_2 - R_1, R_2 - \frac{1}{2}R_1$. $E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix}$.

$\Rightarrow \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 2 \end{bmatrix}$ by $R_3 - \frac{1}{2}R_2$. $E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}$.

$U = \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 4 \end{bmatrix}$ and $U = E_2 * E_1 * A \Rightarrow A = (E_1)^{-1} * (E_2)^{-1} * U$

So, $L = (E_1)^{-1} * (E_2)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$

Then $Ax = b \Rightarrow LUx = b$. Let $Ux = z$, then $Lz = b$.

$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} * z = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$. Initialize them to a matrix $\begin{bmatrix} 1 & 0 & 0 & 2 \\ 1 & 1 & 0 & 4 \\ \frac{1}{2} & \frac{1}{2} & 1 & 6 \end{bmatrix}$

$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & \frac{1}{2} & 1 & 5 \end{bmatrix}$ by $R_2 - R_1$ and $R_3 - \frac{1}{2}R_1$.

$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 4 \end{bmatrix} \text{ by } R_3 - \frac{1}{2}R_2.$$

Use the back substitution: $z_3 = 4, z_2 = 2, z_1 = 2$

$$Ux = z \Rightarrow \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 2 \end{bmatrix} * x = \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix}. \text{ Initialize them to a matrix } \begin{bmatrix} 4 & 2 & 0 & 2 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 4 & 4 \end{bmatrix}.$$

Use the back substitution: $x_3 = 2, x_2 = \frac{2-(2*2)}{2} =- 1, x_1 = \frac{2-2*(-1)}{4} = 1,$

Q3.

```
function [L, U, z, x] = lu_guass(A, b)

[n, m] = size(A);
% The number of rows should be equal to columns
if n ~= m
    error(1)
end
for k = 1:n-1
    if A(k, k) == 0
        error(1)
    end
    for i = k+1:n
        m = A(i, k)/A(k, k);
        A(i, k) = m; % lower triangle
        for j=k+1:n
            A(i, j) = A(i, j)-m*A(k, j);
        end
    end
end
L = eye(n) + tril(A, -1); % replace the diagnose with identity matrix
U = triu(A);

Z = [L b];
z = zeros(n, 1);
for i=1:n
    tmp = Z(i,n+1);
    for j = 1:i-1
        tmp = tmp-Z(i,j)*z(j,1);
    end
    z(i,1)= tmp/Z(i,i);
end

R = [U z]
x = zeros(n, 1)
for i=n:-1:1
    tmp = R(i,n+1);
    for j = n:-1:i+1
        tmp = tmp-R(i,j)*x(j,1)
    end
    x(i,1)= tmp/R(i,i);
end
```

The code in the red box is modification (addition) part.

Q4.

```
function Inv = inverse(A)

[n, m] = size(A);
% The number of rows should be equal to columns

if n ~= m
    error(1)
end

A_I = [A eye(n)];
[r c] = size(A_I);
for k = 1:n-1
    if A_I(k, k) == 0
        error(1)
    end
    for i = k+1:n
        a = A_I(i, k)/A_I(k, k)
        i,k
        for j=k:c
            A_I(i, j) = A_I(i, j)-a*A_I(k, j);
        end
    end

```

```
    for k = r:-1:1
        if A_I(k, k) == 0
            error(1)
        end
        for i = k:c
            A_I(k, i) = A_I(k, i)/A_I(k, k);
        end
        for j = k-1:-1:1
            tmp = A_I(j, k)/A_I(k,k);
            for l=k:c
                A_I(j,l)=A_I(j,l)-tmp*A_I(k,l)
            end
        end
    end
end

Inv=A_I(:, c/2+1:c)
end
```

```
function x = inverse_solve(A, b)
if det(A)==0
    error('A is not consistent!')
end
[n m] = size(A);
if n~=m
    error(1)
end
inv = inverse(A);
x = inv*b
end
```

Q5.

The result of the $lu\_guass()$ method in question 3:

```
>> [l u z x]=lu_guass(A, b)

l =

    1.0000         0         0         0         0
    0.5000    1.0000         0         0         0
         0   -0.5714    1.0000         0         0
    0.5000   -1.0000   -4.6667    1.0000         0
    3.0000    0.2857   33.3333   -7.3333    1.0000


u =

    2.0000    1.0000   -5.0000    1.0000    4.0000
         0   -3.5000    2.5000   -6.5000         0
         0         0    0.4286   -1.7143   17.0000
         0         0         0   -9.0000   77.3333
         0         0         0         0   43.4444


z =

    8.0000
    5.0000
   -2.1429
   -9.0000
   -8.0000


x =

    4.7801
   -0.3649
   -0.0247
   -0.5823
   -0.1841
```

The result of the $inverse\_solve()$ method in question 4:

```
>> A=[2 1 -5 1 4; 1 -3 0 -6 2; 0 2 -1 2 17; 1 4 -7 6 0; 6 2 0 10 55]

A =

     2     1    -5     1     4
     1    -3     0    -6     2
     0     2    -1     2    17
     1     4    -7     6     0
     6     2     0    10    55

>> b=[8; 9; -5; 0; 12]

b =

     8
     9
    -5
     0
    12

>> x = inverse(A)

x =

    3.6573   -2.3478    0.1355   -2.6317   -0.2225
    4.6718   -3.1739    0.7442   -3.4433   -0.4544
    1.6436   -1.2174    0.1509   -1.3384   -0.1219
   -1.8065    1.0870   -0.3427    1.3393    0.1978
   -0.2404    0.1739    0.0205    0.1688    0.0230
```

Q6.

```
>> A=[1 2 3 4 5;2 3 4 5 6;3 4 5 6 7;4 5 6 7 8;5 6 7 8 9]

A =

     1     2     3     4     5
     2     3     4     5     6
     3     4     5     6     7
     4     5     6     7     8
     5     6     7     8     9

>> inv(A)
警告: 矩阵接近奇异值，或者缩放错误。结果可能不准确。RCOND =  1.874961e-18。

ans =

   1.0e+15 *

  -2.2001    3.3076   -1.2403    1.3585   -1.2256
   3.3076   -4.4888    1.6833   -3.1304    2.6283
  -1.2403    1.6833   -1.7571    3.4257   -2.1115
   1.3585   -3.1304    3.4257   -2.8941    1.2403
  -1.2256    2.6283   -2.1115    1.2403   -0.5316

>> inverse(A)
错误使用 error
信息必须指定为字符矢量或信息结构体。

出错 inverse (line 14)
        error(1)
```

The result of direct $inv(A)$ and the function in Q4 $inverse(A)$ are not same. Because in the $inverse()$, we determine whether the input matrix is consistent or not. In the print screen we can see it catch this error. So, there are some differences between them.