实验六 线性方程组的直接法实验

一. 实验目的

（1）深入理解线性方程组的直接法的设计思想，掌握不同方法的矩阵分解手续，以及解决某些实际的线性方程组求解问题。

（2）熟悉 Matlab 编程环境，利用 Matlab 解决具体的线性方程组求解问题。

二. 实验要求

用 Matlab 软件实现线性方程组求解的高斯选主元素法、矩阵分解法、Cholesky 法和追赶法，并用实例在计算机上计算。

三. 实验内容

1. 实验题目

3-1：利用高斯选主元素法求解下列方程组：

$$\begin{cases} x_1+x_2+\quad\quad x_4 = 2 \\ 2x_1+x_2-3x_3+x_4 = 1 \\ 4x_1-x_2-2x_3+2x_4 = 0 \\ 3x_1-x_2-\quad x_3+2x_4 = -3 \end{cases}$$

3-2：用杜利特尔分解法求解方程组：

$$\begin{bmatrix} 2 & 2 & 3 \\ 4 & 7 & 7 \\ -2 & 4 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} 3 \\ 1 \\ -7 \end{bmatrix}$$

3-3：用 Cholesky 方法求解方程组：

$$\begin{cases} 4x_1-2x_2+4x_3=10 \\ -2x_1+17x_2+10x_3=3 \\ 4x_1+10x_2+9x_3=-7 \end{cases}$$

3-4：用追赶法求解方程组：

$$\begin{cases} 136.01x_1+90.860x_2 = -33.254 \\ 90.860x_1+98.810x_2-67.590x_3 = 49.709 \\ -67.590x_2+132.01x_3+46.260x_4 = 28.067 \\ 46.260x_3+177.17x_4 = -7.3244 \end{cases}$$

2. 设计思想

要求针对上述题目，详细阐述每种方法的设计思想。

（1） 利用高斯选主元素法

对于所要消去的第 i 个元素，首先比较同列的还有待消去的其它行中元素的值，选择其中最大的换到改行，再继续消元，而后回代；消元和回代的思想为，通过将一个方程乘以或除以某个常数，以及将两个方程相加减这两种手续，逐步消去方程 中的变元，而将所给方程加工成便于求解的三角方程组乃至对角方程组的形式；

（2）矩阵分解法

将系数矩阵分解为下三角阵和上三角阵的乘积则 L(Ux)=b 可化归为两个三角方程组 Ly=b，Ux=y 这两个方程组都是容易求解的，用追和赶的过程便可得到最终的结果；

（3）Cholesky 方法

同样是将系数矩阵分解，但分解为三个矩阵　A=LDL’ 其中 D 是对角阵，L 是单位下三角阵，通过适当的方法可以求出 D、L，然后通过　解两个方程组：Ly=b、L’x=y/D　　即可得出最终的结果，并且相比平方根方法，该方法分解确实不在含有开放运算；

（4）追赶法

无论是消元过程还是回代过程，它们都是规模缩减技术的具体运用；设计机理是将所给的三对角方程组化归为简单的二对角方程组来求解，来达到化繁为简的目的。

3. 对应程序

列出每种方法的程序。

```
function x = Gauss_pivot(A,b)
n=length(b);
x=zeros(n,1);
c=zeros(1,n);
d1=0;
for i=1:n-1
    max = abs(A(i,i));
    m = i;
    for j=i+1:n
        if max<abs(A(j,i))
            max=abs(A(j,i));
            m=j;
        end
    end
    if(m~=i)
        for k=i:n
            c(k)=A(i,k);
            A(i,k)=A(m,k);
            A(m,k)=c(k);
        end
        d1=b(i);
        b(i)=b(m);
        b(m)=d1;
    end
    for k=i+1:n
        for j=i+1:n
            A(k,j)=A(k,j)-A(i,j)*A(k,i)/A(i,i);
        end
        b(k)=b(k)-b(i)*A(k,i)/A(i,i);
        A(k,i)=0;
    end
end
x(n)=b(n)/A(n,n);
for i=n-1:-1:1
    sum=0;
    for j=i+1:n
```

```
        sum = sum + A(i,j)*x(j);
    end
    x(i) = (b(i)-sum)/A(i,i);
end
function x = Chol_decompose(A,b)
N=length(A);
L=zeros(N,N);
D=zeros(1,N);
for i=1:N
    L(i,i)=1;
end
D(1)=A(1,1);
for i = 2:N
    for j=1:i-1
        if j==1
            L(i,j)=A(i,j)/D(j);
        else
            sum1=0;
            for k=1:j-1
                sum1=sum1+L(i,k)*D(k)*L(j,k);
            end
            L(i,j)=(A(i,j)-sum1)/D(j);
        end
    end
    sum2=0;
    for k=1:i-1
        sum2=sum2+L(i,k)^2*D(k);
    end
    D(i)=A(i,i)-sum2;
end
y=zeros(1,N);
y(1)=b(1);
for i=2:N
    sumi=0;
    for k=1:i-1
        sumi=sumi+L(i,k)*y(k);
    end
    y(i)=b(i)-sumi;
end

x=zeros(1,N);
x(N)=y(N)/D(N);

for i=N-1:-1:1
    sumi=0;
    for k=i+1:N
        sumi=sumi+L(k,i)*x(k);
```

```
    end
    x(i)=y(i)/D(i)-sumi;
end
function x =lu_decompose(A,b)
n=length(b);
L=eye(n);U=zeros(n,n);
x=zeros(n,1);y=zeros(n,1);
for i=1:n
    U(1,i)=A(1,i);
    if i==1
        L(i,1)=1;
    else
        L(i,1)=A(i,1)/U(1,1);
    end
end
for i=2:n
    for j=i:n
        sum=0;
        for k=1:i-1
            sum=sum+L(i,k)*U(k,j);
        end
        U(i,j)=A(i,j)-sum;
        if j~=n
            sum=0;
            for k=1:i-1
                sum = sum + L(j+1,k)*U(k,i);
            end
            L(j+1,i)=(A(j+1,i)-sum)/U(i,i);
        end
    end
end

y(1)=b(1);
for k=2:n
    sum=0;
    for j=1:k-1
        sum = sum + L(k,j)*y(j);
    end
    y(k) = b(k)-sum;
end
x(n)=y(n)/U(n,n);
for k=n-1:-1:1
    sum=0;
    for j=k+1:n
        sum = sum+U(k,j)*x(j);
    end
    x(k)=(y(k)-sum)/U(k,k);
```

```
end
function x= threedia(a,b,c,f)
N=length(f);
x=zeros(1,N);y=zeros(1,N);
d=zeros(1,N);u=zeros(1,N);
d(1)=b(1);
for i=1:N-1
    u(i)=c(i)/d(i);
    d(i+1)=b(i+1)-a(i+1)*u(i);
end

y(1)=f(1)/d(1);

for i=2:N
    y(i)=(f(i)-a(i)*y(i-1))/d(i);
end

x(N)=y(N);
for i=N-1:-1:1
    x(i)=y(i)-u(i)*x(i+1);
end

A=[1,1,0,1;
    2,1,-3,1;
    4,-1,-2,2;
    3,-1,-1,2];
b = [2,1,0,-3]';
x = Gauss_pivot(A,b);
disp("高斯选主元素法:")
disp(x)
A=[2,2,3;
    4,7,7;
    -2,4,5];
b=[3,1,-7]';
x=lu_decompose(A,b);
disp("杜利特尔分解法:")
disp(x)

A=[4,-2,4;
    -2,17,10;
    4,10,9];
b=[10,3,-7]';
x=Chol_decompose(A,b)';
disp("Cholesky 方法:")
disp(x)

a=[0,90.860,-67.590,46.260];
```

```
b=[136.01,98.810,132.01,177.17];
c=[90.860,-67.590,46.260,0];
f=[-33.254,49.709,28.067,-7.3244];
disp("追赶法:")
disp(threedia(a,b,c,f)')
```

4. 实验结果

列出相应的运行结果截图。

```
高斯选主元素法:
    5.000000000000000
    3.333333333333333
    2.000000000000000
   -6.333333333333334

杜利特尔分解法:
     2
    -2
     1

Cholesky方法:
   -5.156250000000000
   -3.812500000000000
    5.750000000000000

追赶法:
    1.0e+03 *

   -2.954703119253691
    4.422583284720388
    2.492697439677929
   -0.650897488059497
```

四. 实验分析

对实验过程进行分析总结，指出每种方法的设计要点及应注意的事项，以及通过实验所获得的对线性方程组求解问题的各种解法的理解。

实验过程中重要的是对公式的理解，求解的过程顺序很重要，在看懂代码的基础上，再打代码，这往往也不会出现一些小的错误，比如少写或错写代码等。

在矩阵分解方法中采用的是 Doolittle 分解，再者在编写代码时并不一定要按公式死死地写代码，可以进行一定的灵活改变，比如矩阵的对角元素是 1 时，可以直接定义为单位阵；在循环中需要对两个矩阵进行处理（L、U），当 L 的前两行已经知道，而 U 只知道一行，此时知道的不必再插入循环中进行求解，进行适当的代码处理，直接处理从 U 的第二行，L 的第三行开始处理； 通过实验，很好地复习了第六章的内容，深入地理解了线性方程组的直接法，设计的过程都是要把方程组直接加工成某个三角方程组乃至

对角方程组，而这些方程组可以通过一定的技巧来解出，从而能够比较精确和迅速地得到方程的解。

（注：不要改变实验报告的结构，写清页码和题号，源程序以自己的姓名命名，如3-1题可命名为“zhangsan_3-1.m”，运行截图中应出现自己的姓名和题号）