《计算方法》上机作业报告

学生信息：

|  |  |  |
| --- | --- | --- |
| 姓名：何宇昊 | 学号：3190102182 | 授课教师：郑太英 |

## 上机作业内容：

**列主元高斯消去法与三角分解法**

## 算法目的和意义

**算法目的与介绍**

高斯消去法是一个古老的求解线性方程组的方法，基本思路是用初等行变换将线性方程组AX=B中的矩阵A化为上三角阵，再回代求解。由消元过程和回代过程求解线性方程组的方法称为高斯消去法，虽然他思路简单，但由它改进、变形得到的选主元素消去法、三角分解法仍然是目前计算机上常用的有效方法**。**

**主元算法的意义**

用高斯消去法解Ax=b时，其中设A为非奇异矩阵，可能出现的情况，这时必须进行带行交换的高斯消去法。但在实际计算中即使值很小时，用作除数，会导致中间结果矩阵A(k)元素数量级严重增长和舍入误差的扩散，使得最后的计算结果不可靠。因此对一般矩阵方程组，需要引进选主元的技巧，即在高斯消去法的每一步应该选取系数矩阵或消元后的低阶矩阵中绝对值最大的元素作为主元素，保持乘数，以便减少计算过程中舍入误差对计算解的影响。

完全主元素消去法是解低阶稠密矩阵方程组的有效方法，但完全主元素消去法在选取主元时要花费一定的计算机时间。（而且交换了未知数的位置），因此在实际应用中，我们常采取列主元进行消去，即是每次选主元时，仅依次按列选取绝对值最大的元素作为主元素。

**三角分解法的意义**

由于行变换以及行之间相减相当于左乘初等矩阵，因此实际上高斯消去法就是将系数矩阵分解为一个下三角矩阵和一个上三角矩阵的乘积。

这种分解也叫做杜利特尔分解，从而将方程组转化为求解两个三角形方程组

而在程序运算中，高斯消元法每对一行进行运算，其下面的每一行元素都需要更新，而在三角分解法的程序实现过程中，每个元素只需要进行一次的更新就可以达成目标，这样很大程度上减少了运算次数，减小了时间复杂度。

## 计算公式

**高斯消去法**

对接下来每行进行更新，有公式

得到上三角阵之后，对x回代求解，有公式

**三角分解法**

由于两个矩阵可以由一个矩阵的变换得来，因此可以得到如下更新公式

## 程序设计

程序设计过程中，主要有以下几点需要注意：

1. **行交换**  
   由于是列主元法，因此每次迭代都需要判断当前列往下的当列的最大元，并考虑是否交换，在这种情况下，我的做法是现将矩阵进行分割，然后用max和find函数找到最大值和最大值所处行数，再考虑是否交换。
2. **矩阵无解**需要考虑矩阵没有解的情况，或者列主元交换之后仍然主元为0，这时候矩阵无解，考虑直接退出。
3. **行的更新**在矩阵元素的更新过程中有许多的重复批量操作，可以利用matlab的矩阵运算特性按行操作。在三角分解法中也可以通过构建的矩阵乘法来简化表达式。
4. **外层循环的次数**虽然要遍历每个点，但是每次迭代都要从寻找列主元开始，因此每次都是从对角线开始判断，因此最外层其实只需要n次。

函数的输入输出：由于方程组有解的需求就默认其能解，因此默认系数矩阵为方阵。本人设计的函数都是两个输入，分别是增广矩阵函数以及增广矩阵的行数。返回方程组的解x以及三角分解还会返回L和U，并输出P。

程序框图（三角分解和高斯消去只有在矩阵元素更新处不同）

|  |  |
| --- | --- |
|  |  |

## 结果讨论与分析

对课本P227 第三题进行计算求解

【高斯消去法】



【三角分解法】

得到P =

[ 0 0 1

1 0 0

0 1 0]得到L=

[ 1 0 0

1 1 0

0.5 -0.5 1]得到U=

[ 2 1 2

0 3 4

0 0 2]

并且有解为

1.166666666666667

-0.333333333333333

0.500000000000000

**程序评价**

经过验证，两个算法结果无误。对程序的评价而言，这两个函数都比较好的完成了高斯消去法和三角分解法的任务。

亮点：

1. 可以在程序启动的时候输入增广矩阵和求解阶数，而不必在程序内修改。
2. 设计巧妙的矩阵乘法以化简了代码。
3. 主函数的设计同时可以用来验证雅可比，高斯赛德尔与逐次超松弛法解决问题，并且方便将不同的算法放在一起比较。另外几种函数也都写好并置于文件夹中，可以通过主函数Lab\_1.m调用。

除了这些之外，这个程序没有考虑系数矩阵的秩小于阶数的情况，即默认方程有解。而在计算中，最好还是考虑这种情况并提前退出计算。

**复杂度分析：**

空间复杂度S(N)为O(N)，和输入规模为相同数量级。时间复杂度T(N)由于不了解matlab内部的函数复杂度以及matlab矩阵运算的复杂度，因此无法计算。

## Appendix（coding in matlab）

【主函数】

clc;

clear;

close all;

% %

r = input("输入方阵行数 : ");

a = input("输入矩阵 :\n");

% r = 3;

% a = [0 3 4 1;1 -1 1 2; 2 1 2 3];

%%

 [L,u,x] = Tri\_decomposition(a,r);

% x = Gaussian\_eli(a,r);

% disp(L);

disp(x);

% disp(u);

%% Jacobi

%

% X = SOR(a,r,0.001,1);

% disp(X);

% result = Gaussian\_eli(a,r);

% disp(result);

【列主元高斯消去法】

function x = Gaussian\_eli(A,rows)

%这里是默认系数矩阵为方阵

    i = 2;

    x = zeros(1,rows);

    m = zeros(1,rows);

    while(i<=rows)

        %%从这里开始是开始迭代

    %这个地方是用于寻找列主元

        max\_col\_i = max(A(:,i-1));

        max\_l = find(abs(A(:,i-1) -max\_col\_i)<0.01);

        if max\_l ~= i-1

        %交换两列

            A([max\_l i-1],:) = A([i-1 max\_l],:);

        end

    %%如果变换之后仍然是首项为0，则该循环直接跳过，并且方程可能直接无解

        if abs(A(i-1, i-1) - 0)<0.01

            continue;

        end

    %%

    %找到主元之后，对每一行求系数以将最左列消去

        for j = i:rows

            m(j) = A(j,i-1)/A(i-1,i-1);

            A(j,:) = A(j,:) - m(j)\*A(i-1,:);

        end

        i = i+1;

    end

    %%这里就是将x求解出来

    x(rows)=A(rows,rows+1)/A(rows,rows);

    for k = rows-1:-1:1

        former\_add = sum(x.\*A(k,1:rows));

        x(k)=(A(k,rows+1)-former\_add)/A(k,k);

    end

end

【列主元三角分解法】

function [L,u,x] = Tri\_decomposition(A,rows)

%这里是为了计算u\_ij

    u=A;

    s = zeros(rows);%这里定义一个s用于放置列主元判断时的量

    p = diag(ones(1,rows));

    for k=1:rows

       %%这个部分是（列主元）用的，如果不需要列主元，只是单纯的三角分解，就可以跳过这一部分（8~24行）

        s(k:rows,k) = u(k:rows,k);%赋值，在s中判断

        for i = k:rows

            s(i,k)=A(i,k)-u(i,1:k-1) \* u(1:k-1,k);

        end

        max\_u = max(s(k:rows,k));%找到最大值

%         max\_u

        max\_u\_r = find(abs(s(:,k)-max\_u)<0.001,1); % 找到最大值index

        if max\_u\_r ~= k%如果和当前不同就交换

            u([max\_u\_r k],:)=u([k max\_u\_r],:);

            A([max\_u\_r k],:)=A([k max\_u\_r],:);

            p([max\_u\_r k],:)=p([k max\_u\_r],:);

        end

%         clear s\_i;

%         k

%         disp(u);

        %%

        for j=k:rows + 1%更新列

            u(k,j)=A(k,j)-u(k,1:k-1) \* u(1:k-1,j);%这里用了矩阵的乘法，乘出来是一个常数

        end

        for i=k+1:rows %更新行

            u(i,k)=(A(i,k)-u(i,1:k-1) \* u(1:k-1,k))/u(k,k);

        end

    end

    %后面的是把x，L，U求出来

    y = u(:,rows+1);

    u = u(:,1:rows);

    x = zeros(rows,1);

    x(rows) = y(rows)/u(rows,rows);

    for i = rows-1:-1:1 %这里是把x求出来

        f\_d = u(i,i+1:rows) \* x((i+1):rows);

        x(i)=(y(i)-f\_d)/u(i,i);

    end

    L = zeros(rows);

    for i = 1:rows

        L(i+1:rows,i)=u(i+1:rows,i);

    end

    u = u-L;

    L = L+diag(ones(1,rows));

    disp(p);

end