1 问题描述

编写求解线性代数方程组的原始高斯消去算法程序

验证程序正确后，利用此程序完成以下任务：

已知函数$y = c_0 + c_1x + c_2x^2 + c_3x^3 + c_4x^4$通过点
(2.16)，$(3, 81)$，$(4, 256)$，$(5, 625)$，$(6, 1296)$，则函数可通过求解以下方程组得到

$$\begin{bmatrix} 1 & 2 & 4 & 8 & 16 \\ 1 & 3 & 9 & 27 & 81 \\ 1 & 4 & 16 & 64 & 256 \\ 1 & 5 & 25 & 125 & 625 \\ 1 & 6 & 36 & 216 & 1296 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} 16 \\ 81 \\ 256 \\ 625 \\ 1296 \end{bmatrix}$$

- 求解此方程组
- 将系数矩阵中的1296改为1295，求出结果
- 分析你的结果。

2 问题解析

原始高斯消元法的原理，即对增广矩阵矩阵，从第二行开始，每一次消元消去前面若干项数，使其最终的系数矩阵成为一个上三角矩阵，再依次从下到上依次回代求解。
共需要三层循环，第一层循环表示一共需要操作的次数，定义变量k说明从哪一行开始操作(即保留到哪一行)；第二层是操作的行数，从k+1行开始，到最后一行结束；第三层是对该行每一个数进行操作，均对第k行进行线性运算消去第k项。完成后再从下而上对x依次求解。

对题目中的方程组，分析其解的情况：

- 当方程组的系数矩阵的秩与方程组增广矩阵的秩相等且均等于方程组中未知数个数n的时候，方程组有唯一解；
- 当方程组的系数矩阵的秩与方程组增广矩阵的秩相等目均小于方程组中未知数个数n的时候方程组有无穷多解；
- 当方程组的系数矩阵的秩小于方程组增广矩阵的秩的时候，方程组无解。

```
a=[1 2 4 8 16;1 3 8 27 81;1 4 16 64 256;1 5 25 125 625;1 6 36 216 1296];
%系数矩阵
b=[16 81 256 625 1296];%此时b为行向量 计算时需要转置
aplusb=[a,b']; %增广矩阵
disp(rank(aplusb)); %求秩
disp(rank(a));
```

得到系数矩阵与增广矩阵的秩都为5，即未知数个数；且当1296改为1295时也成立，故只有唯一解。通过计算得出当为1296时，解的情况为$(0\ 0\ 0\ 0\ 1)^T$

3 代码实现

3.1 原始高斯消元法的实现

```
function x=Gauss(a,b)
    [m,n]=size(a);%得到a的形状
    for k=1:n-1
        for i=k+1:n
            factor=a(i,k)/a(k,k); %消去项对应的系数
            for j=k+1:n
                a(i,j)=a(i,j)-factor*a(k,j); %每一项都要进行线性变换
            end
            b(i)=b(i)-factor*b(k); %常数项变换
        end
    end
    x(n)=b(n)/a(n,n);
    for i=n-1:-1:1 %自下而上依次求x
        sum=b(i);
       for j=i+1:n
           sum=sum-a(i,j)*x(j);
       end
       x(i)=sum/a(i,i);
    end
end
```

其中函数输出为解，输入a为系数矩阵，b为常数项向量。

3.2 验证程序

用如下方程组进行验证

$$\begin{cases} 2x+3y-z=5 \\ x-y+4z=11 \\ 3x+2y+z=10 \end{cases}$$

有且仅有唯一解x=1 y=2 z=3

```
a_test=[2 3 -1;1 -1 4;3 2 1];
b_test=[5 11 10];
x_test=Gauss(a_test,b_test);
for i=1:3
    fprintf("%.16f\n",x_test(i));
end
```

输出为

```
1.0000000000000000
2.0000000000000000
3.0000000000000000
```

3.3 求解原方程组及修改后的方程组

```
a=[1 2 4 8 16;1 3 8 27 81;1 4 16 64 256;1 5 25 125 625;1 6 36 216 1296];%或1295
b=[16 81 256 625 1296];
c=Gauss(a,b);
```

4 结果分析

```
[m,n]=size(a);
for i=1:n  %将结果回代回每个方程组，得到对应的函数值
    sum=0;
    for j=1:n
        sum=sum+a(i,j)*c(j);
    end
    fprintf("%.16f\n",c(i));
end
for i=1:n %展示得到的结果
    fprintf("%.16f\n",c(i));
end
fprintf("\n");

x=linsolve(a,b'); %用matlab自带的函数求解与高斯消元法对比
for i=1:n
    sum=0;
    for j=1:n
        sum=sum+a(i,j)*x(j);
    end
    fprintf("%.16f\n",x(i));
end
for i=1:n
    fprintf("%.16f\n",x(i));
end
```

4.1 结果一

当系数矩阵处为1296时

原始高斯消元法：

```
16.0000000000000000
81.0000000000000000
256.0000000000000000
625.0000000000000000
1296.0000000000000000

0.0000000000000000
0.0000000000000000
0.0000000000000000
0.0000000000000000
1.0000000000000000
```

上面是回代的结果，下面是解得的结果。

用linsolve方法：

```
16.0000000000000000
81.0000000000000000
256.0000000000000000
625.0000000000000000
1296.0000000000000000

0.0000000000000000
0.0000000000000000
-0.0000000000000000
0.0000000000000000
1.0000000000000000
```

上面是回代的结果，下面是解得的结果。

可以发现两者结果并几乎没有显著不同，除了c2的值用linsolve解得的值带负号，说明在16位小数之外有微小误差，可以忽略不记。两者计算结果与手算结果完全相同，回代结果也完全吻合。

4.2 结果二

当结果为1295

原始高斯消元法：

```
16.0000000000000000
81.0000000000000000
256.0000000000000000
625.0000000000000000
1296.0000000000000000

-1.4763552479815338
0.8027681660899617
0.1637831603229500
-0.1280276816608986
1.0149942329873125
```

上面是回代的结果，下面是解得的结果。

用linsolve方法：

```
16.0000000000000000
81.0000000000000284
256.0000000000000568
625.0000000000001137
1296.0000000000000000

-1.4763552479816451
0.8027681660900239
0.1637831603229500
-0.1280276816609008
1.0149942329873127
```

上面是回代的结果，下面是解得的结果。对比发现linsolve的方法的精度不如原始高斯消元法。

4.3 分析

1295与1296的解进行对比，发现对系数微小的扰动导致解发生了显著的变化，虽然回代结果显示是准确的，但是如果我们最结果只保留四位小数再回代的话

```
c_4=round(c,4);
for i=1:n
    sum=0;
    for j=1:n
        sum=sum+a(i,j)*c_4(j);
    end
    fprintf("%.16f\n",sum);
end
```

```
16.0003999999999991
81.0013999999999896
256.0035999999999490
625.0075999999999112
1296.0141999999998461
```

这样与原式子的偏差变大了，在实际应用场景中一般不会用到太多的小数位，因此常需要四舍五入，如果用原始高斯消元法求这类方程组的解时，一般会产生较大的舍入误差，这类方程组即病态方程组。

将1295得到的解回代1296的系数矩阵

```
a_2=[1 2 4 8 16;1 3 8 27 81;1 4 16 64 256;1 5 25 125 625;1 6 36 216
1296];
for i=1:n
    sum=0;
    for j=1:n
        sum=sum+a_2(i,j)*c(j);
    end
    fprintf("%.16f\n",sum);
end
```

```
16.0000000000000000
81.0000000000000000
256.0000000000000000
625.0000000000000000
1297.0149942329871919
```

发现只有最后的常数项出现了误差，这也体现出了对系数矩阵的微小扰动造成的误差传递效果。

5 个人心得

病态方程组对系数矩阵的扰动非常敏感，微小的变化也会引起结果的显著变化，结果可能由整数变为小数，在实际应用中保留有效数字回导致较大的舍入误差，因此原始高斯消元法不适用于这种大型或病态方程组的问题，使用前应先检查方程组是否为病态，是否会出现数值稳定性问题。