# 实习六实习报告

**计21 战裕隆 2012011388**

## 实习要求

**编程序生成矩阵，以及n维向量，其中为所有分量都是1的向量。**

**计算n为3和4的时候的无穷范数。**

**N=10的时候，用分解算法求解方程，得到近似解，计算残差和误差的-范数。**

**在完成上一个任务之后，让向量b产生10-7的误差，随后观察误差，残差的变化。**

**研究n的大小和误差大小的关系，n大约在多少的时候误差会达到100%。**

## 算法描述

1. **生成希尔伯特矩阵**
2. **生成希尔伯特矩阵逆矩阵**
3. **根据范数定义求取要求的范数以及条件数**
4. **求取所需要的temp矩阵**
5. **求取x1和x向量，并进行计算，算出误差和残差数组**
6. **根据大小求出所求得量**

## 程序清单

**可以调节是否使用干扰以进行计算**

**#include<iostream>**

**#include<cstdlib>**

**#include<cmath>**

**#include<memory>**

**#include<iomanip>**

**using namespace std;**

**int n;**

**double use[17][17];**

**double ni[17][17];**

**double ni\_temp[17][40];**

**double b[17];**

**double temp[17][17];**

**double x1[15],x[15];**

**double temp\_1[15];**

**double cancha[15],wucha[15];**

**int main()**

**{**

**cin>>n;**

**for(int i=0;i<15;i++)**

**{**

**for(int j=0;j<15;j++)**

**{**

**use[i][j]=1.0/(i+j+1);**

**}**

**}**

**for(int i=0;i<15;i++)**

**{**

**for(int j=0;j<15;j++)**

**{**

**ni\_temp[i][j]=use[i][j];**

**}**

**}**

**for(int i=0;i<15;i++)**

**{**

**for(int j=15;j<30;j++)**

**{**

**if(i==(j-15))**

**{**

**ni\_temp[i][j]=1;**

**}**

**else**

**{**

**ni\_temp[i][j]=0;**

**}**

**}**

**}**

**for(int i=0;i<15;i++)**

**{**

**if(ni\_temp[i][i]==0)**

**{**

**for(int k=i;k<15;k++)**

**{**

**if(ni\_temp[k][k]!=0)**

**{**

**for(int j=0;j<30;j++)**

**{**

**double temp1;**

**temp1=ni\_temp[i][j];**

**ni\_temp[i][j]=ni\_temp[k][j];**

**ni\_temp[k][j]=temp1;**

**}**

**break;**

**}**

**}**

**}**

**for(int j=30-1;j>=i;j--)**

**{**

**ni\_temp[i][j]/=ni\_temp[i][i];**

**}**

**for(int k=0;k<15;k++)**

**{**

**if(k!=i)**

**{**

**double temp2=ni\_temp[k][i];**

**for(int j=0;j<30;j++)**

**{**

**ni\_temp[k][j]-=temp2\*ni\_temp[i][j];**

**}**

**}**

**}**

**}**

**for(int i=0;i<15;i++)**

**{**

**for(int j=15;j<30;j++)**

**{**

**ni[i][j-15]=ni\_temp[i][j];**

**}**

**}//求出逆矩阵**

**double sum3=0.0,sum4=0.0;**

**double sum33=0.0,sum44=0.0;**

**for(int i=0;i<3;i++)**

**{**

**sum3=sum3+use[i][0];**

**}**

**for(int i=0;i<4;i++)**

**{**

**sum4=sum4+use[i][0];**

**}**

**/\*for(int i=0;i<3;i++)**

**{**

**for(int j=0;j<3;j++)**

**{**

**cout<<ni[i][j]<<" ";**

**}**

**cout<<endl;**

**}\*/**

**double he[5]={0.0};**

**for(int i=0;i<3;i++)**

**{**

**for(int j=0;j<3;j++)**

**{**

**if(ni[i][j]>0)**

**he[i]=he[i]+ni[i][j];**

**else**

**he[i]=he[i]-ni[i][j];**

**}**

**}**

**sum33=he[0];**

**for(int i=0;i<3;i++)**

**{**

**if(he[i]>sum33)**

**{**

**sum33=he[i];**

**}**

**}**

**for(int i=0;i<=4;i++)**

**{**

**he[i]=0.0;**

**}**

**for(int i=0;i<4;i++)**

**{**

**for(int j=0;j<4;j++)**

**{**

**if(ni[i][j]>0)**

**he[i]=he[i]+ni[i][j];**

**else**

**he[i]=he[i]-ni[i][j];**

**}**

**}**

**sum44=he[0];**

**for(int i=0;i<4;i++)**

**{**

**if(he[i]>sum44)**

**{**

**sum44=he[i];**

**}**

**}**

**cout<<"H3的范数 "<<sum3<<endl;**

**cout<<"H4的范数 "<<sum4<<endl;//完成第一个任务，求取H3，H4的无穷范数。**

**double cond1,cond2;**

**cond1=sum3\*sum33;**

**cond2=sum4\*sum44;**

**cout<<"H3的条件数 "<<cond1<<endl;**

**cout<<"H4的条件数 "<<cond2<<endl;**

**for(int i=0;i<15;i++)**

**{**

**b[i]=0.0;**

**}**

**for(int i=0;i<10;i++)**

**{**

**for(int k=0;k<10;k++)**

**{**

**b[i]=b[i]+use[i][k];**

**}**

**}**

**for(int i=0;i<10;i++)**

**{**

**b[i]=b[i]+0.0000001;**

**}**

**for(int i=0;i<n;i++)**

**{**

**for(int k=0;k<n;k++)**

**{**

**temp[i][k]=use[i][k];**

**}**

**}**

**for(int i=0;i<n;i++)**

**{**

**for(int k=0;k<i;k++)**

**{**

**temp[i][i]=temp[i][i]-temp[i][k]\*temp[i][k];**

**}**

**temp[i][i]=sqrt(temp[i][i]);**

**for(int j=i+1;j<n;j++)**

**{**

**for(int k=0;k<i;k++)**

**{**

**temp[j][i]=temp[j][i]-temp[i][k]\*temp[j][k];**

**}**

**temp[j][i]=temp[j][i]/temp[i][i];**

**}**

**}//分解**

**/\*for(int i=0;i<n;i++)**

**{**

**for(int k=0;k<n;k++)**

**{**

**cout<<temp[i][k]<<" ";**

**}**

**cout<<endl;**

**}\*/**

**for(int i=0;i<n;i++)**

**{**

**x1[i]=b[i];**

**for(int j=0;j<i;j++)**

**{**

**x1[i]=x1[i]-(temp[i][j]\*x1[j]);**

**}**

**x1[i]=x1[i]/temp[i][i];**

**}**

**/\*for(int i=0;i<n;i++)**

**{**

**cout<<x1[i]<<endl;**

**} \*/**

**for(int i=9;i>=0;i--)**

**{**

**x[i]=x1[i];**

**for(int j=9;j>i;j--)**

**{**

**x[i]=x[i]-(temp[j][i]\*x[j]);**

**}**

**x[i]=x[i]/temp[i][i];**

**}**

**/\*for(int i=0;i<n;i++)**

**{**

**cout<<x[i]<<endl;**

**}\*/**

**for(int i=0;i<n;i++)**

**{**

**wucha[i]=x[i]-1.0;**

**}//误差计算完毕**

**for(int i=0;i<n;i++)**

**{**

**for(int j=0;j<n;j++)**

**{**

**temp\_1[i]=temp\_1[i]+use[i][j]\*x[j];**

**}**

**cancha[i]=b[i]-temp\_1[i];**

**}**

**double wucha\_max=abs(wucha[0]),cancha\_max=abs(cancha[0]);**

**for(int i=1;i<n;i++)**

**{**

**if(abs(wucha[i])>wucha\_max)**

**{**

**wucha\_max=abs(wucha[i]);**

**}**

**if(abs(cancha[i])>cancha\_max)**

**{**

**cancha\_max=abs(cancha[i]);**

**}**

**}**

**cout<<"残差："<<cancha\_max<<endl;**

**cout<<"误差："<<wucha\_max<<endl;**

**system("pause");**

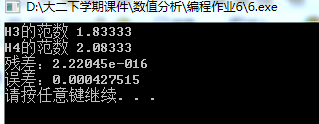
**return 0;**

**}**

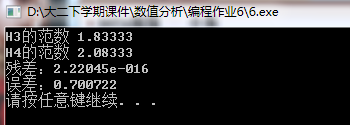
## 体会与问题

1. **这个程序的变量名比较容易搞混，还是取一些比较容易区分的名称才好，我就是一个地方写错，跟同学对拍才找出问题。**
2. **实验结果如下：**

**没有扰动的n=10的输出结果如下：**

****

**而n=10的时候，b向量添加扰动的输出结果如下：**

****

1. **当n增大的时候，误差也跟着变大（可以在程序中调整n的输入进行尝试）**
2. **两个条件数的值都比较大，并且随着n的增大不断增大，这是病态矩阵。**
3. **当n的取值大约为6的时候，误差将近100%，可见希尔伯特矩阵确实是一个病态矩阵**
4. **残差的大小变化基本和扰动差不多，所以在以后的应用中使用残差未尝不是一个好的方法。**