# 实习报告

## 实习要求

用幂法求出下列矩阵按模最大的特征值及其对应的特征向量*x*1，使得两个相邻的特征值逼近量小于10的-5次方

A={5，-4,1，-4,6，-4,1，-4,7}

B={25，-41,10，-6，-41,68，-17,10,10，-17,5，-3，-6,10，-3,2}

## 算法描述

1. 生成u0，v0初始值
2. 通过条件判断进入循环
3. 用u和矩阵A迭代求取v向量
4. 对u向量进行规范化
5. 判断uk值是否符合精度要求，当符合要求的时候跳出循环，输出结果

## 程序清单

#include<iostream>

#include<cmath>

#include<memory.h>

using namespace std;

double a[6][6];

double v[10000][6]={0.0},u[10000][6]={0.0};

double u\_max[10000]={0.0};

int cishu=1;

int n;

int biaoji;

void mifa()

{

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

{

v[0][i]=1.0;

u[0][i]=1.0;

}//初值

biaoji=0;

while(biaoji==0)

{

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

{

for(int k=1;k<=n;k++)

{

v[cishu][i]=v[cishu][i]+a[i][k]\*u[cishu-1][k];

}

}

u\_max[cishu]=v[cishu][1];

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

{

if(v[cishu][i]>u\_max[cishu])

{

u\_max[cishu]=v[cishu][i];

}

}

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

{

u[cishu][i]=v[cishu][i]/u\_max[cishu];

}

if(abs(u\_max[cishu]-u\_max[cishu-1])<1e-5)

{

biaoji=1;

break;

}

else

{

cishu++;

continue;

}

}

}

int main()

{

cin>>n;

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

{

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

{

cin>>a[i][j];

}

}

mifa();

cout<<u\_max[cishu]<<endl;

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

{

cout<<u[cishu][i]<<" ";

}

cout<<endl;

cout<<cishu<<endl;

system("pause");

return 0;

}

1.cpp是我自己写的源代码，2.cpp则是从网上找的检验程序，经过检验，无误。

## 体会与问题

本题应该来说是所有编程作业里面的难度最小的了，一开始我只输出了主特征值和特征向量，并没有发现什么，之后我输出了迭代次数，发现三阶的迭代次数为17，而四阶的矩阵迭代次数仅为5，并非是矩阵阶数越大，迭代次数就必定越多，这是跟矩阵内部的值有着直接关系。

这种算法的收敛速度按照书上的结论是由第二个特征值和第一特征值的比值大小决定的，但是我希望可以找到一个和矩阵内部的值有着直接关系的结论。

我在网上查找了很多资料，并没有提到幂法的迭代次数和阶数有什么关系或者和其他什么因素有关，题目中所给的是两个对称矩阵，通过观察我猜测有着一定倍数关系的矩阵可能迭代次数会少一些，元素值比较相近可能比较消耗迭代次数。