**数值分析第二次大作业**

BY1702103 斯志仲

具体思路：（我用的是第三版数，页码以此为准）

1. 对矩阵A进行拟上三角化，用书上P61页下方的算法，得到An-1。
2. 再写QR分解算法，用书上P59页算法。
3. 再利用书上P62页的算法，用上面QR分解的方法，循环迭代，得到迭代结束后的举证。
4. 再利用类似于书上P63，P64页的方法，得出实数特征值和复数特征值。
5. 对于实数特征值，再利用选主元方法，进行高斯消去。得到特征值。

这里有一个trick，就是A-lambda\*I的矩阵的秩是10-1=9，所以高斯消去后，最后一行一定是十个0，所以直接令特征值的最后一位为1，从倒数第二行回代即可。最后再做归一化。

编程里面的一些特点

1. 由于C语言返回指针数组的时候。容易导致内存管理混乱，得出错误结论。我所有需要返回数组的函数，均是采用在输入参数的最后一个加入一个数值指针，通过这个参数，返回结果数值，这样同时可以返回多个参数。
2. 防止在底层函数里面修改输入矩阵的数值，所有底层函数的矩阵运算，都对输入的矩阵做了个深拷贝，然后利用这个矩阵进行运算。
3. 在具体算法实现前，都写了很多矩阵和向量的基本运算函数，直接调用。虽然这个代码很长，但是非常易读。
4. 由于结果太长，不适宜输出到控制台上，我全部输出到了matrix.txt文件中。

具体代码和结果如下：

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

#define SIZE 10

//符号函数

double sgn(double s)

{

if(s<0)

return -1;

else

return 1;

}

//矩阵乘以向量，输出矩阵

void matrix\_multipy\_vector(double input\_matrix[SIZE][SIZE],double input\_vector[SIZE],double output\_vector[SIZE])

{

int i,j;

for(i=0;i<SIZE;i++)

{

output\_vector[i]=0;

for(j=0;j<SIZE;j++)

output\_vector[i]=output\_vector[i]+input\_matrix[i][j]\*input\_vector[j];

}

}

//向量点积，输出标量

double vector\_dot\_multiply\_vector(double input\_vector1[SIZE],double input\_vector2[SIZE])

{

double result=0;

int i=0;

for(i=0;i<SIZE;i++)

result+=input\_vector1[i]\*input\_vector2[i];

return result;

}

//纵向量乘横向量，返回矩阵

void vector\_multiply(double input\_vector1[SIZE],double input\_vector2[SIZE],double output\_matrix[SIZE][SIZE])

{

int i=0,j=0;

for(i=0;i<SIZE;i++)

{

for(j=0;j<SIZE;j++)

output\_matrix[i][j]=input\_vector1[i]\*input\_vector2[j];

}

}

//向量减法，A-t\*B返回向量

void vector\_sub(double input\_vector1[SIZE],double input\_vector2[SIZE],double t,double output\_vector[SIZE])

{

int i;

for(i=0;i<SIZE;i++)

output\_vector[i]=input\_vector1[i]-t\*input\_vector2[i];

}

//矩阵减法,返回矩阵

void matrix\_sub(double input\_matrix1[SIZE][SIZE],double input\_matrix2[SIZE][SIZE],double output\_matrix[SIZE][SIZE])

{

int i=0,j=0;

for(i=0;i<SIZE;i++)

{

for(j=0;j<SIZE;j++)

output\_matrix[i][j]=input\_matrix1[i][j]-input\_matrix2[i][j];

}

}

//矩阵转置

void matrix\_transpose(double input\_matrix[SIZE][SIZE],double output\_matrix[SIZE][SIZE])

{

int i,j;

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

output\_matrix[i][j]=input\_matrix[j][i];

}

//向量除标量

void vector\_div\_scalar(double input\_vector[SIZE],double s,double output\_vector[SIZE])

{

int i;

for(i=0;i<SIZE;i++)

output\_vector[i]=input\_vector[i]/s;

}

//矩阵乘法

void matrix\_multiply(double input\_matrix1[SIZE][SIZE],double input\_matrix2[SIZE][SIZE],double output\_matrix[SIZE][SIZE])

{

int i,j,k;

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

{

output\_matrix[i][j]=0;

for(k=0;k<SIZE;k++)

output\_matrix[i][j]+=input\_matrix1[i][k]\*input\_matrix2[k][j];

}

}

//对输入矩阵进行拟上三角化

void Hessenberg( double input\_matrix[SIZE][SIZE], double output\_matrix[SIZE][SIZE])

{

int i,j,k,all\_zero\_flag;

double dr,cr,hr,tr;

double ur[SIZE], pr[SIZE],qr[SIZE],wr[SIZE];

double temp\_matrix[SIZE][SIZE];

double temp\_matrix\_trans[SIZE][SIZE];

double v\_multi\_v\_matirx[SIZE][SIZE];

//利用中间矩阵进行计算，防止破坏原来输入矩阵；

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

temp\_matrix[i][j]=input\_matrix[i][j];

//对矩阵的每一列做运算

for (j=0;j<SIZE-2;j++)

{

//判断是不是所有元素都是0

for (i=j+2;i<SIZE;i++)

all\_zero\_flag=all\_zero\_flag\*(temp\_matrix[i][j]==0);

if(all\_zero\_flag==1) continue;//如果全部为0，跳到书中P62页步骤5

else

{

dr=0;

for (i=j+1;i<SIZE;i++)

dr+=temp\_matrix[i][j]\*temp\_matrix[i][j];

dr=sqrt(dr);

if(temp\_matrix[j+1][j]==0)

cr=dr;

else

cr=-sgn(temp\_matrix[j+1][j])\*dr;

hr=cr\*cr-cr\*temp\_matrix[j+1][j];

for(k=0;k<j+1;k++)

ur[k]=0;

ur[j+1]=temp\_matrix[j+1][j]-cr;

for(k=j+2;k<SIZE;k++)

ur[k]=temp\_matrix[k][j];

matrix\_transpose(temp\_matrix,temp\_matrix\_trans);

matrix\_multipy\_vector(temp\_matrix\_trans,ur,pr);

vector\_div\_scalar(pr,hr,pr);

matrix\_multipy\_vector(temp\_matrix,ur,qr);

vector\_div\_scalar(qr,hr,qr);

tr=vector\_dot\_multiply\_vector(pr,ur)/hr;

vector\_sub(qr,ur,tr,wr);

vector\_multiply(wr,ur,v\_multi\_v\_matirx);

matrix\_sub(temp\_matrix,v\_multi\_v\_matirx,temp\_matrix);

vector\_multiply(ur,pr,v\_multi\_v\_matirx);

matrix\_sub(temp\_matrix,v\_multi\_v\_matirx,temp\_matrix);

}

}

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

{

output\_matrix[i][j]=temp\_matrix[i][j];

if(fabs(output\_matrix[i][j])<pow(10,-15))

output\_matrix[i][j]=0;

}

}

void QR\_decomposition(double input\_matrix[SIZE][SIZE],double Q[SIZE][SIZE],double R[SIZE][SIZE])

{

int i,j,k,all\_zero\_flag;

double dr,cr,hr;

double ur[SIZE], pr[SIZE],wr[SIZE];

double temp\_matrix[SIZE][SIZE];

double temp\_matrix\_trans[SIZE][SIZE];

double v\_multi\_v\_matirx[SIZE][SIZE];

//Q=I

for (i=0;i<SIZE;i++)

{

for(j=0;j<SIZE;j++)

{

if(i==j) Q[i][j]=1;

else Q[i][j]=0;

}

}

//利用中间矩阵进行计算，防止破坏原来输入矩阵；

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

temp\_matrix[i][j]=input\_matrix[i][j];

//对矩阵的每一列做运算

for (j=0;j<SIZE-1;j++)

{

//判断是不是所有元素都是0

for (i=j+1;i<SIZE;i++)

all\_zero\_flag=all\_zero\_flag\*(temp\_matrix[i][j]==0);

if(all\_zero\_flag==1) continue;//如果全部为0，跳到书中P62页步骤5

else

{

dr=0;

for (i=j;i<SIZE;i++)

dr+=temp\_matrix[i][j]\*temp\_matrix[i][j];

dr=sqrt(dr);

if(temp\_matrix[j][j]==0)

cr=dr;

else

cr=-sgn(temp\_matrix[j][j])\*dr;

hr=cr\*cr-cr\*temp\_matrix[j][j];

for(k=0;k<j;k++)

ur[k]=0;

ur[j]=temp\_matrix[j][j]-cr;

for(k=j+1;k<SIZE;k++)

ur[k]=temp\_matrix[k][j];

matrix\_multipy\_vector(Q,ur,wr);

vector\_div\_scalar(ur,hr,ur);

vector\_multiply(wr,ur,v\_multi\_v\_matirx);

matrix\_sub(Q,v\_multi\_v\_matirx,Q);

matrix\_transpose(temp\_matrix,temp\_matrix\_trans);

matrix\_multipy\_vector(temp\_matrix\_trans,ur,pr);

vector\_div\_scalar(ur,1/hr,ur);

vector\_multiply(ur,pr,v\_multi\_v\_matirx);

matrix\_sub(temp\_matrix,v\_multi\_v\_matirx,temp\_matrix);

}

}

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

{

R[i][j]=temp\_matrix[i][j];

if(fabs(R[i][j])<pow(10,-15))

R[i][j]=0;

}

}

//用QR分解法，循环迭代

void QR\_method(double input\_matrix[SIZE][SIZE],double output\_matrix[SIZE][SIZE])

{

double R[SIZE][SIZE],Q[SIZE][SIZE];

double Ak[SIZE][SIZE];

int i,j;

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

Ak[i][j]=input\_matrix[i][j];

while(fabs(Ak[7][6])>pow(10,-20))

{

QR\_decomposition(Ak,Q,R);

matrix\_multiply(R,Q,Ak);

}

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

output\_matrix[i][j]=Ak[i][j];

}

//二阶子阵，求特征值

void two\_order\_matrix(double a,double b,double c,double d,double \*real,double \*image)

{

double p1,p2,p3,delta;

p1=1;

p2=-(a+d);

p3=a\*d-b\*c;

delta=p2\*p2-4\*p1\*p3;

if(delta>0) printf("error,2-order matrix give a real number eign value\n");

else

{

\*real=-p2/(2\*p1);

\*image=sqrt(-delta)/(2\*p1);

}

}

//eign value 用二维数组保存，分别保存实部和虚部，如果是实数，这虚部为0

void eign\_value(double input\_vector[SIZE][SIZE],double eignvalue[SIZE][2])

{

int i=0;

double real,image;

//不能算到0，因为不存在input\_vector[0][0-1]这个单元

for(i=SIZE-1;i>0;i--)

{

//得到一个实数特征值

if(fabs(input\_vector[i][i-1])<pow(10,-12))

{

eignvalue[i][0]=input\_vector[i][i];

eignvalue[i][1]=0;

}

else

{

//得到一对复数特征值

two\_order\_matrix(input\_vector[i-1][i-1],input\_vector[i-1][i],input\_vector[i][i-1],input\_vector[i][i],&real,&image);

eignvalue[i][0]=real;

eignvalue[i][1]=image;

eignvalue[i-1][0]=real;

eignvalue[i-1][1]=-image;

i--;

}

}

if(i==0)

{

eignvalue[0][0]=input\_vector[0][0];

eignvalue[0][1]=0;

}

}

//用列主要元素高斯消去法，在已知特征值的情况下，求特征向量

void eign\_vector(double input\_matrix[SIZE][SIZE],double lambda,double eignvector[SIZE])

{

double A[SIZE][SIZE];

int i,j,k,column\_pivot\_row\_number;

double temp,mik;

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

{

if(i==j)

A[i][j]=input\_matrix[i][j]-lambda;

else

A[i][j]=input\_matrix[i][j];

}

//消去过程

for(i=0;i<SIZE-1;i++)

{

column\_pivot\_row\_number=i;

//选出列主元

for(k=i+1;k<SIZE;k++)

if(fabs(A[k][i])>fabs(A[column\_pivot\_row\_number][i]))

column\_pivot\_row\_number=k;

//交换

for(j=0;j<SIZE;j++)

{

temp=A[i][j];

A[i][j]=A[column\_pivot\_row\_number][j];

A[column\_pivot\_row\_number][j]=temp;

}

for(k=i+1;k<SIZE;k++)

{

mik=A[k][i]/A[i][i];

for(j=i;j<SIZE;j++)

A[k][j]=A[k][j]-mik\*A[i][j];

}

}

//回代过程

eignvector[SIZE-1]=1;

for(i=SIZE-2;i>=0;i--)

{

temp=0;

for(j=i+1;j<SIZE;j++)

temp+=eignvector[j]\*A[i][j];

eignvector[i]=-temp/A[i][i];

}

//单位化特征向量

temp=0;

for(i=0;i<SIZE;i++)

temp+=eignvector[i]\*eignvector[i];

temp=sqrt(temp);

for(i=0;i<SIZE;i++)

eignvector[i]=eignvector[i]/temp;

}

//数据太长，将数据保存到一个txt文档中

void matrix\_print(double input\_matrix[SIZE][SIZE],FILE \*fp)

{

int i=0,j=0;

for (i=0;i<SIZE;i++)

{

fprintf(fp,"\nthe %dth row is:\n",i+1);

for(j=0;j<SIZE;j++)

fprintf(fp,"%.14e\t",input\_matrix[i][j]);

fprintf(fp,"\n");

}

}

int main()

{

double A[SIZE][SIZE],Hessenberg\_A[SIZE][SIZE],Q[SIZE][SIZE],R[SIZE][SIZE];

double QR\_method\_A[SIZE][SIZE];

double eignvalue[SIZE][2];

double eignvector[SIZE];

int i,j;

FILE \*fp;

fp=fopen("matrix.txt","a+");

for(i=0;i<SIZE;i++)

for(j=0;j<SIZE;j++)

{

if(i==j)

A[i][j]=1.52\*cos(i+1+1.2\*j+1.2);//i+1,j+1

else

A[i][j]=sin(0.5\*i+0.2\*j+0.7);

}

fprintf(fp,"\n\n原矩阵A是:\n");

matrix\_print(A,fp);

Hessenberg(A,Hessenberg\_A);

fprintf(fp,"\n\n拟上三角化矩阵A(n-1)是:\n");

matrix\_print(Hessenberg\_A,fp);

QR\_decomposition(Hessenberg\_A,Q,R);

fprintf(fp,"\n\nA(n-1)QR分解Q是:\n");

matrix\_print(Q,fp);

fprintf(fp,"\n\nA(n-1)QR分解R是:\n");

matrix\_print(R,fp);

QR\_method(Hessenberg\_A,QR\_method\_A);

fprintf(fp,"\n\nA(n-1)的QR迭代结束后矩阵是is:\n");

matrix\_print(QR\_method\_A,fp);

eign\_value(QR\_method\_A,eignvalue);

fprintf(fp,"\n\n特征值是:\n");

for(i=0;i<SIZE;i++)

{

if(eignvalue[i][1]==0)

fprintf(fp,"特征值%.2d是\t%.14e\n",i+1,eignvalue[i][0]);

else

{

fprintf(fp,"特征值%.2d是\t(%.14e) + (%.14e)i\n",i,eignvalue[i][0],eignvalue[i][1]);

}

}

fprintf(fp,"\n\n实数特征值的特征向量是:\n\n");

for(i=0;i<SIZE;i++)

{

//对实数根求特征向量

if(eignvalue[i][1]==0)

{

eign\_vector(A,eignvalue[i][0],eignvector);

fprintf(fp,"特征值%.14e的特征向量是:\n",eignvalue[i][0]);

for(j=0;j<SIZE;j++)

fprintf(fp,"%.14e\t",eignvector[j]);

fprintf(fp,"\n\n");

}

}

return 0;

}

最后结果是：

原矩阵A是:

the 1th row is:

-8.94521698228126e-001 7.83326909627483e-001 8.91207360061435e-001 9.63558185417193e-001 9.97494986604054e-001 9.91664810452469e-001 9.46300087687414e-001 8.63209366648874e-001 7.45705212176720e-001 5.98472144103957e-001

the 2th row is:

9.32039085967226e-001 -4.67145962367197e-001 9.99573603041505e-001 9.73847630878195e-001 9.09297426825682e-001 8.08496403819590e-001 6.75463180551151e-001 5.15501371821464e-001 3.34988150155905e-001 1.41120008059867e-001

the 3th row is:

9.91664810452469e-001 9.46300087687414e-001 1.44435353977697e+000 7.45705212176720e-001 5.98472144103957e-001 4.27379880233830e-001 2.39249329213982e-001 4.15806624332905e-002 -1.57745694143248e-001 -3.50783227689620e-001

the 4th row is:

8.08496403819590e-001 6.75463180551151e-001 5.15501371821464e-001 -1.23286138137372e+000 1.41120008059867e-001 -5.83741434275801e-002 -2.55541102026831e-001 -4.42520443294852e-001 -6.11857890942719e-001 -7.56802495307928e-001

the 5th row is:

4.27379880233830e-001 2.39249329213982e-001 4.15806624332905e-002 -1.57745694143248e-001 6.72706094183719e-003 -5.29836140908493e-001 -6.87766159183974e-001 -8.18277111064410e-001 -9.16165936749455e-001 -9.77530117665097e-001

the 6th row is:

-5.83741434275801e-002 -2.55541102026831e-001 -4.42520443294852e-001 -6.11857890942719e-001 -7.56802495307928e-001 1.22494361561349e+000 -9.51602073889516e-001 -9.93691003633464e-001 -9.96164608835841e-001 -9.58924274663138e-001

the 7th row is:

-5.29836140908493e-001 -6.87766159183974e-001 -8.18277111064410e-001 -9.16165936749455e-001 -9.77530117665097e-001 -9.99923257564101e-001 -1.44848843366851e+000 -9.25814682327732e-001 -8.32267442223901e-001 -7.05540325570392e-001

the 8th row is:

-8.71575772413588e-001 -9.51602073889516e-001 -9.93691003633464e-001 -9.96164608835841e-001 -9.58924274663138e-001 -8.83454655720153e-001 -7.72764487555987e-001 4.79930507477250e-001 -4.64602179413757e-001 -2.79415498198926e-001

the 9th row is:

-9.99923257564101e-001 -9.82452612624332e-001 -9.25814682327732e-001 -8.32267442223901e-001 -7.05540325570392e-001 -5.50685542597638e-001 -3.73876664830236e-001 -1.82162504272096e-001 8.83609153957942e-001 2.15119988087816e-001

the 10th row is:

-8.83454655720153e-001 -7.72764487555987e-001 -6.31266637872322e-001 -4.64602179413757e-001 -2.79415498198926e-001 -8.30894028174964e-002 1.16549204850494e-001 3.11541363513378e-001 4.94113351138608e-001 -1.51994045611985e+000

拟上三角化矩阵A(n-1)是:

the 1th row is:

-8.94521698228126e-001 -9.93313649182606e-002 -1.09983175887715e+000 -7.66503870907691e-001 1.70760114145622e-001 -1.93488255888865e+000 -8.39020870524530e-002 9.13256511314323e-001 -6.40797700918769e-001 1.94673367868470e-001

the 2th row is:

-2.34787836241606e+000 2.37205792159777e+000 1.82799855231583e+000 3.26655688471351e-001 2.08236058363522e-001 2.08898700994067e+000 1.84786191028881e-001 -1.26301526608010e+000 6.79069466849898e-001 -4.67215088650002e-001

the 3th row is:

0.00000000000000e+000 1.73595446994556e+000 -1.16502336747706e+000 -1.24674444351819e+000 -6.29822548908405e-001 -1.98482018099160e+000 2.97575006079980e-001 6.33930059659465e-001 -1.30851892877229e-001 3.04030103609525e-001

the 4th row is:

0.00000000000000e+000 0.00000000000000e+000 -1.29293756392402e+000 -1.12623922590204e+000 1.19078291192443e+000 -1.30877298389537e+000 1.86015166266588e-001 4.23673393688071e-001 -1.01960082654530e-001 1.94366091450536e-001

the 5th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 1.57771115303190e+000 8.16935832816027e-001 4.46153172382751e-001 -4.36509254160898e-002 -4.66597916718791e-001 2.94123156618385e-001 -1.03442111366465e-001

the 6th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -7.72897513498899e-001 -1.60102824404581e+000 -2.91268547482648e-001 -2.43433785832148e-001 6.73628608450988e-001 2.62477290493710e-001

the 7th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -7.29677394636161e-001 -7.96545627982814e-003 9.71073910200658e-001 -1.29896736857445e-001 2.78024208124104e-002

the 8th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 7.94553961297629e-001 -4.52514345460606e-001 5.04890152757524e-001 -1.21121019351202e-001

the 9th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 7.03991137351415e-001 1.26753552349843e-001 -3.71469673551342e-001

the 10th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -4.91958687221390e-001 4.08150976639909e-001

A(n-1)QR分解Q是:

the 1th row is:

-3.56027250057119e-001 4.43987395278495e-001 -6.93593924883413e-001 6.59751328748562e-002 3.70104288733476e-001 1.87368025302213e-001 -1.61684625386048e-002 1.14220906542088e-001 4.84614753427747e-002 -5.43528154600276e-002

the 2th row is:

-9.34475573365492e-001 -1.69155423540644e-001 2.64253389570390e-001 -2.51359648117855e-002 -1.41006587981288e-001 -7.13856249411974e-002 6.16004678917333e-003 -4.35171944717122e-002 -1.84634101647628e-002 2.07079606708212e-002

the 3th row is:

0.00000000000000e+000 -8.79921380306561e-001 -4.00770866599404e-001 3.81216014553637e-002 2.13852819649197e-001 1.08264566887620e-001 -9.34242430722824e-003 6.59988648348338e-002 2.80018996317970e-002 -3.14060203997524e-002

the 4th row is:

0.00000000000000e+000 0.00000000000000e+000 -5.37103645445220e-001 -1.26009650246059e-001 -7.06883183794930e-001 -3.57864824318144e-001 3.08810641332194e-002 -2.18156991232702e-001 -9.25593218574384e-002 1.03811526670178e-001

the 5th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 9.88778932149381e-001 -1.26609221642631e-001 -6.40968520667788e-002 5.53108007522973e-003 -3.90739056877660e-002 -1.65782182470757e-002 1.85935906958250e-002

the 6th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 5.30747773050478e-001 -6.85161829090356e-001 5.91243535211584e-002 -4.17679618069880e-001 -1.77212483467854e-001 1.98755761004241e-001

the 7th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -5.88603201003163e-001 -9.58135578050108e-002 6.76867785380382e-001 2.87180451325196e-001 -3.22092259144006e-001

the 8th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -9.92951758014159e-001 -9.99370029990069e-002 -4.24011221175398e-002 4.75557202799097e-002

the 9th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 5.37578904347965e-001 -5.61156323438901e-001 6.29374691471286e-001

the 10th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 7.46400204792486e-001 6.65497358586595e-001

A(n-1)QR分解R是:

the 1th row is:

2.51250907924776e+000 -2.18126551364497e+000 -1.31664991864811e+000 -3.23554964599634e-002 -2.55386763893328e-001 -1.26323641724277e+000 -1.42806752484401e-001 8.55112710619579e-001 -4.06432386088524e-001 3.67292064029650e-001

the 2th row is:

0.00000000000000e+000 -1.97285178971416e+000 2.27601662382715e-001 7.01463453179538e-001 5.94785406231737e-001 5.34058763390266e-001 -3.30351665591430e-001 6.13116488382732e-002 -2.84235007213628e-001 -1.02058100687656e-001

the 3th row is:

0.00000000000000e+000 0.00000000000000e+000 2.40724034343925e+000 1.72252834609436e+000 -4.50570407729081e-001 3.39244953104379e+000 -1.12144461785076e-001 -1.44880245684980e+000 7.31104559393583e-001 -4.84728580489276e-001

the 4th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 1.59561566466866e+000 6.39740664943640e-001 3.50237521521432e-001 -6.54370162142797e-002 -3.98583369940887e-001 2.39336671678428e-001 -9.05957658391911e-002

the 5th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -1.45624259345765e+000 -1.41620832333719e+000 -2.74025854657333e-001 2.82047417428750e-001 3.14637556271610e-002 2.17959389834064e-001

the 6th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 1.23967622566877e+000 1.43789368978907e-001 -1.96587654404655e-001 -5.50158837389961e-001 -1.56386794706025e-001

the 7th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -8.00193921693322e-001 3.23924067788705e-001 -4.34813370195586e-001 1.29686351399900e-001

the 8th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 1.30955871158169e+000 -4.52230033793115e-001 -2.54129774134552e-001

the 9th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -6.59108456914430e-001 4.90000710377961e-001

the 10th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 6.37333049005875e-002

A(n-1)的QR迭代结束后矩阵是is:

the 1th row is:

3.38961343881590e+000 8.02488074767549e-001 9.68945061669654e-001 8.83786659159944e-002 2.66219628377969e-001 -1.49315951793331e+000 1.20050820922893e+000 -1.07605926086027e-001 1.00470908854903e+000 -4.94758462747495e-001

the 2th row is:

2.34809943135332e-117 -2.32545916311937e+000 2.37282691474305e+000 1.58087880959695e-001 -4.76364165206190e-002 -2.03448554434184e+000 4.42214678843713e-001 1.42141927966870e-001 1.73478458088008e+000 -9.37528316136692e-001

the 3th row is:

0.00000000000000e+000 -3.36460037662303e-001 -2.34827270135765e+000 6.19924839499045e-001 -7.07167573241708e-003 1.58070491364486e+000 3.57083517747292e-002 -3.06237233807940e-001 -1.13375931449882e+000 1.99301384879556e-001

the 4th row is:

0.00000000000000e+000 0.00000000000000e+000 -2.07992595062006e-175 1.59031345880661e+000 -1.54642273213987e-002 6.84804991603965e-001 -2.36573760764005e-001 -2.69109589820564e-002 -5.12958735261066e-001 2.74410039455899e-001

the 5th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -1.04898945204601e-024 -1.49314708091493e+000 9.68734942491829e-002 -5.20431183122219e-002 6.38914814806945e-003 -3.47625905952872e-002 4.49286570719177e-002

the 6th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 1.17736987635185e-157 -7.43047298560115e-001 -5.94295473593872e-001 2.61539053547119e-001 -1.20537985514073e-001 3.89947794777855e-001

the 7th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 1.21683586982545e-001 -1.23518139438447e+000 -9.60688904735403e-002 -4.20651316598666e-001 2.16651685011453e-001

the 8th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -8.35769822314506e-021 9.43287957276840e-001 1.88977063328717e-001 -1.40375246444547e-001

the 9th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 -5.18718515391146e-146 6.48948820211121e-001 -2.77098602041316e-001

the 10th row is:

0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 0.00000000000000e+000 4.95499092363340e-002

特征值是:

特征值01是 3.38961343881590e+000

特征值01是 (-2.33686593223851e+000) + (-8.93437921021285e-001)i

特征值02是 (-2.33686593223851e+000) + (8.93437921021285e-001)i

特征值04是 1.59031345880661e+000

特征值05是 -1.49314708091493e+000

特征值05是 (-9.89114346472292e-001) + (-1.08475863150191e-001)i

特征值06是 (-9.89114346472292e-001) + (1.08475863150191e-001)i

特征值08是 9.43287957276840e-001

特征值09是 6.48948820211121e-001

特征值10是 4.95499092363340e-002

实数特征值的特征向量是:

特征值3.38961343881590e+000的特征向量是:

-1.04871999320445e-001 -2.17676976319569e-001 -4.74694012241461e-001 -2.59383624650750e-001 -3.04665248520650e-001 -2.59451746661726e-001 8.68664182733747e-002 4.05258126692653e-001 5.09628289643149e-001 2.39514692166035e-001

特征值1.59031345880661e+000的特征向量是:

6.23768976129162e-002 -1.12312295278585e-002 -2.52846032094263e-001 -1.30987581361361e-001 -3.81985138640923e-001 8.15575288836252e-001 -1.23376782911006e-001 -6.77214519898163e-002 2.71944611154593e-001 1.00282224999279e-001

特征值-1.49314708091493e+000的特征向量是:

-5.61340981697835e-001 7.78192357457906e-001 1.43637166587695e-002 -2.77601903748001e-001 3.56807241902725e-003 -2.54834165597275e-003 -2.20608987819480e-002 -1.17582711695849e-002 -1.31734984814371e-002 3.50159577287591e-002

特征值9.43287957276840e-001的特征向量是:

7.96197316848905e-002 4.54205684404738e-002 -1.82719542763806e-002 -4.79609167138929e-002 -3.49567427069969e-001 2.07214771155896e-001 -1.52312073429939e-001 8.20633710404115e-001 -3.55466329432117e-001 2.88659534097272e-002

特征值6.48948820211121e-001的特征向量是:

1.08434798576865e-001 7.13441259542963e-002 3.82501666947168e-001 -4.71003433310313e-002 -7.17803600564618e-001 1.81518546648647e-001 -2.26005938413477e-001 3.88381467696121e-001 2.89696424845602e-001 2.43327682952278e-002

特征值4.95499092363340e-002的特征向量是:

-2.13767977958844e-001 -2.06773621698937e-001 3.86828983510447e-001 -3.11123946363279e-002 -3.80938960237301e-001 -1.25173726811675e-001 6.44715735838699e-001 -3.08201272966526e-001 -2.95976727012453e-001 4.37229510135515e-002