**矩阵分析编程作业**

**[题1]**

1. 注：由于迭代次数较多，故将矩阵的迭代序列存入文件中。

A，B，C三个矩阵的QR算法迭代序列分别见文件：A\_qr\_sequence.txt

B\_qr\_sequence.txt和C\_qr\_sequence.txt

[](矩阵分析)

其中矩阵经过19次迭代可以收敛于上三角矩阵 其中精度

同样的算法作用于矩阵B,矩阵经过44次迭代可以收敛于上三角矩阵其中精度

同样的算法分别作用于矩阵B,矩阵酉相似化为上Hessenberg矩阵后经过14次迭代可以收敛于上三角矩阵（备注：更精确的值见文件C\_qr\_sequence.txt）其中精度

b) 利用QR迭代算法求出特征值，利用其中分别为(Page 54)的最大和最小特征值。矩阵A, B, C的特征值和矩阵2范数分别为：

*; 2984.1*

*; 103.3104*

*;1.4767\*10^7*

**[题2]**

1. 证明

∵ , = [],其中,其中

∴ 有： =

进而： = 整理有： =

∴ 是 的特征向量，的特征值；

又∵ 是实对称矩阵；

∴ 其特征向量组成的矩阵的列相互正交；

∴ 相互正交；

1. 的特征值和特征向量分别求出并存入文件：eigenvalues.txt和eigenvectors.txt中，如下：



在本程序中利用QR算法进行特征分解所耗时为：58.110251; 共经历了：9811次迭代；

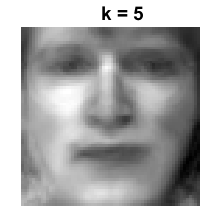
协方差矩阵最大5个特征值及相应的特征向量的结果存入了文件large\_val\_vect.txt中，如下：



这5个特征向量展成的图片为：

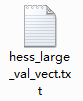


K=5的情况下将其重构为：



1. 经过hess变换后QR算法运行所需为：11.576720，共：10100次迭代；

协方差矩阵最大5个特征值及相应的特征向量的结果存入了文件hess\_large\_val\_vect.txt中，如下：



这5个特征向量展成的图片为：



**[源代码附件]**

% QR Algorithm to get eigenvalues

function T = **getFeature**(A)

% 酉相似化为上Hessenberg矩阵H

% H = hess(A);

format long e

H = A;

%print(H);

% show\_Mtrx\_in\_file(A,'A\_qr\_sequence.txt');

% 对A做第一次QR分解

[Q1,R1] = qr(H);

count = 1;

disp(['Iterator',' : ',num2str(count)]);

T = R1\*Q1;

% QR算法求特征值

while (checkValid(T))

% show\_Mtrx\_in\_file(T,'A\_qr\_sequence.txt');

[Q,R] = qr(T);

count = count + 1;

disp(['Iterator',' : ',num2str(count)]);

T = R\*Q;

end

% to check whether T is upper triangle matrix

function res = **checkValid**(T)

[n,m] = size(T);

res = 0;

for j = 1:m - 1

for i = j + 1:n

if (abs(T(i,j)) >= 10^(-10))

res = 1;

break;

end

end

end

% In this funiton, input a Matrix and store the Matrix in a file;

function **storeMatrix**(A, str)

fid = fopen(str,'a+');

format long e

[n, m] = size(A);

for i = 1:1:n

for j = 1:1:m

% 文件格式化输出

fprintf(fid,'%-.10f\t',A(i,j));

end

fprintf(fid,'\n');

end

fprintf(fid,'\n');

fclose(fid);

% In matlab, cond(M,n) can calculate the n-norm condition number

function [v,cond2] = **condNum**(A)

V1 = getFeature(A)

[n,m] = size(V1)

for i = 1:n

v(i) = V1(i,i);

end

v

T = A'\*A

V2 = getFeature(T)

max = V2(1,1)

min = V2(1,1)

[n,m] = size(V2)

for i = 1:n

if (max < V2(i,i))

max = V2(i,i);

end

if (min > V2(i,i))

min = V2(i,i);

end

end

cond2 = sqrt(max/min)

% **PCA Algorthm**

clear

str = load('yale\_face.mat');

data = str.X;

% 求均值

u = mean(data,2);

[n, m] = size(data);

for j = 1:m

for i = 1:n

X(i,j) = data(i,j) - u(i);

end

end

A = (1/m) \* X' \* X;

% [P,H] = hess(A);

% QR求特征值

tic

N = getFeature(A);

toc

% 获取特征值

for i = 1:m

v(i) = N(i,i);

end

% 特征值递减序排列

v = sort(v, 'descend')

t = v(1);

y = A - t \* eye(m);

V = null(y);

% 根据特征值求特征向量(基础解系）

for i = 2:m

% 去重特征值

if (abs(t - v(i)) > 10^(-10))

t = v(i);

y = A - t \* eye(m);

% 求基础解析，拼接成特征矩阵

yy = null(y);

V = [V,yy];

end

end

R1 = X\*V;

% 获取变换矩阵W

W = R1(:,1:5);

% 重构图片

% R2 = W \* W' \* data(:,1);

% imshow(reshape(R2(:,1),[64 64]),[]);

% 显示图片

for i = 1:5

subplot(1,5,i);imshow(reshape(W(:,i),[64 64]),[]);

end

% 文件输出结果

fid1 = fopen('eigenvalues.txt','wt'); % 写入文件路径

for i = 1:1:m

fprintf(fid1,'%-.10f\n',v(i));

end

fclose(fid1);

fid2 = fopen('eigenvectors.txt','wt');

for i = 1:1:m

for j = 1:1:m

fprintf(fid2,'%-.10f\t',V(i,j));

end

fprintf(fid2,'\n');

end

fclose(fid2);

fid3 = fopen('large\_val\_vect.txt','wt');

fprintf(fid3,'%s\n','The Largest 5 Eigenvalues of Σ : ');

for i = 1:1:5

fprintf(fid3,'%-.10f\t',v(i));

end

fprintf(fid3,'\n');

fprintf(fid3,'%s\n','The Corresponding Eigenvectors of Σ : ');

for i = 1:1:n

for j = 1:1:5

fprintf(fid3,'%-.10f\t',W(i,j));

end

fprintf(fid3,'\n');

end

fclose(fid3);