Homework 2: Computing Eigenvalues/Eigenvectors

1. This exercise asks you to write a program based on Jacobi transformations (Givens rotations) to compute eigenvalues and corresponding eigenvectors of real symmetric matrices.

***(a)** Prove or realize that all eigenvalues of a real symmetric matrix are real.

(b) Write a C/C++, Java, or Matlab program to compute eigenvalues/eigenvectors with the precision up to 10^{-4} and test the following matrices T and B.

(c) Verify your results using Matlab.

$$T = \begin{bmatrix} 1.0 & 0.6 & 0.36 & 0.216 & 0.1296 \\ 0.6 & 1.0 & 0.6 & 0.36 & 0.216 \\ 0.36 & 0.6 & 1.0 & 0.6 & 0.36 \\ 0.216 & 0.36 & 0.6 & 1.0 & 0.6 \\ 0.1296 & 0.216 & 0.36 & 0.6 & 1.0 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 4 \end{bmatrix}$$

2. Implement the following simple version of QR iteration with shift for computing the eigenvalues of a general real matrix $A = [a_{ij}]$.

Repeat

(a) $\sigma = a_{nn}$

(b) Compute QR factorization $A - \sigma I = QR$

(c) $A \leftarrow RQ + \sigma I$

Until Convergence

Q1. What convergence test should you use?

Q2. Test your program on the following matrices A and C.

Q3. Test your program on the matrices T and B in Problem **1**.

$$A = \begin{bmatrix} 11 & -12 & 8 & -4 \\ 25 & -25 & 16 & -8 \\ 7 & -6 & 2 & 0 \\ -9 & 9 & -8 & 6 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & -3 & 2 & -1 \\ 12 & -12 & 10 & -5 \\ 15 & -15 & 14 & -7 \\ 6 & -6 & 6 & -3 \end{bmatrix}.$$

Homework 2e: MatLab Programming Practice

Under Matlab environments, try to figure out the results of the following exercises.

1. linspace(a,b,n)

(a) linspace(0,10,11) (b) linspace(0,10,51)

(c) linspace(-12,12,25) (d) linspace(10,1,10)

2. eye, ones, triu, tril

(a) 3*eye(4,4)-(ones(4,4)-triu(ones(4,4),2)-tril(ones(4,4),-2))

(b) eye(5,5)-tril(ones(5,5),-1)+[1,1,1,1,0]'*[0,0,0,0,1]

3. cos, sin, mesh, meshgrid

```
R=linspace(0,5,20);
Theta=linspace(0,2*pi,11);
X=R'*cos(Theta);
Y=R'*sin(Theta);
t=linspace(-5,5,20);
[X Y]=meshgrid(t,t);
Z=2+X.^2+Y.^2;
mesh(Z)
title('An example of mesh plot')
xlabel('r*cos(x)')
ylabel('r*sin(x)')
```

4. 100-meter race

```
% Modelling the race for 100-meter athletes
fin=fopen('sprint.dat','r');
m=3;  n=11;    % fgetL(fin); skip one input line
A=fscanf(fin,'%f',[m n]);
X=0:10:100;    % X=linspace(0,100,11);
for i=1:n
  Y0(i)=A(1,i); Y1(i)=A(2,i); Y2(i)=A(3,i);
% Y0=A(1,:);
end
a=14.4;  tau=0.739;
T=0:0.2:10;  % T=linspace(0,10,21);
for j=1:51
  t=0.2*(j-1);
  X1(j)=a*tau*(t-tau*(1-exp(-t/tau)));
end
plot(T,X1,'g-',Y1,X,'bo',Y2,X,'rx'); ...
axis([0,10, 0,100]); grid
legend('Model','Carl Lewis','Ben Johnson',2)
title('A model for athletes in a race')
xlabel('Time in seconds')
ylabel('Running distance in meters')
```

5. Fast Fourier Transform (FFT)

```
% fftBox.m - Fourier Transform for the Box function
X1=linspace(0,1,17);    % X1=0:1/16:1;
X2=linspace(1,16,241);  % X2=1:1/16:16;
Y1=ones(1,length(X1));
Y2=zeros(1,length(X2));
X=[X1 X2];  Y=[Y1 Y2];
W=abs(fftshift(fft(Y)));
subplot(2,1,1)
plot(X,Y,'r');  axis([0 16,  0,1.2])
title('Box function')
subplot(2,1,2)
plot(W,'b-');
title('Fourier Spectrum of Box function')
```