22S-CSB150-EEB159 Lab2

LISA WANG

TOTAL POINTS

97 / 100

QUESTION 1

1 Ex.1 20 / 20

✓ **- 0 pts** *Correct*

- 5 pts Your final answer is correct, but I need to see your working

- 5 pts Condition for both eigenvalues to be negative is: $$b_1^2 > b_2 b_3$$

- 5 pts Condition for one positive and one negative eigenvalue is: $$b_1^2 < b_2 b_3$$

- 2 pts Need to simplify fully

QUESTION 2

2 Ex.2a 20 / 20

✓ **- 0 pts** *Correct*

- 10 pts Code for the 6 iterations should be:

```matlab
v=[1;2;0;3;1];
u1=af*v/norm(af*v);
u2=af*u1/norm(af*u1);
u3=af*u2/norm(af*u2);
u4=af*u3/norm(af*u3);
u5=af*u4/norm(af*u4);
u6=af*u5/norm(af*u5);
u6'*af*u6
```

- 10 pts `u6' * af * u6 = -3.0000`

QUESTION 3

3 Ex.2b 10 / 10

✓ **- 0 pts** *Correct*

- 5 pts `eig(af) = -3.0000`

QUESTION 4

4 Ex.3 47 / 50

- 0 pts Correct

- 5 pts Create `5 x 5` matrix and random vector:

```matlab
A = rand(5);
A = A .* A'; % random symmetric matrix
vold1=rand(5,1); % a random vector
vold2=rand(5,1); % another random vector
ll1 = []; % place holder for each iteration for first random vector
ll2 = []; % place holder for each iteration for second random vector
```

- 10 pts Correctly write `for` loop to iterate and find largest eigenvalue:

```matlab
for ii = 1:100
vnew1=A*vold1/norm(A*vold1);
ll1(ii) = double(vnew1'*A*vnew1);
vold1=vnew1;
vnew2=A*vold2/norm(A*vold2);
ll2(ii) = double(vnew2'*A*vnew2);
vold2=vnew2;
```

end

```

**- 5 pts** Verify results with `eig()`

**- 10 pts** Repeat process with a **different** random vector

**- 10 pts** Compare results for the two initial vectors

**- 3 pts** Correct answer: rate of convergence depends on initial vector. It may not look this way if you chose simple matrices and similar vectors or if you look at all 200 iterations in one plot.

**- 10 pts** Make plot of eigenvalue estimates for both cases:

```matlab
figure
hold on
plot(ll1,'r--o')
plot(ll2,'g-*')
xlabel('iteration number')
ylabel('estimated leading eigenvalue')
legend(strcat('',num2str(vold1')),strcat('',num2str(vold2')))
hold off
```

✓ **- 3 pts** *Axes labels and title*

- 50 pts not attempted
```

```
syms b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16
eig([b1, b2; b3, b4])
```

ans =

$$\begin{pmatrix} \frac{b_1}{2} + \frac{b_4}{2} - \frac{\sqrt{{b_1}^2 - 2\,b_1\,b_4 + {b_4}^2 + 4\,b_2\,b_3}}{2} \\ \frac{b_1}{2} + \frac{b_4}{2} + \frac{\sqrt{{b_1}^2 - 2\,b_1\,b_4 + {b_4}^2 + 4\,b_2\,b_3}}{2} \end{pmatrix}$$

```
eig([-b1, b2; b3, -b1])
```

ans =

$$\begin{pmatrix} -b_1 - \sqrt{b_2\,b_3} \\ \sqrt{b_2\,b_3} - b_1 \end{pmatrix}$$

```
%exericse 1 solution: -b1 - sqrt(b2b3) <0, so both eigenvalues will be
%negative when sqrt(b2*b3) < b1. One eigenvalue will be negative and the
%other will be positive when sqrt(b2*b3) > b1
af = [-1, 1.5; 1.5, -1];
eig(af)
```

```
ans = 2×1
   -2.5000
    0.5000
```

```
% one + one -

af = [-1, 0.5; 0.5, -1];
eig(af)
```

```
ans = 2×1
   -1.5000
   -0.5000
```

```
eig2 = @(b1, b2, b3, b4) b1./2 + b4./2 - (b1.^2 - 2*b1.*b4 + 4*b2.*b3 + b4.^2)^(1/2)/2;

eig([b1,b2,b3; b2,b4,b5; b3,b5,b6])
```

ans =

$$\begin{pmatrix} \frac{b_1}{3}+\frac{b_4}{3}+\frac{b_6}{3}+\frac{\sigma_3}{\sigma_2}+\sigma_2 \\ \frac{b_1}{3}+\frac{b_4}{3}+\frac{b_6}{3}-\frac{\sigma_3}{2\,\sigma_2}-\frac{\sigma_2}{2}-\sigma_1 \\ \frac{b_1}{3}+\frac{b_4}{3}+\frac{b_6}{3}-\frac{\sigma_3}{2\,\sigma_2}-\frac{\sigma_2}{2}+\sigma_1 \end{pmatrix}$$

where

$$\sigma_1 = \frac{\sqrt{3}\,\left(\frac{\sigma_3}{\sigma_2}-\sigma_2\right)\,\mathrm{i}}{2}$$

$$\sigma_2 = \left(\sqrt{\left(\sigma_4+\sigma_8+\sigma_7+\sigma_6-\sigma_5-b_2\,b_3\,b_5-\frac{b_1\,b_4\,b_6}{2}\right)^2-\sigma_3{}^3}-\sigma_8-\sigma_7-\sigma_6-\sigma_4+\sigma_5+b_2\,b_3\,b_5+\frac{b_1\,b_4\,b}{2}\right.$$

$$\sigma_3 = \frac{b_2{}^2}{3}-\frac{b_1\,b_6}{3}-\frac{b_4\,b_6}{3}-\frac{b_1\,b_4}{3}+\frac{b_3{}^2}{3}+\frac{b_5{}^2}{3}+\frac{(b_1+b_4+b_6)^2}{9}$$

$$\sigma_4 = \frac{(b_1+b_4+b_6)\,\left(-b_2{}^2-b_3{}^2-b_5{}^2+b_1\,b_4+b_1\,b_6+b_4\,b_6\right)}{6}$$

$$\sigma_5 = \frac{(b_1+b_4+b_6)^3}{27}$$

$$\sigma_6 = \frac{b_2{}^2\,b_6}{2}$$

$$\sigma_7 = \frac{b_3{}^2\,b_4}{2}$$

$$\sigma_8 = \frac{b_1\,b_5{}^2}{2}$$

```
% a much more complex answer here

%eig([b1, b2, b3, b4; b2, b5, b6, b7; b3, b6, b8, b9; b4, b7, b9, b10])
```

```
syms c;
det([b1-c, b2; b3, b4-c])
```

ans = $b_1\,b_4-b_2\,b_3-b_1\,c-b_4\,c+c^2$

```
p = [1 -(b1 + b4) b1*b4-b2*b3];
r = roots(p);

p2 = []; %write the coefficients for your 5th-rder polynomials here.
r = roots(p2);

af=[-1, 0.5, 0.5, 0.5, 0.5;
0.5, -1, 0.5, 0.5, 0.5;
0.5, 0.5, -1, 0.5, 0.5;
0.5, 0.5, 0.5, -1, 0.5;
0.5, 0.5, 0.5, 0.5, -1];
eig(af)
```

```
ans = 5×1
   -1.5000
   -1.5000
   -1.5000
   -1.5000
    1.0000
```

```

% these 0.5s are coefficients for my 5th order polynomial, matrix vector
% form, theorem does not apply

af=[-1, -0.5, -0.5, -0.5, -0.5;
-0.5, -1, -0.5, -0.5, -0.5;
-0.5, -0.5, -1, -0.5, -0.5;
-0.5, -0.5, -0.5, -1, -0.5;
-0.5, -0.5, -0.5, -0.5, -1];
eig(af)
```

```
ans = 5×1
   -3.0000
   -0.5000
   -0.5000
   -0.5000
   -0.5000
```

```
m = [1, 10; 10, 1];
v = [1;1];
i1 = m*v / norm(m*v)
```

```
i1 = 2×1
    0.7071
    0.7071
```

```
i1'*m*i1
```

```
ans = 11.0000
```

```

eig(m)
```

```
ans = 2×1
```

```
   -9.0000
   11.0000
```

```
m = [2, 4;7,3];
v = [1;1];
i1 = m*v / norm(m*v);
c(1) = double(i1'*m*i1)
```

c =

$$\frac{129}{17}$$

```
i2=m*i1/norm(m*i1);
c(2)=double(i2'*m*i2)
```

c =

$$\begin{pmatrix}\frac{129}{17} & \frac{3884}{493}\end{pmatrix}$$

```
i3=m*i2/norm(m*i2);
c(3)=double(i3'*m*i3)
```

c =

$$\begin{pmatrix}\frac{129}{17} & \frac{3884}{493} & \frac{1096313283462169}{140737488355328}\end{pmatrix}$$

```
i4=m*i3/norm(m*i3);
c(4)=double(i4'*m*i4)
```

c =

$$\begin{pmatrix}\frac{129}{17} & \frac{3884}{493} & \frac{1096313283462169}{140737488355328} & \frac{275277734408835}{35184372088832}\end{pmatrix}$$

```
i5=m*i4/norm(m*i4);
c(5)=double(i5'*m*i5)
```

c =

$$\begin{pmatrix}\frac{129}{17} & \frac{3884}{493} & \frac{1096313283462169}{140737488355328} & \frac{275277734408835}{35184372088832} & \frac{8795377287796131}{1125899906842624}\end{pmatrix}$$

```
i6=m*i5/norm(m*i5);
c(6)=double(i6'*m*i6)
```

c =

$$\begin{pmatrix}\frac{129}{17} & \frac{3884}{493} & \frac{1096313283462169}{140737488355328} & \frac{275277734408835}{35184372088832} & \frac{8795377287796131}{1125899906842624} & \frac{2200071254027361}{281474976710656}\end{pmatrix}$$

```
i7=m*i6/norm(m*i6);
c(7)=double(i7'*m*i7)
```

c =

$$\left(\frac{129}{17} \quad \frac{3884}{493} \quad \frac{1096313283462169}{140737488355328} \quad \frac{275277734408835}{35184372088832} \quad \frac{8795377287796131}{1125899906842624} \quad \frac{2200071254027361}{281474976710656} \quad \frac{2199}{2814}\right.$$

```
i8=m*i7/norm(m*i7);
c(8)=double(i8'*m*i8)
```

c =

$$\left(\frac{129}{17} \quad \frac{3884}{493} \quad \frac{1096313283462169}{140737488355328} \quad \frac{275277734408835}{35184372088832} \quad \frac{8795377287796131}{1125899906842624} \quad \frac{2200071254027361}{281474976710656} \quad \frac{2199}{2814}\right.$$

```
i9=m*i8/norm(m*i8);
c(9)=double(i9'*m*i9)
```

c =

$$\left(\frac{129}{17} \quad \frac{3884}{493} \quad \frac{1096313283462169}{140737488355328} \quad \frac{275277734408835}{35184372088832} \quad \frac{8795377287796131}{1125899906842624} \quad \frac{2200071254027361}{281474976710656} \quad \frac{2199}{2814}\right.$$

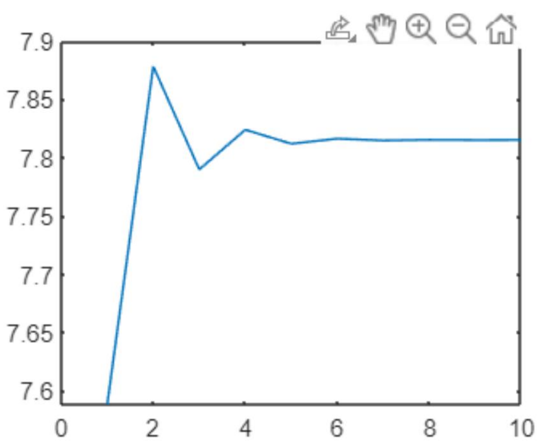
```
i10=m*i9/norm(m*i9);
c(10)=double(i10'*m*i10)
```

c =

$$\left(\frac{129}{17} \quad \frac{3884}{493} \quad \frac{1096313283462169}{140737488355328} \quad \frac{275277734408835}{35184372088832} \quad \frac{8795377287796131}{1125899906842624} \quad \frac{2200071254027361}{281474976710656} \quad \frac{2199}{2814}\right.$$

```
% the outputs eigenvalues are infinitely closing into the value 7.78977 or
% approximately, this is called convergence

plot(c)
```




```
% from the plot on the right, we can see that the eigenvalue vector
% undulates then converges around 7.7-7.8 value range
```

```
%exercise 2
```

```
af = [-1, -0.5, -0.5, -0.5, -0.5;
    -0.5, -1, -0.5, -0.5, -0.5;
    -0.5, -0.5, -1, -0.5, -0.5;
    -0.5, -0.5, -0.5, -1, -0.5;
    -0.5, -0.5, -0.5, -0.5, -1];

v= [1;2;0;3;1];
u1 = af*v/norm(af*v);
u2 = af*u1/norm(af*u1);
u3 = af*u2/norm(af*u2);
u4 = af*u3/norm(af*u3);
u5 = af*u4/norm(af*u4);
u6 = af*u5/norm(af*u5);


u1'*af*u1
```

```
ans = -2.9637
```

```
u2'*af*u2
```

```
ans = -2.9990
```

```
u3'*af*u3
```

```
ans = -3.0000
```

```
u4'*af*u4
```

```
ans = -3.0000
```

```
u5'*af*u5
```

```
ans = -3.0000
```

```
u6'*af*u6
```

```
ans = -3.0000
```

```
eig(af)
```

```
ans = 5×1
   -3.0000
   -0.5000
   -0.5000
   -0.5000
   -0.5000
```

```
eigtrue = eig(af);
difference = abs(u6'*af*u6) - max(abs(eigtrue));

%answering the question in 2b: how close is the largest eigenvalue from the
%calculated approximation in magnitude?
%
```

```
% The substraction shows that the result is a miniscule number, although
% not zero, so we can conclude that the approximation is very close but not
% the same.
```

```
% exercise 3
af = rand(5,5);

v= rand(5,1);

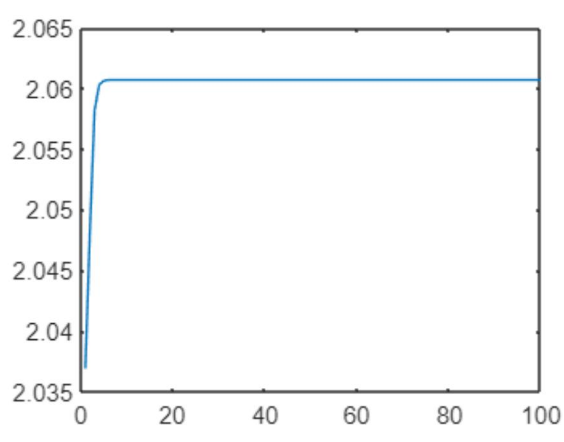
u1 = af*v/norm(af*v);
%u = size(100);
v2 = rand(5,1);
array = [];

for i = 1:100
   % u(i+1) = af*u(i) / norm(af*u(i));
   v2 = af*v/norm(af*v);
   array(i) = v2'* af *v2;
   % u(i)' * af*u(i);
   v =v2;
end

eig(af)
```

```
ans = 5×1 complex
   2.0607 + 0.0000i
  -0.1945 + 0.0000i
   0.2540 + 0.0000i
   0.1031 + 0.0542i
   0.1031 - 0.0542i
```

```
plot(array);
```




```
%next, construct a new random vector

af = rand(5,5);
v = rand(5,1);
```

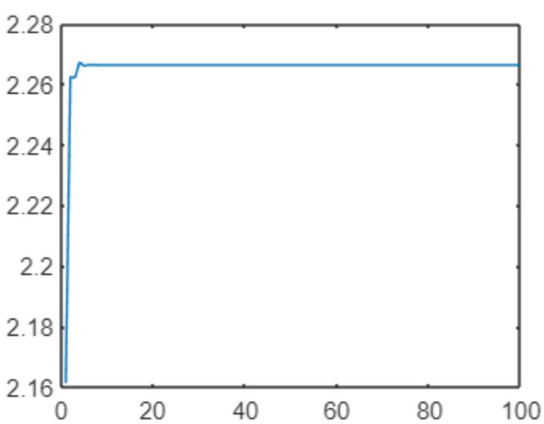
```
u1 = af*v/norm(af*v);
%u = size(100);
v2 = rand(5,1);
array = [];

for i = 1:100
   % u(i+1) = af*u(i) / norm(af*u(i));
   v2 = af*v/norm(af*v);
   array(i) = v2'* af *v2;
   % u(i)' * af*u(i);
   v =v2;
end

eig(af)
```

```
ans = 5×1 complex
   2.2663 + 0.0000i
  -0.7731 + 0.0000i
   0.1610 + 0.1571i
   0.1610 - 0.1571i
  -0.3282 + 0.0000i
```

```
plot(array);
```




```
%a plot of eigenvalue estimates for all iterations to look at convergence
%in both cases
%Answer to the question in 3: in both cases, because I was told that af
%should be a random 5x5 matrix, the convergence goes to different values,
%so it is appr. 2. 06 in the first and 2.26 in the second.The initial guess
%does not affect the convergence because eventually over many iterations,
%the true eigenvalue will be aligned to the estimation. The convergence
%value should be the same, but the rate of convergence may differ depending
%on the initial choice of the vector v.

%If you choose a non-symmetric matrix, and your eigenvalue with largest real part is a complex
%happens as you iterate? Note: upper bound of oscillations is magnitude of largest eigenvalue.
%Answer: After iteration, the complex conjugate pair does not take on the
```

```
%imaginary part (c1*i) where c1 denotes the coefficient of the imaginary
%part.
```