MATLAB PROJECT 4

Please include this page in your Group file, as a front page. Type in the group number and the names of all members WHO PARTICIPATED in this project.

GROUP # \_\_\_\_3\_\_\_\_\_

FIRST & LAST NAMES (UFID numbers are NOT required):

\_1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Kevin Nguyen \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Chamara Gunaratne\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ­­­

\_3.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Cayle Gao\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_4.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Carl Lin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_5.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Nicolas Miller\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_6.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**By signing your names above, each of you had confirmed that you did the work and agree with the work submitted**.

%Exercise 1

type closetozeroroundoff\_1

function B=closetozeroroundoff\_1(A)

[m,n]=size(A);

for i=1:m

for j=1:n

if abs(A(i,j))<10^(-12)

A(i,j)=0;

end

end

end

B=A;

end

type jord

function J=jord(n,r)

A=ones(n);

J=tril(triu(A),1);

for i=1:n

J(i,i)=r;

end

end

type eigenval

function [] = eigenval(A)

P=ones(size(A));

while any(any(P)~= 0)

[Q,R]=qr(A);

A=R\*Q;

P=closetozeroroundoff\_1(A-triu(A));

end

B=A;

disp('The eigenvalues of A from QR factorization are:')

e=sort(diag(B))

disp('The eigenvalues of A from a MATLAB in-built function are:')

p=sort(eig(A))

if norm(e-p)<10^(-5)

disp('The eigenvalues match')

r=poly(p)

R=poly2sym(r)

disp('The coefficients of the MATLAB characteristic polynomial are:')

C=poly(A)

disp('The eigenvalues from the MATLAB characteristic polynomial are:')

c=sort(roots(C))

else

disp('Check the code!')

end

end

A=[3 3; 0 3]

A =

3 3

0 3

eigenval(A)

The eigenvalues of A from QR factorization are:

e =

3

3

The eigenvalues of A from a MATLAB in-built function are:

p =

3

3

The eigenvalues match

r =

1 -6 9

R =

x^2 - 6\*x + 9

The coefficients of the MATLAB characteristic polynomial are:

C =

1 -6 9

The eigenvalues from the MATLAB characteristic polynomial are:

c =

3.0000 - 0.0000i

3.0000 + 0.0000i

A=[4 0 0 0;1 3 0 0; 0 -1 3 0; 0 -1 5 4]

A =

4 0 0 0

1 3 0 0

0 -1 3 0

0 -1 5 4

eigenval(A)

The eigenvalues of A from QR factorization are:

e =

3.0000

3.0000

4.0000

4.0000

The eigenvalues of A from a MATLAB in-built function are:

p =

3.0000 - 0.0000i

3.0000 + 0.0000i

4.0000 - 0.0000i

4.0000 + 0.0000i

The eigenvalues match

r =

1.0000 -14.0000 73.0000 -168.0000 144.0000

R =

x^4 - (492581209243671\*x^3)/35184372088832 + (1284229581242485\*x^2)/17592186044416 - (2955487255462281\*x)/17592186044416 + 5066549580792679/35184372088832

The coefficients of the MATLAB characteristic polynomial are:

C =

1.0000 -14.0000 73.0000 -168.0000 144.0000

The eigenvalues from the MATLAB characteristic polynomial are:

c =

3.0000 - 0.0000i

3.0000 + 0.0000i

4.0000 - 0.0000i

4.0000 + 0.0000i

A=ones(5)

A =

1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

1 1 1 1 1

eigenval(A)

The eigenvalues of A from QR factorization are:

e =

0

0

0

0.0000

5.0000

The eigenvalues of A from a MATLAB in-built function are:

p =

0

0

0

0.0000

5.0000

The eigenvalues match

r =

Columns 1 through 5

1.0000 -5.0000 0.0000 0 0

Column 6

0

R =

x^5 - 5\*x^4 + (2517588727560787\*x^3)/2535301200456458802993406410752

The coefficients of the MATLAB characteristic polynomial are:

C =

Columns 1 through 5

1.0000 -5.0000 0.0000 0 0

Column 6

0

The eigenvalues from the MATLAB characteristic polynomial are:

c =

0

0

0

0.0000

5.0000

A=jord(4,3)

A =

3 1 0 0

0 3 1 0

0 0 3 1

0 0 0 3

eigenval(A)

The eigenvalues of A from QR factorization are:

e =

3

3

3

3

The eigenvalues of A from a MATLAB in-built function are:

p =

3

3

3

3

The eigenvalues match

r =

1 -12 54 -108 81

R =

x^4 - 12\*x^3 + 54\*x^2 - 108\*x + 81

The coefficients of the MATLAB characteristic polynomial are:

C =

1 -12 54 -108 81

The eigenvalues from the MATLAB characteristic polynomial are:

c =

2.9996 - 0.0004i

2.9996 + 0.0004i

3.0004 - 0.0004i

3.0004 + 0.0004i

%Vectors c and p both consider imaginary values in the eigenvalues

% Exercise 2

type eigen

function [] = eigen(A)

L = sort(transpose(closetozeroroundoff(eig(A))))

M = unique(L)

W = eye(size(A));

P = [];

for i=1:size(M,2)

count = 0;

for j=1:size(L,2)

if(L(j) == M(i))

count = count + 1;

end

end

fprintf('Eigenvalue %d has multiplicity %i\n',M(i),count)

W = null(A-(M(i)\*eye(size(A))),'r');

fprintf('A basis for eigenspace for lambda = %d is: ', M(i))

W

P = [P W];

d = size(W,2);

fprintf('Dimension of eigenspace for lambda = %d is %i\n',M(i),d)

end

if (count == d)

D = diag(L)

P

F = closetozeroroundoff(A\*P - P\*D);

if (F == zeros(size(F)))

disp('Great! I got my diagonalization!')

else

disp('Oops! I got a bug in my code!')

end

else

disp('Matrix A is not diagonalizable')

return;

end

end

type closetozeroroundoff

function B = closetozeroroundoff(A)

[m,n]=size(A);

for i=1:m

for j=1:n

if abs(A(i,j))<10^(-7)

A(i,j)=0;

end

end

end

B=A;

end

type jord

function J = jord(n,r)

A=ones(n);

J=tril(triu(A),1);

for i=1:n

J(i,i)=r;

end

end

% A

A = [3 3; 0 3]

A =

3 3

0 3

eigen(A)

L =

3 3

M =

3

Eigenvalue 3 has multiplicity 2

A basis for eigenspace for lambda = 3 is: W =

1

0

Dimension of eigenspace for lambda = 3 is 1

Matrix A is not diagonalizable

% B

A = [4 0 0 0; 1 3 0 0; 0 -1 3 0; 0 -1 5 4]

A =

4 0 0 0

1 3 0 0

0 -1 3 0

0 -1 5 4

eigen(A)

L =

3 3 4 4

M =

3 4

Eigenvalue 3 has multiplicity 2

A basis for eigenspace for lambda = 3 is: W =

0

0

-0.2000

1.0000

Dimension of eigenspace for lambda = 3 is 1

Eigenvalue 4 has multiplicity 2

A basis for eigenspace for lambda = 4 is: W =

0

0

0

1

Dimension of eigenspace for lambda = 4 is 1

Matrix A is not diagonalizable

% C

A = jord(5,4)

A =

4 1 0 0 0

0 4 1 0 0

0 0 4 1 0

0 0 0 4 1

0 0 0 0 4

eigen(A)

L =

4 4 4 4 4

M =

4

Eigenvalue 4 has multiplicity 5

A basis for eigenspace for lambda = 4 is: W =

1

0

0

0

0

Dimension of eigenspace for lambda = 4 is 1

Matrix A is not diagonalizable

% D

A = diag([3, 3, 3, 2, 2, 1])

A =

3 0 0 0 0 0

0 3 0 0 0 0

0 0 3 0 0 0

0 0 0 2 0 0

0 0 0 0 2 0

0 0 0 0 0 1

eigen(A)

L =

1 2 2 3 3 3

M =

1 2 3

Eigenvalue 1 has multiplicity 1

A basis for eigenspace for lambda = 1 is: W =

0

0

0

0

0

1

Dimension of eigenspace for lambda = 1 is 1

Eigenvalue 2 has multiplicity 2

A basis for eigenspace for lambda = 2 is: W =

0 0

0 0

0 0

1 0

0 1

0 0

Dimension of eigenspace for lambda = 2 is 2

Eigenvalue 3 has multiplicity 3

A basis for eigenspace for lambda = 3 is: W =

1 0 0

0 1 0

0 0 1

0 0 0

0 0 0

0 0 0

Dimension of eigenspace for lambda = 3 is 3

D =

1 0 0 0 0 0

0 2 0 0 0 0

0 0 2 0 0 0

0 0 0 3 0 0

0 0 0 0 3 0

0 0 0 0 0 3

P =

0 0 0 1 0 0

0 0 0 0 1 0

0 0 0 0 0 1

0 1 0 0 0 0

0 0 1 0 0 0

1 0 0 0 0 0

Great! I got my diagonalization!

% E

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

eigen(A)

L =

-8.9443 0 8.9443 34.0000

M =

-8.9443 0 8.9443 34.0000

Eigenvalue -8.944272e+00 has multiplicity 1

A basis for eigenspace for lambda = -8.944272e+00 is: W =

-0.4570

-0.0287

-0.5143

1.0000

Dimension of eigenspace for lambda = -8.944272e+00 is 1

Eigenvalue 0 has multiplicity 1

A basis for eigenspace for lambda = 0 is: W =

-1

-3

3

1

Dimension of eigenspace for lambda = 0 is 1

Eigenvalue 8.944272e+00 has multiplicity 1

A basis for eigenspace for lambda = 8.944272e+00 is: W =

-2.1882

1.1254

0.0627

1.0000

Dimension of eigenspace for lambda = 8.944272e+00 is 1

Eigenvalue 3.400000e+01 has multiplicity 1

A basis for eigenspace for lambda = 3.400000e+01 is: W =

1.0000

1.0000

1.0000

1.0000

Dimension of eigenspace for lambda = 3.400000e+01 is 1

D =

-8.9443 0 0 0

0 0 0 0

0 0 8.9443 0

0 0 0 34.0000

P =

-0.4570 -1.0000 -2.1882 1.0000

-0.0287 -3.0000 1.1254 1.0000

-0.5143 3.0000 0.0627 1.0000

1.0000 1.0000 1.0000 1.0000

Great! I got my diagonalization!

% F

A = ones(4)

A =

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

eigen(A)

L =

0 0 0 4.0000

M =

0 4.0000

Eigenvalue 0 has multiplicity 3

A basis for eigenspace for lambda = 0 is: W =

-1 -1 -1

1 0 0

0 1 0

0 0 1

Dimension of eigenspace for lambda = 0 is 3

Eigenvalue 4.000000e+00 has multiplicity 1

A basis for eigenspace for lambda = 4.000000e+00 is: W =

1.0000

1.0000

1.0000

1.0000

Dimension of eigenspace for lambda = 4.000000e+00 is 1

D =

0 0 0 0

0 0 0 0

0 0 0 0

0 0 0 4.0000

P =

-1.0000 -1.0000 -1.0000 1.0000

1.0000 0 0 1.0000

0 1.0000 0 1.0000

0 0 1.0000 1.0000

Great! I got my diagonalization!

% G

A = magic(5)

A =

17 24 1 8 15

23 5 7 14 16

4 6 13 20 22

10 12 19 21 3

11 18 25 2 9

eigen(A)

L =

-21.2768 -13.1263 13.1263 21.2768 65.0000

M =

-21.2768 -13.1263 13.1263 21.2768 65.0000

Eigenvalue -2.127677e+01 has multiplicity 1

A basis for eigenspace for lambda = -2.127677e+01 is: W =

-0.1440

-0.5200

-0.8114

0.4753

1.0000

Dimension of eigenspace for lambda = -2.127677e+01 is 1

Eigenvalue -1.312628e+01 has multiplicity 1

A basis for eigenspace for lambda = -1.312628e+01 is: W =

-2.4172

2.2511

-1.4952

0.6613

1.0000

Dimension of eigenspace for lambda = -1.312628e+01 is 1

Eigenvalue 1.312628e+01 has multiplicity 1

A basis for eigenspace for lambda = 1.312628e+01 is: W =

-0.4137

-0.2736

0.6186

-0.9313

1.0000

Dimension of eigenspace for lambda = 1.312628e+01 is 1

Eigenvalue 2.127677e+01 has multiplicity 1

A basis for eigenspace for lambda = 2.127677e+01 is: W =

5◊0 empty <a href="matlab:helpPopup double" style="font-weight:bold">double</a> matrix

Dimension of eigenspace for lambda = 2.127677e+01 is 0

Eigenvalue 6.500000e+01 has multiplicity 1

A basis for eigenspace for lambda = 6.500000e+01 is: W =

1.0000

1.0000

1.0000

1.0000

1.0000

Dimension of eigenspace for lambda = 6.500000e+01 is 1

D =

-21.2768 0 0 0 0

0 -13.1263 0 0 0

0 0 13.1263 0 0

0 0 0 21.2768 0

0 0 0 0 65.0000

P =

-0.1440 -2.4172 -0.4137 1.0000

-0.5200 2.2511 -0.2736 1.0000

-0.8114 -1.4952 0.6186 1.0000

0.4753 0.6613 -0.9313 1.0000

1.0000 1.0000 1.0000 1.0000

{\_Error using <a href="matlab:matlab.internal.language.introspective.errorDocCallback('mtimes')" style="font-weight:bold"> \* </a>

Inner matrix dimensions must agree.

Error in <a href="matlab:matlab.internal.language.introspective.errorDocCallback('eigen', 'M:\eigen.m', 24)" style="font-weight:bold">eigen</a> (<a href="matlab: opentoline('M:\eigen.m',24,0)">line 24</a>)

F = closetozeroroundoff(A\*P - P\*D);}

\_

% The matrix P in part G is incorrect because one of the eigenvalues had created a 5x0 empty matrix.

type eigen\_1

function [] = eigen\_1(A)

L = sort(transpose(closetozeroroundoff(eig(A))))

M = unique(L)

W = eye(size(A));

P = [];

for i=1:size(M,2)

count = 0;

for j=1:size(L,2)

if(L(j) == M(i))

count = count + 1;

end

end

fprintf('Eigenvalue %d has multiplicity %i\n',M(i),count)

W = null(A-(M(i)\*eye(size(A))));

fprintf('A basis for eigenspace for lambda = %d is: ', M(i))

W

P = [P W];

d = size(W,2);

fprintf('Dimension of eigenspace for lambda = %d is %i\n',M(i),d)

end

if (count == d)

D = diag(L)

P

F = closetozeroroundoff(A\*P - P\*D);

if (F == zeros(size(F)))

disp('Great! I got my diagonalization!')

else

disp('Oops! I got a bug in my code!')

end

else

disp('Matrix A is not diagonalizable')

return;

end

end

% G (Attempt #2)

A = magic(5)

A =

17 24 1 8 15

23 5 7 14 16

4 6 13 20 22

10 12 19 21 3

11 18 25 2 9

eigen\_1(A)

L =

-21.2768 -13.1263 13.1263 21.2768 65.0000

M =

-21.2768 -13.1263 13.1263 21.2768 65.0000

Eigenvalue -2.127677e+01 has multiplicity 1

A basis for eigenspace for lambda = -2.127677e+01 is: W =

-0.0976

-0.3525

-0.5501

0.3223

0.6780

Dimension of eigenspace for lambda = -2.127677e+01 is 1

Eigenvalue -1.312628e+01 has multiplicity 1

A basis for eigenspace for lambda = -1.312628e+01 is: W =

-0.6330

0.5895

-0.3915

0.1732

0.2619

Dimension of eigenspace for lambda = -1.312628e+01 is 1

Eigenvalue 1.312628e+01 has multiplicity 1

A basis for eigenspace for lambda = 1.312628e+01 is: W =

0.2619

0.1732

-0.3915

0.5895

-0.6330

Dimension of eigenspace for lambda = 1.312628e+01 is 1

Eigenvalue 2.127677e+01 has multiplicity 1

A basis for eigenspace for lambda = 2.127677e+01 is: W =

0.6780

0.3223

-0.5501

-0.3525

-0.0976

Dimension of eigenspace for lambda = 2.127677e+01 is 1

Eigenvalue 6.500000e+01 has multiplicity 1

A basis for eigenspace for lambda = 6.500000e+01 is: W =

-0.4472

-0.4472

-0.4472

-0.4472

-0.4472

Dimension of eigenspace for lambda = 6.500000e+01 is 1

D =

-21.2768 0 0 0 0

0 -13.1263 0 0 0

0 0 13.1263 0 0

0 0 0 21.2768 0

0 0 0 0 65.0000

P =

-0.0976 -0.6330 0.2619 0.6780 -0.4472

-0.3525 0.5895 0.1732 0.3223 -0.4472

-0.5501 -0.3915 -0.3915 -0.5501 -0.4472

0.3223 0.1732 0.5895 -0.3525 -0.4472

0.6780 0.2619 -0.6330 -0.0976 -0.4472

Great! I got my diagonalization!

%Exercise 3

type diagon

function []=diagon(A)

[~,n]=size(A);

[P,D]=eig(A)

if A\*P==P\*D

disp('A is diagonalizable')

fprintf('An eigenvector basis for R^%i is\n',n)

P

else

('A is not diagonalizable')

fprintf('P is not a basis for R^%i \n',n)

end

L=eig(D)

end

type jord

function J=jord(n,r)

A=ones(n);

J=tril(triu(A),1);

for i=1:n

J(i,i)=r;

end

end

%(a)

A= [3 3; 0 3]

A =

3 3

0 3

diagon(A)

P =

1.0000 -1.0000

0 0.0000

D =

3 0

0 3

ans =

'A is not diagonalizable'

P is not a basis for R^2

L =

3

3

%(b)

A= [4 0 0 0; 1 3 0 0; 0 -1 3 0; 0 -1 5 4]

A =

4 0 0 0

1 3 0 0

0 -1 3 0

0 -1 5 4

diagon(A)

P =

0 0 0 0.0000

0 0 0.0000 0.0000

0 0.1961 0.1961 -0.0000

1.0000 -0.9806 -0.9806 1.0000

D =

4 0 0 0

0 3 0 0

0 0 3 0

0 0 0 4

ans =

'A is not diagonalizable'

P is not a basis for R^4

L =

3

3

4

4

%(c)

A = jord(5,4)

A =

4 1 0 0 0

0 4 1 0 0

0 0 4 1 0

0 0 0 4 1

0 0 0 0 4

diagon(A)

P =

Columns 1 through 4

1.0000 -1.0000 1.0000 -1.0000

0 0.0000 -0.0000 0.0000

0 0 0.0000 -0.0000

0 0 0 0.0000

0 0 0 0

Column 5

1.0000

-0.0000

0.0000

-0.0000

0.0000

D =

4 0 0 0 0

0 4 0 0 0

0 0 4 0 0

0 0 0 4 0

0 0 0 0 4

ans =

'A is not diagonalizable'

P is not a basis for R^5

L =

4

4

4

4

4

%(d)

A = diag([3, 3, 3, 2, 2, 1])

A =

3 0 0 0 0 0

0 3 0 0 0 0

0 0 3 0 0 0

0 0 0 2 0 0

0 0 0 0 2 0

0 0 0 0 0 1

diagon(A)

P =

0 0 0 0 0 1

0 0 0 1 0 0

0 0 0 0 1 0

0 1 0 0 0 0

0 0 1 0 0 0

1 0 0 0 0 0

D =

1 0 0 0 0 0

0 2 0 0 0 0

0 0 2 0 0 0

0 0 0 3 0 0

0 0 0 0 3 0

0 0 0 0 0 3

A is diagonalizable

An eigenvector basis for R^6 is

P =

0 0 0 0 0 1

0 0 0 1 0 0

0 0 0 0 1 0

0 1 0 0 0 0

0 0 1 0 0 0

1 0 0 0 0 0

L =

1

2

2

3

3

3

%(e)

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

diagon(A)

P =

-0.5000 -0.8236 0.3764 -0.2236

-0.5000 0.4236 0.0236 -0.6708

-0.5000 0.0236 0.4236 0.6708

-0.5000 0.3764 -0.8236 0.2236

D =

34.0000 0 0 0

0 8.9443 0 0

0 0 -8.9443 0

0 0 0 0.0000

ans =

'A is not diagonalizable'

P is not a basis for R^4

L =

-8.9443

0.0000

8.9443

34.0000

%(f)

A= ones(4)

A =

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

diagon(A)

P =

0.0846 0.4928 0.7071 0.5000

0.0846 0.4928 -0.7071 0.5000

-0.7815 -0.3732 0 0.5000

0.6124 -0.6124 0 0.5000

D =

-0.0000 0 0 0

0 -0.0000 0 0

0 0 0 0

0 0 0 4.0000

ans =

'A is not diagonalizable'

P is not a basis for R^4

L =

-0.0000

-0.0000

0

4.0000

%(g)

A = magic(5)

A =

17 24 1 8 15

23 5 7 14 16

4 6 13 20 22

10 12 19 21 3

11 18 25 2 9

diagon(A)

P =

Columns 1 through 4

-0.4472 0.0976 -0.6330 0.6780

-0.4472 0.3525 0.5895 0.3223

-0.4472 0.5501 -0.3915 -0.5501

-0.4472 -0.3223 0.1732 -0.3525

-0.4472 -0.6780 0.2619 -0.0976

Column 5

-0.2619

-0.1732

0.3915

-0.5895

0.6330

D =

Columns 1 through 4

65.0000 0 0 0

0 -21.2768 0 0

0 0 -13.1263 0

0 0 0 21.2768

0 0 0 0

Column 5

0

0

0

0

13.1263

ans =

'A is not diagonalizable'

P is not a basis for R^5

L =

-21.2768

-13.1263

13.1263

21.2768

65.0000

%the outputs for the matrix in (g) in this exercise and for the function eigen\_1 in Exercise 2 do not match

% Exercise 4

type shrink

function B=shrink(A)

format compact

[~,pivot]=rref(A);

B=A(:,pivot);

end

type proj

function [p,z]=proj(A,b)

format compact

A=shrink(A);

b=transpose(b);

[m,~]=size(A);

if numel(b)~=m

disp('No solution:dimensions of A and b disagree')

p=[];

z=[];

return

end

if rank(A) == rank([A b])

p=transpose(b);

z=transpose(b)-p;

disp('b is in the Col A')

return

end

C=0;

for i=1:size(A,2)

C(:,i) = dot(A(:,i),b);

end

C=sum(C); if C == 0

p=transpose(b);

z=transpose(b)-p;

disp('b is orthogonal to the Col A')

return

end

x=A\b;

p=A\*x;

z=b-p;

if closetozeroroundoff(transpose(p)\*z)==0

disp ('Yes, p and z are orthogonal! Great Job!')

else

disp('What is wrong?!')

end

end

%(a)

A= magic(4); A=A( : , 1 : 3), b = (1 : 4)

A =

16 2 3

5 11 10

9 7 6

4 14 15

b =

1 2 3 4

[p,z]=proj(A,b)

Yes, p and z are orthogonal! Great Job!

p =

1.3000

2.9000

2.1000

3.7000

z =

-0.3000

-0.9000

0.9000

0.3000

%(b)

A= magic(6), E= eye(6); b = E( 6, :)

A =

35 1 6 26 19 24

3 32 7 21 23 25

31 9 2 22 27 20

8 28 33 17 10 15

30 5 34 12 14 16

4 36 29 13 18 11

b =

0 0 0 0 0 1

[p,z]=proj(A,b)

Yes, p and z are orthogonal! Great Job!

p =

-0.2500

0.0000

0.2500

0.2500

-0.0000

0.7500

z =

0.2500

-0.0000

-0.2500

-0.2500

0.0000

0.2500

%(c)

A = magic(6), b = (1 : 5)

A =

35 1 6 26 19 24

3 32 7 21 23 25

31 9 2 22 27 20

8 28 33 17 10 15

30 5 34 12 14 16

4 36 29 13 18 11

b =

1 2 3 4 5

[p,z]=proj(A,b)

No solution:dimensions of A and b disagree

p =

[]

z =

[]

%(d)

A = magic(5), b = rand(1,5)

A =

17 24 1 8 15

23 5 7 14 16

4 6 13 20 22

10 12 19 21 3

11 18 25 2 9

b =

Columns 1 through 4

0.8147 0.9058 0.1270 0.9134

Column 5

0.6324

[p,z]=proj(A,b)

b is in the Col A

p =

Columns 1 through 4

0.8147 0.9058 0.1270 0.9134

Column 5

0.6324

z =

0 0 0 0 0

%(e)

A= ones(4); A( : ) = 1 : 16, b = [1,0,1,0]

A =

1 5 9 13

2 6 10 14

3 7 11 15

4 8 12 16

b =

1 0 1 0

[p,z]=proj(A,b)

Yes, p and z are orthogonal! Great Job!

p =

0.8000

0.6000

0.4000

0.2000

z =

0.2000

-0.6000

0.6000

-0.2000

%(f)

B=ones(4); B( : ) = 1 : 16; A= null(B,'r'), b = ones(1,4)

A =

1 2

-2 -3

1 0

0 1

b =

1 1 1 1

[p,z]=proj(A,b)

b is orthogonal to the Col A

p =

1 1 1 1

z =

0 0 0 0

%Exercise 5

type polyplot

function [] = polyplot(a,b,p)

x=(a:(b-a)/50:b)';

y=polyval(p,x);

plot(x,y);

end

type lstsqline

function c = lstsqline(x,y)

format rat

x=x';

y=y';

a=x(1);

m=length(x);

b=x(m);

X=[x,ones(m,1)];

c=lscov(X,y);

c1=(inv(X'\*X))\*(X'\*y)

c2=(X'\*X)\(X'\*y)

N=norm(y-X\*c)

plot(x,y,'\*'),hold

polyplot(a,b,c');

P=poly2sym(c)

end

x=[0,1,2,3,4,5,6];,y=[2,1,3,3,2,4,5];

c=lstsqline(x,y)

c1 =

1/2

19/14

c2 =

1/2

19/14

N =

6049/3080

Current plot held

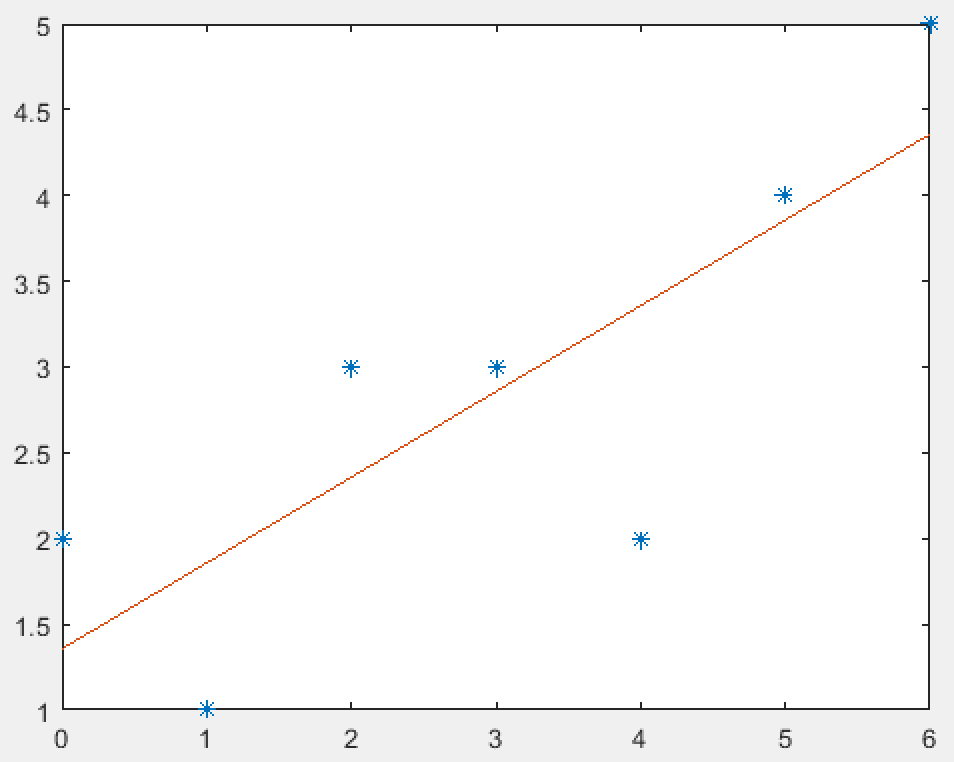
P =

x/2 + 19/14

c =

1/2

19/14



%Exercise 6

type lstsqpoly

function c = lstsqpoly(x, y, n)

format rat

x=x';

y=y';

a=x(1);

m=length(x);

b=x(m);

X=[];

for s=n:-1:1

X=[X x.^s];

end

X=[X ones(m,1)];

c=lscov(X,y);

c1=(inv(X'\*X))\*(X'\*y)

c2=(X'\*X)\(X'\*y)

N=norm(y-X\*c)

plot(x,y,'\*'),hold

polyplot(a,b,c');

P=poly2sym(c)

end

x=[0,1,2,3,4,5,6];, y=[2,1,3,3,2,4,5];

c=lstsqpoly(x,y,1)

c1 =

1/2

19/14

c2 =

1/2

19/14

N =

6049/3080

Current plot held

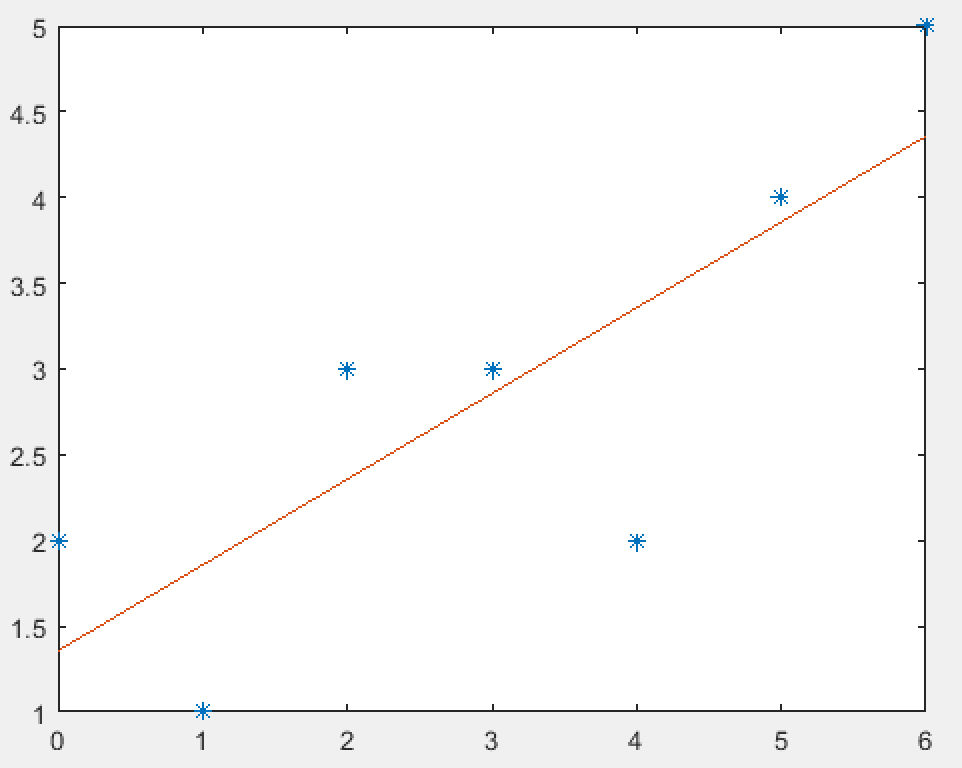
P =

x/2 + 19/14

c =

1/2

19/14



% c, c1, and c2 are equivalent with the vector being

% [1/2; 19/14]

% The plots for both degree 1 and the original are equivalent because both

% functions create the same matrix X which is used to create the same plot.

c=lstsqpoly(x,y,2)

c1 =

2/21

-1/14

11/6

c2 =

2/21

-1/14

11/6

N =

943/536

Current plot held

P =

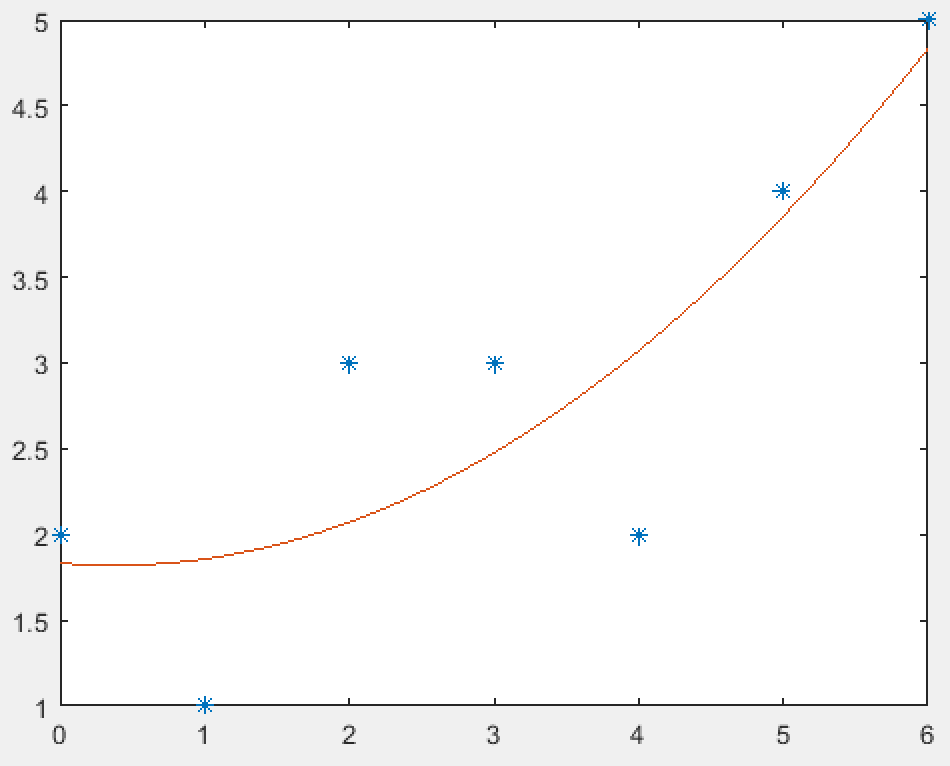
(2\*x^2)/21 - x/14 + 11/6

c =

2/21

-1/14

11/6



% c, c1, and c2 are equivalent with the vector being

% [2/21; -1/14; 1/6]

c=lstsqpoly(x,y,3)

c1 =

1/36

-13/84

61/126

5/3

c2 =

1/36

-13/84

61/126

5/3

N =

2134/1247

Current plot held

P =

x^3/36 - (13\*x^2)/84 + (61\*x)/126 + 5/3

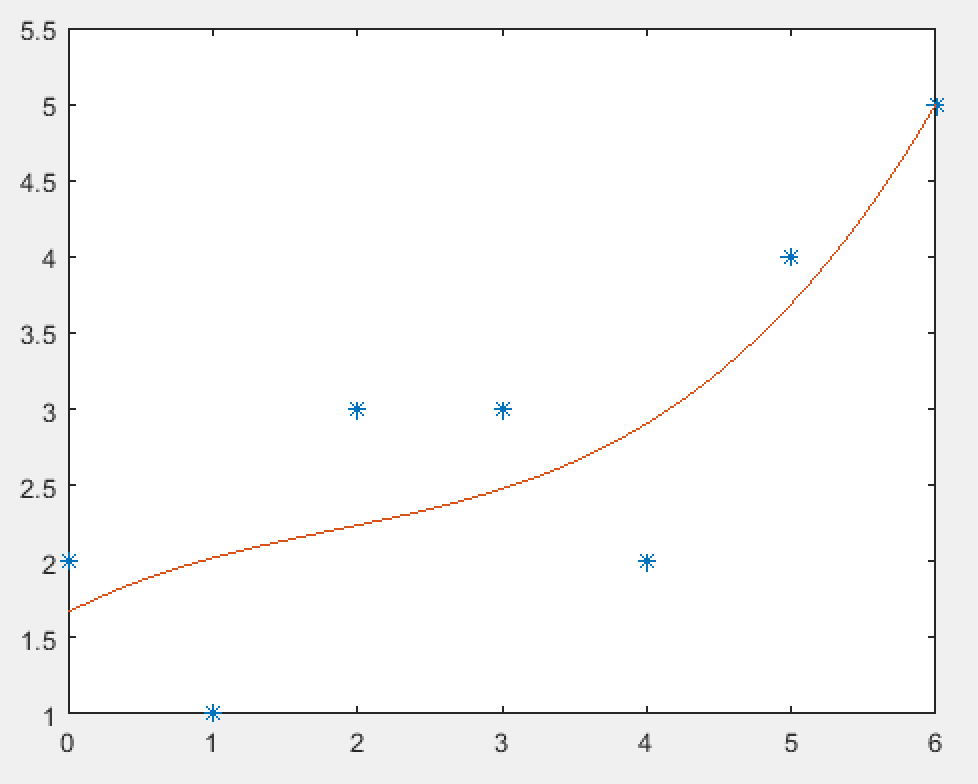
c =

1/36

-13/84

61/126

5/3



% c, c1, and c2 are equivalent with the vector being

% [1/36; -13/84; 61/126; 5/3]

c=lstsqpoly(x,y,4)

c1 =

3/88

-151/396

359/264

-491/396

851/462

c2 =

3/88

-151/396

359/264

-491/396

851/462

N =

2463/1589

Current plot held

P =

(3\*x^4)/88 - (151\*x^3)/396 + (359\*x^2)/264 - (491\*x)/396 + 851/462

c =

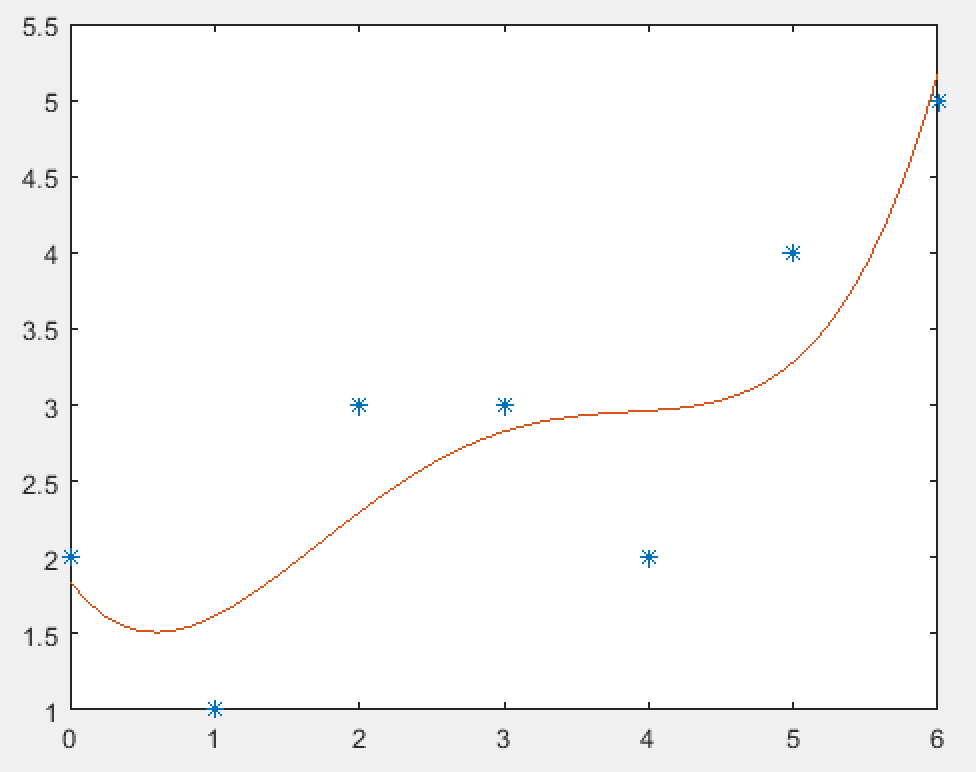
3/88

-151/396

359/264

-491/396

851/462



% c, c1, and c2 all have the same vector of

% [3/88; -151/396; 359/264; -491/396; 851/462]

% The polynomial of degree 4 fits the data the best because it

% forms a residual line that is the closest to the true residual

% plot for the data. Additionally, the plot moves between data

% points creating a smaller distance between the data points and

% the plot.

%Exercise 7

type invalprob

function [] = invalprob(A,x0)

L = sort(transpose(closetozeroroundoff(eig(A))));

M = unique(L);

W = eye(size(A));

P = [];

for i=1:size(M,2)

count = 0;

for j=1:size(L,2)

if(L(j) == M(i))

count = count + 1;

end

end

W = null(A-(M(i)\*eye(size(A))),'r');

P = [P W];

end

if (count == size(W,2))

disp('There exists a unique solution to the initial value problem')

L

P

fprintf('P forms an eigenvector basis for R^%d\n',size(P,2))

syms t;

E = sym(zeros(size(P)));

for i=1:size(P,2)

E(:,i) = P(:,i)\*exp(L(i)\*t);

end

E

fprintf('The columns of E are the eigenfunctions for A.\n')

C = rref([P x0]);

C = C(:,size(C,2))

attrDir = L(~(abs(L) - min(abs(L))));

replDir = L(~(abs(L) - max(abs(L))));

if(size(C,1) == 2)

if(all(abs(L(:)) < 1))

disp('Origin is an attractor')

fprintf('The direction of greatest attraction corresponds to vector ')

P(:,~(L-attrDir))

elseif(all(abs(L(:)) > 1))

disp('Origin is a repellor')

fprintf('The direction of greatest repulsion corresponds to vector ')

P(:,~(L-replDir))

else

disp('Origin is a saddle point')

fprintf('The direction of greatest attraction corresponds to vector ')

P(:,~(L-attrDir))

fprintf('The direction of greatest repulsion corresponds to vector ')

P(:,~(L-replDir))

end

end

else

disp('There is no unique solution to the initial value problem')

return;

end

end

%(A)

A=[-2 -5;1 4], x0=[2;3]

A =

-2 -5

1 4

x0 =

2

3

invalprob(A,x0)

There exists a unique solution to the initial value problem

L =

-1 3

P =

-5 -1

1 1

P forms an eigenvector basis for R^2

E =

[ -5\*exp(-t), -exp(3\*t)]

[ exp(-t), exp(3\*t)]

The columns of E are the eigenfunctions for A.

C =

-1.2500

4.2500

Origin is a saddle point

The direction of greatest attraction corresponds to vector ans =

-5

1

The direction of greatest repulsion corresponds to vector ans =

-1

1

%(B)

A=[7 -1;3 3], x0=[2;3]

A =

7 -1

3 3

x0 =

2

3

invalprob(A,x0)

There exists a unique solution to the initial value problem

L =

4 6

P =

0.3333 1.0000

1.0000 1.0000

P forms an eigenvector basis for R^2

E =

[ exp(4\*t)/3, exp(6\*t)]

[ exp(4\*t), exp(6\*t)]

The columns of E are the eigenfunctions for A.

C =

1.5000

1.5000

Origin is a repellor

The direction of greatest repulsion corresponds to vector ans =

1

1

%(C)

A=[-1.5 .5; 1 -1], x0=[5;4]

A =

-1.5000 0.5000

1.0000 -1.0000

x0 =

5

4

invalprob(A,x0)

There exists a unique solution to the initial value problem

L =

-2.0000 -0.5000

P =

-1.0000 0.5000

1.0000 1.0000

P forms an eigenvector basis for R^2

E =

[ -exp(-2\*t), exp(-t/2)/2]

[ exp(-2\*t), exp(-t/2)]

The columns of E are the eigenfunctions for A.

C =

-2

6

Origin is a saddle point

The direction of greatest attraction corresponds to vector ans =

0.5000

1.0000

The direction of greatest repulsion corresponds to vector ans =

-1

1

%(D)

A=[3 3;0 3], x0=[2;3]

A =

3 3

0 3

x0 =

2

3

invalprob(A,x0)

There is no unique solution to the initial value problem

%(E)

A=diag([1,2,2,3,3,3]), x0=ones(6,1)

A =

1 0 0 0 0 0

0 2 0 0 0 0

0 0 2 0 0 0

0 0 0 3 0 0

0 0 0 0 3 0

0 0 0 0 0 3

x0 =

1

1

1

1

1

1

invalprob(A,x0)

There exists a unique solution to the initial value problem

L =

1 2 2 3 3 3

P =

1 0 0 0 0 0

0 1 0 0 0 0

0 0 1 0 0 0

0 0 0 1 0 0

0 0 0 0 1 0

0 0 0 0 0 1

P forms an eigenvector basis for R^6

E =

[ exp(t), 0, 0, 0, 0, 0]

[ 0, exp(2\*t), 0, 0, 0, 0]

[ 0, 0, exp(2\*t), 0, 0, 0]

[ 0, 0, 0, exp(3\*t), 0, 0]

[ 0, 0, 0, 0, exp(3\*t), 0]

[ 0, 0, 0, 0, 0, exp(3\*t)]

The columns of E are the eigenfunctions for A.

C =

1

1

1

1

1

1