MATLAB PROJECT 2

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 # 27

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

\_1.\_Shanil Patel\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_2.\_Sean Tromans\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

\_3. Harrison Paparatto\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_4.\_Adam Wood\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_5.\_Joseph Samman\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_6.\_Mason Taylor\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

diary on

% #Exercise1

format compact

type eigen

function [ ] = eigen( A )

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

M = unique(L)

disp("The sum of all multiplicities is Q=")

Q = size(L,2)

N=0;

P=[];

for i=1:size(M,2)

W = null((A-(M(1,i)\*eye(size(A)))),'r')

N = N + size(W,2);

P = [P W];

end

disp("The total sum of the dimensions of the eigenspaces is N=")

N

if N == Q

disp("Yes, matrix A is diagonalizable since N=Q")

P

D = diag(L)

if closetozeroroundoff(A\*P-P\*D)==zeros(Q)

disp("Great! I got a diagonalization!")

else

disp("Oops! I got a big in my code.")

end

else

disp("No, matrix A is not diagonalizable since N<Q")

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;

type jord

function J = jord (n, r)

A = ones(n);

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

for i = 1: n

J(i, i) = r;

end

eigen([2 2;0 2])

L =

2 2

M =

2

The sum of all multiplicities is Q=

Q =

2

W =

1

0

The total sum of the dimensions of the eigenspaces is N=

N =

1

No, matrix A is not diagonalizable since N<Q

eigen([4 0 0 0;1 3 0 0;0 -1 3 0;0 -1 5 4])

L =

3 3 4 4

M =

3 4

The sum of all multiplicities is Q=

Q =

4

W =

0

0

-0.2000

1.0000

W =

0

0

0

1

The total sum of the dimensions of the eigenspaces is N=

N =

2

No, matrix A is not diagonalizable since N<Q

eigen(jord(5,3))

L =

3 3 3 3 3

M =

3

The sum of all multiplicities is Q=

Q =

5

W =

1

0

0

0

0

The total sum of the dimensions of the eigenspaces is N=

N =

1

No, matrix A is not diagonalizable since N<Q

eigen(diag([3,3,3,2,2,1]))

L =

1 2 2 3 3 3

M =

1 2 3

The sum of all multiplicities is Q=

Q =

6

W =

0

0

0

0

0

1

W =

0 0

0 0

0 0

1 0

0 1

0 0

W =

1 0 0

0 1 0

0 0 1

0 0 0

0 0 0

0 0 0

The total sum of the dimensions of the eigenspaces is N=

N =

6

Yes, matrix A is diagonalizable since N=Q

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

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

Great! I got a diagonalization!

eigen(magic(4))

L =

-8.9443 0 8.9443 34.0000

M =

-8.9443 0 8.9443 34.0000

The sum of all multiplicities is Q=

Q =

4

W =

-0.4570

-0.0287

-0.5143

1.0000

W =

-1

-3

3

1

W =

-2.1882

1.1254

0.0627

1.0000

W =

1.0000

1.0000

1.0000

1.0000

The total sum of the dimensions of the eigenspaces is N=

N =

4

Yes, matrix A is diagonalizable since N=Q

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

D =

-8.9443 0 0 0

0 0 0 0

0 0 8.9443 0

0 0 0 34.0000

Great! I got a diagonalization!

eigen(ones(5))

L =

0 0 0 0 5

M =

0 5

The sum of all multiplicities is Q=

Q =

5

W =

-1 -1 -1 -1

1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1

W =

1

1

1

1

1

The total sum of the dimensions of the eigenspaces is N=

N =

5

Yes, matrix A is diagonalizable since N=Q

P =

-1 -1 -1 -1 1

1 0 0 0 1

0 1 0 0 1

0 0 1 0 1

0 0 0 1 1

D =

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 0

0 0 0 0 5

Great! I got a diagonalization!

eigen(magic(5))

L =

-21.2768 -13.1263 13.1263 21.2768 65.0000

M =

-21.2768 -13.1263 13.1263 21.2768 65.0000

The sum of all multiplicities is Q=

Q =

5

W =

-0.1440

-0.5200

-0.8114

0.4753

1.0000

W =

-2.4172

2.2511

-1.4952

0.6613

1.0000

W =

-0.4137

-0.2736

0.6186

-0.9313

1.0000

W =

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

W =

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

The total sum of the dimensions of the eigenspaces is N=

N =

3

No, matrix A is not diagonalizable since N<Q

% the result for g cannot be correct because it returns empty vectors as basises for the eigenspaces for two of the eigenvalues.

type eigen\_1

function [ ] = eigen( A )

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

M = unique(L)

disp("The sum of all multiplicities is Q=")

Q = size(L,2)

N=0;

P=[];

for i=1:size(M,2)

W = null((A-(M(1,i)\*eye(size(A)))))

N = N + size(W,2);

P = [P W];

end

disp("The total sum of the dimensions of the eigenspaces is N=")

N

if N == Q

disp("Yes, matrix A is diagonalizable since N=Q")

P

D = diag(L)

if closetozeroroundoff(A\*P-P\*D)==zeros(Q)

disp("Great! I got a diagonalization!")

else

disp("Oops! I got a big in my code.")

end

else

disp("No, matrix A is not diagonalizable since N<Q")

end

% The only change made to fix the "problem" was changing null( ,'r') to null( )

eigen\_1(magic(5))

L =

-21.2768 -13.1263 13.1263 21.2768 65.0000

M =

-21.2768 -13.1263 13.1263 21.2768 65.0000

The sum of all multiplicities is Q=

Q =

5

W =

0.0976

0.3525

0.5501

-0.3223

-0.6780

W =

-0.6330

0.5895

-0.3915

0.1732

0.2619

W =

-0.2619

-0.1732

0.3915

-0.5895

0.6330

W =

0.6780

0.3223

-0.5501

-0.3525

-0.0976

W =

-0.4472

-0.4472

-0.4472

-0.4472

-0.4472

The total sum of the dimensions of the eigenspaces is N=

N =

5

Yes, matrix A is diagonalizable since N=Q

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

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

Great! I got a diagonalization!

diary off

diary on

format compact

%Exercise 2

type diagonal

function L = diagonal( A )

n = size( A , 1 );

[ P, D] = eig( A );

k = rank( D );

d = rank(P);

if d ~= length(P)

k=d;

k

disp( 'A is not diagonalizable ' )

disp( 'A does not have enough linearly independent eigenvectors to create a basis for R^n' )

return;

end

disp( 'The number of linearly independent columns in P is' )

k

if ( k==n )

disp( 'A is diagonalizable' )

disp( 'A basis for R^n is' )

P

else

disp( 'A is not diagonalizable ' )

disp( 'A does not have enough linearly independent eigenvectors to create a basis for R^n' )

end

end

A=[2,2;0,2]

A =

2 2

0 2

diagonal(A)

k =

1

A is not diagonalizable

A does not have enough linearly independent eigenvectors to create a basis for R^n

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

diagonal(A)

k =

2

A is not diagonalizable

A does not have enough linearly independent eigenvectors to create a basis for R^n

A=jord(5,3)

A =

3 1 0 0 0

0 3 1 0 0

0 0 3 1 0

0 0 0 3 1

0 0 0 0 3

diagonal(A)

k =

1

A is not diagonalizable

A does not have enough linearly independent eigenvectors to create a basis for R^n

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

diagonal(A)

The number of linearly independent columns in P is

k =

6

A is diagonalizable

A basis for R^n 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

A=magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

diagonal(A)

The number of linearly independent columns in P is

k =

3

A is not diagonalizable

A does not have enough linearly independent eigenvectors to create a basis for R^n

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

diagonal(A)

The number of linearly independent columns in P is

k =

1

A is not diagonalizable

A does not have enough linearly independent eigenvectors to create a basis for R^n

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

diagonal(A)

The number of linearly independent columns in P is

k =

5

A is diagonalizable

A basis for R^n is

P =

-0.4472 0.0976 -0.6330 0.6780 -0.2619

-0.4472 0.3525 0.5895 0.3223 -0.1732

-0.4472 0.5501 -0.3915 -0.5501 0.3915

-0.4472 -0.3223 0.1732 -0.3525 -0.5895

-0.4472 -0.6780 0.2619 -0.0976 0.6330

diary off

%Exercise 3

diary on

format compact

%Exercise 3

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);

if size(A,1) == size(b,2)

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

p = b;

disp(p);

z = b - p;

disp(z);

disp('b is in the col A');

quit;

else

r = colspace(sym(A));

t = 0;

for i = 1:size(r,2)

t = t + dot(r(:,i),b);

end

t = closetozeroroundoff(t);

if t == 0

z = b;

p = z - b;

disp(p);

disp(z);

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

quit;

else

p = (A\*((transpose(A)\*A)\transpose(A)))\*b;

disp(p);

z = b - p';

D = dot(p,z);

if abs(D) < 1E-7

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

else

disp('Oops! Is there a bug in my code?');

end

end

end

else

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

p = [];

z = [];

return

end

end

%(a)

A = magic(6);

A = A(:,1:4), b = (1:6)

A =

35 1 6 26

3 32 7 21

31 9 2 22

8 28 33 17

30 5 34 12

4 36 29 13

b =

1 2 3 4 5 6

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

0.9492

2.1599

2.9492

3.9180

5.1287

5.9180

Yes, p and z are orthogonal! Great job!

p =

0.9492

2.1599

2.9492

3.9180

5.1287

5.9180

z =

0.0508 -0.1599 0.0508 0.0820 -0.1287 0.0820

%(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)

-0.2500

-0.0000

0.2500

0.2500

-0.0000

0.7500

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(4), b = (1:5)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

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 =

0.8147 0.9058 0.1270 0.9134 0.6324

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

0.8147 0.9058 0.1270 0.9134 0.6324

0 0 0 0 0

b is in the col A

%(e)

A = ones(6); A(:) = 1:36, b = [1,0,1,0,1,0]

A =

1 7 13 19 25 31

2 8 14 20 26 32

3 9 15 21 27 33

4 10 16 22 28 34

5 11 17 23 29 35

6 12 18 24 30 36

b =

1 0 1 0 1 0

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

0.7143

0.6286

0.5429

0.4571

0.3714

0.2857

Yes, p and z are orthogonal! Great job!

p =

0.7143

0.6286

0.5429

0.4571

0.3714

0.2857

z =

0.2857 -0.6286 0.4571 -0.4571 0.6286 -0.2857

%(f)

A = ones(6); A(:) = 1:36; A = null(A,'r'), b = ones(1,6)

A =

1 2 3 4

-2 -3 -4 -5

1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1

b =

1 1 1 1 1 1

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

0 0 0 0 0 0

1 1 1 1 1 1

b is orthogonal to Col A

diary off

%Exercise4

diary on

format compact

type shrink

function B = shrink(A)

format compact,

[~, pivot] = rref(A);

B = A(: , pivot);

end

type solvemore

function X = solvemore(A,b)

format long,

A=shrink(A);

[m, n] = size(A);

B=closetozeroroundoff(A'\*A-eye(n));

Z = zeros(n);

%Determine rank to check for consistency

rank1 = rank(A);

rank2 = rank([A,b]);

%consistent: b is in colA

if(rank1 == rank2)

disp("The equation is consistent – look for the exact

solution");

%check(1) not orthonormal

if(isequal(B,Z) == 0)

R = rref([A,b]);

x1 = R(:,n+1);

disp("A does not have orthonormal columns");

X=x1;

%check(2) orthonormal but not orthogonal

elseif(isequal(B,Z) && (m ~= n))

R = rref([A,b]);

x1 = R(:,n+1);

disp("A has orthonormal columns but is not orthogonal");

X=x1;

%check(3) orthogonal

else

R = rref([A,b]);

x1 = R(:,n+1);

x2 = A'\*b;

N = norm(x1 -x2);

disp("A is orthogonal");

X = [x1,x2];

disp("The norm of the difference between the two

solutions is N=")

N

end

%inconsistent

else

disp("The system is inconsistent - look for the least

squares solution");

%part(1)

R = rref([A'\*A,A'\*b]);

x3 = R(:,n+1);

n1 = norm(b-A\*x3);

disp("The solution of the normal equation is x3=");

x3

disp("The least squares error of the approximation is n1=");

n1

%part2

%find an orthonormal basis for Col A

if(isequal(B,Z))

disp("A has orthonormal columns: an orthonormal basis

for Col A is U=A");

U = A;

else

disp("An orthonormal basis for Col A is U=");

U = orth(A);

U

end

%Find projection of b

b1 = U\*U'\*b;

disp("The projection of b onto Col A is");

b1

%find solution of Ax = b1 and its error

R = rref([A,b1])

x4 = R(:,n+1);

disp("The least-squares solution by using the projection

onto Col A is x4 = ");

x4

n2 = norm(b-b1);

disp("The least-squares error of this approximation is n2 =

");

n2

%find error between x3 and x4

n3 = norm(x3-x4);

disp("The norm of the difference between x3 and x4 is n3 =

");

n3

%prove x3 and x4 are closest

x=rand(n,1);

n4 = norm(b-A\*x);

disp("An error of approximation of b by Ax for a random

vector x in R^n is");

n4

X = [x3, x4];

end

end

%(a)

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

b =

16

5

9

4

A =

-0.500000000000000 0.670820393249937 0.500000000000000

-0.500000000000000 -0.223606797749979 -0.500000000000000

-0.500000000000000 0.223606797749979 -0.500000000000000

-0.500000000000000 -0.670820393249937 0.500000000000000

X = solvemore(A,b)

The equation is consistent – look for the exact solution

A has orthonormal columns but is not orthogonal

X =

-16.999999999999989

8.944271909999154

2.999999999999998

0

%(b)

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

A =

-0.447213595499958 -0.545634873129948 0.511667273601714

0.195439507584854 -0.449758363151198

-0.447213595499958 -0.449758363151205 -0.195439507584838 -

0.511667273601691 0.545634873129969

-0.447213595499958 -0.000000000000024 -0.632455532033676

0.632455532033676 -0.000000000000002

-0.447213595499958 0.449758363151189 -0.195439507584872 -

0.511667273601694 -0.545634873129966

-0.447213595499958 0.545634873129987 0.511667273601672

0.195439507584856 0.449758363151196

b =

0.964888535199277

0.157613081677548

0.970592781760616

0.957166948242946

0.485375648722841

X = solvemore(A,b)

The equation is consistent – look for the exact solution

A is orthogonal

The norm of the difference between the two solutions is N=

N =

9.209666375651935e-16

X =

-1.581184933186388 -1.581184933186389

0.097967085300906 0.097967085300906

-0.089676113299936 -0.089676113299936

0.326899233575111 0.326899233575112

-0.651929403475534 -0.651929403475534

%(c)

A= magic(4), b = ones(4,1)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

b =

1

1

1

1

X = solvemore(A,b)

The equation is consistent – look for the exact solution

A does not have orthonormal columns

X =

0.058823529411765

0.117647058823529

-0.058823529411765

0

%(d)

A = magic(4), b = rand(4, 1)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

b =

0.795199901137063

0.186872604554379

0.489764395788231

0.445586200710899

X = solvemore(A,b)

The system is inconsistent - look for the least squares solution

The solution of the normal equation is x3=

x3 =

0.045094059718030

-0.067745817096681

0.079046554501903

The least squares error of the approximation is n1=

n1 =

0.125009990505856

An orthonormal basis for Col A is U=

U =

-0.363225569906992 -0.839773278980323 0.335928601456289

-0.511952614823082 0.201051551215513 -0.497476425501395

-0.413098103234259 -0.228706948321817 -0.571876812690966

-0.659789104673461 0.449502219631666 0.559129763025005

The projection of b onto Col A is

b1 =

0.823152984800833

0.270731855545687

0.405905144796922

0.417633117047130

The least-squares solution by using the projection onto Col A is

x4 =

x4 =

0.045094059718030

-0.067745817096682

0.079046554501904

0

The least-squares error of this approximation is n2 =

n2 =

0.125009990505856

The norm of the difference between x3 and x4 is n3 =

n3 =

5.792210254404242e-16

An error of approximation of b by Ax for a random vector x in

R^n is

n4 =

35.779216930112220

X =

0.045094059718030 0.045094059718030

-0.067745817096681 -0.067745817096682

0.079046554501903 0.079046554501904

%(e)

A= magic(4); A = orth(A), b = rand(4,1)

A =

-0.500000000000000 0.670820393249937 0.500000000000000

-0.500000000000000 -0.223606797749979 -0.500000000000000

-0.500000000000000 0.223606797749979 -0.500000000000000

-0.500000000000000 -0.670820393249937 0.500000000000000

b =

0.276025076998578

0.679702676853675

0.655098003973841

0.162611735194631

X = solvemore(A,b)

The system is inconsistent - look for the least squares solution

The solution of the normal equation is x3=

x3 =

-0.886718746510362

0.070578210436368

-0.448081934317153

The least squares error of the approximation is n1=

n1 =

0.041865310519941

A has orthonormal columns: an orthonormal basis for Col A is U=A

The projection of b onto Col A is

b1 =

0.266663708976407

0.651618572787158

0.683182108040358

0.171973103216803

The least-squares solution by using the projection onto Col A is

x4 =

x4 =

-0.886718746510363

0.070578210436368

-0.448081934317153

0

The least-squares error of this approximation is n2 =

n2 =

0.041865310519941

The norm of the difference between x3 and x4 is n3 =

n3 =

6.901322157724297e-16

An error of approximation of b by Ax for a random vector x in

R^n is

n4 =

1.782748638667547

X =

-0.886718746510362 -0.886718746510363

0.070578210436368 0.070578210436368

-0.448081934317153 -0.448081934317153

diary off

%Exercise 5

diary on

format compact

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=[0,2,3,5,6]

x =

Columns 1 through 3

0 2 3

Columns 4 through 5

5 6

y=[4,3,2,1,0]

y =

Columns 1 through 3

4 3 2

Columns 4 through 5

1 0

c=lstsqline(x,y)

c1 =

-25/38

78/19

c2 =

-25/38

78/19

N =

514/1417

Current plot held

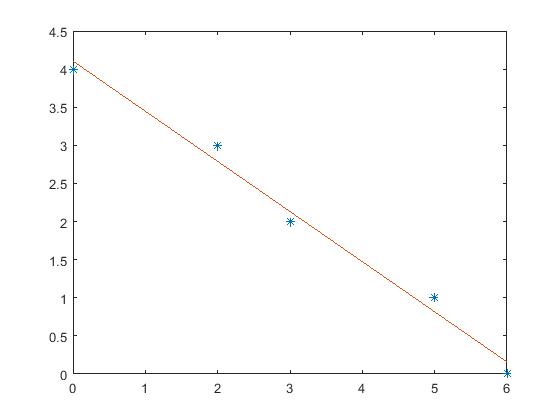
P =

78/19 - (25\*x)/38

c =

-25/38

78/19



diary off

%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);

for i=0:n

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

end

c = lscov(X, y);

%the same result will be obtained if c = (X^T \* X)^-1 \* (X^T \*y) or c =

%(X^T \* X)\(X^T\*y)

%and we are verifying it:

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

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

%the next command calculates the 2-norm of the residual vector

N=norm(y-X\*c)

%plot data points and the least-squares regression line:

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

polyplot(a,b,c');

%output the polynomial:

P=poly2sym(c)

end

x=[0,2,3,5,6];

y=[4,3,2,1,0];

c=lstsqpoly(x,y,1)

c1 =

-25/38

78/19

c2 =

-25/38

78/19

N =

514/1417

Current plot held

P =

78/19 - (25\*x)/38

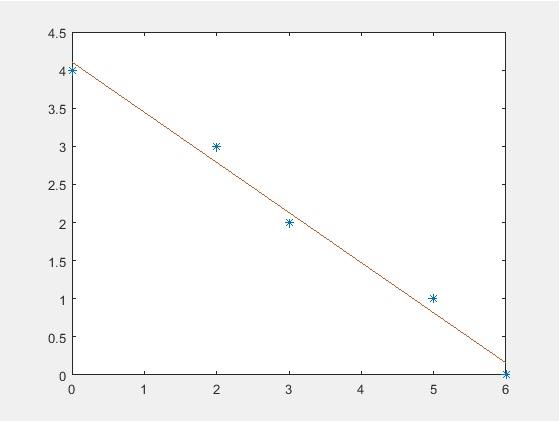
c =

-25/38

78/19

%c, c1, and c2 are equal

%Both functions must be consistent because both show the same output for n=1.



c=lstsqpoly(x,y,2)

c1 =

-155/1538

2685/769

c2 =

-155/1538

2685/769

N =

522/659

Current plot held

P =

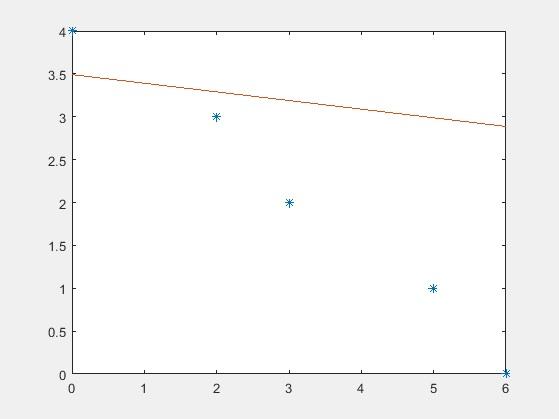
2685/769 - (155\*x)/1538

c =

-155/1538

2685/769

%c, c1, and c2 are equal.



c=lstsqpoly(x,y,3)

c1 =

-197/12487

1966/617

c2 =

-197/12487

1966/617

N =

501/433

Current plot held

P =

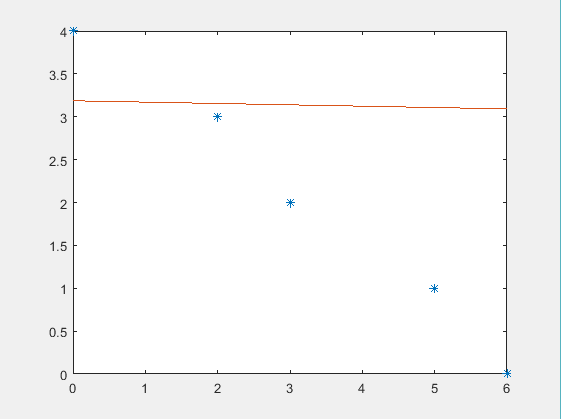
1966/617 - (915\*x)/57998

c =

-197/12487

1966/617

%c, c1, and c2 are equal.



c=lstsqpoly(x,y,4)

c1 =

-16/6311

2733/904

c2 =

-16/6311

2733/904

N =

394/287

Current plot held

P =

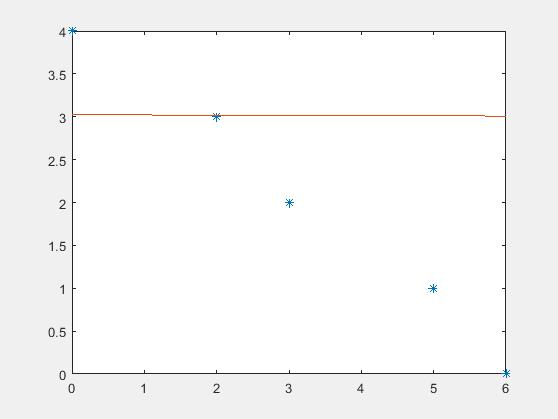
6807709392713421/2251799813685248 - (1461476957204631\*x)/576460752303423488

c =

-16/6311

2733/904

%c, c1, and c2 are equal.



%The degree 1 polynomial fits the data the best.

diary off