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 # 18

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

1. Leeson Chen
2. Antoine Gordon Alvarez
3. Ari Heyd
4. Adam Hochberger
5. Madison Holt

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

Project 4

Group 18

% Exercise1

diary Project4

diary on

format compact

Type eigen

function [] = eigen(A)

L = round(transpose(eig(A)));

L = closetozeroroundoff(L)

M = unique(L)

[~,Q] = size(L);

[~,U] = size(M);

fprintf('%s\n', 'The sum of multiplicities of the eigenvalues is ')

disp(Q)

V = size(1,1);

B = A;

[~,y] = size(A);

N = 0;

P = zeros(y,1);

for i = 1:U

for j = 1:y

B(j,j) = B(j,j) - M(1,i);

end

W = null(B,'r');

[~,t] = size(W);

for j = 1:t

N = N + 1;

V(1,N) = [M(1,i)];

end

P = [P W];

end

if Q == N

fprintf('Yes, matrix A is diagonalizable since N=Q')

P(:,1) = []

D = diag(V)

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

if F == 0

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

else

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

end

else

fprintf('No, matrix A is not diagonalizable since N<Q.')

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;

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 = [2 2; 0 2]

A =

2 2

0 2

eigen(A)

L =

2 2

M =

2

The sum of multiplicities of the eigenvalues is

2

No, matrix A is not diagonalizable since N<Q.

% (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 =

4 3 3 4

M =

3 4

The sum of multiplicities of the eigenvalues is

4

No, matrix A is not diagonalizable since N<Q.

% (c)

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

eigen(A)

L =

3 3 3 3 3

M =

3

The sum of multiplicities of the eigenvalues is

5

ans =

'No, matrix A is not diagonalizable since N<Q.'

% (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

The sum of multiplicities of the eigenvalues is

6

ans =

'No, matrix A is not diagonalizable since N<Q.'

% (e)

A = magic(4)

A =

16 2 3 13

5 11 10 8

9 7 6 12

4 14 15 1

eigen(A)

L =

34 9 -9 0

M =

-9 0 9 34

The sum of multiplicities of the eigenvalues is

4

ans =

'No, matrix A is not diagonalizable since N<Q.'

% (f)

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

eigen(A)

L =

0 0 0 0 5

M =

0 5

The sum of multiplicities of the eigenvalues is

5

Yes, matrix A is diagonalizable since N=QP =

-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!

% (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 =

65 -21 -13 21 13

M =

-21 -13 13 21 65

The sum of multiplicities of the eigenvalues is

5

ans =

'No, matrix A is not diagonalizable since N<Q.'

% The outputs for g are correct because there are not enough rational diagonal eigenvalues.

type eigen\_1

function [] = eigen\_1(A)

L = round(transpose(eig(A)));

L = closetozeroroundoff(L)

M = unique(L)

[~,Q] = size(L);

[~,U] = size(M);

fprintf('%s\n', 'The sum of multiplicities of the eigenvalues is ')

disp(Q)

V = size(1,1);

B = A;

[~,y] = size(A);

N = 0;

P = zeros(y,1);

for i = 1:U

for j = 1:y

B(j,j) = B(j,j) - M(1,i);

end

W = null(B);

[~,t] = size(W);

for j = 1:t

N = N + 1;

V(1,N) = [M(1,i)];

end

P = [P W];

end

if Q == N

fprintf('Yes, matrix A is diagonalizable since N=Q')

P(:,1) = []

D = diag(V)

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

if F == 0

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

else

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

end

else

sprintf('No, matrix A is not diagonalizable since N<Q.')

end

end

% I changed null(A, 'r') to null(A)

eigen\_1(A)

L =

65 -21 -13 21 13

M =

-21 -13 13 21 65

The sum of multiplicities of the eigenvalues is

5

ans =

'No, matrix A is not diagonalizable since N<Q.'

diary off

% Exercise2

diary Project4

diary on

format compact

type diagonal

function [ L ] = diagonal( A )

n = size(A,1)

[P, D] = eig(A);

%[~,colind] = rref(P);

%PP = A(:, colind);

%[j,k] = size(PP);

k = rank(P);

fprintf('The number of linearly independent columns in P is k = ');

disp(k)

%d = det(P);

if k ~= n

% P not invertible, A not diagonalizable

fprintf('A is not diagonalizable\n');

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

else

% d =/= 0

% P is invertible, A is diagonalizable

fprintf('A is diagonalizable\n');

fprintf('A basis for R^n is\n');

disp(P)

end

L=transpose(diag(D));

End

% (a)

A = [2 2 ; 0 2]

A =

2 2

0 2

diagonal( A )

n =

2

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

ans =

2 2

% (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

diagonal( A )

n =

4

The number of linearly independent columns in P is k = 2

A is not diagonalizable

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

ans =

4 3 3 4

% (c)

A = jord(5, 3);

diagonal(A)

n =

5

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

ans =

3 3 3 3 3

% (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

diagonal(A)

n =

6

The number of linearly independent columns in P is k = 6

A is diagonalizable

A basis for R^n is

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

ans =

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

diagonal(A)

n =

4

The number of linearly independent columns in P is k = 4

A is diagonalizable

A basis for R^n is

-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

ans =

34.0000 8.9443 -8.9443 0.0000

% (f)

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)

n =

5

The number of linearly independent columns in P is k = 5

A is diagonalizable

A basis for R^n is

0.8333 -0.1667 -0.1667 0.2236 0.4472

-0.1667 0.8333 -0.1667 0.2236 0.4472

-0.1667 -0.1667 0.8333 0.2236 0.4472

-0.5000 -0.5000 -0.5000 0.2236 0.4472

0 0 0 -0.8944 0.4472

ans =

0 0 0 0 5

% (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

diagonal(A)

n =

5

The number of linearly independent columns in P is k = 5

A is diagonalizable

A basis for R^n is

-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

ans =

65.0000 -21.2768 -13.1263 21.2768 13.1263

diary off

% Exercise 3

diary Project4

diary on

format compact

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, n] = size(A);

r = size(b,1);

if m == r

%find if b is in A 1 = true 0 = false

isIn = ismember(b',A','rows');

if isIn == 1

disp('b is in the Col A')

p = b;

z = zeros(r,1);

else

x = 0;

for y = 1:n

temp = A(:,y);

x = x+dot(temp,b);

end

if x == 0

disp('b is orthogonal to Col A')

p = zeros(r,1);

z = b;

else

P = A\*inv(A'\*A)\*A';

p = P\*b;

z = b - p;

if dot(p,z) < 10^-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 = [ ];

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)

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)

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)

Yes, p and z are orthogonal! Great Job!

p =

0.8147

0.9058

0.1270

0.9134

0.6324

z =

1.0e-15 \*

0.1110

-0.2220

-0.0833

-0.4441

-0.8882

%%(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)

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)

b is orthogonal to Col A

p =

0

0

0

0

0

0

z =

1

1

1

1

1

1

% Exercise4

diary Project4

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);

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

disp('The equations is consistent')

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

if( B ~= zeros(n))

disp('does not have orthonomrmal columns')

X = A\b

else if ( m ~= n)

disp('has orthonormal columns but is not orthogonal')

X = A\b

else

disp('is orthogonal')

x1 = A\b

x2 = A'\*b

X = [x1,x2]

N = norm(x1 - x2)

end

end

else

disp('The syststem is inconsistent - look for the least-squares solution')

x3 = (A.'\*A)\ (A.'\*b);

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

disp('The solution of the normal equations is ', x3)

disp('The least-square error of the approximation is n1 = ', n1)

if( B ~= zeros(n))

disp('A has orthonormal columns: an orthonormal basis for Col A is U = A')

U = A

else

U = orth(A);

disp('An orthonormal basis for Col A is U = ', U)

end

b1 = U\*U.'\*b;

disp('The projection of b onto Col A is ', b1)

x4 = A\(b1);

disp('The least-square solution by using projection onto Col A is x4 = ',x4)

n2 = norm(b - A\*x4);

disp('The least squares error of this approximation is n2 = ', n2)

n3 = norm(x3 - x4);

disp('The norm n3 of the difference between the soultions x3 and x4 is ', n3)

x = rand(n,1);

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

disp('An error of approximatinon of b by Ax for a random vector x in R^n is ', n4)

X = [x3, x4]

end

end

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

b =

16

5

9

4

A =

-0.5000 0.6708 0.5000

-0.5000 -0.2236 -0.5000

-0.5000 0.2236 -0.5000

-0.5000 -0.6708 0.5000

solvemore(A,b)

The equations is consistent

has orthonormal columns but is not orthogonal

X =

-16.999999999999993

8.944271909999161

3.000000000000000

ans =

-16.999999999999993

8.944271909999161

3.000000000000000

%(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 =

0.814723686393179

0.905791937075619

0.126986816293506

0.913375856139019

0.632359246225410

solvemore(A,b)

The equations is consistent

is orthogonal

x1 =

-1.517501961599936

-0.096093467150121

0.304574206628231

-0.567677934732546

-0.086157982822825

x2 =

-1.517501961599936

-0.096093467150121

0.304574206628231

-0.567677934732546

-0.086157982822825

X =

-1.517501961599936 -1.517501961599936

-0.096093467150121 -0.096093467150121

0.304574206628231 0.304574206628231

-0.567677934732546 -0.567677934732546

-0.086157982822825 -0.086157982822825

N =

4.266150252213274e-16

ans =

-1.517501961599936 -1.517501961599936

-0.096093467150121 -0.096093467150121

0.304574206628231 0.304574206628231

-0.567677934732546 -0.567677934732546

-0.086157982822825 -0.086157982822825

%(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

solvemore(A,b)

The equations is consistent

does not have orthonomrmal columns

X =

0.058823529411765

0.117647058823529

-0.058823529411765

ans =

0.058823529411765

0.117647058823529

-0.058823529411765

%(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 =

0.097540404999410

0.278498218867048

0.546881519204984

0.957506835434298

solvemore(A,b)

The syststem is inconsistent - look for the least-squares solution

The solution of the normal equations is

x3 =

-0.001240611967882

-0.031004149639099

0.087551437445385

The least-square error of the approximation is n1 =

n1 =

0.372331330756435

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

U =

16 2 3

5 11 10

9 7 6

4 14 15

The projection of b onto Col A is

b1 =

1.0e+02 \*

2.904297176698675

4.911505464772355

3.731176504390088

6.445284057845472

The least-square solution by using projection onto Col A is x4 =

x4 =

11.705098588907832

20.491827548051422

20.721495050413061

The least squares error of this approximation is n2 =

n2 =

9.371449304906834e+02

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

n3 =

31.368529251477824

An error of approximatinon of b by Ax for a random vector x in R^n is

x =

0.964888535199277

0.157613081677548

0.970592781760616

n4 =

34.842978007481335

X =

-0.001240611967882 11.705098588907832

-0.031004149639099 20.491827548051422

0.087551437445385 20.721495050413061

ans =

-0.001240611967882 11.705098588907832

-0.031004149639099 20.491827548051422

0.087551437445385 20.721495050413061

%(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 =

0.957166948242946

0.485375648722841

0.800280468888800

0.141886338627215

solvemore(A,b)

The syststem is inconsistent - look for the least-squares solution

The solution of the normal equations is

x3 =

-1.192354702240901

0.617321717584815

-0.093301415370740

The least-square error of the approximation is n1 =

n1 =

0.028942288916205

An orthonormal basis for Col A is U =

The projection of b onto Col A is

b1 =

0.963638640787053

0.504790726355163

0.780865391256478

0.135414646083108

The least-square solution by using projection onto Col A is x4 =

x4 =

-1.192354702240901

0.617321717584816

-0.093301415370740

The least squares error of this approximation is n2 =

n2 =

0.028942288916205

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

n3 =

4.695944831791985e-16

An error of approximatinon of b by Ax for a random vector x in R^n is

x =

0.421761282626275

0.915735525189067

0.792207329559554

n4 =

1.865310860582851

X =

-1.192354702240901 -1.192354702240901

0.617321717584815 0.617321717584816

-0.093301415370740 -0.093301415370740

ans =

-1.192354702240901 -1.192354702240901

0.617321717584815 0.617321717584816

-0.093301415370740 -0.093301415370740

diary off

% Exercise5

diary Project4

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);

% 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, '\*')

polyplot(a, b, c');

P = poly2sym(c)

end

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

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

c = lstsqline(x, y)

c1 =

-25/38

78/19

c2 =

-25/38

78/19

N =

514/1417

P =

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

c =

-25/38

78/19

diary off

% Exercise 6

diary Project4

diary on

format compact

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 i = 1:(n+1)

X = [X x.^n]

n = n-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, '\*')

polyplot(a, b, c');

P = poly2sym(c)

end

x = [0. 2. 3. 5. 6]

x =

0 2 3 5 6

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

y =

Columns 1 through 3

4 3 2

Columns 4 through 5

1 0

a)

n = 1

n =

1

c = lstsqpoly(x, y, n)

X =

0

2

3

5

6

X =

0 1

2 1

3 1

5 1

6 1

c1 =

-25/38

78/19

c2 =

-25/38

78/19

N =

514/1417

P =

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

c =

-25/38

78/19

%The coefficients c, c1, and c2 match

%The values for c, c1, and c2 match exercise 5

b)

n = 2

n =

2

c = lstsqpoly(x, y, n)

X =

0

4

9

25

36

X =

0 0

4 2

9 3

25 5

36 6

X =

0 0 1

4 2 1

9 3 1

25 5 1

36 6 1

c1 =

-3/154

-83/154

309/77

c2 =

-3/154

-83/154

309/77

N =

703/2181

P =

309/77 - (3\*x^2)/154 - (83\*x)/154

c =

-3/154

-83/154

309/77

%The coefficients c, c1, and c2 match

c)

n = 3

n =

3

c = lstsqpoly(x, y, n)

X =

0

8

27

125

216

X =

0 0

8 4

27 9

125 25

216 36

X =

0 0 0

8 4 2

27 9 3

125 25 5

216 36 6

X =

Columns 1 through 3

0 0 0

8 4 2

27 9 3

125 25 5

216 36 6

Column 4

1

1

1

1

1

c1 =

-1/228

5/266

-983/1596

535/133

c2 =

-1/228

5/266

-983/1596

535/133

N =

454/1425

P =

- x^3/228 + (5417864213378195\*x^2)/288230376151711744 - (983\*x)/1596 + 535/133

c =

-1/228

5/266

-983/1596

535/133

%The coefficients c, c1, and c2 match

d)

n = 4

n =

4

c = lstsqpoly(x, y, n)

X =

0

16

81

625

1296

X =

0 0

16 8

81 27

625 125

1296 216

X =

0 0 0

16 8 4

81 27 9

625 125 25

1296 216 36

X =

Columns 1 through 3

0 0 0

16 8 4

81 27 9

625 125 25

1296 216 36

Column 4

0

2

3

5

6

X =

Columns 1 through 3

0 0 0

16 8 4

81 27 9

625 125 25

1296 216 36

Columns 4 through 5

0 1

2 1

3 1

5 1

6 1

c1 =

-1/40

19/60

-51/40

59/60

4

c2 =

-1/40

19/60

-51/40

59/60

4

N =

1/77464268540332

P =

- x^4/40 + (19\*x^3)/60 - (51\*x^2)/40 + (59\*x)/60 + 4

c =

-1/40

19/60

-51/40

59/60

4

%The coefficients c, c1, and c2 match

%Because the norm of the n = 4 polynomial plot is the lowest, this

%polynomial line fits the data the best

diary off