HW2

Problem 1:

A.

function [U, S, V] = mysvd(A)

n = size(A, 1);

p = size(A, 2);

[V, L] = jacobi(A' \* A);

L = diag(L);

%sort V, L simultaneously

for i=1:p

max = i; %index of current max eigenvalue

for j=i+1:p

if (L(j) > L(i))

max = j;

end

end

if max ~= i %swap columns

tmp\_L = L(i);

L(i) = L(max);

L(max) = tmp\_L;

tmp\_V = V(:,i);

V(:,i) = V(:,max);

V(:,max) = tmp\_V;

end

end

S = sqrt(diag(L));

if p > n

S = S(1:n, 1:p);

else

S = [S; zeros(p-n, p)];

end

U = A \* V \* mypinv(S);

%------------------------------

function Sminus = mypinv(S)

m = size(S, 1);

n = size(S, 2);

Sminus = zeros(n, m);

p = min(m, n);

for i = 1:p

if S(i,i) ~= 0

Sminus(i,i) = 1/S(i,i);

end

end

%Testing: ------------------------------

C = gallery('chow', 8);

F = hadamard(8);

H = hilb(8);

P = pascal(8);

[U, S, V] = mysvd(C)

U =

Columns 1 through 6

0.1491 + 0.0000i -0.3899 + 0.0000i -0.5050 + 0.0000i 0.5073 + 0.0000i 0.4341 + 0.0000i 0.3133 + 0.0000i

0.2195 + 0.0000i -0.4824 + 0.0000i -0.3681 + 0.0000i -0.0134 + 0.0000i -0.3917 + 0.0000i -0.5379 + 0.0000i

0.2839 + 0.0000i -0.4483 + 0.0000i 0.0527 + 0.0000i -0.5137 + 0.0000i -0.2766 + 0.0000i 0.3415 + 0.0000i

0.3405 + 0.0000i -0.2964 + 0.0000i 0.4329 + 0.0000i -0.2298 + 0.0000i 0.5029 + 0.0000i 0.1222 + 0.0000i

0.3876 + 0.0000i -0.0667 + 0.0000i 0.4792 + 0.0000i 0.4049 + 0.0000i 0.0743 + 0.0000i -0.4903 + 0.0000i

0.4241 + 0.0000i 0.1806 + 0.0000i 0.1560 + 0.0000i 0.4215 + 0.0000i -0.5328 + 0.0000i 0.4746 + 0.0000i

0.4489 + 0.0000i 0.3804 + 0.0000i -0.2875 + 0.0000i -0.2053 + 0.0000i 0.1401 + 0.0000i -0.0873 + 0.0000i

0.4489 + 0.0000i 0.3804 + 0.0000i -0.2875 + 0.0000i -0.2053 + 0.0000i 0.1401 + 0.0000i -0.0873 + 0.0000i

Columns 7 through 8

-0.1635 + 0.0000i 0.0000 + 2.1500i

0.3749 + 0.0000i 0.0000 + 0.9389i

-0.5086 + 0.0000i 0.0000 + 1.9009i

0.5368 + 0.0000i 0.0000 + 1.1321i

-0.4537 + 0.0000i 0.0000 + 0.9270i

0.2765 + 0.0000i 0.0000 + 0.1855i

-0.0420 + 0.0000i 0.0000 + 0.1914i

-0.0420 + 0.0000i 0.0000 + 0.1914i

S =

Columns 1 through 6

6.0208 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 1.9513 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 1.1387 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.8094 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.6452 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.5575 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

Columns 7 through 8

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.5135 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

V =

0.4489 -0.3804 -0.2875 0.2053 0.1401 0.0873 -0.0420 0.7071

0.4489 -0.3804 -0.2875 0.2053 0.1401 0.0873 -0.0420 -0.7071

0.4241 -0.1806 0.1560 -0.4215 -0.5328 -0.4746 0.2765 0.0000

0.3876 0.0667 0.4792 -0.4049 0.0743 0.4903 -0.4537 -0.0000

0.3405 0.2964 0.4329 0.2298 0.5029 -0.1222 0.5368 0.0000

0.2839 0.4483 0.0527 0.5137 -0.2766 -0.3415 -0.5086 0.0000

0.2195 0.4824 -0.3681 0.0134 -0.3917 0.5379 0.3749 0.0000

0.1491 0.3899 -0.5050 -0.5073 0.4341 -0.3133 -0.1635 -0.0000

[U, S, V] = mysvd(F)

U =

0.3536 0.3536 0.3536 0.3536 0.3536 0.3536 0.3536 0.3536

0.3536 -0.3536 0.3536 -0.3536 0.3536 -0.3536 0.3536 -0.3536

0.3536 0.3536 -0.3536 -0.3536 0.3536 0.3536 -0.3536 -0.3536

0.3536 -0.3536 -0.3536 0.3536 0.3536 -0.3536 -0.3536 0.3536

0.3536 0.3536 0.3536 0.3536 -0.3536 -0.3536 -0.3536 -0.3536

0.3536 -0.3536 0.3536 -0.3536 -0.3536 0.3536 -0.3536 0.3536

0.3536 0.3536 -0.3536 -0.3536 -0.3536 -0.3536 0.3536 0.3536

0.3536 -0.3536 -0.3536 0.3536 -0.3536 0.3536 0.3536 -0.3536

S =

2.8284 0 0 0 0 0 0 0

0 2.8284 0 0 0 0 0 0

0 0 2.8284 0 0 0 0 0

0 0 0 2.8284 0 0 0 0

0 0 0 0 2.8284 0 0 0

0 0 0 0 0 2.8284 0 0

0 0 0 0 0 0 2.8284 0

0 0 0 0 0 0 0 2.8284

V =

1 0 0 0 0 0 0 0

0 1 0 0 0 0 0 0

0 0 1 0 0 0 0 0

0 0 0 1 0 0 0 0

0 0 0 0 1 0 0 0

0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 0

0 0 0 0 0 0 0 1

[U, S, V] = mysvd(H)

U =

Columns 1 through 6

0.7203 + 0.0000i -0.6295 + 0.0000i 0.2775 + 0.0000i 0.0865 + 0.0000i 0.0207 + 0.0000i 0.0038 + 0.0000i

0.4325 + 0.0000i 0.1257 + 0.0000i -0.6449 + 0.0000i -0.5501 + 0.0000i -0.2661 + 0.0000i -0.0858 + 0.0000i

0.3188 + 0.0000i 0.2864 + 0.0000i -0.3352 + 0.0000i 0.3363 + 0.0000i 0.6241 + 0.0000i 0.4188 + 0.0000i

0.2552 + 0.0000i 0.3276 + 0.0000i -0.0513 + 0.0000i 0.4592 + 0.0000i -0.0320 + 0.0000i -0.5797 + 0.0000i

0.2139 + 0.0000i 0.3321 + 0.0000i 0.1440 + 0.0000i 0.2756 + 0.0000i -0.4194 + 0.0000i -0.1232 + 0.0000i

0.1845 + 0.0000i 0.3235 + 0.0000i 0.2729 + 0.0000i 0.0113 + 0.0000i -0.3655 + 0.0000i 0.4185 + 0.0000i

0.1625 + 0.0000i 0.3103 + 0.0000i 0.3570 + 0.0000i -0.2490 + 0.0000i -0.0180 + 0.0000i 0.3524 + 0.0000i

0.1453 + 0.0000i 0.2956 + 0.0000i 0.4116 + 0.0000i -0.4773 + 0.0000i 0.4780 + 0.0000i -0.4083 + 0.0000i

Columns 7 through 8

0.0006 + 0.0000i 0.0000 - 0.0005i

-0.0199 + 0.0000i 0.0000 + 0.0173i

0.1676 + 0.0000i 0.0000 - 0.1454i

-0.5264 + 0.0000i 0.0000 + 0.4552i

0.6052 + 0.0000i 0.0000 - 0.5156i

0.0778 + 0.0000i 0.0000 - 0.0916i

-0.6072 + 0.0000i 0.0000 + 0.5527i

0.3027 + 0.0000i 0.0000 - 0.2726i

S =

Columns 1 through 6

1.6959 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.2981 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0262 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0015 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0001 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

Columns 7 through 8

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

0.0000 + 0.0000i 0.0000 + 0.0000i

V =

0.7203 -0.6295 0.2775 0.0865 0.0207 0.0038 0.0005 0.0000

0.4325 0.1257 -0.6449 -0.5501 -0.2661 -0.0858 -0.0190 -0.0006

0.3188 0.2864 -0.3352 0.3363 0.6241 0.4188 0.1613 -0.0065

0.2552 0.3276 -0.0513 0.4592 -0.0320 -0.5797 -0.5175 0.0953

0.2139 0.3321 0.1440 0.2756 -0.4194 -0.1232 0.6391 -0.3843

0.1845 0.3235 0.2729 0.0113 -0.3655 0.4185 -0.0404 0.6901

0.1625 0.3103 0.3570 -0.2490 -0.0180 0.3523 -0.4797 -0.5773

0.1453 0.2956 0.4116 -0.4773 0.4780 -0.4083 0.2560 0.1833

[U, S, V] = mysvd(P)

U =

0.0004 -0.0101 -0.1115 0.5279 0.7379 0.3852 0.1234 -0.0246

0.0026 -0.0469 -0.2879 0.5622 -0.0478 -0.6097 -0.4490 0.1523

0.0110 -0.1313 -0.4686 0.3041 -0.4085 0.0148 0.5796 -0.4093

0.0345 -0.2747 -0.5254 -0.1083 -0.1928 0.4431 -0.1412 0.6181

0.0909 -0.4554 -0.3298 -0.3584 0.2497 -0.0005 -0.4067 -0.5650

0.2107 -0.5801 0.1203 -0.1490 0.2768 -0.4362 0.4669 0.3122

0.4434 -0.4285 0.4886 0.3698 -0.3213 0.2984 -0.2066 -0.0965

0.8657 0.4213 -0.2171 -0.1170 0.0837 -0.0628 0.0345 0.0128

S =

1.0e+03 \*

4.5437 0 0 0 0 0 0 0

0 0.1488 0 0 0 0 0 0

0 0 0.0119 0 0 0 0 0

0 0 0 0.0020 0 0 0 0

0 0 0 0 0.0005 0 0 0

0 0 0 0 0 0.0001 0 0

0 0 0 0 0 0 0.0000 0

0 0 0 0 0 0 0 0.0000

V =

0.0004 -0.0101 -0.1115 0.5279 0.7379 0.3852 0.1234 -0.0246

0.0026 -0.0469 -0.2879 0.5622 -0.0478 -0.6097 -0.4490 0.1523

0.0110 -0.1313 -0.4686 0.3041 -0.4085 0.0148 0.5796 -0.4094

0.0345 -0.2747 -0.5254 -0.1083 -0.1928 0.4431 -0.1412 0.6182

0.0909 -0.4554 -0.3298 -0.3584 0.2497 -0.0005 -0.4067 -0.5651

0.2107 -0.5801 0.1203 -0.1490 0.2768 -0.4362 0.4669 0.3123

0.4434 -0.4285 0.4886 0.3698 -0.3213 0.2984 -0.2066 -0.0965

0.8657 0.4213 -0.2171 -0.1170 0.0837 -0.0628 0.0345 0.0128

B.

Using the singular values in the S matrices printed above, here are the conditional numbers of each matrix:

Chow: 6.0208 / 0.6452 = 9.33

Hadamard: 2.8284/2.8284 = 1

Hilbert: 1.6959/0.0001 = 16959

Pascal: 4.5437 / 0.0001 = 45437

C.

A = double(imread('Joseph\_Fourier\_1820.jpg'))/255;

R = A(:,:,1);

G = A(:,:,2);

B = A(:,:,3);

A = (R+G+B)/3;

[U, S, V] = mysvd(A);

S(1:5, 1:5) = 0;

Serror5 = S;

Aerror5 = U\*Serror5\*V';

imshow(Aerror5);



S(1:10, 1:10) = 0;

Serror10 = S;

Aerror10 = U\*Serror10\*V';

imshow(Aerror10);

S(1:20, 1:20) = 0;

Serror20 = S;

Aerror20 = U\*Serror20\*V';

imshow(Aerror20);



S(1:50, 1:50) = 0;

Serror50 = S;

Aerror50 = U\*Serror50\*V';

imshow(Aerror50);

------------------------------------------------------------------------------------------------------------------------------------------

Problem 2:

A.

function r = quaternionmultiply(p, q)

r = zeros(1,4);

r(1) = q(1)\*p(1) - q(2)\*p(2) - q(3)\*p(3) - q(4)\*p(4);

r(2) = q(1)\*p(2) + q(2)\*p(1) - q(3)\*p(4) + q(4)\*p(3);

r(3) = q(1)\*p(3) + q(2)\*p(4) + q(3)\*p(1) - q(4)\*p(2);

r(4) = q(1)\*p(4) - q(2)\*p(3) + q(3)\*p(2) + q(4)\*p(1);

%---------------------------------------------------

p = [3 2 5 4];

q = [4 5 3 1];

r = quaternionmultiply(p,q);

r =

-17 16 47 0

B.

function qinv = quaternioninverse(q)

qinv = [q(1) -q(2:4)] / (q\*q');

%---------------------------------------------------

q = [8 2 3 1];

qinv = quaternioninverse(q)

qinv =

0.1026 -0.0256 -0.0385 -0.0128

qinv \* (q\*q')

ans =

8.0000 -2.0000 -3.0000 -1.0000

qinv \* norm(q)

ans =

0.9058 -0.2265 -0.3397 -0.1132

C.

function q = rotationquaternion(theta, v)

v = v/norm(v);

q = [cos(theta/2) sin(theta/2)\*v];

%------------------------------------------

x = [1 0 0];

theta = pi/6;

z = [0 0 1];

x = [0 x];

q = rotationquaternion(theta, z);

tmp = quaternionmultiply(quaternioninverse(q), x);

w = quaternionmultiply(tmp, q)

w =

0 0.8660 -0.5000 0

D.

u = [1 2 3];

theta = pi/4;

v = [1 1 1];

u = [0 u];

q = rotationquaternion(theta, v);

tmp = quaternionmultiply(quaternioninverse(q), u);

w = quaternionmultiply(tmp, q)

q1 = quiver3(0,0,0, 1, 2, 3);

hold on;

q2 = quiver3(0,0,0, w(1), w(2), w(3));

hold on;

%rotate axes

tmp = quaternionmultiply(quaternioninverse(q), [0 1 0 0]);

x = quaternionmultiply(tmp, q);

tmp = quaternionmultiply(quaternioninverse(q), [0 0 1 0]);

y = quaternionmultiply(tmp, q);

tmp = quaternionmultiply(quaternioninverse(q), [0 0 0 1]);

z = quaternionmultiply(tmp, q);

q3 = quiver3([0,0,0], [0,0,0], [0,0,0], [x(1) y(1) z(1)], [x(2) y(2) z(2)], [x(3), y(3), z(3)]);

legend([q1 q2 q3], 'original u', 'rotated u', 'rotated axes');



E.

theta = pi/3;

v = [1 1 1];

q = rotationquaternion(theta, v);

tmp = quaternionmultiply(quaternioninverse(q), [0 1 0 0]);

x = quaternionmultiply(tmp, q);

tmp = quaternionmultiply(quaternioninverse(q), [0 0 1 0]);

y = quaternionmultiply(tmp, q);

tmp = quaternionmultiply(quaternioninverse(q), [0 0 0 1]);

z = quaternionmultiply(tmp, q);

graph = quiver3([0,0,0], [0,0,0], [0,0,0], [x(1) y(1) z(1)], [x(2) y(2) z(2)], [x(3), y(3), z(3)]);

graph.ShowArrowHead = 'off';



p = -q;

tmp = quaternionmultiply(quaternioninverse(p), [0 1 0 0]);

x = quaternionmultiply(tmp, p);

tmp = quaternionmultiply(quaternioninverse(p), [0 0 1 0]);

y = quaternionmultiply(tmp, p);

tmp = quaternionmultiply(quaternioninverse(p), [0 0 0 1]);

z = quaternionmultiply(tmp, p);

graph = quiver3([0,0,0], [0,0,0], [0,0,0], [x(1) y(1) z(1)], [x(2) y(2) z(2)], [x(3), y(3), z(3)]);

graph.ShowArrowHead = 'off';



The two graphs look the same, using rotation q and p = -q.

-------------------------------------------------------------------------------------------------------------------------------

Problem 3:

A.

A = zeros(14800, 13);

for i=1:13

F = convertimage(sprintf('faces/basis/f%d.jpg', i));

f = F(:);

A(:,i) = f/norm(f);

end

%-------------------------------------

function A = convertimage(Img)

A = double(imread(Img))/255;

R = A(:,:,1);

G = A(:,:,2);

B = A(:,:,3);

A = (R+G+B)/3;

B.

T2 = convertimage('faces/tests/t2.jpg');

b = T2(:);

C.

x = pinv(A'\*A)\*A'\*b;

v = A\*x;

v = v/(max(v));

imshow(reshape(v,148,100));



imshow(T2);

%squared error

norm(v-b)^2

ans =

295.5664

D.

x =

3.3883

6.2665

-2.7240

32.5422

13.7122

12.4288

-6.4815

-3.6926

14.9529

-16.5462

12.0888

0.5861

23.4545

Based on the numbers above, the three basis images corresponding to absolute largest coefficients are: f4, f13, f10.

E.

RHO = corr(A)

Columns 1 through 12

1.0000 0.7795 0.7890 0.7363 0.6467 0.7252 0.7518 0.6863 0.7336 0.6895 0.7474 0.6838

0.7795 1.0000 0.8511 0.7576 0.7304 0.7522 0.8225 0.7635 0.8120 0.7794 0.8189 0.7193

0.7890 0.8511 1.0000 0.8183 0.7568 0.7670 0.8861 0.7723 0.8914 0.8243 0.8492 0.7969

0.7363 0.7576 0.8183 1.0000 0.8242 0.8538 0.7551 0.6620 0.8038 0.7517 0.7610 0.7618

0.6467 0.7304 0.7568 0.8242 1.0000 0.7922 0.7513 0.6919 0.8157 0.8221 0.7577 0.7650

0.7252 0.7522 0.7670 0.8538 0.7922 1.0000 0.7258 0.6858 0.7477 0.7390 0.7161 0.7190

0.7518 0.8225 0.8861 0.7551 0.7513 0.7258 1.0000 0.7801 0.9060 0.8846 0.8955 0.8116

0.6863 0.7635 0.7723 0.6620 0.6919 0.6858 0.7801 1.0000 0.7774 0.7322 0.8152 0.8004

0.7336 0.8120 0.8914 0.8038 0.8157 0.7477 0.9060 0.7774 1.0000 0.8889 0.9135 0.8539

0.6895 0.7794 0.8243 0.7517 0.8221 0.7390 0.8846 0.7322 0.8889 1.0000 0.8354 0.7966

0.7474 0.8189 0.8492 0.7610 0.7577 0.7161 0.8955 0.8152 0.9135 0.8354 1.0000 0.8171

0.6838 0.7193 0.7969 0.7618 0.7650 0.7190 0.8116 0.8004 0.8539 0.7966 0.8171 1.0000

0.7091 0.7694 0.8120 0.7698 0.7801 0.7246 0.8482 0.8308 0.8887 0.8168 0.8945 0.8915

Column 13

0.7091

0.7694

0.8120

0.7698

0.7801

0.7246

0.8482

0.8308

0.8887

0.8168

0.8945

0.8915

1.0000

RHO = corr(A);

RHO2 = RHO;

RHO2(logical(eye(13))) = 0;

[Y I] = max(RHO2)

Highest correlation: 0.9135, between images f9 and f11.

F.

COEFF = pca(RHO)

COEFF =

-0.1624 -0.4603 -0.3172 0.3390 -0.3971 -0.1007 0.1143 -0.0415 -0.2349 -0.1375 0.0713 0.1211

0.0608 -0.2289 -0.3685 -0.2964 0.4712 -0.3551 -0.2131 0.5408 0.0310 -0.0088 -0.0076 0.0912

0.1687 -0.0541 -0.3998 0.1601 0.2570 0.1229 -0.4365 -0.4040 -0.3343 -0.1357 0.1276 -0.4170

-0.1984 0.3628 -0.2328 0.4436 0.2679 -0.1811 -0.0451 -0.2250 0.5976 -0.0847 0.0121 0.1113

-0.0296 0.5594 -0.0013 -0.2822 -0.1098 -0.4657 -0.0314 -0.2086 -0.4063 0.1995 0.1102 -0.0038

-0.3115 0.2780 -0.1059 0.0980 0.4081 0.5306 0.3567 0.2281 -0.3711 -0.0227 -0.0006 0.0159

0.3621 -0.0210 -0.3243 -0.0465 -0.0571 0.3071 0.0610 -0.0744 0.1564 0.6759 0.3360 0.2341

0.2917 -0.2622 0.4251 -0.1785 0.4179 0.1052 0.0145 -0.4128 0.0227 -0.1619 0.0106 0.2035

0.3501 0.1691 -0.2319 0.1126 -0.0155 -0.0429 0.0617 -0.0787 -0.2459 -0.2608 -0.3330 0.6699

0.2865 0.2702 -0.2471 -0.3710 -0.3257 0.3300 -0.0337 0.1216 0.2639 -0.4833 0.0059 -0.1770

0.3839 -0.0460 -0.1402 0.1012 0.0985 -0.2098 0.5675 -0.0492 0.0368 0.1494 -0.4571 -0.4543

0.3124 0.1701 0.2923 0.4421 -0.0759 0.1179 -0.4706 0.3666 -0.1004 0.1873 -0.2841 -0.0662

0.3720 0.0951 0.1804 0.3091 0.0642 -0.2085 0.2614 0.2504 -0.0758 -0.2702 0.6729 -0.0578

The first two principal components are the first 2 columns of the above matrix.

G.

c1 = COEFF(:,1);

c2 = COEFF(:,2);

%pairwise distances

D = squareform(pdist([c1 c2]))

D(logical(eye(13))) = Inf;

m = min(D)

Images f7 and f11 have coordinates closest to each other.

------------------------------------------------------------------------------------------------------------------------------------------Problem 4:

A.

[time ixic] = read\_stock('IXIC.csv');



B.

[time ixic] = read\_stock('IXIC.csv');

t1 = time(2:7350);

y1 = ixic(2:7350);

t2 = time(9611:11287);

y2 = ixic(9611:11287);

%Polynomial Fit:

%First time period:

X4 = [t1.^4 t1.^3 t1.^2 t1 t1.^0];

c1p = X4 \ y1; %find coefficients of degree 4 polynomial

%Second time period:

X4 = [t2.^4 t2.^3 t2.^2 t2 t2.^0];

c1p = X4 \ y2; %find coefficients of degree 4 polynomial

%Exponential fit:

%First time period:

Ylog = log(y1);

T1 = [t1.^0 t1];

c1log = Ylog \ T1

%Second time period:

Ylog = log(y2);

T2 = [t2.^0 t2];

c2log = Ylog \ T2

%Polynomial Fit:

%First time period:

c1p = polyfit(t1, y1, 4); %find coefficients of degree 4 polynomial

%Second time period:

c2p = polyfit(t2, y2, 4); %find coefficients of degree 4 polynomial

%Exponential fit:

%First time period:

ylog = log(y1);

c1log = polyfit(t1, ylog, 1);

polylog1 = polyval(c1log, t1);

%Second time period:

ylog = log(y2);

c2log = polyfit(t2, ylog, 1);

polylog2 = polyval(c2log, t2);

hold on;

plot(t1, polyval(c1p, t1), 'r', t2, polyval(c2p, t2), 'r');

hold on;

plot(t1, exp(polylog1), 'g', t2, exp(polylog2), 'g');

hold on;

title(sprintf('poly1: %d %d %d %d %d, poly2: %d %d %d %d %d \nexp1: %d %d, exp2: %d %d', c1p, c2p, c1log, c2log))



Squared Errors:

%squared errors:

polyerror1 = norm(polyval(c1p, t1) - y1)^2

polyerror2 = norm(polyval(c2p, t2) - y2)^2

experror1 = norm(exp(polylog1) - y1)^2

experror2 = norm(exp(polylog2) - y2)^2

polyerror1 =

1.5922e+09

polyerror2 =

9.2822e+04

experror1 =

3.3657e+09

experror2 =

4.4669e+05

It looks like the polynomial function fits better, since it has smaller errors for both time periods.

-------------------------------------------------------------------------------------------------------------------------------Problem 5:



The power spectrum value at the spike is 7.9806, based on the output of log\_power\_spectrum(2:floor(n/2)), which shows a suddenly low value in the middle of values around ~10.

I = find(v < 8)

I =

259

The index is at 259, so looking at the frequency vector, the frequency corresponding to this spike is

frequencies(259)

ans =

0.2051

Per month, this value is 0.2051 \* 252/12 = 4.3071.