**COMPUTER PROBLEMS:**

**Q 3-1)**

Using curve fitting tool in MATLAB to obtain following polynomial and plot them.

Linear model Poly1:

f(x) = p1\*x + p2

Coefficients (with 95% confidence bounds):

p1 = 1.706 (1.261, 2.15)

p2 = 1.319 (-0.02622, 2.664)



Linear model Poly2:

f(x) = p1\*x^2 + p2\*x + p3

Coefficients (with 95% confidence bounds):

p1 = -0.09464 (-0.4578, 0.2685)

p2 = 2.179 (0.2873, 4.071)

p3 = 1.004 (-1.007, 3.015)



Linear model Poly3:

f(x) = p1\*x^3 + p2\*x^2 + p3\*x + p4

Coefficients (with 95% confidence bounds):

p1 = 0.0713 (-0.3303, 0.4729)

p2 = -0.6294 (-3.687, 2.429)

p3 = 3.156 (-2.998, 9.309)

p4 = 0.7897 (-2.378, 3.958)



Linear model Poly4:

f(x) = p1\*x^4 + p2\*x^3 + p3\*x^2 + p4\*x + p5

Coefficients (with 95% confidence bounds):

p1 = 0.1062 (-0.5202, 0.7327)

p2 = -0.9912 (-7.296, 5.313)

p3 = 2.634 (-17.34, 22.61)

p4 = 0.12 (-20.79, 21.03)

p5 = 0.9718 (-4.7, 6.643)



Linear model Poly5:

f(x) = p1\*x^5 + p2\*x^4 + p3\*x^3 + p4\*x^2 + p5\*x + p6

Coefficients:

p1 = -0.05917

p2 = 0.8458

p3 = -4.212

p4 = 8.304

p5 = -3.178

p6 = 1



Fifth order polynomal capture the general trend of daha better if we think that polynomal passes near the data point.

**Q 3-4)**

**a)**

Using cholesky factorization:

clear all;

clc;

b=[0.26;0.28;3.31];

A=[0.16,0.1;0.17,0.11;2.02,1.29];

b=A'\*b;

[m,n]=size(A);

B=zeros(n);

%find B=A'\*A

for i=1:m

for j=1:n

for k=1:n

B(j,k)=B(j,k)+A(i,k)\*A(i,j);

end

end

end

%cholesky factorization

[m,n]=size(B);

for k=1:n

B(k,k)=sqrt(B(k,k));

for i=k+1:n

B(i,k)=B(i,k)/B(k,k);

end

for j=k+1:n

for i=k+1:n

B(i,j)=B(i,j)-B(i,k)\*B(j,k);

end

end

end

% L=B and U=L'

%forward subsitition

% Ly=b

for j=1:n

if B(j,j)==0

break;

end

y(j)=b(j)/B(j,j);

for i=j+1:n

b(i)=b(i)-B(i,j)\*y(j);

end

end

%backward subsitition

% Ux=y

U=B';

for j=n:-1:1

if U(j,j)==0

break;

end

x(j)=y(j)/U(j,j);

for i=1:j-1

y(i)=y(i)-U(i,j)\*x(j);

end

end

Solution is x = [1.0000 1.0000]

**b)**

When we change the right side vector b, then solution is

x = [7.0089 -8.3957]

c) Results of a and b differ from each other, this mean that condition number of the AT\*A is too greater than 1. Condition number of A is equal to

cond(A)=1.0975e+03

**Q 3-4)**

**a)**

clear all;

clc;

%given values

x=[1.02 0.95 0.87 0.77 0.67 0.56 0.44 0.30 0.16 0.01]';%x

y=[0.39 0.32 0.27 0.22 0.18 0.15 0.13 0.12 0.13 0.15]';%y

z=[1,1,1,1,1,1,1,1,1,1]';

%obtain A and f such taht Ax=f

A=[y.^2,x.\*y,x,y,z];

f=x.^2;

%solve the least square solution

%this solution gives coefficients of ellipse

sol=A\f; % sol=[a b c d e]'

%coefficients

a=sol(1);

b=sol(2);

c=sol(3);

d=sol(4);

e=sol(5);

%obtain boundry of x1 parameter from solve(B^2-4aC)

x1=-0.4897:0.001:1.1302;

%ellipse

B=b\*x1+d;

C=e+c\*x1-x1.^2;

y1=(-B+sqrt(B.^2-4\*a\*C))/(2\*a); % lower part of ellipse

y2=(-B-sqrt(B.^2-4\*a\*C))/(2\*a); % upper part of ellipse

%plot ellipse and x,y valuse on the same figure

plot(x1,y1)

xlabel('x values');

ylabel('y values');

hold on

plot(x1,y2);

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

Ellipse and given values are plotted on the same figure.



**b)**

We perturb the input data and plot the original and new ellipse on the same figüre.

clear all;

clc;

%given values

x=[1.02 0.95 0.87 0.77 0.67 0.56 0.44 0.30 0.16 0.01]';%x

y=[0.39 0.32 0.27 0.22 0.18 0.15 0.13 0.12 0.13 0.15]';%y

z=[1,1,1,1,1,1,1,1,1,1]';

%perturb the input data

n=length(x);

xpert=x+(rand([n,1])\*0.01-0.005);

ypert=y+(rand([n,1])\*0.01-0.005);

%obtain A and f such taht Ax=f

A=[ypert.^2,xpert.\*ypert,xpert,ypert,z];

f=xpert.^2;

%solve the least square solution

%this solution gives coefficients of ellipse

sol=A\f; % sol=[a b c d e]'

%coefficients

a=sol(1);

b=sol(2);

c=sol(3);

d=sol(4);

e=sol(5);

%obtain boundry of x1 parameter from solve(B^2-4aC)

x1=-0.4897:0.001:1.1302;

%ellipse

B=b\*x1+d;

C=e+c\*x1-x1.^2;

y1=(-B+sqrt(B.^2-4\*a\*C))/(2\*a); % lower part of ellipse

y2=(-B-sqrt(B.^2-4\*a\*C))/(2\*a); % upper part of ellipse

%plot ellipse and x,y valuse on the same figure

plot(x1,y1)

xlabel('x values');

ylabel('y values');

hold on

plot(x1,y2);

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

n=length(x);

x\_per=x+(rand([n,1])\*0.01-0.005);

y\_per=y+(rand([n,1])\*0.01-0.005);



We note that the parameters that define the ellipse should be approximately the same and the ellipses should look approximately the same. But the plot of the ellipses reveals remarkably visual differences, even though the perturbation is small. Computing the condition number of the original matrix A, we notice that this is very large and that the problem thus is nearly rank deficient, so small perturbations of the data can result in loss of data in the computing process.

**d)**

Compute the singular decomposition of A:

clear all;

clc;

%given values

x=[1.02 0.95 0.87 0.77 0.67 0.56 0.44 0.30 0.16 0.01]';%x

y=[0.39 0.32 0.27 0.22 0.18 0.15 0.13 0.12 0.13 0.15]';%y

z=[1,1,1,1,1,1,1,1,1,1]';

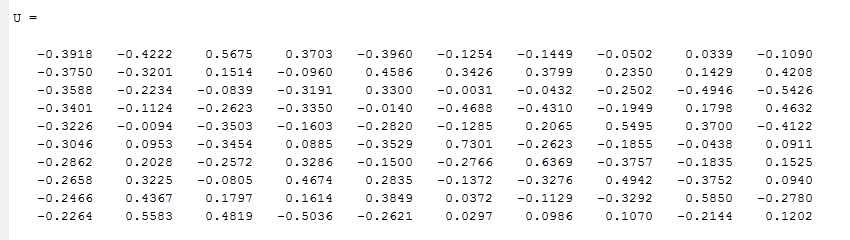
%obtain A and f such taht Ax=f

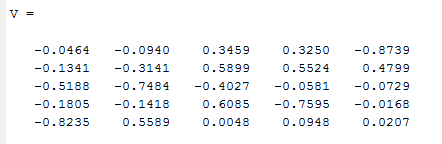
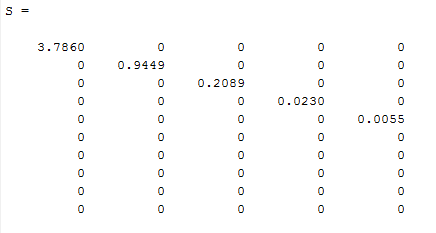
A=[y.^2,x.\*y,x,y,z];

f=x.^2;

%singular value decomposition

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





**f)**

Compute the singular decomposition of distributed A:

clear all;

clc;

%given values

x=[1.02 0.95 0.87 0.77 0.67 0.56 0.44 0.30 0.16 0.01]';%x

y=[0.39 0.32 0.27 0.22 0.18 0.15 0.13 0.12 0.13 0.15]';%y

z=[1,1,1,1,1,1,1,1,1,1]';

%perturb the input data

n=length(x);

xpert=x+(rand([n,1])\*0.01-0.005);

ypert=y+(rand([n,1])\*0.01-0.005);

%obtain A and f such taht Ax=f

A=[ypert.^2,xpert.\*ypert,xpert,ypert,z];

f=xpert.^2;

%singular value decomposition

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

