close all

clear all

load hw1data

%% a) solve the system of equations to find a, b, c in terms of

% J1 J2 J3 J4 K1 K2 K3 K4 L1 L2 L3 L4, where:

% % coefficient for eq1

J1 = transpose(x.^2)\*y;

J2 = sum(x.^4);

J3 = sum(x.^3);

J4 = sum(x.^2);

% coefficient for eq2

K1 = transpose(x)\*y;

K2 = sum(x.^3);

K3 = sum(x.^2);

K4 = sum(x);

% coefficient for eq3

L1 = sum(y);

L2 = sum(x.^2);

L3 = sum(x);

L4 = 100;

% Uncomment this and comment all coefficents above if you

% want to solve a, b, c with parameters

% syms a b c J1 J2 J3 J4 K1 K2 K3 K4 L1 L2 L3 L4

% eq1 = J1 - J2 \* a - J3 \* b - J4 \* c == 0;

% eq2 = K1 - K2 \* a - K3 \* b - K4 \* c == 0;

% eq3 = L1 - L2 \* a - L3 \* b - L4 \* c == 0;

% S = solve(eq1, eq2, eq3);

% S = [S.a, S.b, S.c];

% Uncomment this is you want to solve a, b, c numerically

syms a\_q b\_q c\_q

eq1 = J1 - J2 \* a\_q - J3 \* b\_q - J4 \* c\_q == 0;

eq2 = K1 - K2 \* a\_q - K3 \* b\_q - K4 \* c\_q == 0;

eq3 = L1 - L2 \* a\_q - L3 \* b\_q - L4 \* c\_q == 0;

S = solve(eq1, eq2, eq3);

S = vpa([S.a\_q, S.b\_q, S.c\_q]);

a\_q = S(1);

b\_q = S(2);

c\_q = S(3);

% Noise with Gaussian Distribution

mu = 0;

sigma = 1;

N\_norm = (1/(sqrt(2\*pi)\*sigma)) \* exp(-(y-a\_q\*x.^2-b\_q\*x-c\_q).^2./2);

% quadratic approximation, analytical solution

Y\_q = a\_q \* x.^2 + b\_q \* x + c\_q + N\_norm;

% plot

figure(1)

scatter(x, y , 'r+')

grid on

hold on

scatter(x, Y\_q, 'bo')

legend('input data', 'analytical quadratic approximation')

title('Quadratic fitting subject to AWGN~(0,1)')

hold off

%% b) solve linear programming problem

A = [x.^2 x ones(100,1)]; % coef. of the ineq.

t = y; % constraint, <=

f = [-sum(x.^2) -sum(x) -100]; % objective func., find its min

output = linprog(f,A,t); % output, optimized a, b, and c

% optimized a, b, and c after linear programming

a\_l = output(1);

b\_l = output(2);

c\_l = output(3);

N\_exp = exp(-(y-a\_l\*x.^2-b\_l\*x-c\_l));

Y\_l = a\_l \* x.^2 + b\_l \* x + c\_l + N\_exp;

figure(2)

scatter(x, y , 'r+')

grid on

hold on

scatter(x, Y\_l, 'go')

legend('input data', 'analytical quadratic approximation')

title('Quadratic fitting subject to exp(-z) (z>=0) noise')

hold off

