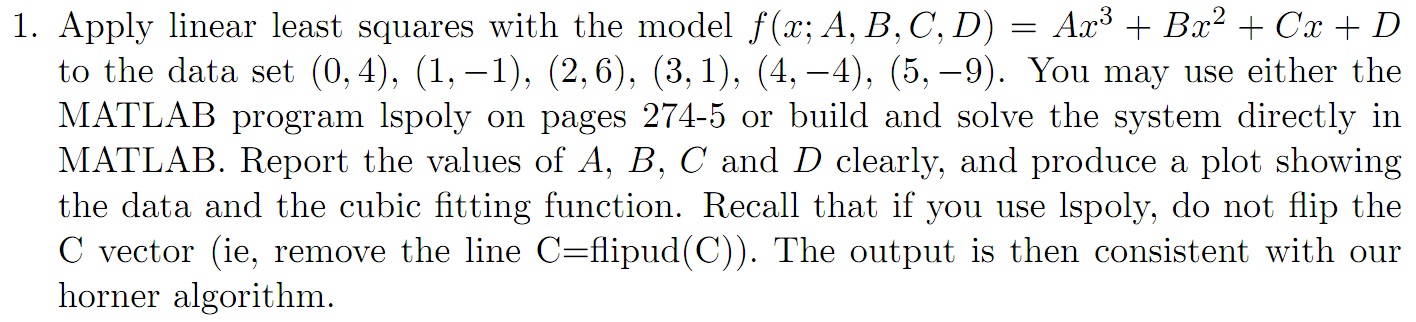
Ryan Brosnahan

Homework 5





function C = lspoly(X,Y,M)

n = length(X);

B = zeros(1:M+1);

F = zeros(n,M+1);

for k = 1:M+1

F(:,k) = X'.^(k-1);

end

A = F'\*F;

B = F'\*Y';

C = A\B;

D = 2.952380952380991e+00

C = -1.015873015873156e+00

B = 8.095238095238816e-01

A = -2.222222222222315e-01

>> X = [0 1 2 3 4 5];

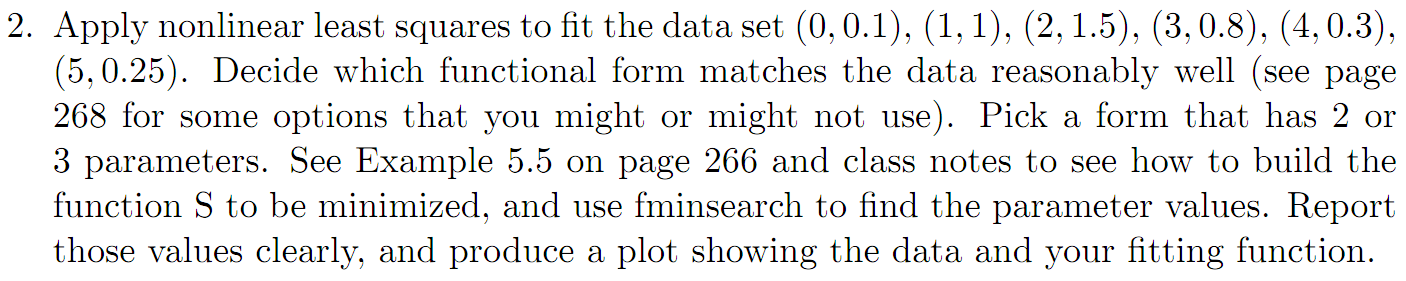
>> Y = [4 -1 6 1 -4 -9];

>> c = lspoly(X, Y, 3)

>> x = 0:.01:5

>> f = c(1) + c(2)\*x + c(3)\*x.^2 + c(4)\*x.^3

>> plot(X, Y, 'o', x, f)



Try



Nope!

>> X = 0:5;

Y = [0.1 1 1.5 0.8 0.3 0.25];

s = @(R) sum( (R(2)\*exp(R(1)\*X)-Y).^2 );

initguess = [1, 1];

u = fminsearch(s, initguess)

x = 0:0.01:5;

y = u(2) \* exp(u(1)\*x);

plot(X, Y, 'o', x, y)

A = -.05843

B = .7601

Function looks like the form

But the code is wrong somehow:

% y = Ax \* e^(Bx) + C -> R(1) = A R(2) = B R(3) = C

X = 0:5;

Y = [0.1 1 1.5 0.8 0.3 0.25];

s = @(R) sum( ( R(1) \* X \* exp( R(2)\*X ) + R(3) - Y ).^2 );

initguess = [1, 1, 1];

u = fminsearch(s, initguess)

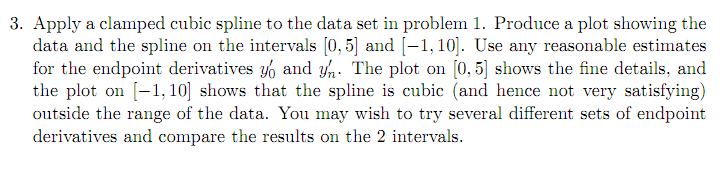
x = 0:0.01:5;

y = u(1)\* x \* exp(u(2)\*x + u(3));

plot(X, Y, 'o', x, y)



This plot give some indication that the model wouldn’t be the best, but not terrible either.





Plot from -1 to 10 clamped with y’left = -0.5 and y’right = -1



Same, on 0 to 5

x = 0:5;

y = [4 -1 6 1 -4 -9];

ycl = [-0.5 y -1];

xp = -1:.01:10 % or 0:0.01:5

y1 = spline(x, ycl, xp);

plot(x, y, 'o', xp, y1)