MATH 313 Spring 2009, Homework 7, Computer-based problems

As we had discussed in previous assignments, the "\" operator solves $Ax = b$ when you execute "x=A\b". It does this by gaussian elimination if it can. But what if the system is going to be overdetermined (A has more rows than columns)? Gaussian elimination isn't going to be very useful then... In this case, Matlab computes the least-squares solution! (Recall the least-squares pseudoinverse, it's like that. Though actually, Matlab does something faster...) In this way, Matlab always finds a solution unless A is square and singular.

This assignment examines least-squares solutions to polynomial interpolation, which is a very handy thing. Suppose you have N data points of the form $\{(x_i, y_i)\}_{i=1}^N$. If you want to fit a degree d polynomial to this, what would you do? Well, you'd like to find $d+1$ coefficients $\{a_j\}_{j=0}^d$ such that

$$y_i = \sum_{k=0}^{d} a_j x_i^j$$

for each $i = 1...N$.

Problem 1 Turn this problem specification into a matrix equation involving the column vector of coefficients

$$A = \begin{pmatrix} a_0 \\ a_1 \\ ... \\ a_d \end{pmatrix},$$

the column vector of outputs

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_N \end{pmatrix},$$

and some matrix V involving powers of the x_i. Write a Matlab function to solve this equation for A given a vector of x_i values (perhaps call it X) and Y. Hint: V is called a "Vandermonde matrix."

Problem 2 Once you've got the function in Problem 1 working, write another function for evaluating a polynomial with coefficients vector A at any point x you specify. (This doesn't actually depend on solving Problem 1 successfully.)

Problem 3 Write a script to demonstrate the above two functions you've written. In particular, you should:

(1) Randomly create a list of data points
(2) Choose a degree for your polynomial (you can make this a hard-coded value)
(3) Compute the polynomial coefficient vector
(4) Compute finely spaced points on the polynomial
(5) Plot the data points and the polynomial. You should use two different colors or symbols (type "help plot" to get some ideas)
(6) Label your axes, and print the output figure

Problem 4 (Optional Challenge) Generalize the above to produce a polynomial surface from a set of random data points $\{(x_i, y_i, z_i)\}_{i=1}^N$. There are several ways to think about this! You can talk to me about this if you like – it's a solution I've used in some actual signal processing algorithms...