Least Squares Approximation - computer problem

Consider the problem of approximating a set of data points $(x_1, y_1), ..., (x_m, y_m)$ by a function of the form

$$\phi(x) = c_1\phi_1(x) + \cdots + c_n\phi_n(x)$$

where $m > n$. The *basis functions* $\phi_j(x)$ are chosen according to the type of approximation desired. For example, to generate an $n - 1^{\underline{\text{st}}}$ degree polynomial, one might choose the *monomials* $\phi_j(x) = x^{j-1}$, $j = 1, ..., n$. Actually, this choice of basis functions is an exceedingly bad one from the standpoint of numerical stability, as you will see. [The reason is that, over a given approximation interval $[a, b]$, they become increasingly (exponentially) dependent on one another as n increases. A much better choice would be the *Legendre* polynomials, which have the orthogonality property $\int_a^b \phi_i(x)\,\phi_j(x)\,dx = 0$ for $i \neq j$.]

The conditions for an exact fit at all the data points, $\phi_j(x_i) = y_i$, $i = 1, ..., m$, can be written:

$$A_{m\times n}\, c_{n\times 1} = y_{m\times 1}$$

where $A_{i,j} = \phi_j(x_i)$, $1 \le i \le m$, $1 \le j \le n$. But $m > n$, so in general this is an overdetermined system with no solution. Instead, we seek a *least squares solution* by minimizing

$$E(c) \equiv \sum_k [\phi(x_k) - y_k]^2 = \|Ac - y\|_2^2.$$

The conditions for a minimum are $\frac{\partial\phi}{\partial c_i} = 0$, $i = 1, .., n$. Taking these derivatives, we obtain the following $n \times n$ system for the least squares solution to $Ac = y$:

$$A^T Ac = A^T y.$$

These are known as the *normal equations.*

Now consider the problem of approximating the data points,

$$(x_k, y_k),\ k = 1, ..., 51, \qquad x_k = .02(k - 1),\ y_k = 1 + x_k,$$

in the least squares sense by polynomials of degree $4, 8, 12$ ($n = 5, 9, 13$), represented in terms of monomial basis functions. Do this by setting up and solving the corresponding normal equations. Note that this is a situation where it's possible to get an exact fit (the data points are co-linear), and that is what must happen, apart from roundoff error. Thus what you'll be observing here is the sensitivity of the normal equations to roundoff error, a condition number dependent effect. For each computed solution $\hat{c}$, print the corresponding relative error $\frac{\|\hat{c}-c\|_2}{\|c\|_2}$ where c is the exact solution, an estimate of the condition number $\kappa(A^T A)$ of the coefficient matrix, and 'residual' $\|A\hat{c} - y\|_2$. Interpret your results. Is $\kappa(A^T A)$ a good predictor of the roundoff error present in $\hat{c}$?

Also, compute the same least squares solutions by a second method, via an SVD of A. Print the resulting coefficients $\hat{c}$, relative errors $\frac{\|\hat{c}-c\|_2}{\|c\|_2}$ and residuals $\|A\hat{c} - y\|_2$. Interpret your results, and compare to the normal equations approach.

If you use Matlab (highly recommended!), you'll find the following the following functions very useful: *norm, lu, cond, svd, diary.* Type '*help <function name>*' for documentation. A possible alternative would be to use routines from the Lapack library on Netlib (http://www.netlib.org/) - a primary resource for high quality numerical software.