Math 685/CSI 700/OR 682 Project 3
given 04/05/10, due in class 04/26/10

Each group should submit a report, which should include a short 2-5 page account on the implementation and solution, plus any necessary supplements in the form of graphs and/or tables, in class on the date listed above. Matlab codes should be sent to me via email (in archived form if necessary). Late submissions will bear a penalty.

Problem 1.
Compute all the roots of the polynomial $p(t) = 24 - 40t + 35t^2 - 13t^3 + t^4$ by forming the companion matrix and then calling an eigenvalue routine `eig` in MATLAB to compute its eigenvalues. Design your own eigenvalue solver appropriate for this calculation by noting that the companion matrix is already in Hessenberg form. Compare the results of your code with those obtained by MATLAB, both in terms of accuracy and speed. Then, compare the speed and accuracy of the companion matrix method with those of a library routine (such as `roots` in MATLAB) designed specifically for computing roots of polynomials. Try increasing the degree of the polynomial and comment on the differences you notice.

Problem 2.
Choose one of the datasets provided on the webpage and solve the associated eigenvalue problem. You need to implement your own version of a power method and comment on its accuracy/stability, as well as the results you obtain.

Problem 3.
Lorenz derived a simple system of ordinary differential equations describing buoyant convection in a fluid as a crude model for atmospheric circulation. At steady state, the convective velocity x, temperature gradient y and heat flow z satisfy the system of nonlinear equations

$$\begin{array}{rcrcrcl} & & \sigma(y & - & x) & = & 0, \\ rx & - & y & - & xz & = & 0, \\ & & xy & - & bz & = & 0, \end{array}$$

where σ (the Prandtl number), r(the Rayleigh number) and b are positive constants that depend on the properties of the fluid, the applied temperature gradient and the geometry of the problem. Typical values are $\sigma = 10, r = 28, b = 8/3$. Write a program using Newton's and Broyden's methods to solve this system of nonlinear equations. Try computing solutions for several initial guesses and modify the Newton's step if necessary to obtain a more robust implementation. You should be able to find three different solutions. Comment on the accuracy and speed of each of the methods you used.

Problem 4.
Write a program to find a minimum of the Rosenbrock's function

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

using each of the following methods:
1) steepest descent
2) Newton
3) damped Newton (Newton's method with a line search)

You should try each of the methods from each of the three starting points $(-1, 1), (0, 1), (2, 1)$. Plot the path taken in the plane by the appropriate solutions for each method from each starting point. Compare your results with those given by MATLAB functions `fminsearch` and `fminunc`. Comment on your observations.