MATH 609-602: Numerical Methods

Lecturer:	Prof. Wolfgang Bangerth Blocker Bldg., Room 507D (979) 845 6393 bangerth@math.tamu.edu
Teaching Assistant:	Seungil Kim Blocker Bldg., Room 507A (979) 862 3259 sgkim@math.tamu.edu

Homework assignment 6 – due Tuesday 10/18/2005

Problem 1 (Power method for extremal eigenvalue). Let N be the size of the matrix defined by

$$A_{ij} = \begin{cases} 2 + \frac{1}{N^2} & \text{if } i = j, \\ -1 & \text{if } i = j \pm 1, \\ 0 & \text{otherwise.} \end{cases}$$

This is a typical matrix in the numerical solution of partial differential equations and we can learn a great deal from it by looking at its eigenvalues.

Implement the power method for finding the largest eigenvalue of a matrix. Apply it to above matrix for the cases where $N = 10, 50, 100, 200, 500, 1000$.

Next implement the inverse power method for finding the smallest eigenvalue. (For the inverse power method, you need to multiply repeatedly with A^{-1}, i.e. compute $x_{k+1} = A^{-1}x_k$; you may use Matlab to actually compute A^{-1}, or use any of the methods we have learned in class to solve the linear system $Ax_{k+1} = x_k$ for x_{k+1}.) Apply it to the same set of matrices as above.

Generate a table that shows, for above values of N:

- the maximum eigenvalue of A
- the minimum eigenvalue of A
- the condition number of A in the l_2 norm (if you recall the formula for the condition number, you will see how to compute it from the maximum and minimum eigenvalues)
- the number of Steepest Descent iterations that would be required to solve for an accuracy of $\varepsilon = 10^{-8}$ (we had a formula that expressed this number in terms of the condition number)
- the number of Conjugate Gradient (CG) iterations that would be required to solve for an accuracy of $\varepsilon = 10^{-8}$ (same here).

What do we learn from this prototypical example concerning the behavior of matrices as they become larger and larger? **(7 points)**

Problem 2 (Polynomial interpolation). Consider the four points $(x_1, y_1) = (0, 0)$, $(x_2, y_2) = (1, 1)$, $(x_3, y_3) = (2, 2)$, $(x_4, y_4) = (3, 0)$. Compute, by hand, both the Lagrange and Newton form of the polynomial that exactly interpolates these points. **(4 points)**

Problem 3 (Polynomial interpolation). Write a program that computes the polynomial of order $N - 1$ that exactly interpolates N given points (x_i, y_i).

Define the following set of N points:

- for $1 \leq i < N$ let $x_i = \frac{i-1}{N-1}$, $y_i = 0$
- for $i = N$ let $x_N = 1$, $y_i = 1$

(For example, for $N = 4$, the four points are $\{(0, 0), (1/3, 0), (2/3, 0), (1, 1)\}$.) Apply your program to find the polynomials that interpolate these points for $N = 4, 8, 12, 20$. Plot the four polynomials you have found for the four choices of N on the interval $0 \leq x \leq 1$, verify that they indeed interpolate the given points, and describe their behavior between the interpolation points.

(6 points)