CS 322: Introduction to Scientific Computing
Spring 2003
Problem Set 5

Handed out: Wed., Apr. 9.

Due: Wed., Apr. 16 in lecture.

The policies for this (and other problem sets) are as follows:

- You should hand in your on-time problem set in lecture in the box at the front of the room on the day it is due. Problem sets handed in elsewhere (TA's office, Upson 303, etc.) will be considered late. See the next bullet.
- Late papers may be handed in up to 24 hours late. For instance, this problem set may be handed in up to 11:00 am on Apr. 17. You can hand in a late paper in Upson 303. Late papers get an automatic deduction of 10%. The full late penalty is applied even if you turn in part of the solution on time.
- Problem sets may be done individually or in teams of two. Put your name or names on the front page. Re-read the academic integrity statement on the web for the policy concerning working in larger groups.
- Problem sets count for 20% of the final course grade. The lowest scoring problem set will be dropped.
- You need Matlab for some of the questions. Matlab is available in the following CIT labs: Upson and Carpenter.
- If you need clarification for a homework question, please either ask your question in section, lecture, or office hours or else post it to the newsgroup cornell.class.cs322. The professor reads this newsgroup and will post an answer.
- Write your section number (like this: "Section 2") at the top of the front page of your paper, and circle it. This is the section where your graded paper will be returned. As a reminder, Section 1 is Th 12:20, Section 2 is Th 3:35, Section 3 is F 2:30, Section 4 is F 3:35.

1. Let A be an $n \times n$ lower triangular matrix. Consider reducing it to upper triangular form using Givens rotations as described in lecture. Determine how many flops are required, accurate to the leading term. Be sure to indicate the order in which the rotations are to be applied, and which entries change from nonzero to zero on each step, and vice versa.

2. In lecture it was claimed that the Givens and Householder algorithms both yield a more accurate answer than the method of normal equations when A is ill-conditioned. Test this claim with a computational experiment as follows.

   First, construct an $m \times n$ matrix A, $m > n$, with m and n both reasonably small, such that A is quite ill-conditioned. (You can make such an A by making one column a small

perturbation of a linear combination of the other column(s).) Make up a nonzero m-vector $\mathbf{b}$ for the right-hand side. Solve this using Matlab's built-in least-squares solver described in lecture and in the book. Then solve it using the method of normal equations, which in Matlab is like this `x=(A'*A)\(A'*b)`. Finally, compare both solutions to the exact solution. You can get the exact solution in Matlab by switching to symbolic arithmetic in the symbolic toolbox. For example, to make a symbolic version of A, use the statement `A2=sym(A)`. (This statement works only on a machine where the symbolic toolbox is installed.) You can then solve the normal equations with symbolic entities using the same statement as above. Finally, you can convert a symbolic object to ordinary double precision by applying the `double` operation to it. Carry out a study that relates the condition number of A (as measured by `cond`) to the accuracy of both Matlab's built-in solver and the method of normal equations. Hand in a trace of your Matlab experiments, including copies of any scripts or functions that you wrote and a relevant plot. Also hand in a paragraph of conclusions.

3. Consider approximating e^x over $[-.5, .5]$ using a degree-8 polynomial. One approach would be the Taylor series expansion $1 + x + x^2/2 + \cdots$. Another would be to fit the best polynomial in a least-squares sense. For instance, e^x could be evaluated at 20 evenly-spaced points in $[-.5, .5]$, and then a degree-8 polynomial could be fit to those data points using Matlab's `polyfit` function. Try both approaches, and see which one yields a more accurate fit. To measure accuracy, try to estimate the maximum difference between the fitted polynomial and the true value of e^x over the interval $[-.5, .5]$. Hand in listings of m-files you write and estimates of the two errors.