CS 322: Introduction to Scientific Computing
Spring 2003
Problem Set 3

Handed out: Wed., Mar. 5.

Due: Wed., Mar. 12 in lecture.

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

- You should hand in your on-time problem set at the beginning or end of lecture on the day it is due in the box at the front of the room. 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 on Mar. 13. 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 names and section number (like this: "Section 2") at the top of the front page of your paper. 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. Sometimes it is necessary to integrate a function over an infinite interval, say $[0, \infty)$. One strategy is to apply a change of variables, say $y = x/(x+1)$, so that the interval is reduced to a finite length. Then a quadrature rule can be used on that finite interval. Implement this strategy in Matlab for approximating

$$\int_0^\infty \exp(-x^2)\, dx$$

(In other words, first, on paper, figure out a change of variables that transforms the infinite interval to a finite interval, and then analytically substitute the change of variables before beginning any computation). Once the interval is finite, use the `quad` function in matlab to evaluate the integral. Try various tolerances. How accurate is this method? Note that the exact value of this integral is $\sqrt{\pi}/2$. Does the tolerance that is input to `quad` correspond to the actual accuracy (that is, the difference between the output of `quad` and the true solution)?

Hand in a listing of m-files that you wrote, and comparison of the tolerance to the actual accuracy.

2. Run the `quad` routine on the function $\sqrt{x}$ integrated over the interval [0,1] accurate to 10^{-8}. Turn on tracing. (Type `help quad` for information on tracing.) Then make a plot showing the function $\sqrt{x}$ over $[0, 1]$ and asterisks on the square-root curve indicating the interval endpoints used by quad at the leaves of the recursion. In other words, on top of a plot of the square-root curve, plot $(a, \sqrt{a})$ for each a appearing in the trace, where a is the left endpoint of an interval and is the second entry of each row printed out by the trace. One way to capture the output of the trace of quad is to use copy-and-paste in the Matlab command window. Copy the numbers output by `quad` and paste them back in the command window like this:

```
>> r = [
 (paste numbers here)
];
```

The plot should show a clustering at $(0, 0)$. Hand in your plot and a description, including traces, of how you generated it. Explain why this clustering happens, based on the discussion in lecture on adaptive quadrature.

3. A "skyline" $n \times n$ upper triangular matrix is one in which the nonzero entries above the diagonal occur in consecutive vertical positions (possibly some 0's mixed in), but with all zeros above those entries. The "height" of the skyline is a vector of n integers. Its ith entry is the highest row index of a nonzero number in column i. For example, here is a 7×7 skyline upper triangular matrix and its height vector:

$$U = \begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 & -5 \\ & 1 & 0 & 2 & 0 & 0 & 7 \\ & & 3 & -1 & 0 & 0 & -2 \\ & & & 4 & 0 & 6 & 3 \\ & & & & 9 & 2 & 0 \\ & & & & & 3 & 1 \\ & & & & & & 2 \end{pmatrix}, \quad h = [1, 1, 3, 2, 5, 4, 1].$$

(a) Describe (on paper) a scheme for storing a skyline upper triangular matrix using Matlab vectors so that storage space is not wasted, i.e., there is no space used by the zero entries outside the skyline in your scheme.

(b) Write a vectorized m-file that carries out back-substitution efficiently to solve $U\mathbf{x} = \mathbf{b}$ when U is a skyline upper triangular matrix. The m-file should take as input the vector(s) describing matrix as defined in part (a), as well as a vector holding the right-hand side. It should return the solution vector $\mathbf{x}$. Try your m-file on the above matrix, with right-hand side $(1 : 7)'$.