CS 421: Numerical Analysis
Fall 2002
Problem Set 2

Handed out: Wed., Sep. 25.

Due: Fri., Oct. 4 in lecture.

1. Let U be an $n \times n$ nonsingular upper triangular matrix. (a) Show that $\|U^{-1}\|_\infty \geq 1/\min_i |U(i,i)|$. This fact leads to a simple but not very reliable condition-number estimator (namely, $\|U^{-1}\|_\infty \approx 1/\min_i |U(i,i)|$) for upper triangular matrices. (b) In fact, show that this estimator is not reliable by constructing a 2×2 upper triangular matrix U in which $\|U^{-1}\|_\infty \geq 10^8/\min_i |U(i,i)|$.

2. Consider solving $A\mathbf{x} = \mathbf{b}$, let $\mathbf{x}$ be the exact solution and let $\mathbf{x}_1$ be an approximate solution. Let $\mathbf{r}_1$ be the residual of $\mathbf{x}_1$, that is, $\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1$.

   (a) Derive two inequalities, the first of the form $\|\mathbf{x} - \mathbf{x}_1\| \leq X \cdot \|\mathbf{r}_1\|$ and the second of the form $\|\mathbf{r}_1\| \leq Y \cdot \|\mathbf{x} - \mathbf{x}_1\|$. Here, X, Y are two scalars that both depend on A.

   (b) Confirm in Matlab that these two inequalities are both satisfied for both $\mathbf{x}_1$ and $\mathbf{x}_2$ in Example 2.8 on p. 63 of the text.

   (c) [**Note correction to this question 9/27/02.**] An example of this kind (like 2.8) is possible only if A is fairly ill-conditioned. Derive an inequality that explain why. [Hint: come up with an upper bound on

$$\frac{\|\mathbf{x}_2 - \mathbf{x}\|/\|\mathbf{x}_1 - \mathbf{x}\|}{\|\mathbf{r}_2\|/\|\mathbf{r}_1\|}.]$$

3. Let A be a symmetric positive *semidefinite* matrix.

   (a) Show that $A(1,1)$ must be nonnegative.

   (b) Show that if $A(1,1) = 0$, then the whole first row and column of A must be all zeros.

   These two facts play a role in an efficient algorithm for testing whether a matrix is positive semidefinite.

4. Write a Matlab function `invlower` that computes L^{-1} given a lower triangular matrix L by applying backsubstitution to the columns of the identity matrix. Make sure the inner loop is vectorized, and make sure that unnecessary operations on 0's are omitted.

   Then write an m-file called `mycond` that computes the condition number of a lower triangular matrix by multiplying its norm (the matrix 2-norm, which is computed by `norm`) in matlab by the norm of the inverse as computed by `invlower`. Compare this to the builtin cond function. They should nearly identical answers for reasonably well-conditioned matrices, e.g., the matrix returned by `tril(randn(10,10))`. Which seems to be more accurate for extremely ill-conditioned lower triangular matrices? You can make a lower triangular matrix ill-conditioned by putting a number very close to 0 (say 1e-40) on the main diagonal, or by putting a very big number in an off-diagonal position, or both. You can get some idea of which routine (`cond` vs `mycond`) is more accurate by checking whether the inequalities of question 1 are satisfied by the results. Hand in listings of all m-files, some sample runs, and a paragraph of conclusions.