

CS 421: Numerical Analysis  
Fall 2005  
**Problem Set 2**

Handed out: Wed., Sep. 21.

Due: Fri., Sep. 30 in lecture.

1. Consider a program to evaluate

$$F(a, b) = \sqrt{a^2 + b^2} - |a|.$$

Implementing this formula directly in Matlab (i.e., as `sqrt(a^2+b^2)-abs(a)`) is prone to overflow (e.g., in the case  $a, b \approx 10^{180}$ ), underflow (e.g., in the case  $a, b \approx 10^{-180}$ ), and severe cancellation (e.g., in the case  $|a| \gg |b|$ ). Write (on paper) a matlab program to evaluate  $F$  that should be more robust against overflow, underflow and cancellation than the direct implementation. It is OK if your program needs some **if** statements.

[Hint: first get rid of the cancellation error by multiplying and dividing by the same factor to come up with a mathematically equivalent formula. See, e.g., p. 26 of the text. Then deal with the overflow problem by scaling and then unscaling entries under the square root sign. See, e.g., p. 128 of the text.]

2. Let  $U$  be an  $n \times n$  nonsingular upper triangular matrix, and suppose we wish to know its condition number in the 1-norm.

(a) To compute the condition number exactly, we need to compute  $U^{-1}$ . Write down (on paper) an efficient algorithm for computing  $U^{-1}$ . How many flops (accurate to the leading term) are required? It is suggested that your algorithm carry out backsubstitution using  $U$  on the columns of  $I$ . Note that many flops can be saved because of special structure—i.e., avoid unnecessary operations on 0's. For example, since  $U^{-1}$  is upper triangular, there is no need to compute the entries below the main diagonal.

(b) For many applications, it is adequate to have merely an estimate of the condition number, and in particular, to estimate  $\|U^{-1}\|_\infty$ . (Note that computation of  $\|U\|_\infty$  requires only  $O(n^2)$  flops, so this can be done exactly for a low cost.) One way to come up with such an estimate is to generate a random nonzero vector  $\mathbf{x}$ , solve  $U\mathbf{y} = \mathbf{x}$  via backsubstitution, and then estimate  $\|U^{-1}\|_\infty \approx \|\mathbf{y}\|_\infty / \|\mathbf{x}\|_\infty$ . Explain how this estimate relates to the true value of  $\|U^{-1}\|_\infty$  (i.e., what inequality is satisfied).

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 two Matlab functions, one that computes the condition number in the  $\infty$  norm of a matrix exactly using the technique of 2(a) and one that approximates it using the max over five random vectors as in 2(b). Compare your program with built-in `cond` and `condest`. (Note that `condest` computes the 1-norm, so you need to transpose to lower-triangular before using `condest`.) Test all four routines on random upper triangular matrices of various condition numbers. Note: you can make an arbitrarily ill conditioned upper triangular matrix by either setting the diagonal entries to be widely varying in magnitude or else by including entries above the diagonal that have much larger magnitude than diagonal entries (or both).

Hand in listings of all m-files, a relevant table or plot, and a paragraph explaining how well your condition number estimator works compared to Matlab's.