CS 621: Matrix Computations
Fall 2001
Problem Set 4

Handed out: Mon., Oct. 22.

Due: Wed., Oct. 31 in lecture.

1. Let A be an $m \times n$ matrix of rank n, and let $\hat{A}$ be the matrix
$$\hat{A} = \begin{pmatrix} 0 \\ A \end{pmatrix}$$
where the block of zeros is $n \times n$. Show that Householder's QR factorization algorithm applied to $\hat{A}$ performs the same sequence of operations on the last m rows of $\hat{A}$ that MGS would perform on the original A, except possibly for signs. [This fact is used to analyze the stability of the MGS algorithm – see e.g. Higham's or Björck's recent books.]

2. Learn about the trick that avoids the recomputation of column norms in QR factorization with column pivoting, for instance, by reading pp. 249-250 of GVL. Explain why this trick has a problem with accuracy in the presence of roundoff when $\|\mathbf{z}^{(j)}\|_2 \ll \|\mathbf{z}^{(j-1)}\|_2$. For example, your analysis should show that there will be an expected relative error of about 100% in the computed value of $\|\mathbf{z}^{(j)}\|_2$ if $\|\mathbf{z}^{(j)}\|_2 \leq 10^{-8}\|\mathbf{z}^{(j-1)}\|_2$ when IEEE double-precision arithmetic is used.

3. Develop a formula for a (complex) unitary Givens rotation that has the ability to zero out any one entry of a given complex matrix. Hint: the complex Givens rotation is similar to the real Givens rotation except that it has some complex-conjugate symbols in key locations.

4. Although we have treated backward error analysis as primarily a theoretical tool, the backward error can be explicitly computed for many problems. Write m-files that carry out Gaussian elimination without pivoting, with partial pivoting and with complete pivoting using simulated single precision (24 bits). Write also forward and back substitution in 24-bit arithmetic. Apply all three versions of GE to solve a linear system $A\mathbf{x} = \mathbf{b}$. Then, for the computed solution $\hat{\mathbf{x}}$, find the smallest matrix E (in the Frobenius norm) such that $(A + E)\hat{\mathbf{x}} = \mathbf{b}$ by solving a least squares problem (in usual double precision). Determine the backward error for all three algorithms. Try values of n up to 30. Can you construct examples where the backward error for plain GE is much worse than for the other two? (Hint: if any upper left $k \times k$ submatrix of A is nearly singular, then plain GE will be expected to have a much larger backward error than the other two.) Hand in listings of all m-files, some printouts and some conclusions.

   (a) The routine `rnd24` on the class home page will probably be helpful for this project.

(b) This project requires considerably more programming than previous assignments. Be sure to follow guidelines for good code such as documenting your m-files with comments, breaking up your code into sensible routines, etc.

(c) Vectorize your code. Hint: vectorizing the forward substitution can be done either as saxpy or as inner product. Because the rnd24 operation must be invoked after every arithmetic operation, only one of these two (saxpy or inner product) can be vectorized. Which one? Write your code accordingly.

(d) You must use QR factorization for the underdetermined least squares problem. Implement your own underdetermined least squares solver using the `qr` function for QR factorization.

(e) Note that there is a possibility for notational confusion because the coefficient matrix for the undetermined least squares problem is not the same "A" as in the question. What is the coefficient matrix to find E? Hint: this matrix is $n \times n^2$ in size.