CS 421: Numerical Analysis
Fall 2000
Problem Set 3

Handed out: Mon., Oct. 2.

Due: Mon., Oct. 16.

1. Let L_1 and L_2 be two lines in $\mathbf{R}^n$. Assume that L_1 is written in "parametric form" as $L_1 = \{\mathbf{v}_1 + t_1\mathbf{w}_1 : t_1 \in \mathbf{R}\}$ where $\mathbf{v}_1, \mathbf{w}_1$ are given, and assume L_2 has the analogous form. Assume further that the lines are not parallel, i.e., $\mathbf{w}_1, \mathbf{w}_2$ are linearly independent. Consider the problem of finding the closest pair of points on L_1 and L_2. Show that this is a least-squares problem, and develop an algorithm for solving it.

2. Suppose that $A \in \mathbf{R}^{m\times n}$ has rank n. The *orthogonal projection* onto the rangespace of A is defined to be the matrix $P = A(A^TA)^{-1}A^T$.

   (a) Suppose A is factored as QR. Write a formula for P in terms of Q and R. By simplifying the formula, show that R is unneeded, and that P can be written in terms of Q alone.

   (b) Given the factorization $A = QR$ where Q is represented implicitly as a product of Householder reflections, propose an algorithm to compute $A(A^TA)^{-1}A^T\mathbf{x}$ for an arbitrary vector $\mathbf{x}$ using the Householder reflections (i.e., without explicitly forming Q). How many flops are required for your algorithm, accurate to the leading term (not counting the flops for factorization)?

3. Let A be a square nonsingular matrix QR-factored as $A = QR$. Prove that $\|Q\|_F \cdot \|R\|_F$ is not much larger than $\|A\|_F$.

4. Implement classical Gram-Schmidt for solving least-squares problems. Use BLAS level 2 where possible. Compare it to the method of normal equations. For these experiments, assume that Matlab's built-in least-squares solver (i.e., the backslash operator) returns the "exact answer." How do CGS and normal equations compare in terms of accuracy?

   Try all three algorithms for randomly-generated problems of size 60×40 for varying condition numbers, where the condition number varies from 1 to 10^{16}. In class it will be explained how to generate a random matrix with known condition number.

   Hand in: listings of all m-files, sample runs (if relevant) and at least one interesting plot showing how the errors in the CGS versus normal equations behave as the condition is varied.