COT 4501: Fall 2009 Midterm

Tuesday, October 13, 2009. This is a 115 minute exam with 8 problems on 8 pages. No computers. Calculators are OK. If you write on the back of any page, please write "OVER $\Longrightarrow$" on the front of the page. If you remove the staple, please write your name on each page.

Name (please print. No ID numbers): ______________________________

Problem 1 (10 points)

Consider the Householder reflection matrix $H - \rho uu^T$. If x is a column vector, what is the most efficient way to compute Hx? What is the most efficient way to compute $x^T H$?

Solution:

- $Hx = (I - \rho uu^T)x = x - u(\rho(u^T x))$
- $x^T H = x^T(I - \rho uu^T) = x^T - (\rho(x^T u))u^T$

Problem 2 (10 points)

Define *backward error* and *forward error* and describe how they are different.

Solution: Let $y = f(x)$ be the quantity being computed. Forward error is the difference between the true solution y and the approximate solution $\hat{y}$. For the backward error, pretend that the approximate solution $\hat{y}$ is the exact solution to some "mangled" or "nearby" problem $\hat{x}$. The backward error is then the difference between the true problem x and the mangled problem $\hat{x}$.

One way of describing the difference between the two is to consider an ill-conditioned problem. If a problem is ill-conditioned, the forward error can be very high while the backward error is at the same time very low. That is, you solved nearly the right problem, but got a very different answer.

Another way of describing the difference is to draw the diagram on page 12 of the book, or to give an example. Some kind of compare/contrast should be given here; there are many possible ways of discussing the difference.

Problem 3 (10 points)

Define *relative error* and *absolute error* and describe how they are different.

Solution:

The absolute error between the true value x and its approximation $\hat{x}$ is the difference $|\hat{x} - x|$ (you can leave out the absolute value, or put them in; either definition is fine). The relative error is the absolute error divided by the true value x, or

$$\frac{|\hat{x} - x|}{x}.$$

There are many ways of discussing the difference between the two.

The relative error is a better indicator of the accuracy of $\hat{x}$ (unless $x = 0$).

Another way of describing their difference is with an example. Suppose the true answer is 10^{-10} but you get the approximate result 10^{-8}. The absolute error looks small (about 0.99×10^{-8}), but the relative error is huge, about 99. So in this case, $\hat{x}$ is an awful approximation to x.

Another example: the absolute error can be huge but the relative error can be small (suppose $x = 100,000$ but $\hat{x} = 100,001$). The absolute error is 1, which seems large, but the relative error is just 10^{-5}.

Another thing that can be stated: the number of decimal digits of accuracy in the result is roughly the $\log_{10}$ of the relative error. The same cannot be said of the absolute error.

Problem 4 (10 points)

True or False (and state your reasons). If A is any n-by-n matrix and P is any n-by-n permutation matrix, then $PA = AP$ always holds.

Solution:

This is false. There are many examples. For example:

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Then if A is a 2-by-2 matrix,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

The product PA swaps the two rows of A:

$$PA = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}$$

but the product AP swaps the two columns of A:

$$AP = \begin{bmatrix} 2 & 1 \\ 4 & 3 \end{bmatrix}$$

Problem 5 (10 points)

True or False (and state your reasons). If A is non-singular, then the condition number of A is always the same as the condition number of A^{-1}.

Solution: This is true. The condition number of a matrix is $\text{cond}(A) = ||A|| \times ||A^{-1}||$. If we let $C = A^{-1}$, then $\text{cond}(C) = ||C|| \times ||C^{-1}|| = ||A^{-1}|| \times ||A|| = \text{cond}(A)$.

Problem 6 (10 points)

Suppose that two sides of a system of linear equations $Ax = b$ are multiplied by a nonzero scalar α.

1. Does this change the true solution x?

   Solution: no.

2. Does this change the residual vector $r = b - Ax$ for a given x? If so, how?

   Solution: yes. The residual becomes $r = \alpha(b - Ax)$. The norm of r can change arbitrarily, depending on α.

3. What conclusions can be drawn from r about assessing the quality of a computed solution x?

   Solution: The short (and sufficient) answer is "none." If the residual is small, it could just be due to a tiny α. If the residual is huge, it could just be due to a large α. In either case, you can't say much about the accuracy of x.

Problem 7 (20 points)

Consider the least squares problem $Ax \approx b$, and the QR factorization

$$A = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}.$$

The solution is found by solving $Q_1^T b - R_1 x = 0$ for x. Recall that $A^T = R^T Q^T$. The residual is $r = b - Ax$.

Use these facts to prove that $A^T r = 0$.

Solution:

$$\begin{aligned}
A^T r &= A^T(b - Ax) \\
&= R^T Q^T (b - QRx) \\
&= R^T Q^T b - R^T Q^T QRx \\
&= R^T Q^T b - R^T Rx \\
&= R^T (Q^T b - Rx) \\
&= \begin{bmatrix} R_1^T & 0 \end{bmatrix} \begin{bmatrix} Q_1^T b - R_1 x \\ Q_2^T b \end{bmatrix} \\
&= R_1^T (Q_1^T b - R_1 x) + (0)(Q_2^T b) \\
&= R_1^T (0) + (0) \\
&= 0
\end{aligned}$$

Problem 8 (20 points)

Consider the linear system $Ax = b$, a computed solution $\overline{x}$, and the computed residual $r = b - A\overline{x}$. Let $x = \overline{x} + \Delta x$ be the true solution, where Δx is the (unknown) error in the computed solution.

State and derive the algorithm for *iterative refinement*, which can improve the solution to this problem. Use MATLAB notation to state the algorithm, and note where the factorization of A can be reused. You do not need to show exactly how to reuse the factorization, just where it would be reused.

Solution: $Ax = A(\overline{x} + \Delta x) = A\overline{x} + A\Delta x = b$.

Thus, $A\Delta x = b - A\overline{x} = r$. We can solve this for Δx.

Algorithm:

```
x = A\b ;           % note: save the LU factorization here
r = b - A*x ;
dx = A\r ;          % note: reuse the LU factorization here
x = x + dx ;
```

This can be rewritten so that the statement `dx=A\r` can reuse the factorization computed when `x=A\b` is solved.