Homework #2
Due Friday, July 18, 2003

Walking The Straight Line

Answer all questions. Submit any pertinent hand calculations and/or Matlab code in a neat manner in order to help us give you partial credit. Readings: Van Loan 5.2.5, 6.

1) Consider the following matrix:

$$A = \begin{pmatrix} 1 & 3 & 2 \\ 2 & 15 & 5 \\ -2 & 39 & 3 \end{pmatrix}.$$

(a) Calculate the LU decomposition of A without any pivoting. That is, calculate

$$M_2 M_1 A = U,\ L = M_1^{-1} M_2^{-1},$$

where M_1 and M_2 are elementary elimination matrices as given in section 6.3.2 of the text, and L and U are upper and lower triangular matrices, respectively. Show how you arrive at your M_1, M_2, U and L. (b) Calculate the LU decomposition with partial pivoting. That is, calculate

$$M_2 P_2 M_1 P_1 A = U$$

or, equivalently,

$$PA = LU.$$

Write out M_1, M_2, P_1,P_2, P, L and U explicitly. Input A into Matlab and ask it to compute the LU decomposition using the built-in Matlab function lu: $[L,\, U] = lu(A)$;. Compare your hand-caculations of L and U to these, if the results are different explain the differences. Now compute L and U again, this time using the following syntax: $[L,\, U,\, P] = lu(A)$;. Does this result agree with your hand-calculations? Explain any differences. (c) The theory says that if you consider $A\mathbf{x} = \mathbf{b}$ with the equations reordered as given by $PA\mathbf{x} = P\mathbf{b}$, where $P = P_2 P_1$ and P_1 and $P_2$are from part (b), one can perform Guassian elimination without pivoting and still have all the elimination matrices with entries of magnitude less than or equal to 1. Show that this is true in this example.

2) Assume that A and C are given $n - by - n$ matrices and that A is non-singular. Assume that $\mathbf{g}$ and $\mathbf{h}$ are given $n - by - 1$ vectors. Consider the system

$$\begin{aligned} A^T\mathbf{y} + C\mathbf{z} &= \mathbf{g} \\ A\mathbf{z} &= \mathbf{h} \end{aligned}.$$

Show how to solve the system for $\mathbf{z}$ and $\mathbf{y}$ using two forward substitutions, two back substitutions and one LU decomposition. You may assume that pivoting is not needed anywhere.

3) Consider the $n - by - n$ Hilbert matrix H with entries $H_{ij} = 1/(i + j - 1)$, $i, j = 1, 2, \ldots, n$. Note that Matlab has a function that constructs such matrices. Just type $H = hilb(n)$. Suppose that we wish to solve $H\mathbf{x} = \mathbf{b}$, where $\mathbf{b} = \mathbf{e}_1$, the first unit vector in R^n. Write a Matlab program that performs the following study:

First note that H is symmetric positive definite, which means that a Cholesky factorization exists. (a) Compute the Cholesky factorization $H = U^T U$, where U is upper triangular. In Matlab, just type $U = chol(H)$. (b) Solve for x by forward and back substitution. Let's call this solution $\tilde{\mathbf{x}}$. Do this for $n = 2, 3, \ldots, 14$. (c) For the purpose of camparison, calculate the true solution $\mathbf{x} = H^{-1}\mathbf{b}$. Note that Matlab calculates exactly the inverse of H, H^{-1}. To get it, type $invhilb(n)$. (d) Plot the ∞-norm of the residual, $\|\mathbf{b} - A\tilde{\mathbf{x}}\|_\infty$, versus matrix size, n, for $n = 2, 3, \ldots, 14$. (e) Plot the relative error, $\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|_\infty}{\|\mathbf{x}\|_\infty}$, versus matrix size, n, for $n = 2, 3, \ldots, 14$. How large is n when the relative error is bigger than 1? (f) Plot the condition number of H versus matrix size, n, for $n = 2, 3, \ldots, 14$. To get the condition number of a matrix A in Matlab type $cond(A)$. Comment on the trend showed in this plot. (g) Note how big the condition number of H is when the relative error is bigger than 1.

Revision: Matlab may have trouble computing the *Cholesky* factorization of our *Hilbert* matrices when n gets bigger than a certain value and this is machine dependent; my machine went up to $n = 14$, while at least one student in our class could go up to only $n = 12$. To circumvent this problem, go up to the n when Matlab fails for you. In part (e), instead of making comparisons of the relative error against 1 compare against the largest observable relative error and denote this number by RE. Now in part (g) note how big the condition number of H is when the relative error is bigger than RE.