COMP 540 Matrix Computations

Assignment #2: Solving Linear Systems.
Date Given: Monday, 21 September. **Date Due:** Monday, 5 October, 2015.

1. (5 points) (**Cholesky factorization with pivoting**). In some problems we have singular or nearly singular symmetric nonnegative denite matrices ($A \in \mathbb{R}^{n\times n}$ is said to be symmetric nonnegative denite if $A^T = A$ and $x^T Ax \geq 0$ for any $x \in \mathbb{R}^n$). In this case, we need to incorporate the so called symmetric pivoting (or diagonal pivoting) strategy into the Cholesky factorization (outer-product formulation). At each step $k = 1, 2, \ldots, r$ where $r = \text{rank}(A)$, we search the remaining (altered) diagonal (i.e., $a_{kk}^{(k)}, ..., a_{nn}^{(k)}$) for the largest element, say $a_{mm}^{(k)}$, and exchange rows (and columns) k and m. Briey outline the algorithm and show that the algorithm produces a permutation matrix $P \in \mathbb{R}^{n\times n}$ and a lower triangular matrix $L \in \mathbb{R}^{n\times r}$ (i.e., $l_{ij} = 0$ for $i < j$) with positive diagonal entries such that $P^T AP = LL^T$. In practice when $\max_{k\leq i\leq n} a_{ii}^{(k)} \leq \epsilon \max_{1\leq i\leq n} a_{ii}^{(1)}$, where $\epsilon = nu$, the algorithm terminates and the rank r is determined to be $k - 1$.

2. (15 points + 5 bonus points) (**Ill conditioned problem**). Program in single precision the Cholesky factorization $A = LL^T$ and solution of equations for a symmetric positive definite matrix A. If your program is about to divide by zero, or take the square root of a negative number, it should instead output an appropriate message and stop. Write each component in subroutine or procedure forms and use them in the solution of systems $Ax = b$, where A are $n \times n$ Hilbert Matrices.

$$A = \left(\frac{1}{i + j - 1}\right), \qquad b = Ae \;\text{ computed,}$$

   for $n = 2, 4, 6, 8$. Note that $e^T = [1, 1, \ldots, 1]$. Let the computed solution be denoted by x_c, compute

   i. The relative error in the solution, $\|x_c - e\|_2/\|x_c\|_2$.
   ii. The relative residual, $\|b - Ax_c\|_2/(\|A\|_F\|x_c\|_2)$.
   iii. The relative matrix residual, $\|A - L_cL_c^T\|_F/\|A\|_F$.

   Comment on and justify these results (hint: using the rounding error analysis results and perturbation analysis results to justify i and using the rounding error analysis results to justify iii). Print error and residual norms in floating point format so you can see what is happening. Note that James Wilkinson (1968) showed that Cholesky factorization runs to completion if $20n^{3/2}u\kappa_2(A) \leq 1$ (an improved condition was given by James Demmel (1989)). The condition numbers $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2$ for Hilbert matrices with $n = 2, 4, 6, 8$ are about 19, 16,000, 15,000,000 and 15,000,000,000 respectively.

   If you use MATLAB, find out how to use single precision. If you use a high level language rather than a MATLAB-like language, you can get up to 5 bonus points.

   Please submit your code with a readme file through myCourses, which explains how to run your code.

3. (a) (Bonus, 5 points) Show in GEPP the growth factor $\rho \leq 2^{n-1}$ and the upper bound can be reached for the following $n \times n$ matrix

$$A_0 = \begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ \vdots & \vdots & \ddots & & 1 \\ -1 & -1 & \cdots & 1 & 1 \\ -1 & -1 & \cdots & -1 & 1 \end{bmatrix}$$

(b) (5 points) Write a MATLAB program to solve a general linear system $Ax = b$ by the LU factorization with partial pivoting. Your program should include the computation of the growth factor.

(c) (10 points) Use your code to solve $n \times n$ $Ax = b$, where $A = A_0 + 10^{-8} * I$ with A_0 defined in 3(a), $b = A * e$, $e \equiv [1, 1, \ldots, 1]^T$, and $n = 30$. So $x = e$ is the exact solution. Suppose your computed solution is x_c, compute $\|x - x_c\|_\infty / \|x_c\|_\infty$.

Note: The reason we consider solving $Ax = b$ is that if we solve $A_0 x = b$, we may not see any rounding errors because the elements of A are integers, so cannot see how the growth factor affect the accuracy of the computed solution.

(d) (5 points) Iterative refinement is an established technique for improving a computed solution x_c to a linear system $Ax = b$. The process consists of 3 steps:

1 Compute $r = b - Ax_c$.

2 Solve $Ad = r$.

3 Update $y = x_c + d$.
(Repeat from step 1 if necessary with x_c replaced by y).

If there were no rounding errors in the computation of r, d and y, then y would be the exact solution to the system. The idea behind iterative refinement is that if r and d are computed accurately enough then some improvement in the accuracy of the solution will be obtained. Notice when we solve $Ad = r$ we can reuse the LU factorization which was used in computing x_c. So iterative refinement is economic if only a few number of iterative steps are required.

Now use the iterative refinement technique to refine your computed solution x_c obtained in 3(c). Your iteration should stop when $\|d\|_2 / \|x_c\|_2 \leq \tau$, where τ is a tolerance and try $\tau = 10^{-15}$ in your computation, and x_c is the latest computed solution. Compute the relative error $\|x_c - x\|_\infty / \|x_c\|_\infty$ for your final solution, and report the number of iterative steps.