Homework 3: Due October 23rd, 2017

First, do these problems. These will be graded for completion.

- 1.5: 4, 11, 12
- 1.7: 10 b,c, 34
- Write a function in MATLAB that takes as input a tridiagonal matrix given as three vectors: an $n \times 1$ vector $v$ representing the main diagonal, an $(n - 1) \times 1$ vector $w$ representing the upper diagonal, and a $(n - 1) \times 1$ vector $z$ representing the lower diagonal. Have this function output the LU factorization (without pivoting) with the $U$ as two vectors and the $L$ as one vector representing the nontrivial diagonals. Use only basic programming.
- 1.8: 4, 9, 10 (the only part you need to submit is the equation relating $L$ with $K$.)

Then do these problems. These will be graded for correctness.

1. Write a function in MATLAB that takes as input a symmetric tridiagonal matrix represented as two vectors: an $1 \times n$ vector $v$ representing the main diagonal and an $(n - 1) \times 1$ vector $w$ representing the upper (and lower) diagonal. Have this function output the Cholesky factor of the matrix as a vector for the main diagonal and a vector for the upper diagonal. Only submit the code, but to make sure your code works, if you input $v = [4, 4, 4, 4, 4]$ and $w = [1, 1, 1, 1]$, your program should output the vectors $[2, 1.94, 1.93, 1.93, 1.93, 1.93]$ and $[.5, .516, .518, .518, .518]$. For big matrices, this algorithm is more efficient than the standard Cholesky method. For a $100 \times 100$ matrix, what percentage of the memory do we need to store our matrix, compared to the naive method? About what percentage of flops do we compute, compared to the naive method? (Use the approximations for flops we gave in class.)

2. For sparse matrices (matrices with mostly zero entries, such as banded matrices), explain why trying to calculate $A^{-1}$ in order to solve $Ax = b$ is even more inefficient (and thus foolish) than usual. (As compared to, say, using the Cholesky decomposition.) (Hint: Look at the paragraph above Exercise 1.5.12.)

3. Write a function in MATLAB that takes as input an $n \times 1$ vector $p$ of rearranged integers from 1 to $n$ representing a permutation matrix $P$ whose $i$-th row is the $p(i)$-th row of the identity matrix; an $n \times n$ matrix $B$ whose upper triangular portion stores $U$ and strictly lower triangular portion stores $L$ of the $LU$ factorization of the matrix $PA$; and an $n \times 1$ vector $b$. Have this function output the solution to $Ax = b$. Use only basic programming. Only submit the code, but to make sure your code works, if you input $p = [3; 1; 2]$, $B = [2, -1, 3; 0.4, -3, 3; 0.5, -0.2, 4]$ and $b = [2; -1; 1]$, you should output $x = [0.53; -0.83; -0.30]$.

4. Explain why pivoting is necessary for Gaussian elimination to work on a computer. (There are two reasons.)

For additional practice, here are some optional problems. These should not be turned in.

- 1.5: 9, 12, 14
- 1.6: 5
- 1.7: 18, 35, 36, 37