Homework 5. Due Wednesday, Oct. 2.

1. Calculate the number of flops required for computing a matrix inverse as a function of n where $n \times n$ is the size of the matrix. Consider two algorithms.

   (a) **(5 pts) Algorithm 1.** Define an $n \times 2n$ matrix $M := (A, I)$ where I is the $n \times n$ identity matrix. Subject M to row operations to transform it into a matrix of the form (I, B). Then $B = A^{-1}$. Write this algorithm as a pseudocode. For simplicity assume that pivoting is not needed (anyway, row swaps do not involve flops). Calculate the number of flops. An answer of the form $W(n) = Cn^p + O(n^{p-1})$ is good enough. You need to determine the constants C and p.

   (b) **(5 pts) Algorithm 2.** Decompose A to $A = LU$. The cost of this is $W_1(n) = \frac{2}{3}n^3 + O(n^2)$. Compute L^{-1} and U^{-1} and calculate $A^{-1} = U^{-1}L^{-1}$. Write this algorithm as a pseudocode. For simplicity assume that pivoting is not needed. Start it with calling the LU algorithm (you do not need to write a pseudocode for LU, just add its cost to your result). Calculate the number of flops. An answer should be of the form $W(n) = Cn^p + O(n^{p-1})$. You need to determine the constants C and p.

2. (a) **(4 pts)** Consider the set $\mathcal{L}$ of all $n \times n$ lower-triangular matrices with positive diagonal entries.

   i. Prove that the product of any two matrices in $\mathcal{L}$ is also in $\mathcal{L}$.

   ii. Prove that the inverse of any matrix in $\mathcal{L}$ is also in $\mathcal{L}$.

   This means that the set of all $n \times n$ lower-triangular matrices with positive diagonal entries forms a group with respect to matrix multiplication.

   (b) **(2 pts)** Prove that the Cholesky decomposition for any $n \times n$ symmetric positive definite matrix is unique. *Hint. Proceed from converse. Assume that there are two Cholesky decompositions* $A = LL^\top$ *and* $A = MM^\top$. *Show that then* $M^{-1}LL^\top M^{-\top} = I$. *Conclude that* $M^{-1}L$ *must be orthogonal. Then use item (a) of this problem to complete the argument.*

3. **(5 pts)** The Cholesky algorithm is the cheapest way to check if a symmetric matrix is positive definite.

   (a) Program the Cholesky algorithm. If any L_{jj} turns out to be either complex or zero, make it terminate with a message: "The matrix is not positive definite".

   (b) Generate a symmetric 100×100 matrix as follows: generate a matrix $\tilde{A}$ with entries being random numbers uniformly distributed in $(0, 1)$ and define $A := \tilde{A} + \tilde{A}^\top$. Use the Cholesky algorithm to check if A is symmetric positive definite. Compute the eigenvalues of A using a standard command (e.g. `eig` in MATLAB), find minimal eigenvalue, and check if the conclusion of your Cholesky-based test for positive definiteness is correct. If A is positive definite, compute its Cholesky factor using a standard command (e.g. see this help page for MATLAB) and print the norm of the difference o the Cholesky factors computed by your routine and by the standard one.

(c) Repeat item (b) with A defined by $A = \tilde{A}^\top \tilde{A}$. The point of this task is to check that your Cholesky routine works correctly.