Homework 5. Due Friday Oct. 4 by 10am in my mailbox

Always provide complete, well written solutions.

The solution to **two** of the following problems (of your choice) have to be typed in Latex, while the rest of the solutions can be handwritten.

Always submit printouts of the output of your codes - either as figures, or as tables, or as numbers - and always provide complete description of these outputs. Moreover your matlab codes have to be submitted via ELMS.

1. (a) Count the exact number of flops to solve an upper triangular linear system of size n by backward substitution. What is the asymptotic number of flops as $n \to \infty$?

   (b) The matrix product is associative. What is the most efficient way to perform the matrix product $x^T y z^T$, for $x, y, z \in \mathbb{K}^n$? Justify your choice.

   (c) Verify the formula presented in class for the L factor of the LU decomposition in terms of the multipliers $m_{i,j}$.

2. Ref. [1], Chapter 5, Problem 1 (page 123).

   The solution to $Au = b$ may be written $u = A^{-1}b$. The goal of this exercise is to show that you should NOT write the command `u = inv(A)*b` in Matlab as it is more than twice as expensive to execute as `u = A\ b`. Below you will calculate the computational cost of finding `B = inv(A)`.

   (a) Show that about $(2/3)n^3$ flops reduces $AB = I$ to $UB = L^{-1}$.

   (b) Show that computing the entries of B from $UB = L^{-1}$ by back substitution takes about n^3 flops.

   (c) Use this to verify the claim that computing A^{-1} is more than twice as expensive as solving $Au = b$ by LU factorization.

3. Ref. [1], Chapter 5, Problem 4 (page 124).

   A square matrix A has *bandwidth* $2k+1$ if $a_{jk} = 0$ whenever $|j-k| > k$. A *subdiagonal* or *superdiagonal* is a set of matrix elements on one side of the main diagonal (below for sub, above for super) with $j-k$, the distance to the diagonal, fixed. The bandwidth is the number of nonzero bands. A bandwidth 3 matrix is *tridiagonal*, bandwidth 5 makes *pentadiagonal*, etc.

   (a) Show that a SPD matrix with bandwidth $2k+1$ has a Cholesky factor with nonzeros only on the diagonal and up to k bands below.

   (b) Show that the Cholesky decomposition algorithm computes this L in work proportional to kn^2 (if we skip operations on entries of A outside its nonzero bands).

   (c) Write a procedure for Cholesky factorization of tridiagonal SPD matrices, and apply it to the matrix that is (-1) times the matrix of Exercise 11 from Chapter 4 in [1], compare the running time with matlab's `chol` implementation, and your implementation of the Choleski factorization presented in class. Of course, check that the answer is the same, up to roundoff.

4. Let A be $N \times N$ symmetric matrix. Use Householder matrices to show that there exists an orthogonal matrix Q such that $T := Q^T AQ$ is tridiagonal. *Hint: Let $N \geq 3$. Design a Householder matrix Q_1 such that $Q_1 A$ has all zeros in the first column below row 2. Then show that $A_1 := Q_1 A Q_1^T$ has zeros in the first row after column 2. If $N > 3$, design a Householder matrix Q_2 such that $Q_2 A_1$ has all zeros in column 1 below row 2 and in column 2 below row 3. Set $A_2 := Q_2 A_1 Q_2^T$ and show that it has zeros all zeros in row 1 after column 2 and in row 2 after column 3. And so on. At the end, set $Q := Q_{N-2} \dots Q_2 Q_1$.* You can look up Householder transformations in [2] (Section 3.4.1).

   Remark If an $N \times N$ matrix A is symmetric then there exists an orthogonal matrix V and a diagonal matrix D such that $A = VDV^T$ (the eigenvalue decomposition). However, if $N > 4$, the matrix V in principle cannot be found exactly at a finite number of steps except for some special cases. This is due to the fact that the roots of a polynomial of degree ≥ 5 cannot be expressed as any finite algebraic expression involving the polynomial coefficients. Contrary to this, if the goal is more modest, i.e., to reduce A to a tridiagonal matrix rather than to a diagonal one, it can be always achieved in a finite number of iterations ($\leq N - 2$).

References

[1] Bindel and Goodman, Principles of scientific computing

[2] James Demmel, Applied Numerical Linear Algebra, SIAM 1997 (available online)