Sep 10, 2024 (Due: 08:00 Sep 24, 2024)

1. In the lecture we have shown that many matrix operations, such as matrix–matrix multiplication and LU factorization, share the property that the number of additions/subtractions is equal to the number of of multiplications. Does this property hold for complex matrices, assuming complex arithmetic has to be implemented using real arithmetic? Justify your answer.

2. A real symmetric and positive definite matrix A admits a special variant of the LU factorization (known as the *Cholesky factorization*), $A = LL^\top$, where L is a lower triangular matrix (not necessarily with unit diagonal elements). Design two different algorithms to compute L from A.

3. Let $A \in \mathbb{C}^{n\times n}$ be *strictly diagonally dominant*, i.e.,

$$|a_{ii}| > \sum_{\substack{1\le j\le n\\ j\neq i}} |a_{ij}|$$

for $1 \le i \le n$. Show that after one step of Gaussian elimination, the $(n-1)\times(n-1)$ trailing submatrix (known as the *Schur complement*) is still strictly diagonally dominant.

4. Implement Gaussian elimination with two pivoting strategies—partial pivoting and complete pivoting. Make sure the LU factorization is properly encoded by overwritting the original matrix. Measure the execution time of your program in terms of matrix dimensions and visualize the result by a log–log scale plot.

(optional) Compare the performance of your implementation with a linear solver from a well-developed math library (`linsolve` from MATLAB/Octave, `DGESV` from LAPACK, `numpy.linalg.solve` from NumPy, etc.).

5. You are given $A \in \mathbb{R}^{m\times n}$ and $x \in \mathbb{R}^n$, both already stored in floating-point format. Show that there exists a "small" matrix $E \in \mathbb{R}^{m\times n}$ such that $\mathrm{fl}(Ax) = (A+E)x$. Try to bound the entries of E as tight as you can. You may assume that there is no overflow or (gradual) underflow in the calculation.

6. (H) In the lecture we have mentioned that the IEEE-754 floating-point standard ensures Wilkinson's rounding model if no overflow or (gradual) underflow occurs. It can be shown that complex arithmetic, if implemented properly, also satisfies Wilkinson's rounding model, with a slightly larger machine epsilon. Give an estimate on the machine epsilon for complex arithmetic.

7. (optional) Suppose that you have a BLAS library with an inefficient implementation of `STRSM`. Try to design an efficient `STRSM` subroutine by making use of `SGEMM`. You do not have to really implement it. Just describe your algorithm.

Hint: Search online if you do not know what `STRSM` and `SGEMM` stand for. Useful information can be found at, e.g., `http://www.netlib.org/blas/`.

8. (optional, H) Download OpenBLAS and LAPACK from the internet and compile them on your own computer. Write a program in Fortran/C/C++ to solve a linear system using `DGESV` from LAPACK. Make sure your program correctly links with LAPACK and OpenBLAS libraries you just compiled.