Numerical Mathematics: Homework 2

Due on T.B.D at 24:00pm

Problem 1

Recall (from your linear algebra class) the singular value decomposition of a matrix $A \in \mathbb{R}^{n\times n}$:

$$A = U\Sigma V^T$$

where $U, V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}, \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0.$$

is a diagonal matrix. Find an expression of the condition number $\|A\| \left\|A^{-1}\right\|$ in terms of the singular values. Also, what is the condition number of a unitary matrix? What does this mean from a computational standpoint?

Problem 2

Make a function to solve a lower triangular system using forward substitution problem. Call this function `solve_lowertriangular.m`. It should take in two arguments: the first argument is a square matrix A and the second argument a vector b. The output should be x. Do the same for an upper triangular system. Call this function `solve_uppertriangular.m`.

Obviously, do not use the backslash operator (only can be used to checking purposes). For the output x, it is important to initialize an array of zeros first. Matab allows one to dynamically extend the length of a vector dynamically, but this slows down the code.
Can you give a rough estimate of what the computational complexity is of this algorithm depending on problem size? Use big-O notation.

Problem 3

Make a function that finds the LU factorization of a matrix A. Call this function `find_LU.m`. It should take as argument a matrix A and return two outputs: the first one should be L and the second one is U. If the algorithm encounters a zero pivot, make the function print a message. Obviously you should break the loop, because it makes no sense to continue after that.

Problem 4

Use the code written in problem 1 and problem 2 to close the deal. Write a function called `solve_linearsystem.m` that solves a square linear system. It should take in two arguments: the first argument is a square matrix A and the second argument a vector b. The output should be x. You must call the functions you have made in problems 1 and 2 in your code.

(extra) Problem 5

Repeat problems 3 and 4, but now add partial pivoting to it. Call the new functions `find_LUpartialpivot.m` and `solve_linearsystems_partialpivot.m` respectively. Note that there is NO NEED to explicitly construct a permutation matrix!