Updated on June 28, 2021

Math 3607: Homework 4

Due: 10:00PM, Wednesday, July 7, 2021

TOTAL: 30 points You will be writing several MATLAB functions for this assignment. Include all your functions at the end of your live script.

1. (Improved triangular substitutions; adapted from **FNC** 2.3.5) If $B \in \mathbb{R}^{n\times p}$ has columns $\mathbf{b}_1, \ldots, \mathbf{b}_p$, then we can pose p linear systems at once by writing $AX = B$, where $X \in \mathbb{R}^{n\times p}$ whose jth column $\mathbf{x}_j$ solves $A\mathbf{x}_j = b_j$ for $j = 1, \ldots, p$:

$$A \underbrace{\left[\begin{array}{c|c|c|c} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_p \end{array}\right]}_{=X} = \underbrace{\left[\begin{array}{c|c|c|c} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_p \end{array}\right]}_{=B}.$$

   (a) Modify `backsub.m` and `forelim.m` from pp. 36-7 of Module 2 lecture slides so that they solve the case where the second input is an $n \times p$ matrix, for $p \geqslant 1$. Include the programs at the end of your live script.

   (b) If $AX = I$, then $X = A^{-1}$. Use this fact to write a MATLAB function `ltinverse` that uses your modified `forelim` to compute the inverse of a lower triangular matrix. Test your function using the following matrices, that is, compare your numerical solutions against the given exact solutions.

$$L_1 = \begin{bmatrix} 2 & 0 & 0 \\ 8 & -7 & 0 \\ 4 & 9 & -27 \end{bmatrix}, \qquad L_1^{-1} = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ \frac{4}{7} & -\frac{1}{7} & 0 \\ \frac{50}{189} & -\frac{1}{21} & -\frac{1}{27} \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{3} & 1 & 0 & 0 \\ 0 & \frac{1}{3} & 1 & 0 \\ 0 & 0 & \frac{1}{3} & 1 \end{bmatrix}, \qquad L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 & 0 \\ \frac{1}{9} & -\frac{1}{3} & 1 & 0 \\ -\frac{1}{27} & \frac{1}{9} & -\frac{1}{3} & 1 \end{bmatrix}$$

2. (PLU factorization) Complete the program `myplu.m` on p. 56 of Module 2 lecture slides. Include the program at the end of your live script. Then test your code by running it on a 500×500 matrix with random entries, *e.g.*, generated by `rand`, `randi`, or `randn`.

3. (**FNC** 2.4.6) When computing the determinant of a matrix by hand, it is common to use cofactor expansion and apply the definition recursively. But this is terribly inefficient as a function of the matrix size.

   (a) Explain why, if $A = LU$ is an LU factorization,

$$\det(A) = u_{11}u_{22}\cdots u_{nn} = \prod_{i=1}^{n} u_{ii}.$$

This part is an analytical question. Do it by hand.

(b) Using the result of part (a), write a MATLAB function `determinant` that computes the determinant of a given matrix `A` using `mylu` from p. 55 of Module 2 lecture slides. Include the function at the end of your live script. Use your function and the built-in `det` on the matrices `magic(n)` for $n = 3, 4, \ldots, 7$, and make a table (using `disp` or `fprintf`) showing n, the value from your function, and the relative error when compared to `det`.

4. (FLOP Counting) Do **LM** 10.1–12(a,b,d). Justify your calculation of p and c for each part.

5. (Properties of norms) Do **LM** 10.2–1. Justify all your answers.