Math 3607: Homework 5

Due: 11:59PM, Monday, February 22, 2021

TOTAL: 20 points

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) Modify `backsub.m` and `forelim.m` from Lecture 13 so that they solve the case where the second input is an $n \times p$ matrix, for $p \geq 1$.

   (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 on at least two nontrivial matrices, that is, compare the numerical solutions against the exact solutions.

2. (PLU factorization) Complete the program `myplu.m` on p. 20 of Lecture 14 slides. Then test your code by running it on a 500×500 matrix with random entries, *e.g.*, generated by `rand`, `randi`, or `randn`.

   Hint. Read Section 2.1–2.7 of **NCM**, which is freely available at `https://www.mathworks.com/moler/chapters.html`. The code `lutx` found in Section 2.7 may be particularly helpful.

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 and you may need to review related linear algebra.)

   (b) Using the result of part (a), write a MATLAB function `determinant` that computes the determinant of a given matrix `A` using `mylu` from Lecture 14. 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. (Row and column operations) Read and study the appendix to Lecture 14. Then do **LM** 10.1–8.

5. Do **LM** 10.1–12(a,b,d).

6. Do **LM** 10.2–1.

7. **Graphics exercise of the week: Koch snowflake**

   This is an *optional* problem for those interested in further developing programming skills and creating cool graphics. Read **LM** 7.2 on recursion and do **LM** 7.2–12 and 7.2–13 to generate Koch snowflakes.

   The figures below are all generated with `level=6`.

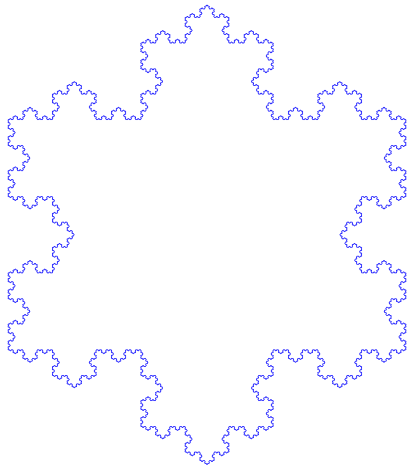

Figure 1: Koch curves around a *negatively* oriented triangle. Generated by letting the angles be $0, -120$, and -240.

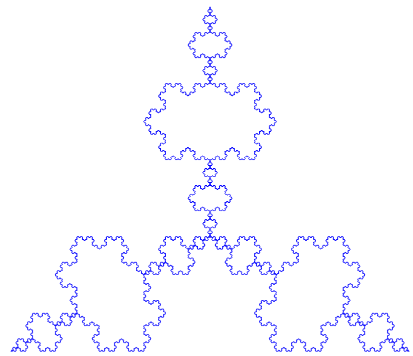

Figure 2: Koch curves around a *positively* oriented trianlge. Generated by letting the angles be $0, 120$, and 240.

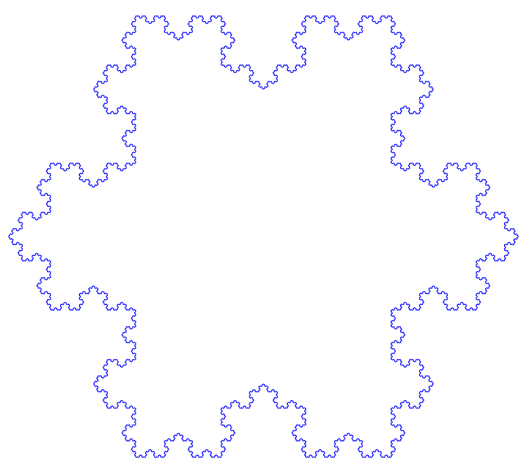

Figure 3: Koch curves around a hexagon. Generated by letting the angles be $0, 60, \ldots, 300$.