Assignment #5

Due Wednesday, 9 October 2024, at the start of class

This Assignment is based on Chapters 5 and 7 of the textbook,[1] and mostly on the latter Chapter. Please read these Chapters! Understanding the (long) section 7.2 is especially important, though subsections 7.2.4 and 7.2.5 less so.

When you turn in homework problems, please put the problems in the order they appear below. Also, two expectations always apply:
1. When you use MATLAB, or other language of your choice, the commands you used must be shown, along with the results.
2. Please minimize use of paper; edit your result to remove extra space. However, please keep a clear distinction between codes, input commands, and computed results and/or figures.

Do the following exercises from the textbook:

CHAPTER 5
- Exercise 15 on page 123.

CHAPTER 7
- Exercise 2 on page 175.
- Exercise 3 on pages 175–176.
- Exercise 4 on page 176.
- Exercise 6 on page 176.

Do the following additional problems:

P4. **(a)** Consider the linear system

$$2x_1 + 3x_2 - x_3 = 5$$
$$4x_1 - 3x_2 + 2x_3 = 1$$
$$2x_1 + x_2 + x_3 = 3$$

Perform Gaussian elimination with back substitution *by hand* to solve the system. (*Do not swap rows. Please indicate the row operations you do, and the result of each elimination stage, but otherwise there is no need to show arithmetic.*)

(b) Let M be the augmented matrix $M = \begin{bmatrix} A|b \end{bmatrix}$, and also define

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \qquad L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{2}{9} & 1 \end{bmatrix}.$$

---

[1]Greenbaum & Chartier, *Numerical Methods: Design, Analysis, and Computer Implementation of Algorithms*, Princeton University Press 2012).

Compute $L_2 L_1 M$ *by hand*. Confirm that the resulting augmented matrix is the upper triangular system which you solved, i.e. by back substitution, in part **(a)**.

In the remaining parts feel free to use MATLAB *(or language of your choice).*

(c)   The original system has form $Ax = b$. Enter A and **b** into MATLAB etc. and confirm your by-hand solution by `x = A \ b` (or equivalent in your language).

(d)   Using the matrices defined in part **(b)**, compute $U = L_2 L_1 A$ and $L = (L_2 L_1)^{-1}$. Confirm that $LU = A$.

(e)   In section 7.2 it says "...to solve $Ax = LUx = b$, one first solves the lower triangular system $Ly = b$ (to obtain $y = Ux$) and then the upper triangular system $Ux = y$." Using U and L from part **(d)**, solve the two triangular systems using the backslash operator (or equivalent in your language). Confirm that you get the same answer this way as in **(a)**.

P5.   *Completely revised! Do Exercise 2 (page 175) first, before doing this problem.*

(a)   Open a new m-file[2] called `ge.m`. Type in the code from section 7.2, at the top of page 137, "`% Gaussian elimination without pivoting`". It is recommended that you make this a function,
```
function [newA, newb] = ge(A, b, n)
```
which returns the new A, b, but this is not required. Of course, you don't need to type in the comments, but please be careful with the loop indices and other details. However you write it, the variables `A,b,n` must be defined and of the right size.

(b)   Consider the 4×4 linear system $Ax = b$:

$$
\begin{aligned}
x_1 + 2x_2 + 3x_3 + 4x_4 &= 7 \\
2x_1 + x_2 \qquad\quad - x_4 &= -1 \\
x_1 \qquad\qquad\quad + x_4 &= 4 \\
2x_2 - 2x_3 \qquad &= -8
\end{aligned}
$$

Run `ge.m` on this example. Show the resulting/new A and **b**. (*These will be different from the ones you typed in!*)

(c)   As a result of part **(b)**, the system is now upper triangular. Solve the system using the code `usolve` which you wrote in Exercise 2.

(d)   Assuming `x` is what you computed in **(c)**, and using the original A and **b**, confirm using `norm(A*x - b)` that x is correct.

(e)   Confirm using `A \ b`, on the original A and **b**, that you have the correct solution x. (*This is simply another way to check what you already know from part **(d)**).*

---

[2]If you are using a language other than MATLAB, please interpret this problem in a reasonable manner. That is, implement basic Gaussian elimination using `for` loops and direct array indexing.