Homework 3.

Amath 352
Applied Linear Algebra and Numerical Analysis

Due: 1/27/23 at 11:59pm to Gradescope

Directions:

Complete all component skills exercises as neatly as possible. Up to 2 points may be deducted for homework that is illegible and/or poorly organized. You are encouraged to type homework solutions, and a half bonus point will be awarded to students who use LaTeX. (Check out my LaTeX beginner document and overleaf.com if you are new to LaTeX.) If you prefer not to type homeworks, I ask that **homeworks be scanned.** (I will not accept physical copies.) In addition, **homeworks must be in .pdf format.**

Pro-Tips:

- You have access to some solutions of the textbook exercises and are encouraged to use them. Note that these solutions are not always correct, so double check your work just in case.
- Teamwork makes the dream work, but please indicate at the top of your assignment who your collaborators are.
- Don't wait until the last minute. ☺

Component Skills Exercises

Exercise 1. (CS2.3)

From Olver and Shakiban, complete Exercises 1.3.21(e) and (f).

Exercise 2. (CS2.3)

Create a MATLAB script that computes the LU factorization of a regular linear system. For concreteness, your script should start with the regular matrix

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

and return the matrices L and U such that $A = LU$. You are not allowed to use MATLAB's built-in function for the LU decomposition.

Hint: Modify the code provided in Homework 2 for regular Gaussian elimination. This will easily get you the U matrix. The L matrix comes from the negatives of the scalar multiples used for the elimination steps. Use these to build up your L matrix.

Exercise 3. (CS2.4)

From Olver and Shakiban, complete Exercises 1.4.21(b) and (c). **Use partial pivoting to determine whether to exchange rows.** You only need to compute the PLU decomposition. Do not solve the systems.

Exercise 4. (CS2.4)

Consider the linear system

$$\begin{cases} \varepsilon x_1 + x_2 = 3 \\ x_1 + 2x_2 = 5 \end{cases},$$

where $|\varepsilon| \ll 1$ is a very small parameter. In this exercise, we will solve this linear system by regular Gaussian elimination and then by Gaussian elimination with partial pivoting. We will see that Gaussian elimination with partial pivoting is preferred when numerical round-off errors are present.

a. What is the solution of the linear system when $\varepsilon = 0$? We can use this solution as a proxy for the exact solution when $\varepsilon \neq 0$ but $|\varepsilon| \ll 1$.

b. Let's solve the system by regular Gaussian elimination, subject to numerical round-off.

   i. Reduce the system to triangular form by regular Gaussian elimination.

   ii. Since $|\varepsilon| \ll 1$, expressions like $a + b/\varepsilon$ are often rounded to just b/ε on a computer[1]. Apply this approximation to your triangular system. Carry out back substitution as usual. Compare your solution with (a).

c. Let's solve the system by Gaussian elimination with partial pivoting, subject to numerical round-off.

   i. Reduce the system to triangular form by Gaussian elimination with partial pivoting.

   ii. Since $|\varepsilon| \ll 1$, expressions like $a + b\varepsilon$ are often rounded to just a on a computer. Apply this approximation to your triangular system. Carry out back substitution as usual. Compare your solution with (a).

Multi-Step Problem

Suppose A has a LU decomposition. We can use this decomposition to solve linear systems in a slightly more efficient way. In particular, given

$$A\mathbf{x} = \mathbf{b},$$

we can use the LU decomposition of A to arrive at

$$LU\mathbf{x} = \mathbf{b}.$$

Now we introduce an intermediate vector $\mathbf{y} = U\mathbf{x}$. This intermediate vector divides our method of solving the linear system into two steps:

1. Solve $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$ by forward substitution.

2. Solve $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$ by backward substitution.

"What is forward substitution?" you may ask. It is precisely the same method as backward substitution, except now your coefficient matrix is lower-triangular and so you start by solving the first equation in your linear system and work down the hierarchy of equations. Here's a MATLAB code that performs forward substitution on a lower-triangular, augmented matrix:

```
% Forward substitution on a m by m+1 lower triangular matrix L.

[m,n] = size(L);
x = L(:,n); % Initialize column vector of unknowns (to be updated).
```

[1]Here, a and b are real numbers that are much smaller in size than $1/\varepsilon$ but much larger in size than ε.

```
x(1) = L(1,n)/L(1,1); % Solve first equation of augmented matrix.
for i = 2:m % i counts up from 2 to m in intervals of 1.
   SUM = 0;
   for j = 1:i-1
      SUM = SUM + L(i,j)*x(j);
   end
   x(i) = (L(i,n) - SUM)/L(i,i); % Updates the ith entry of x.
end
x % x is the solution of the linear system.
```

a. Compute the flop count of forward substitution. Please show the major steps of your calculation.

b. Compute the flop count of solving the linear system using the LU decomposition of A. (This will be the sum of flops for both the forward and backward substitution steps.) Please use results from your previous homework.

c. Based on your answers to (b) and previous homework questions, which of the following is more expensive: solving a linear system by Gaussian elimination or solving a linear system by LU decomposition, assuming the decomposition is known ahead of time?

Remark: Of course, the LU decomposition is not usually known ahead of time, and it requires just as many flops to compute as it does to perform Gaussian elimination. However, once you've computed the LU decomposition once, you know it for all time. Thus, if you happen to be solving $A\mathbf{x} = \mathbf{b}$ for several different $\mathbf{b}$, you can use Gaussian elimination to determine the LU decomposition and then use this decomposition to solve your systems. Doing so requires fewer flops than using Gaussian elimination to solve each system.