SI231 - Matrix Computations, 2022 Fall

Homework Set #2

Prof. Yue Qiu

Acknowledgements:

1) Deadline: **2022-10-28 10:59:59**
2) **Late Policy details** can be found on piazza.
3) Submit your homework in **Homework 2** on **Gradscope**. Entry Code: **4V2N55**. Make sure that you have correctly select pages for each problem. If not, you probably will get 0 point.
4) No handwritten homework is accepted. You need to write LaTeX. (If you have difficulties in using LaTeX, you are allowed to use **MS Word or Pages** for the first and the second homework to accommodate yourself.)
5) Use the given template and give your solution in English. Solution in Chinese is not allowed.
6) Your homework should be uploaded in the PDF format, and the naming format of the file is not specified.

I. LU DECOMPOSITION

Problem 1. (Jianguo Huang, 10 points $\times$ 3)

1) Given the 3×3 matrix: $A = \begin{bmatrix} 2 & -1 & 3 \\ 4 & 2 & 1 \\ -6 & -1 & 2 \end{bmatrix}$. Compute the LU decomposition of A.(you must complete this task by Gaussian Elimination and show the necessary details, but it isn't necessary to use complete pivoting or partial pivoting.)
2) Based on previous LU decomposition, can you solve the linear system $Ax = b$ where $x = (x_1, x_2, x_3)^T$ and $b = (4, 7, -5)^T$.
3) Compute the LU factorization of $A = \begin{bmatrix} 2 & 1 & 0 \\ -4 & 0 & 4 \\ 2 & 5 & 10 \end{bmatrix}$ with partial pivoting. Be sure to also give P.

(**You are highly required to write down your solution procedures in detail. And all values must be represented by integers or fractions, floating point numbers are not accepted.**)

Solution:

1)

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \qquad A^{(1)} = \begin{bmatrix} 2 & -1 & 3 \\ 0 & 4 & -5 \\ 0 & -4 & 11 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \qquad A^{(2)} = \begin{bmatrix} 2 & -1 & 3 \\ 0 & 4 & -5 \\ 0 & 0 & 6 \end{bmatrix}$$

then,U is $A^{(2)}$. And, $L = M_1^{-1} M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -1 & 1 \end{bmatrix}$

2) Let $Ly = b$, where $y = Ux$,by solving the following equations,

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} y_1 \\ y_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ -5 \end{bmatrix}$$

we can get that $y = (4, -1, 6)^T$(5 points) . and continuing to solve the following equations,

$$\begin{bmatrix} 2 & -1 & 3 \\ 0 & 4 & -5 \\ 0 & 0 & 6 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ 6 \end{bmatrix}$$

Finally, $x = (1, 1, 1)^T$.(5 points)

3) Here we have, $A = \begin{bmatrix} 2 & 1 & 0 \\ -4 & 0 & 4 \\ 2 & 5 & 10 \end{bmatrix} \xrightarrow{R_1 \longleftrightarrow R_2} \begin{bmatrix} -4 & 0 & 4 \\ 2 & 1 & 0 \\ 2 & 5 & 10 \end{bmatrix} \xrightarrow[R_3 \longleftarrow R_3 + \frac{1}{2}R_1]{R_1 \longleftarrow R_2 + \frac{1}{2}R_1} \begin{bmatrix} -4 & 0 & -4 \\ 0 & 1 & 2 \\ 0 & 5 & 12 \end{bmatrix} \xrightarrow{R_2 \longleftrightarrow R_3}$
$\begin{bmatrix} -4 & 0 & -4 \\ 0 & 5 & 12 \\ 0 & 1 & 2 \end{bmatrix} \xrightarrow{R_3 \longleftarrow R_3 - \frac{1}{5}R_2} \begin{bmatrix} -4 & 0 & 4 \\ 0 & 5 & 12 \\ 0 & 0 & -\frac{2}{5} \end{bmatrix} = U.$(3 points)

By the end, we would have $\mathbf{r}^T = \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}$ so

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{2} & \frac{1}{5} & 1 \end{bmatrix} (3points), \quad P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} (4points)$$

II. Permutation matrix

Problem 2. (Yuhuang Meng, 10 points $\times$ 3) A permutation matrix is a product of elementary matrices for row swaps.

1) If $\mathbf{P}$ is the $n \times n$ elementary matrix for a row swap, explain why $\mathbf{P} = \mathbf{P}^T = \mathbf{P}^{-1}$.
2) If P is any permutation matrix, show that $\mathbf{P}^T = \mathbf{P}^{-1}$.
   Hint: Permutation matrix $\mathbf{P} = \mathbf{P}_1\mathbf{P}_2 \cdots \mathbf{P}_n$ for elementary row swaps $\mathbf{P}_i$.
3) If $\mathbf{A}^T = \mathbf{A}^{-1}$, is A necessarily a permutation matrix? If yes, prove your answer. If no, give a counterexample.

Solution:

1) $\mathbf{P}$ acts by moving row i to row j for each column k. Then $\mathbf{P}^T$ acts by moving row j to row i for each column k and $\mathbf{P}^{-1}$ acts by moving row j to row i for each column k. Therefore, $\mathbf{P} = \mathbf{P}^T = \mathbf{P}^{-1}$. (10 points)
2) A permutation matrix is a product of elementary matrices for row swaps. Suppose $\mathbf{P} = \mathbf{P}_1\mathbf{P}_2 \cdots \mathbf{P}_n$, where each $\mathbf{P}_i$ interchanges some two rows of the identity matrix. We have $\mathbf{P}_i^T = \mathbf{P}_i$, $\mathbf{P}_i^2 = \mathbf{I}$. Therefore,
$$\mathbf{P}\mathbf{P}^T = (\mathbf{P}_1\mathbf{P}_2 \cdots \mathbf{P}_n)(\mathbf{P}_1\mathbf{P}_2 \cdots \mathbf{P}_n)^T = \mathbf{P}_1\mathbf{P}_2 \cdots \mathbf{P}_n\mathbf{P}_n^T \cdots \mathbf{P}_2^T\mathbf{P}_1^T = \mathbf{I},$$
   which implies $\mathbf{P}^T = \mathbf{P}^{-1}$. (10 points)
3) No. Let $\mathbf{A} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$. It is obvious that $\mathbf{A}^{-1} = \mathbf{A}^T = \mathbf{A}$. But $\mathbf{A}$ is not a permutation matrix, because it can't be obtained by swapping rows of the identity matrix. (10 points)

III. Banded matrix

Problem 3. (Bin Li,15 points)

Suppose an $m \times m$ matrix A is written in the block form $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$, where A_{11} is $n \times n$ and A_{22} is $(m-n) \times (m-n)$. Assume that the upper-left $k \times k$ block $A_{1:k,1:k}$ is nonsingular.

1) Verify the formula

$$\begin{pmatrix} I & \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix}$$

for "elimination" of the block A_{21} . The matrix $A_{22} - A_{21}A_{11}^{-1}A_{12}$ is known as the Schur complement of A_{11} in A.

2) Suppose A_{21} is eliminated row by row by means of n steps of Gaussian elimination. Show that the bottom-right $(m-n) \times (m-n)$ block of the result is again $A_{22} - A_{21}A_{11}^{-1}A_{12}$.

Solution:

1)

$$\begin{aligned} & \begin{pmatrix} I & \\ -A_{21}A_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \\ = & \begin{pmatrix} A_{11} & A_{12} \\ -A_{21}A_{11}^{-1}A_{11} + A_{21} & -A_{21}A_{11}^{-1}A_{12} + A_{22} \end{pmatrix} \\ = & \begin{pmatrix} A_{11} & A_{12} \\ & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix} \end{aligned}$$

(5 points)

2) The n steps of Gaussian elimination is equivalent to multiply a lower triangular matrix to A such that A_{11} become an upper triangular matrix and the block A_{21} become 0. Then consider the LU factorization of A_{11}. Suppose $A_{11} = L_{11}U_{11}$, then the lower triangular matrix for A can be represented as

$$L = \begin{pmatrix} L_{11}^{-1} & 0 \\ X & I \end{pmatrix}$$

where L_{11} is $n \times n$, I is $m-n$ Identity matrix, X is $(m-n) * n$ unknown matrix. Then

$$LA = \begin{pmatrix} L_{11}^{-1} & 0 \\ X & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} U_{11} & L_{11}^{-1}A_{12} \\ XA_{11} + A_{21} & XA_{12} + A_{22} \end{pmatrix},$$

where $XA_{11} + A_{21} = 0$. A_{11} is nonsingular, then $X = -A_{21}A_{11}^{-1}$. Hence, the bottom-right block is

$$XA_{12} + A_{22} = -A_{21}A_{11}^{-1}A_{12} + A_{22} = A_{22} - A_{21}A_{11}^{-1}A_{12},$$

The same result of (a).(10 points)

IV. Programming

Problem 4.(5 points + 10 points)

In this problem, we explore the efficiency of the LU method together with the classical linear system solvers we have learnt in linear algebra.

1) Derive the complexity of the LU decomposition. Particularly, how many flops does the LU decomposition require? The corresponding pseudo code (in Matlab) is provided as follows[1]:

Algorithm 1 Pseudo-code of LU decomposition

function NAIVE_LU($\mathbf{A}$)
    $n = size(\mathbf{A}, 1)$
    $\mathbf{L} = eye(n)$
    $\mathbf{U} = \mathbf{A}$
    for $k = 1 \to n-1$ **do**
        for $j = k+1 \to n$ **do**
            $\mathbf{L}(j,k) = \mathbf{U}(j,k)/\mathbf{U}(k,k)$
            $\mathbf{U}(j,k:n) = \mathbf{U}(j,k:n) - \mathbf{L}(j,k) * \mathbf{U}(k,k:n)$
        end for
    end for
    $\mathbf{U} = triu(\mathbf{U})$
end function

2) Randomly generate a non-singular matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and a vector $\mathbf{b} \in \mathbb{R}^{n\times 1}$, then program the following methods to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$:
   - **LU decomposition.** We first find the LU decomposition of $\mathbf{A}$, then we solve $\mathbf{L}\mathbf{y} = \mathbf{b}$ and $\mathbf{U}\mathbf{x} = \mathbf{y}$.
   - **The inverse method:** Use the inverse of $\mathbf{A}$ to solve the problem, which can be written as,

   $$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}\,.$$

   In your homework, you are required to submit the time-consuming plot (**one figure**) of given methods against the size of matrix $\mathbf{A}$ (i.e., n), where $n = 100, 200, \ldots, 1000$ (you can try larger n and see what will happen, be careful with the memory use of your PC!).

   Remarks:
   - You can use any language you like to program, but do not use built-in functions which are highly optimized to compute the LU decomposition or the matrix inverse (for example, Matlab function lu() and inv()). Otherwise, your results will contradict the complexity analysis, and your score will be discounted. You can implement the simplest version of these methods by yourself.

[1] $triu(\mathbf{U})$ is the Upper triangular part of the matrix $\mathbf{U}$

- In Matlab, to randomly generate a matrix or a vector, you can use randn function to generate normally distributed random numbers.
- The definition of *flop* is: **The float operations of float numbers**. So the division($/$), multiplication($\times$), addition($+$) and subtraction($-$) should be taken into consideration. However, the assignment ($=$) is not an operation on float numbers by convention.
- When handing in your homework in gradescope, package all your codes into **your_student_id+hw2_code.zip** and upload. In the package, you also need to include a file named README.txt/md to clearly identify the function of each file. Make sure that your codes can run and are consistent with your solutions.

Solution:

1) According to the pseudo code, the number of required flops F can be represented by

$$F = \sum_{k=1}^{n-1} \sum_{j=k+1}^{n} [1 + 2(n-k+1)] = \sum_{k=1}^{n-1} (n-k)[3 + 2(n-k)]$$

Let $t = n - k$. By $\sum_{t=1}^{n-1} t = \frac{1}{2}n(n-1)$ and $\sum_{t=1}^{n-1} t^2 = \frac{1}{6}n(n-1)(2n-1)$, we can derive that

$$F = \sum_{t=1}^{n-1} \left(2t^2 + t\right) = \frac{2}{3}n^3 + \frac{1}{2}n^2 - \frac{7}{6}n.$$

Hence, the LU decomposition algorithm requires $\mathcal{O}\left(\frac{2}{3}n^3\right)$ *flops*.

2) The figure is shown in Figure 1, which is consistent with the complexity analysis. That is, LU decomposition is the most efficient one as the size of matrix grows

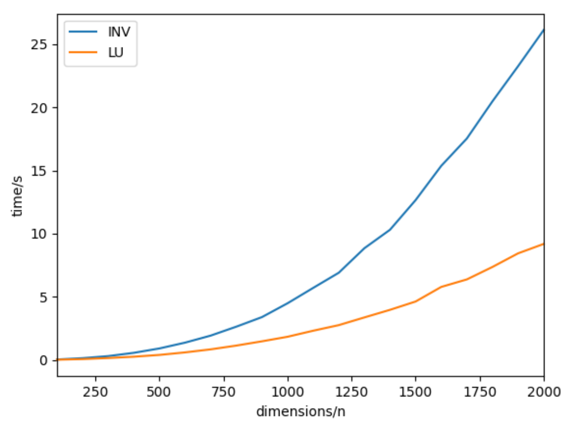

Figure 1: One example solution to compare the LU decomposition and the inverse method.