Numerical Analysis
Homework 2. LU Decomposition

EE17 102061125 陳冠鈞

1. **Objective**

In this assignment, I will use LU decomposition method to solve linear systems

$$\mathbf{Ax} = \mathbf{b}.$$

The advantage of LU decomposition over Gaussian elimination is that when the right hand side vector **b** is changed, Gaussian elimination method needs to be carried out again, but LU decomposition doesn't need to.

Therefore, LU decomposition is more effective in solving the linear system with different **b** vectors.

2. **Approach**

LU decomposition assumes matrix **A** can be factorized to be the product of two matrices **L** and **U**, where **L** is a lower triangular matrix and **U** is an upper triangular matrix.

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ \ell_{21} & u_{22} & u_{23} \\ \ell_{31} & \ell_{32} & u_{33} \end{bmatrix}$$

Briefly introduce the steps in LU decomposition:

- Form $\mathbf{u}_i$ row by copy $\mathbf{a}_{ij}$ to $\mathbf{u}_{ij}$
- Form $\mathbf{l}_j$ column by divide $\mathbf{a}_{ij}$ by $\mathbf{u}_{ii}$
- Update lower-right submatrix of **A**

Once the LU matrix is found, the solution to the linear system can be obtained using forward substitution and backward substitution.

$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{LUx} = \mathbf{b}$$

- Let **Ux = y**
  - ➔ **Ly = b**

$$\begin{bmatrix} 1 & 0 & 0 \\ \ell_{21} & 1 & 0 \\ \ell_{31} & \ell_{32} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$, and **y** can be obtained.

We call this is **forward substitution**.

- Once **y** is obtained,

  ➔ **Ux = y**

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

, and x is solved.

We call this is **backward substitution**.

3. Computational Complexity

- LU decomposition
  - The outer loop is carried out n times, i: 0 -> n-1.
  - In each iteration
    - Division is performed n-i-1 times
    - Multiplication and subtraction are performed $(n\text{-}i\text{-}1)^2$ times
  - Overall $O(n^3)$
    - Division is repeated

$$\sum_{i=0}^{n-1} n-i-1 = \sum_{j=0}^{n-1} j = \frac{n(n-1)}{2}$$

    - Subtraction and multiplication are repeated

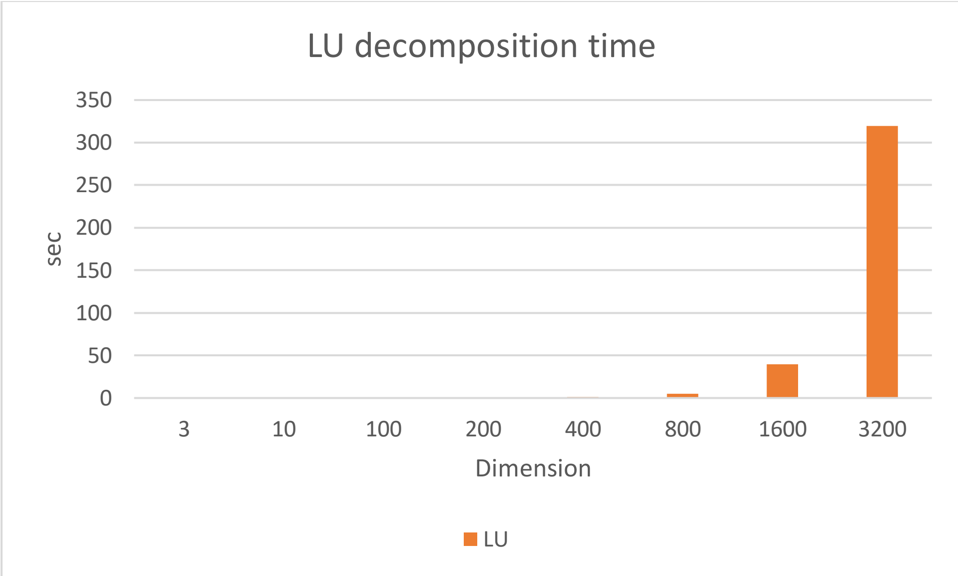
$$\sum_{i=0}^{n-1} (n-i-1)^2 = \sum_{j=0}^{n-1} j^2 = \frac{n^3-n}{3}$$

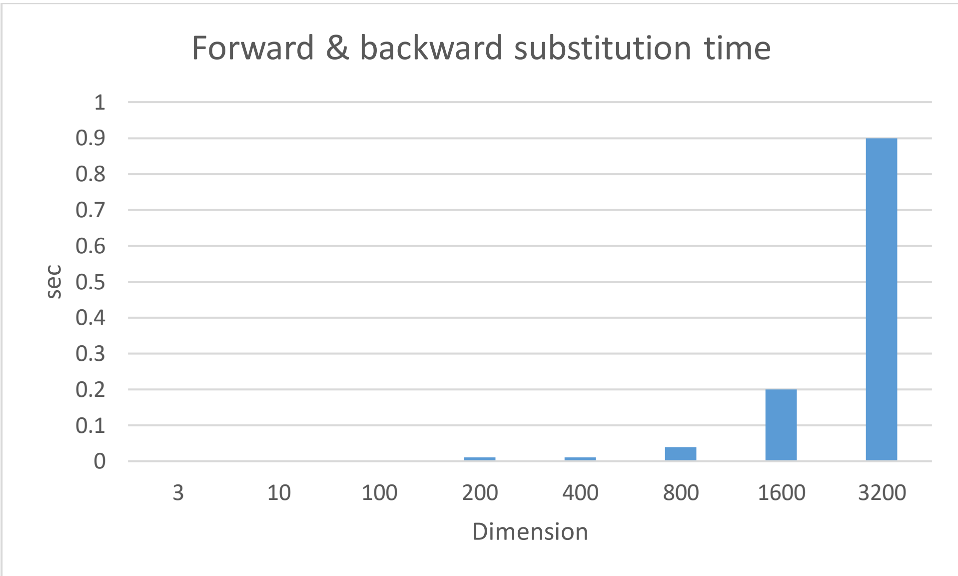
- Forward and backward substitution
  - Overall $O(n^2)$
    - Subtraction and multiplication are repeated

$$\sum_{i=0}^{n-1} n-i-1 = \sum_{j=0}^{n-1} j = \frac{n(n-1)}{2}$$

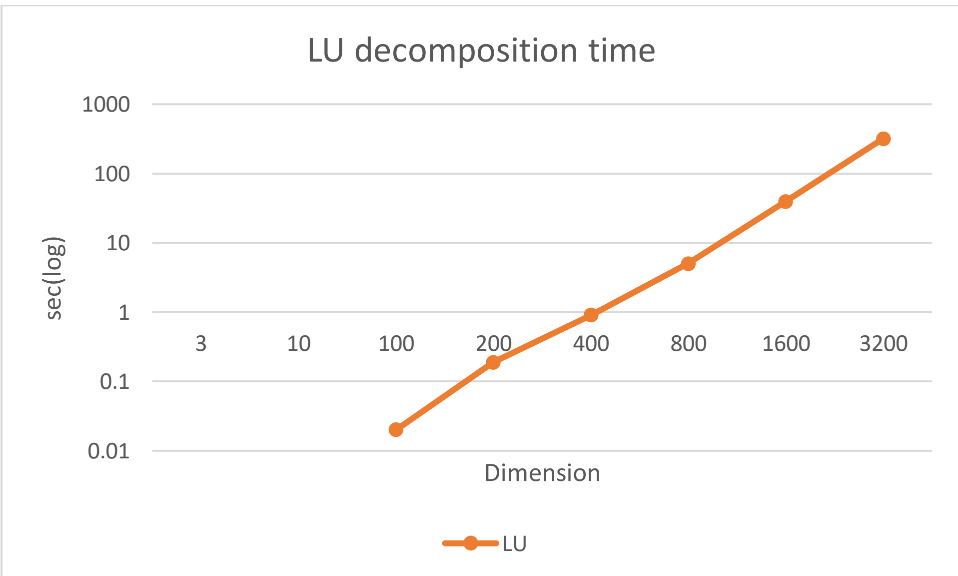
4. Results

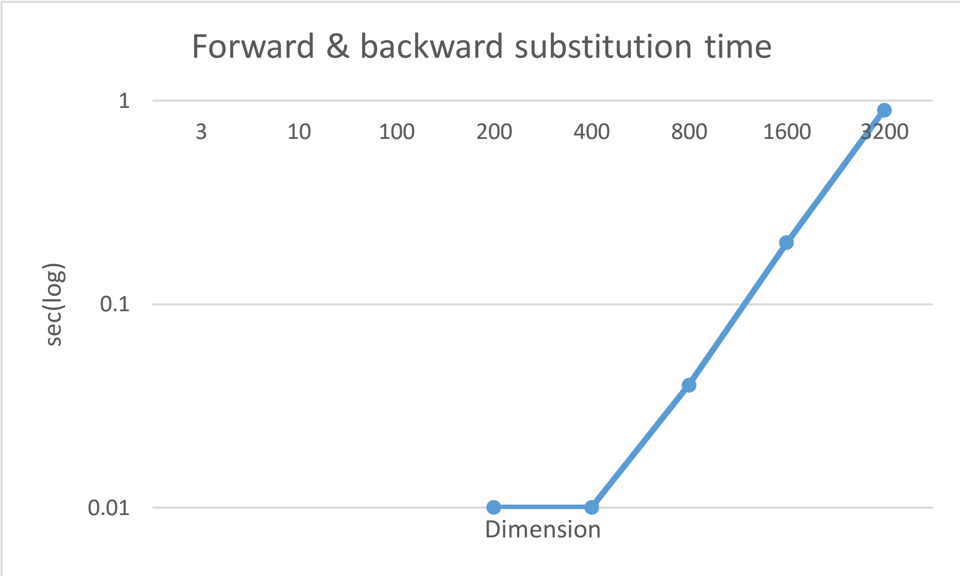
I record the CPU time of LU decomposition and forward & backward substitution, and the plots are below:





Now, I plot them as log scatter:





5. Observations

The CPU time of LU decomposition and forward & backward substitution increase exponentially. These results match the computational complexity.