CS211 hw2 Zi_Yang Report
name: Zi Yang
studentId: 862477486
Git: https://github.com/UCR-HPC/cs211-hw2-solving-large-linear-system-ZiYang-ucr

Q 1. Since $a[i,i] = 1$, all elements below $a[i,i]$ need to be divided by $a[i,i]$.

So, the matrix A becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ \frac{4}{1} & 13 & 18 \\ \frac{7}{1} & 54 & 78 \end{bmatrix}$$

Next, update $a[2,2]$ as follows:

$$a[2,2] = a[2,2] - a[2,1] \times a[1,2] = 5$$

So, the matrix A becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 18 \\ 7 & 54 & 78 \end{bmatrix}$$

Similarly, update $a[3,2]$:

$$a[3,2] = a[3,2] - a[3,1] \times a[2,1] = 40$$

Now, divide $a[3,2]$ by $a[2,2]$:

$$\frac{40}{5} = 8$$

So the matrix becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 18 \\ 7 & 8 & 78 \end{bmatrix}$$

update $a[2,3]$:

$$a[2,3] = a[2,3] - a[1,3] \times a[2,1] = 6$$

So the matrix becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 78 \end{bmatrix}$$

the last $a[3,3]$:

$$a[3,3] = a[3,3] - (a[1,3] \times a[3,1] + a[2,3] \times a[3,2]) = 9$$

the final matrix becomes :

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

so the L =

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 8 & 1 \end{bmatrix}$$

so the U =

$$U = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & 6 \\ 0 & 0 & 9 \end{bmatrix}$$

Q 2.

```
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=1000, pad=1
time=0.038398s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=2000, pad=1
time=0.203500s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=3000, pad=1
time=0.640567s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=4000, pad=1
time=1.304248s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=5000, pad=1
time=2.620849s
```

Figure 1: lapack execute time

```
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=1000, pad=1
time=0.133985s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=2000, pad=1
time=1.285116s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=3000, pad=1
time=5.610805s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.
n=5000, pad=1
time=28.703882s
[zyang253@cluster-001-login-node cs211-hw2-solving-large-linear-system-ZiYang-ucr]$ python3 starter.py
make: 'main' is up to date.

n=4000, pad=1
time=14.194812s
```

Figure 2: my execute time

$$\text{Gflops} = \frac{\text{flops}\,(2 \times n^3)}{\text{Execution Time} \times 10^9}$$

so the Lappack Gflop is :

n	Execution Time (s)	Gflops
1000	0.038398	52.096
2000	0.203500	78.615
3000	0.640567	84.312
4000	1.304248	98.154
5000	2.620849	95.378

Table 1: LAPACK Performance: Execution Time and Gflops for different n values

And the mydtrsv and mydgetrf Glop is :

Matrix Size (n)	Execution Time (s)	Gflops
1000	0.133985	14.93
2000	1.285116	12.45
3000	5.610805	9.625
4000	14.194812	9.017
5000	28.703882	8.710

Table 2: My Performance: Execution Time and Gflops

Q 3.

Since the block size is 2, we first compute the first two columns. Starting with $a[1,1] = a_{1,1}$, all elements below $a_{1,1}$ in the first column need to be divided by $a_{1,1}$.

So the matrix A becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ \frac{2}{1} & 9 & 12 & 15 \\ \frac{3}{1} & 26 & 41 & 49 \\ \frac{5}{1} & 40 & 107 & 135 \end{bmatrix}$$

Next, we compute $a[2,2]$ by subtracting the product of $a[1,2] \times a[2,1]$ from $a[2,2]$:

$$a[2,2] = a_{2,2} - a_{1,2} \times a_{2,1} = 5$$

Similarly, for the third and fourth rows:

$$a[3,2] = a_{3,2} - a_{1,2} \times a_{3,1} = 20$$
$$a[4,2] = a_{4,2} - a_{1,2} \times a_{4,1} = 30$$

Thus, the updated matrix A becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 12 & 15 \\ 3 & 20 & 41 & 49 \\ 5 & 30 & 107 & 135 \end{bmatrix}$$

Next, the elements below $a[2,2]$ in the second column are divided by $a[2,2$. So the matrix becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 12 & 15 \\ 3 & 4 & 41 & 49 \\ 5 & 6 & 107 & 135 \end{bmatrix}$$

by updating $a[2,3]$ using the formula $a[2,3] = a[2,3] - a[1,3] \times a[2,1] = 6$. Similarly, we compute $a[2,4]$:

$$a[2,4] = a[2,4] - a[1,4] \times a[2,1] = 7$$

Now we move on to compute the final block:

$$a[3,3] = a[3,3] - (a[1,3] \times a[3,1] + a[2,3] \times a[3,2]) = 8$$

Similarly, for $a[3,4]$:

$$a[3,4] = a[3,4] - (a[1,4] \times a[3,1] + a[2,4] \times a[3,2]) = 9$$

At this point, the matrix A becomes:

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 6 & 7 \\ 3 & 4 & 8 & 9 \\ 5 & 6 & 107 & 135 \end{bmatrix}$$

Finally, we update $a[4,3]$ and $a[4,4]$:

$$a[4,3] = a[4,3] - (a[1,3] \times a[4,1] + a[2,3] \times a[4,2]) = 7$$
$$a[4,4] = a[4,4] - (a[1,4] \times a[4,1] + a[2,4] \times a[4,2] + a[3,4] \times a[3,4]) = 10$$

final A =

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 6 & 7 \\ 3 & 4 & 8 & 9 \\ 5 & 6 & 7 & 10 \end{bmatrix}$$

so L =

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ 5 & 6 & 7 & 1 \end{bmatrix}$$

U =

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 5 & 6 & 7 \\ 0 & 0 & 8 & 9 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$