# OpenMP

kevin Lopez Chavez

November 22, 2024

## 1 Report deliverables

- The file is named `report.pdf`.
- An explanation of why I think this application is a good choice for OpenMP (Include the link to the exact location where the original program is referenced from) in Sections 2.
- An explanation of the code, the program's functionality, flow, and OpenMP acceleration in Section 3.
- Explained the estimated speed up in Section 4.
- Proof of achieved speedup with expectations in Section 5.

## 2 Explanation why data normalization is a good choice of program for OpenMP

OpenMP can achieve a high speed over independent for-loop interactions, and it's the idea of why it can achieve a high speedup on z-score normalization (or any normalization strategy, also *batch normalization*). In the case of z-score normalization, the total(sum) must be calculated, which can be parallelized. Calculate the difference to the mean, which can also be parallelized, and update all the current data, which can also be normalized. With that being said, it's possible to parallelize all the different operations in z-score normalization.

### 2.1 Links to Exact Locations

I referred to the following articles to understand the implementation and usage of Z-Score Normalization.

**Explanation and example usage of data normalization**:

- `https://developers.google.com/machine-learning/crash-course/numerical-data/normalization#z-score_scaling`

**The code implementation of the Z-score normalization (Scikit-learn repository):**

- `https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/preprocessing/_data.py`

## 3 Code Explanation

Here, I will explain the original code implementation in Section 3.1 and the OpenMP code implementation in Section 3.2.

## 3.1 Original Algorithm

The original algorithm (not optimized) uses the basic functions of mean and standard deviation to update each value individually and one by one. Since this needs to be run a single thread or process at a time, it takes $n$ times to finish executing the algorithm, where n is the number of data points. I have made functions to find the mean and standard deviation. And used in the main normalization function. Each function is defined as follows:

- The following formula is used to calculate the *Z-score normalization*: $Z = \frac{\vec{X}-\mu}{\sigma}$. where: $\vec{X}$ is the original input, $\mu$ is the mean of $\vec{X}$, $\sigma$ is the standard deviation of $\vec{X}$.
- The mean is calculated by $\mu = \frac{\sum_{i=1}^{n} x_i}{n}$; where: $x_i$ is the $i$th data point, and $n$ is the total points.
- The standard deviation is calculated by: $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}}$.

## 3.2 OpenMP

It is possible to use OpenMP to handle multiple data points simultaneously. In the case of parallelizing z-score-normalization, it is possible to parallelize a lot of the work done on the data. Below are the different loops that can be parallelized and provide a high speed-up.

- Calculating the sum (needed for the mean ).
- Calculating the sum of squared deviations (needed for the standard deviation)
- Z-score Normalizing the actual data.

### 3.2.1 Matrix Normalization

The following formula is used to calculate the *Z-score normalization*: $Z = \frac{\vec{X}-\mu}{\sigma}$. where: $\vec{X}$ is the original input; $\mu$ is the mean of $\vec{X}$; $\sigma$ is the standard deviation of $\vec{X}$; In my case of normalizing data with *Z-score normalization*, I need to calculate the mean (Section 3.2.2)and Standard Deviation (Section 3.2.3 ), which is parallelized using OpenMP.

Once I have the mean and standard deviation, we must update every value based on the formula. To speed up the process of updating every value, it is also possible to use OpenMP, handling n data points at a time. Where n is the number of cores allowed to run parallel done by the function`omp_set_num_threads(n)`. It is possible to get the best speed up when using the maximum number of cores in the PC by using the function `omp_get_max_threads()`

To actually normalize the data, it is possible to use the basic formula from the original function. Since this function does not have any dependencies, it is possible to optimize it using OpenMP basic for loop `#pragma omp parallel for`. Using this OpenMP macro allows the compiler to distribute the work between the different processors configured and achieve higher speed up compared to the single-threaded program.

### 3.2.2 Mean

The mean is calculated by $\mu = \frac{\sum_{i=1}^{n} x_i}{n}$; where: $x_i$ is the $i$th data point, and $n$ is the total number of data points.

To speed up the mean calculation, it is possible to calculate the sum in parallel; the only issue is that the sum variable is a value shared across all the running processes. OpenMP does allow to speed this up even with the dependency by using the **reduction** keyword along with the variable shared. The complete macro to speed up the sum calculation is `#pragma omp parallel for reduction(+:sum)` where the sum is the shared variable(each process would have its own copy of it where they accumulate the sum). In the end, OpenMP reduces all the sum variables copied into the processes into one global variable.

### 3.2.3 Standard Deviation

The standard deviation is calculated by: $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}}$; where: $x_i$ is the $i$th data point, $\mu$ is the mean, $n$ is the total points.

With the mean of the whole data, it is possible to get the deviation of every data point and sum to the variable *sum*. Again, since it is necessary to write to the same variable name, this would cause race conditions; in this case, we again use the same method **reduction** from opemMP to handle it. The full line to optimize all the operations is `#pragma omp parallel for reduction(+:sum)` where *sum* variable is the sum of squared deviations. In the end, to get the variance, we just divide it by the number of elements.

## 3.3 Compilation Steps and Flags

I used the following commands to compile the code

- **Original version:**
  - `$ g++ ./z_score_norm_original.cpp -o original.exe`
- **OpenMP version:**
  - `$ g++ -fopenmp ./z_score_norm_OpenMP.cpp -o OpenMP.exe`

Where the flags mean the following:

- `g++`: Compiler used to compile the C++ code.
- `-fopenmp`: This flag enables OpenMP.
- `./z_score_norm_original.cpp ./z_score_norm_OpenMP.cpp`: The source file.
- `-o original.exe`: The `-o`: name of the output executable file.
- `&& ./original.exe`: Execute the program after compiling.

to run the code and get meaningful results the code was run 20 times in a row, for this I used the bash script below.

```bash
$ echo -e "-----\n\n\n\n\n ################### ORIGINAL ###################"
g++ ./z_score_norm_original.cpp -o original.exe
for i in {1..20}
do
  echo -e "-----\n\n\n\nRun #$i"
  ./original.exe

echo -e "-----\n\n\n\n\n ################### OpenMP ###################"
g++ -fopenmp ./z_score_norm_OpenMP.cpp -o OpenMP.exe
for i in {1..20}
do
  echo -e "-----\n\n\n\nRun #$i"
  ./OpenMP.exe
done
```

# 4 Estimated Speedup

Looking down from the big picture, we might assume that this should get a speed up of $n$ where n is the number of processors, but there is one thing that we still need to take into account, and there are still some portions of the code that need to be handled sequentially. In the **reduce** method of OpenMP, the multiple processors need to create private locations of the sum variable that allow it to accumulate the sum value across different processors, and it would need to reduce it back to one single sum. This would take some time with the addition of the rest of the code that needs to run in a single core. For example, entering functions, making the last set of operations(divisions), and other operations.

# 5 Proof of Achieved Speedup

To prove the speed-up achieved, I have attached the logs and Figures 1,2. I posted the logs (run the programs 20 times) for the original version in section 5.1 and the oepnMP version in section 5.2.

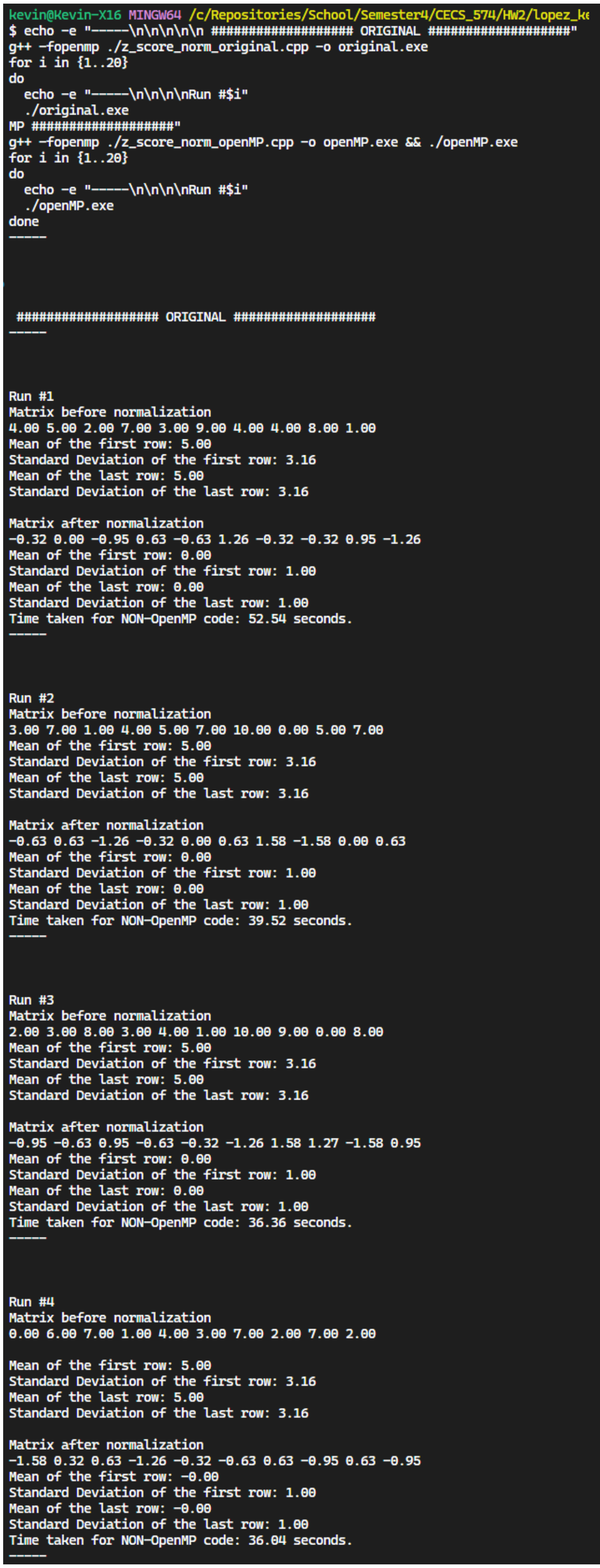
The programs were executed 20 times, the numbers are shown in table 1:

Table 1: OpenMP and Original Comparison

| Run_number | original_time | OpenMP_time |
|---|---:|---:|
| run #1 | 52.54 | 6.04 |
| run #2 | 39.52 | 5.84 |
| run #3 | 36.36 | 6.01 |
| run #4 | 36.04 | 6.46 |
| run #5 | 33.96 | 6.37 |
| run #6 | 52.62 | 5.22 |
| run #7 | 52.31 | 6.12 |
| run #8 | 31.13 | 6.70 |
| run #9 | 31.12 | 5.68 |
| run #10 | 30.87 | 5.74 |
| run #11 | 36.23 | 5.19 |
| run #12 | 60.24 | 4.43 |
| run #13 | 36.28 | 5.55 |
| run #14 | 37.14 | 6.05 |
| run #15 | 33.95 | 8.20 |
| run #16 | 36.74 | 8.13 |
| run #17 | 60.59 | 8.15 |
| run #18 | 63.19 | 7.24 |
| run #19 | 36.95 | 8.33 |
| run #20 | 38.10 | 8.09 |
| **average** | **41.79** | **6.48** |
| standard_deviation | 10.71 | 1.16 |
| speed_up | 1.00 | **6.45** |

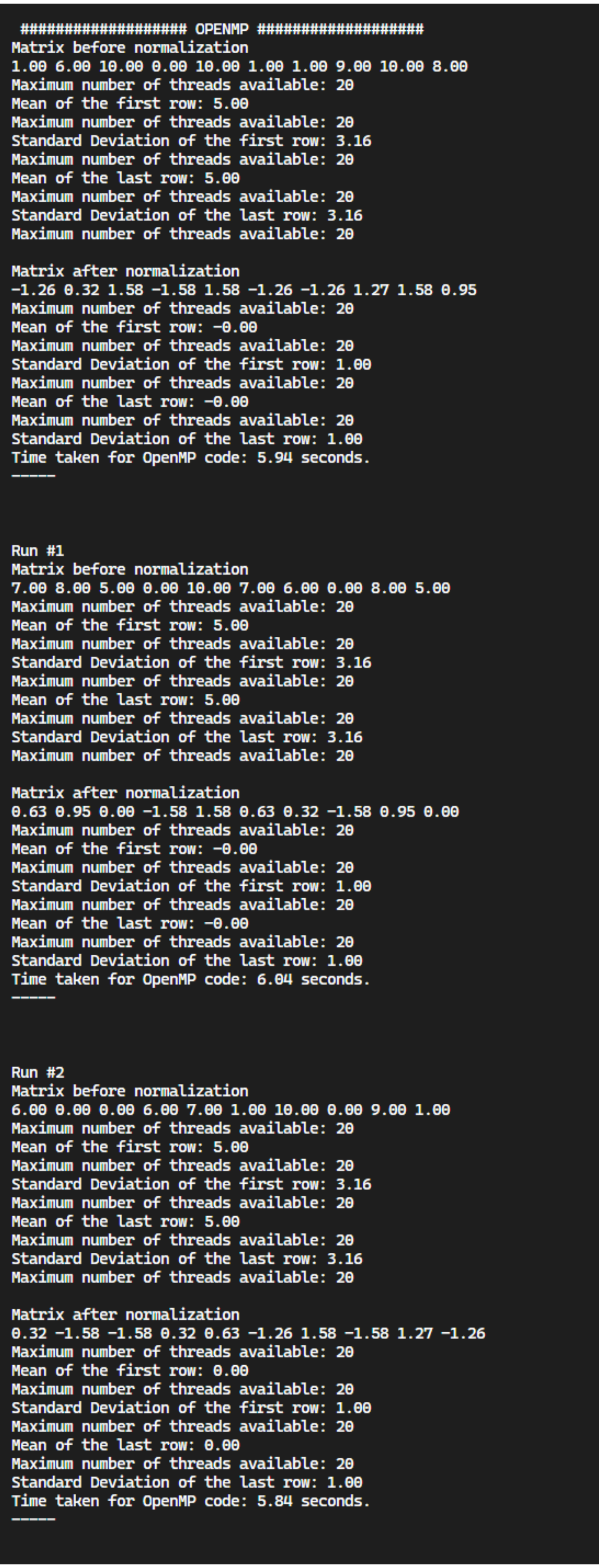
$$\text{Speed-up} = \frac{\text{Time for Original Version}}{\text{Time for OpenMP Version}} = \frac{41.79}{6.48} \approx 6.45$$

One can assume that the speed-up would be close to the number of parallel processes running. When looking in, we can see that there is more to why it won't run at the expected theoretical max. There are portions of the code that need to run in a single process, and also, the OpenMP instructions can add some overhead(The reduce method would need to make copies of the sum variable in each process and then reduce it back to one). Therefore, getting a speedup of ≈ 6.45 is a valid number for this application.



```
kevin@Kevin-X16 MINGW64 /c/Repositories/School/Semester4/CECS_574/HW2/lopez_ke
$ echo -e "-----\n\n\n\n\n ################### ORIGINAL ###################"
g++ -fopenmp ./z_score_norm_original.cpp -o original.exe
for i in {1..20}
do
  echo -e "-----\n\n\n\nRun #$i"
  ./original.exe
MP ###################"
g++ -fopenmp ./z_score_norm_openMP.cpp -o openMP.exe && ./openMP.exe
for i in {1..20}
do
  echo -e "-----\n\n\n\nRun #$i"
  ./openMP.exe
done
-----




 ################### ORIGINAL ###################
-----




Run #1
Matrix before normalization
4.00 5.00 2.00 7.00 3.00 9.00 4.00 4.00 8.00 1.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.32 0.00 -0.95 0.63 -0.63 1.26 -0.32 -0.32 0.95 -1.26
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 52.54 seconds.
-----



Run #2
Matrix before normalization
3.00 7.00 1.00 4.00 5.00 7.00 10.00 0.00 5.00 7.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.63 0.63 -1.26 -0.32 0.00 0.63 1.58 -1.58 0.00 0.63
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 39.52 seconds.
-----



Run #3
Matrix before normalization
2.00 3.00 8.00 3.00 4.00 1.00 10.00 9.00 0.00 8.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.95 -0.63 0.95 -0.63 -0.32 -1.26 1.58 1.27 -1.58 0.95
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 36.36 seconds.
-----



Run #4
Matrix before normalization
0.00 6.00 7.00 1.00 4.00 3.00 7.00 2.00 7.00 2.00

Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-1.58 0.32 0.63 -1.26 -0.32 -0.63 0.63 -0.95 0.63 -0.95
Mean of the first row: -0.00
Standard Deviation of the first row: 1.00
Mean of the last row: -0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 36.04 seconds.
-----
```

Figure 1: Proof of speed up(Original)



```
 ################### OPENMP ###################
Matrix before normalization
1.00 6.00 10.00 0.00 10.00 1.00 1.00 9.00 10.00 8.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-1.26 0.32 1.58 -1.58 1.58 -1.26 -1.26 1.27 1.58 0.95
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.94 seconds.
-----



Run #1
Matrix before normalization
7.00 8.00 5.00 0.00 10.00 7.00 6.00 0.00 8.00 5.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.63 0.95 0.00 -1.58 1.58 0.63 0.32 -1.58 0.95 0.00
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.04 seconds.
-----



Run #2
Matrix before normalization
6.00 0.00 0.00 6.00 7.00 1.00 10.00 0.00 9.00 1.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.32 -1.58 -1.58 0.32 0.63 -1.26 1.58 -1.58 1.27 -1.26
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.84 seconds.
-----
```

Figure 2: Proof of speed up (OpenMP)

## 5.1 Logs for Original File Execution

```

kevin@Kevin-X16 MINGW64 /c/Repositories/School/Semester4/CECS_574/HW2/lopez_kevin_013378831 (main)
$ echo -e "-----\n\n\n\n\n ################### ORIGINAL ###################"
g++ -fopenmp ./z_score_norm_original.cpp -o original.exe
for i in {1..20}
do
  echo -e "-----\n\n\n\nRun #$i"
  ./original.exe

echo -e "-----\n\n\n\n\n ################### OPENMP ###################"
g++ -fopenmp ./z_score_norm_openMP.cpp -o openMP.exe && ./openMP.exe
for i in {1..20}
do
  echo -e "-----\n\n\n\nRun #$i"
  ./openMP.exe
done
-----




 ################### ORIGINAL ###################
-----



Run #1
Matrix before normalization
4.00 5.00 2.00 7.00 3.00 9.00 4.00 4.00 8.00 1.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.32 0.00 -0.95 0.63 -0.63 1.26 -0.32 -0.32 0.95 -1.26
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 52.54 seconds.
-----



Run #2
Matrix before normalization
3.00 7.00 1.00 4.00 5.00 7.00 10.00 0.00 5.00 7.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.63 0.63 -1.26 -0.32 0.00 0.63 1.58 -1.58 0.00 0.63
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 39.52 seconds.
-----



Run #3
Matrix before normalization
2.00 3.00 8.00 3.00 4.00 1.00 10.00 9.00 0.00 8.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.95 -0.63 0.95 -0.63 -0.32 -1.26 1.58 1.27 -1.58 0.95
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
```

```
Time taken for NON-OpenMP code: 36.36 seconds.
-----



Run #4
Matrix before normalization
0.00 6.00 7.00 1.00 4.00 3.00 7.00 2.00 7.00 2.00

Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-1.58 0.32 0.63 -1.26 -0.32 -0.63 0.63 -0.95 0.63 -0.95
Mean of the first row: -0.00
Standard Deviation of the first row: 1.00
Mean of the last row: -0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 36.04 seconds.
-----



Run #5
Matrix before normalization
4.00 6.00 1.00 10.00 7.00 8.00 10.00 8.00 6.00 0.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.32 0.32 -1.26 1.58 0.63 0.95 1.58 0.95 0.32 -1.58
Mean of the first row: -0.00
Standard Deviation of the first row: 1.00
Mean of the last row: -0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 33.96 seconds.
-----



Run #6
Matrix before normalization
6.00 8.00 8.00 6.00 0.00 4.00 6.00 6.00 3.00 2.00

Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
0.32 0.95 0.95 0.32 -1.58 -0.32 0.32 0.32 -0.63 -0.95
Mean of the first row: -0.00
Standard Deviation of the first row: 1.00
Mean of the last row: -0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 52.62 seconds.
-----



Run #7
Matrix before normalization
8.00 8.00 7.00 2.00 5.00 8.00 8.00 2.00 2.00 8.00

Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
0.95 0.95 0.63 -0.95 0.00 0.95 0.95 -0.95 -0.95 0.95
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 52.31 seconds.
-----
```

```



Run #8
Matrix before normalization
8.00 8.00 5.00 1.00 3.00 1.00 5.00 4.00 4.00 3.00

Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
0.95 0.95 0.00 -1.26 -0.63 -1.26 0.00 -0.32 -0.32 -0.63
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 31.13 seconds.
-----



Run #9
Matrix before normalization
9.00 2.00 4.00 7.00 0.00 3.00 3.00 8.00 1.00 7.00

Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
1.27 -0.95 -0.32 0.63 -1.58 -0.63 -0.63 0.95 -1.26 0.63
Mean of the first row: -0.00
Standard Deviation of the first row: 1.00
Mean of the last row: -0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 31.12 seconds.
Standard Deviation of the last row: 3.16

Matrix after normalization
1.26 -0.95 -1.26 -0.32 -0.32 -1.58 0.00 -0.95 0.32 1.26
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 30.87 seconds.
-----



Run #11
Matrix before normalization
9.00 2.00 10.00 0.00 7.00 9.00 7.00 9.00 0.00 0.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
1.26 -0.95 1.58 -1.58 0.63 1.26 0.63 1.26 -1.58 -1.58
Mean of the first row: -0.00
Standard Deviation of the first row: 1.00
Mean of the last row: -0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 36.23 seconds.
-----



Run #12
Matrix before normalization
0.00 6.00 1.00 3.00 1.00 2.00 0.00 8.00 5.00 7.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-1.58 0.32 -1.26 -0.63 -1.26 -0.95 -1.58 0.95 0.00 0.63
```

```
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 60.24 seconds.
-----



Run #13
Matrix before normalization
10.00 4.00 10.00 2.00 10.00 7.00 3.00 6.00 3.00 2.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
1.58 -0.32 1.58 -0.95 1.58 0.63 -0.63 0.32 -0.63 -0.95
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 36.28 seconds.
-----



Run #14
Matrix before normalization
7.00 4.00 2.00 9.00 5.00 5.00 6.00 5.00 7.00 1.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
0.63 -0.32 -0.95 1.27 0.00 0.00 0.32 0.00 0.63 -1.26
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 37.14 seconds.
-----



Run #15
Matrix before normalization
0.00 9.00 8.00 4.00 8.00 10.00 6.00 6.00 8.00 2.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-1.58 1.26 0.95 -0.32 0.95 1.58 0.32 0.32 0.95 -0.95
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 33.95 seconds.
-----



Run #16
Matrix before normalization
3.00 7.00 8.00 8.00 1.00 3.00 1.00 9.00 1.00 6.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.63 0.63 0.95 0.95 -1.26 -0.63 -1.26 1.26 -1.26 0.32
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 36.74 seconds.
```

```
-----



Run #17
Matrix before normalization
1.00 4.00 0.00 2.00 10.00 9.00 4.00 6.00 7.00 3.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-1.26 -0.32 -1.58 -0.95 1.58 1.27 -0.32 0.32 0.63 -0.63
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 60.59 seconds.
-----



Run #18
Matrix before normalization
0.00 0.00 3.00 9.00 2.00 8.00 1.00 8.00 2.00 3.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-1.58 -1.58 -0.63 1.26 -0.95 0.95 -1.26 0.95 -0.95 -0.63
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 63.19 seconds.
-----



Run #19
Matrix before normalization
7.00 2.00 6.00 7.00 10.00 5.00 2.00 1.00 2.00 5.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
0.63 -0.95 0.32 0.63 1.58 0.00 -0.95 -1.26 -0.95 0.00
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 36.95 seconds.
-----



Run #20
Matrix before normalization
2.00 9.00 2.00 0.00 1.00 9.00 2.00 0.00 6.00 6.00
Mean of the first row: 5.00
Standard Deviation of the first row: 3.16
Mean of the last row: 5.00
Standard Deviation of the last row: 3.16

Matrix after normalization
-0.95 1.26 -0.95 -1.58 -1.26 1.26 -0.95 -1.58 0.32 0.32
Mean of the first row: 0.00
Standard Deviation of the first row: 1.00
Mean of the last row: 0.00
Standard Deviation of the last row: 1.00
Time taken for NON-OpenMP code: 38.10 seconds.
-----



```

## 5.2 Logs for OpenMP File Execution

```

 ################## OPENMP ##################
Run #1
Matrix before normalization
7.00 8.00 5.00 0.00 10.00 7.00 6.00 0.00 8.00 5.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.63 0.95 0.00 -1.58 1.58 0.63 0.32 -1.58 0.95 0.00
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.04 seconds.
-----



Run #2
Matrix before normalization
6.00 0.00 0.00 6.00 7.00 1.00 10.00 0.00 9.00 1.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.32 -1.58 -1.58 0.32 0.63 -1.26 1.58 -1.58 1.27 -1.26
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.84 seconds.
-----



Run #3
Matrix before normalization
5.00 3.00 7.00 2.00 3.00 6.00 3.00 1.00 8.00 9.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.00 -0.63 0.63 -0.95 -0.63 0.32 -0.63 -1.26 0.95 1.26
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
```

```
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.01 seconds.
-----



Run #4
Matrix before normalization
10.00 10.00 4.00 1.00 5.00 0.00 6.00 10.00 10.00 7.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
1.58 1.58 -0.32 -1.26 0.00 -1.58 0.32 1.58 1.58 0.63
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.46 seconds.
-----



Run #5
Matrix before normalization
8.00 3.00 0.00 8.00 1.00 5.00 9.00 10.00 10.00 4.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.95 -0.63 -1.58 0.95 -1.26 0.00 1.26 1.58 1.58 -0.32
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.37 seconds.
-----



Run #6
Matrix before normalization
4.00 4.00 6.00 8.00 0.00 10.00 3.00 1.00 9.00 0.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-0.32 -0.32 0.32 0.95 -1.58 1.58 -0.63 -1.26 1.27 -1.58
Maximum number of threads available: 20
Mean of the first row: -0.00
```

```
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.22 seconds.
-----



Run #7
Matrix before normalization
10.00 5.00 1.00 8.00 0.00 4.00 7.00 3.00 7.00 7.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
1.58 0.00 -1.26 0.95 -1.58 -0.32 0.63 -0.63 0.63 0.63
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.12 seconds.
-----



Run #8
Matrix before normalization
2.00 6.00 6.00 1.00 3.00 9.00 0.00 6.00 5.00 2.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-0.95 0.32 0.32 -1.26 -0.63 1.26 -1.58 0.32 -0.00 -0.95
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.70 seconds.
-----



Run #9
Matrix before normalization
5.00 5.00 1.00 5.00 6.00 3.00 6.00 9.00 3.00 8.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
```

```
0.00 0.00 -1.26 0.00 0.32 -0.63 0.32 1.27 -0.63 0.95
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.68 seconds.
-----



Run #10
Matrix before normalization
3.00 8.00 2.00 6.00 2.00 10.00 1.00 8.00 5.00 0.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-0.63 0.95 -0.95 0.32 -0.95 1.58 -1.26 0.95 0.00 -1.58
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.74 seconds.
-----



Run #11
Matrix before normalization
1.00 10.00 4.00 8.00 9.00 5.00 8.00 7.00 9.00 3.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-1.26 1.58 -0.32 0.95 1.26 0.00 0.95 0.63 1.26 -0.63
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.19 seconds.
-----



Run #12
Matrix before normalization
2.00 3.00 6.00 5.00 3.00 10.00 4.00 5.00 2.00 6.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
```

```
Maximum number of threads available: 20

Matrix after normalization
-0.95 -0.63 0.32 0.00 -0.63 1.58 -0.32 0.00 -0.95 0.32
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 4.43 seconds.
-----



Run #13
Matrix before normalization
7.00 4.00 7.00 0.00 2.00 6.00 1.00 5.00 3.00 8.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.63 -0.32 0.63 -1.58 -0.95 0.32 -1.26 0.00 -0.63 0.95
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 5.55 seconds.
-----



Run #14
Matrix before normalization
4.00 10.00 0.00 1.00 1.00 0.00 6.00 0.00 10.00 2.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-0.32 1.58 -1.58 -1.26 -1.26 -1.58 0.32 -1.58 1.58 -0.95

Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 6.05 seconds.
-----



Run #15
Matrix before normalization
2.00 4.00 6.00 7.00 8.00 5.00 10.00 0.00 10.00 9.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
```

```
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-0.95 -0.32 0.32 0.63 0.95 0.00 1.58 -1.58 1.58 1.26
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 8.20 seconds.
-----



Run #16
Matrix before normalization
5.00 9.00 8.00 5.00 1.00 9.00 10.00 1.00 8.00 1.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.00 1.27 0.95 0.00 -1.26 1.27 1.58 -1.26 0.95 -1.26
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 8.13 seconds.
-----



Run #17
Matrix before normalization
9.00 10.00 7.00 3.00 4.00 2.00 1.00 5.00 1.00 2.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
1.26 1.58 0.63 -0.63 -0.32 -0.95 -1.26 0.00 -1.26 -0.95
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 8.15 seconds.
-----



Run #18
Matrix before normalization
1.00 6.00 9.00 0.00 10.00 6.00 1.00 6.00 0.00 5.00
Maximum number of threads available: 20
```

```
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-1.26 0.32 1.26 -1.58 1.58 0.32 -1.26 0.32 -1.58 0.00
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 7.24 seconds.
-----



Run #19
Matrix before normalization
0.00 10.00 10.00 1.00 6.00 9.00 2.00 7.00 6.00 6.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
-1.58 1.58 1.58 -1.26 0.32 1.26 -0.95 0.63 0.32 0.32
Maximum number of threads available: 20
Mean of the first row: -0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: -0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 8.33 seconds.
-----



Run #20
Matrix before normalization
5.00 0.00 8.00 10.00 8.00 3.00 3.00 1.00 1.00 7.00
Maximum number of threads available: 20
Mean of the first row: 5.00
Maximum number of threads available: 20
Standard Deviation of the first row: 3.16
Maximum number of threads available: 20
Mean of the last row: 5.00
Maximum number of threads available: 20
Standard Deviation of the last row: 3.16
Maximum number of threads available: 20

Matrix after normalization
0.00 -1.58 0.95 1.58 0.95 -0.63 -0.63 -1.26 -1.26 0.63
Maximum number of threads available: 20
Mean of the first row: 0.00
Maximum number of threads available: 20
Standard Deviation of the first row: 1.00
Maximum number of threads available: 20
Mean of the last row: 0.00
Maximum number of threads available: 20
Standard Deviation of the last row: 1.00
Time taken for OpenMP code: 8.09 seconds.
```