

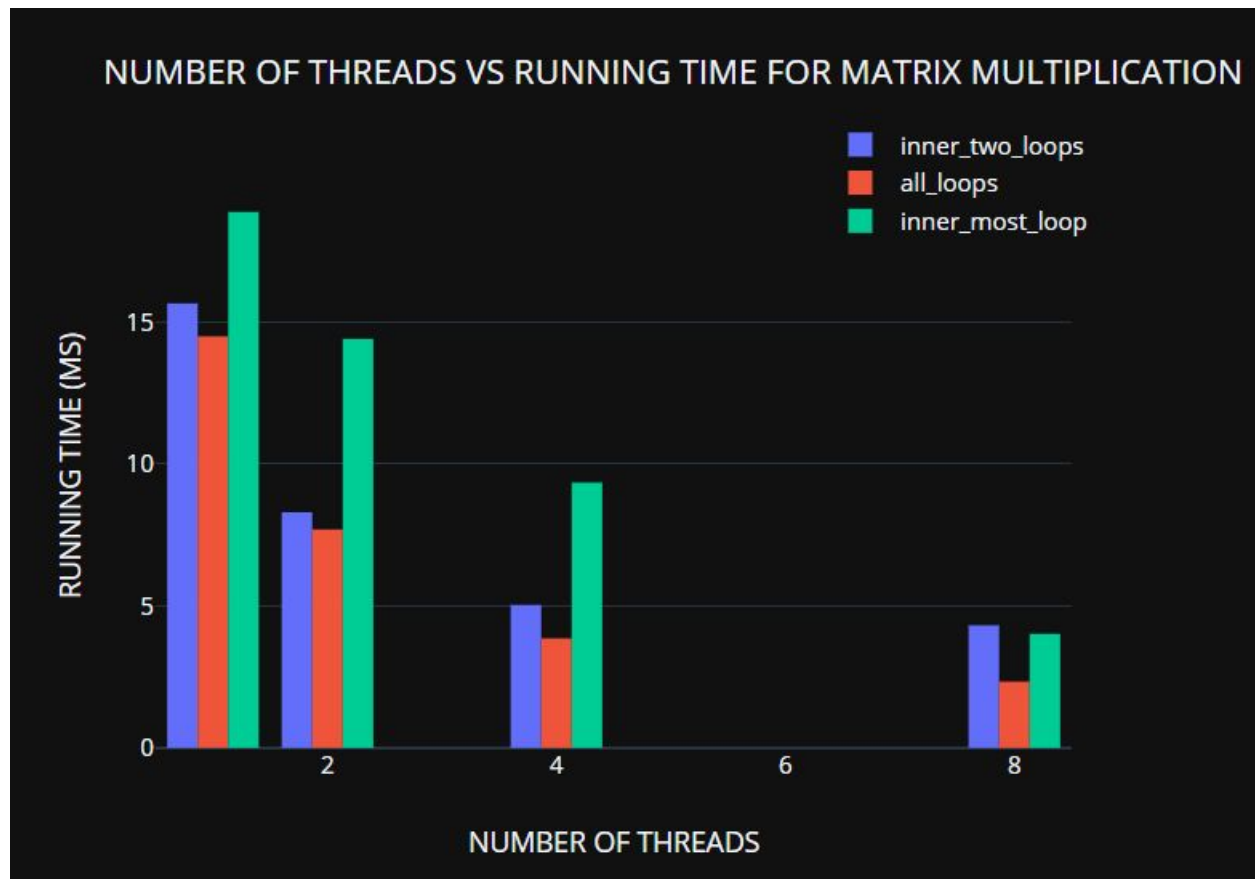
## PROJECT 2

### 1. MATRIX MULTIPLICATION PROBLEM

The matrix multiplication problem has been used and parallelized for three of the test cases. When I tried to parallelize the loops based on three cases as given in the solution, I found that the inner\_most loops run time was higher compared to other two cases and the most efficient one being while parallelizing all three loops. So the runtime for **inner\_most\_loop** > **inner\_two\_loops** > **all\_loops**. But the results for runtime were unstable for various times it was run for different threads. While the results were better in comparison to different cases, but for different number of threads it was unstable, but for some instances the results were recorded and averaged and the graph is plotted. (note: It has not been tested for 6 threads). The program has been tested and tried on different openmp constructs. Creating proper private variables for each of the thread proved to get better results in comparison to the openmp constructs which was generally written for the loops. Seperate files have been created for 3 of the cases and tested for 1,2,4,8 threads.

#### PLOT FOR THE BELOW 3 CASES.

(i) only the innermost loop is parallelized; (ii) the inner two loops are parallelized; and (iii) all three loops are parallelized.

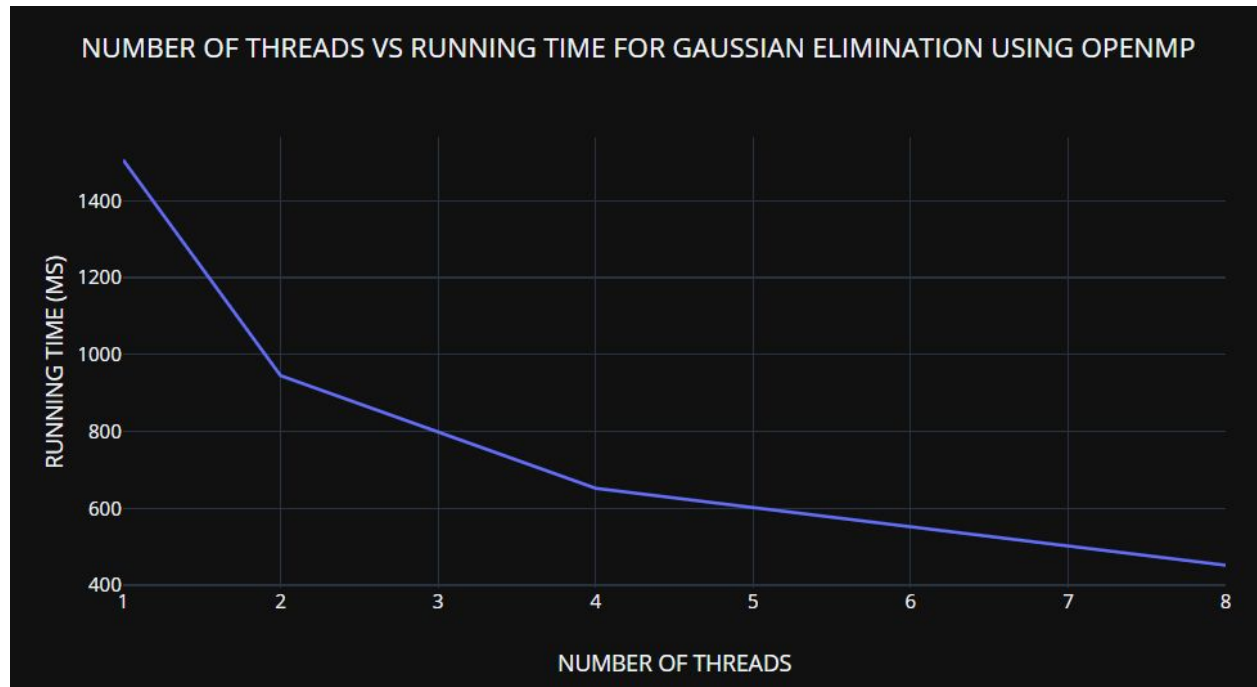


## PROJECT 2

### Gaussian Elimination using Open MP

The program has been tested for 1,2,4 and 8 threads and the run time was greatly reduced as the number of threads were increased. The usage of synchronisation primitive using barrier has been done and have been recorded. The runtime has been increased when the barrier has been used showing that the part of the code has been serialized, but successively as the thread size was increased the deviation in run time was huge, in comparison to without using any synchronisation primitive. This code performed better for large values of matrix dimensions example:500,1000 but the performance was poor was for smaller matrix size, with increment in number of threads. **Runtime for the same program with barrier > runtime of the same program without the barrier.** But there has been cases that **run time with barrier has performed better than the run time without barrier , for higher dimensions.**

### PLOT FOR THE GAUSSIAN ELIMINATION PROBLEM



A makefile has been created and run and all of the outputs were recorded too.