Assignment 3

1.1 Serial Algorithm (MM-ser) Implementation: The algorithm is designed using 3 nested for loops.

1. Assuming m, n and q as 100 the Serial Execution time = 0.030000 seconds.
2. Assuming m, n and q as 200 the Serial Execution time = 0.170000 seconds.
3. Assuming m, n and q as 300 the Serial Execution time = 0.340000 seconds.
4. Assuming m, n and q as 400 the Serial Execution time = 0.930000 seconds.
5. Assuming m, n and q as 500 the Serial Execution time = 1.280000 seconds.

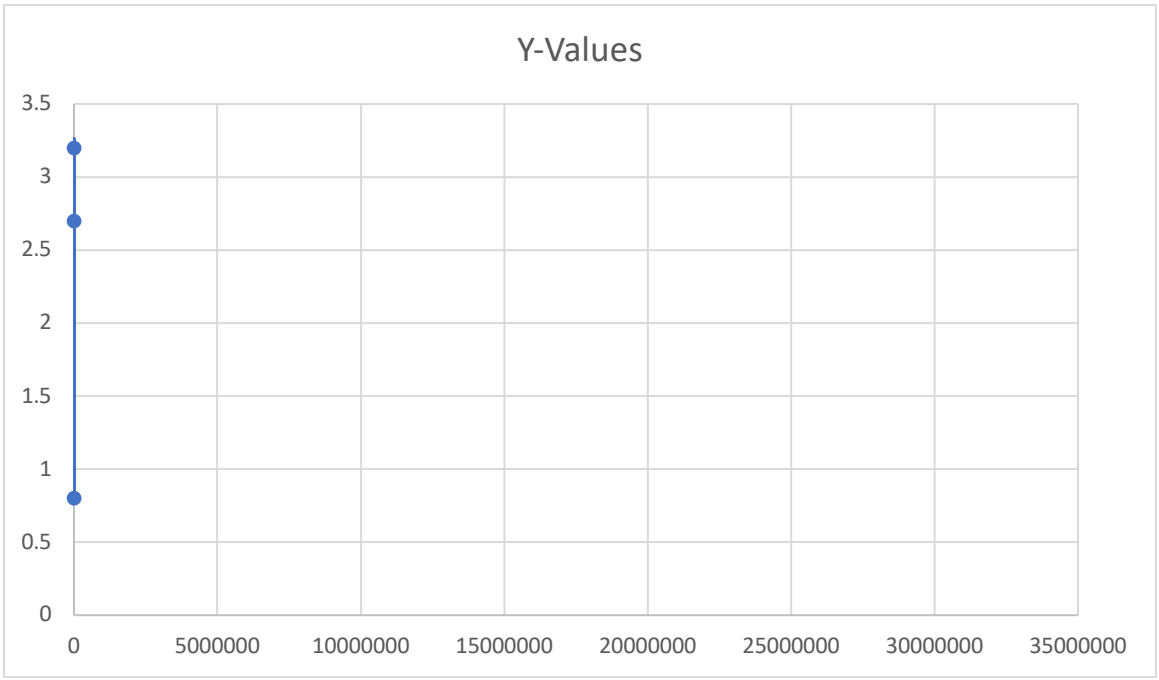
1.2 Simple Parallel Algorithm (MM-Par) Implementation: parallelizing the for loops in the MM-ser implementation using relevant OpenMP directives. This is different from the MM-1D and MM-2D implementation is that there is no explicit mapping of tasks and data items to individual threads

1. Assuming m, n, and q as 100 and p = 4 the parallel Execution time = 0.109215
2. Assuming m, n, and q as 200 and p = 8 the parallel Execution time = 0.505937
3. Assuming m, n, and q as 300 and p = 16 the parallel Execution time = 0.697202
4. Assuming m, n, and q as 400 and p = 32 the parallel Execution time = 0.330076
5. Assuming m, n, and q as 500 and p = 64 the parallel Execution time = 0.465227

1.4 1D parallel Algorithm (MM-2D) Implementation:

1. Assuming m, n, and q as 100 and p = 4 the parallel Execution time = .031966900
2. Assuming m, n, and q as 200 and p = 8 the parallel Execution time =.31967
3. Assuming m, n, and q as 300 and p = 16 the parallel Execution time = .316511
4. Assuming m, n, and q as 400 and p = 32 the parallel Execution time = .4122111
5. Assuming m, n, and q as 500 and p = 64 the parallel Execution time = .518898

Graph :


Y-Values
3.5
3
2.5
2
1.5
1
0.5
0
0
5000000
10000000
15000000
20000000
25000000
30000000
35000000