# Final Year B. Tech., Sem VII 2025-26

High Performance Computing Lab

**Practical No. 7**

**PRN: 22510039**

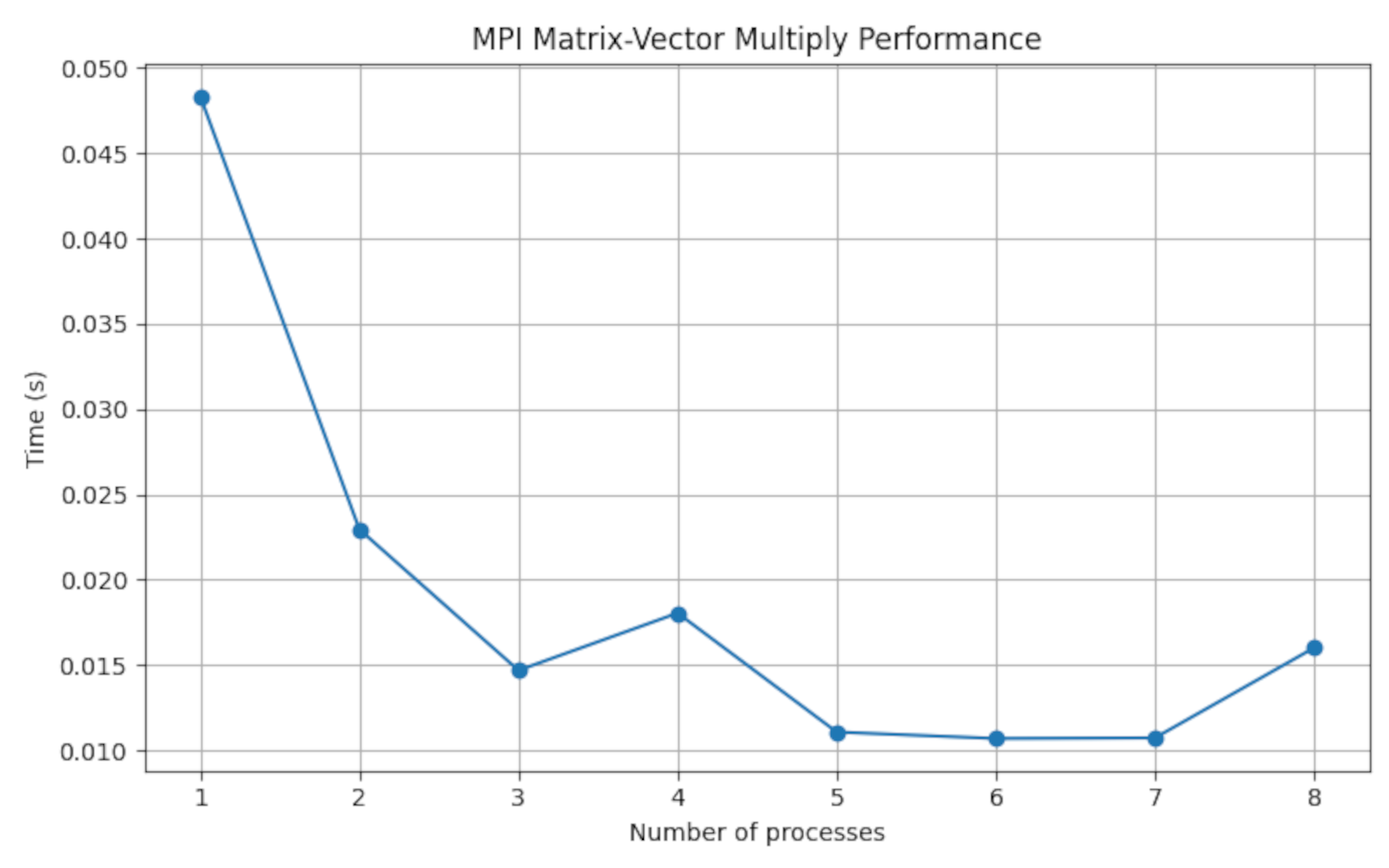
## Implement Matrix-Vector Multiplication using MPI. Use different number of processes and analyze the performance.

Ans.

Code :



Result graph:



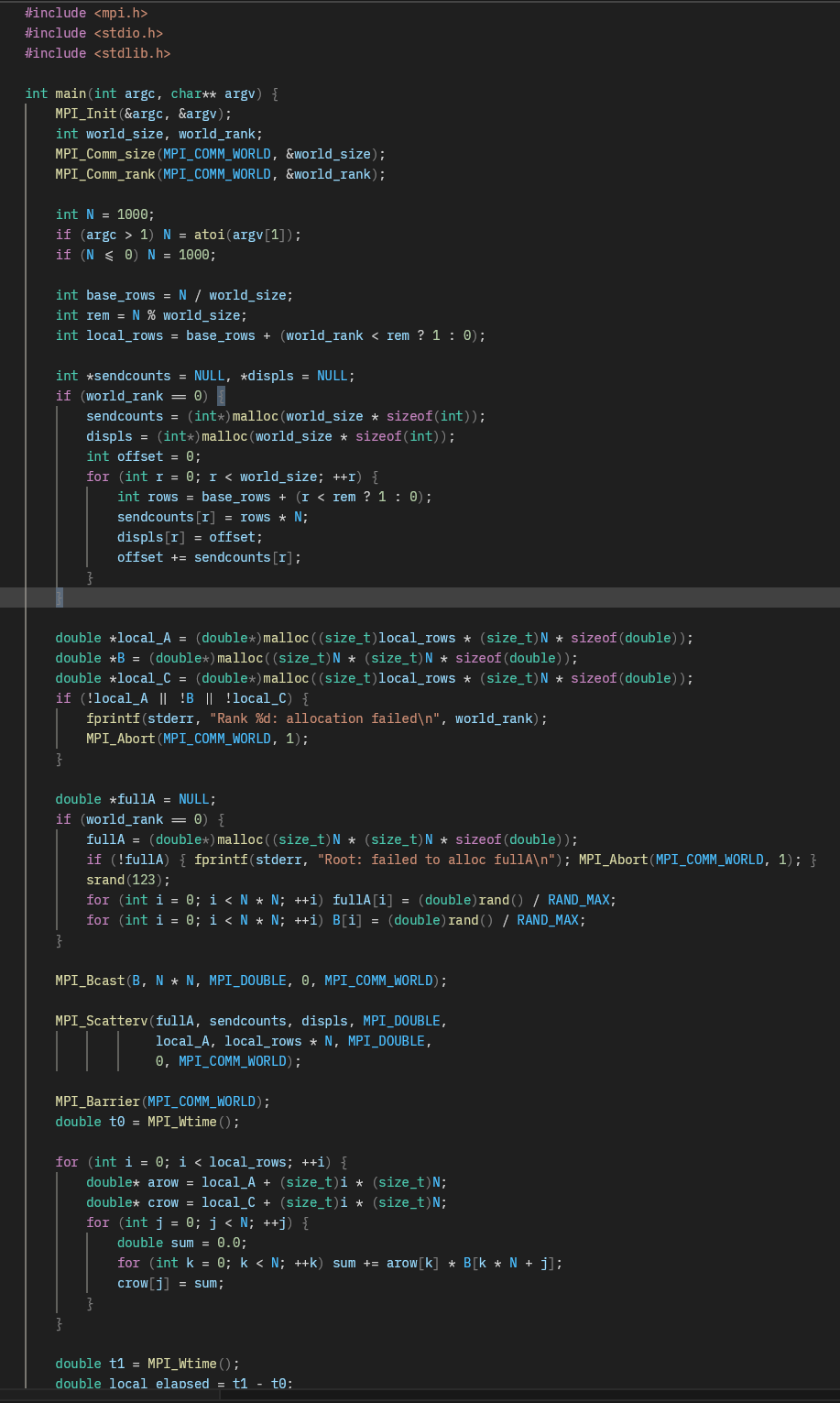
Analysis:

The graph shows that the execution time decreases significantly as the number of processes increases from 1 to about 5, indicating efficient parallelization of the matrix-vector multiplication using MPI. Beyond that point, performance gains flatten and even slightly fluctuate due to communication overhead and load imbalance dominating over computation time. Overall, the trend demonstrates good scalability for small to moderate process counts but diminishing returns as processes increase further.

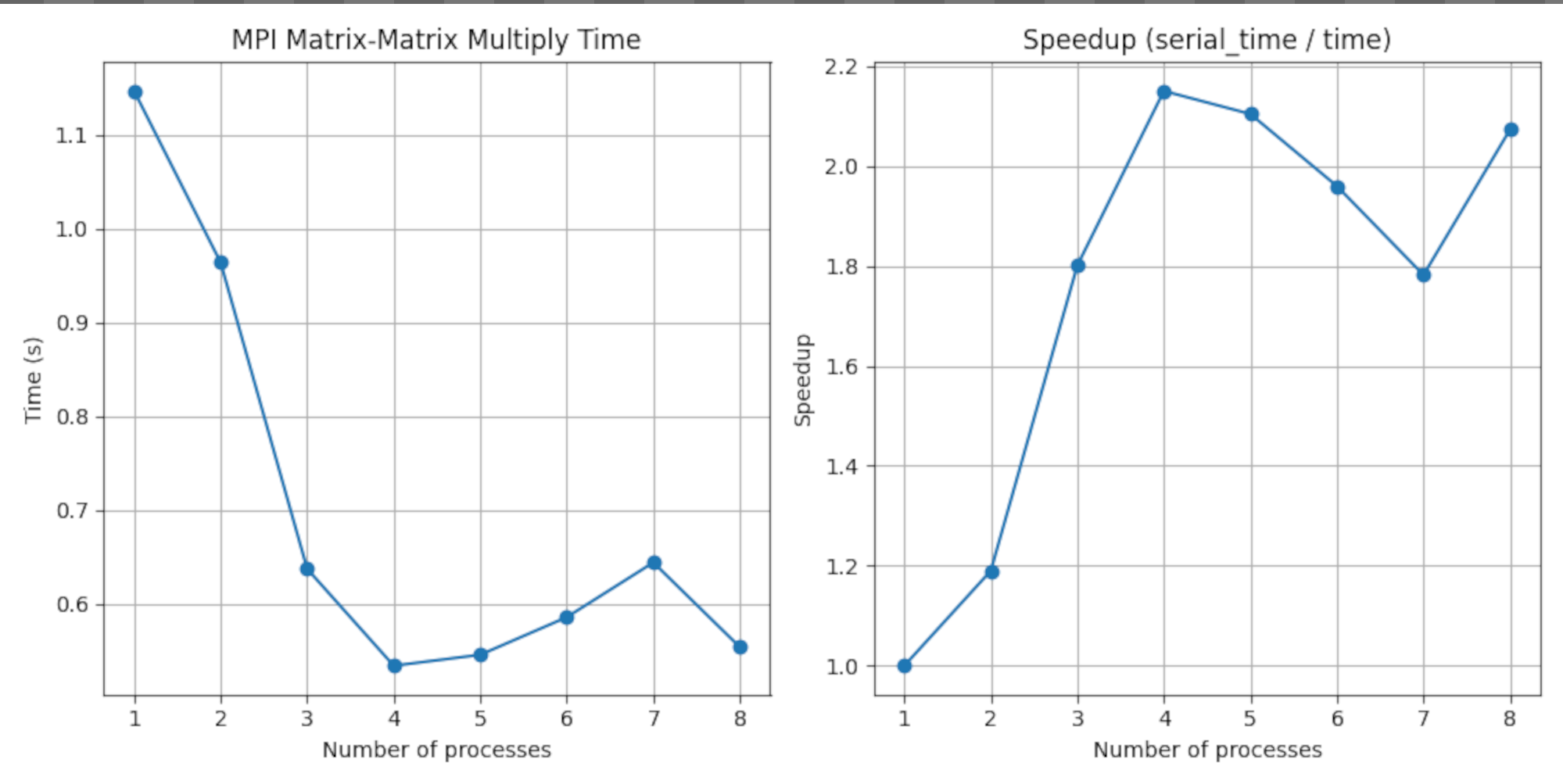
## Implement Matrix-Matrix Multiplication using MPI. Use different number of processes and analyze the performance.

Ans.

Code:



Result graph:



Analysis:

The graphs show the performance of MPI-based Matrix–Matrix Multiplication with varying process counts.

* Left (Execution Time): The time decreases sharply from 1 to 4 processes, showing effective parallelization and workload distribution. After 4 processes, the execution time fluctuates slightly due to communication and synchronization overhead, which becomes more significant as process count increases relative to computation.
* Right (Speedup): Speedup improves steadily up to 4 processes, reaching over 2× faster than serial execution, indicating good scalability. Beyond that, speedup plateaus and slightly drops, showing diminishing returns as inter-process communication starts dominating computation time.

Overall, the results demonstrate that the matrix–matrix multiplication scales efficiently up to a moderate number of processes ( approx 4), beyond which the benefits taper off due to MPI overhead.