**PROJECT REPORT**

**Calculating Area Under the Curve Using Trapezoidal & Simpson Rule**

**Project members:**

1. **K214869 – Yasir Raza**
2. **K213440 – Murtaza Ahmed**
3. **K214915 – Farzeen Khan**

**Submitted to:**

**Miss Ayesha Ali**

**Project Background**

The computation of the area under a curve is a fundamental mathematical task with applications in various fields. However, this process becomes computationally intensive, particularly for complex functions and large datasets. This project aims to address this challenge by leveraging parallel and distributed computing through MPI and OpenMP.

**Objectives**

1. Develop a parallel MPI-based solution for calculating the area under mathematical functions using the Trapezoid Rule and Simpson Rule.
2. Showcase the advantages of parallel and distributed computing in reducing computation time for complex functions.
3. Gain practical insights into parallel computing paradigms, algorithms, and frameworks.

**Scope and Deliverables**

* Develop a user-friendly program for inputting functions and defining integration intervals.
* Implement the Trapezoid Rule and Simpson algorithm in a parallel MPI-based solution.
* Test and benchmark the solution for efficiency, scalability, and accuracy.
* MPI and OpenMP programs for efficient area estimation.
* Efficiency and scalability analysis reports.
* A user-friendly interface for function input and result visualization.

**Methodology**

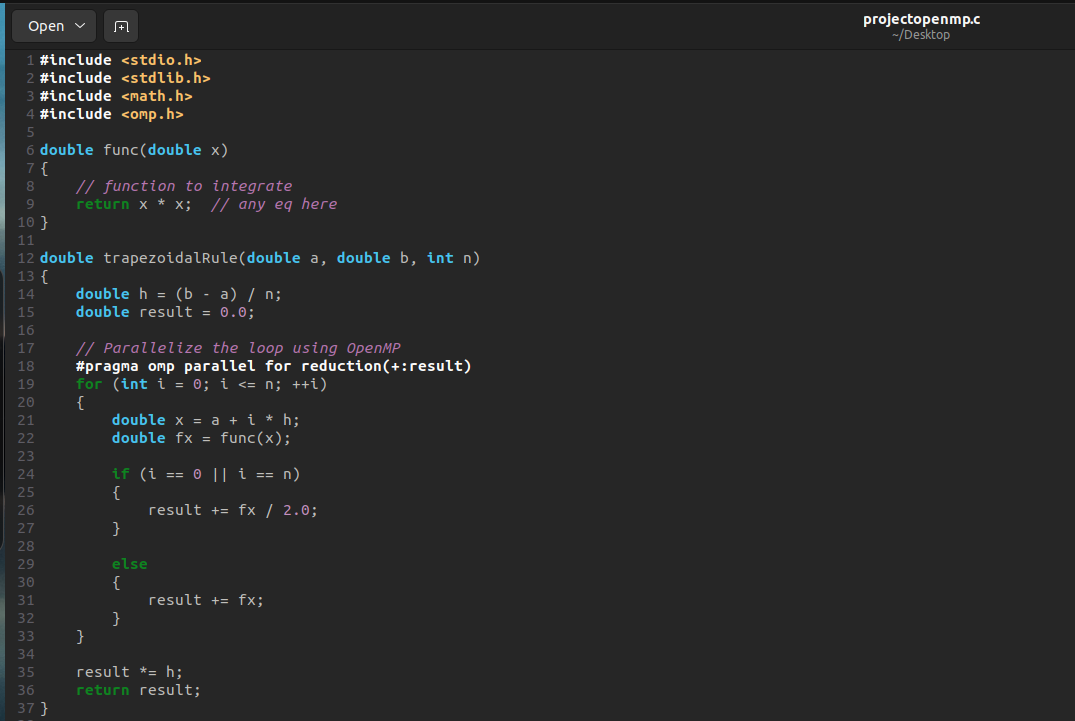
1. **Parallel MPI and OpenMP Solution:**

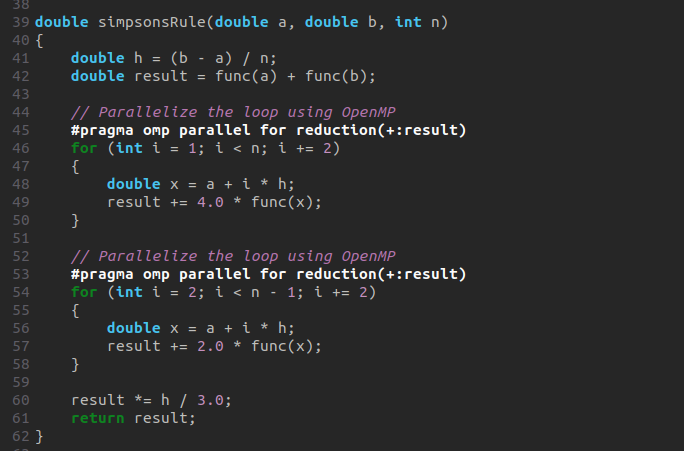
* Develop parallel solutions for both algorithms using MPI and OpenMP.
* Optimize code for efficient workload distribution and parallel execution.

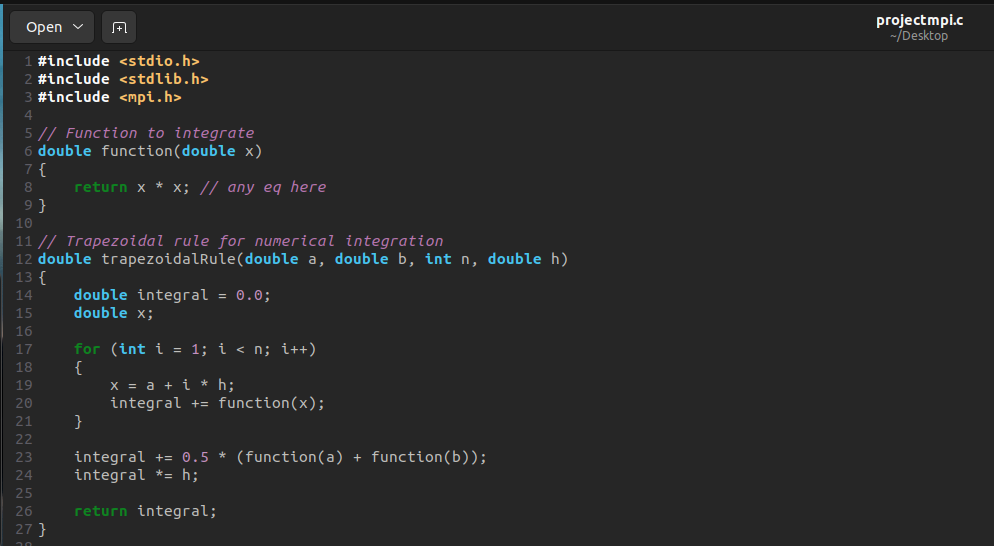
1. **Evaluation:**

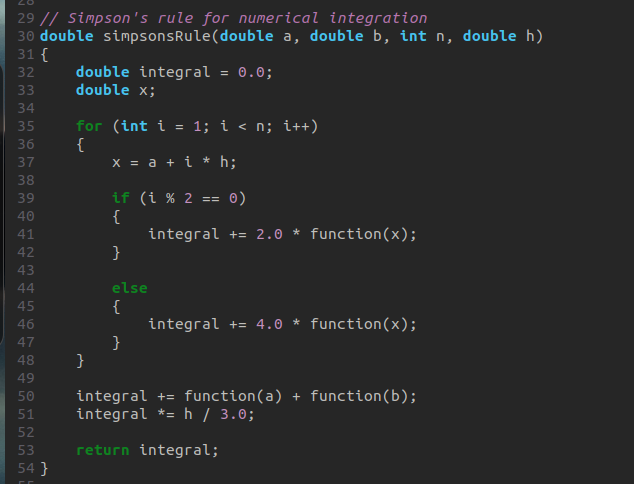
* Evaluate efficiency and scalability across different complexities and dataset sizes.
* Conduct validation tests to ensure accuracy.

**Code Screenshots:**

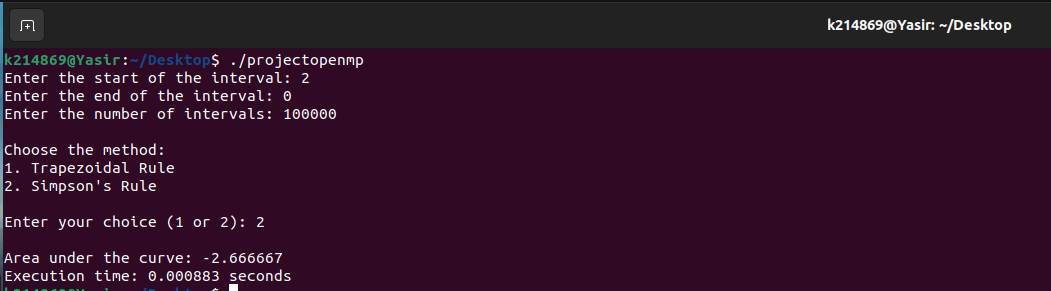
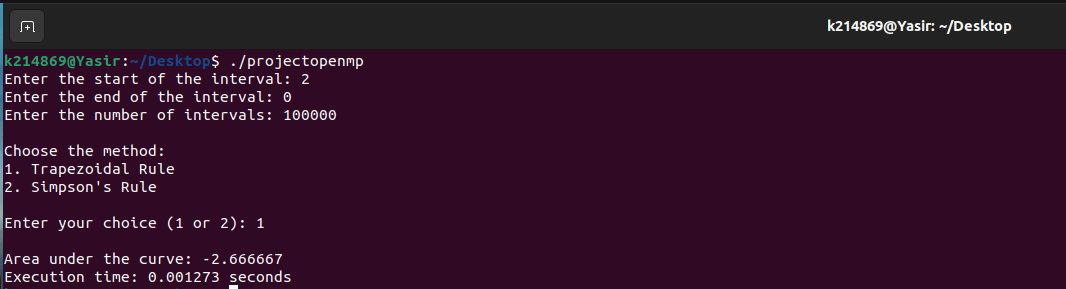
Trapezoidal Rule in OpenMP

Simpson Rule in OpenMP

Trapezoidal Rule in MPI

Simpson Rule in MPI

**Results and Outcomes**

****Outcomes for OpenMP (same data)

Outcomes for MPI (same data)



Notably, OpenMP consistently outperformed MPI, exhibiting shorter execution times on the same datasets and intervals. The parallelization efficiency of OpenMP was evident in significantly reduced computation times, emphasizing its advantage in this specific context.

**Conclusion**

This project successfully leverages parallel and distributed computing using MPI and OpenMP to address the computational challenge of finding the area under a curve. The OpenMp solution significantly reduces computation time, allowing for the efficient handling of complex functions and large datasets. Beyond the scope of this project, this work serves as a valuable educational experience in parallel and distributed computing course. It highlights the importance of parallel computing in accelerating scientific and engineering computations, contributing to our understanding of high-performance computing techniques.