Simulating Heat Distribution in a Metal Plate Using Monte Carlo and MPI

Artur Sahakyan

Erasmus+ WICT

Introduction

This seminar aims to simulate heat distribution (transfer) across a 2d grid. Simulation uses particles, which can then be visualized as pixels with temperature values. The process is stochastic, meaning particles move randomly from the heat source. The farther away particles go from the source, the less heat they carry (depending on conductivity). The process is transient, and it will reach a steady state (in this case thermal equilibrium). The goal of the work is to demonstrate the effectiveness of parallel computation through the experiment.

Methodology

The seminar implementation uses parallel processing techniques to solve the problem more efficiently. 2D grid is partitioned among multiple processes to distribute the tasks into subtasks. Grid is divided into horizontal slices, with each MPI process solving for its own slice. When number of processes increases, it leads to potentially (potentially, since the process is random) faster execution time.

`MPI.COMM_WORLD` is a communicator for representing processes in the MPI environment, `comm.Get_rank()` retrieves the rank (unique id) of the calling process within the communicator. It identifies which slice of grid a process should solve for, `comm.Get_size()` retrieves the total number of processes in the communicator, `comm.Scatter(sendbuf, recvbuf, root=0)` distributes data chunks from root process to other processes in communicator. `comm.Gather(sendbuf, recvbuf, root =0)` combines data slices after the simulation.

Results And Discussion

The results show that for smaller grid size (100) and steps (100) there is no big difference between the code with mpi4py and the one without it. However, once the grid size and steps increase the difference is obvious.

Results for parallel computing

execution time = 0.26 seconds
memory usage = 0.17 MB

grid size = 100
number of steps for computation = 100

with increased parameters

execution time = 9.41 seconds
memory usage = 0.63 MB
grid size = 200
number of steps for computation = 400

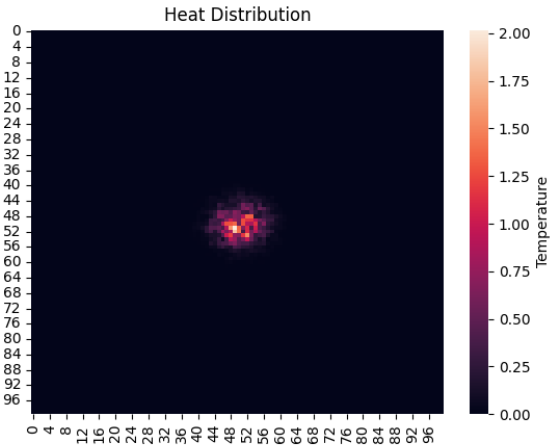
Results without parallel computing

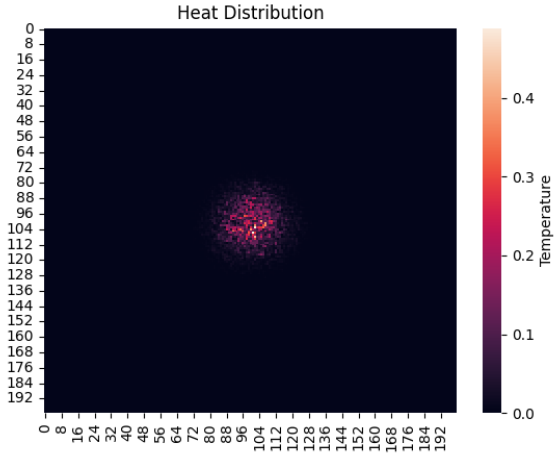
execution time = 0.25 seconds
memory usage = 0.16 MB
grid size = 100
number of steps for computation = 100

with increased parameters

execution time = 15.62 seconds
memory usage = 0.61 MB
grid size = 200
number of steps for computation = 400

Heatmaps for simulations of stochastic heat transfer





It is also worth noting that due to the increase in grid size and steps memory consumption has slightly risen (+2MB) in HPC execution.

Conclusion

The seminar project modelled the heat transfer. Parallel, regular computational techniques were applied and then compared by their execution parameters, results. The goal of showing the effectiveness of parallel processing has been achieved through multiple tests.