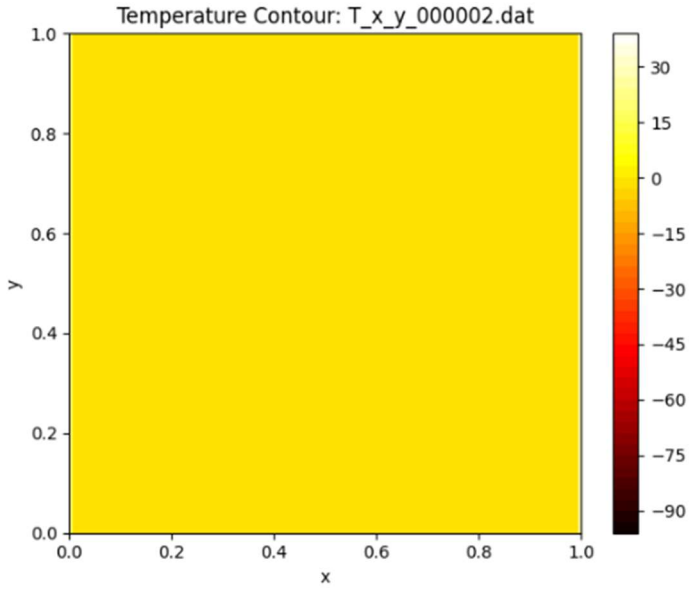
ME5470 : Introduction to Parallel Scientific Computing

HOMEWORK 5 Report

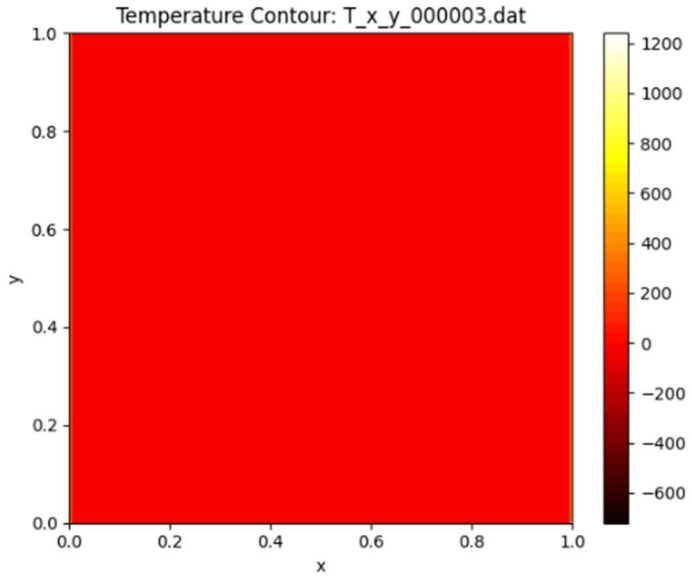
Abhinav kalala

Co21btech11007

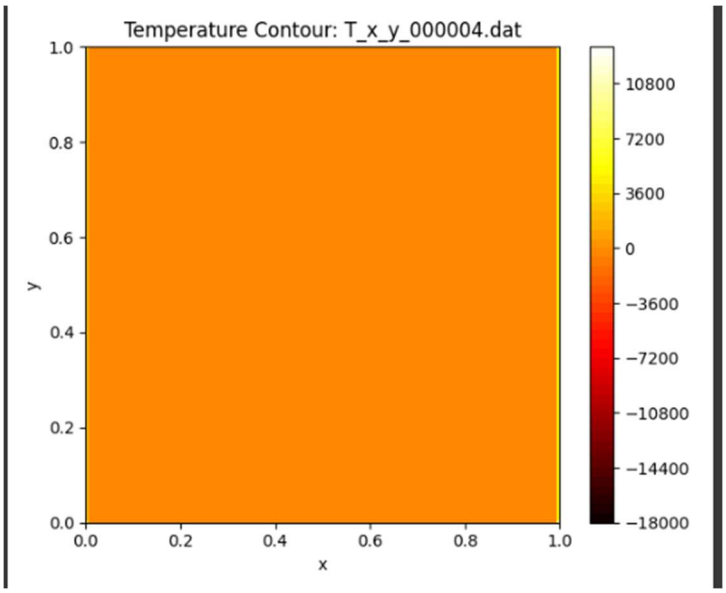
Plots for serial code:



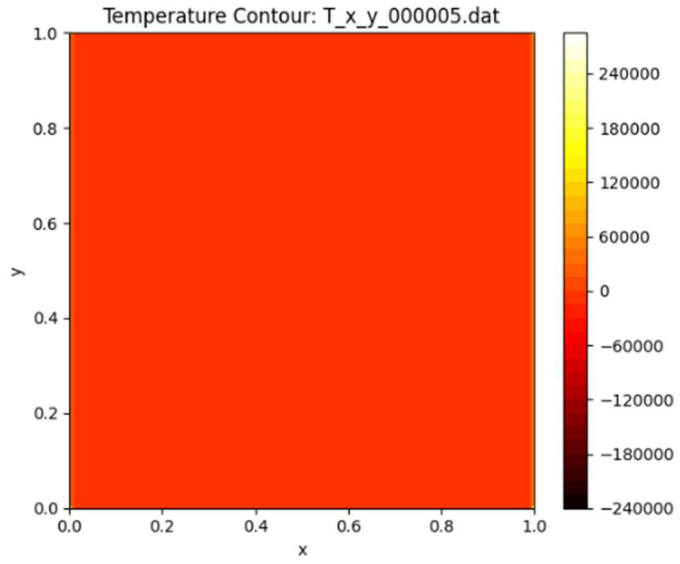
At T_x_y_000002



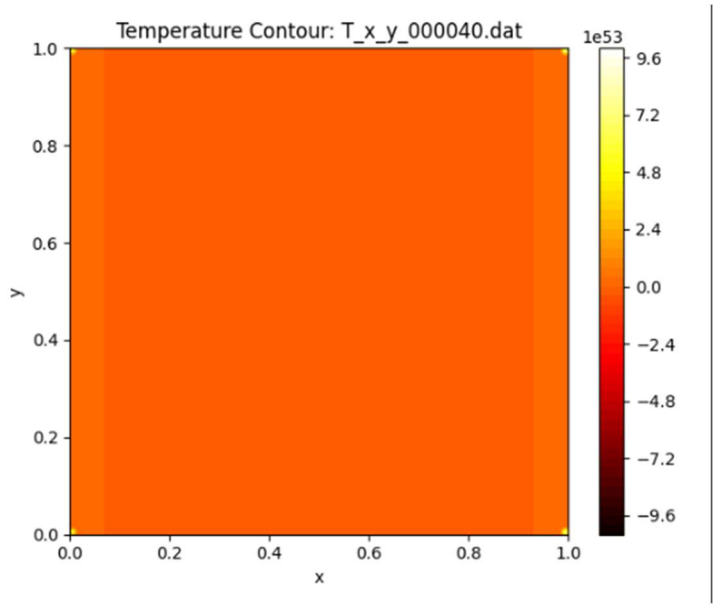
At T_x_y_000003



At T_x_y_000004

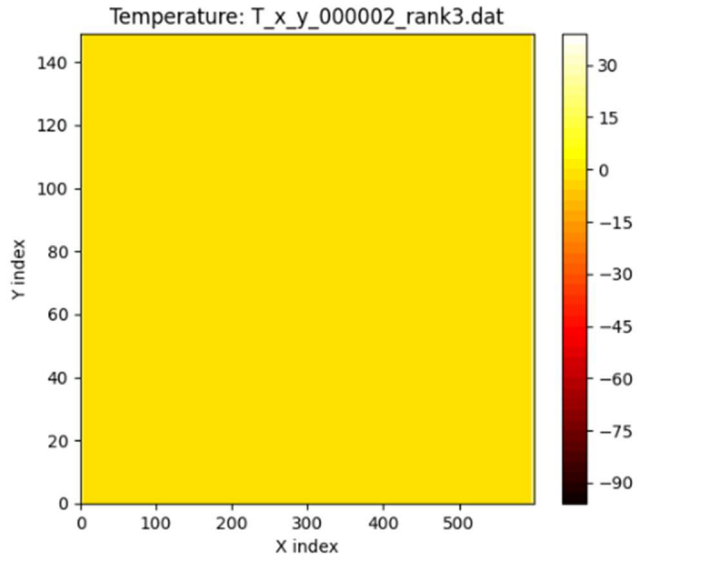


At At T_x_y_000005

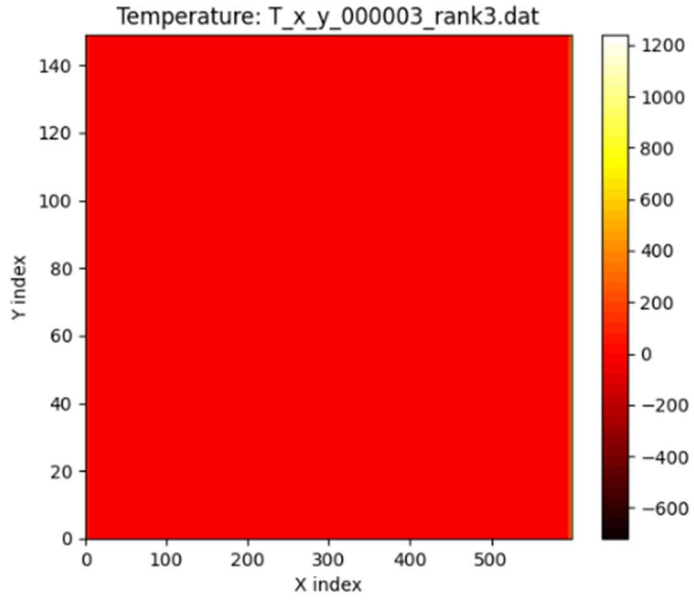


At At T_x_y_000040

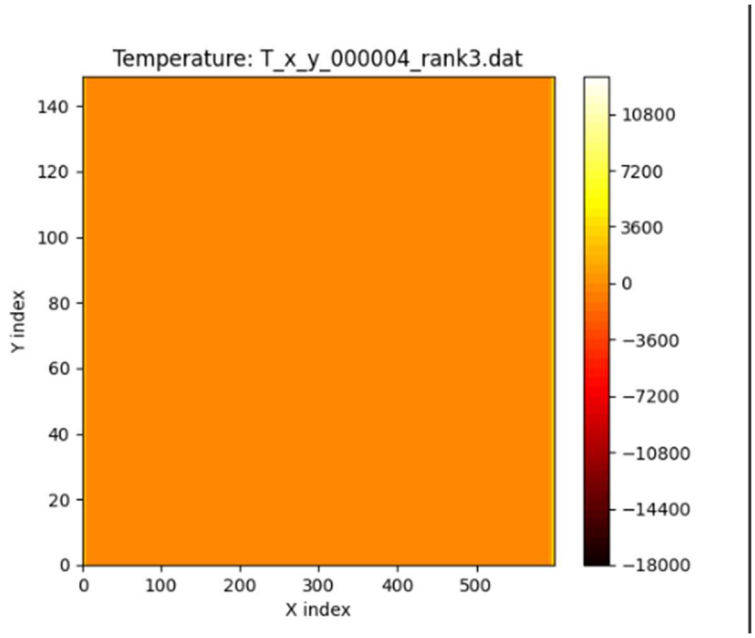
Plots for parallel code:



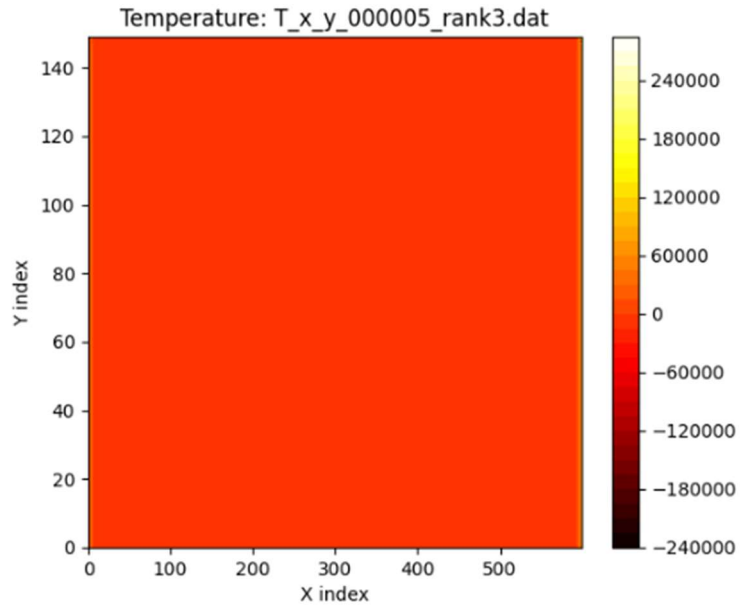
At At T_x_y_000002



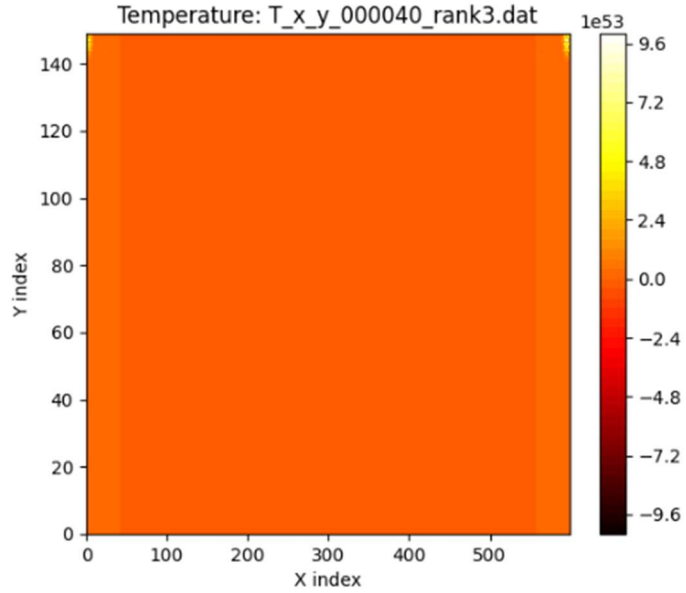
At At T_x_y_000003



At T_x_y_000004



At At T_x_y_000005



At At T_x_y_000040

Yes by seeing the plots obtained by both serial and parallel we can say that the differences should be almost close to machine precision.

Serial Version (hc2d.c)

Implementation Summary:

- The entire grid is stored on a single processor.
- Finite difference method is applied at each timestep to compute new temperature values.

- Boundary conditions are applied directly at each step.
- .dat output files are generated for every time step to visualize temperature distribution.

Output:

- Files: T_x_y_000000.dat, T_x_y_000001.dat, ...
- Each file contains temperature values at each grid point.

Key Parameters:

- Grid: 600x600
- dx, dy calculated from domain dimensions.
- Stability condition: dt should satisfy the CFL condition.

Parallel Version (parhc2d_skel.c)

Implementation Summary:

- The grid is divided among MPI processes along the y-direction.
- Each process handles a block of rows (adjusting for uneven division).
- MPI communication (Sendrecv) is used to exchange ghost rows with neighbors.
- Local matrices T and T_new store temperature for each subdomain.
- Output files are generated per process: T_x_y_000005_rank0.dat, etc.

Communication:

- Top and bottom boundaries require communication with neighboring ranks.
- Left and right boundaries are handled locally.

Output:

- Multiple .dat files per timestep (one per rank)
- Can be post-processed and merged to reconstruct the global temperature field