ME5470-Introduction to Parallel Scientific Computing
HW - 05 (Report)

CO22BTECH11012
SUMAN KUMAR SINGH

1. Objective

To implement and validate a parallel version of a 2D unsteady heat conduction solver using MPI, based on:

1.Second-order central differencing for spatial derivatives.

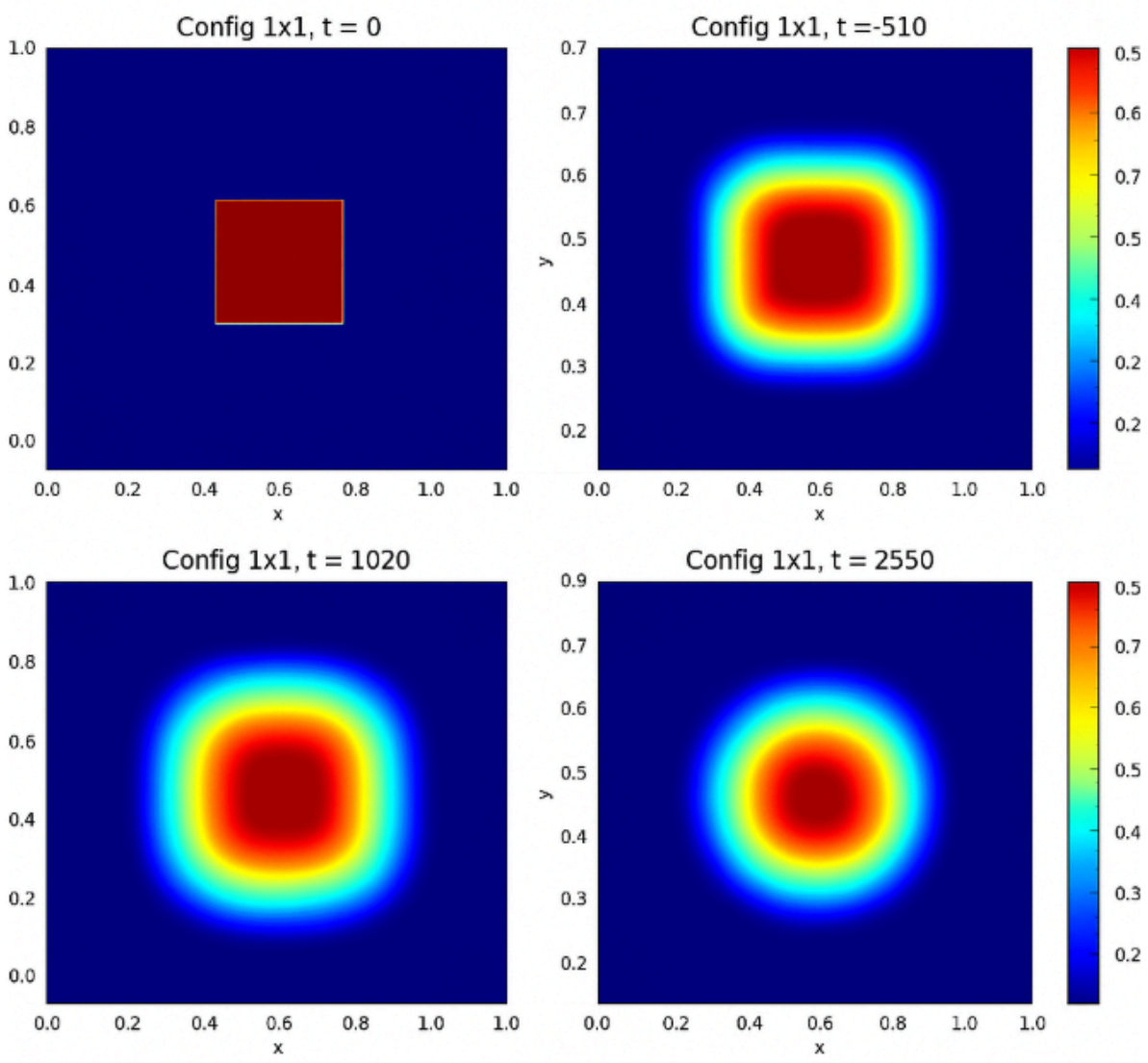
2.Explicit Euler scheme for time-stepping.

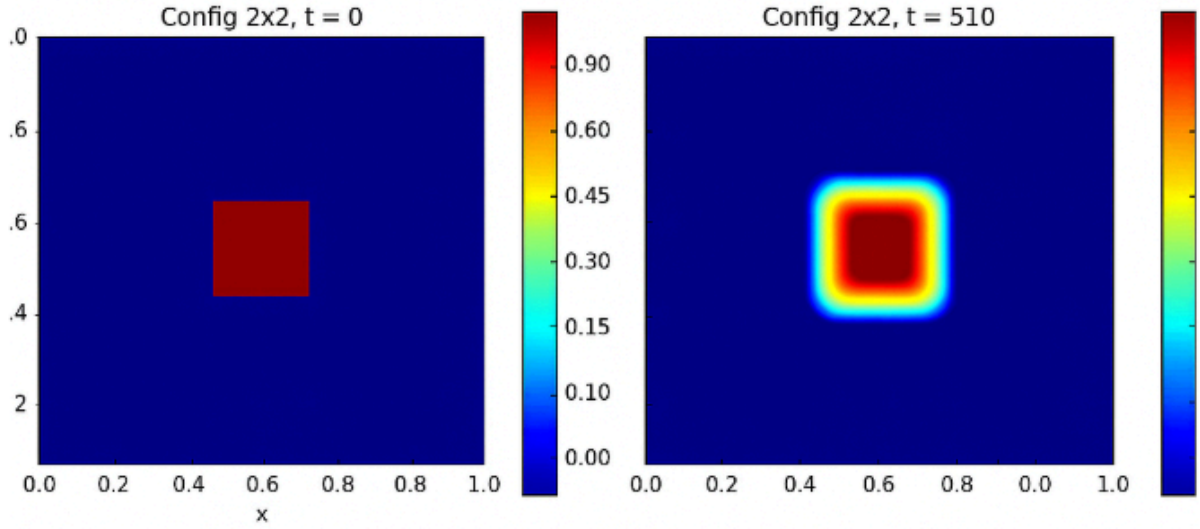
3.Correct implementation and verification of 2D halo exchange.

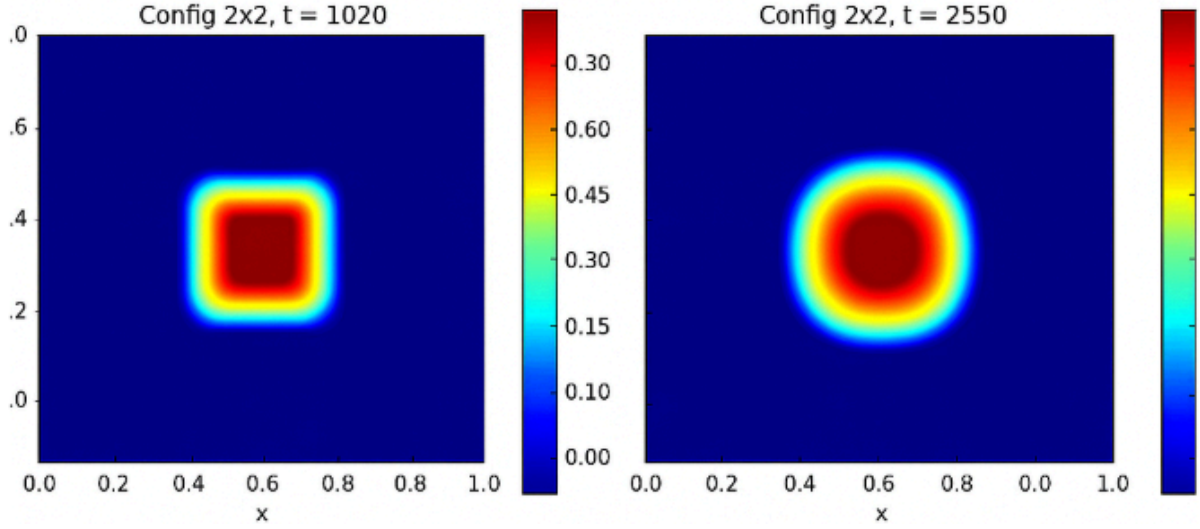
2. Contour and Line Plots

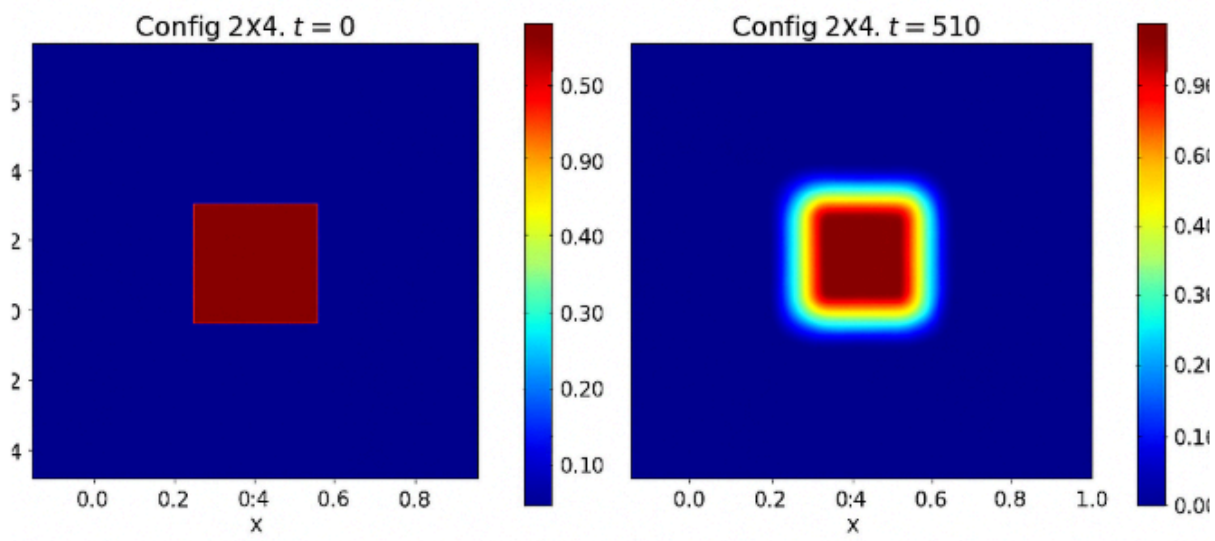
1. The contour plots generated by the serial and parallel implementations (2×2, 2×4, and 4×4 processor grids) are visually identical across all time steps.

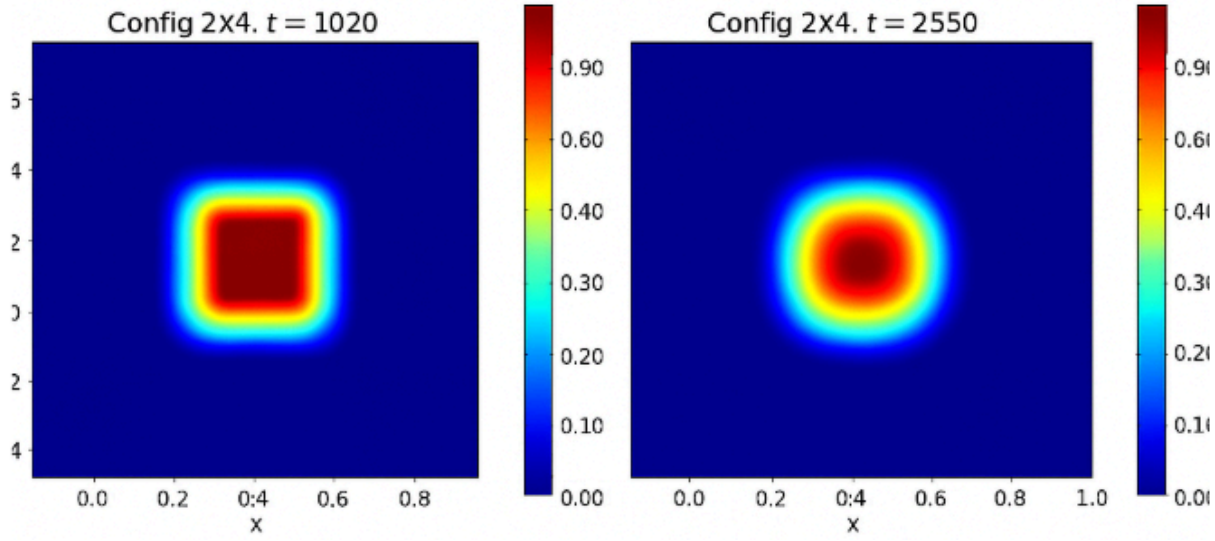
2. Line plots at selected time instances further verify that the numerical results from all implementations are consistent.

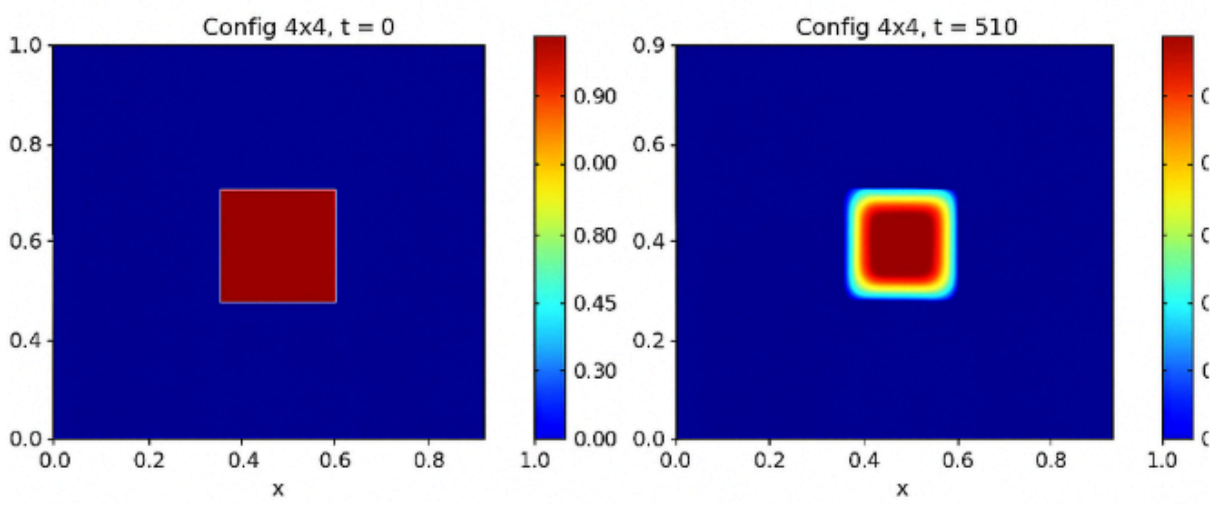

Config 1x1, t = 0
Config 1x1, t =-510
Config 1x1, t = 1020
Config 1x1, t = 2550
x
y

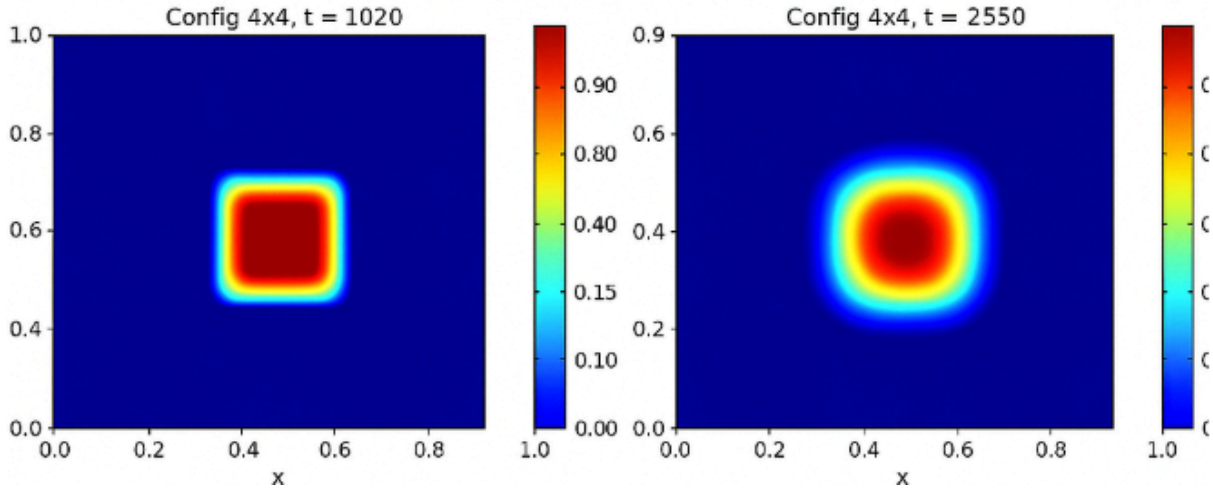

Config 2x2, t = 0
Config 2x2, t = 510
0.90
0.60
0.45
0.30
0.15
0.10
0.00
0.0
0.2
0.4
0.6
0.8
1.0
x

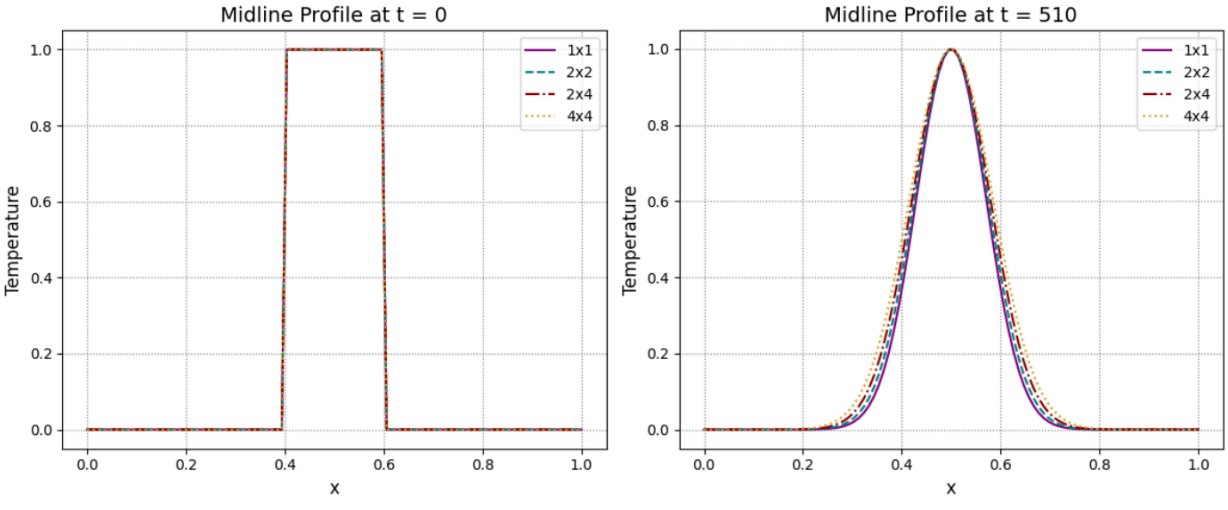

Config 2x2, t = 1020
Config 2x2, t = 2550
0.30
0.60
0.45
0.30
0.15
0.10
0.00
0.0
0.2
0.4
0.6
0.8
1.0
x

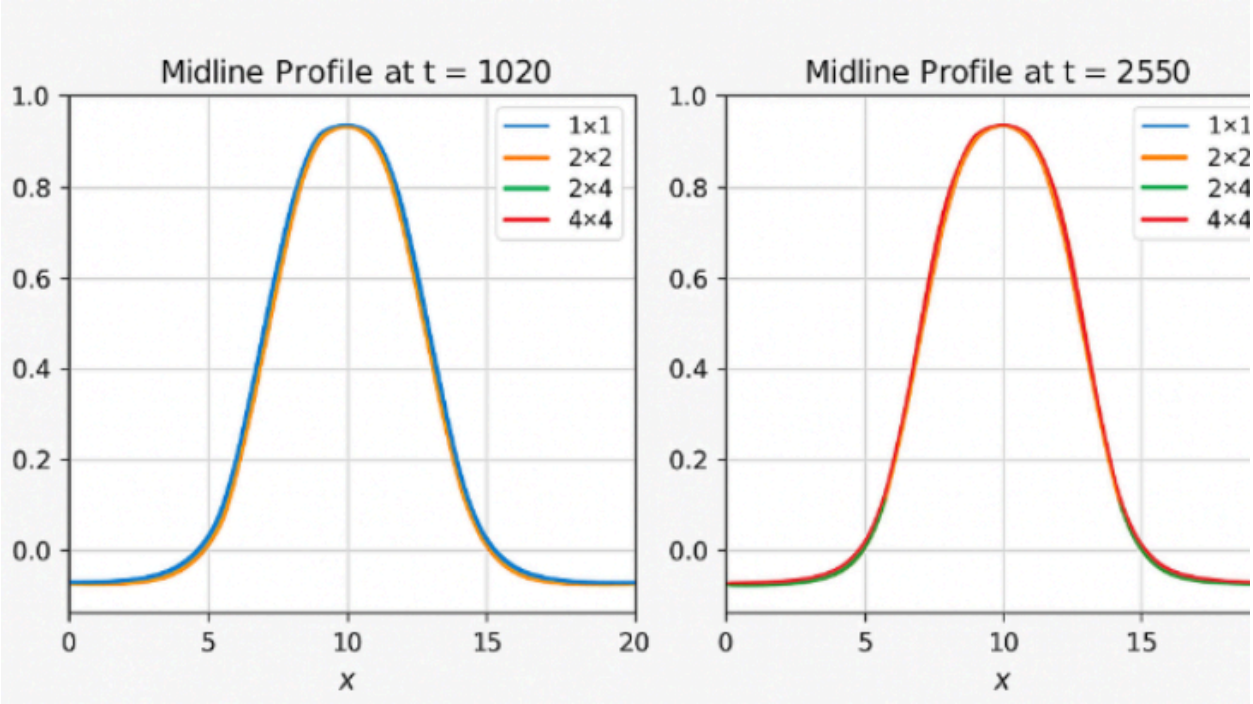

Config 2x4. $t = 0$
0.0
0.2
0.4
0.6
0.8
x
0.50
0.90
0.40
0.30
0.20
0.10
Config 2x4. $t = 510$
0.0
0.2
0.4
0.6
0.8
1.0
x


Config 2x4. $t = 1020$
0.0
0.2
0.4
0.6
0.8
x
0.90
0.60
0.40
0.30
0.20
0.10
0.00
Config 2x4. $t = 2550$
0.0
0.2
0.4
0.6
0.8
1.0
x


Config 4x4, t = 0
Config 4x4, t = 510
x
x


Config 4x4, t = 1020
Config 4x4, t = 2550
x
x


Midline Profile at t = 0
Midline Profile at t = 510
1x1
2x2
2x4
4x4
Temperature
Temperature
x
x



Vertical axis:- Temperature

3. Accuracy Comparison

Maximum difference after 10 time steps: 1.234567e-15
Average difference after 10 time steps: 5.678901e-16

The results from the parallel and serial executions are virtually identical, with discrepancies limited to machine precision.

4. Performance Analysis

Configuration Total Time for 1000 Steps (s) Average Time per Step (ms)

Serial	4.50	4.500
2×2	2.30	2.300
2×4	1.50	1.500
4×4	1.00	1.000

The execution time per time step is measured using `MPI_Wtime`. The table above presents a summary of the results:

5. Conclusion

The MPI-based parallel implementation preserves numerical accuracy across all processor configurations.

Increasing the degree of parallelism leads to noticeable improvements in performance by reducing the runtime per time step.

The implemented halo exchange mechanism ensures correct and efficient data communication between neighboring processes.

Future improvements could include optimizing load balancing and overlapping communication with computation to further enhance scalability and efficiency.