TP 3: Solving the 2D Heat Equation with MPI

October 27, 2023

1 Introduction

In this assignment, you will explore the numerical solution of the 2D heat equation using the Finite Difference Method with the FCTS (Forward-Time Centered-Space) scheme. The 2D heat equation describes the distribution of heat over a 2D spatial domain and is given by:

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{1}$$

Where $u(x, y, t)$ is the temperature at position (x, y) and time t, and α is the thermal diffusivity.

1	1	1	1	1	**1**
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	0	0	0	0	1
1	1	1	1	1	**1**

Figure 1: Initial conditions on 6x6 2D domain

To solve this equation numerically, you will use the FCTS method

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \alpha \left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2} \right) \Delta t \tag{2}$$

A sequential implementation is provided in the file `heatSeq.cpp`. The code simulates the heat equation's evolution over time. Your task is to extend this code to work on multiple processor cores using MPI.

2 Instructions

1. Download the sample code `heatSeq.cpp` provided with this assignment.
2. Study the provided sequential code (`heatSeq.cpp`) to understand the algorithm and how the 2D domain is discretized.
3. Parallelize the code using MPI. You'll need to divide the domain among multiple processes and implement the necessary communication between them.
4. Test your MPI implementation with various domain sizes and processor counts. Use the following parameters as a starting point:
   - Domain size: Vary the dimensions of the 2D domain (e.g., 64x64, 128x128, 256x256).
   - Processor count: Vary the number of MPI processes (e.g., 1, 2, 4, 8).
   - You can assume that the size of the domain is always divisible by the number of processors
   - The number of time iterations can be fixed to be sufficiently large 10^5
5. Record the execution times for different domain sizes and processor counts. Analyze the results and discuss the following:
   - How does the execution time change as the domain size increases?
   - How does the execution time change as the number of processors increases?
   - Discuss the trade-offs and challenges of parallelizing this problem.
6. Write a report summarizing your findings and observations. Include any insights you gained from the analysis of the results. Include graphs to support your observations

3 Submission Guidelines

Submit your assignment as a ZIP file containing the following:

1. Your modified C++ code (with MPI parallelization)(include comments) .
2. A Makefile (No Makefile means no points for the code)
3. A PDF report summarizing your findings and observations.

Please make sure to include your name and the due date in your report.

4 Resources

You may find the following resources helpful for this assignment:

- MPI Documentation: `https://www.mpi-forum.org/docs/`
- MPI Tutorials: `https://mpitutorial.com/`