CAC Instructor(s): Fernando Hernandez
Course Instructor: Stephen Hughes

Due date: Sun 14th March (midnight) [**nominal deadline**]

Topics: CAC, MPI, Parallelized Heat Equation, SLURM

Task: go through the questions and coding exercises below and then write up a **small physics report** in the LaTeX template provided.

LaTeX File of Assignment: (for convenience with your own write up) PS4

Marks: Codes (12), Report (8), Results (10). Total: (30)

We decided to keep the LaTeX part of this assignment, as it is important for you to explain the results, but it should not be an overly long report, and not too many graphs and equations to show.

Reminder: All codes must run under Python 3.x. and for this assignment should run on CAC.

Background

The extension of the simple approximation of the 1D heat equation to more than one dimension is straightforward. Assuming that we have the same heat conductivity in both directions, and ignoring any external source terms we have:

$$\frac{\partial U}{\partial t} = D\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right), \tag{1}$$

where D is the thermal diffusivity in the appropriate units. This can be discretized similar to how we did the 1d case, yielding

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

We will use MPI parallel code for the 1D diffusion equation, as well as the 2D serial code (discussed in the lectures) as starting points. In this assignment, we will write a parallel program for the 2D case. The parallelization should be done in one dimension only, which means that each process computes a "strip" of values for U in each time step.

Communication is necessary when "neighbouring" values that are computed with another process are required. For the 1D case, this communication was a single scalar value. In the 2D case, it will be a one-dimensional array of values.

Enable the program to take "snapshot" of the computed 2D distribution in the form of contour plots. This will require additional communication for those specific time steps. Use Python packages such as `matplotlib` for this, which you are already familiar with. Save these at high quality pdf files with physics-journal-friendly fonts, etc. You can also save data, and then make the plots later, such as on your own computer (but it is nice to save a few snapshots juts to check things are working).

Deliverables

1. Write a Python/MPI program for parallel computation of the 2D diffusion equation with variable boundary conditions and initial values.

2. Examine the time development of a test case with the following conditions:

   - $0 \le x, y \le 20.48$ (units of x, y at mm, time units are seconds).
   - Initial conditions: $U_0 = 2000 \cos^4(4r)$ if $r < 5.12$ and $U_0 = 300$ K if $r \ge 5.12$, where r is the distance from the centre of the square domain.
   - Boundary conditions: $U = 300$ K at the edges of the domain.
   - $D = 4.2$ mm^2/s.

3. For a grid with 1024×1024 points, conduct scaling experiments based on internal timing and 100 (or 1000) time steps with the above settings. Report values excluding I/O and separating computation from communication. Do so for 1, 2, 4, 8 processes on a single node, and for 1, 2, 4 nodes with 8 processes per node. Investigate speedup and efficiency for these and report them in table/graph form.

4. Follow the run for 1000 time steps and produce contour plots of the domain for step 0, 100, 200 $\cdots$ 1000.

5. For the "snapshot" time steps, communication to rank 0 is required. UseGatherv, Send/Receive, or Reduce to do this.

Remarks

- The time steps should be adapted to the grid density and diffusity. This means that you should use a value of less than $\Delta t_{\max} = \Delta x^2 \Delta y^2 / (2D(\Delta x^2 + \Delta y^2))$. Larger values will cause stability issues.
- The simulation should run long enough to show the "washing-out" of the ripples in the initial condition.

Any questions or concerns on the assignment or CAC, please first email Fernando: f.hernandezleiva@queensu.ca