Exercises in Parallel Programming in Fortran and C/C++

Dr. Sara Collins SoSe 2021

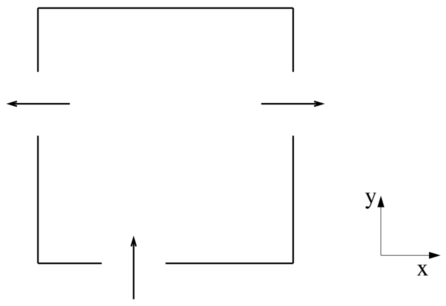
Project

Please consider the following when completing the project.

- You may write your program in either Fortran, C or C++.
- What you submit should enable all parts of the project to be produced and illustrated without requiring modification to your code. Instructions should be provided on how to run the code and visualise the output.
- The program should be able to run on the CIP-pool machines (where the program will be tested when it is graded) and any output should be able to be visualised on these machines.
- The project must be *entirely your own work*. For example, using another person's code, even with some modification is not acceptable.
- Some marks will be given for good programming practice and well organised code.
- If the program consists of more than one file a makefile should be provided.

A simple fluid dynamics simulation

The aim of the project is to perform a simple fluid dynamics simulation of an incompressible fluid in 2-dimensions. The fluid is flowing within a square cavity with one inlet and two outlets:



We work with the *stream function* ϕ, which satisfies Laplace's equation

$$\Delta\phi(\vec{r}) = 0 \qquad \vec{r} \in \Omega, \quad \Omega \subset \mathbb{R}^2 \tag{1}$$

where $\vec{r} = (x, y)$,

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2},$$

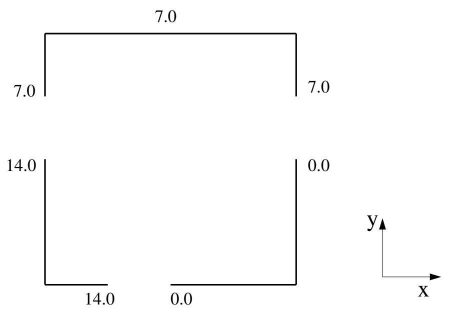
and the cavity has length L ($0 \le x, y < L$). The flow velocities in the x and y directions are given by

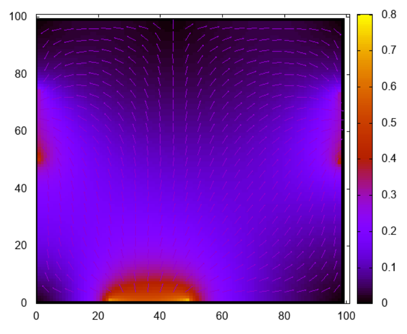
$$v_x = \frac{\partial\phi}{\partial y}, \quad \text{and} \quad v_y = -\frac{\partial\phi}{\partial x}, \tag{2}$$

respectively.
The boundary conditions on ϕ are

$$\begin{array}{lll}
y = 0 & \phi(\vec{r}) = 14 & 0 \le x \le L/4 \\
 & \phi(\vec{r}) = 14 - (x - L/4)56/L & L/4 < x < L/2 \\
 & \phi(\vec{r}) = 0 & L/2 \le x < L \\
x = L & \phi(\vec{r}) = 0 & 0 \le y \le L/2 \\
 & \phi(\vec{r}) = (y - L/2)28/L & L/2 < y < 3L/4 \\
 & \phi(\vec{r}) = 7 & 3L/4 \le y < L \\
y = L & \phi(\vec{r}) = 7 & 0 \le x \le L \\
x = 0 & \phi(\vec{r}) = 14 & 0 \le y \le L/2 \\
 & \phi(\vec{r}) = 7 + (3L/4 - y)28/L & L/2 < y < 3L/4 \\
 & \phi(\vec{r}) = 7 & 3L/4 \le y < L
\end{array}$$

The boundary conditions are displayed below on the left and the resulting flow (after solving Laplace's equation) on the right.





Use the Jacobi algorithm to find the function ϕ with the stopping criterium:

$$\left(\sum_{x,y} \left[\phi^{k+1}(x,y) - \phi^{k}(x,y) \right]^2 \right)^{1/2} < \epsilon$$

where $\phi^k(x,y)$ is the field after k iterations of the algorithm and ϵ is provided by the user. Use ϕ and equation 2 to compute the flow velocities.
Implement the above 2-dimensional problem as a parallel program using MPI:

- First implement a serial version of the program (without parallelisation) that has the correct functionality which will serve as a reference for the parallel program.
- The 2-dimensional grid should be parallelised in both dimensions. The size of the grid should be input to the program and the distribution of the grid to the processes should be performed automatically.
- Modify your parallel program to include OpenMP directives (= hybrid parallelisation: MPI+OpenMP). Hybrid programs can be compiled and run using (for the example of a Fortran program where the number of threads is not fixed in the code)

  ```
  $ mpif90 -fopenmp flow.f90
  $ export OMP_NUM_THREADS=4
  $ mpirun -np 10 ./a.out
  ```

- The output of the program should be a single binary file containing the values of the flow velocities ($v_x(x,y)$, $v_y(x,y)$). Use MPI-IO to achieve this. The ordering of the data within the file should be: $v_x(0,0), v_y(0,0), \ldots$

The binary file can be displayed in gnuplot using the gnuplot script `binary.gnu` available on the course website. The direction of the flow velocities is indicated by the arrows, while the magnitude of the flow velocity is shown via the background colour. The script assumes the binary data file name is `vec.bin`. If you output is in double precision then you should change `float32` to `float64` in the gnuplot script.
For debugging purposes, the script `ascii.gnu` is also provided which will display the flow velocity in the same way for data stored in text format (x, y, v_x, v_y):

```
2 2   -0.899791455490906E-02    0.904296083108846E-02
3 2   -0.180403266388467E-01    0.911107797690214E-02
4 2   -0.271728012617070E-01    0.922646355853196E-02
....
```

in a file with the name `vec.txt`.