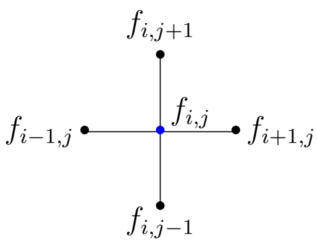
Parallelisation of Diffusion Equation

Date: 16.08.2024

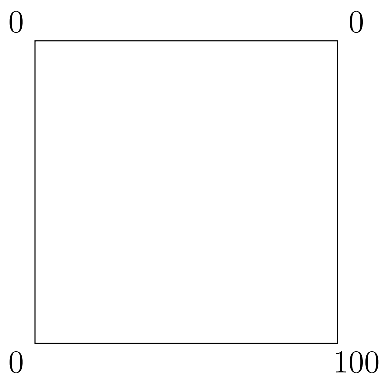
The template code is found in `$PROJECT_training2423/materials/tutorial`. The code loops through all the grid points that are discretised evenly across a square 2D geometry, while updating the temperature value of each point by averaging the adjacent 4 points with the following expression:

$$f_{i,j} = \frac{1}{4}\left(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}\right) \tag{1}$$

The indices i and j refers to the grid numbering in x and y direction respectively. It should be mentioned that Eq. 1 is a simplified equation to demonstrate diffusion.

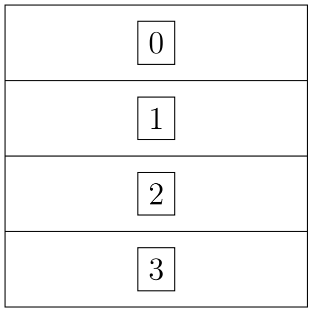


In this numerical setup, the 2D square geometry is discretised into 1000 points in both the x and y direction respectively. The initial condition of the grid points is such that the temperature values are all zero. Only the left and bottom boundaries of the 2D square domain is defined with constant normalised temperature values. The temperature increases linearly from 0 to 100 starting from the bottom left to the bottom right corner of the domain. The temperature profile is similarly increasing from the top right to the bottom right corner.

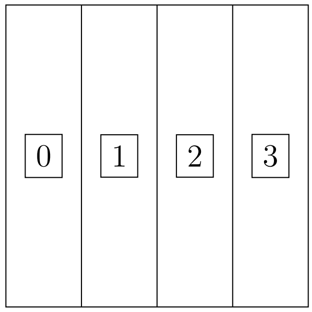


Temperature boundary condition sketch.

For ease of implementation of the MPI parallelisation and load-balancing, the grids are distributed to participating ranks in slices. The following images show the slices and the corresponding MPI ranks that are responsible for the temperature values update.



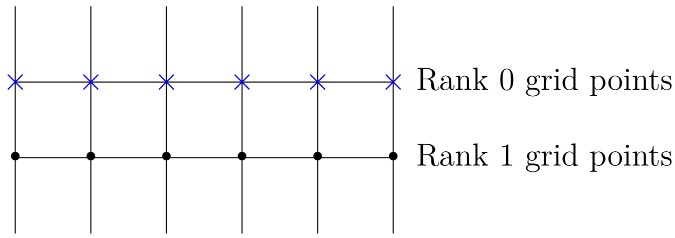
Slice orientation and rank distribution for 4 MPI ranks in C implementation.



Slice orientation and rank distribution for 4 MPI ranks in Fortran.

The code already includes the initialisation and distribution of the grid points to respective ranks, as well as the looping update of the grid point values.

The task is to program the MPI calls that handle the communication of ghost grid points between each rank. Ghost grid points are a subset of points to provide a rank's neighbours the updated values for the next iteration of temperature updates. This subset of points will have to be sent to the neighbour before the updates are calculated.



Aside from the actual grid points which Rank 1 is responsible for, it needs to also hold ghost points to store the values from Rank 0 before temperature update can be done. The opposite is true.