Parallel Computing: Exercise 3

- Implement a parallel heat transfer simulation using a 2-dimensional finite difference Gauss-Seidel successive over-relaxation scheme as described in the lectures
- Use a square grid of size *NxN*
  - the value of *N* is given as input by the user
- Use a 2-dimensional process grid with *P* = *qxq* processes
  - 2-dimensional block decomposition
  - *N* is not necessarily evenly divisible by *q*
- Use safe communication between the processes
  - no risk for deadlocks, regardless of the message size

Graphical output

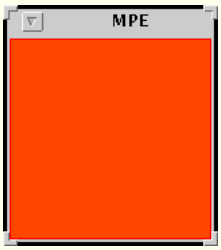
- Present the temperature distribution graphically in a MPE graphics window
  - one point corresponds to one pixel
- Use a colour array of 64 colours
  - the two first colours are *MPE_WHITE* and *MPE_BLACK*
- Update the display every *k* iterations
  - *k* can be 10–20
- Each time the graphical window is updated, check if the user has clicked in the window
  - left button – print out the temperature value of the point
  - middle button – display the iteration number
  - right button – terminate

Termination

- The computation terminates either
  - when the computation has converged
    no temperature value has changed more than ε since last iteration
  - when the user wants to terminate the computation
    the user clicks with the right button in the output window
- Use as convergence criteria for instance ε = 0.001
- Use as the over-relaxation parameter ω = 1.2

Examples

- All four edges initialised to the edge temperature



- Three edges initialised to the edge temperature