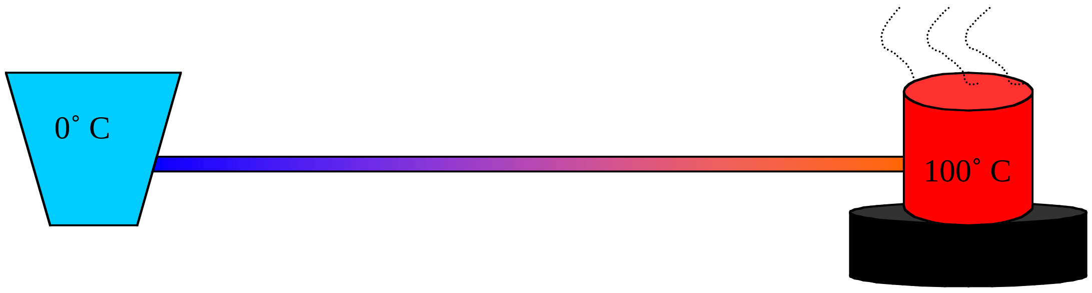
High Performance Computing

Exercise 2. One-dimensional finite difference calculation

A thin, heat-conducting metal wire has the left end connected to a bucket of ice water with a temperature of 0˚ C and the right end connected to a cattle of boiling water at 100˚ C.



The temperature of the wire is represented as a one-dimensional array *X* of temperature values in *N* discrete, equally spaced points along the wire. The temperature at the left end, *X[0]* is always 0˚ C and the temperature at the right end, *X[N-1]*, is always +100˚ C. Initially, all other points have a temperature of +20˚ C (room temperature).

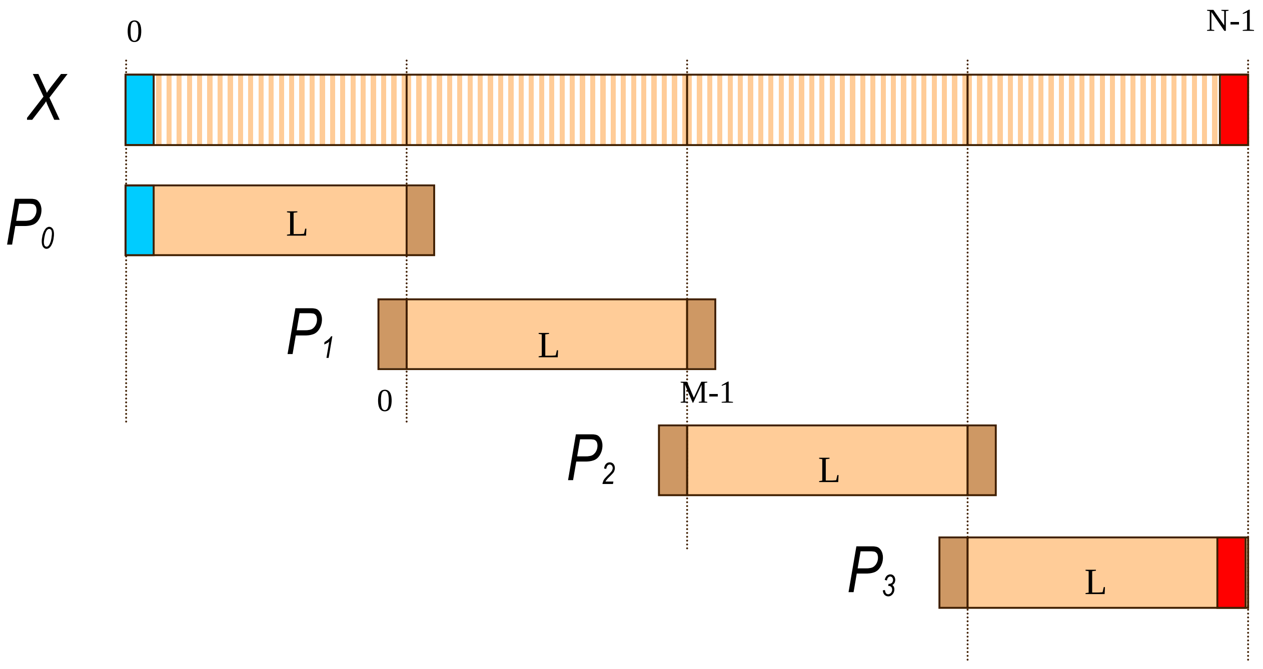
The temperature in an arbitrary point, *X[i]* at time *t+1* can be calculated as

$$X_{t+1}(i) = \frac{X_t(i-1) + X_t(i+1)}{2}$$

i.e. at each internal point the temperature is equal to the arithmetic mean of the temperatures at its two neighbouring points.

Write a parallel program that simulates the temperature distribution in the wire using this simplified model. Use an array of 200 floating point numbers to represent the temperature distribution. Use data decomposition to decompose the problem, as discussed in part 2 of the lectures (see the lecture slides from part 2, pages 46 – 60). You can assume that the number of points is evenly divisible by the number of processors (use 2, 4 or 8 processors).

Observe that the leftmost and rightmost processor has one point less than the other processors in the local array it uses to store its own part of the temperature values, since these only have on neighbour to communicate with. The following figure illustrates how the array *X* is distributed among the processes.



The computation has converged when no temperature value in any process has changed with more than a value ε (epsilon) since the last iteration. A suitable value of epsilon is 0,01.

The program should print out a table of the temperature distribution, the number of iterations it took to reach convergence and the value of epsilon that was used.

If you want to, and if you feel like you don't otherwise have enough to do, you may also present the temperature distribution graphically on the screen during the computation.