UNIVERSITY of York

MSc Information Technology Examinations, 2016-2017

Distributed Performance Computing (DIPC)

Open Examination

Issued at:
Wednesday 8th February 2017

Submission due:
Wednesday 15th March 2017

Feedback and marks due:
Wednesday 26th April 2017

All students should submit their answers through the electronic submission system: http://www.cs.york.ac.uk/student/assessment/submit/ by 12 noon, **Wednesday 15th March 2017**. An assessment (or part of an assessment) submitted after this deadline will be marked initially as if it had been handed in on time, but the Board of Examiners will normally apply a lateness penalty to the whole assessment.

The feedback and marks date is guided by departmental policy but, in exceptional cases, there may be a delay. In these cases, all students expecting feedback will be emailed by the module owner with a revised feedback date. The date that students can expect to see their feedback is published on the module descriptor: http://www.cs.york.ac.uk/modules/

Your attention is drawn to the Guidelines on Mutual Assistance and Collaboration in the Departmental Statement on Assessment: http://www.cs.york.ac.uk/student/assessment/policies/.

All queries on this assessment should be addressed to
Simon O'Keefe, simon.okeefe@york.ac.uk.

Your examination number must be written on the front of your submission.
You must not identify yourself in any other way.

Answers must not exceed 10 A4 pages in total; this limit includes any title page, diagrams, references, and so on. Excess pages will not be marked.

In addition to your written answers, you **MUST** submit the code for your solution. This is not part of the 12-page limit.

Answer *all* parts of this question. Your answers must address all of the points to gain full marks. You must evaluate thoroughly the performance of the system. You must provide justification for the choices you make. If your system performs poorly, you will still get credit for attempting to explain the performance.

This assessment requires you to write a program to solve a problem using the facilities of YARCC to distribute processing over a number of processors. You may use any appropriate software to create the program. A full justification for your choice of system and evaluation of its performance are more important than the ability of the program to produce "correct" results.

The problem is to solve the 'steady state' heat equation on a rectangular region, representing a plate heated along three edges to 100°C, and cooled to 0°C along the fourth edge. The physical region, and the boundary conditions, are suggested by the diagram in figure 1, in which W represents temperature:

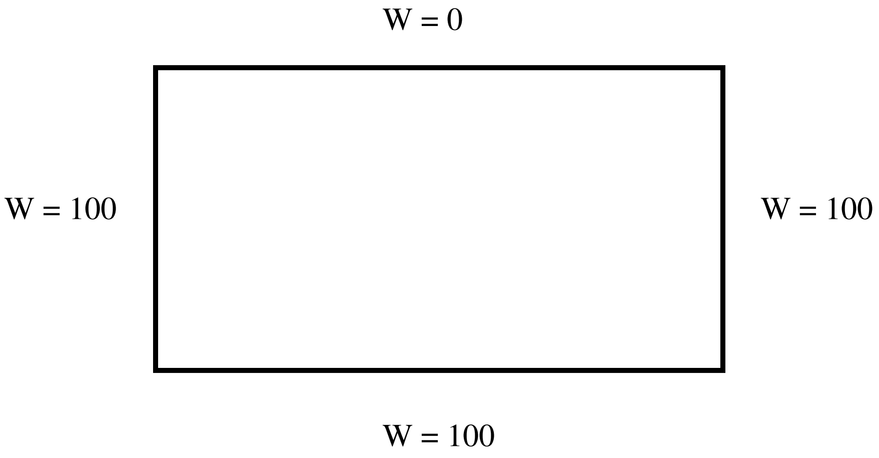

Figure 1: Physical region indicating temperature W maintained along each boundary.

The region is covered with a grid of M by N nodes, and an M by N array W is used to record the temperature. The correspondence between array indices and locations in the region is suggested by giving the indices of the four corners, as is shown in figure 2.

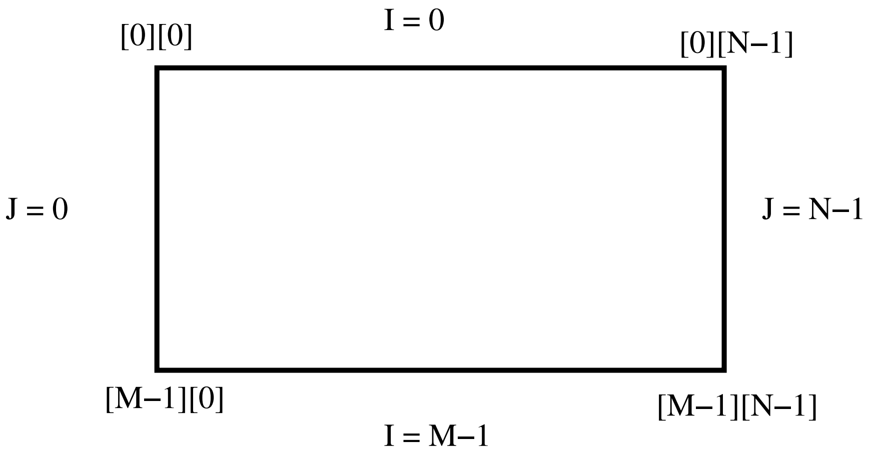

Figure 2: Correspondence between the array indices and locations in the region.

The steady state solution to the discrete heat equation satisfies the following condition at an interior grid point:

```
W[Central] = (1/4) * (W[North] + W[South] + W[East] + W[West])
```

where ‘Central’ is the index of the grid point, ‘North’ is the index of its immediate neighbor to the ‘north’, and so on. Given an approximate solution of the steady state heat equation, a ‘better’ solution is given by replacing each interior point by the average of its 4 neighbors - in other words, by using the condition as an assignment statement.

If this process is repeated often enough, the difference between successive estimates of the solution will go towards zero.

Your program should carry out such an iteration, using a tolerance specified by the user, and write the final estimate of the solution to a file that can be used to generate graphic representation of the solution.

1. [30 marks] The introduction above tells you how to calculate the steady state temperature at every point on the plate. Discuss at least two possible methods that you could use to distribute this processing over multiple processors [15 marks], your choice of method [3 marks], and the reasons for your choice [12 marks].

2. [70 marks] Write a program to implement your chosen method to solve the given heat diffusion problem in a distributed manner.

   i You must fully and explicitly document a detailed description of the implementation of the method you have chosen, including how the system was tested. [45 marks]

   ii For an implementation using a fixed problem size (size of plate) and fixed tolerance (specified by the user), calculate the temperature distribution calculated by (a) a single processor, and (b) four processors. For both solutions show the calculated temperature distribution as an image. Comparing the two solutions, calculate the standard deviation of differences in calculated temperature at each point in the two solutions. [25 marks]