PH-M31 Assignment Question 2

Tom Bourton

E-mail: 701329@swansea.ac.uk

1. Question 2 - Describe the structure of the parallel code, comparing it to the serial code.

We initialise the program from command line arguments, and define the size of communicating group and assign a label to each node called myid.

We then get node 0 only to read command line values for β, size, etc and print logs to a file.

Then broadcast values of size, beta, thermalisation and measurements to all nodes.

Then we do a test to ensure that the size can be divided into the number of procesors with no remainder.

We then initialise the random numbers, unlike the serial code, we need to take care to ensure that each node generates its own unique set of pseudo random numbers by giving each node its own unique seed, because of the nature of pseudo random numbers, if each node uses the same seed then they will generate the exact same set of random numbers.

We then get node 0 only to compute the boltzmann weights, and then broadcast the weights to all nodes, then each node initializes it's local lattice, each node creates (size/numprocess)+2 number of rows and then for each row we populate the row with empty spin values, this is so each node has sizenp2 rows of the global lattice, these 2 extra rows are because we need two rows on the boundaries to communicate between nodes.

Then we initalise each spin on the local lattices depending on hot or cold starts the same as the serial code, also set each spin on the top and bottom rows of the local lattice to contain no value(0) spin.

Then we thermalise the lattice thermalisation times, for each even node it sends the (top - 1) row of spins to the node above's (bottom + 1) row, and vice versa, there is

a parity if statement included to ensure that we do not get a block, where eg, node x and node y are both trying to send at the same time which will cause a block in the code.

Then each node sends the bottom row to the top row of the node above, and also sends the top row to the bottom row of the node below, then each node updates the spins on the local lattice apart form those spins in the outside rows by connecting the spins to the heatbath as in the serial code.then each node creates a new 1D arrays of doubles to store observable values.

Then we update the lattices in the same way as the thermalisation process, but this time (measurement) times, we also take measurements at each run through the for loop each node computes the magnetisation and energy values on each node's local lattice. Then after the measurements have been taken we do an mpi reduce to add all the energy and magnetisation values together from each node. Then all measurements are written to file.

The main differences between the serial and mpi code is that for the mpi code we must initialise the random number generators with a different seed for each core to ensure that we are not producing the same set of random numbers for each processor, as opposed to the serial code where we only require one seed. Also we need to worry about communicating lattice points between processors using the MPI_SEND functions.