PH-M31 Assignment Question 1

Tom Bourton

E-mail: 701329@swansea.ac.uk

1. Question 1 - Describe the serial code, and in particular specify which sections are suitable for parallelization and which sections are not

The program begins by reading input from the command line for Lattice size (size), β (beta), Number of thermalisation sweeps (thermalisation), Number of measurements (measurements), cold or hot start (istart), istart=1 $\Rightarrow$ hot start - spins are randomly assigned, istart=0 $\Rightarrow$ cold start - all spins start locked in +1 position.

A size*size grid is allocated and each lattice point is ran over and assigned spin values, if a cold start then all spins are locked in +1 position, if a hot start then spins are thermally disordered and so spins are randomly assigned +1 or -1 using the random number generator

The system is then thermalized by bringing the system into contact with a heatbath. The heatbath is generated by first specifying the periodic boundary conditions (staple function) and then summing over all nearest neighbours of the spin[i][j].

Then generate a random number and compare it to the Boltzmann weight, if the random number is less then the Boltzmann weight then set spin[i][j]= -1, else spin[i][j]= +1 this is done for each point on the lattice. The system is bought into contact with the heatbath (thermalisation) times, this step is done to thermalize the system

Then initialise observable variables, an energy and magnetisation value for each measurement and calculate the volume of the lattice (size)*(size).

Then bring the system into contact with the heatbath in the same way as for thermalisation, but this time it is brought into contact (measurements) times. For each measurement compute the magnetisation by summing over all spins, compute the energy by summing over links, each lattice point has 4 links with nearest neighbours, however we do not want to overcount links so we only consider 4/2 links per lattice point, in this case we consider the link to the right and the link above.

Then perform jackknife analysis on the magnetization and energy values. Begin by

taking the average of the observable by summing the observables for each measurement then dividing by the number of measurements. Then divide the observables into bins of length (slice), then form the jackknife bins to remove bias. Then compute the jackknife error by taking the (observable average - jackknife bin)^2 for each bin, then normalise with the (n-1)/n factor then take the square root.

Then print magnetisation and energy results to the console and write to a file.

The jackknifing algorithm may be suitable for parallelization because we can divide up the bins, then allow each node to independently work on the bins, e.g. if we have 30 bins and 10 nodes, each node can execute the jacknifing algorithm on 10 bins each. However it may not be beneficial to parallelize as in general we are working on only a small number of bins.

We can also parallelize the lattice initialisation, however we must ensure that each node generates a separate set of pseudo random numbers. We can parallelize the update algorithm to work in parallel, as well as the thermalisation and measurement sweeps by dividing up the global lattice into a set of local sub-lattices and allowing each node to work on the sub-lattices, using MPI to communicate the sub-lattice boundaries to other nodes.

The calculation of the Boltzmann weights does not need to be done in parallel as we are computing only a small set of values.