

Ising Model Monte Carlo

0.1 Level 1

Use the blocks provided to set the microscopic coordinates of all the spins in the system equal to 1

0.2 Level 2

Use the blocks provided to set the microscopic coordinates of all the odd numbered spins in the system to 1 and all the even numbered spins in the system to -1.

0.3 Level 3

Use the blocks provided to generate a random configuration for the microscopic coordinates of all the spins.

0.4 Level 4

Use the blocks provided to generate a random configuration for the microscopic coordinates of all the spins. Then select one of the spin coordinates at random and flip the value of this spin.

0.5 Level 5

Use the blocks provided to generate a random configuration for the microscopic coordinates of all the spins. Then make your code repeat the process of flipping one randomly chosen spin a total of 10 times.

0.6 Level 6

At the end of the previous level we implemented a Monte Carlo integrator that accepted all moves and that therefore operated at infinite temperature. To operate at finite temperature we are going to have to calculate the energy. Use the blocks provided to generate a random configuration of spins and then calculate the energy of this microstate using the Ising model Hamiltonian. Once the energy has been calculated you should report it to the screen using the print energy block.

0.7 Level 7

Lets now generate a trajectory of 11 frames using what we learnt while programming the code for the last but one level. Generate a first configuration at random and then calculate the energy using the Ising model Hamiltonian for each of the microstates you generate by flipping, randomly-chosen individual spins and report these to screen. You will need to include the code to calculate the energy of the system twice in this program so you will probably find it useful to write a function that calculates the energy and returns it.

0.8 Level 8

The arrangement of blocks you should have just arrived at provides almost everything we need for our Monte Carlo integrator. Can you finish off the code by implementing the Metropolis accept/reject criteria as well. Make sure you only report the energies of the accepted configurations to screen.