Two-Dimensional Ising Model Simulation

PHY2027 – Scientific Programming in C – Final Project Report

Miguel de Sousa

November 22, 2025

1 Theory

[1] The Ising model, introduced by Ernst Ising in 1925, is a mathematical model consisting of a lattice of 'spin' variables σ_α which can take discrete values of ± 1. These spins represent magnetic dipole moments of atomic spins that can be in one of two states: up (+1) or down (-1). The model is used to study phase transitions and critical phenomena in statistical mechanics. [1]Any two of these spins have a mutual interaction Energy

$$-E_{\alpha\beta}\sigma_\alpha\sigma_\beta \tag{1}$$

[1]Additionally, an external magnetic field B can interact with each spin, contributing an energy term of

$$-B\sigma_\alpha \tag{2}$$

The total energy of a configuration of spins on a lattice can be expressed as

$$E = -\sum_{\langle\alpha,\beta\rangle} E_{\alpha\beta}\sigma_\alpha\sigma_\beta - B\sum_\alpha \sigma_\alpha \tag{3}$$

where the first sum is over all pairs of neighbouring spins. The Ising model exhibits a phase transition at a critical temperature T_c, below which the spins tend to align, resulting in spontaneous magnetisation, and above which the spins are disordered.

2 Implementation

2.1 Data Structure and Initialisation (Part a)

I began by defining a structure type to represent the 2D Ising lattice, which included all necessary parameters as structure members for the model such as its N-dimeniionsionality, values for temperature T, external magnetic field B, total energy and magnetisation of the lattice, and a pointer to an array of spins.

```
1  typedef char Spin;
2  static const Spin SPIN_UP   =  1;
3  static const Spin SPIN_DOWN = -1;
4
5  typedef struct{
6      int N;
7      Spin *spin;
8      double T;
9      double B;
10     unsigned long long step;
11     double energy;
12     double magnetisation;
13 } IsingLattice;
```

Lines 1-3: Assigning a char type to Spin, with constants for SPIN_UP and SPIN_DOWN defined with this new type. The char data type forces each spin to occupy only 1 byte of memory, which is efficient for large n by n lattices. The Spin type is used throughout the code, so defining it at the start improves code readability.

Lines 5-14: Creating the IsingLattice structure type with members for lattice size N, pointer to spin array, temperature T, external magnetic field B, current step number, total energy, and total magnetisation. Structures are useful for grouping related data together, so its perfect for representing the lattice and its properties. I used double for T, B, energy, and magnetisation to allow for fractional values and higher precision in calculations. The step member is an unsigned long long to allow a large number of Monte Carlo steps. Proving its reuseability, the spin pointer defined in the sructure types is allocatd to my new Spin data type.

After defining the structure type, I implemented the initialisation function to set up the lattice with random n by n spins.

```
1  IsingLattice *create_lattice(int N, double T, double B){
2
3      IsingLattice *lattice = malloc(sizeof(IsingLattice));
4
5      if (lattice == NULL){
6          fprintf(stderr, "Memory allocation failed\n");
7          exit(EXIT_FAILURE);
8      }
9
10     lattice->N = N;
11     lattice->T = T;
12     lattice->B = B;
13     lattice->step = 0;
14     lattice->spin = malloc(N * N * sizeof(char));
15
16     if (lattice->spin == NULL){
17         fprintf(stderr, "Memory allocation for spin failed\n");
18         free(lattice);
19         exit(EXIT_FAILURE);
20     }
21
22     for (int i = 0; i < N * N; i++){
23         lattice->spin[i] = (rand() % 2) ? SPIN_UP : SPIN_DOWN;
24     }
25
26     return lattice;
27 }
```

Lines 1-8: The create_function takes lattice parameters as arguments and it allocates enough memory for the new lattice instance using malloc and the size of the IsingLattice structure type. It checks if the allocation was successful, printing an error message and exiting if not. The structure type pointer is initalised within the fucntion as *lattice.

Lines 10-13: Assign the input parameters N, T, and B to the corresponding structure members in the IsingLattice using -¿ structure pointer notation.

Lines 14-24: Allocate memory for the spin array within the lattice structure using malloc. The size is n by n multiplied by the size of a char (1 byte) since each spin is represented as a char. It checks for successful allocation, freeing the previously allocated lattice memory location if the spin allocation fails. I then assigned each lattice spin array value in the lattice to a random value of SPIN_UP or SPIN_DOWN using a count controlled loop with boundaries at 0 and n by n. The rand() function acted as the randomiser, with modulus 2 giving eitehr values of 0 or 1, which were mapped to either SPIN_UP or SPIN_DOWN.

Lines 26: Finally, the function returns a pointer to the newly created and initialised lattice instance.

References

[1] B. M. McCoy and T. T. Wu, *The Two-Dimensional Ising Model*, Harvard University Press, Reprint 2014.