

Computational Physics (physics760)

Exercise 2

Ajay S. Sakthivasan, Dongjin Suh

November 4, 2022

1 Simulating the 2-D Ising Model

In this homework, we implemented the **Metropolis-Hastings** algorithm to simulate the 2D Ising Model. The calculation of total energy of a particular configuration is much more efficient in our implementation, due to the possibility of manipulating *numpy* arrays. Figure 1 shows the evaluation times for arbitrary configurations with $N = 1, 2, \dots, 20$, timed using `timeit` on `Python`. We see that the evaluation time grows only very weakly with N . This is due to the efficiency of *numpy* manipulations.

Similarly, calculations of energy before and after a spin flip shows a similar

```
46.4 µs ± 1.51 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
53.5 µs ± 3.62 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
57.2 µs ± 7.82 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
55.8 µs ± 2.63 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
52.8 µs ± 1.74 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
58.4 µs ± 5.69 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
56.6 µs ± 3.38 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
61.7 µs ± 7.97 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
75.7 µs ± 32.7 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
82.6 µs ± 37.8 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
64.5 µs ± 19.6 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
59 µs ± 4.06 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
65.1 µs ± 1.86 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
58.4 µs ± 5.29 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
55.8 µs ± 2.62 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
61.7 µs ± 6.6 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
60.9 µs ± 4.01 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
66.7 µs ± 3.67 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
71.7 µs ± 10.5 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
63.7 µs ± 2.16 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
```

Figure 1: Evaluation time for arbitrary configurations with $N = 1, 2, \dots, 20$

trend, as can be seen from the code. However, this step is implemented inside a *for* – *loop*, which makes it grow roughly quadratically with the number of lattice points. This step is at the heart of the **Metropolis-Hastings** algorithm.

The critical coupling, J_c , gives the value of the coupling beyond which spontaneous magnetisation takes place for a given non-zero external magnetic field. This is an example of a phase transition, whose universality across different physical systems has been well studied. This can be seen from figure 2. The value of J_c is about 0.44. Figure 3 shows the analytical result for energy vs. coupling factor.

Figure 4 shows magnetisation vs. external magnetic field for a fixed value of J .

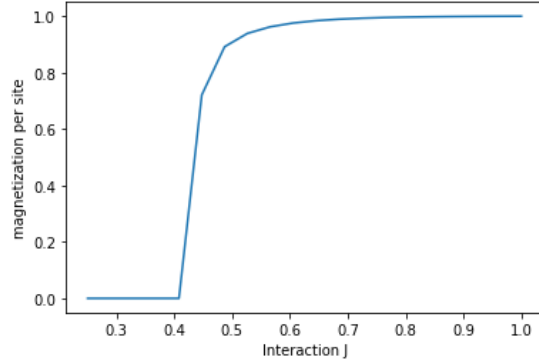


Figure 2: Analytical result for Magnetisation vs. Coupling factor

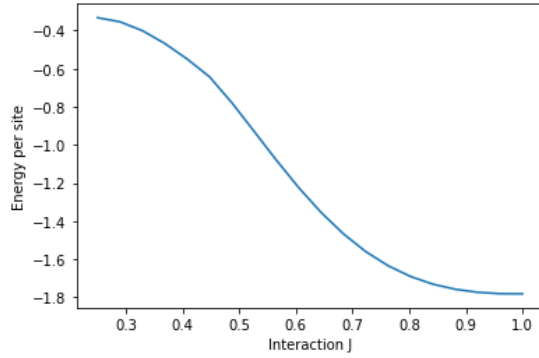


Figure 3: Analytical result for Energy vs. Coupling factor

Figure 5 shows the numerical result for magnetisation vs. coupling factor and figure 6 shows the numerical result for energy vs. coupling factor. As can be seen, unfortunately we were unable to simulate the expected phase transitions in our numerical simulations. We are actively trying to understand what went wrong with our simulations, and hopefully we are able to fix it in the near future. Our best guess at this point is that we are overlooking some trivial mistake in the implementation of the algorithm. To the question what would happen if we plot $\langle m \rangle$, instead of $\langle |m| \rangle$, we would no longer observe any phase transition in

that case.

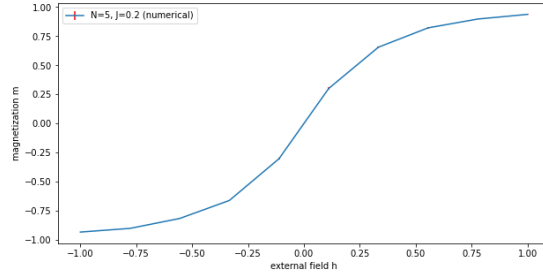


Figure 4: Magnetisation vs. External field for a given coupling factor

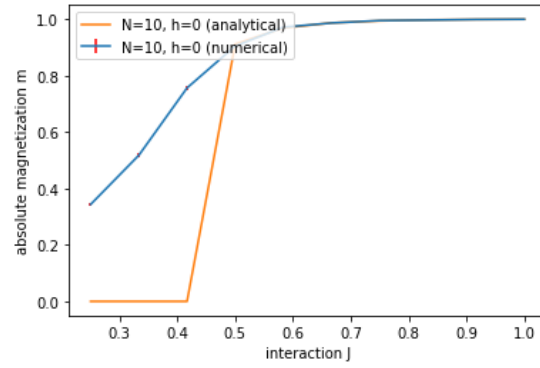


Figure 5: Magnetisation vs. Coupling factor (Simulation)

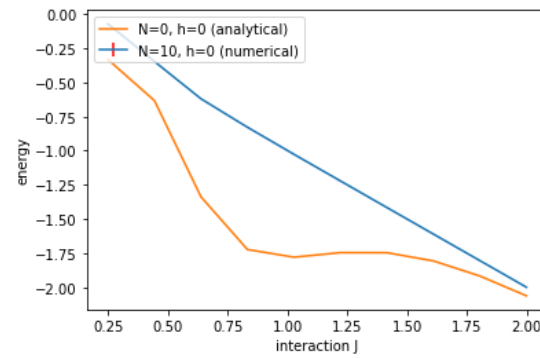


Figure 6: Energy vs. Coupling factor (Simulation)