Exercise 03: Applying HMC to the long-range Ising model

Computational Physics WS20/21

Marc Vöhringer Carrera und Helen Meyer

November 25, 2020

1 Introduction

After simulating the Ising model in one and two dimensions in the last exercises, we now want to apply the Hamiltonian Monte Carlo (HMC) algorithm to the long range Ising model in one dimension. The Hamiltonian for this system is

$$H(s,h) = -\frac{1}{2}\hat{J}\sum_{i,j} s_i s_j - h\sum_i s_i, \tag{1}$$

where $\hat{J} = J/N$ and the coupling term is not restricted, but summed over *all* spins, not just next-neighbours.

2 Hubbard-Stratonovich Transformation

In order to apply the HMC algorithm, which works only on continuous spaces, to the Ising model, which is discrete, we employ the Hubbard-Stratonovich transformation. We want to linearize the argument of the exponential in the partition function

$$Z = \sum_{s_i=\pm 1} e^{\beta J\left(\frac{1}{2N}\sum_{i,j} s_i S_j + \frac{h}{J}\sum_i s_i\right)} \tag{2}$$

by applying

$$e^{-\frac{1}{2}|U|s^2} = \int_{-\infty}^{\infty} \frac{\mathrm{d}\phi}{\sqrt{2\pi|U|}} e^{-\frac{\phi^2}{2|U|}\pm i\phi s} \tag{3}$$

$$e^{\frac{1}{2}|U|s^2} = \int_{-\infty}^{\infty} \frac{\mathrm{d}\phi}{\sqrt{2\pi|U|}} e^{-\frac{\phi^2}{2|U|}\pm \phi s}, \tag{4}$$

with $s = \sum_i s_i$ the total spin of the system. After some calculations we get

$$Z = \begin{cases} \int_{-\infty}^{\infty} \frac{\mathrm{d}\phi}{\sqrt{2\pi\beta\hat{J}}} e^{-\frac{\phi^2}{2\beta\hat{J}} + N\log(2\cosh(\beta h \pm \phi))} & J > 0 \\ \int_{-\infty}^{\infty} \frac{\mathrm{d}\phi}{\sqrt{2\pi\beta\left|\hat{J}\right|}} e^{-\frac{\phi^2}{2\beta\left|\hat{J}\right|} + N\log(2\cosh(\beta h \pm \phi))} & J < 0 \end{cases} . \tag{5}$$

From now on, we assume $J > 0$.

1: *The expectation value of some operator O is given by*

$$\langle O \rangle = \frac{1}{Z} \int \frac{\mathrm{d}\phi}{\sqrt{2\pi\beta\left|\hat{J}\right|}} O[\phi] e^{-S[\phi]} \tag{6}$$

Define

$$S[\phi] = \frac{\phi^2}{2\beta\left|\hat{J}\right|} - N\log(2\cosh(\beta h \pm \phi)). \tag{7}$$

The average magnetization per site and energy per site are given by

$$\langle m \rangle = \frac{1}{N\beta} \frac{\partial}{\partial h} \log(Z) \tag{8}$$

$$\langle \epsilon \rangle = -\frac{1}{N} \frac{\partial}{\partial \beta} \log(Z) \tag{9}$$

We want to find the functions $m[\phi]$ and $\epsilon(\phi)$, so we calculate equations 8 explicitly and find:

$$\langle m \rangle = \frac{1}{Z} \int \frac{\mathrm{d}\phi}{\sqrt{2\pi\beta\hat{J}}} \tanh(\beta h \pm \phi) e^{-S[\phi]} \tag{10}$$

$$\langle \epsilon \rangle = \frac{1}{Z} \int \frac{\mathrm{d}\phi}{\sqrt{2\pi\beta\hat{J}}} \left(\frac{1}{2\beta N} - \frac{\phi^2}{2J\beta^2} - h\tanh(\beta h \pm \phi) \right) e^{-S[\phi]} \tag{11}$$

By comparing to eq. (6), we find

$$m[\phi] = \tanh(\beta h \pm \phi) \tag{12}$$

$$\epsilon[\phi] = \frac{1}{2\beta N} - \frac{\phi^2}{2J\beta^2} - h\tanh(\beta h \pm \phi) \tag{13}$$

We define the artificial Hamiltonian as a function of ϕ and the conjugate momentum p as

$$\mathcal{H}(p, \phi) = \frac{p^2}{2} + \frac{\phi^2}{2\beta\hat{J}} - N\log(2\cosh(\beta h + \phi)). \tag{14}$$

2: The equations of motion for this Hamiltonian are

$$\dot{\phi} = \frac{\partial}{\partial p}\mathcal{H} = p \tag{15}$$

$$\dot{p} = -\frac{\partial}{\partial \phi}\mathcal{H} = -\frac{\phi}{\beta\hat{J}} + N\tanh(\beta h + \phi) \tag{16}$$

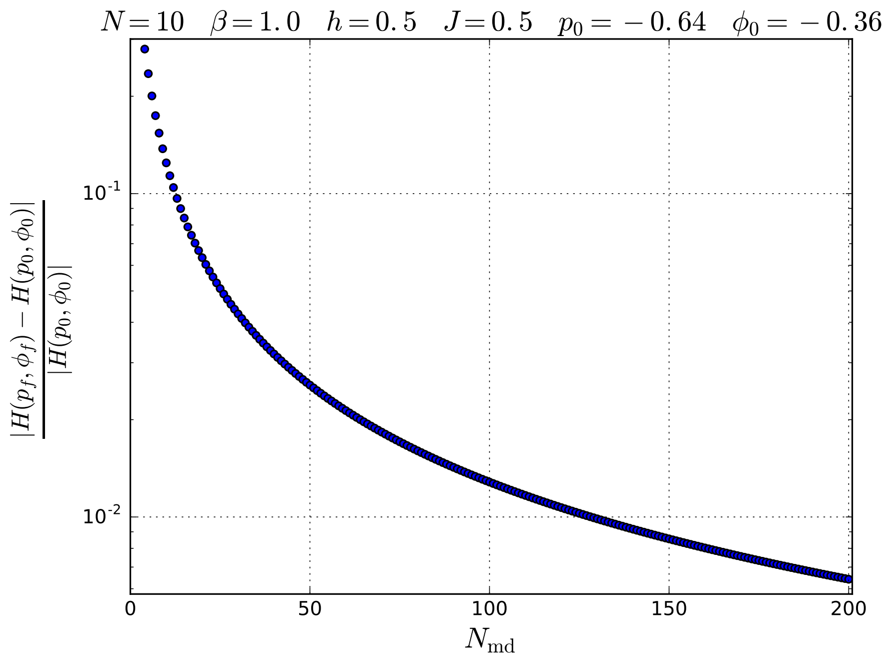

Figure 3.0.1: Test of the leapfrog algorithm.

3 Leapfrog Algorithm

These equations of motion we want to integrate using the leapfrog algorithm implemented in the file `leapfrog.py`. The goal is to change ϕ and p in a way that keeps the Hamiltonian constant. The convergence of our leapfrog algorithm for $J =$ is shown in Fig. 3.0.1.

4 HCM Algorithm and Results

Now, in order to create a sample of ϕ's, we implement the HMC algorithm (in `hmc.py`), starting with $\phi = 0$. First we generate a $p \in \mathcal{N}_{(0,1)}$ and apply the leapfrog algorithm to p and ϕ. The new pair $(p\prime, \phi\prime)$ is accepted with a probability $P = \min\{1, e^{\mathcal{H}(p,\phi)-\mathcal{H}(p\prime,\phi\prime)}\}$, and the new value of ϕ is stored (when rejected, we keep the old value of ϕ). This we repeat N_{cfg} times after a thermalization period of N_{therm} steps. Having generated a sample of ϕ's, the expectation value of an operator O can be calculated:

$$\langle O \rangle = \frac{1}{N_{\mathrm{cfg}}} \sum_{\{\phi\}} O[\phi]. \tag{17}$$

We want to calculate the average magnetization per site $\langle m \rangle$ and the average energy per site $\langle \epsilon \rangle$ as a function of the coupling $J \in [0.2, 2]$, with $h = \beta h = 0.5$. The results an be seen in Figs. 4.0.1 – 4.0.4.

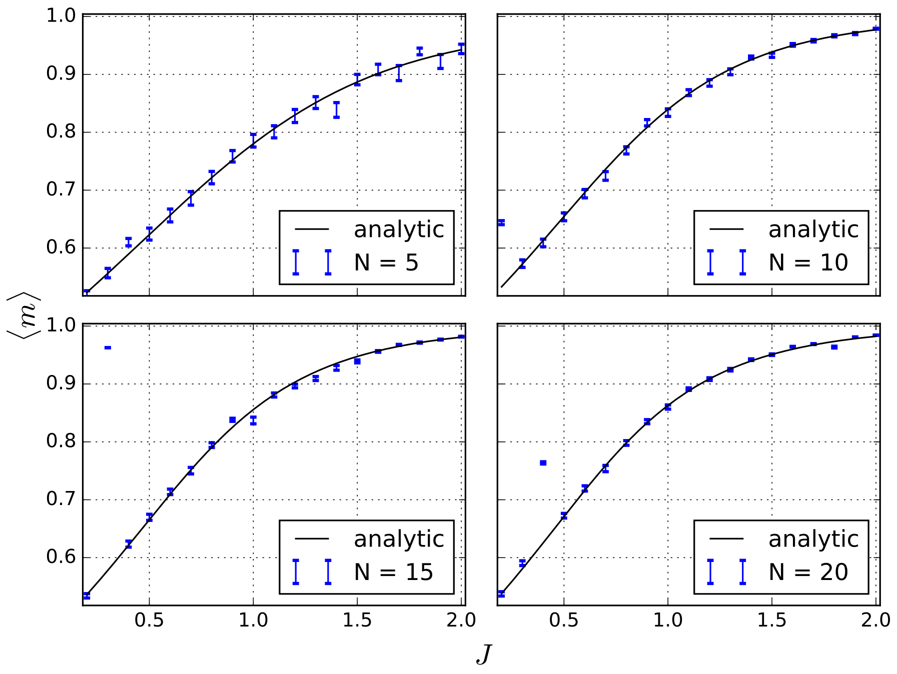

Figure 4.0.1: $\langle m \rangle$ vs. J for $N = 5, 10, 15, 20$. Numeric results shown as errorbars and analytic solution shown as solid line.

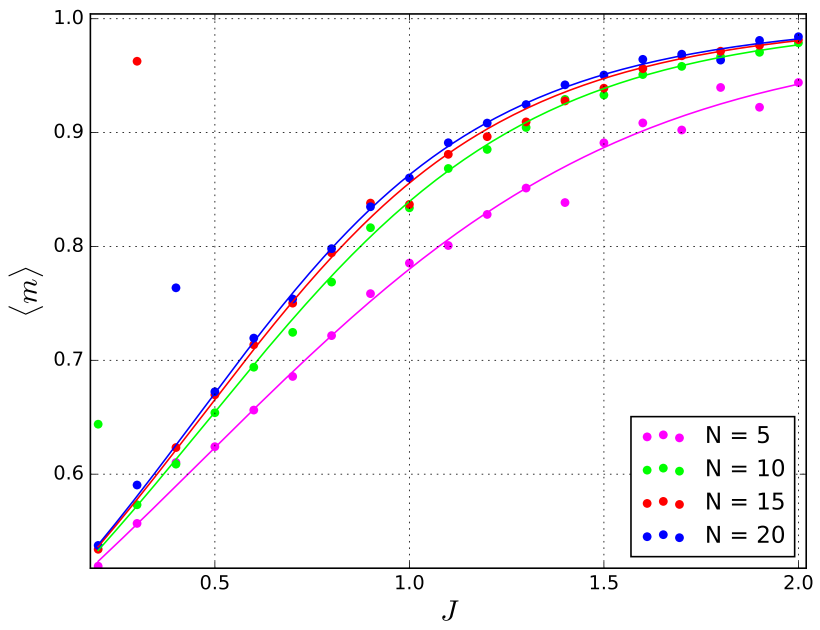

Figure 4.0.2: Overlaid version of Fig. 4.0.1. Numeric results shown as errorbars and analytic solution shown as solid line.

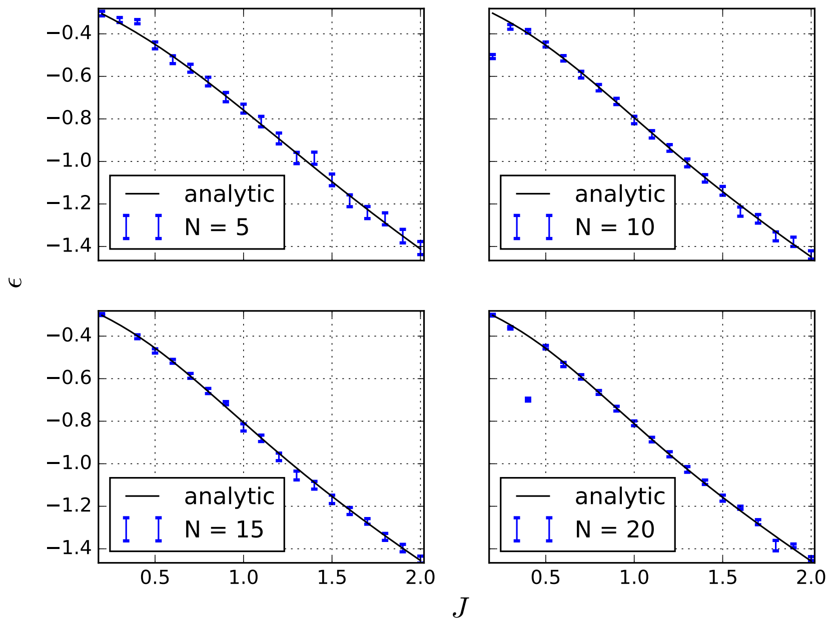

Figure 4.0.3: ϵ vs. J for $N = 5, 10, 15, 20$. Numeric results shown as errorbars and analytic solution shown as solid line.

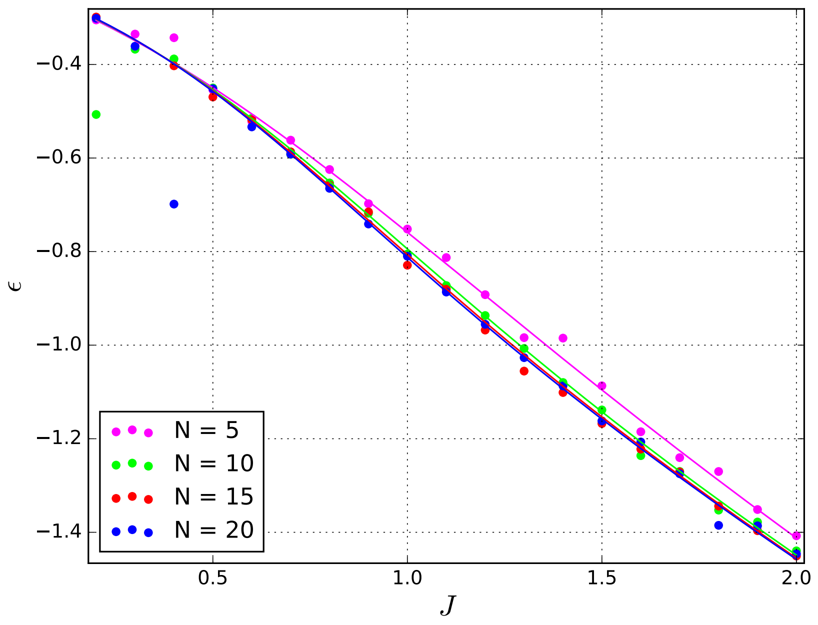

Figure 4.0.4: Overlaid version of Fig. 4.0.3. Numeric results shown as errorbars and analytic solution shown as solid line.

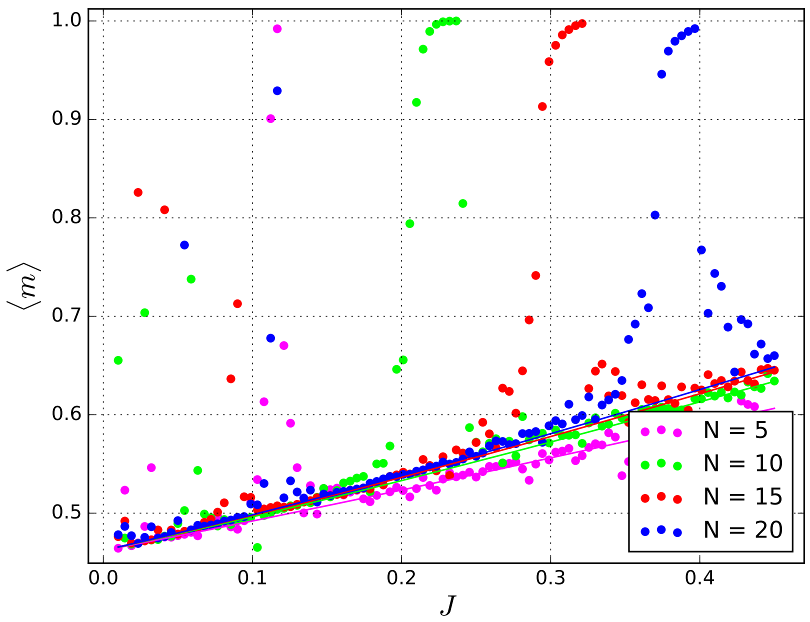

Figure 4.0.5: Irregularity in the algorithm

The numerical results obtained with our algorithm follow the analytical solution quite nicely except for some points at low values of J. It can also be seen that the general deviation from the analytic results decrases with increasing N. As we found a weird behaviour of our results for certain values of N and J, we first suspected that *pathological runs* were happening, as discussed in the tutorial. To prevent these, before the first run for a certain N, we did an extra of $50 \cdot N_{\text{therm}}$ thermalization runs, and used the resulting ϕ as starting point for the simulation. We then used the ϕ of the final run as starting point for the next value of J, after jiggling it a little bit by adding a random value $\in \mathcal{N}_{(-0.25,0.25)}$. Between each value of J we still perform Nterm thermalization runs. As the deviations from the analytical solution keep happening, we suspect that the deviations have some deeper origin lying in the algorithm. Unfortunately we did not have time to dig deeper by looking at the Monte Carlo histories of $\langle m \rangle$. In Fig. 4.0.5 we plotted the irregularities found in the algorithm with a closer look. It seems that at certain values of J (depending proportionally? on N) the obtained results deviate from the analytical solution during a short window.