KFU Graz
Dénes Sexty (denes.sexty@uni-graz.at)
CP1: Monte Carlo Methods WS 2023/24

13. Exercise: Continuum extrapolation and Improved Langevin update

First, use the simulation in the exercise before (you can download `Langevin.py` from the moodle page) to measure $\langle x^2 \rangle$ for different $\Delta\tau$ values, and carry out the continuum extrapolation.
Second, use the following update for the Maxwell-Boltzmann simulation with the Langevin equation:

$$\begin{aligned} x' &= x(\tau) + \partial_x S(x)\Delta\tau + \eta_\tau \sqrt{2\Delta\tau} \\ x(\tau + \Delta\tau) &= x(\tau) + \frac{1}{2}\left(\partial_x S(x) + \partial_x S(x')\right)\Delta\tau + \eta_\tau \sqrt{2\Delta\tau} \end{aligned} \tag{1}$$

Note that the noise variable is the same in both equations. Now measure $\langle x^2 \rangle$ for several different $\Delta\tau$ values. It should depend on $\Delta\tau$ as

$$\langle x^2 \rangle(\Delta\tau) = \langle x^2 \rangle(\Delta\tau = 0) + C\Delta\tau^n, \tag{2}$$

with some constant C. What is n (the order of the update)?

14. Exercise: Ising model

Here we aim to build an MC simulation of the Ising model. The spins will be stored in a 2D array of integers.
Fill the lattice with random spins.
First implement a subroutine calculating the action and the total magnetization:

$$H = -J \sum_{neigh} s_i s_j - h \sum s_i, \quad M = \frac{1}{N^2} \sum s_i \tag{3}$$

Use periodic boundary conditions. Now implement a subroutine `deltaE(x,y)` which calculates the change of the energy if the spin at x, y is flipped. Make sure it is consistent with the algorithm above calculating the total energy.
Next is a routine for a Monte-Carlo hit of the spin at x, y. Calculate the acceptance probability and accordingly update the spin. Next you need a subroutine for a sweep (random or one-by-one).
Take $J = 1, h = 0$ and try various β values.
You can at look some of the the following questions:
Measure the magnetization as the function of the number of sweeps. How long thermalization is needed? Can you observe the metastability of the magnetization at low temperatures? How does the frequency of tunnelings depend on the lattice size? What is the acceptance rate as the function of temperature? Calculate the susceptibility of the magnetization. Try to find the critical β value and calculate histograms of the magnetization and the energy.

15. Exercise: Bonus exercise: Autocorrelation

Suppose we have a set of N values $\Phi(x)$ with $x \in \{0, 1, \dots N-1\}$. We define the Discrete Fourier Transformation (DFT) and its inverse as:

$$\Phi_k = \sum_x \Phi_x e^{-ikx2\pi/N}, \quad \Phi_x = \frac{1}{N}\sum_k \Phi_k e^{ikx2\pi/N} \tag{4}$$

where we also have $k \in \{0, 1, \dots, N-1\}$. There is a fast algorithm (called Fast Fourier Transform: FFT) which calculates Φ_k from Φ_x (or Φ_x from Φ_k) in $O(N\log(N))$ steps.
We can use this to calculate autocorrelations (using real fields satisfying $\Phi_x = \Phi_x^*$)

$$\begin{aligned} \sum_x \Phi_{x+\Delta}\Phi_x &= \sum_x \Phi_{x+\Delta}\Phi_x^* = \frac{1}{N^2}\sum_{x,k,k'} \Phi_k e^{ik(x+\Delta)2\pi/N}\Phi_{k'}^* e^{-ik'x2\pi/N} = \\ &= \frac{1}{N^2}\sum_{k,k'} N\delta_{k,k'}\Phi_k\Phi_{k'}^* e^{ik\Delta 2\pi/N} = \frac{1}{N}\sum_k \Phi_k\Phi_k^* e^{ik\Delta 2\pi/N} \end{aligned} \tag{5}$$

(So we calculate the inverse Fourier Transformation of the absolute value squared of the fourier transformation of Φ.) This assumes the data is periodic: $\Phi_{x+N} = \Phi_x$, so we get extra terms because of this: e.g. $\sum_x \phi_{x+2}\phi_x$ includes $\phi_1\phi_{N-1}$, which should not contribute to the autocorrelations. If we want to avoid these extra terms, we can expand our data set with N zeroes, than no unwanted terms occurs if we take the autocorrelation above for $\Delta < N$. (The new dataset is thus $\phi'_i,\ i \in \{0, 1, \dots, 2N-1\}$, $\quad \phi'_i = \phi_i$ for $i < N$ and $\phi'_i = 0$ for $i \geq N$.) In this case we have to take care of the correct normalization by hand: $\sum_x \Phi_{x+\Delta}\Phi_x$ has only $N - \Delta$ terms. Also, we have to substract the average of the data set beforehand in case it is nonzero, to calculate the autocorrelations. Using these steps we can calculate the autocorrelations in $O(2N\log(2N))$ steps.
Implement the calculation of autocorrelations using the ideas above and the `scipy.fft` and `scipy.ifft` functions, which calculate the Fourier transformation (and its inverse) with the same conventions of the formulas above. Verify that this gives the same result as the direct but slow calculation (which takes $O(N^2)$ steps).