Computer simulations in statistical physics

Sheet 1 - **hand in by Friday, 5.5.2023 12:00**

R. Cortes-Huerto, M. Girard, L. Stelzl, P. Virnau, M. Schmitt SoSe 2023

Notes for submissions

- Please pack your solution into a **single zip-archive**.
- If there is something non-obvious necessary for compiling or running the code, please add a README file.

Exercise 1: Random number generators *(5 points)*

You can choose which programming language you are going to use. However, Python functions which help in solving the tasks will be given as hints in the following. The uniform random number generator of Python is the uniform-function of the random package.

(a) Generate 100000 uniformly distributed random numbers between 0 and 1 and save the output.

(b) Write a program which calculates the average value, the variance and the error of the mean of given data and use it to evaluate the random numbers generated in (a). Do NOT use already implemented functions.

$$\text{VAR} = \left(< X^2 > - < X >^2 \right), \quad \text{ERR} = \left(\sqrt{\frac{VAR}{n}} \right) \tag{1}$$

(c) **Uniformity Test:** Create a histogram of the data with 100 bins and plot it.
Hint: In Python, you can generate a histogram using numpy.histogram(). Otherwise, you can easily plot a simple data file using xmgrace or gnuplot.

(d) **Screen Pixel Test:** Generate 2×100000 random numbers. Visualize the result by interpreting each pair of random numbers as x and y coordinates.

(e) Program your own linear congruential random number generator:

$$X_{n+1} = (a \cdot X_n + c) \mod m \tag{2}$$

Hint 1: To generate numbers between 0 and 1, the result needs to be divided by m.
Hint 2: Numerical Recipes in C advocate:

$$a = 1664525, \quad c = 1013904223, \quad m = 2^{32}$$

Exercise 2: Your first Monte Carlo Simulation *(5 points)*

Write a Monte Carlo program which calculates π.

Exercise 3: Metropolis Criterion *(4 points)*

(a) Verify that the Metropolis criterion enforces detailed balance. Distinguish between $\Delta E < 0$ and $\Delta E > 0$.

(b) An alternative to the Metropolis criterion is the heat bath algorithm. For the canonical ensemble the probability of jumping from state S_i to state S_j is:

$$\omega_{ij} = \frac{e^{-\beta \Delta E}}{e^{-\beta \Delta E} + 1}, \quad \Delta E = \left(E_j - E_i\right). \tag{3}$$

Show that this criterion fulfills detailed balance for the canonical ensemble:

$$\frac{\omega_{ij}}{\omega_{ji}} = \frac{P_j}{P_i} = e^{-\beta \Delta E} \tag{4}$$

(c) Write down the Metropolis criterion and the heat bath criterion for hard disks.
Hint: Hard disks are 2D objects with the following pair potential (for diameter σ):

$$V(\vec{r_1}, \vec{r_2}) = \begin{cases} 0 & |\vec{r_1} - \vec{r_2}| \geq \sigma \\ \infty & |\vec{r_1} - \vec{r_2}| < \sigma, \end{cases} \tag{5}$$

Exercise 4: Ising model *(6 points)*

(a) Take a look at https://mattbierbaum.github.io/ising.js/. What happens at low and high temperatures, respectively? At which temperature does the phase transition occur approximately? For which temperatures would the use of the single-spin-flip (Metropolis) algorithm and the Wolff-algorithm, respectively, be appropriate?

(b) Write a pseudo code for the single-spin-flip algorithm for the Ising model. Which different values for the energy difference ΔE occur in 1D and 2D, respectively?