*Note:* You will be given time to solve the exercises during the tutorial.

## Exercise 1.1: Estimate $\pi$ with Monte Carlo

a) Estimate $\pi$ by "shooting" (i.e., drawing random numbers) $N$ times uniformly on a square and counting the number of points hitting a disc target: the ratio of hits to $N$ should correspond to the ratio of the areas of the target to the area you shoot on.

b) Estimate the variance of the error for given $N$ by repeating this a few times.

c) Plot the variance of the error versus $N$ on a log-log scale. What is the scaling of the error?

d) Generalize your code to estimate the volume of a $d$-dimensional sphere. Does it run much slower for large $d$?

## Exercise 1.2: Importance sampling with Monte Carlo

The goal of this exercise is to find a Monte Carlo estimate of $I = \int_{a=0}^{\infty} \mathrm{d}x \frac{\mathrm{e}^{-x}}{1+(x-1)^2}$. (It is fine to cut off the upper limit at some value e.g. b=10.)

a) Write a function that calculates $I$ by using that $I = \overline{f} \times (b-a)$ with $f(x) = \frac{\mathrm{e}^{-x}}{1+(x-1)^2}$. (You can find an estimate of $\overline{f}$ by uniformly generating some $x_i \in [a,b)$ and averaging over $f(x_i)$.) Get an idea of the error of this estimate.

b) Now we will use importance sampling. The p.d.f. $g(x) = \alpha \mathrm{e}^{-\alpha x}$ turns out to be a good choice. In order to maximally improve our estimate of $I$, we will need the optimal value of $\alpha$. Write a function that determines the optimal $\alpha$, or use the function in the script `integral.py`.

c) Use importance sampling to estimate $I$ over the same number of samples as in (a). How much improved your result?

## Exercise 1.3: Metropolis algorithm for the 2D Ising model

Download the script `metropolis.py` from the homepage, which implements the Metropolis algorithm for the classical 2D Ising model $H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$ with $J \equiv 1$. The 2D Ising model has a critical point at $T_c = 2J/\ln(1+\sqrt{2}) \approx 2.269$.

a) What is the script plotting?

b) What are "typical" configurations at temperatures $T \gg T_c$, $T \approx T_c$ and $T \ll T_c$?

c) Plot the energy $E$ and specific heat $C_V$ versus temperature $T$ for different system sizes $L$.

d) Adjust the script to measure the magnetization $M = \frac{1}{L^2} \sum_{i,j} \sigma_{i,j}$. Plot how $M$ changes with simulation time (=the number of updates performed) for $T > T_c$, $T \approx T_c$ and $T < T_c$. Which time scales can you recognize? In which cases do you still get the correct expectation value $\langle M \rangle = 0$? Plot $\langle |M| \rangle$ (i.e. taking the absolute value of $M$ before averaging) versus $T$ to see the transition.

e) Include a magnetic field $h$ coupling to the spins with a term $H' = -h \sum_i \sigma_i$. Plot $\langle M \rangle$ versus $T$.

Bonus Some further ideas for playing around:

- Instead of restarting from a random state for each new $\beta$, re-use the last state of the previous simulation. You should still perform sweeps without measurements for the thermalization. Is it better to start with large $\beta$ or small $\beta$?
- Change the lattice.
- Optimize the code.
- ...