TUTORIAL: MONTE CARLO METHODS

I. Simulation of two repulsive particles in a harmonic potential

We consider two particles 1 and 2 of equal mass m in a one-dimensional harmonic potential of stiffness κ at temperature T. The two particles also interact repulsively with a typical interaction strength γ. The Hamiltonian of the system reads

$$\mathcal{H} = \frac{p_1^2}{2m} + \frac{p_2^2}{2m} + \frac{1}{2}\kappa\left(x_1^2 + x_2^2\right) - \gamma|x_1 - x_2|. \tag{1}$$

The first two terms stand for the kinetic energy (with p_a the momentum of particle a), the third term corresponds to the quadratic potential energy of each particle, the fourth term is the interaction potential between the two particles which depends on their relative distance.

Integrating over the momentum degrees of freedom, the Boltzmann distribution gives the joint probability distribution density of the positions (x_1, x_2) of the two particles:

$$p(x_1, x_2) = \frac{1}{\mathcal{Z}}\exp\left[-\frac{\kappa}{2k_\mathrm{B}T}\left(x_1^2 + x_2^2\right) + \frac{\gamma}{k_\mathrm{B}T}|x_1 - x_2|\right] \propto \exp\left[-\frac{V(x_1, x_2)}{k_\mathrm{B}T}\right], \tag{2}$$

with $\mathcal{Z}$ the normalization constant. After some algebra (try it at home, this is a good statistical mechanics exercise!), we can prove that $\langle x_1 \rangle = \langle x_2 \rangle = 0$ while

$$\langle x_1^2 \rangle = \langle x_2^2 \rangle = \frac{k_\mathrm{B}T}{\kappa}\left[1 + \frac{\gamma}{\sqrt{\pi\kappa k_\mathrm{B}T}}\frac{\exp\left(-\dfrac{\gamma^2}{\kappa k_\mathrm{B}T}\right)}{1 + \mathrm{erf}\left(\dfrac{\gamma}{\sqrt{\kappa k_\mathrm{B}T}}\right)} + \frac{\gamma^2}{\kappa k_\mathrm{B}T}\right]. \tag{3}$$

In particular, when $\gamma = 0$, we recover the equipartition theorem ($\langle x_1^2 \rangle = \langle x_2^2 \rangle = k_\mathrm{B}T/\kappa$).

We want to use the Metropolis algorithm to sample this distribution. We propose the following Markov chain:

1. Pick at random one of the two particles.
2. Propose a new trial position y_a for the chosen particle a by randomly shifting its current position x_a by a random number between $-\epsilon$ and ϵ.
3. Accept or reject this trial position following Metropolis procedure.

For the simulations, we set $m = 1$, $\kappa = 1$ and $k_\mathrm{B} = 1$ (rescaled units).

Question 1: Implement the above Markov chain. You can choose $(x_1, x_2) = (0, 0)$ to initiate the chain.

Question 2: We start with $\gamma = 0$.

a. For several values of $T \in [0.2,\, 2]$ and for several values of $\epsilon \in [1,\, 10]$, perform $N = 10000$ steps and compute the average acceptance rate of trial moves. For each temperature, plot the acceptance rate as a function of ϵ and display all curves on the same graph. How should you choose ϵ such that the acceptance rate is roughly between $20\,\%$ and $50\,\%$ for all temperatures?

b. For several values of $T \in [0.2,\, 2]$, perform $N = 100000$ steps and plot the energy $V(x_1, x_2)$ as a function of the number of steps. How should you choose the number of steps $N_\mathrm{b-i}$ in the burn-in phase?

c. For the same simulations, compute the correlation function of x_1 in the equilibrium phase and plot the curve as a function of the number of steps. Estimate the number of steps N_corr over which the samples are correlated. How should you choose the number of steps N_eq in the equilibrium phase (with respect to N_corr)?

d. For several values of $T \in [0.2, 2]$, compute the variance of x_1 in the equilibrium phase and plot $\langle x_1^2 \rangle$ as a function of T. Is the equipartition theorem verified?

Question 3: We now study the case $\gamma = 1.0$ (in rescaled units).

a. For $T = 0.2$, perform $N = 10000$ steps and plot x_1 and x_2 as a function of the number of steps on the same graph. What do you observe? Can you rationalize this behavior? How should you choose the number of steps $N_{\mathrm{b-i}}$ in the burn-in phase?

b. For the same temperature, perform $N = 100000$ steps, compute the correlation function of x_1 in the equilibrium phase and plot it as a function of the number of steps. Estimate the number of steps N_{corr} over which the samples are correlated. How should you choose the number of steps N_{eq} in the equilibrium phase (with respect to N_{corr})? Compare with the case $\gamma = 0$.

c. For several values of $T \in [0.2, 2]$, compute the variance of x_1 in the equilibrium phase and plot $\langle x_1^2 \rangle$ as a function of T. Check that Eq. (3) is verified.

II. Monte Carlo integration

Question 1: Compute the volume of the unit ball (the set of vectors of norm smaller or equal than 1) in D dimensions using a Monte Carlo integration. Compare with the exact results $4\pi/3$ and $\pi^5/120$ for $D = 3$ and $D = 10$ respectively.

Question 2: We now want to compute numerically the integral

$$I = \int_0^1 \left(\frac{1}{x^{1/3}} + \frac{x}{10} \right) \mathrm{d}x. \tag{4}$$

Its exact value is $31/20$.

a. Compute the integral using a Monte Carlo integration with the uniform distribution for $N = 100$, $N = 1000$ and $N = 10000$ samples. For each value of N, you can do the calculation 20 times and average the result. Plot $|I - 31/20|$ as a function of N in a loglog plot. Comment on the accuracy of the method as a function of N (recall that the accuracy of Monte Carlo integration is of order $N^{-1/2}$).

b. We want to perform another Monte Carlo integration using the probability distribution density $p(x) = 2/(3x^{1/3})$ of support $I = [0, 1[$. Plot the functions $p(x)$ and $f(x) = 1/x^{1/3} + x/10$ on the same graph. Justify the choice of $p(x)$ (importance sampling).

c. Generate random numbers following the distribution $p(x)$ using the inverse transform sampling.

d. From the set of N random numbers obtained in the previous question, compute an estimate of I for $N = 100$, $N = 1000$ and $N = 10000$ samples (you can do the calculation 20 times and again average the result). Plot $|I - 31/20|$ as a function of N in a loglog plot. Comment on the accuracy of the method as a function of N and compare with the uniform distribution.

III. The best concert tour

A band wants to make a concert tour in France and has selected 13 cities among the biggest in Metropolitan France. The group wants to start from Paris, perform only once in each city and come back to Paris eventually. Because of ecological concerns, the band wants to travel the least number of kilometers. The geographical coordinates of the 13 French cities where they want to perform are listed in the array below. We recall that on Earth, given two locations of coordinates (λ_1, ϕ_1) and (λ_2, ϕ_2) (with λ_a the longitudes measuring the W/E deviation from the Greenwich Meridian and ϕ_a the latitudes measuring the N/S deviation from the Equator), their relative distance reads

$$d_{12} = R \arccos\left[\cos\phi_1 \cos\phi_2 \cos(\lambda_1 - \lambda_2) + \sin\phi_1 \sin\phi_2\right], \tag{5}$$

City	Longitude (O/E) λ	Latitude (N/S) ϕ
Paris	$2^\circ\ 21'\ 07''$ (E)	$48^\circ\ 51'\ 24''$ (N)
Lyon	$4^\circ\ 49'\ 56''$ (E)	$45^\circ\ 45'\ 28''$ (N)
Toulouse	$1^\circ\ 26'\ 38''$ (E)	$43^\circ\ 36'\ 16''$ (N)
Nice	$7^\circ\ 16'\ 17''$ (E)	$43^\circ\ 41'\ 45''$ (N)
Nantes	$1^\circ\ 33'\ 10''$ (O)	$47^\circ\ 13'\ 05''$ (N)
Montpellier	$3^\circ\ 52'\ 38''$ (E)	$43^\circ\ 36'\ 43''$ (N)
Strasbourg	$7^\circ\ 45'\ 08''$ (E)	$48^\circ\ 34'\ 24''$ (N)
Bordeaux	$0^\circ\ 34'\ 46''$ (O)	$44^\circ\ 50'\ 16''$ (N)
Lille	$3^\circ\ 03'\ 48''$ (E)	$50^\circ\ 38'\ 14''$ (N)
Le Havre	$0^\circ\ 06'\ 00''$ (E)	$49^\circ\ 29'\ 24''$ (N)
Clermont-Ferrand	$3^\circ\ 04'\ 56''$ (E)	$45^\circ\ 46'\ 33''$ (N)
Limoges	$1^\circ\ 15'\ 00''$ (E)	$45^\circ\ 51'\ 00''$ (N)
Orléans	$1^\circ\ 54'\ 32''$ (E)	$47^\circ\ 54'\ 09''$ (N)

Table 1: **Geographical coordinates of the 13 French cities chosen by a band for their French concert tour.** The longitudes λ and latitudes ϕ are given in degrees ($^\circ$), minutes ($'$) and seconds ($''$). We recall that $1^\circ = 60' = 3600''$.

with $R = 6371\,\mathrm{km}$ the average radius of the Earth.

A direct brute-force test of the $12! = 479001600$ possible paths takes about 30 minutes to converge and shows that the smallest tour is: Paris $\to$ Orléans $\to$ Limoges $\to$ Clermont-Ferrand $\to$ Lyon $\to$ Nice $\to$ Montpellier $\to$ Toulouse $\to$ Bordeaux $\to$ Nantes $\to$ Le Havre $\to$ Lille $\to$ Strasbourg $\to$ Paris. The objective of this exercise is to find circuits which minimize the total length travelled using a simulated annealing algorithm.

Question 1: Start from the Python code `band_circuit.py` and construct the $n \times n$ symmetric matrix d_{ij} of all distances between cities i and j (with n the number of cities).

Question 2: Compute the length L of the smallest tour described above.

Question 3: We propose the following simulated annealing scheme to find an approximation of the optimal tour. We start from a random circuit connecting all cities once, starting from Paris and coming back to Paris. At each step, we propose the following procedure:

1. Try to swap two cities in the path except Paris and compute the new length of the circuit.
2. Accept and reject the new trial path following the Metropolis criterion assuming that the probability of a path of total length d is $\propto e^{-d/T}$ (with T the temperature).
3. Update the temperature T by decreasing it by a factor Δ at each step.

The simulated annealing scheme starts with an initial temperature T_{i} and ends with a final temperature T_{f}. Implement the above procedure.

Question 4: The performance of the algorithm depends on the annealing rate Δ.

a. For $T_{\mathrm{i}} = 100$ and $T_{\mathrm{f}} = 10^{-5}$, run the algorithm 20 times for $\Delta = 10^{-1},\ 10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}$ and compute the average length $\tilde{L}(\Delta)$ of the optimal path.

b. Plot $\tilde{L}(\Delta) - L$ as a function of Δ in a loglog plot. How does the difference scales with Δ?

Question 4: For $\Delta = 10^{-4}$, $T_{\mathrm{i}} = 100$ and $T_{\mathrm{f}} = 10^{-5}$, run the simulated annealing algorithm and plot the entire path at the end of the simulated annealing procedure. Is the simulated annealing algorithm able to find the minimal path?

IV. Parameter inference from experimental data

We are interested in biological data on blood types, see the array below. The observed blood types A, B, AB or O (phenotype) depend on the pair of two alleles of the genotype and on their dominance properties. We want to estimate the probabilities p_{A}, p_{B} and p_{O} that an allele is represented in the population (where $p_{\mathrm{A}}+p_{\mathrm{B}}+p_{\mathrm{O}}=1$).

Genotype	Probability	Phenotype	Number of observed phenotypes
AA AO	p_{A}^2 $2p_{\mathrm{A}}p_{\mathrm{O}}$	A	$N_{\mathrm{A}}=186$
BB BO	p_{B}^2 $2p_{\mathrm{B}}p_{\mathrm{O}}$	B	$N_{\mathrm{B}}=38$
AB	$2p_{\mathrm{A}}p_{\mathrm{B}}$	AB	$N_{\mathrm{AB}}=13$
OO	p_{O}^2	O	$N_{\mathrm{O}}=284$

Table 2: **Results from a biology experiment.** The blood types of $N=521$ people and their corresponding genotype are reported.

Given the allele probabilities p_{A}, p_{B}, p_{O}, the probability to observe such values of N_{A}, N_{B}, N_{AB} and N_{O} are given by a multinomial distribution:

$$P(N_{\mathrm{A}},N_{\mathrm{B}},N_{\mathrm{AB}},N_{\mathrm{O}})=\frac{N!}{N_{\mathrm{A}}!N_{\mathrm{B}}!N_{\mathrm{AB}}!N_{\mathrm{O}}!}(p_{\mathrm{A}}^2+2p_{\mathrm{A}}p_{\mathrm{O}})^{N_{\mathrm{A}}}(p_{\mathrm{B}}^2+2p_{\mathrm{B}}p_{\mathrm{O}})^{N_{\mathrm{B}}}(2p_{\mathrm{A}}p_{\mathrm{B}})^{N_{\mathrm{AB}}}p_{\mathrm{O}}^{2N_{\mathrm{O}}}, \tag{6}$$

with $N=N_{\mathrm{A}}+N_{\mathrm{B}}+N_{\mathrm{AB}}+N_{\mathrm{O}}$. To estimate the allele probabilities, we propose to maximise the above probability with respect to p_{A}, p_{B} and p_{O}, which amounts to maximizing the following likelihood

$$L(p_{\mathrm{A}},p_{\mathrm{B}},p_{\mathrm{O}})=N_{\mathrm{A}}\ln(p_{\mathrm{A}}^2+2p_{\mathrm{A}}p_{\mathrm{O}})+N_{\mathrm{B}}\ln(p_{\mathrm{B}}^2+2p_{\mathrm{B}}p_{\mathrm{O}})+N_{\mathrm{AB}}\ln(p_{\mathrm{A}}p_{\mathrm{B}})+2N_{\mathrm{O}}\ln(p_{\mathrm{O}}) \tag{7}$$

with the constraint $p_{\mathrm{A}}+p_{\mathrm{B}}+p_{\mathrm{O}}=1$.
A direct calculation using a Lagrange multiplier to enforce the above constraint leads to tedious algebra and multiple solutions. We instead propose an algorithm to find the maximum. We introduce the two hidden variables Z_{A} and Z_{B} corresponding to the number of allele pairs (A,A) and (B,B) respectively. Given Z_{A} and Z_{B}, the new likelihood to maximize reads

$$\begin{aligned}L'(p_{\mathrm{A}},p_{\mathrm{B}},p_{\mathrm{O}};Z_{\mathrm{A}},Z_{\mathrm{B}})=&\,2Z_{\mathrm{A}}\ln(p_{\mathrm{A}})+(N_{\mathrm{A}}-Z_{\mathrm{A}})\ln(2p_{\mathrm{A}}p_{\mathrm{O}})+2Z_{\mathrm{B}}\ln(p_{\mathrm{B}})+(N_{\mathrm{B}}-Z_{\mathrm{B}})\ln(2p_{\mathrm{B}}p_{\mathrm{O}})\\&+N_{\mathrm{AB}}\ln(p_{\mathrm{A}}p_{\mathrm{B}})+2N_{\mathrm{O}}\ln(p_{\mathrm{O}}).\end{aligned} \tag{8}$$

By construction Z_{A} is a random variable which follows a binomial distribution of parameters N_{A} (number of trials) and $p_{\mathrm{A}}^2/(p_{\mathrm{A}}^2+2p_{\mathrm{A}}p_{\mathrm{O}})$ [relative probability to observe a pair of alleles (A,A)]. Similarly, Z_{B} follows a binomial distribution of parameters N_{B} and $p_{\mathrm{B}}^2/(p_{\mathrm{B}}^2+2p_{\mathrm{B}}p_{\mathrm{O}})$.

We propose the following Expectation–Maximization algorithm to find the allele probabilities, starting from random values of p_{A}, p_{B} and p_{O}.

1. At step n, for the estimated values $p_{\mathrm{A}}^{(n)}$, $p_{\mathrm{B}}^{(n)}$ and $p_{\mathrm{O}}^{(n)}$ of the allele probabilities, compute the average of L' with respect to Z_{A} and Z_{B}. We denote this average $L_{\mathrm{avg}}(p_{\mathrm{A}},p_{\mathrm{B}},p_{\mathrm{O}})$.

2. Update the allele probabilities by setting their values at step $n+1$ to the location of the maximum of the average of L_{avg} computed before:

$$(p_{\mathrm{A}}^{(n+1)},p_{\mathrm{B}}^{(n+1)},p_{\mathrm{O}}^{(n+1)})=\mathrm{argmax}_{(p_{\mathrm{A}},p_{\mathrm{B}},p_{\mathrm{O}})}L_{\mathrm{avg}}(p_{\mathrm{A}},p_{\mathrm{B}},p_{\mathrm{O}}). \tag{9}$$

3. Stop the algorithm when you reach fixed points for the allele probabilities, namely, when the relative change of the probabilities between two steps is smaller than a given threshold ϵ. If such a criterion cannot be met, stop the algorithm after a given number of steps.

Question 1: Implement the function $L_{\mathrm{avg}}(p_{\mathrm{A}}, p_{\mathrm{B}}, p_{\mathrm{O}})$. You can simplify its expression using pen and paper before coding it, or you can simply generate samples of Z_{A} and Z_{B} following binomial distributions.

Question 2: Implement the part of the algorithm which finds the location of the maximum of L_{avg}. You can first use pen and paper to find its expression before coding it, or you can directly use optimization routines implemented in Python.

Question 3: Find the best estimates of p_{A}, p_{B} and p_{O} using the above algorithm.