Analyse Numérique / Computational Physics (M1) - T. Roscilde, F. Caleca

General rule for the TD sessions: the TD sessions are fully *hands-on* – namely, in every TD session you are supposed to *write computer codes* to learn about the phenomenology and efficiency of important algorithms, or, more ambitiously, to learn the physics of simple, yet fundamental models. You should choose a programming platform (Python, Matlab, Mathematica, C, Fortran, etc.), and you should be able to plot your results in the form of two-dimensional functions $y = f(x)$ (using matplotlib in Python, the plotting utilities of Matlab and Mathematica, Gnuplot, etc.), or occasionally in a more complicated form. We assume that you have some familiarity with at least one programming platform; if this is not the case, you should be able to familiarize yourself rapidly *e.g.* by attending online tutorials.

TD2: Monte Carlo

In this exercise sheet, we shall get acquainted with the statistical analysis of a series of correlated random variables; and we will later use the Monte Carlo scheme to calculate some integrals of interest in physics.

1 Statistical analysis of a series of random variables

In this programming exercise, we will make use of a *random number generator* (RNG) to revise some elementary aspects of the statistics of a series of random variables, and to extract their correlation time.
To be able to work out the exercise, as well as all the exercises in this sheet, you need to get a handle on a RNG, namely a function (or subroutine) which produces (pseudo-)random numbers z uniformly distributed in the interval $[0, 1]$. Python and Matlab (for instance) contain a built-in function producing (pseudo-)random numbers; the built-in RNG of Fortran and C is generally not recommendable, but online you can easily find subroutines to include in your code (e.g. the *ran2* subroutine from Numerical Recipes).

1.1

In this first exercise we will do a simple analysis of the output of the RNG. Write a program in which you perform $M = n_b M_b$ subsequent extractions of a random number z_i ($i = 1, ..., M$) structured in n_b blocks of length M_b each, and define the block variables Z_a ($a = 1, ..., n_b$)

$$Z_a = \frac{1}{M_b} \sum_{j=1}^{M_b} z_{i=(a-1)*M_b+j} \tag{1}$$

which average the variables in the a-th block. Calculate numerically the averages

$$\begin{aligned} \langle z \rangle &= \frac{1}{M} \sum_{i=1}^{M} z_i & \langle z^2 \rangle &= \frac{1}{M} \sum_{i=1}^{M} z_i^2 \\ \langle Z \rangle &= \frac{1}{n_b} \sum_{a=1}^{n_b} Z_a & \langle Z^2 \rangle &= \frac{1}{n_b} \sum_{a=1}^{n_b} Z_a^2 \end{aligned} \tag{2}$$

and build the corresponding variances

$$\sigma_z^2 = \langle z^2 \rangle - \langle z \rangle^2 \qquad \sigma_Z^2 = \langle Z^2 \rangle - \langle Z \rangle^2 \tag{3}$$

You should verify numerically that, for $M_b \gg 1$ ($M_b \gtrsim 10$, sufficiently large for the central-limit theorem to apply), then

$$\sigma_z^2 \approx M_b \; \sigma_Z^2 \tag{4}$$

This offers some evidence that your RNG produces uncorrelated random numbers.

1.2

We now introduce some artificial correlations in the random numbers, by generating a random sequence θ_i with the following procedure: given the random number at step $i-1$, θ_{i-1}

- extract a random number ξ;
- if ξ and θ_{i-1} are both comprised in $[0, 1/2]$ or in $[1/2, 1]$, then $\theta_i = \xi$;
- otherwise, extract a random number $\eta \in [0, 1]$: if $\eta < p$ then $\theta_i = \theta_{i-1}$, otherwise $\theta_i = \xi$.

In this way the numbers θ_{i-1} and θ_i have a higher probability of being in the same half of the $[0, 1]$ interval than in different halves. You can start by choosing with $p = 1/2$ (but it may be interesting to vary p, see below).
In the exact same way as in the previous exercise, introduce the block variables

$$\Theta_a = \frac{1}{M_b} \sum_{j=1}^{M_b} \theta_{i=(a-1)*M_b+j} \tag{5}$$

and calculate $\langle \theta \rangle$, $\langle \theta^2 \rangle$, $\langle \Theta \rangle$, $\langle \Theta^2 \rangle$ and use them to calculate σ_θ^2 and σ_Θ^2.
You should observe that

$$\sigma_\theta^2 < M_b \; \sigma_\Theta^2 \; . \tag{6}$$

Use these quantities to estimate the *autocorrelation time*

$$\tau = \frac{1}{2} \; \frac{\sigma_\Theta^2}{\sigma_\theta^2} \; M_b \quad (> 1/2) \; . \tag{7}$$

You can control τ by varying p (the larger p, the larger τ).

1.3

If you plot the sequence of values θ_i you should observe that they vary over a characteristic "time scale" which is comparable with τ. Can you see this?

2 Fourier transforms with Monte Carlo

In this exercise we shall calculate numerically the Fourier transform of a couple of relevant functions using Markov-chain Monte Carlo. The Fourier transform of the $f(x)$ function is defined as:

$$\tilde{f}(k) = \int \frac{dx}{\sqrt{2\pi}} \; f(x) \; e^{-ikx} \; . \tag{8}$$

If $f(x) = f(-x)$, then $\tilde{f}(k) \in \mathbb{R}$ and one can replace e^{-ikx} with $\cos(kx)$.
In particular we have that for a Gaussian function $f(x) = \exp(-x^2)$, $\tilde{f}(k) = e^{-k^2/4}/\sqrt{2}$.

2.1

As a first exercise, we would like first to write a Markov-chain Monte Carlo scheme in order to verify the latter result on the Fourier transform of a Gaussian, by using the normalized probability distribution

$$p(x) = \frac{e^{-x^2}}{\sqrt{\pi}} \tag{9}$$

so that

$$\tilde{f}(k) = \frac{1}{\sqrt{2}} \int dx\ p(x)\ \cos(kx) = \frac{1}{\sqrt{2}} \langle\ \cos(kx)\ \rangle_p\ . \tag{10}$$

Build a Markov-chain Monte Carlo scheme with the following ingredients:

- Equilibration phase (comprising $M_{\rm eq}$ iterations);
  - 1) pick an initial position x (e.g. $x = 0$);
  - 2) propose a new position $x' = x + \delta x$ with δx a random number in the interval $[-\Delta, \Delta]$ [1];
  - 3) accept the move with Metropolis-Hastings probability $A(x \to x') = \min(1, p(x')/p(x))$;
  - 4) go to point 2);
- Measurement phase (comprising M iterations)
  - go over points 2)-3) of the equilibration phase, but in addition accumulate the values of $\cos(kx)/\sqrt{2}$ to build the Monte Carlo estimate

$$\tilde{f}_M(k) = \frac{1}{M} \sum_{i=1}^{M} \frac{\cos(kx_i)}{\sqrt{2}} \approx \tilde{f}(k)\ . \tag{11}$$

How to choose $M_{\rm eq}$ and M? You can take $M_{\rm eq} \approx 1000$ and $M \approx 10000$ (or smaller/larger, depending on the speed of execution of your computer).
Repeating the above procedure for a grid of k points (e.g. $k = 0, 0.1, 0.2, ..., 1.0$) verify that you obtain results coherent with the Fourier transform of the Gaussian.

2.2 (Bonus question)

If you could successfully implement the previous calculation, you are now all set to calculate Fourier transforms of functions that cannot be calculated exactly. One such example is given by

$$f(x) = \exp(-x^2 - \epsilon x^4) \tag{12}$$

whose Fourier transform can also be thought of as

$$\tilde{f}(k) = \frac{1}{\sqrt{2}} \left\langle\ \cos(kx)\ e^{-\epsilon x^4}\ \right\rangle_p\ . \tag{13}$$

The Monte Carlo calculation of $\tilde{f}(k)$ goes exactly as in the previous question, with a simple modification of the function to average. You can choose e.g. $\epsilon = 0.1$, and compare your result for $\tilde{f}(k)$ with the Fourier transform of a Gaussian.

[1] You can choose $\Delta = 0.1$ as a starter; taking a much smaller Δ will increase the acceptance rate of the updates but it introduces correlations between successive steps, while taking a much larger Δ should suppress the acceptance rate quite significantly.

3 Deviation from equipartition in a non-linear harmonic oscillator

Consider a classical anharmonic oscillator in one dimension, with unit mass and frequency, whose Hamiltonian reads:

$$\mathcal{H} = \frac{p^2}{2} + \frac{x^2 + \epsilon x^4}{2} \ . \tag{14}$$

Imagining the oscillator to be at equilibrium with a thermal bath at temperature T, the average potential energy reads

$$\left\langle \frac{x^2 + \epsilon x^4}{2} \right\rangle_T = \frac{1}{\mathcal{N}} \int dx \left(\frac{x^2 + \epsilon x^4}{2} \right) e^{-\frac{1}{2k_B T}(x^2 + \epsilon x^4)} \tag{15}$$

where $\mathcal{N}$ is a normalization factor.
In the absence of anharmonicity ($\epsilon = 0$) this average value satisfies the so-called energy equipartition

$$\left\langle \frac{x^2}{2} \right\rangle_T = \frac{1}{2} k_B T \ . \tag{16}$$

3.1

Starting from the case $\epsilon = 0$, build a Markov-chain Monte Carlo scheme as in the previous exercise in order to calculate $\left\langle \frac{x^2}{2} \right\rangle_T$; verify the energy equipartition by repeating the calculation for several values of $k_B T$.

3.2

Adding a small $\epsilon = 0.2$, calculate the non-quadratic potential energy $\left\langle \frac{x^2 + \epsilon x^4}{2} \right\rangle_T$ by using Markov-chain Monte Carlo, and show numerically that equipartition is violated at sufficiently high temperature (namely, the average potential energy is no longer linear in T). This should happen when $\langle x^2 \rangle_T \approx \epsilon \langle x^4 \rangle_T$.