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.

TD3: Numerical optimization

In this exercise sheet, we shall focus our attention on the search of the ground-state configuration for two charged particles moving in one dimension, immersed in an external potential. This is a simple model e.g. of the physics of two ions trapped in the same common potential, as realized in many atomic-physics experiments.
The cost function to be minimized is the potential energy expressed in terms of the positions x_1 and x_2 of the two particles, which reads

$$V(x_1, x_2) = \frac{1}{|x_1 - x_2|} + a_2(x_1^2 + x_2^2) + a_4(x_1^4 + x_2^4) \ . \tag{1}$$

Here the first term is the Coulomb repulsion between the two particles (in dimensionless units), while the last two terms define a quadratic-plus-quartic potential to which both particles are subject.

1 Gradient descent

In this part of the exercise sheet we will use the gradient method to find the minimum-energy configuration.

1.1

Calculate the gradient of the cost function $\boldsymbol{\nabla} V(x_1, x_2)$ (to calculate the derivative of $|x_1 - x_2|$ it may be convenient to express is at $\sqrt{(x_1 - x_2)^2}$).

1.2

Let us start from the case of a simple quadratic potential ($a_2 = 1$, $a_4 = 0$). Plot the surface $V(x_1, x_2)$: you should conclude that there is a single minimum (modulo a particle-exchange symmetry).

1.3

Write a program which picks an initial point at random $\boldsymbol{x}_0 = (x_1^{(0)}, x_2^{(0)})$ in the interval $[-2, 2] \times [-2, 2]$ (for instance), and updates it according to the gradient-descent method

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \epsilon \boldsymbol{\nabla} V(\boldsymbol{x}_k) \ . \tag{2}$$

The choice of ϵ is up to you – it should be small enough to not jump too far ahead and guarantee that $V(\boldsymbol{x}_{k+1}) \leq V(\boldsymbol{x}_k)$, but also large enough so that it does not take too long to get to the minimum. You could choose $\epsilon = 0.1$ as a starting value.
You can decide to stop the algorithm at a step k^* such that $|\boldsymbol{\nabla} V(\boldsymbol{x}_{k^*})| \leq \eta$ with η a parameter of your choice (e.g. $\eta = 10^{-3}$). You can test whether this will bring you sufficiently close to the minimum that you have already identified visually by plotting the function.

1.4

Save the sequence of values $V_k = V(\boldsymbol{x}_k)$ that the algorithm explores, and plot it as a function of k; you should see that $V_k - V_{k^*}$ converges to zero: can you say how it does so? You can check this by plotting V_k vs. k in a log-log scale (which should single out a power-law decay) or in a log-lin scale (which should single out an exponential decay).

1.5

You can also save the sequence of points $\boldsymbol{x}_k$, and plot $x_2^{(k)}$ vs. $x_1^{(k)}$ to understand what the algorithm did on the way to finding the minimum.

1.6

You can now turn on the quartic part of the potential and change the sign of the quadratic one, for instance you can take: $a_2 = -6$, $a_4 = 1$. Visualizing the function $V(\boldsymbol{x})$ again, you should observe that a new set of minima has appeared: can you make sense of them physically?

1.7

Now you can play with the new potential by repeating the gradient-descent optimization: you should find that, depending on the initial point $\boldsymbol{x}_0$, you can fall in one of the three inequivalent minima of the potential energy. This should show you how the gradient search can fail to find the global minimum.

2 Improvements on gradient descent

We now modify the cost function in order to observe improvements on the gradient descent. Let us consider two following function

$$U(x_1, x_2) = (x_1 - x_2/2)^4 + x_1^6 \tag{3}$$

which has a shallow minimum at $(x_1, x_2) = (0, 0)$.

2.1

To start, search for the minimum with the gradient descent algorithm again, checking how $U_k - U_{k^*}$ converges to zero (as you did in the previous exercise).

2.2

Let us now modify the algorithm to add a "momentum term" *à la* Polyak. The "heavy-ball" method amounts to the new update rule:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \epsilon \boldsymbol{\nabla} U(\boldsymbol{x}_k) \ + \beta(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \ . \tag{4}$$

You can implement it by producing $\boldsymbol{x}_0$ as a random point, $\boldsymbol{x}_1$ from $\boldsymbol{x}_0$ via the gradient algorithm, and use the heavy-ball approach starting from $\boldsymbol{x}_2$.
Using the new potential U, it would be useful that you use the same $\boldsymbol{x}_0$ point as in the case of gradient descent, so that you can compare the results of the two approaches. You can choose a small $\beta = 0.1$, or a larger one $\beta = 0.5$ (or an even larger one *e.g.* $\beta = 0.9$); and compare the results with respect to the case $\beta = 0$, by plotting U_k vs. k for the various methods when starting from the same initial value U_0. You should not be surprised to see that, under the addition of the momentum term, U_k is no longer monotonically decreasing (this is the effect of "inertia" on the dynamics!).

2.3

We can further modify the algorithm by accelerating it *à la* Nesterov, namely using the following update scheme:

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{y}_{k-1} - \epsilon \boldsymbol{\nabla} U(\boldsymbol{y}_{k-1}) \\ \boldsymbol{y}_k &= \boldsymbol{x}_k + \frac{k-1}{k+2} (\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \end{aligned} \tag{5}$$

which you can implement (starting from $\boldsymbol{x}_2$) after having picked $\boldsymbol{x}_0$ at random, and $\boldsymbol{x}_1$ using one step of gradient descent.
Choosing one of the cases before (either the quadratic potential, or the quadratic + quartic potential), you can monitor the decrease of $U_k - U_{k^*}$ vs. k, and verify whether you see the expected acceleration with respect to the two previous methods (simple gradient and heavy ball).

3 Simulated annealing

As we have seen above, the simple gradient descent is a deterministic algorithm which brings you systematically to the local minimum which is "closest" to the initial point (in the sense that the initial point falls in its basin of attraction). A similar problem is shared with the improved versions of the gradient descent.
An alternative to these minimization algorithms is offered by simulated annealing, which, in the problem at hand, amounts to simulating the equilibrium statistical physics of the two particles in the potential $V(\boldsymbol{x})$ by making a Monte Carlo simulation at variable temperature. We now go back to the first cost function we studied (the one motivated by the search for the equilibrium position of two charged particles).

3.1

Picking an initial point $\boldsymbol{x}_0$ at random, and introducing a fictitious temperature T, update the point according to the Metropolis algorithm:

- pick a random displacement $\delta\boldsymbol{x} = (\delta x_1, \delta x_2)$ in the box $[-\Delta, \Delta] \times [-\Delta, \Delta]$;
- calculate the energy variation $\Delta V = V(\boldsymbol{x}_k + \delta\boldsymbol{x}) - V(\boldsymbol{x}_k)$;
- extract a random number $z \in [0, 1]$; if $z \leq \min[1, \exp(-\Delta V/T)]$ then $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \delta\boldsymbol{x}_k$, otherwise $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k$.

The value of Δ (the size of the box within which you propose the next point) should be carefully chosen in order for the proposed moves to be accepted at a good rate. For the moment you can choose $\Delta = 0.1$.

For a fixed value of the temperature, you should repeat this procedure several times (say $M = 100$ for instance), and then lower the temperature to repeat the operation. How you decrease the temperature defines the so-called "annealing schedule". You can start from $T = 10$, and then decrease T in steps of ΔT (with $\Delta T = 0.2$, for instance). The algorithm stops when you have reached $T = 0$.

To test simulated annealing, the most interesting situation is to choose the quadratic-plus-quartic potential ($a_2 = -1, a_4 = 6$), and see whether a particular annealing schedule is able to find the absolute minimum for you. By repeating the simulation for different values of ΔT and M, you should observe that you are less likely to get trapped in the local minimum for a slow annealing schedule (a smaller ΔT and/or a larger M).