Introduction to data assimilation
Practical session Day 1: elementary tools

Exercice D1.0.
In this course we will compute the solutions of ODEs or ODE systems using the function `scipy.integrate.solve_ivp`.
Read the official documentation on this function.

Exercice D1.1. **Numerical solution of an ODE: logistic growth**
We consider the following ODE that models the logistic growth of a population:

$$\begin{cases} y'(t) = \alpha y(t)(1 - y(t)/K), & t \in [0, T], \\ y(t = 0) = y_0, & t = 0. \end{cases}$$

for some given constants $\alpha > 0$ and $K > 0$. The initial population is $y_0 > 0$.

a) implement the numerical solution of this ODE, and plot the solution.

b) change the values of α, K and y_0 (one parameter at a time !).

Note: an analytical solution of the logistic equation can be calculated. We propose not to use this analytical solution.

Exercice D1.2. **Numerical solution of an ODE system: SIR epidemiologic model**
We consider the following ODE system (SIR model) that models the spread of an epidemics in a population:

$$\begin{cases} S'(t) = -\tau S(t)I(t), \\ I'(t) = \tau S(t)I(t) - \nu I(t) \\ R'(t) = \nu I(t), & t \in [0, T], \end{cases}$$

together with the initial condition: $S(t = 0) = N, I(t = 0) = 1, R(t = 0) = 0$.
The given constants $\tau > 0$ and $\nu > 0$ represent respectively the force of the infection and the recovery/death rate. Note that in this simple model a person that was Infected becomes Removed at a rate ν, and the model does not distinguish between people that recovered from the infection and those that died !

a) Implement the numerical solution of this ODE system on the time interval $[0, 250]$ for the following values: $\tau = 3.10^{-9}, \nu = 10^{-1}$ and the initial conditions $N = 6.10^7$, and plot the solution.

b) Change the values of τ and ν (one parameter at a time !).

We aim at modelling a public intervention to stop the pandemics spread. This public intervention (e.g. lock down) has the effect on the spread force τ that is multiplied by a factor $f \in (0, 1)$ between the dates t_1 and t_2.

c) Modify the function that implements the SIR model to incorporate the effect of multiplying τ by the factor f between the dates t_1 and t_2.

d) Apply on the previous simulation the spread force reduction by a factor of 10 between $t_1 = 80$ and $t_2 = 130$. Comment the results.

e) Modify the factor f.

Exercice D1.3. Optimization of a quadratic function: gradient method, Gauss-Newton method

We consider a symmetric positive definite (sdp) matrix A of size $n \times n$, and a vector $\mathbf{x}_* \in \mathbf{R}^n$. The objective function that is defined by

$$J(\mathbf{x}) = \frac{1}{2}\|A(\mathbf{x} - \mathbf{x}_*)\|^2$$

attains its minimum value at the point $\mathbf{x}_*$. Therefore we know the 'exact' minimizer and we will be able to evaluate the performance of methods designed to find this minimizer.

a) Let $\mathbf{h} \in \mathbf{R}^n$. Compute the quantity

$$DJ(\mathbf{x}).\mathbf{h}.$$

What is the gradient of J at the point $\mathbf{x}$?

b) One can prove that the Lipschitz constant of the gradient of J is the largest eigenvalue L of the matrix $A^T A$. Propose an implementation of the gradient descent method applied to J with starting point $\mathbf{0}$.

c) Choose a 3×3 sdp matrix A. Implement 100 iterations of the gradient descent method. For each iterate $\mathbf{x}^k$ compute

- the distance to the true minimizer $\|\mathbf{x}^k - \mathbf{x}_*\|$,
- the value of the cost function $J(\mathbf{x}^k)$,

and plot these quantities vs k in two different graphics. One can use a semi-log scale.

d) Choose the following 2×2 sdp matrix:

$$A = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}$$

and perform the same study. Plot in the plane the iterates $\mathbf{x}^k$ together with the level curves of the function J and the true minimizer $\mathbf{x}_*$.

We aim to minimize the same quadratic function using a Gauss-Newton method. One can note that

$$J(\mathbf{x}) = \frac{1}{2}\|F(\mathbf{x})\|^2, \qquad \text{with} \qquad F(\mathbf{x}) = A(\mathbf{x} - \mathbf{x}_*).$$

e) What is the Jacobian matrix DF of F ? Propose an implementation of the Gauss-Newton descent method applied to J.

f) Perform the same studies as questions c) and d) above.

Exercice D1.4. Optimization of a non linear function (Rosenbrock function): gradient method, GN method.

We consider the Rosenbrock function in dimension 2, defined by

$$j(x, y) = (x - 1)^2 + 100(y - x^2)^2.$$

The minimizer is the point $\mathbf{x}^* = (x^*, y^*) = (1, 1)$, it is located at the bottom of a valley in the shape of a parabola. This function (and higher dimensional generalizations) is an example of a 'difficult' function to minimize.

a) What is the gradient of j at the point $\mathbf{x} = (x, y)$?

b) The gradient of j is not Lipschitz in the whole plane $\mathbf{R}^2$ but it is in bounded domains. We assume that $x, y \in [-M, M]$. Check that the Lispchitz constant of ∇j in this domain can be bounded by

$$L \leq 202 + 800M + 600M^2.$$

c) Implement 100 iterations of the gradient descent method with fixed step size. Use as a starting point a random point with coordinates between 0 and 1. For each iterate $\mathbf{x}^k$ compute

- the distance to the true minimizer $\|\mathbf{x}^k - \mathbf{x}^*\|$,
- the value of the cost function $j(\mathbf{x}^k)$,

and plot these quantities vs k in two different graphics. One can use a semi-log scale.

d) Increase the number of iterates to reach the minimizer up to machine precision.

We now aim to implement Gauss-Newton method.

e) Check that one can write

$$j(\mathbf{x}) = \frac{1}{2}\|F(\mathbf{x})\|^2,$$

for a well chosen function F to be determined.

f) What is the Jacobian matrix DF of F ? Propose an implementation of the Gauss-Newton descent method applied to J.

g) Perform the same studies as questions c) and d) above.