Numerical Optimization Lab 03:
Steepest Descent and the Lotka-Volterra Model

Francesco Della Santa

Abstract

In this lesson, we will learn to use the *steepest descent* optimization method for a real world case study.

1 Introduction

In the working folder of the Numerical Optimization laboratories, from the web page of the course download the files: *Lotka_Volterra_E_Espl.m* and *PoliTO_preypreds_guided.mat*.

2 Exercises

Exercise 1 (Finding Coefficients of the Lotka-Volterra Model)**.** Write in the command window `help Lotka_Volterra_E_Espl` and read which are the meanings of the inputs and outputs of the function (before, read Appendix A.1). Then, using the variables inside *PoliTO_preypreds_guided.mat*, run the command

```
[xn,yn,x_eq,y_eq,tn] = Lotka_Volterra_E_Espl(a,b,c,d,x0,y0,t0,T,N)
```

and plot the numerical solutions as illustrated in Figure 1.

Let us assume that the solutions just computed represent the evolution of a population of deers and a population of wolves in a natural park. Let the table `preypred_samples`[1] represents the population samplings made by the natural scientists.

Using the values in `preypred_samples`, implement the procedure described in Appendix A.2 using the steepest descent function defined in the previous laboratory. Assume that the only known data are the ones inside the table `preypred_samples`.

In particular, when you try to find the coefficients of the Lotka-Volterra model, use the following parameters for the steepest descent:

- $(a_0, b_0, c_0, d_0) = (1, 1, 1, 1)$, $\alpha = 10^{-6}$, $k_{\max} = 10^5$, $toll = 10^{-16}$;
- $(a_0, b_0, c_0, d_0) = (0.1, 0.1, 0.1, .0.1)$, $\alpha = 10^{-6}$, $k_{\max} = 10^5$, $toll = 10^{-16}$;

How the result changes, varying the steepest descent initialization?

Suggestion: use the functions `contour`, `dist`, `min` and `round` for the computation of the better approximation $(\hat{x}_0, \hat{y}_0)$ of the starting populations.

[1]variable saved in *PoliTO_preypreds_guided.mat*.

A Preys-Predators Interaction Analysis

In this appendix we briefly introduce the Lotka-Volterra model for the analysis of the evolution of two animal populations: a population $x(t)$ of preys and a population $y(t)$ of predators.

A.1 The Lotka-Volterra Model

The Lotka-Volterra model, tells us that the evolution of two interacting prey-predator populations follows these dynamic rules:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}x(t) = a\,x(t) - b\,x(t)y(t) \\ \frac{\mathrm{d}}{\mathrm{d}t}y(t) = -c\,y(t) + d\,x(t)y(t) \end{cases}, \tag{1}$$

where the coefficients are such that:

$a \in \mathbb{R}$**:** is the *reproduction* coefficient of the preys;

$b > 0$**:** is the *death* coefficient of the preys due to predators' hunting;

$c > 0$**:** is the *death* coefficient of the predators;

$d > 0$**:** is the *survival & reproduction* coefficient of the predators due to successful huntings.

The solutions $x(t)$ and $y(t)$ are not analytically defined but good approximations can be numerically computed.

Given a starting time t_0, given the starting populations $(x(t_0), y(t_0))$ and *given the coefficients* a, b, c, d, an approximation of $x(t)$ and $y(t)$ can be computed using the explicit Euler method, i.e.

$$\begin{cases} x_{n+1} = x_n + (a\,x_n - b\,x_n y_n)\Delta t \\ y_{n+1} = y_n + (-c\,y_n + d\,x_n y_n)\Delta t \end{cases}, \quad \forall\, n \geq 0\,, \tag{2}$$

where Δt is a *"very thin"* time step, $(x_0, y_0) = (x(t_0), y(t_0))$, and $x_n \approx x(t_0 + n\Delta t)$, $y_n \approx y(t_0 + n\Delta t)$ for each $n \geq 1$.

Assuming $a > 0$, if we plot the sequences $\{x_n\}_{n \in \mathbb{N}}$ and $\{y_n\}_{n \in \mathbb{N}}$ with respect to time we observe a periodic behavior for both the populations (see Figure 1-left); if we plot instead the approximated solutions in the (x, y) plane, we observe a closed curve around the stable equilibrium point $(x_{eq}, y_{eq}) = (c/d, a/b)$ (see Figure 1-right)

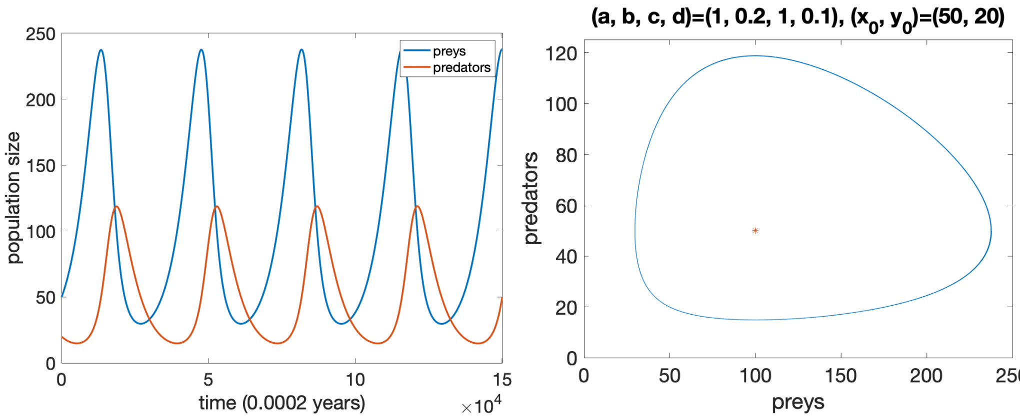

Figure 1: Approximated solutions for the Lotka-Volterra model.

At last, a deeper analysis of the model reveals that the orbit plotted in Figure 1-right is actually a level curve of the function

$$H(x(t), y(t)) := c\log(x(t)) - dx(t) - by(t) + a\log(y(t))\,; \tag{3}$$

i.e., for each instant $t > 0$, $H(x(t), y(t))$ is constant and equal to $H(x_0, y_0)$.

A.2 Finding Coefficients from Population Samplings: an "Everyday-Life" Problem

The dynamic system of Lotka-Volterra is extremely useful in studying and predicting the behavior of two animal populations. However, in practice, natural scientists often have a set of population samplings but no idea about the values $a = a^*, b = b^*, c = c^*, d =^*$ of the coefficients characterizing the model; therefore, they can't use the model at all!

Using numerical optimization methods, we can help the natural scientists finding an approximation of a^*, b^*, c^*, d^*, given some population samples.

Let us assume that the scientists have sampled N times the prey and predator populations at the time instants $t_0 < \cdots < t_{N-1}$, obtaining the set

$$\{(\tilde{x}_0, \tilde{y}_0), \dots, (\tilde{x}_{N-1}, \tilde{y}_{N-1})\}\,, \tag{4}$$

where $(\tilde{x}_i, \tilde{y}_i)$ is an approximation of the unknown real population values $(x(t_i), y(t_i))$, for each $i = 0, \dots, (N-1)$.

Due to the approximation errors of the samplings, we have that very probably $H(\tilde{x}_i, \tilde{y}_i) \neq H(x_0, y_0)$, for each $i = 0, \dots, (N-1)$; however, we can assume that *in the average* (especially if $N \gg 1$) the values $H(\tilde{x}_i, \tilde{y}_i)$ approximate $H(x_0, y_0)$, i.e.

$$\bar{H} := \frac{1}{N} \sum_{i=0}^{N-1} H(\tilde{x}_i, \tilde{y}_i) \approx H(x_0, y_0)\,. \tag{5}$$

Therefore, considering the coefficients a, b, c, d as variables of the function H and the population samples $\tilde{x}_i, \tilde{y}_i$ as parameters, we can find an approximation of the unknown coefficients minimizing the *mean squared error* between the values $H(\tilde{x}_i, \tilde{y}_i)$ and $\bar{H}$; more specifically, minimizing the loss function:

$$\mathcal{L}(a, b, c, d) := \frac{1}{N} \sum_{j=0}^{N-1} \left(H(a, b, c, d\ ;\ \tilde{x}_j, \tilde{y}_j) - \bar{H}(a, b, c, d)\right)^2\,. \tag{6}$$

A.2.1 The Gradient of the Loss Function

We remember that the variables of the loss function $\mathcal{L}$ in (6) are the coefficients a, b, c, d and, therefore, for the implementation of an optimization method, the derivatives must be computed with respect to them.

For example, the gradient of $\mathcal{L}$ is such that:

$$\nabla \mathcal{L}(\boldsymbol{k}) = \left[\frac{\partial}{\partial a}\mathcal{L}(\boldsymbol{k})\,,\ \frac{\partial}{\partial b}\mathcal{L}(\boldsymbol{k})\,,\ \frac{\partial}{\partial c}\mathcal{L}(\boldsymbol{k})\,,\ \frac{\partial}{\partial d}\mathcal{L}(\boldsymbol{k})\right]^\top \in \mathbb{R}^4\,, \tag{7}$$

where $\boldsymbol{k} = [\,a, b, c, d\,]^\top$ and

$$\frac{\partial}{\partial k}\mathcal{L}(\boldsymbol{k}) = \frac{1}{N} \sum_{j=0}^{N-1} \left[2\left(H(\boldsymbol{k}; \tilde{x}_j, \tilde{y}_j) - \bar{H}(\boldsymbol{k})\right)\left(\frac{\partial}{\partial k} H(\boldsymbol{k}; \tilde{x}_j, \tilde{y}_j) - \frac{\partial}{\partial k}\bar{H}(\boldsymbol{k})\right)\right]\,, \tag{8}$$

for each $k = a, b, c, d$.

Moreover, observe that:

1. $\frac{\partial}{\partial k}\bar{H}(\boldsymbol{k}) = (1/N)\sum_{i=0}^{N-1} \frac{\partial}{\partial k} H(\boldsymbol{k}; \tilde{x}_i, \tilde{y}_i)$, for each $k = a, b, c, d$;
2. the partial derivatives of $H(\boldsymbol{k}; x, y)$ with respect to a, b, c, d are such that:

$$\frac{\partial}{\partial k} H(\boldsymbol{k}; x, y) = \begin{cases} \log(y)\,, & \text{if } k = a \\ -y\,, & \text{if } k = b \\ \log(x)\,, & \text{if } k = c \\ -x\,, & \text{if } k = d \end{cases}\,.$$