Neural, Parallel, and Scientific Computations, **26**, No. 3 (2018), 311-326 ISSN: 1061-5369

STATE-SPACE SYSTEM IDENTIFICATION PROBLEM FOR SOME NONLINEAR BIOLOGICAL MODELS

ANDREY ANTONOV[1], SVETOSLAV NENOV[2],
AND TZVETELIN TZVETKOV[3]

[1,2,3]Department of Mathematics
University of Chemical Technology and Metallurgy
8, Kliment Ohridski, Blvd., Sofia, 1756, BULGARIA

ABSTRACT: The article is concerned with the problem of parameter estimation in the general case of dynamical systems in the form of state-space. Supposing the existence of first integral(s) of the system, we proved some conditions for solvability of the problem.

As a set of examples, we considered the classical Lotka-Volterra system. We proved that a solution of state-space identification problem is also a solution of a set of nonlinear equations. Based on this result, some applications (using CAS Maple) are presented.

Key Words: state-space system identification, problem of parameter estimation, dynamical systems, first integral, Lotka-Volterra system, numerical examples

Received: August 23, 2018; **Accepted:** November 11, 2018;
Published: November 17, 2018 **doi:** 10.12732/npsc.v26i3.7
Dynamic Publishers, Inc., Acad. Publishers, Ltd. https://acadsol.eu/npsc

1. INTRODUCTION AND STATEMENT

Let $\mathcal{D}$ be a domain in $\mathbb{R}^d$; $\boldsymbol{x}_0 \in \mathcal{D}$; and Ω be a domain in $\mathbb{R}^n$.

Consider the initial value problem

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{\alpha}), \quad \boldsymbol{x} \in \mathcal{D},\ \boldsymbol{\alpha} \in \Omega, \tag{1}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2}$$

where f is uniformly local Lipschitz continuous in $\boldsymbol{x}$, i.e. for any compact set $\mathcal{D}' \subseteq \mathcal{D}$,

there exists a positive constant $L_{\mathcal{D}'} > 0$ such that:

$$\|f(\boldsymbol{x}_1, \boldsymbol{\alpha}) - f(\boldsymbol{x}_2, \boldsymbol{\alpha})\| \leq L_{\mathcal{D}'} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \text{ for all } \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{D}'$$

and f is continuous in $\boldsymbol{\alpha}$. Under these assumptions it is well known that for any $\boldsymbol{x}_0 \in \mathcal{D}$ there exists a unique solution $\boldsymbol{x} = \boldsymbol{x}(t; \boldsymbol{x}_0, \boldsymbol{\alpha})$ of (1) with initial condition (2), see [4]. Below, for simplification of notations, we suppose that all solutions of (1), (2) are defined in time-interval $[0, T)$, $T > 0$. Moreover, we suppose that the domain $\mathcal{D}$ is an invariant set for the system (1).

Let $\Xi = \{\boldsymbol{y}_i : i = 1, \ldots, m\}$ be a given set of points in $\mathcal{D}$.

Now, let us set the problem (state-space system identification, see for example [15], [1], [8]): Find a parameter $\boldsymbol{\alpha}^* \in \Omega$, initial condition $\boldsymbol{x}_0^* \in \mathcal{D}$, and numbers $0 \leq t_1^* \leq \cdots \leq t_m^* \leq T$ such that

$$\min \left\{ \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{y}_i)^2 : \boldsymbol{x}_0 \in \mathcal{D},\ \boldsymbol{a} \in \Omega \right\} = \sum_{i=1}^{m} (\boldsymbol{x}_i^* - \boldsymbol{y}_i)^2 , \tag{3}$$

where

$$\boldsymbol{x}_i = \boldsymbol{x}(t_i; \boldsymbol{x}_0, \boldsymbol{\alpha}), \quad \boldsymbol{x}_i^* = \boldsymbol{x}(t_i^*; \boldsymbol{x}_0^*, \boldsymbol{\alpha}^*), \qquad i = 1, \ldots, m.$$

2. FIRST INTEGRAL AND STATE-SPACE PROBLEMS

Assume that there exists a first integral $\Phi(\boldsymbol{x}, \boldsymbol{\alpha})$ of (1) in $\mathcal{D}$.

Let us set

$$\Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) = \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{y}_i)^2 .$$

If the vector-parameter $\boldsymbol{\alpha}^*$, initial condition $\boldsymbol{x}_0^*$, and numbers $t_1^* \leq \cdots \leq t_m^*$ satisfy the problem (3), then the same vectors and numbers satisfy the following constraint problem

$$\Psi(\boldsymbol{x}(t_1^*; \boldsymbol{x}_0^*, \boldsymbol{\alpha}^*), \ldots, \boldsymbol{x}(t_m^*; \boldsymbol{x}_0^*, \boldsymbol{\alpha}^*)) = \min \Big\{ \Psi(\boldsymbol{x}(t_1; \boldsymbol{x}_0, \boldsymbol{\alpha}), \ldots, \boldsymbol{x}(t_m; \boldsymbol{x}_0, \boldsymbol{\alpha}),) : \\ \boldsymbol{x}_0 \in \mathcal{D},\ 0 \leq t_1 \leq \cdots \leq t_m \leq T,\ \boldsymbol{\alpha} \in \Omega \Big\},$$

subject to additional constrains:

$$\boldsymbol{x}_i = \boldsymbol{x}(t_i; \boldsymbol{x}_0, \boldsymbol{\alpha}), \quad i = 1, \ldots, m.$$

or

$$\Psi(\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_m^*) = \min \{ \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) : \boldsymbol{x}_i \in \mathcal{D},\ i = 0, \ldots, m \},$$

subject to additional constrains:

$$\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha}) = 0, \quad i = 1, \ldots, m.$$

Indeed:

1. If $\boldsymbol{x}_i = \boldsymbol{x}(t_i; \boldsymbol{x}_0, \boldsymbol{\alpha})$ and $\Phi(\boldsymbol{x}, \boldsymbol{\alpha})$ is a first integral of (1) in $\mathcal{D}$, then $\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) = \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha})$ for all $i = 1, \ldots, m$.

2. The domain $\mathcal{D}$ is invariant. Therefore, $\boldsymbol{x}_0 \in \mathcal{D}$ implies $\boldsymbol{x}_i = \boldsymbol{x}(t_i; \boldsymbol{x}_0, \boldsymbol{\alpha}) \in \mathcal{D}$.

Using the method of Lagrange multipliers, let us define

$$L(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{\alpha}, \boldsymbol{\lambda}) = \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) - \sum_{i=1}^{m} \lambda_i \left(\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha})\right),$$

where $\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 & \cdots & \lambda_m \end{pmatrix}^t$, $\boldsymbol{x}_0 \in \mathcal{D}$. Then $\boldsymbol{x}_0^*$, $\boldsymbol{x}_i^*$, $\boldsymbol{\alpha}^*$, and $\boldsymbol{\lambda}$ are solutions of the following system

$$\nabla_{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{\alpha}, \boldsymbol{\lambda}} L(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{\alpha}, \boldsymbol{\lambda}) = 0. \tag{4}$$

i.e.

$$\nabla_{\boldsymbol{x}_0} \left(\sum_{i=1}^{m} \lambda_i \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha}) \right) = \boldsymbol{0}, \tag{5}$$

$$\nabla_{\boldsymbol{x}_i} \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) - \lambda_i \nabla_{\boldsymbol{x}_i} \Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) = \boldsymbol{0}, \ i = 1, \ldots, m, \tag{6}$$

$$\nabla_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{m} \lambda_i \left(\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha})\right) \right) = \boldsymbol{0}, \tag{7}$$

$$\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha}) = 0, \ i = 1, \ldots, m. \tag{8}$$

The system above contains $d + dm + n + m$ equations and the same number of unknowns: $\boldsymbol{x}_0$, $\boldsymbol{x}_i$, $i = 1, \ldots, m$, $\boldsymbol{\alpha}$, and $\boldsymbol{\lambda}$.

Therefore, we have the following result.

Theorem 1. *Let:*

1. *The parameter $\boldsymbol{\alpha}^*$, initial condition $\boldsymbol{x}_0^* \in \mathcal{D}$, and points $\boldsymbol{x}_i^* \in \mathcal{D}$, $i = 1, \ldots, m$ are a solution of state-space system identification problem* (3).

2. *The vectors $\{\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}_i^*, \boldsymbol{\alpha}) : i = 0, \ldots, m\}$ are linearly independent.*

Then there exists a nonzero vector $\boldsymbol{\lambda}$ such that the vectors $\boldsymbol{x}_0^*$, $\boldsymbol{x}_i^*$, $i = 1, \ldots, m$, $\boldsymbol{\alpha}^*$, and $\boldsymbol{\lambda}$ satisfy the following nonlinear system

$$\nabla_{\boldsymbol{x}_0} \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha}) \sum_{i=1}^{m} \lambda_i = \boldsymbol{0}, \tag{9}$$

$$\nabla_{\boldsymbol{x}_i} \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) - \lambda_i \nabla_{\boldsymbol{x}_i} \Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) = \boldsymbol{0}, \ i = 1, \ldots, m, \tag{10}$$

$$\sum_{i=1}^{m} \lambda_i \left(\nabla_{\boldsymbol{\alpha}} \Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \nabla_{\boldsymbol{\alpha}} \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha})\right) = \boldsymbol{0}, \tag{11}$$

$$\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha}) = 0, \ i = 1, \ldots, m. \tag{12}$$

Corollary 1. *Let:*

1. *The parameter* $\boldsymbol{\alpha}^*$*, initial condition* $\boldsymbol{x}_0^* \in \mathcal{D}$*, and points* $\boldsymbol{x}_i^* \in \mathcal{D}$*,* $i = 1, \ldots, m$ *are a solution of state-space system identification problem* (3)*.*

2. *The vectors* $\{\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}_i^*, \boldsymbol{\alpha}) : i = 0, \ldots, m\}$ *are linearly independent, where* $\boldsymbol{x}_i^* = \boldsymbol{x}(t_i^*; \boldsymbol{x}_0^*, \boldsymbol{\alpha}^*)$*,* $i = 1, \ldots, m$*.*

3. $f(\boldsymbol{x}_i^*, \boldsymbol{\alpha}^*) \neq \mathbf{0}$ *for all* $i = 1, \ldots, m$*.*

Then:

1. *The vector* $\boldsymbol{\lambda}$ *is orthogonal to* $(1\ 1\ \cdots\ 1)^t$*.*

2. *The vector* $\boldsymbol{x}_i - \boldsymbol{y}_i$ *is orthogonal to* $f(\boldsymbol{x}_i, \boldsymbol{\alpha}^*)$ *for all* $i = 1, \ldots, m$*.*

Proof. The first claim follows directly from (9) and assumption (2) of the corollary, i.e. $\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha}) \neq \mathbf{0}$.

Obviously

$$\nabla_{\boldsymbol{x}_i} \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) = 2(\boldsymbol{x}_i - \boldsymbol{y}_i), \qquad i = 1, \ldots, m.$$

Hence (see (10))

$$\begin{aligned} \langle \boldsymbol{x}_i - \boldsymbol{y}_i, f(\boldsymbol{x}_i, \boldsymbol{\alpha}^*) \rangle =& \frac{1}{2} \langle \nabla_{\boldsymbol{x}_i} \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m), f(\boldsymbol{x}_i, \boldsymbol{\alpha}^*) \rangle \\ =& \frac{\lambda_i}{2} \langle \nabla_{\boldsymbol{x}_i} \Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}), f(\boldsymbol{x}_i, \boldsymbol{\alpha}^*) \rangle \\ =& 0, \end{aligned}$$

because $\Phi(\boldsymbol{x}, \boldsymbol{\alpha})$ is a first integral of the system (1). □

3. APPLICATION TO BIOLOGICAL MODELS: LOTKA-VOLTERRA SYSTEM

Consider the two dimensional Lotka-Volterra system

$$\dot{x}_1 = x_1(\quad - a_{12} x_2 + b_1), \tag{13}$$

$$\dot{x}_2 = x_2(a_{21} x_1 \quad - b_2), \tag{14}$$

where a_{12}, b_1, a_{21}, and b_2 are strictly positive numbers. Let us start with a very simple example.

t_i	1	5	10	14	16
$\boldsymbol{y}_i$	(30, 4)	(47.2, 50.1)	(70.2, 9.8)	(77.4, 35.2)	(30, 75)

Table 1: Initial data for the predator-prey model

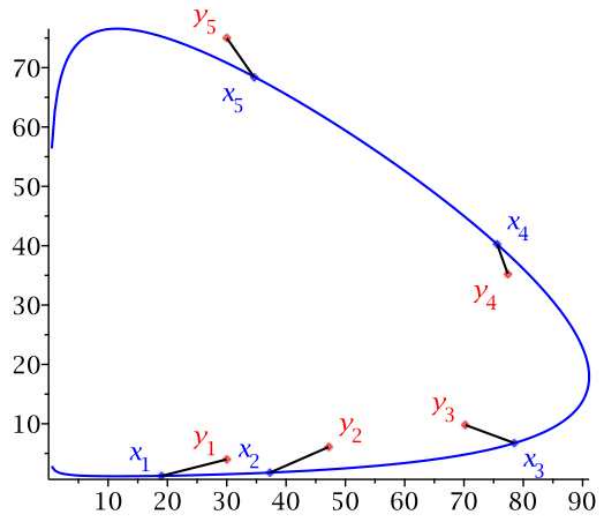

Figure 1: Red points – initial data; blue points – calculated data

Example 1. Let us have the initial data for the predator-prey evolution, given in Table 1.

The problem: Find all parameters a_{12}, b_1, a_{21}, b_2, and initial conditions $\boldsymbol{x}_0$ such that the solution $\boldsymbol{x}(t, \boldsymbol{x}_0)$ that best fits the data in Table 1.

Following [1], the better known approach is to solve system (13), (14) numerically to obtain an approximate solution. A most popular method is to use Fehlberg fourth-fifth order Runge-Kutta method with degree four interpolant. To do these calculations, we use a simple Maple code (see Maple worksheet 1 in Addendum section).

On Figure 1 an arc of obtained orbit is presented. Let us mark that the obtained error is huge: the sum of squared distances between initial data $\boldsymbol{y}_i$ and obtained points $\boldsymbol{x}_i$ is more then 191.97. On Figure 1 the geometric interpretation of residuals are marked as black segments. It is not hard to analyze that the mentioned error above is huge because the time moments t_i are fixed. Also, the orbit obtained is not the optimal solution: there are "black" segments that are not perpendicular to flow, see Corollary 1.

Using Theorem 1 it is not hard to write a Maple code determining the coefficients of the system (13), (14), $\boldsymbol{x}_0$, and $\boldsymbol{x}_i$, $i = 1, \ldots, m$, see Maple worksheet 2 in Addendum section. Here, we have to underline several facts:

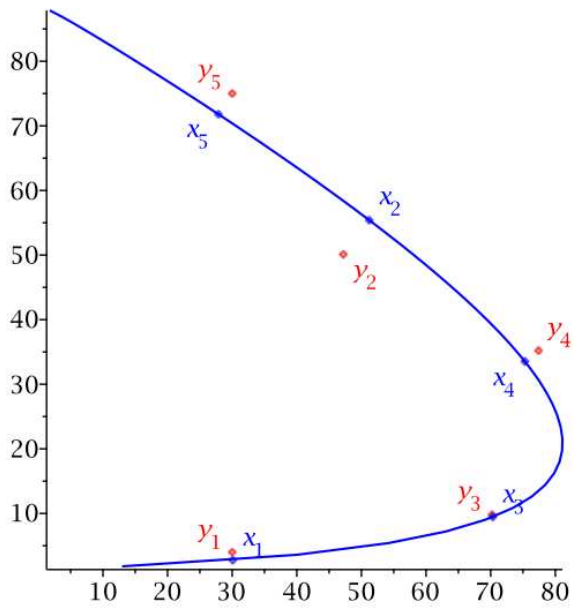

Figure 2: Red points – initial data; blue points – calculated data

1. The main problem is to solve system (9)-(12). This is a nonlinear system and we use the package `GlobalOptimization` to minimize Ψ, under additional conditions (8). The minimization process returns 67.842... as error, i.e. the sum of squared distances betwen $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ is less than 67.843:

$$\sum_{i=1}^{m=5} (\boldsymbol{x}_i - \boldsymbol{y}_i)^2 < 67.843.$$

2. The points $\boldsymbol{x}_i$, $i = 1, \ldots, m$ are obtained numerically, i.e. we cannot expect all points to lie in one and the same orbit of the system (13), (14). Our calculations: $|\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_0)| \leq 9 \times 10^{-11}$, where $\Phi(\boldsymbol{x})$ is the calculated first integral of (13), (14):

$$\begin{aligned}\Phi(\boldsymbol{x}) =& a_{21}x_1 - b_2 \ln(x_1) + a_{12}x_2 - b_1 \ln(x_2)\\ \approx& 0.228x_1 - 0.1 \ln(x_1) + 0.479x_2 - 10 \ln(x_2).\end{aligned}$$

An arc of obtained orbit is plotted on Figure 2.

It is not difficult to simplify Theorem 1 in the case of Lotka-Volterra two dimensional system (13), (14).

Corollary 2. *Let:*

1. *The parameters* a_{12}, a_{21}, b_1, b_2, *initial condition* $\boldsymbol{x}_0 = (x_{10}\ x_{20})^t \in \mathcal{D}$, *and points* $\boldsymbol{x}_i = (x_{1i}\ x_{2i})^t \in \mathcal{D}$, $i = 1, \ldots, m$ *are a solution of state-space system identification problem* (3) *for the Lotka-Volterra system* (13), (14).

2. *The vectors* $\{\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}_i^*, \boldsymbol{\alpha}) : i = 0, \ldots, m\}$ *are linearly independent,* $\boldsymbol{\alpha} = (a_{12}\ b_1\ a_{21}\ b_2)^t$.

Then

$$0 < x_{10} < \frac{b_2}{a_{21}}, \quad 0 < x_{20} < \frac{b_1}{a_{12}}, \tag{15}$$

$$\sum_{i=1}^{m} \lambda_i = 0, \tag{16}$$

$$2x_{1i}^2 + (a_{21}\lambda_i - 2y_{1i})\, x_{1i} - b_2\lambda_i = 0,\ i = 1, \ldots, m, \tag{17}$$

$$2y_{1i}^2 + (a_{12}\lambda_i - 2y_{2i})\, y_{1i} - b_1\lambda_i = 0,\ i = 1, \ldots, m, \tag{18}$$

$$\sum_{i=1}^{m} \lambda_i x_{2i} = \sum_{i=1}^{m} \lambda_i x_{i1} = \sum_{i=1}^{m} \lambda_i \ln(x_{2i}) = \sum_{i=1}^{m} \lambda_i \ln(x_{i1}) = 0, \tag{19}$$

$$b_2\,(\ln(x_{10}) - \ln(x_{1i})) + b_1\,(\ln(x_{20}) - \ln(x_{2i})) + a_{12}(x_{21} - x_{20}) - a_{21}(x_{10} - x_{1i}) = 0,\ i = 1, \ldots, m. \tag{20}$$

Proof. Obviously ($\boldsymbol{x} = (x_1\ x_2)^t$, $\boldsymbol{\alpha} = (a_{12}\ b_1\ a_{21}\ b_2)^t$)

$$\Phi(\boldsymbol{x}, \boldsymbol{\alpha}) = a_{21}x_1 - b_2 \ln(x_1) + a_{12}x_2 - b_1 \ln(x_2)$$

is the first integral of (13), (14).

Then:

$$\frac{\partial \Phi(\boldsymbol{x}, \boldsymbol{\alpha})}{\partial x_1} = a_{21} - \frac{b_2}{x_1}, \quad \frac{\partial \Phi(\boldsymbol{x}, \boldsymbol{\alpha})}{\partial x_2} = a_{12} - \frac{b_1}{x_2},$$

$$\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}, \boldsymbol{\alpha}) = \begin{pmatrix} a_{21} - \dfrac{b_2}{x_1} \\ a_{12} - \dfrac{b_1}{x_2} \end{pmatrix} \neq \boldsymbol{0}.$$

Therefore (9) yields (16).

Consider the equation (10). We have:

$$\nabla_{\boldsymbol{x}_i} \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) = \begin{pmatrix} \boldsymbol{x}_1 - \boldsymbol{y}_1 \\ \boldsymbol{x}_2 - \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{x}_m - \boldsymbol{y}_m \end{pmatrix}.$$

Therefore, we receive the following equations

$$2x_{1i}^2 + (a_{21}\lambda_i - 2y_{1i})\, x_{1i} - b_2\lambda_i = 0$$

$$2y_{1i}^2 + (a_{12}\lambda_i - 2y_{2i})\, y_{1i} - b_1\lambda_i = 0.$$

Similarly, using (11) and (12), we receive

$$\sum_{i=1}^{m} \lambda_i x_{2i} = \sum_{i=1}^{m} \lambda_i x_{i1} = 0$$

$$\sum_{i=1}^{m} \lambda_i \ln(x_{2i}) = \sum_{i=1}^{m} \lambda_i \ln(x_{i1}) = 0$$

and

$$b_2\,(\ln(x_{10}) - \ln(x_{1i})) + b_1\,(\ln(x_{20}) - \ln(x_{2i})) + a_{12}(x_{21} - x_{20}) - a_{21}(x_{10} - x_{1i}) = 0,$$

respectively. □

Remark 1. Assume, we know the parameters of the Lotka-Volterra system. Moreover, let us set $C = \Phi(\boldsymbol{x}_0, \boldsymbol{\alpha})$. Then, we have to minimize $\Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$, $\boldsymbol{x}_i \in \mathcal{D}$, $i = 0, \ldots, m$, subject to

$$\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - C = 0, \quad i = 1, \ldots, m.$$

Again, let

$$L(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{\lambda}, C) = \Psi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) - \sum_{i=1}^{m} \lambda_i\,(\Phi(\boldsymbol{x}_i, \boldsymbol{\alpha}) - C))$$

be the Lagrangian.

Obviously (varying C, the optimal value of L will vary, too):

$$\frac{\partial L(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{\lambda}, C)}{\partial C} = \sum_{i=1}^{m} \lambda_i = 0,$$

see (16).

Hence, at the extrema points, the rate of change of the quantity being optimized, with respect to the constraint value C is very slow. Hence, the computation time is huge.

Example 2. An real example of Lotka-Volterra models is seen in Canadian northern forests: the populations of the lynx and the snowshoe hare, see [5], [18].

The initial data is present in Table 2.

Using the Maple code (see Maple worksheet 3 in Addendum section), analogous to that one in previous example, we receive the parameters of the system and initial data:

$$a_{12} \approx 0.002712, \quad a_{21} \approx 0.002978, \quad b_1 \approx 0.067756, \quad b_2 \approx 0.105185,$$

year	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911
hare (× 1000)	30	47.2	70.2	77.4	36.3	20.6	18.1	21.4	22	25.4	27.1	40.3
lynx (× 1000)	4	6.1	9.8	35.2	59.4	41.7	19	13	8.3	9.1	7.4	8

year	1912	1913	1914	1915	1916	1917	1918	1919	1920
hare (× 1000)	57	76.6	52.3	19.5	11.2	7.6	14.6	16.2	24.7
lynx (× 1000)	12.3	19.5	45.7	51.1	29.7	15.8	9.7	10.1	8.6

Table 2: The hare-lynx dynamics, see [5]

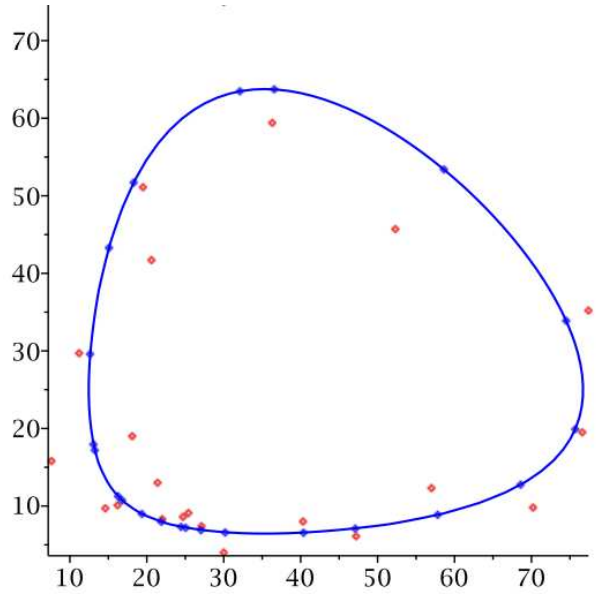

Figure 3: Hare-lynx populations: red points – initial data; blue points – calculated data

$$\boldsymbol{x}_0 \approx (46.792697, 60.941209)^t .$$

The sum of squares error is less than 428.8905, see Figure 3.

Let us note that in [5], the sum of squares error is 593.28 (computed by a MatLab code). The difference in the results is based on the following:

1. In [5] is minimized the sum of squares with respect to the conditions $\boldsymbol{x}_i = \boldsymbol{x}_i(t_i, \boldsymbol{x}_0)$, where t_i are the years, $i = 1, \ldots, m = 21$, see Table 2. Hence, the unknown parameters are the coefficients of the system and initial conditions.

2. In our results, the unknown parameters are the coefficients of the system, the initial conditions $\boldsymbol{x}_0$, and the points $\boldsymbol{x}_i$ subject to additional constrains $\Phi(\boldsymbol{x}_i) =$

$\Phi(\boldsymbol{x}_0)$, $i = 1, \ldots, m = 21$.

ADDENDUM: SOME USED MAPLE WORKSHEETS

MAPLE WORKSHEET 1

```
> restart:  with(plots):  with(plots, implicitplot):
with(plottools):  with(GlobalOptimization):
> T := [1, 5, 10, 14, 16]:
y1 := [30, 47.2, 70.2, 77.4, 30]:
y2 := [4, 6.1, 9.8, 35.2, 75]:
m := nops(y1);
data := [seq([T[i], y1[i], y2[i]], i = 1 ..  m)];
d := 2:
X := Matrix(m, 2, (i, j) -> x[i, j]):
local Psi := (X) -> add((X[i, 1]-y1[i])^2+(X[i, 2]-y2[i])^2,
  i = 1 ..  m):
LotkaVolterraModel := [
  diff(x1(t), t) = x1(t)*(-a12*x2(t)+b1),
  diff(x2(t), t) = x2(t)*(a21*x1(t)-b2)];
InitialConditions := [x1(0) = ic1, x2(0) = ic2];
RungeKuttaFehlbergMethod := dsolve(
  InitialConditions[], LotkaVolterraModel[],
  numeric, parameters = [ic1, ic2, a12, b1, a21, b2],
  method = rkf45, abserr = 1^(-100), relerr = 1^(-100));
objective_function := proc (ic1, ic2, a12, b1, a21, b2)
  RungeKuttaFehlbergMethod(
  parameters = [ic1, ic2, a12, b1, a21, b2]);
  add((rhs(select(has,
  RungeKuttaFehlbergMethod(data[i, 1]), x1)[])-data[i, 2])^2
  +(rhs(select(has,
  RungeKuttaFehlbergMethod(data[i, 1]), x2)[])-data[i, 3])^2,
  i = 1 ..  m)
end proc;
results := GlobalSolve(
  'objective_function'(ic1, ic2, a12, b1, a21, b2),
  ic1 = 0 ..  100, ic2 = 0 ..  100, a12 = 0 ..  1,
```

```
  a21 = 0 ..  1, b1 = 0 ..  1, b2 = 0 ..  1)
```

$$m = 5$$

$$\text{data} := [[1, 30, 4], [5, 47.2, 6.1], [10, 70.2, 9.8], [14, 77.4, 35.2], [16, 30, 75]]$$

```
Warning, relerr reduced from 1.  to 1e-3
```

$$\text{RungeKuttaFehlbergMethod:=}\textbf{proc}(\text{x\_rkf45})\ \ldots\ \textbf{end proc}$$

$$\text{objective\_function} := \textbf{proc}(\text{ic1, ic2, a12, b1, a21, b2})\ \ldots\ \textbf{end proc}$$

$$\begin{aligned}\text{results} := [&418.760829426850592, [\text{a12} = 0.0102178094044135,\\ &\text{a21} = 0.00597528126417812, \text{b1} = 0.183251112997071,\\ &\text{b2} = 0.0688611041499360, \text{ic1} = 15.9981960988228,\\ &\text{ic2} = 1.16979062354762]]\end{aligned}$$

MAPLE WORKSHEET 2

```
> restart:  with(plots):  with(plots, implicitplot):
with(plottools):  with(GlobalOptimization):
> Y1 := [30, 47.2, 70.2, 77.4, 30]:
Y2 := [4, 50.1, 9.8, 35.2, 75]:
m := nops(Y1); n := 4:  d := 2:  X := Vector(m, i -> x[i]):
Y := Vector(m, i -> y[i]):
```

$$m = 5$$

```
> local Phi := proc (x, y, a12, b1, a21, b2)
  -> a21*x-b2*ln(x)+a12*y-b1*ln(y);
```

$$\Phi := (x, y, a12, b1, a21, b2) -> a21 * x - b2 * ln(x) + a12 * y - b1 * ln(y)$$

```
> infolevel[GlobalOptimization] := 3:
results := GlobalSolve(Psi(X, Y),
  seq(Phi(x[i], y[i], a12, b1, a21, b2)
  = Phi(x[0], y[0], a12, b1, a21, b2), i = 1 ..  m),
  initialpoint = [a12 = .479106761299915, a21 = .228154646344869,
  b1 = 10., b2 = .100000000000000, lambda[1] = 1.20760156744517,
  lambda[2] = 4.44124935584076, lambda[3] = 3.86066634005239,
  lambda[4] = 6.54432037106769, lambda[5] = 9.70161776085386,
```

```
  x[0] = 19.2347258743833, x[1] = 30.0885752977728,
  x[2] = 51.2113096071932, x[3] = 70.3347634103879,
  x[4] = 75.2958407016812, x[5] = 27.8789375298377,
  y[0] = 77.3996516178174, y[1] = 2.76279022791571,
  y[2] = 55.3948864749378, y[3] = 9.45620990954325,
  y[4] = 33.5227914656997, y[5] = 71.7904399352332],
  x[0] = 0 ..  100, y[0] = 0 ..  100, a12 = .1 ..  3,
  a21 = .1 ..  1, b1 = 6 ..  10, b2 = .1 ..  1,
  seq(x[i] = 0 ..  100, i = 1 ..  m),
  seq(y[i] = 0 ..  100, i = 1 ..  m));
```

GlobalSolve: calling NLP solver
GlobalSolve: calling global optimization solver
GlobalSolve: number of problem variables 16
GlobalSolve: number of nonlinear inequality constraints 0
GlobalSolve: number of nonlinear equality constraints 5
GlobalSolve: method OptimusDEVOL
GlobalSolve: maximum iterations 525
GlobalSolve: population size 160
GlobalSolve: average stopping stepwidth 0.1e-3
GlobalSolve: time limit 2100
GlobalSolve: trying evalhf mode
GlobalSolve: performing local refinement

results := [67.8420105856509252, [a12 = 0.479106843024773, a21 = 0.228154817908553, b1 = 10., b2 = 0.100000000000000, x[0] = 21.9495670638483, x[1] = 30.0885752761301, x[2] = 51.2113049574526, x[3] = 70.3347652327639, x[4] = 75.2958333817766, x[5] = 27.8789384186806, y[0] = 2.24546752284065, y[1] = 2.76278808245235, y[2] = 55.3948860040821, y[3] = 9.45621186695482, y[4] = 33.5227810865563, y[5] = 71.7904436757884]]

```
> F := (x, y) -> Phi(x, y, rhs(select(has, results[2], a12)[]),
  rhs(select(has, results[2], b1)[]),
  rhs(select(has, results[2], a21)[]),
  rhs(select(has, results[2], b2)[])):
evalf[3](seq(abs(F(rhs(select(has, results[2], x[i])[]),
  rhs(select(has, results[2], y[i])[]))
  -F(rhs(select(has, results[2], x[0])[]),
  rhs(select(has, results[2], y[0])[]))),
  i = 0 ..  m));
```

0., 8.20*10^(-11), 5.14*10^(-12), 4.79*10^(-11), 1.92*10^(-11), 4.31*10^(-12)

MAPLE WORKSHEET 3

```
> restart:  with(plots):  with(plots, implicitplot):
with(plottools):  with(GlobalOptimization):
hare := [30, 47.2, 70.2, 77.4, 30, 36.3, 20.6, 18.1, 21.4, 22,
  25.4, 27.1, 40.3, 57, 76.6, 52.3, 19.5, 11.2, 7.6, 14.6,
  16.2, 24.7]:
lynx := [4, 6.1, 9.8, 35.2, 75, 59.4, 41.7, 19, 13, 8.3, 9.1,
  7.4, 8, 12.3, 19.5, 45.7, 51.1, 29.7, 15.8, 9.7, 10.1, 8.6]:
m := nops(hare):  n := 4:  d := 2:
X := Vector(m, i -> x[i]):  Y := Vector(m, i -> y[i]):

local Phi := (x, y, a12, b1, a21, b2)
  -> a21*x-b2*ln(x)+a12*y-b1*ln(y);
local Psi := (X, Y)
  -> add((x[i]-hare[i])^2+(y[i]-lynx[i])^2, i = 1 ..  m);
```

$$\Phi := (x, y, a12, b1, a21, b2) -> a21x - b2ln(x) + a12y - b1ln(y)$$

$$\Psi := (X, Y) -> add((x[i] - hare[i])^2 + (y[i] - lynx[i])^2, i = 1..m)$$

```
> infolevel[GlobalOptimization] := 3;
results := GlobalSolve(Psi(X, Y),
  seq(Phi(x[i], y[i], a12, b1, a21, b2)
  = Phi(x[0], y[0], a12, b1, a21, b2), i = 1 ..  m),
  x[0] = 0 ..  100, y[0] = 0 ..  100,
  a12 = 0.1e-4 ..  1, a21 = 0.1e-2 ..  1,
  b1 = 0.1e-2 ..  1, b2 = 0.1e-2 ..  1,
  seq(x[i] = 0 ..  100, i = 1 ..  m),
  seq(y[i] = 0 ..  100, i = 1 ..  m),
  evaluationlimit = 10^6,
  avgstopstepwidth = 10^(-5),
  populationsize = 1000)
GlobalSolve: calling NLP solver
GlobalSolve: calling global optimization solver
GlobalSolve: number of problem variables 50
GlobalSolve: number of nonlinear inequality constraints 0
GlobalSolve: number of nonlinear equality constraints 22
GlobalSolve: method OptimusDEVOL
```

GlobalSolve: maximum iterations 1000
GlobalSolve: population size 1000
GlobalSolve: average stopping stepwidth 1/100000
GlobalSolve: time limit 7200
GlobalSolve: trying evalhf mode
GlobalSolve: performing local refinement

results := [428.890440602430431, [a12 = 0.00271172103096504, a21 = 0.00297772479740680, b1 = 0.0677555686059305, b2 = 0.105184817849744, x[0] = 55.1951201839620, x[1] = 30.1737388133019, x[2] = 47.0933197686964, x[3] = 68.5725772068865, x[4] = 74.5090710505102, x[5] = 32.0978961007733, x[6] = 36.5648063452672, x[7] = 15.0735698853647, x[8] = 13.0664336382384, x[9] = 19.3568046664196, x[10] = 21.8926441290475, x[11] = 25.0530684079186, x[12] = 27.0344756440903, x[13] = 40.3705511523999, x[14] = 57.8073503494856, x[15] = 75.6880380276483, x[16] = 58.6109942360203, x[17] = 18.2999106147010, x[18] = 12.6231535832025, x[19] = 13.2374388060624, x[20] = 16.2487672837392, x[21] = 16.7812037633146, x[22] = 24.4393864586380, y[0] = 8.29669326200963, y[1] = 6.58879482110436, y[2] = 7.08262806986796, y[3] = 12.7387412041133, y[4] = 33.8853412185394, y[5] = 63.4739545157908, y[6] = 63.7192900708077, y[7] = 43.2836506658494, y[8] = 17.9438409227003, y[9] = 8.97788818949608, y[10] = 7.95918334814176, y[11] = 7.19412932173964, y[12] = 6.88939758819259, y[13] = 6.55470755243952, y[14] = 8.85823678085712, y[15] = 19.8981814793112, y[16] = 53.3963444048431, y[17] = 51.7071334732456, y[18] = 29.5879101307840, y[19] = 17.1943495495269, y[20] = 11.2592924682180, y[21] = 10.7358847899826, y[22] = 7.31196018591283]]

```
> F := (x, y) -> Phi(x, y,
  rhs(select(has, results[2], a12)[]),
  rhs(select(has, results[2], b1)[]),
  rhs(select(has, results[2], a21)[]),
  rhs(select(has, results[2], b2)[])):
F(x,y);
# First Integrals at calculated points:
evalf[3](seq(abs(F(rhs(select(has, results[2], x[i])[]),
  rhs(select(has, results[2], y[i])[]))
  -F(rhs(select(has, results[2], x[0])[]),
  rhs(select(has, results[2], y[0])[]))), i = 0 ..  m)):
```

0.297772479740680e-2*x-.105184817849744*ln(x)+0.271172103096504e-2*y-0.677555686059305e-1*ln(y)

0., 6.01*10^(-13), 9.40*10^(-13), 2.09*10^(-12), 3.34*10^(-12), 2.34*10^(-12), 2.05*10^(-12), 5.84*10^(-12), 6.33*10^(-12), 2.57*10^(-12), 2.45*10^(-12), 1.56*10^(-12), 2.05*10^(-12), 9.36*10^(-13), 1.45*10^(-12), 3.26*10^(-12), 2.26*10^(-12), 4.73*10^(-12), 6.39*10^(-12), 6.30*10^(-12), 4.80*10^(-12), 7.45*10^(-12), 1.62*10^(-12)

ACKNOWLEDGMENTS

We used the computational resources and software provided by the Mathematical Models and Computer Simulations Center at University of Chemical Technology and Metallurgy.

REFERENCES

[1] Daniel Brewer, Martino Barenco, Robin Callard, Michael Hubank, Jaroslav Stark, Fitting ordinary differential equations to short time course data, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* (2008), doi.org/10.1098/rsta.2007.2108.

[2] E.B. Cox, M.A. Woodbury, L.E. Myers, A new model for tumor growth analysis based on a postulated inhibitory substance, *Comp. Biomed. Res.*, **13** (1980), 437.

[3] Angel Dishliev, Katya Dishlieva, Svetoslav Nenov, *Specific Asymptotic Properties of the Solutions of Impulsive Differential Equations. Methods and Applications*, Academic Publications, 2012, available at http://www.acadpubl.eu/ap/node/3.

[4] P. Hartman, Ordinary differential equations, New York, Wiley, 1964.

[5] Joseph M. Mahaffy, *Lotka-Volterra Models* (2009), https://jmahaffy.sdsu.edu/courses/f09/math636/lectures/lotka/qualde2.html

[6] Miljenko Marušić, *Mathematical Models of Tumor Growth*, Lecture presented at the Mathematical Colloquium in Osijek organized by Croatian Mathematical Society, Division Osijek, June 7, 1996.

[7] M. Marušić, Z. Bajzer, J.P. Freyer, S. Vuk-Pavlovic, Analysis of growth of multicellular tumour spheroids by mathematical models, *Cell Prolif.*, **27** (1994), 7394.

[8] T. Mckelvey, Fully parametrized statespace models in system identification, *IFAC Proceedings Volumes*, **27**, No. 8 (1994), doi: 10.1016/S1474-6670(17)47786-X.

[9] S. Nenov, Impulsive controllability and optimization problems. Lagrange's method and applications, *ZAA – Zeitschrift für Analysis und ihre Anwendungen*, Heldermann Verlag, Berlin, **17**, No. 2 (1998), 501-512.

[10] S. Piantadosi, A model of growth with first-order birth and death rates, *Comp. Biomed. Res.*, **18** (1985), 220-232.

[11] Janos D. Pinter, *Global Optimization with Maple*, Pintér Consulting Services Inc., Maplesoft (2006), ISBN: 1-897310-15-3.

[12] Janos D. Pinter, *Global Optimization: Scientific and Engineering Case Studies*, Springer (2006), ISBN: 0387304088.

[13] Janos D. Pinter, *Global Optimization in Action: Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*, Springer (1995), ISBN: 0792337573.

[14] Katarzyna A. Rejniak, A single cell approach in modeling the dynamics of tumor microregions, *Mathematical Biosciences and Engineering*, http://math.asu.edu/˜ mbe/

[15] Thomas B. Schön, Adrian Willsb, Brett Ninness, System identification of nonlinear state-space models, *Automatica*, **47**, No. 1 (2011), doi: 10.1016/j.automatica.2010.10.013.

[16] A. Tsoularis, Analysis of logistic growth models, *Res. Lett. Inf. Math. Sci.*, **2** (2001), 23-46.

[17] T.E. Wheldon, J. Kirk, W.M. Grey, Mitotic autoregulation, growth control and neoplasia, *J. Theor. Biol.*, **38** (1973), 627.

[18] Uri Wilensky, Kenneth Reisman, Thinking like a wolf, a sheep or a firefly: Learning biology through constructing and testing computational theories – an embodied modeling approach, Center for Connected Learning and Computer-Based Modeling, Northwestern University (1992), http://ccl.northwestern.edu/papers/bio/long/