Applied Mathematics and Computation 390 (2021) 125644

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A Modified Multiple Shooting Algorithm for Parameter Estimation in ODEs Using Adjoint Sensitivity Analysis

Ozgur Aydogmus[a,*], Ali Hakan TOR[b]

[a] Social Sciences University of Ankara, Department of Economics Ankara, Turkey
[b] Abdullah Gül University, Department of Applied Mathematics Kayseri, Turkey

ARTICLE INFO

Article history:
Received 9 March 2020
Revised 28 June 2020
Accepted 23 August 2020
Available online 7 September 2020

Keywords:
Parameter estimation
Multiple shooting algorithm
Adjoint method

ABSTRACT

To increase the predictive power of a model, one needs to estimate its unknown parameters. Almost all parameter estimation techniques in ordinary differential equation models suffer from either a small convergence region or enormous computational cost. The method of multiple shooting, on the other hand, takes its place in between these two extremes. The computational cost of the algorithm is mostly due to the calculation of directional derivatives of objective and constraint functions. Here we modify the multiple shooting algorithm to use the adjoint method in calculating these derivatives. In the literature, this method is known to be a more stable and computationally efficient way of computing gradients of scalar functions. A predator-prey system is used to show the performance of the method and supply all necessary information for a successful and efficient implementation.

1. Introduction

Modeling by ordinary differential equations (ODEs) is used to accurately depict the physical state of a system in many areas of applied sciences and engineering. Accurately describing the system and finding its parameters allow for future behavior to be predicted. To estimate parameters in ODEs, we use partially observed noisy data. Fitting (partially) observed noisy data is a useful way for estimating the unknown parameters of a system of ODEs. Implementing such a procedure requires using a convenient ODE solver and an optimization routine.

There are several stochastic and deterministic optimization routines. A detailed discussion of stochastic methods for parameter estimation in ODEs is given in [1]. Deterministic optimization procedures such as sequential quadratic programming (SQP), Newton methods or quasi-Newton methods can also be employed to estimate the unknown parameters of an ODE system. Contrary to stochastic algorithms, deterministic ones are computationally efficient but tend to converge to local minima.

For parameter estimation problems, the convergence to local minima is prevalent when the single shooting method (also known as initial value approach) is employed. This parameter estimation method uses a single initial condition to produce a trajectory that attempts to fit the noisy data points by minimizing a maximum-likelihood functional with respect to parameters. Hence, the method is computationally efficient but its convergence is highly sensitive to the initial guesses.

[*] Corresponding author.
E-mail addresses: ozgur.aydogmus@asbu.edu.tr (O. Aydogmus), hakantor@gmail.com (A.H. TOR).

https://doi.org/10.1016/j.amc.2020.125644
0096-3003/© 2020 Elsevier Inc. All rights reserved.

The above discussion suggests that there is a trade-off between computational cost and stability of the algorithm. Multiple shooting approach for estimating parameters takes its place in between these two extremes. The method was introduced in [2] in 1970s and was substantially enhanced and mathematically analysed in [3,4]. In this approach, one allows multiple initial values and the problem is considered as a multi-point boundary value problem. This is to say that the parameter space is enlarged and discontinuous trajectories are allowed during the optimization process. Employing this method increases the convergence significantly; hence the above-mentioned computationally efficient deterministic optimization routines can be used to estimate the parameters. We provide the details of this algorithm in the following section.

We would like to note that the multiple shooting method is effective when a gradient based nonlinear programming (NLP) solver such as SQP [5,6], quasi-Newton [7] or generalized Gauss-Newton methods [6,8] is employed. The reason behind the choice of gradient based NLP solvers is that one can compute the sensitivity equations which is useful in finding the gradient of the objective and/or constraint functions. With that being said, the computational cost of shooting methods is strongly dependent on the efficiency of ODE and sensitivity solvers [9].

It was claimed in [10] that there are three feasible approaches for calculating the derivatives of trajectories with respect to the parameters. These methods are external differentiation, internal differentiation and the simultaneous solutions of the sensitivity equations. The latter approach is claimed to be the most effective approach among these three. This method requires the simultaneous solution of the original ODE system with the sensitivity systems obtained by differentiating the original system with respect to each parameter. If the number of parameters is relatively small this approach may be efficient. On the other hand, the forward sensitivity approach is intractable when the number of state variables and the number of parameters are large. In [11,12], it was claimed that the sensitivities with respect to parameters can be computed more efficiently by employing the adjoint method if the number of parameters is large. Moreover, it was discussed in [13] that the adjoint method is the most suitable method to compute sensitivities of a function provided that this function is scalar.

The adjoint method is used in [14] to calculate sensitivities for estimating the parameters via single shooting algorithm. In addition, the adjoint and multiple shooting strategies for optimal control of differential algebraic equations systems are combined in [15]. To the best of our knowledge, the adjoint method has not been employed to implement a multiple shooting algorithm for estimating the parameters of a system of ODEs. In this study, we aim to modify the classical multiple shooting algorithm so that the adjoint method can be applied efficiently.

The structure of this article is as follows: In the following section, we give a detailed description of the multiple shooting parameter estimation method for systems of ODEs. In Section 3, we modify the multiple shooting algorithm and find the sensitivities with respect to parameters using the adjoint method. In Section 4, we consider a Lotka-Volterra system and implement the modified algorithm to estimate its parameters from disturbed data. Lastly, we discuss and summarize our findings in Section 5.

2. The Classical Multiple Shooting Method for Parameter Estimation

In this section, we recall the classical multiple shooting method whose detailed mathematical analysis was performed in [3,4]. The method is used to estimate the parameters of the system and some applications of this method to measured data are given in [16–19].

We start by considering a $d-$dimensional state variable $\mathbf{x}(\mathbf{t}) \in \mathbb{R}^{\mathbf{d}}$ at time $t \in I = [t_0, t_f]$ of a continuous time ordinary differential equation (ODE) satisfying the following initial value problem:

$$\dot{\mathbf{x}}(t) = f\Big(\mathbf{x}(t), t, \mathbf{p}\Big), \qquad \mathbf{x}(t_0) = \mathbf{x_0}. \tag{1}$$

We would like to note that the right hand side of the former equality depends on the parameter vector $\mathbf{p} \in \mathbb{R}^m$. The multiple shooting method requires using an enlarged parameter space. This ensures that the procedure has more flexibility for searching the parameter space and circumventing local minima. This enlargement is realized by subdividing the time interval with multiple shooting nodes as follows:

$$t_0 = \tau_0 < \tau_1 < \cdots < \tau_K = t_f.$$

By introducing the discrete trajectory $\{\mathbf{s}_j := \mathbf{x}(\tau_j) | j = 0, 1, \cdots, K\}$, one can define the extended parameter vector as: $\mathbf{q} = (\mathbf{s}_0, \mathbf{s}_1, \cdots, \mathbf{s}_K, \mathbf{p})$. In each subinterval $I_j = [\tau_j, \tau_{j+1}]$ for $j = 0, 1 \cdots, K-1$, we consider K independent initial value problems

$$\dot{\mathbf{x}}_{\mathbf{j}}(t) = f\Big(\mathbf{x}_j(t), t, \mathbf{p}\Big), \quad \mathbf{x}_j(\tau_j) = \mathbf{s}_j, \qquad t \in [\tau_j, \tau_{j+1}] \tag{2}$$

where $\mathbf{x}_j(t) = \mathbf{x}_j(t; \tau_j, \mathbf{s}_j, \mathbf{p}) \in \mathbb{R}^d$.

In [7], it was required that each subinterval I_j contains at least one measurement. We require that the following assumption holds throughout the paper.

Assumption 1. Observable variables are indexed by $i = 1, 2, \cdots, obs$ and measurements are collected at times τ_j, $j \in M \subset \{0, 1, 2, \cdots, K\}$.

Here we remark that the number of observable variables are smaller than or equal to d. We also would like to note that Assumption 1 guarantees that all measurement points are also multiple shooting nodes. As was noted in [20], these two sets can be taken equal i.e. $M = \{0, 1, 2, \cdots, K\}$. This assumption will be used in the next section to simplify the objective function in such a way that it reduces the computational complexity of the algorithm. For a detailed discussion, see Remark 2.

We now introduce the classical multiple shooting algorithm in the following lines following [6,7]. Let the measurements for a general function of the state variables $\mathbf{x}(t)$ be given as follows:

$$\eta_{ij} = g_{ij}\big(\mathbf{x}(\tau_j), \mathbf{q}\big) + \varepsilon_{ij} \tag{3}$$

where measurement errors ε_{ij} are independent, Gaussian with zero mean and variances σ_{ij}^2 for $i \in \{1, 2, \cdots, obs\}$ and $j \in M$. Hence $[\eta_{ij}]$ is a $obs \times |M|$ matrix where $|.|$ is used to denote the cardinality of a set. Here the least squares objective function for discontinuous trajectories is defined as follows:

$$L(\mathbf{q}) := \sum_{i=1}^{obs} \sum_{j \in M} \big(l_{ij}(\mathbf{q})\big)^2$$

where $l_{ij}(\mathbf{q}) = \big(\eta_{ij} - g_{ij}\big(\mathbf{x}_j(\tau_j), \mathbf{q}\big)\big)/\sigma_{ij}$. Then we need to consider the following constrained nonlinear optimization problem:

$$\begin{aligned} &\underset{\mathbf{q}}{\text{minimize}} && L(\mathbf{q}) \\ &\text{subject to} && G_j\big(\mathbf{x}_j, \mathbf{q}\big) = 0 \quad \text{for } j = 0, 1, \cdots, K-1. \end{aligned} \tag{4}$$

where the continuity constrains are given by $G_j\big(\mathbf{x}_j, \mathbf{q}\big) = \mathbf{x}_j(\tau_{j+1}; \tau_j, \mathbf{s}_j, \mathbf{p}) - \mathbf{s}_{j+1} \in \mathbb{R}^d$. Note that each g_j is a $1 \times d$ vector and does not specify a scalar valued function. We also note that some optional equality or inequality constraints may be added to this optimization problem (4) (see e.g. [7]).

Non-linear optimization problem (4) can be solved iteratively using the generalized-quasi-Newton method [21, pp.24-25]. An update step $\Delta\mathbf{q} = (\Delta\mathbf{s}_0, \Delta\mathbf{s}_1, \cdots, \Delta\mathbf{s}_K, \Delta\mathbf{p})$ can be calculated by solving the following Linearized Constrained Least Squares Problem:

$$\begin{aligned} &\underset{\mathbf{q}}{\text{minimize}} && \big(l_{ij}(\mathbf{q}^0) + J_{\mathbf{q}} l_{ij}(\mathbf{q}^0)\Delta\mathbf{q}\big)^2 \\ &\text{subject to} && G_j\big(\mathbf{x}_j, \mathbf{q}^0\big) + J_{\mathbf{q}} G_j\big(\mathbf{x}_j, \mathbf{q}^0\big)\Delta\mathbf{q} = 0 \quad \text{for } j = 0, 1, \cdots, K-1. \end{aligned} \tag{5}$$

for some initial guess $\mathbf{q}^0 = \big(\mathbf{s}_0^0, \mathbf{s}_1^0, \cdots, \mathbf{s}_K^0, \mathbf{p}^0\big)$. Here J_q denotes the Jacobian with respect to the parameters $\mathbf{q}$ of the corresponding function. Hence $\Delta\mathbf{q}$ is used to update parameters as follows: $\mathbf{q}^l = \mathbf{q}^{l-1} + \Delta\mathbf{q}$. We would like to note that quasi-Newton algorithms has been used to implement multiple shooting procedures. More detailed description of this iterative process is given in [8, pp.12-17].

Note that one needs to calculate jacobians $J_{\mathbf{q}} G_j(\mathbf{x}_j, \mathbf{q}^0)$ in the above-given algorithm. This calculation requires finding the following derivatives of the trajectory with respect to parameters:

$$\frac{\partial \mathbf{x}_j(\tau_{j+1}; \tau_j, \mathbf{s}_j, \mathbf{p})}{\partial p_l} \quad \text{and} \quad \frac{\partial \mathbf{x}_j(\tau_{j+1}; \tau_j, \mathbf{s}_j, \mathbf{p})}{\partial s_{jk}} \tag{6}$$

for $l = 1, 2, \cdots, m$ and $k = 1, 2, \cdots, d$. To find these derivatives, one needs to solve sensitivity equations that is a flexible and quite efficient approach (compared to internal or external differentiation [10]). The cost of calculating the sensitivities is the simultaneous integration of the sensitivity equations by solving a system of $m + d$ differential equations at each iteration.

3. The Modified Algorithm and Sensitivity Analysis Using the Adjoint Method

Here our aim is to employ the adjoint method [12] so that the computational cost of the algorithm decreases. First, we would like to note that gradients of objective and constraint functions given in (4) can be calculated using the adjoint method. However, the computational cost of calculating the gradients will be larger than the forward sensitivity analysis since the adjoint method is only effective to find gradients of scalar valued functions as noted in [13]. This is to say that one needs to solve d adjoint equations for each one of d elements of vector g_j to calculate the gradient of each g_j. This implies that finding a solution to (4) with no modifications requires solving d^2 differential equations in each subinterval per iteration.

The above discussion implies that the adjoint method cannot be effectively used to calculate the derivatives given in (6). To reduce the computational cost of the algorithm, we need to consider an equivalent optimization problem. First, recall that one needs to solve a system of d linear differential equations to find the directional derivatives for each scalar function of the state variables $\mathbf{x}$. Hence we need to reduce the number of appearances of the state vectors $\mathbf{x}$ in objective and constraint functions.

Using Assumption 1 along with the initial conditions given in (2) and (3), objective function of (4) can be written as follows:

$$l_{ij}(\mathbf{q}) = \sigma_{ij}^{-1}\big(\eta_{ij} - g_{ij}\big(\mathbf{s}_j, \mathbf{q}\big)\big). \tag{7}$$

Remark 2. With this simplification, the objective function does depend on the parameter vector **q** but not on the state variables **x**. Hence, there is no need to use the adjoint method when calculating the gradient of the objective function $L(\mathbf{q})$.

Now we consider the following optimization problem:

$$\begin{aligned} &\underset{\mathbf{q}}{\text{minimize}} && L(\mathbf{q}) \\ &\text{subject to} && h^{(j)}(\mathbf{x}_j, \mathbf{q}) = \|G_j\big(\mathbf{x}_j, \mathbf{q}\big)\|_2^2 = 0 \quad \text{for } j = 0, 1, \cdots, K-1. \end{aligned} \tag{8}$$

One can easily see that this problem is equivalent to (4). The difference between these two, on the other hand, is that there is a scalar constraint for each subinterval in the latter while the former contains vector valued constraints.

Here our aim is to compute the sensitivities

$$\frac{dh^{(j)}}{d\mathbf{q}}\big(\mathbf{x}_j(\tau_{j+1}), \mathbf{q}\big) \text{ for } j = 0, 1, \cdots, K-1 \tag{9}$$

using the adjoint method. This calculation requires solution of d differential equations on each subinterval rather than $m + d$ differential equations as in the case of sensitivity equations.

The Adjoint Method

Here we consider the initial value problem (2). For any interval number j between 0 and $K-1$, we have the trajectory $\mathbf{x}_j(t) = \mathbf{x}_j(t; \tau_i, \mathbf{s}_i, \mathbf{p})$ then, we consider the following function:

$$H_j\big(\mathbf{q}\big) = \int_{\tau_j}^{\tau_{j+1}} h^{(j)}\big(\mathbf{x}_j(t), \mathbf{q}\big)\, dt$$

to obtain desired sensitivities (9). Let $\boldsymbol{\lambda}(t)$ be any vector valued function of dimension d defined for $t \in [\tau_j, \tau_{j+1}]$. Now we consider the following augmented function:

$$\mathcal{L}(\mathbf{q}) = H_j(\mathbf{q}) + \int_{\tau_j}^{\tau_{j+1}} \lambda^T(t)\big(\dot{\mathbf{x}}_j(t) - f\big(\mathbf{x}_j(t), t, \mathbf{p}\big)\big)\, dt. \tag{10}$$

Here we use integration by parts to obtain

$$\int_{\tau_j}^{\tau_{j+1}} \lambda^T \dot{\mathbf{x}}_j\, dt = \lambda^T \mathbf{x}_j \Big|_{\tau_j}^{\tau_{j+1}} - \int_{\tau_j}^{\tau_{j+1}} \dot{\lambda}^T \mathbf{x}_j\, dt.$$

Using this equality in (10), we obtain

$$\mathcal{L}(\mathbf{q}) = \int_{\tau_j}^{\tau_{j+1}} \Big(h^{(j)}\big(\mathbf{x}_j(t), \mathbf{q}\big) - \dot{\lambda}^T(t)\mathbf{x}_j(t) - \lambda^T(t) f\big(\mathbf{x}_j(t), t, \mathbf{p}\big)\Big)\, dt + \lambda^T \mathbf{x}_j \Big|_{\tau_j}^{\tau_{j+1}} \tag{11}$$

We would like to note that $\frac{d\mathcal{L}}{d\mathbf{q}} = \frac{dH_j}{d\mathbf{q}}$. Hence, taking the total derivative of augmented objective function (11), we obtain:

$$\frac{d\mathcal{L}}{d\mathbf{q}} = \int_{\tau_j}^{\tau_{j+1}} \Big(\big(h^{(j)}_{\mathbf{x}_j} - \dot{\lambda}^T(t) - \lambda^T(t) f_{\mathbf{x}_j}\big)\mathbf{x}_{j_{\mathbf{q}}} + h^{(j)}_{\mathbf{q}} - \lambda^T f_{\mathbf{q}}\Big)\, dt + \big(\lambda^T \mathbf{x}_j \big|_{\tau_j}^{\tau_{j+1}}\big)_{\mathbf{q}}. \tag{12}$$

To obtain sensitivity of $H_j(\mathbf{q})$ with respect to parameters **q**, we require that $\lambda(t)$ satisfies the following initial value problem

$$\dot{\lambda}^T(t) = h^{(j)}_{\mathbf{x}_j} - \lambda^T(t) f_{\mathbf{x}_j}, \qquad \lambda^T(\tau_{j+1}) = 0. \tag{13}$$

Hence, by (12), sensitivity of H_j can be calculated as follows:

$$\frac{dH_j}{d\mathbf{q}} = \frac{d\mathcal{L}}{d\mathbf{q}} = \int_{\tau_j}^{\tau_{j+1}} \Big(h^{(j)}_{\mathbf{q}} - \lambda^T f_{\mathbf{q}}\Big)\, dt - \lambda^T(\tau_j)\big(\mathbf{s}_j\big)_{\mathbf{q}}. \tag{14}$$

As noted before, we need to calculate the sensitivities of $h^{(j)}$ at time τ_{j+1} (see eq. (9)) rather than sensitivities of H_j. By Leibnitz integral rule, we have the following equality:

$$\frac{dh^{(j)}}{d\mathbf{q}}\big(\mathbf{x}_j(\tau_{j+1}), \mathbf{q}\big) = \frac{d}{d\tau_{j+1}} \frac{dH_j}{d\mathbf{q}}$$

Using (14) in the above equality, we obtain:

$$\frac{dh^{(j)}}{d\mathbf{q}}\big(\mathbf{x}_j(\tau_{j+1}), \mathbf{q}\big) = \Big(h^{(j)}_{\mathbf{q}} - \lambda^T f_{\mathbf{q}}\Big)(\tau_{j+1}) - \int_{\tau_j}^{\tau_{j+1}} \lambda^T_{\tau_{j+1}} f_{\mathbf{q}}\, dt - \lambda^T_{\tau_{j+1}}(\tau_j)(\mathbf{s}_j)_{\mathbf{q}} \tag{15}$$

where $\lambda^T_{\tau_{j+1}} = \frac{\partial \lambda^T}{\partial \tau_{j+1}}$. By (13), this newly introduced quantity satisfies the following differential equation:

$$\dot{\lambda}^T_{\tau_{j+1}} = -\lambda^T_{\tau_{j+1}} f_{\mathbf{x}_j}. \tag{16}$$

By taking the total derivative of the equality $\lambda^T(\tau_{j+1}) = 0$ as in [12], we obtain $\lambda^T_{\tau_{j+1}}(\tau_{j+1}) = -\dot{\lambda}^T(\tau_{j+1})$. This implies by (13) that $\lambda^T_{\tau_{j+1}}(\tau_{j+1}) = -\dot{\lambda}^T(\tau_{j+1}) = -h^{(j)}_{\mathbf{x}_j}(\tau_{j+1})$. Hence, (15) can be written in the following form:

$$\frac{dh^{(j)}}{d\mathbf{q}}(\mathbf{x}_j, \mathbf{q})|_{t=\tau_{j+1}} = h^{(j)}_{\mathbf{q}}(\tau_{j+1}) - \int_{\tau_j}^{\tau_{j+1}} \lambda^T_{\tau_{j+1}} f_{\mathbf{q}}\, dt - \lambda^T_{\tau_{j+1}}(\tau_j)(\mathbf{s}_j)_{\mathbf{q}}. \tag{17}$$

The above-given formula for the calculation of gradients of constraint functions $h^{(j)}$ contains only $2d + m$ non-zero elements. The fact that gradient vector of $h^{(j)}$ is sparse can be used to efficiently implement an algorithm to solve optimization problem (8). In the following section, we calculate the sensitivities for a Lotka-Volterra system using the above equality.

4. An Application: Classical Lotka-Volterra Predator Prey System

To describe an ecological system consisting of one predator and one prey, one can consider the following model of Lotka and Volterra

$$\begin{aligned} \dot{x}^{(1)} &= -p_1 x^{(1)} + p_2 x^{(1)} x^{(2)} \\ \dot{x}^{(2)} &= p_3 x^{(2)} - p_4 x^{(1)} x^{(1)}. \end{aligned} \tag{18}$$

The measurements at times $\tau_j = j$ for $j = 1, 2, \cdots, 10$ are simulated by numerical integration of (18) with initial conditions $y_1(0) = 0.4$ and $y_2(0) = 1$ and parameter values $\mathbf{p} = (1, 1, 1, 1)$ then they are diturbed by normally distributed pseudo-random noise ($N(0, \sigma^2)$ with $\sigma^2 = 0.05$.)

The Lotka-Volterra system has been used as a test example for parameter estimation algorithms (see, for instance, [8,21–23]). The solution has singularities at various combinations of parameters values. We follow [8, p.6] and take $\mathbf{p_0} = (0.5, 0.5, 0.5, -0.2)$ which results in a pole near $t = 3.3$, where any ODE solver breaks down [21]. Hence, the single shooting approach with the above-given initial guess $\mathbf{p}_0$ must fail to estimate the parameters $\mathbf{p}$.

For the above-given parameters, denote the measurements for $x^{(1)}$ and $x^{(2)}$ at τ_j by d^1_j and d^2_j, respectively. Then the constrained nonlinear minimization problem can be written explicitly as follows :

$$\begin{aligned} &\underset{\mathbf{q}}{\text{minimize}} \quad \sum_{j=0}^{10} \left((s^1_j - d^1_j)^2 + (s^2_j - d^2_j)^2 \right) \\ &\text{subject to} \quad h^{(j)}(\mathbf{x}_j, \mathbf{q}) = 0 \quad \text{for } j = 0, 1, \cdots, 9. \end{aligned} \tag{19}$$

where $h^{(j)}(\mathbf{x}_j, \mathbf{q}) = \left(\mathbf{x}^{(1)}_j(\tau_{j+1}) - s^1_{j+1}\right)^2 + \left(\mathbf{x}^{(2)}_j(\tau_{j+1}) - s^1_{j+1}\right)^2$.

Here our aim is to calculate the gradient of each function in this nonlinear optimization problem. Since the objective function contains only the parameters of the discrete trajectory $\mathbf{s}_j$ for $j = 0, 1, \cdots, 10$ one can easily find its gradient. On the other hand, computing the gradient of $h^{(j)}$ with respect to parameters $\mathbf{q}$ requires using formula (17).

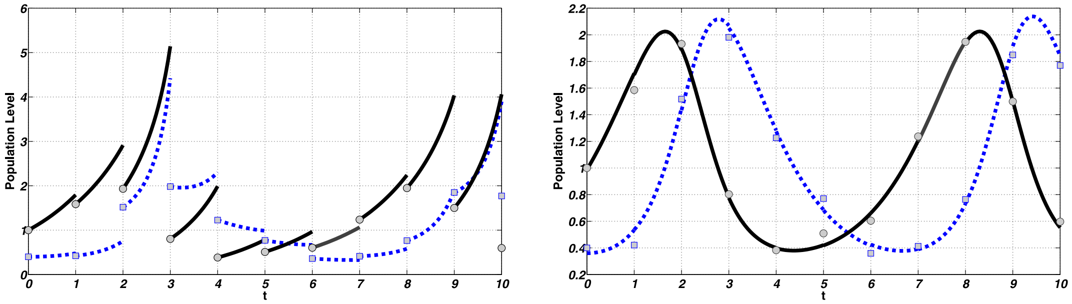
It is obvious from the definition of $h^{(j)}$ and (17) that the gradient of each constraint function has a special form. In particular, one needs to calculate the nonzero elements of the gradient vector $\nabla h^{(j)}$ i.e. the derivatives with respect to $\mathbf{s}_j$, $\mathbf{s}_{j+1}$ and $\mathbf{p}$. Finding the derivatives with respect to $\mathbf{s}_j$ and $\mathbf{p}$ requires integration of the following system of nonautonomous linear differential equations from τ_{j+1} to τ_j:

$$\begin{bmatrix} \dot{\Lambda}_1(t) & \dot{\Lambda}_2(t) \end{bmatrix} = \begin{bmatrix} \Lambda_1(t) & \Lambda_2(t) \end{bmatrix} \begin{bmatrix} -p_1 + p_2 x^{(2)}_j(t) & p_2 x^{(1)}_j(t) \\ -p_4 x^{(2)}_j(t) & p_3 - p_4 x^{(1)}_j(t) \end{bmatrix}$$

satisfying $\Lambda_i(\tau_{j+1}) = -2\left(\mathbf{x}^{(i)}_j(\tau_{j+1}) - s^i_{j+1}\right)$ for $i = 1, 2$.

Now one can compute the nonzero elements of the gradient $\nabla h^{(j)}$as follows:

$$\begin{aligned} \frac{dh^{(j)}}{ds^i_{j+1}} &= -2\left(\mathbf{x}^{(i)}_j(\tau_{j+1}) - s^i_{j+1}\right) \text{ for } i = 1, 2 \\ \frac{dh^{(j)}}{ds^i_j} &= \Lambda_i(\tau_j) \text{ for } i = 1, 2 \\ \frac{dh^{(j)}}{dp_1} &= \int_{\tau_j}^{\tau_{j+1}} \Lambda_1(t) x^{(1)}_j(t)\, dt \\ \frac{dh^{(j)}}{dp_2} &= -\int_{\tau_j}^{\tau_{j+1}} \Lambda_1(t) x^{(1)}_j(t) x^{(2)}_j(t)\, dt \\ \frac{dh^{(j)}}{dp_3} &= -\int_{\tau_j}^{\tau_{j+1}} \Lambda_2(t) x^{(2)}_j(t)\, dt \\ \frac{dh^{(j)}}{dp_4} &= \int_{\tau_j}^{\tau_{j+1}} \Lambda_2(t) x^{(1)}_j(t) x^{(2)}_j(t)\, dt \end{aligned}$$



(a) (b)

Fig. 1. Multiple shooting trajectories. Panel (a) illustrates the solutions to (18) simulted in each subinterval for the initial guess $\mathbf{p}_0 = (0.5, 0.5, 0.5, -0.2)$ along with the data points. Similarly, panel (b) shows the same solutions with estimated parameters $p_f = (1.0263, 1.04640.9784, 0.9688)$ in 8 iterations.

Remark 3. As seen in the above formulas, calculation of directional derivatives with respect to parameters $\mathbf{p}$ requires integrating some multiplications of state variables $\mathbf{x}_j(t)$ and $\Lambda_i(t)$ for $i = 1, 2$ over the interval I_j. Note that using an ODE solver with constant step size provides a computationally efficient way of approximating these integrals.

We computed the gradients of the constraint functions $h^{(j)}$ using the above-defined formulas and obtained the trajectories presented in Figure 1. In the panel (a), solutions to the 10 initial value problems with parameters p_0 are illustrated as the initial multiple shooting trajectories. In Panel (b), on the other hand, solutions to the same initial value problems are shown for the estimated parameters after 8 iterations of the active set (sqp with a line search) algorithm in Matlab 2015a. Note that this algorithm uses the quasi-Newton approximation of the Hessian matrix; hence it can be used as the optimization routine that is needed for the parameter estimation process.

We also obtained the following estimations of the parameters $p = (1, 1, 1, 1)$ after 8 iterations of the active set algorithm. We disturbed the data obtained from the simulation of (18) with a Gaussian noise term with $\sigma = 0.05$. Repeating parameter estimation process with 10 times with different realizaions of the noise term we obtained the following means and standard deviations for the parameters $\mathbf{p}$.

	p_1	p_2	p_3	p_4
mean	0.9832	0.9870	1.0254	1.0290
standard deviation	0.0554	0.0590	0.0397	0.0371

Note that these results are comparable with the results obtained in [8].

5. Conclusion

In this paper, we have modified the classical multiple shooting algorithm for parameter estimation in ODEs. In particular, the objective function is taken as a function of only the parameters identifying the discrete trajectory $\mathbf{s}_j$ and the continuity constraints are taken as scalar functions of the continuous trajectories $\mathbf{x}_j(t; \tau_j, \mathbf{s}_j, \mathbf{p})$ on each subinterval I_j. This allowed us to use so-called the adjoint method to find the gradient vector of each continuity constraint in each of the intervals I_j.

Recall that the classical multiple shooting algorithms require solutions to sensitivity equations to determine the Jacobian matrix. Since the number of sensitivity equations increases as the number of parameters increases, the adjoint method provides a computationally efficient alternative for finding such directional derivatives. In addition to this, as claimed in [12], the adjoint method is more tractable (compared to the forward sensitivity equations) when the number of parameters and state variables become large. Finally, we applied this new method to a well-known predator-prey system which has a singularity for some parameter values. This implies that any ODE solver breaks down, and hence the single shooting method cannot be used to estimate its parameters. We give the explicit expressions for the adjoint equations and nonzero components of the gradient vectors for this system.

Finally, there are several interesting issues that should be further explored or extended. In this paper, we focused on using the adjoint method to estimate the unknown parameters of a system of ODEs using observed data. We would like to note that both the adjoint method, and the multiple shooting method for parameter estimation has been developed for delay differential equations [14,24–26], differential algebraic equations [12,27] and partial differential equations [21,28–30]. Hence all of these parameter estimation algorithms for qualitatively different systems can be modified, as done in this paper, to find directional derivatives with respect to unknown parameters using the adjoint method. Moreover, the adjoint method can also be employed to solve constrained optimal control problems (see e.g. [31,32]).

References

[1] J.R. Banga, C.G. Moles, A.A. Alonso, Global optimization of bioprocesses using stochastic and hybrid methods, in: Frontiers in global optimization, Springer, 2004, pp. 45–70.

[2] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, New York, 1993.

[3] H.G. Bock, Numerical treatment of inverse problems in chemical reaction kinetics, in: Modelling of chemical reaction systems, Springer, 1981, pp. 102–125.

[4] H.G. Bock, Recent advances in parameter identification techniques for ODE, in: Numerical treatment of inverse problems in differential and integral equations, Springer, 1983, pp. 95–121.

[5] P. Drag, K. Styczeń, Multiple shooting SQP algorithm for optimal control of DAE systems with inconsistent initial conditions, in: Recent Advances in Computational Optimization, Springer, 2015, pp. 53–65.

[6] H.G. Bock, S. Körkel, J.P. Schlöder, Parameter estimation and optimum experimental design for differential equation models, in: Model based parameter estimation, Springer, 2013, pp. 1–30.

[7] M. Peifer, J. Timmer, Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting, IET Systems Biology 1 (2) (2007) 78–88.

[8] H.G. Bock, E. Kostina, J.P. Schlöder, Direct multiple shooting and generalized gauss-newton method for parameter estimation problems in ODE models, in: Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 1–34.

[9] L.T. Biegler, Nonlinear programming: concepts, algorithms, and applications to chemical processes, Siam, 2010. Vol. 10

[10] M. Peifer, J. Timmer, Parameter estimation in ordinary differential equations using the method of multiple shooting–a review, 2005.

[11] Y. Cao, S. Li, L. Petzold, Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software, Journal of Computational and Applied Mathematics 149 (1) (2002) 171–191.

[12] Y. Cao, S. Li, L. Petzold, R. Serban, Adjoint sensitivity analysis for differential-algebraic equations: The adjoint DAE system and its numerical solution, SIAM Journal on Scientific Computing 24 (3) (2003) 1076–1089.

[13] B. Sengupta, K.J. Friston, W.D. Penny, Efficient gradient computation for dynamical models, NeuroImage 98 (2014) 521–527.

[14] J.J. Calver, Parameter estimation for systems of ordinary differential equations, 2019 Ph.d. thesis.

[15] M. Jeon, Parallel optimal control with multiple shooting, constraints aggregation and adjoint methods, Journal of Applied Mathematics and Computing 19 (1-2) (2005) 215.

[16] O. Richter, P. Nörtersheuser, W. Pestemer, Non-linear parameter estimation in pesticide degradation, Science of the Total Environment 123 (1992) 435–450.

[17] J. Timmer, H. Rust, W. Horbelt, H. Voss, Parametric, nonparametric and parametric modelling of a chaotic circuit time series, Physics Letters A 274 (3-4) (2000) 123–134.

[18] A.D. Stirbet, P. Rosenau, A.C. Ströder, R.J. Strasser, Parameter optimisation of fast chlorophyll fluorescence induction model, Mathematics and Computers in Simulation 56 (4-5) (2001) 443–450.

[19] H. von Grünberg, M. Peifer, J. Timmer, M. Kollmann, Variations in substitution rate in human and mouse genomes, Physical Review Letters 93 (20) (2004) 208102.

[20] S.B. Hazra, Large-scale PDE-constrained optimization in applications, Springer Science & Business Media, 2009. Vol. 49

[21] K. Schittkowski, Numerical data fitting in dynamical systems: a practical introduction with applications and software, Vol. 77, Springer Science & Business Media, 2013.

[22] J. Swartz, H. Bremermann, Discussion of parameter estimation in biological modelling: Algorithms for estimation and evaluation of the estimates, Journal of Mathematical Biology 1 (3) (1975) 241–257.

[23] L. Edsberg, P.Å. Wedin, Numerical tools for parameter estimation in ODE-systems, Optimization Methods and Software 6 (3) (1995) 193–217.

[24] J. Calver, W. Enright, Numerical methods for computing sensitivities for ODEs and DDEs, Numerical Algorithms 74 (4) (2017) 1101–1117.

[25] H. Voss, M. Peifer, W. Horbelt, H. Rust, J. Timmer, G. Gousebet, Identification of chaotic systems from experimental data, in: Chaos and its reconstruction', Nova Science Publishers Inc., New York, 2003, pp. 245–286. 2003

[26] W. Horbelt, J. Timmer, H.U. Voss, Parameter estimation in nonlinear delayed feedback systems from noisy data, Physics Letters A 299 (5-6) (2002) 513–521.

[27] H. Bock, M. Diehl, D. Leineweber, J. Schlöder, A direct multiple shooting method for real-time optimization of nonlinear DAE processes, in: Nonlinear model predictive control, Springer, 2000, pp. 245–267.

[28] A.M. Bradley, Pde-constrained optimization and the adjoint method, 2013, Tech. rep., Technical Report. Stanford University. https://cs. stanford.edu/ ambrad.

[29] X. Xun, J. Cao, B. Mallick, A. Maity, R.J. Carroll, Parameter estimation of partial differential equation models, Journal of the American Statistical Association 108 (503) (2013) 1009–1020.

[30] T. Müller, J. Timmer, Parameter identification techniques for partial differential equations, International Journal of Bifurcation and Chaos 14 (06) (2004) 2053–2060.

[31] B.C. Fabien, Numerical solution of constrained optimal control problems with parameters, Applied Mathematics and Computation 80 (1) (1996) 43–62.

[32] M. Diehl, H.G. Bock, H. Diedam, P.B. Wieber, Fast direct multiple shooting algorithms for optimal robot control, in: Fast motions in biomechanics and robotics, Springer, 2006, pp. 65–93.