Parameter estimation in ordinary differential equations using the method of multiple shooting – a review

M. Peifer* and J. Timmer

Freiburg Centre for Data Analysis and Modelling
Eckerstr. 1, 79104 Freiburg, Germany
*peifer@fdm.uni-freiburg.de

May 31, 2005

Abstract

If the parameters of a model are unknown, results from simulation studies can be misleading. Such a scenario can be avoided by estimating the parameters before analysing the system e.g. with simulations. Almost all approaches for estimating parameters in ordinary differential equations are either having a small convergence region or having a immense computational cost. The method of multiple shooting can be situated in between of these extremes. In spite of its good convergence and stability properties, literature regarding the practical implementation and providing some theoretical background is rarely available. This review attempts to supply all necessary information for a successful implementation and discusses the basic facts of the involved numerics. To show the performance of the method, two illustrative examples are discussed after the approach has been presented.

1 Introduction

The problem of estimating parameters in ordinary differential equations (ODEs) from partially observed noisy data appears in many areas of applied sciences and engineering. Since most of the ODEs are nonlinear, all methods regarding parameter estimation are showing an interplay between simulating the trajectory and optimisation. The simulation of the trajectory is usually done by convenient ODE solvers. Whereas the optimisation differs drastically and can be classified of being either a stochastic optimisation procedures or a deterministic. Methods based on stochastic minimisation routines are e.g., random search and adaptive stochastic methods [1, 47, 4, 43], clustering methods [37], evolutionary computation [18] and simulated annealing. A detailed discussion of these methods with respect to parameter identification in ordinary differential equations can be found in [3]. The disadvantage of stochastic optimisers is mainly their immense computational cost which is the price for the flexibility and stability of these methods.

On the other side, deterministic optimisation procedures such as sequential quadratic programming (SQP), Newton methods, quasi-Newton methods, etc. are computationally efficient but they tend to converge to local minima. In case of parameter identification in ODEs the problem of convergence to local minima is predominant if the so called initial value approach is considered. This approach utilises that the trajectory is uniquely determined by the parameters and initial values. Minimising a maximum-likelihood functional with respect to parameters and initial values should therefore solve the inverse problem.

The situation stated above further suggests that there might be a trade-off between computational efficiency and stability for estimating parameters in ODEs. Multiple shooting can be located in between these two extremes. In order to use more information of the measurements, the inverse problem is embedded into a higher dimension and the possibility of having non-continuous trajectories is allowed during the optimisation process. This offers the possibility that the trajectory can stay close to the measurements for the whole runtime of the procedure, which increases the stability significantly. Hence, deterministic optimisation approaches can be used to reduce the computational cost without totally destabilising the method. Multiple shooting was introduced by J. Stoer and R. Bulirsch in the early seventies [40] and was substantially enhanced and mathematically analysed by H.G. Bock [8, 9, 10]. Here, some of the well elaborated mathematical details are presented, but always in scope of practically implementing these ideas. Keeping also track on the algorithmic issues can be regarded as the major intension of this review. Since this aspect is neglected in literature the accessibility and re-implementation of multiple shooting is currently limited.

The remainder of this article is organised as follows: After stating the estimation problem, Sec. 2, multiple shooting is described in detail, Sec. 3. In order of their algorithmic occurrence all components of the procedure are discussed. The treatment of unidentifiable estimation problems is briefly introduced by a regularisation approach in Sec. 4. Finally, two examples emphasis the practical relevance of the method, Sec. 5.

2 The Estimation Problem

Suppose that a dynamical system is given by the d-dimensional state variable $x(t) \in \mathbb{R}^d$ at time $t \in I = [t_0, t_e]$, which is the unique and differentiable solution of the initial value problem

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

The right-hand side of the ODE depends in addition on some parameters $p \in \mathbb{R}^{n_p}$. It is further assumed that f is continuously differentiable with respect to the state x and parameters p. Let $Y^{(k)}$ define the k-th component of some vector Y and the data $(Y_i)_{i=1\cdots n}$ are assumed to satisfy the observation equation

$$Y_i^{(j)} = g^{(j)}(x(t_i), p) + \sigma_{ij}\epsilon_{ij} \qquad j = 1, \cdots, obs\ , \tag{2}$$

for some link function $g : \mathbb{R}^d \to \mathbb{R}^{obs}$, $d \geq obs$, $\sigma_{ij} > 0$, and ϵ_i's are independent and standard Gaussian distributed random variables. The sample points t_i are ordered such that $t_0 \leq t_1 < \cdots < t_n \leq t_e$ and the link function g is again continuously differentiable in both variables. Eqs. (1) and (2) defines an one experiment design but suppose you have several experiments, possibly under different experimental conditions. In this case Eq. (2) depends on each experiment and must be modified in the following manner

$$Y_{ik}^{(j)} = g^{(j)}(x(t_{ij}), p) + \sigma_{ijk}\epsilon_{ijk} \qquad k = 1, \cdots, n_{exp}\ . \tag{3}$$

Certain parameters may depend on the experiments itself, but the treatment of these local parameters and the different experiments requires only obvious modifications of the described procedures and therefore only the one-experiment design $n_{exp} = 1$ is considered.

On the basis of the measurements $(Y_i)_{i=1,\cdots,n}$ the task is now to estimate the initial state x_0 and the parameters p. The principle of maximum-likelihood provides an appropriate cost function which has to be minimised for estimating these parameters. Defining $x(t_i; x_0, p)$ as being the trajectory at time t_i with parameters p and initial value x_0, the cost function is then given by

$$\mathcal{L}(x_0, p) = \sum_{i=1}^{n} \sum_{j=1}^{obs} \frac{\left(Y_i^{(j)} - g^{(j)}(x(t_i; x_0, p), p)\right)^2}{2\sigma_{ij}^2}\ . \tag{4}$$

A direct minimisation of $\mathcal{L}$ with respect to x_0, p leads to the so called initial value approach.

2.1 Initial Value Approach

The development of the initial value approach has a long history, e.g. [31, 38] and recently [2, 3]. Again, deterministic and stochastic optimisation methods can be considered. If stochastic optimisation procedures are used for minimising the likelihood, Eq. (4), an immense computational cost is again needed. On the other hand deterministic or local optimisation algorithms are having a small domain in parameter space for which the method converges to the global minimum. These problems are due to the following difficulties:

1. The optimisation problem is highly non-linear such that local optimisation routines tend to converge to local minima.

2. The solution of the differential equation can become unstable such that the trajectory diverges before the last time point t_n is reached.

An efficient and robust method minimising these effects therefore needs a modification of the optimisation scheme. One possibility of such a modification is multiple shooting.

2.2 Multiple Shooting

A detailed mathematical analysis of the multiple shooting method was done by H.G. Bock [8, 9, 10]. But in spite of the significant reduction of the problems given above, only a few research groups use multiple shooting. Besides the example given in Sec. 5 some applications of the method to measured data are e.g. [36, 41, 20, 45].

The basic idea of multiple shooting is that the parameter space is enlarged during the optimisation process. This offers the possibility to circumvent local minima because the procedure has more flexibility for searching the parameter space. It is realised by subdividing the time interval $I = [t_0, t_e]$ into $n_{ms} < n$ subintervals I_k such that each interval contains at least one measurement. Each of the intervals is assigned to an individual experiment having its own initial values $(x_0^k)_{k=1,\cdots,n_{ms}}$ but sharing the same parameters p. The only difference in the cost function Eq. (4) is that the trajectory $x(t_i; x_0, p)$ is replaced by the interval dependent trajectory $x(t_i; x_0^k, p)$ for all $k = 1, \cdots, n_{ms}$. Since the over-all trajectory for each $t \in I = I_1 \cup \cdots \cup I_{n_{ms}}$ is usually discontinuous at the joins of the subintervals, the fitted curve would not satisfy the smoothness assumption of the model, Eq. (1). To include smoothness of the final trajectory, the optimisation is constrained such that all discontinuties are removed which therefore leads to a constrained non-linear optimisation problem. This has the advantage that further equality and inequality constraints, such as parameter bounds or conservation relations can easily be implemented.

For each $k = 1, \cdots, n_{ms}$ let $t_k^+ = \max\{I_k\}$, $t_k^- = \min\{I_k\}$ and $\theta_k = (x_0^k, p)$, the optimisation problem can then be formulated in the following manner:

$$\begin{array}{ll} & \mathcal{L}(\theta_1, \cdots, \theta_{n_{ms}}) = \frac{1}{2} \sum_{j=1}^{obs} \sum_{k=1}^{n_{ms}} \sum_{\{i:t_i \in I_k\}} \left(R_{ijk}^a(\theta_k) \right)^2 = \min_{\theta_1, \cdots, \theta_{n_{ms}}} \\ \text{subject to} & \\ & x(t_i^+; \theta_i) - x(t_{i+1}^-; \theta_{i+1}) = 0 \qquad i = 1, \cdots, n_{ms} - 1 \\ & R_j^e(\theta_1, \cdots, \theta_{n_{ms}}) = 0 \qquad j = 1, \cdots, n_e \\ & R_k^g(\theta_1, \cdots, \theta_{n_{ms}}) \geq 0 \qquad k = 1, \cdots, n_g \ , \end{array} \tag{5}$$

where the continuity constrains are given at the first row of the constrains-part followed by some optional constraints R_j^e, R_k^g. The cost function $\mathcal{L}(\theta_1, \cdots, \theta_{n_{ms}})$ is equivalent to Eq. (4) if the continuity constraints are satisfied, hence

$$R_{ijk}^a(\theta_k) = \frac{Y_i^{(j)} - g^{(j)}(x(t_i; \theta_k), p)}{\sigma_{ij}} \ .$$

This non-linear programming type of problem can only be solved iteratively. We use the generalised-quasi-Newton method for solving (5), where the cost function is expanded up to the second order with respect to some initial guess $\theta^0 = (\theta_1^0, \cdots, \theta_{n_{ms}}^0)$. All contributions depending on the second derivative of R_{ijk}^a are neglected afterwards. This is possible because these contributions are vanishing asymptotically, $n \rightarrow \infty$, if the model assumptions are correct [10, 35]. From the quadratic approximation an update step $\Delta\theta = (\Delta\theta_1, \cdots, \Delta\theta_{n_{ms}})$ can be calculated by solving the linear programming problem

$$\begin{aligned} & \textstyle\sum_{j=1}^{obs} \sum_{k=1}^{n_{ms}} \sum_{\{i:t_i \in I_k\}} \left(R^a_{ijk}(\theta^0_k) + J_\theta R^a_{ijk}(\theta^0_k)\Delta\theta \right)^2 = \min_{\Delta\theta} \\ \text{subject to} \quad & \\ & x(t_i^+;\theta_i^0) - x(t_{i+1}^-;\theta_{i+1}^0) + J_{\theta_i} x(t_i^+;\theta_i^0)\Delta\theta_i - J_{\theta_{i+1}} x(t_{i+1}^-;\theta_{i+1}^0)\Delta\theta_{i+1} = 0 \\ & R^e_j(\theta^0) + J_\theta R^e_j(\theta^0)\ \Delta\theta = 0 \\ & R^g_k(\theta^0) + J_\theta R^g_k(\theta^0)\ \Delta\theta \geq 0\ , \end{aligned} \tag{6}$$

where J_θ denotes the Jacobian with respect to the parameters θ of the corresponding function. Setting $\theta^l = \theta^{l-1} + \Delta\theta$, $l = 1, \cdots$ and repeating (6) until $\Delta\theta \approx 0$, yields a minimum of Eqs. (5) under the condition that all parameters are identifiable and the constraints are not contradictory. These extra assumptions are necessary to fulfil the so called Kuhn-Tucker conditions for the solvability of constrained, non-linear optimisation problems [23, 10]. In Sec. 4, a regularisation approach is discussed for weakening these restrictions if non-identifiable parameters are present.

In combination with multiple shooting the generalised-quasi-Newton approach has three major advantages:

1. The optimisation is sub-quadratically convergent.
2. A transformation of Eqs. (6) can be found such that the transformed equations are numerically equivalent to the initial value approach, which is called *condensing*.
3. Due to the linearisation of the continuity constraints, they do not have to be fulfilled exactly after each iteration, but only at convergence.

Properties 1. and 2. are yielding the desired speed of convergence whereas 3. is mainly responsible for the stability of multiple-shooting. This results form the possibility that the algorithm can "walk trough forbidden ground" while searching the minimum. The main disadvantage is due to the linearisation of the cost function. It can easily happen that despite the update step $\Delta\theta$ is pointing in the direction of decreasing $\mathcal{L}$ the proposed step is too large. Such an overshooting is common to any simple optimisation procedures based on the local approximation of the cost function. A suitable approach to cure this defect is to *damp* the proposed step, which is realised by relaxing the update scheme to $\theta^l = \theta^{l-1} + \lambda^l \Delta\theta$ for some $\lambda^l \in (0, 1]$. Both, the condensation algorithm and the damping method are necessary for building up a fast and stable parameter estimator for ODEs. These procedures as well as the main program flow are the subject of following section.

3 Detailed Description of Multiple Shooting

In the previous section the basic idea and some aspects of the performance of multiple shooting was displayed without emphasising any algorithmic details of the method. To fill this gap, each module, starting from the initialisation and ending in the output of the procedure is discussed in detail. The different stages of the described method can be extracted from the flow chart, Fig. 1. Beginning at the initialisation, where e.g the multiple shooting mesh as well as the initial values of each interval are set, a first trial trajectory has to be integrated. Using these data, the linearised problem Eq. (6) can be formulated for the initial iteration and condensed in order to accelerate the minimisation process. To prevent overshooting, the relaxation or damping of the obtained update step is done. Then, a convergence criterion, such as $||\Delta\theta|| \approx 0$, decides whether the procedure is converged or a further iteration has to be taken into account by integrating a new trail trajectory, ... If the current iteration is convergent, output such as the parameter estimates, the estimates for the initial values as well as the covariance matrix for a statistical analysis of the solution is provided. The first non-trivial stage in the program flow is the integration of a trial trajectory.

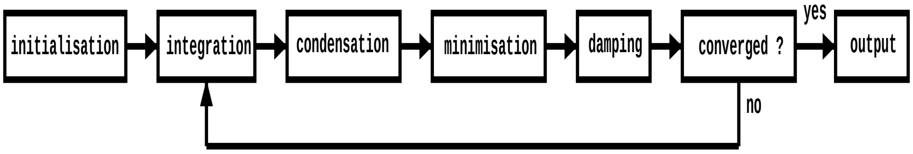

Figure 1: Program flow

3.1 Integration

The choice of the numerical integrator depends on the class of ODE given in Eq. (1) or its numerical stability. There are four major groups to consider:

1. non-stiff ODEs,
2. stiff problems,
3. delay differential equations
4. and differential algebraic equations.

For non-stiff ODEs standard numerical integrators such as the Runge-Kutta method [35] with an appropriate step size control can be used. Whereas, if the solution of the ODE has at least two different time scales which differ over orders of magnitude only stiff integrators are useful. Especially in the case of multiple shooting we propose to use ODESSA [25, 24], because the code is optimised for simultaneously solving the sensitivity equations. The significance of the trajectory's sensitivity is due to the linearisation given in Eq. (6) and will be discussed later. Delay differential equations (DDEs) cannot be represented by (1). Although DDEs are not ordinary differential equations, it is possible to adapt multiple shooting to this class of differential equations [19, 21, 46]. Since the right-hand-side of a DDE depends on the time delayed trajectory or a delay distribution, specially suited integrators are needed. A widely used DDE integrator is e.g. RETARD [16], for a deeper discussion of DDE we refer to [6]. Differential algebraic equations (DAEs) are differential equations in which algebraic relations between the state variables are present. In some cases the algebraic relations can be formulated as equality constants and are thus treated like constrained ODEs. This kind of separation is sometimes not possible such that special DAEs integrators have to be considered [11].

Besides the choice of the integrator, the solution of the sensitivity equations has to be obtained, because the Jacobian $J_\theta R^a_{ijk}(\theta^0_k)$ or $J_\theta x(t_i^+;\theta_i^0)$ in Eq. (6) contains derivatives of the trajectory with respect to the initial values and parameters:

$$\frac{\partial x(t;\theta_k)}{\partial x_0^{k(i)}} \quad \text{and} \quad \frac{\partial x(t;\theta_k)}{\partial p^{(j)}} \qquad t \in I_k,\ k=1,\cdots,n_{ms},\ i=1,\cdots,d,\ j=1,\cdots,n_p\ .$$

In order to calculate these quantities numerically, three approaches are feasible:

1. finite differences, called external differentiation [9, 10],

2. differentiation of the integration scheme, called internal differentiation [9, 10, 16],

3. and the simultaneous solution of the sensitivity equations [19].

The approximation of the derivatives by finite differences such as

$$\frac{\partial x(t;\theta_k)}{\partial x_0^{k(i)}} \approx h^{-1}\left(x(t;\theta_k + e_{i,x_0}h) - x(t;\theta_k)\right)$$

for some $h \ll 1$ and e_{i,x_0} being the i-th unit vector with respect to the initial value, leads to some difficulties. Due to the numerical integration, the trajectory $x(t;\theta_k)$ is corrupted by numerical noise. Since an adaptive integration step size is used, the maximal noise strength can be predefined by some constant $eps \ll 1$. Consequently, h cannot be chosen arbitrarily small without destabilising the method. Arguments based on the expansion of $x(t;\theta_k)$ reveals that the optimal choice is

$$h = \mathcal{O}(\sqrt{eps})\ , \tag{7}$$

see e.g. [22]. Unfortunately, the constant of proportionality in Eq. (7) depends on the second derivative and is therefore not known. Furthermore, a high integration accuracy is needed for achieving a suitable derivative. Thus, external differentiation should be avoided because of the unknown parameter h and the high computational cost.

Differentiating the integration scheme is considerably faster than external differentiation [9, 10] and the problem of adjusting a parameter does not occur. On the other hand, internal differentiation depends highly on the used integrator and has to be adapted whenever one decides to try another integration scheme.

A more flexible and quite efficient approach is the simultaneous integration of the sensitivity equations. Consider again a trajectory $x(t; x_0, p) = x(t; \theta)$ of Eq. (1) and the Jacobian J_θ, where the subscript indicates the variables to be differentiated. The time evolution of the sensitivities $S(t; \theta) = J_\theta x(t; \theta)$ is then given by the solution of

$$\begin{aligned} \frac{d}{dt} S(t;\theta) &= (J_\theta f)(x(t;\theta), t, p) + (J_x f)(x(t;\theta), t, p)\ S(t;\theta) \\ S_0 &= S(t_0;\theta) = \left(\mathbb{1}_{d\times d}, 0_{d\times n_p}\right)\ , \end{aligned} \tag{8}$$

where $\mathbb{1}_{d\times d}$ is the $d \times d$-unity matrix and $0_{d\times n_p}$ the $d \times n_p$-matrix of zeros. Simultaneously integrating Eq. (1) and Eq. (8) yields the trajectory as well as the desired sensitivities. It is further sufficient to restrict the step size control to the main ODE, Eq. (1). Doing this, the speed and the accuracy is comparable to the internal differentiation. It is therefore a matter of taste using either the internal differentiation or the the simultaneous solution of the sensitivity equations, Eq. (8).

Remark on the calculation of derivatives like $J_p f$, $J_x f$, etc.: Calculating such Jacobians by hand can be very time consuming and error-prone for big systems. Some kind of an automatic differentiation should therefore taken into account. One possibility is to generate the derivatives at runtime by using program packages like ADIFOR or ADOLC [7, 15]. Since the derivatives have to be recalculated for every function evaluation, this approach can easily slow down the whole method by a factor of 10. The calculation of the Jacobians should therefore processed before the program is executed which can be realised by using symbolic computation software, e.g. GinNaC [5].

3.2 Condensation

All information is now available for setting up Eq. (6). Suppose that $h_i = x(t_i^+) - x(t_{i+1}^-)$, $\Delta\theta_i = (\Delta x_0^i, \Delta p)$ for all $i = 1, \cdots, n_{ms} - 1$ and because of (8), $J_{x_0^{i+1}} x(t_{i+1}^-) = \mathbb{1}$ then the continuity constraints can be written as

$$h_i + J_{x_0^i} x(t_i^+)\ \Delta x_0^i + J_p x(t_i^+)\ \Delta p = \Delta x_0^{i+1} \qquad i = 1, \cdots, n_{ms} - 1\ . \tag{9}$$

According to Eq. (9) all initial value update steps at the multiple shooting intervals can therefore be related to Δx_0^1 by backward elimination. Inserting all other increments into Eq. (6) yields a system to be solved only for Δx_0^1 and Δp. Let R^a be the $n \cdot n_{ms} \cdot n_{obs}$-dimensional vector with components R^a_{ijk} and R^e, R^g respectively, the condensed problem is thus

$$\begin{array}{ll} & ||u_1^a + E_1^a\ \Delta x_0^1 + P_1^a\ \Delta p||^2 = \min_{\Delta x_0^1,\Delta p} \\ \text{subject to} & \\ & u_1^e + E_1^e\ \Delta x_0^1 + P_1^e\ \Delta p = 0 \\ & u_1^g + E_1^g\ \Delta x_0^1 + P_1^g\ \Delta p \geq 0\ , \end{array} \tag{10}$$

where $u_1^{a/e/g}$ and the matrices $E_1^{a/e/g}, P_1^{a/e/g}$ are determined by the recursion [9, 10]:

$$\begin{array}{lll} \text{Initialisation} & : & u_{n_{ms}}^{a/e/g} = R^{a/e/g}\ ,\ E_{n_{ms}}^{a/e/g} = J_{x_0^{n_{ms}}} R^{a/e/g}\ ,\ P_{n_{ms}}^{a/e/g} = J_p R^{a/e/g} \\ \text{For } i = n_{ms}, \cdots, 2 & : & u_{i-1}^{a/e/g} = u_i^{a/e/g} + E_i^{a/e/g}\ h_{i-1} \\ & & E_{i-1}^{a/e/g} = J_{x_0^{i-1}} R^{a/e/g} + E_i^{a/e/g}\ J_{x_0^{i-1}} h_{i-1} \\ & & P_{i-1}^{a/e/g} = P_i^{a/e/g} + E_i^{a/e/g}\ J_p R^{a/e/g}\ . \end{array} \tag{11}$$

The condensation algorithm eliminates (9) such that problem (10) is of lower dimension than the original, Eq. (6). Since (11) involves only matrix multiplications the desired increase in speed is achieved by solving only the condensed problem. After the solution of Eqs. (10) is determined, the actual update step $\Delta\theta$ is obtained by the recursion given in Eq. (9), which involves again only matrix multiplications.

3.3 Minimisation

The solution of the linear programming problem (10) can be obtained by a minimisation procedure with a sufficient pre-conditioning of the parameters. Appropriate minimisation algorithms for the constrained linear optimisation is e.g. LSEI [17] or the method of Stoer [39].

3.4 Damping

Damping or relaxation of the update is essential for the stability of the whole method. To judge if the proposed update step is descendant, some kind of level function has to be chosen. Such a level function must share the same monotony properties of the cost function close to the global minimum. In case of unconstrained problems, it is feasible to use the cost function $\mathcal{L}$ directly, whereas some modifications are necessary for constrained problems, such as multiple shooting. These modifications are due to the constraints entering the level function via Lagrange multipliers. A possible level function is then

$$T(\theta) = \mathcal{L}(\theta) + \underbrace{\sum_{i=1}^{n_{ms}+n_e-1} \alpha_i\ |R_i^e(\theta)|}_{\text{equality constr.}} + \underbrace{\sum_{i=1}^{n_g} \beta_i\ |\min\{0, R_i^g(\theta)\}|}_{\text{inequality constr.}}\ , \tag{12}$$

where α_i and β_i are bounded below the corresponding Lagrange multipliers. Although, with the aid of this level function, a downhill procedure can always be constructed by some one-dimensional line-search algorithm. According to [8, 9, 10] it turns out that the performance of using $T(\theta)$ is rather bad. This inefficiency is due to

1. line-search has a high computational cost since a new trajectory has to be integrated for each evaluation of Eq. (12) and

2. the local geometry of the minimisation problem is not adapted to the level function, leading to extremely small steps for badly conditioned problems.

To surmount these problems H.G. Bock [8, 9, 10] proposed to replace the line-search by some predictor-corrector method and the level function is changed to include the local geometry.

The adaption to geometrical issue can be realised by the so called natural level function. Suppose that the solution of the linearised minimisation problem, Eq. (6), can be obtained by the linear-operator $G(\theta)$, mapping $R = (R^a, R^e, R^g)$ to the parameter update $\Delta\theta$. For some θ, the natural level function is then given by

$$T_N(\theta) = ||G(\theta^l)R(\theta)||^2 \ , \tag{13}$$

where θ^l is again the initial value for the l-th iteration. It can further be shown that in the vicinity of the minimum, say θ^*, the natural level function is measuring the distance between θ and θ^* up to third order and feasibility condition for level functions is satisfied for every non-singular problem. Therefore all application specific geometric properties of the parameter "landscape" are reduced to the properties of the Euclidean-norm in the vicinity of θ^*, such that an efficient determination of the relaxation coefficient λ is expected.

Finding an appropriate λ for which the minimisation scheme is descendant involves again some kind of line-search to guarantee that $T_N(\theta^l + \lambda\Delta\theta) < T_N(\theta^l)$ is satisfied. Since the evaluation of the natural level function involves the integration of the trajectory and in addition the solution of the whole minimisation procedure, calculating T_N is quite expensive. Fortunately, it is possible to determine the relaxation coefficient without evaluating T_N if the quantity

$$\omega(\theta^l, G, \lambda) = \sup_{s\in(0,\lambda]} \left\{ \frac{||G(\theta^l)\left(J_\theta R(\theta^l + s\Delta\theta) - J_\theta R(\theta^l)\right)\Delta\theta||}{s||\Delta\theta||^2} \right\} \ , \tag{14}$$

for every $\lambda \in (0, 1]$ is known, [10]. Providing some $0 < \eta < 2$, the maximal step length λ^* is then given by the solution of

$$\lambda^* = \min\left\{1, \frac{\eta}{\omega(\theta^l, G(\theta^l), \lambda^*)\ ||\Delta\theta||}\right\} \ . \tag{15}$$

It can further be shown that this choice of λ^* yields $T_N(\theta^l + \lambda^* \Delta\theta) \leq (1 - \lambda^* \, (1 - \eta/2))^2 \; T_N(\theta^l) < T_N(\theta^l)$. Moreover, if the relaxation coefficient is chosen to be $\lambda^l \in [\lambda^*(\eta_1), \lambda^*(\eta_2)]$, for $0 < \eta_1 \leq \eta_2 < 2$, the damped generalised-quasi-Newton method converges to a full-step procedure, $\lambda = 1$, when the parameters are approaching the minimum.

Since $\omega(\theta^l, G, \lambda)$ is a-priori not known a suitable estimation or approximation is necessary. Demanding the coincidence of the estimator with Eq. (14) in the limit $\lambda \to 0$ automatically guarantees an appropriate relaxation scheme whenever a massive damping is needed. The estimator

$$\hat{\omega}(\theta^l, G, \lambda) = 2 \; \frac{||G(\theta^l) R(\theta^l + \lambda\Delta\theta) + (1-\lambda)\Delta\theta||}{||\lambda\Delta\theta||^2} \; , \tag{16}$$

satisfies this desired property, [10]. Although the proposed estimator is sufficient for the damping strategy, the replacement of ω with $\hat{\omega}$ in Eq. (15) leads again to many computationally expensive function evaluations of R. Providing a computationally efficient damping algorithm, it is firstly assumed that $\hat{\omega}(\theta^{l-1}, G, \lambda^{l-1})$ from the previous Gauss-Newton iteration is approximately constant for the actual iteration. If this assumption holds the damping of the current iteration is determined by

$$\lambda^l = \min\left\{1, \frac{\eta_0}{\hat{\omega}(\theta^{l-1}, G, \lambda^{l-1}) \; ||\Delta\theta||}\right\} \; , \tag{17}$$

for some $0 < \eta_0 < 2$. If the assumption is violated such that decreasing of the method cannot be guaranteed, $\hat{\omega}$ has to be recalculated from Eq. (16) but now using λ^l, given in Eq. (17). This procedure has to be repeated until a suitable relaxation coefficient has been obtained. For some $0 < \eta_0 < \eta_2 < 2$, $\tau \in [0.5, 1]$ and $0 < \tau_{min} \ll 1$, the damping procedure can be summarised by the following algorithm:

1. Set $j = 0$ and calculate the predictor $\mu_j = \eta_0 / \left(\hat{\omega}(\theta^{l-1}, G, \lambda^{l-1}) \; ||\Delta\theta||\right)$.

2. The predicted relaxation step is then given by

$$\lambda_j^{pred} = \begin{cases} 1 & \tau < \; \mu_j \\ \mu_j & \tau_{min} \leq \; \mu_j \leq \tau \\ \tau_{min} & \mu_j < \tau_{min} \end{cases} \; .$$

3. If $\hat{\omega}(\theta^l, G, \lambda_j^{pred})\lambda_j^{pred} \leq \eta_2$ then the proposed step $\lambda_j^{pred} = \lambda^l$ yields a descending update and is therefore accepted. Whereas, if the above statement is violated, $j = j + 1$ and

4. the prediction λ_{j-1}^{pred} is corrected by

$$\mu_j = \frac{\eta_0}{\hat{\omega}(\theta^l, G, \lambda_{j-1}^{pred}) \; ||\Delta\theta||} \; .$$

5. Step 2, 3 and 4 are repeated until a sufficient relaxation coefficient λ^l is found or the minimal step length τ_{min} is reached.

In order to ensure the numerical stability of the damping algorithm, a predefined minimal relaxation τ_{min} must be provided. An upper threshold τ is also given, which determines the transition from a damped procedure to a full step approach, $\lambda^l = 1$. Finally, η_0, η_2 are controlling the correction of the prediction, which can be estimated by $\lambda_j^{pred} < (\eta_0/\eta_2)\lambda_{j+1}^{pred}$. A suitable choice of these control parameters is e.g. $\tau_{min} = 0.01, \tau = 0.9, \eta_0 = 1$ and $\eta_2 = 1.8$. Since there is no information about $\hat{\omega}$ for the first Gauss-Newton iteration, one can chose $\hat{\omega}$ such that λ^1 attains the lower bound τ_{min}.

The described damping algorithm reflects the advantageous geometrical properties of the natural level function. Furthermore, the correction of the prediction λ_j^{pred} is rarely activated such that in most of the cases only one extra integration is needed to achieve an appropriate damping. Unfortunately, there are no rigorous proofs that this damping strategy always yields a descending method, which is due to the approximation of ω. But the algorithm provides excellent results in practice, we can therefore highly encourage the use of this damping scheme.

3.5 Output

Beside the pure estimation of parameters and initial values statistical information such as standard errors or confidence intervals for these values are essential in practice. In the case of maximum likelihood estimators the statistical properties can be derived in the asymptotic limit. Under mild conditions, the estimator is converging to the "true" parameters and the parameters are normally distributed [44]. The covariance matrix of the estimates can be obtained from the Fisher information matrix which can be approximated by

$$\mathrm{IF}(\hat{\theta})_{ij} = \frac{\partial^2 \mathcal{L}(\hat{\theta})}{\partial\theta_i \partial\theta_j} , \qquad (18)$$

where $\mathcal{L}$ is the negative logarithm of the likelihood. Inverting $\mathrm{IF}(\hat{\theta})$ then yields the covariance matrix for the estimated parameters $\hat{\theta}$.

The described procedure for estimating parameters in ODEs is a maximum likelihood approach, such that Eq. (18) provides a sufficient approximation of the covariance matrix. Most of the minimiser, e.g. [17], simultaneously calculate this covariance matrix within the quadratic approximation discussed in Sec. 2.2.

All these stages define the basic algorithm of multiple shooting which are valid in case of identifiable problems. As explained, the restriction of having only identifiable parameters is of great importance for the convergence of the algorithm, the damping strategy and the statistical analysis. To judge if the system of interest contains only identifiable parameters several methods can be applied, e.g [12, 14, 27, 34]. Since these methods can involve extremely tedious calculations even for small models, it is often a-priori not feasible to decide whether the system is identifiable. Alternatively, the multiple shooting method can be modified to obtain parameter estimates even if some parameters can not be identified. A possible implementation of such a strategy is described in the next section.

4 Regularisation

If some parameters are not identifiable at a certain domain in the parameter space, the matrix P_1^a of the condensed system (10) does not have its full rank whenever the algorithm tries to enter this region. The central idea of the regularisation approach is to manipulate the estimation process such that the rank criterion above is satisfied. The manipulation we propose can be regarded as heavily damp a specific parameter set such that they appear to be fixed.

A singular value decomposition [35] of $P_1^a = U \ \mathrm{diag}(w_1, \cdots, w_{n_p}) \ V^T$ has to be calculated first to determine if the rank criterion is fulfilled. Both matrices U, V^T are orthogonal, by convention V^T is the transposed matrix of V, and $\mathrm{diag}(w_1, \cdots, w_{n_p})$ is a diagonal matrix containing the positive (also by convention) singular values $w_1, \cdots, w_{n_p}$. It this further assumed that the singular values are in descending order $w_1 \geq \cdots \geq w_{n_p}$. The rank criterion is said to be violated if the condition number w_{n_p}/w_1 is below a given threshold $0 < \epsilon_c \ll 1$. Introducing a threshold is necessary because the numerical error prevents the condition number to vanish exactly. Therefore, the value of ϵ_c should be close to the accuracy of the machine. In order to judge, which parameters contribute to the violation of the rank criterion, the set $M_c = \{i \ : \ w_i/w_{n_p} \leq \epsilon_c\}$ of all singular directions is regarded. Let

$$\Pi_c = \sum_{i \in M_c} e_i \otimes e_i^T$$

be the projection onto the space of all singular directions, the regularisation can be realised by enlarging the corresponding singular values. For this reason, let us choose some $\Delta \gg w_1$. The regularised matrix $\tilde{P}_1^a$ is the given by

$$\tilde{P}_1^a = W \ \left(\mathrm{diag}(w_1, \cdots, w_{n_p}) + \Delta \ \Pi_c\right) \ V^T \ . \tag{19}$$

For a well adjusted value of Δ all parameters contributing to the singular directions are almost kept fixed if P_1^a is replaced by $\tilde{P}_1^a$ in Eq. (10), as desired. Since the described regularisation method is similar to the classical damping procedure of Levenberg and Marquardt [26, 30], regularisation can also be regarded as an individual damping of ill-conditioned directions. Unfortunately, the statistical analysis of regularised parameters is not possible anymore. On the other hand, regularisation can help to remove parameters until the system is identifiable. If once the set of identifiable parameters is found, the statistical analysis of Sec. 3.5 is possible for the remaining parameter estimates. Note that if some initial values are not identifiable, the same procedure can also be applied to the matrix E_1^a in Eq. (10).

5 Examples

To display how multiple shooting operates, we present two examples in the following. The first dataset is synthetically generated and can be considered as some kind of benchmark. Here, a chaotic oscillator is chosen as data generating system which is particularly hard to identify. This is due to the sensitive dependence of the trajectory with respect to the initial values. The the data of the second example can be obtained form the online material of [42]. In this example, biochemical data of the STAT5 pathway are modelled.

5.1 Example 1: Simulated Data

A dataset of the Lorenz system [29] is generated by integrating the system's equation

$$\begin{aligned} dx/dt &= \sigma(y-x) \\ dy/dt &= -y + x(r-z) \\ dz/dt &= xy - bz \end{aligned}$$

using a Runge-Kutta integration scheme with adaptive step size [35]. The three parameters are chosen to be $\sigma = 10$, $r = 28$, $b = 2.66$ and the sampling interval is set to $\Delta t = 0.1$. For this parameter setting the system is situated in the chaotic regime and the sampling guarantees a sufficient amount of data points within a period of the oscillation. Additionally, only one state variable, the x-component, is observed and corrupted with Gaussian white noise having a noise-to-signal ratio of 5%.

For the estimation procedure, the parameters are initially set to $\sigma_0 = 100$, $r_0 = b_0 = 0$ resulting a trajectory which is far away of the domain of convergence for the initial value approach. To assure that the trajectory is lying in the chaotic attractor the time interval $t \in [20, 40]$ is chosen for fitting purposes, yielding a total amount of $n = 200$ data points. Due to the sensitive dependence on the initial values of the chaotic motion it is not possible to fit an arbitrary long dataset. For this reason we divide the dataset in two parts and fitting them simultaneous in a multi-experiment fashion. Therefore, the strict continuity is weakened at the joint of the datasets allowing to fit the whole time span.

Snapshots of three multiple shooting iterations are shown in Fig. 2. Since nearly every data point is used as starting point for the multiple shooting interval the initial curve is highly erratic. The discontinuities are removed to obtain a smooth trajectory after the algorithm has been converged. As expected, the data are perfectly fitted and the

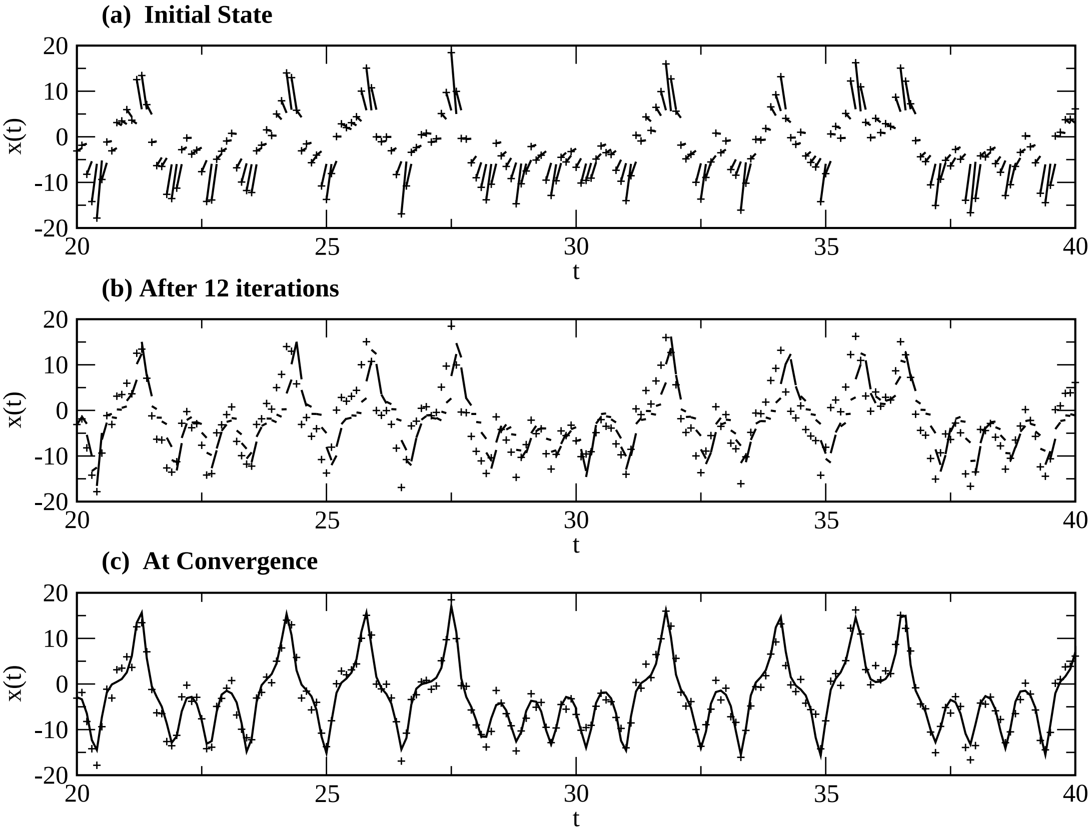

Figure 2: Identification of the Lorenz system using multiple shooting. Due to the large amount of multiples shooting intervals the initial trajectory (a) is highly discontinuous. Since discontinuities are remove during the iterations (b), the trajectory turns out to be continuous at convergence (c).

estimated parameters $\hat{\sigma} = 9.87 \pm 0.51$, $\hat{r} = 27.68 \pm 0.34$, $\hat{b} = 2.73 \pm 0.06$ are in accordance with the true parameters.

5.2 Example 2: Measured Data

So far, only simulated data are considered where the model structure is completely known. If measured data are modelled the choice or selection of a parameterised model which properly captures the underlying dynamics complicates the situation significantly. For the considered data the model selection procedure is thoroughly described in [42]. Here, we only concentrate on the identification of the model. Before doing this, it is necessary to provide a brief description of the model. The biochemical reaction starts at the activation or phosphorylation of the STAT5 molecule. This reaction is driven by the EPO receptor located at the cell membrane. Then, two activated STAT5 molecules can undergo a dimerisation. Only the a STAT5 dimer enters the cell-nucleus and can trigger the transcription of target genes. After the dimer has accomplished its job, the dimer separates and the STAT5 molecule is dephosphorylated. Finally, these single STAT5 molecules are able to reenter the cytoplasm and can again be activated by the receptor.

Assuming that the transport mechanisms from the cell membrane to nucleus are sufficiently fast, such that no concentration gradients can occur, the dynamical behaviour of the pathway can be approximate by an ODE. Since no in-vivo measurements inside of the nucleus are possible, all nuclear processes are condensed into a single step which contains a time delay. Let x_1 be the concentration of unphosphorylated STAT5, x_2 the activated STAT5 and x_3 the STAT5 dimer. The receptor activity is denoted by $EpoR_A(t)$ and x_4 is the concentration of STAT5 molecules staying in the nucleus. Unfortunately, no concentration of the reaction components can be measured directly. Instead, up to a-priori unknown scaling parameters s_1, s_2 the total amount of activated STAT5, $y_1 = s_1\ (x_2+x_3)$ and the total amount of STAT5 $y_2 = s_2\ (x_1 + x_2 + x_3)$ in the cytoplasm is accessible. For a given set of observations, the most simple identifiable ODE capturing all the properties stated above is

$$\begin{aligned}
\dot{x}_1 &= -k_1 x_1 EpoR_A(t) + k_2 x_3(t-\tau) \\
\dot{x}_2 &= -x_2^2 + k_1 x_1 EpoR_A(t) \\
\dot{x}_3 &= -k_2 x_3 + x_2^2 \\
\dot{x}_4 &= -k_2 x_3(t-\tau) + k_2 x_3\ ,
\end{aligned} \tag{20}$$

where k_1, k_2 are rate constants and τ is a delay parameter. Instead of using a "hard" delay in Eq. (20), we decided to use a delay chain approach. A delay chain of length N is a linear ODE of type

$$\begin{aligned}
\dot{q}_1 &= N/\tau\ (in(t) - q_1) \\
\dot{q}_2 &= N/\tau\ (q_1 - q_2) \\
&\dots \\
\dot{q}_{N-1} &= N/\tau\ (q_{N-2} - q_{N-1}) \\
\dot{out} &= N/\tau\ (q_{N-1} - out(t))\ .
\end{aligned}$$

Here, $in(t)$ is the input and $out(t)$ the output of the delay chain. It can be shown that such a chain generates a delay distribution having a mean delay of τ and a variance of τ^2/N. In case of STAT5 we set $in(t) = x_3(t)$, $out(t) = x_3(t-\tau)$, and $N = 8$.

Now, all model ingredients are available for fitting the dataset shown in Fig. 3. But due to the experimental design outlined in [42] it is known that all state variables except x_1 are initially zero, these values are therefore kept fixed throughout the optimisation. In addition, it is not possible to estimate the scaling parameters s_1, s_2 from a single experiment but in a multi-experiment fashion. It turns out that the scaling is $s_1 = 0.33$ and $s_2 = 0.26$ for the displayed dataset. The remaining parameters as well as the initial

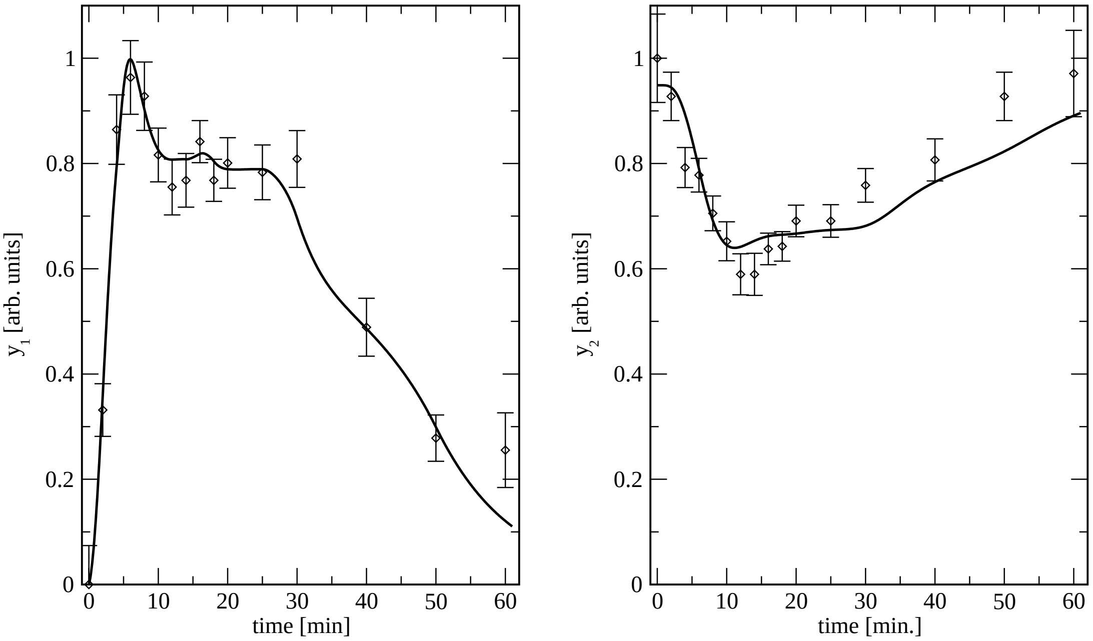

Figure 3: Total activated STAT5, y_1, and total STAT5, y_2, in the cytoplasm of the cell. The trajectory of the best fit is indicated by the solid line.

value of x_1 can then be estimated form a single dataset. They turns out to be $k_1 = (2.12 \pm 0.22)$ min^{-1}mol^{-1}, $k_2 = (0.109 \pm 0.015)$ min^{-1}mol^{-1}, $\tau = (5.2 \pm 0.6)$ min and $x_1(0) = (3.71 \pm 0.07)$ mol. The corresponding trajectory is displayed in Fig. 3. Moreover, the fitted model yields a good description of the data.

6 Summary

The parameter estimation procedure for ordinary differential equations, multiple shooting, is reviewed and detailedly described. In contrast to other attempts of estimating parameters in differential equations, this procedure does not suffer heavily from the attraction to local minima and the speed of convergence is considerably higher than global optimisation methods can achieve. Besides the general idea of embedding the problem into a higher dimensional parameter space, the speed of convergence as well as the stability can only be achieved by sophisticated numerical methods. Especially the condensation algorithm and the damping strategy can be considered as landmarks of this issue. These aspects are thoroughly explained within the remaining issues of the method, such as integration of the ODE, minimisation and the statistical analysis of the estimates. Identifiability of the parameters can be regarded as central assumption for a successful operation of most of the numerical components. A regularisation procedure to weaken this assumption is included to the discussion of multiple shooting. The regularisation can further help to remove all unidentifiable parameters. Towards the end of this review we appended two examples to demonstrate the performance of multiple shooting.

Moreover, the extension of multiple shooting to partial differential equations is also possible, [32, 33]. Additionally, the method can also be used to find an optimal experimental design, see e.g. [28, 13]. This broad applicability of the multiple shooting method marks the relevance of such a tool for a vast range of applied sciences and engineering. The method can also yield a valuable contibution for validating the model, e.g. needed in discrete event simulations. We hope that this review allows an easy re-implementation of the presented ideas, and therefore propagating the availability of the method to a larger community.

Acknowledgements

M. Peifer received financial support from the Deutsche Forschungsgemeinschaft (DFG).

References

[1] M. Ali, C. Storey, and A. Törn. Applications of stochastic global optimization algorithms to practical problems. *J. Optim. Theo. & Appl.*, 95:545–563, 1997.

[2] J.I. Ardenghi, M.C. Maciel, and A.B. Verdiell. A trust-region approach for solving a parameters estimation problem from the biotechnology area. *Appl. Num. Math.*, 47:281–294, 2003.

[3] J.R. Banga, C.G. Moles, and A.A. Alonso. Global optimization of bioprocesses using stochastic and hybrid methods. In *Frontiers in Global Optimization.* Kluwer, 2003.

[4] J.R. Banga and W.D. Seider. Global optimization of chemical processes using stochastic algorithms. In C.A. Floudas and P.M. Pardalos, editors, *State of the Art in Global Optimization*, pages 563–583. Kluwer, Dordrecht, 1996.

[5] C. Bauer, A. Frink, and R. Kreckel. Introduction to the GiNaC framework for symbolic computation within the C++ programming language. *J. Symbolic Com.*, 33:1–12, 2002.

[6] A. Bellen and M. Zennaro. *Numerical Methods for Delay Differential Equations.* Oxford Science Publications, 2003.

[7] C.H. Bischof, A. Carle, G.F. Corliss, A. Griewank, and P. Hovland. ADIFOR: Generating derivative code from Fortran programs. *Scientific Programming*, 1:11–29., 1992.

[8] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In Ebert K., Deuflhard P., and Jäger W., editors, *Modelling of Chemical Reaction Systems*, pages 102–125. Springer, 1981.

[9] H.G. Bock. Recent advances in parameter identification techniques for ordinary differential equations. In Deuflhard P. and Hairer E., editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, pages 95–121. Birkhäuser, 1983.

[10] H.G. Bock. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen.* PhD thesis, Universität Bonn, 1987.

[11] K.E. Brenan, S.L. Campell, and L.R. Petzold. *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations.* North-Holland, New York, 1989.

[12] L. Denis-Vidal, G. Joly-Blanchard, and C. Noiret. Some effective approaches to check the identifiability of uncontrolled nonlinear systems. *Math. and Computers in Sim.*, 57:35–44, 2001.

[13] D. Faller, U. Klingmüller, and J. Timmer. Simulation methods for optimal experimental design in systems biology. *Simulation*, 79:717–725, 2003.

[14] K.R. Godfray and J.J. DiStefano. Identifiability of model parameters. In *Identification and system parameter estimation*, pages 89–114. Pergamon Press, 1985.

[15] A. Griewank, D. Juedes, and J. Utke. Alogorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Trans. on Math. Software*, 22:131–167, 1996.

[16] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I, Nonstiff Problems.* Springer, Berlin, 1993.

[17] R.J. Hanson and K.H. Haskell. Alogorithm 587: Two algorithms for the linearly constrained least squares problem. *ACM Trans. on Math. Software*, 8:323–333, 1982.

[18] J.H. Holland. *Adaption in natural and artificial systems.* MIT Press, Cambridge, MA, 1992.

[19] W. Horbelt. *Maximum likelihood estimation in dynamical systems.* PhD thesis, University of Freiburg, 2001.

[20] W. Horbelt, J. Timmer, M. Bünner, R. Meucci, and M. Ciofini. Identifying physically properties of a CO_2 laser by dynamical modeling of measured time series. *Phys. Rev. E*, 64:016222, 2001.

[21] W. Horbelt, J. Timmer, and H.U. Voss. Parameter estimation in nonlinear delayed feedback systems from noisy data. *Phys. Lett. A*, 299:513–521, 2002.

[22] C.T. Kelly. *Iterative Methods for Optimization.* SIAM, 1999.

[23] H. Kuhn and A. Tucker. Nonlinear programming. In *Proceedings of 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, pages 481–492. University of California Press, 1951.

[24] J.R. Leis and M.A. Kramer. ODESSA-An ordinary differential equation solver with explicit simultaneous sensitivity analysis. *ACM Trans. on Math. Software*, 14:61–67, 1988.

[25] J.R. Leis and M.A. Kramer. The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations. *ACM Trans. on Math. Software*, 14:45–60, 1988.

[26] K.A. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[27] L. Ljung and T. Glad. On global identifiability for arbitrary model parameterization. *Automatica*, 30:265–276, 1994.

[28] T. Lohmann, H.G. Bock, and J.P. Schlöder. Numerical methods for parameter estimation and optimal experimental design in chemical reaction systems. *Industial and Engineering Chemistry Research*, (31):54–57, 1992.

[29] E.N. Lorenz. Deterministic aperiodic flow. *J. Atmos. Sci.*, 20:130, 1963.

[30] D.W. Marquardt. An algorithm for least-squares-estimation of nonlinear problems in least squares. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[31] J. Milstein. Fitting multiple trajectories simultaneously to a model of inducible enzyme synthesis. *Math. Biosciences*, 40:175–184, 1978.

[32] T.G. Müller and J. Timmer. Fitting parameters in partial differential equations from partially observed noisy data. *Physica D*, 171:1–7, 2002.

[33] T.G. Müller and J. Timmer. Parameter identification techniques for partial differential equations. *Int. J. Bif. Chaos*, 14:2053–2060, 2004.

[34] N. Noykova, T.G. Müller, M. Gyllenberg, and J. Timmer. Quantitative analysis of anaerobic wastewater treatment processes: Identifiability and parameter estimation. *Biotech. and Bioengineering*, 78:89–103, 2002.

[35] W.H. Press, B.P. Flannery, S.A. Saul, and W.T. Vetterling. *Numerical Recipes.* Cambridge University Press, Cambridge, 1992.

[36] O. Richter, P. Nörtersheuser, and W. Pestemer. Non-linear parameter estimation in pesticide degradation. *The Science of the Total Environment*, (123/124):435–450, 1992.

[37] A.H.G. Rinnooy-Kan and G.T. Timmer. Stochastic global optimization methods. Part I: Clustering methods. *Math. Prog.*, 39:27–56, 1987.

[38] K. Schittkowski. Parameter estimation in systems of nonlinear equations. *Num. Math.*, 68:129–142, 1995.

[39] J. Stoer. On the numerical solution of constrained least squares. *SIAM J. Numer. Anal.*, 382:282–411, 1971.

[40] J. Stoer and R. Bulirsch. *Introduction to numerical analysis.* Springer, 1993.

[41] A.D. Stribet, P. Rosenau, A.C. Ströder, and R.J. Strasser. Parameter optimisation of fast chlorophyll fluorescence induction model. *Math. & Computers in Sim.*, 56:443–450, 2001.

[42] I. Swameye, T.G. Müller, J. Timmer, O. Sandra, and U. Klingmüller. Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by data-based modeling. *Proc. Natl. Acad. Sci.*, (1000):1028–1033, 2003.

[43] A. Törn, M. Ali, and Viitanen S. Stochastic global optimization: Problem classes and solution techniques. *J. Global Optimization*, 14:437, 1999.

[44] A.W. van der Vaart. *Asymptotic Statistics.* Cambridge University Press, 1998.

[45] H.H. von Grünberg, M. Peifer, J. Timmer, and M. Kollmann. Variations in substitution rate in human and mouse genomes. *Phys. Rev. Lett.*, 93, 2004.

[46] H.U. Voss, M. Peifer, W. Horbelt, H. Rust, and J. Timmer. Identification of chaotic systems from experimental data. In G. Gousebet, editor, *Chaos and its Reconstruction*, pages 245–286. Nova Science Publishers Inc., New York, 2003.

[47] Z.B. Zabinsky and R.L. Smith. Pure adaptive search in global optimization. *Math. Prog.*, 53:323–338, 1992.