Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting

M. Peifer and J. Timmer

Abstract: In silico investigations by simulating dynamical models of biochemical processes play an important role in systems biology. 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. Almost all approaches for estimating parameters in ordinary differential equations have either a small convergence region or suffer from an 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, the literature regarding the practical implementation and providing some theoretical background is rarely available. All necessary information for a successful implementation is supplied here and the basic facts of the involved numerics are discussed. To show the performance of the method, two illustrative examples are discussed.

1 Introduction

The central idea of systems biology to learn about biological systems by the analysis of mathematical models of these systems is hampered by the fact that parameters like rate constants are not known. The challenging problem of estimating parameters in ordinary differential equations (ODEs) from partially observed noisy data appears therefore in systems biology. 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 into global or local optimisation procedures. Methods based on global minimisation routines are for example random search and adaptive stochastic methods [1–4], clustering methods [5], evolutionary computation [6] and simulated annealing. A detailed discussion of these methods with respect to parameter identification in ODEs is given in the work of Banga *et al.* [7]. 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, local optimisation procedures such as sequential quadratic programming (SQP), Newton methods, quasi-Newton methods and so on are computationally efficient, but they tend to converge to local minima. In the 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 the fact 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 earlier further suggests that there is a trade-off between computational efficiency and stability for estimating parameters in ODEs. In comparison to the initial value approach, multiple shooting provides enhanced stability with only a slight increase of the computational cost. The method was introduced by Stoer and Bulirsch in the early seventies [8] and was substantially enhanced and mathematically analysed by Bock [9–11]. Here, some of the well-elaborated mathematical details are presented, but always in scope of practically implementing these ideas. Keeping track on the algorithmic issues can be regarded as the major intension of this article. Since this aspect is neglected in the literature so far, the accessibility and re-implementation of multiple shooting is currently limited.

2 Estimation problem

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

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), t, p) \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{1}$$

The right-hand side of the ODE depends on some parameters $\boldsymbol{p} \in \mathbb{R}^{n_p}$. It is further assumed that f is continuously differentiable with respect to state $\boldsymbol{x}$ and parameters $\boldsymbol{p}$. Let Y_{ij} denote the data of measurement $i = 1, \ldots, n$ and of observable $j = 1, \ldots,$ obs, whereas n represents the total amount of data and obs is the number of observables. Moreover, data Y_{ij} satisfy the following observation equation

$$Y_{ij} = g_j(t_i, \boldsymbol{p}) + \sigma_{ij}\epsilon_{ij} \qquad j = 1, \ldots, \text{obs} \tag{2}$$

for some observation function $\boldsymbol{g}$: $\mathbb{R}^d \to \mathbb{R}^{\text{obs}}$, $d \geq$ obs, $\sigma_{ij} > 0$, and ϵ_{ij}s are independent and standard Gaussian distributed random variables. Sample points t_i are ordered such that $t_0 \leq t_1 < \cdots < t_n < t_f$ and observation function $\boldsymbol{g}(\cdot)$ is

doi:10.1049/iet-syb:20060067

Paper first received 24th April and in revised form 25th September 2006

The authors are with Freiburg Centre for Data Analysis and Modelling, Eckerstr. 1, Freiburg 79104, Germany

E-mail: peifer@fdm.uni-freiburg.de

again continuously differentiable in both variables. The generalisation of (2) to more than one experiment, possibly under different experimental conditions, reads

$$Y_{ijk} = g_j(x(t_{ij}), \boldsymbol{p}) + \sigma_{ijk}\epsilon_{ijk} \qquad k = 1, \ldots, n_{\text{exp}} \tag{3}$$

where n_{exp} is the number of experiments performed. Certain parameters may be different for each experiment, but the treatment of these local parameters and different experiments requires only minor modifications of the described procedures and therefore only the one-experiment design $n_{\text{exp}} = 1$ is considered.

On the basis of measurements Y_{ij}, the task is now to estimate the initial state $\boldsymbol{x}_0$ and parameters $\boldsymbol{p}$. The principle of maximum-likelihood [12] yields an appropriate cost function which has to be minimised with respect to parameters $\boldsymbol{x}_0$ and $\boldsymbol{p}$. Defining $\boldsymbol{x}(t_i; \boldsymbol{x}_0, \boldsymbol{p})$ as being the trajectory at time t_i, the cost function is then given by

$$\mathcal{L}(\boldsymbol{x}_0, \boldsymbol{p}) = \sum_{i=1}^{n} \sum_{j=1}^{\text{obs}} \frac{(Y_{ij} - g_j(x(t_i; \boldsymbol{x}_0, \boldsymbol{p}), \boldsymbol{p}))^2}{2\sigma_{ij}^2} \tag{4}$$

A direct minimisation of $\mathcal{L}$ with respect to $\boldsymbol{x}_0, \boldsymbol{p}$ leads to the so-called initial value approach.

2.1 Initial value approach

The development of the initial value approach has a long history [7, 13–16]. Again, one can distinguish between local and global optimisation methods. If global optimisation procedures are used for minimising the likelihood, (4), the computational cost is rather high. On the other hand, local optimisation algorithms have 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 nonlinear 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 performed by Bock [9–11]. Besides the example given in Section 5, some applications of the method to measured data are for example the works of Richter *et al.* [17], Timmer *et al.* [18], Stribet *et al.* [19], Horbelt *et al.* [20] and von Grünberg *et al.* [21].

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 time interval $I = [t_0, t_f]$ 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 $(\boldsymbol{x}_0^k)_{k=1,\ldots,n_{ms}}$ but sharing the same parameters $\boldsymbol{p}$. The only difference in cost function (4) is that trajectory $x(t_i; \boldsymbol{x}_0, \boldsymbol{p})$ is replaced by the interval dependent trajectory $x(t_i; \boldsymbol{x}_0^k, \boldsymbol{p})$ for all $k = 1, \ldots, n_{ms}$. 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, (1). To enforce smoothness of the final trajectory, the optimisation is constrained such that all discontinuities are eventually removed which therefore leads to a constrained nonlinear 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 $k = 1, \ldots, n_{ms}$ let $t_k^+ = \max\{I_k\}$, $t_k^- = \min\{I_k\}$ and $\boldsymbol{\theta}_k = (\boldsymbol{x}_0^k, \boldsymbol{p})$. The optimisation problem can then be formulated in the following manner

$$\mathcal{L}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{n_{ms}}) = \frac{1}{2} \sum_{j=1}^{\text{obs}} \sum_{k=1}^{n_{ms}} \sum_{\{i:t_i \in I_k\}} (R_{ijk}^a(\boldsymbol{\theta}_k))^2 = \min_{\theta_1, \ldots, \theta_{n_{ms}}}$$

subject to

$$\begin{aligned} &\boldsymbol{x}(t_i^+; \boldsymbol{\theta}_i) - \boldsymbol{x}(t_{i+1}^-; \boldsymbol{\theta}_{i+1}) = 0 && i = 1, \ldots, n_{ms} - 1 \\ &R_j^e(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{n_{ms}}) = 0 && j = 1, \ldots, n_e \\ &R_k^g(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{n_{ms}}) \geq 0 && k = 1, \ldots, n_g \end{aligned} \tag{5}$$

where the continuity constraints are given at the first row of the constraints-part followed by some optional constraints R_j^e, R_k^g, to include for example conservation laws or parameter bounds. Cost function $\mathcal{L}(\theta_1, \ldots, \theta_{n_{ms}})$ is equivalent to (4) if the continuity constraints are satisfied; hence

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

This nonlinear 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 $\boldsymbol{\theta}^0 = (\boldsymbol{\theta}_1^0, \ldots, \boldsymbol{\theta}_{n_{ms}}^0)$. All contributions depending on the second derivative of R_{ijk}^a are neglected afterwards. This is possible because these contributions to the Hessian of $\mathcal{L}$ are vanishing asymptotically, $n \to \infty$, if the model assumptions are correct [11, 22]. From the quadratic approximation an update step for lth iteration $\Delta\boldsymbol{\theta}^l = (\Delta\boldsymbol{\theta}_1^l, \ldots, \Delta\boldsymbol{\theta}_{n_{ms}}^l)$ can be calculated by solving the linear programming problem

$$\frac{1}{2} \sum_{j=1}^{\text{obs}} \sum_{k=1}^{n_{ms}} \sum_{\{i:t_i \in I_k\}} (R_{ijk}^a(\boldsymbol{\theta}_k^l) + \mathrm{d}_\theta R_{ijk}^a(\boldsymbol{\theta}_k^l)\Delta\boldsymbol{\theta}^l)^2 = \min_{\Delta\boldsymbol{\theta}^l}$$

subject to

$$\begin{aligned} &\boldsymbol{x}(t_i^+; \boldsymbol{\theta}_i^l) - \boldsymbol{x}(t_{i+1}^-; \boldsymbol{\theta}_{i+1}^l) + \mathrm{d}_{\theta_i}\boldsymbol{x}(t_i^+; \boldsymbol{\theta}_i^l)\Delta\boldsymbol{\theta}_i^l \\ &\qquad - \mathrm{d}_{\theta_{i+1}}\boldsymbol{x}(t_{i+1}^-; \boldsymbol{\theta}_{i+1}^l)\Delta\boldsymbol{\theta}_{i+1}^l = 0 \\ &R_j^e(\boldsymbol{\theta}^l) + \mathrm{d}_\theta R_j^e(\boldsymbol{\theta}^l)\,\Delta\boldsymbol{\theta}^l = 0 \\ &R_k^g(\boldsymbol{\theta}^l) + \mathrm{d}_\theta R_k^g(\boldsymbol{\theta}^l)\,\Delta\boldsymbol{\theta}^l \geq 0 \end{aligned} \tag{6}$$

where d_θ denotes the derivative with respect to parameters $\boldsymbol{\theta}$ of the corresponding function. Setting $\boldsymbol{\theta}^{l+1} = \boldsymbol{\theta}^l + \Delta\boldsymbol{\theta}^l$, $l = 1, \ldots,$ and iterating (6) until $\Delta\boldsymbol{\theta}^l \simeq 0$ yields a minimum of (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, nonlinear optimisation problems [11, 23]. In Section 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 (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. This allows discontinuous trajectories during the optimisation process, reducing the problem of local minima.

Properties 1 and 2 are yielding the desired speed of convergence whereas 3 is mainly responsible for the stability of multiple-shooting. This is gained by the possibility that the algorithm can circumvent local minima by allowing for discontinuous trajectories while searching the minimum. The main disadvantage results from the linearisation of the cost function. It can easily happen that despite the update step $\Delta\boldsymbol{\theta}^l$ pointing in the direction of decreasing $\mathcal{L}$, the proposed step is too large. Such an overshooting is common to any simple optimisation procedure 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 $\boldsymbol{\theta}^{l+1} = \boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l$ 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 the following section.

3 Detailed description of multiple shooting

In the previous section, the basic idea and some aspects of the performance of multiple shooting were 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 for example 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, linearised problem (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 one decides whether the procedure is converged or a further iteration has to be taken into account by integrating a new trail trajectory, applying a convergence criterion, such as $||\Delta\boldsymbol{\theta}^l|| \simeq 0$. After convergence, 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.

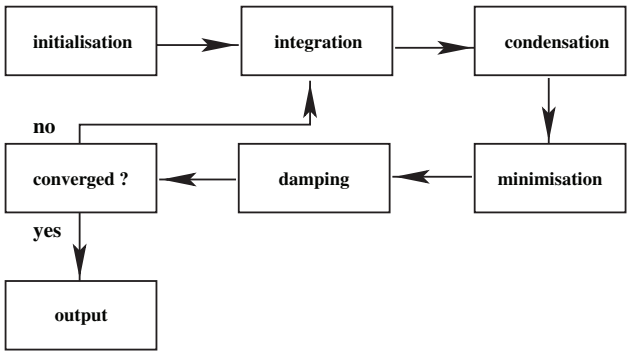

Fig. 1 *Main program flow*
All stages of the algorithm described in Section 3 are displayed

3.1 Integration

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

1. Non-stiff ODEs,
2. Stiff problems,
3. Delay differential equations (DDEs), and
4. Differential algebraic equations (DAEs).

For non-stiff ODEs, standard numerical integrators such as the Runge–Kutta method [22] 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 by orders of magnitude, only stiff integrators are useful. Especially, in the case of multiple shooting, we propose to use ODESSA [24, 25], 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 (6) and will be discussed later. DDEs cannot be represented by (1). Although DDEs are not ODEs, it is possible to adapt multiple shooting to this class of differential equations [26–28]. 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 for example RETARD [29], for a deeper discussion of DDE we refer to the work of Bellen and Zennaro [30]. 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. Sometimes, this kind of separation is not possible such that special DAE integrators have to be considered [31].

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

$$\frac{\partial \boldsymbol{x}(t; \boldsymbol{\theta}_k)}{\partial x_0^{k(i)}} \quad \text{and} \quad \frac{\partial \boldsymbol{x}(t; \boldsymbol{\theta}_k)}{\partial p^{(j)}} \quad t \in I_k,$$

$$k = 1, \ldots, n_{ms}, \quad i = 1, \ldots, d, \quad j = 1, \ldots, n_p$$

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

1. Finite differences, called external differentiation [10, 11],
2. Differentiation of the integration scheme, called internal differentiation [10, 11, 29], and
3. The simultaneous solution of the sensitivity equations [26].

The approximation of the derivatives by finite differences such as

$$\frac{\partial \boldsymbol{x}(t; \boldsymbol{\theta}_k)}{\partial x_0^{k(i)}} \simeq h^{-1}(\boldsymbol{x}(t; \boldsymbol{\theta}_k + \boldsymbol{e}_{i,x_0} h) - x(t; \boldsymbol{\theta}_k))$$

for some $h \ll 1$ and $\boldsymbol{e}_{i,x_0}$ being the ith unit vector with respect to the initial value leads to numerical difficulties.

Due to the numerical integration, trajectory $\boldsymbol{x}(t;\ \boldsymbol{\theta}_k)$ 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 $\boldsymbol{x}(t;\ \boldsymbol{\theta}_k)$ reveals that the optimal choice is

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

see for example the work of Kelly [32]. Unfortunately, the constant of proportionality in (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 unknown parameter h and the high computational cost.

Differentiating the integration scheme is considerably faster than external differentiation [10, 11] 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 $\boldsymbol{x}(t;\ \boldsymbol{x}_0,\ \boldsymbol{p}) = \boldsymbol{x}(t;\ \boldsymbol{\theta})$ of (1) and derivative d_θ, where the subscript indicates the variables to be differentiated. The time evolution of the sensitivities $\boldsymbol{S}(t;\ \boldsymbol{\theta}) = \mathrm{d}_\theta x(t;\ \boldsymbol{\theta})$ is then given by solving

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{S}(t;\ \boldsymbol{\theta}) = (\mathrm{d}_\theta f)(\boldsymbol{x}(t;\ \boldsymbol{\theta}),\ t,\ \boldsymbol{p}) + (\mathrm{d}_x f)(\boldsymbol{x}(t;\ \boldsymbol{\theta}),\ t,\ \boldsymbol{p})\boldsymbol{S}(t;\ \boldsymbol{\theta})$$

$$\boldsymbol{S}_0 = \boldsymbol{S}(t_0;\ \boldsymbol{\theta}) = (\mathbf{1}_{d\times d},\ \mathbf{0}_{d\times n_p}) \tag{8}$$

where $\mathbf{1}_{d\times d}$ is the $d \times d$-unity matrix, $\mathbf{0}_{d\times n_p}$ the $d \times n_p$-matrix of zeroes, and f is the right-hand side of the ODE as introduced in (1). Simultaneously integrating (1) and (8) yields the trajectory as well as the desired sensitivities. It is further sufficient to restrict the step size control to the main ODE, (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 simultaneous solution of the sensitivity equations (8).

The procedure requires the calculation of derivatives like $\mathrm{d}_p f$, $\mathrm{d}_x f$ and so on. Calculating such derivatives by hand can be very time consuming and error-prone for big systems. Therefore automatic differentiation should be applied. One possibility is to generate the derivatives at runtime by using program packages like ADIFOR or ADOLC [33, 34]. Since the derivatives have to be recalculated for every function evaluation, this approach slows down the method significantly. The calculation of the Jacobians should therefore processed before the program is executed which can be realised by using symbolic computation software, for example GinNaC [35].

3.2 Condensation

All information is now available for setting up (6). Suppose that $\boldsymbol{h}_i = \boldsymbol{x}(t_i^+) - \boldsymbol{x}(t_{i+1}^-)$, $\Delta\boldsymbol{\theta}_i^l = (\Delta\boldsymbol{x}_0^i,\ \Delta\boldsymbol{p})$ for all $i = 1, \ldots,\ n_{ms}-1$ and because of (8), $\mathrm{d}_{x_0^i}^{i+1}\boldsymbol{x}(t_{i+1}^-) = \mathbf{1}$, $\mathrm{d}_p\boldsymbol{x}(t_{i+1}^-) = \mathbf{0}$ then the continuity constraints can be written as

$$\boldsymbol{h}_i + \mathrm{d}_{x_0^i}\boldsymbol{x}(t_i^+)\Delta\boldsymbol{x}_0^i\ +\ \mathrm{d}_p\boldsymbol{x}(t_i^+)$$
$$\Delta\boldsymbol{p} = \Delta\boldsymbol{x}_0^{i+1} \quad i = 1, \ldots, n_{ms}-1 \tag{9}$$

According to (9), all initial value update steps at the multiple shooting intervals can therefore be related to $\Delta\boldsymbol{x}_0^1$ by backward elimination. Inserting the increments $\Delta\boldsymbol{x}_0^2, \ldots,$ $\Delta\boldsymbol{x}_0^{n_{ms}}$ obtained by (9) into (6) yields a system to be solved only for $\Delta\boldsymbol{x}_0^1$ and $\Delta\boldsymbol{p}$. Let $\boldsymbol{R}^a$ be the $n \cdot n_{ms} \cdot n_{\text{obs}}$-dimensional vector with components R_{ijk}^a and $\boldsymbol{R}^e$, $\boldsymbol{R}^g$, respectively the condensed problem is thus

$$||\boldsymbol{u}_1^a + \boldsymbol{E}_1^a\,\Delta\boldsymbol{x}_0^1 + \boldsymbol{P}_1^a\,\Delta\boldsymbol{p}||^2 = \min_{\Delta\boldsymbol{x}_0^1,\Delta\boldsymbol{p}}$$

subject to

$$\begin{aligned}\boldsymbol{u}_1^e + \boldsymbol{E}_1^e\,\Delta\boldsymbol{x}_0^1 + \boldsymbol{P}_1^e\,\Delta\boldsymbol{p} &= 0\\ \boldsymbol{u}_1^g + \boldsymbol{E}_1^g\,\Delta\boldsymbol{x}_0^1 + \boldsymbol{P}_1^g\,\Delta\boldsymbol{p} &\geq 0\end{aligned} \tag{10}$$

where $u_1^{a/e/g}$ and matrices $\boldsymbol{E}_1^{a/e/g}$, $\boldsymbol{P}_1^{a/e/g}$ are determined by the recursion [10, 11]

$$\text{Initialisation:}\ \boldsymbol{u}_{n_{ms}}^{a/e/g} = \boldsymbol{R}^{a/e/g},\quad \boldsymbol{E}_{n_{ms}}^{a/e/g} = \mathrm{d}_{x_0^{n_{ms}}}\boldsymbol{R}^{a/e/g},$$
$$\boldsymbol{P}_{n_{ms}}^{a/e/g} = \mathrm{d}_p\boldsymbol{R}^{a/e/g}$$
$$\text{For } i = n_{ms}, \ldots, 2\text{:}\ \boldsymbol{u}_{i-1}^{a/e/g} = \boldsymbol{u}_i^{a/e/g} + \boldsymbol{E}_i^{a/e/g}\,\boldsymbol{h}_{i-1}$$
$$\boldsymbol{E}_{i-1}^{a/e/g} = d_{x_0^{i-1}}\boldsymbol{R}^{a/e/g} + \boldsymbol{E}_i^{a/e/g}\mathrm{d}_{x_0^{i-1}}\boldsymbol{h}_{i-1}$$
$$\boldsymbol{P}_{i-1}^{a/e/g} = \boldsymbol{P}_i^{a/e/g} + \boldsymbol{E}_i^{a/e/g}\,\mathrm{d}_p\boldsymbol{h}_{i-1} \tag{11}$$

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

3.3 Minimisation

The solution of the linear programming problem (10) can be obtained by calculating the generalised inverse $G(\boldsymbol{\theta}^l)$ at $\boldsymbol{\theta}^l$. Since the condensation procedure removes the continuity constraints by partially calculating the generalised inverse using the transformation given earlier, we concentrate on uncondensed problem (6) in the following. The general inverse then solves

$$-\mathrm{d}_\theta\boldsymbol{R}^a(\boldsymbol{\theta}^l)\Delta\boldsymbol{\theta}^l = \boldsymbol{R}^a(\boldsymbol{\theta}^l) \tag{12}$$

subject to all equality and inequality constraints of (6), where $\boldsymbol{R}^a$ is again the $n \cdot n_{ms} \cdot n_{\text{obs}}$-dimensional vector of the actual residuals. Therefore $\Delta\boldsymbol{\theta}^1 = G(\boldsymbol{\theta}^1)R^{\text{a}}(\boldsymbol{\theta}^l)$ and by multiplying the system to solve (12) with $G(\boldsymbol{\theta}^l)$, we obtain $-G(\boldsymbol{\theta}^l)\mathrm{d}_\theta\boldsymbol{R}^a(\boldsymbol{\theta}^l) = \mathbf{1}$. Note that since (12) is overdetermined, the solution as constructed earlier only yields the minimum quadratic norm solution, as desired. Moreover, the equality and violated inequality constraints are handled by projections onto the resulting sub-manifold using Lagrange multipliers.

In practice, any appropriate minimisation algorithm for solving constrained linear optimisation problems, for example the routine E04NCF from the NAG library, LSEI [36] or the method of Stoer [37], can be used.

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 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

$$T(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) + \underbrace{\sum_{i=1}^{n_{ms}+n_e-1} \alpha_i |R_i^e(\boldsymbol{\theta})|}_{\text{equality constr.}} + \underbrace{\sum_{i=1}^{n_g} \beta_i |\min\{0, R_i^g(\boldsymbol{\theta})\}|}_{\text{inequality constr.}} \tag{13}$$

where α_i and β_i are bounded below by their corresponding Lagrange multipliers. Based on this level function, a downhill procedure can always be constructed by some one-dimensional line-search algorithm. According to Bock [9–11], it turns out that the performance of using $T(\boldsymbol{\theta})$ is rather bad. This inefficiency is because

1. Line-search has a high computational cost since a new trajectory has to be integrated for each evaluation of (13) 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, Bock [9–11] proposed to replace the line-search by some predictor-corrector method and the level function is changed to include the local geometry. As prototype for constructing such a level function, we consider the following ideal level function

$$T_{N,\theta^*}(\boldsymbol{\theta}) = ||G(\boldsymbol{\theta}^*)\boldsymbol{R}^a(\boldsymbol{\theta})||^2 \tag{14}$$

where $\boldsymbol{\theta}^*$ is minimum of the cost function $\mathcal{L}$, G is the generalised inverse as defined in Section 3.3 and $\boldsymbol{R}^a$ the vector of residuals at the corresponding point in parameter space. Expanding $\boldsymbol{R}^a(\boldsymbol{\theta})$ about $\boldsymbol{\theta}^*$ up to first order and substituting the obtained expression into (14) yields $T_{N,\theta^*}(\boldsymbol{\theta}) = \|G(\boldsymbol{\theta}^*)(\boldsymbol{R}^a(\boldsymbol{\theta}^*) + \mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*) + \mathcal{O}(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|^2))\|^2$. Since we assume that the Kuhn–Tucker conditions are fulfilled, as described in Section 2.2, $G(\boldsymbol{\theta}^*)\boldsymbol{R}^a(\boldsymbol{\theta}^*) = 0$ and by the properties of the generalised inverse $-G(\boldsymbol{\theta}^l)\mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l) = \mathbf{1}$, according to Section 3.3, we obtain

$$T_{N,\theta^*}(\boldsymbol{\theta}) = ||\boldsymbol{\theta} - \boldsymbol{\theta}^*||^2 + O(||\boldsymbol{\theta} - \boldsymbol{\theta}^*||^3)$$

In conclusion, ideal level function T_{N,θ^*} measures the squared Euclidean distance to the optimum up to third order. Therefore T_{N,θ^*} is in vicinity of $\boldsymbol{\theta}^*$, a distance measure in a Euclidean space which does not depend on application specific geometric properties of the parameter 'landscape'. Moreover, it shares the same monotony properties of the cost function close to the global minimum, as desired. Unfortunately, the knowledge of the minimum $\boldsymbol{\theta}^*$ is needed for constructing T_{N,θ^*}. In order to obtain an applicable level function which has similar properies as T_{N,θ^*}, we replace $\boldsymbol{\theta}^*$ by $\boldsymbol{\theta}^l$. The resulting level function

$$T_N^l(\boldsymbol{\theta}) = ||G(\boldsymbol{\theta}^l)\boldsymbol{R}^a(\boldsymbol{\theta})||^2 \tag{15}$$

is called natural level function which provides an efficient criterion for determining the relaxation coefficient λ^l for the lth iteration.

Again, finding an appropriate λ^l for which the minimisation scheme is descendant involves some kind of line-search to guarantee that $T_N(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l) < T_N(\boldsymbol{\theta}^l)$ 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. To prevent line-search, an upper bound for the level function evaluated at the relaxed update step $T_N(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l)$ can be derived, as shown in the work of Bock [11] and in Section 9.1. This bound reads

$$T_N^l(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l) \leq \left(1 - \lambda^l + \frac{\lambda^{l^2}}{2}\omega(\boldsymbol{\theta}^l, \lambda^l)\right)^2 T_N^l(\boldsymbol{\theta}^l) \tag{16}$$

where the function ω is given by

$$\omega(\boldsymbol{\theta}^l, \lambda^l) = \sup_{s\in(0,\lambda^l]} \left\{ \frac{||G(\boldsymbol{\theta}^l)(\mathrm{d}_\theta R(\boldsymbol{\theta}^l + s\Delta\boldsymbol{\theta}^l) - \mathrm{d}_\theta R(\boldsymbol{\theta}^l))\Delta\boldsymbol{\theta}^l||}{s||\Delta\boldsymbol{\theta}^l||^2} \right\} \tag{17}$$

Now, for some arbitrarily chosen $\eta \in (0, 2]$, every $\lambda^l \in (0, \lambda^*]$ yields a descending step $T_N(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l) < T_N(\boldsymbol{\theta}^l)$, where λ^* can be obtained from the solution of

$$\lambda^* = \min\left\{1, \frac{\eta}{\omega(\boldsymbol{\theta}^l, \lambda^*)\,||\Delta\boldsymbol{\theta}^l||}\right\} \tag{18}$$

This is because $\lambda^l \leq \lambda^* \leq \eta(\omega(\boldsymbol{\theta}^l,\lambda^*)||\Delta\boldsymbol{\theta}^l||)^{-1}$ and by using (16)

$$\begin{aligned} T_N^l(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l) &\leq \left(1 - \lambda^l + \frac{\lambda^l}{2}\eta\right)^2 T_N^l(\boldsymbol{\theta}^l) \\ &= (1 - \lambda^l(1 - \eta/2))^2 T_N^l(\boldsymbol{\theta}^l) < T_N^l(\boldsymbol{\theta}^l) \end{aligned}$$

For a given $\eta \in (0, 2]$, the maximal relaxation parameter leading to a descendant step is therefore λ^*. 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. This requires the local identifiability of all parameters and the boundedness of the second derivative $\mathrm{d}_\theta^2 \boldsymbol{R}^a$ in the vicinity of the minimum, as shown in Section 9.2.

Since $\omega(\boldsymbol{\theta}^l, \lambda^l)$ is a-priori not known, a suitable estimation or approximation is necessary. Demanding the coincidence of the estimator with (17) in the limit $\lambda^l \to 0$ automatically guarantees an appropriate relaxation scheme whenever a massive damping is needed. The estimator

$$\hat{\omega}(\boldsymbol{\theta}^l, \lambda^l) = 2\frac{||G(\boldsymbol{\theta}^l)R(\boldsymbol{\theta}^l + \lambda^l\Delta\boldsymbol{\theta}) - (1 - \lambda^l)\Delta\boldsymbol{\theta}^l||}{||\lambda^l\Delta\boldsymbol{\theta}^l||^2} \tag{19}$$

satisfies this desired property [11]. Replacing ω with $\hat{\omega}$ in (18), a predictor-corrector procedure can be constructed to find a suitable $0 < \lambda^l \leq \lambda^*$. Assuming that $\hat{\omega}(\boldsymbol{\theta}^{l-1}, \lambda^{l-1})$ from the previous Gauss–Newton iteration is approximately constant the damping parameter for the actual iteration can be determined by

$$\lambda^l = \min\left\{1, \frac{\eta_0}{\hat{\omega}(\boldsymbol{\theta}^{l-1}, \lambda^{l-1})||\Delta\boldsymbol{\theta}^l||}\right\} \tag{20}$$

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 (19) but now using λ^l, given in (20). 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 described by the following algorithm:

1. Set $j = 0$ and calculate the predictor $\mu_0 = \eta_0/(\hat{\omega}(\boldsymbol{\theta}^{l-1}, G, \lambda^{l-1})||\Delta\boldsymbol{\theta}^l||)$.
2. The predicted relaxation step is then given by

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

3. If $\hat{\omega}(\boldsymbol{\theta}^l, \lambda_j^{\text{pred}})||\Delta\boldsymbol{\theta}^l||\lambda_j^{\text{pred}} \leq \eta_2$, then the proposed step $\lambda_j^{\text{pred}} = \lambda^l$ yields a descending update and is therefore accepted. Whereas, if the above statement is violated, $j = j + 1$ and
4. Prediction $\lambda_{j-1}^{\text{pred}}$ is corrected by

$$\mu_j = \frac{\eta_0}{\hat{\omega}(\boldsymbol{\theta}^l, \lambda_{j-1}^{\text{pred}})||\Delta\boldsymbol{\theta}^l||} \tag{21}$$

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

In order to ensure the numerical stability of the damping algorithm, a predefined minimal relaxation $\tau_{\min}$ 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 (step 4). Inserting $\hat{\omega}(\boldsymbol{\theta}^l, \lambda_{j-1}^{\text{pred}})||\Delta\boldsymbol{\theta}^l||\lambda_{j-1}^{\text{pred}} > \eta_2$ into (21) and suppose that $\mu_j > \tau_{\min}$, we have $\lambda_j^{\text{pred}} < (\eta_0/\eta_2)\lambda_{j-1}^{\text{pred}}$. Thus, the minimal correction factor is given by the ratio η_0/η_2. A suitable choice of these control parameters is for example $\tau_{\min} = 0.01$, $\tau = 0.5$, $\eta_0 = 1$ 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^l attains the lower bound $\tau_{\min}$.

The described damping algorithm reflects the advantageous geometrical properties of the natural level function. Furthermore, correction step (21) 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

Besides the pure estimation of parameters and initial values, statistical information such as standard errors or confidence intervals for these values is 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 [38]. The covariance matrix of the estimates can be obtained from the Fisher information matrix which can be approximated by

$$\text{IF}(\hat{\boldsymbol{\theta}})_{ij} = \frac{\partial^2 \mathcal{L}(\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}_i \partial\boldsymbol{\theta}_j} \tag{22}$$

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

The described procedure for estimating parameters in ODEs is a maximum likelihood approach, such that (22) provides a sufficient approximation of $\boldsymbol{C}^{-1}$. Most of the minimiser [36] simultaneously calculate this covariance matrix within the quadratic approximation discussed in Section 2.2.

All described stages, integration, condensation and so on 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 [39–42]. 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 cannot be identified. A possible implementation of such a strategy is described in the next section.

4 Regularisation

If some parameters are not identifiable in a certain domain of the parameter space, matrix $\boldsymbol{P}_1^a$ of 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 modified matrix $\tilde{\boldsymbol{P}}_1^a$ attains its full rank. 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 [22] of $\boldsymbol{P}_1^a = \boldsymbol{U}\,\text{diag}(w_1, \ldots, w_{n_p})\boldsymbol{V}^{\text{T}}$ is calculated first to determine if $\boldsymbol{P}_1^a$ has its full rank. Both matrices $\boldsymbol{U}$ and $\boldsymbol{V}^{\text{T}}$ are orthogonal, $\boldsymbol{V}^{\text{T}}$ is the transposed matrix of $\boldsymbol{V}$, and $\text{diag}(w_1, \ldots, w_{n_p})$ is a diagonal matrix containing the positive (by convention) singular values $w_1, \ldots, w_{n_p}$. It is 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 machine accuracy. In order to judge which parameters contribute to the violation of the rank criterion the set $M_c = \{i\colon\ w_{n_p}/w_i \leq \epsilon_c\}$ of all singular directions is regarded. Let

$$\Pi_c = \sum_{i \in M_c} \boldsymbol{e}_i \otimes \boldsymbol{e}_i^{\text{T}}$$

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 $\tilde{\boldsymbol{P}}_1^a$ is then given by

$$\tilde{\boldsymbol{P}}_1^a = \boldsymbol{U}(\text{diag}(w_1, \ldots, w_{n_p}) + \Delta\Pi_c)\boldsymbol{V}^{\text{T}} \tag{23}$$

For a well-adjusted value of Δ, all parameters contributing to the singular directions are almost kept fixed if $\boldsymbol{P}_1^a$ is replaced by $\tilde{\boldsymbol{P}}_1^a$ in (10). Since the described regularisation method is similar to the classical damping procedure of Levenberg and Marquardt [43, 44], regularisation can also be regarded as an individual damping of ill-conditioned directions. If the regularisation is turned off at the last iteration, the singular directions of the covariance matrix can help to find the unidentifiable parameters. Note that if some initial values are not identifiable, the same procedure can also be applied to matrix $\boldsymbol{E}_1^a$ in (10).

5 Examples

To display how multiple shooting performs, we present two examples in the following. The first dataset consists of a simulated trajectory from a model for oscillations in calcium signalling which is also capable of showing complex or chaotic behaviour [45]. For this model, a small simulation study is presented to show superior performance of multiple shooting compared to single shooting as claimed in Section 2.2. As second example measured data obtained form the online material of [46] are considered. Here, biochemical data of the STAT5 pathway are modelled.

5.1 Example 1: simulated data

Calcium ions are an important second messager substance in eucaryotic cells. Thereby, Ca^{2+} is a substantial part of the cellular information processing system. It has been observed that the concentration of the cytoplasmatic calcium ions may exhibit oscillations [47]. A mathematical model of these oscillations is developed in the work of Kummer *et al.* [45] which shows for a specific set of parameters complex or chaotic behaviour. The main stages of the calcium signalling pathway are activation of the phospolipase C (PLC) enzyme by the activated G_α unit of a G-protein linked receptor. This enzyme is attached to the plasma membrane and itself catalyses the hydrolysis of the membrane lipid phosphatidyl inositol-4,5-bisphosphate to build inositol-1,4,5-trisphosphate (IP_3) and diacylglycerol. Then, IP_3 may bind to specific ion-channels in the endoplasmatic reticulum which lead to a massive out-flux of Ca^{2+} from intra-cellular stores.

For the following simulation study we used the most complex mathematical model presented in the work of Kummer *et al.* [45]. This model consists of four state variables representing the concentrations of: (1) the active G_α unit, G_α^*, (2) the active PLC, PLC*, (3) the free calcium in the cytoplasm, Ca_{cyt}, and (4) the calcium in the endoplasmatic reticulum, Ca_{er}. For sake of simplicity, the dynamics of the IP_3 is assumed to follow the dynamics of the active PLC. The dynamics of the remaining state variables is then given by the following differential equation

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}G_\alpha^* &= k_1 + k_2 G_\alpha^* - k_3 \mathrm{PLC}^* \frac{G_\alpha^*}{G_\alpha^* + \mathrm{Km}_1} \\
&\quad - k_4 \mathrm{Ca}_{\mathrm{cyt}} \frac{G_\alpha^*}{G_\alpha^* + \mathrm{Km}_2} \\
\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{PLC}^* &= k_5 G_\alpha^* - k_6 \frac{\mathrm{PLC}^*}{\mathrm{PLC}^* + \mathrm{Km}_3} \\
\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Ca}_{\mathrm{cyt}} &= k_7 \mathrm{PLC}^* \mathrm{Ca}_{\mathrm{cyt}} \frac{\mathrm{Ca}_{\mathrm{er}}}{\mathrm{Ca}_{\mathrm{er}} + \mathrm{Km}_4} + k_8 \mathrm{PLC}^* + k_9 G_\alpha^* \\
&\quad - k_{10} \frac{\mathrm{Ca}_{\mathrm{cyt}}}{\mathrm{Ca}_{\mathrm{cyt}} + \mathrm{Km}_5} - k_{11} \frac{\mathrm{Ca}_{\mathrm{cyt}}}{\mathrm{Ca}_{\mathrm{cyt}} + \mathrm{Km}_6} \\
\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Ca}_{\mathrm{er}} &= -k_7 \mathrm{PLC}^* \mathrm{Ca}_{\mathrm{cyt}} \frac{\mathrm{Ca}_{\mathrm{er}}}{\mathrm{Ca}_{\mathrm{er}} + \mathrm{Km}_4} \\
&\quad + k_{11} \frac{\mathrm{Ca}_{\mathrm{cyt}}}{\mathrm{Ca}_{\mathrm{cyt}} + \mathrm{Km}_6}
\end{aligned} \tag{24}$$

where the 17 parameters are chosen in the following manner: $k_1 = 0.09$, $k_2 = 2$, $k_3 = 1.27$, $k_4 = 3.73$, $k_5 = 1.27$, $k_6 = 32.24$, $k_7 = 2$, $k_8 = 0.05$, $k_9 = 13.58$, $k_{10} = 153$, $k_{11} = 4.85$, $\mathrm{Km}_1 = 0.19$, $\mathrm{Km}_2 = 0.73$, $\mathrm{Km}_3 = 29.09$, $\mathrm{Km}_4 = 2.67$, $\mathrm{Km}_5 = 0.16$ and $\mathrm{Km}_6 = 0.05$. For this specific parameterisation the solution of (24) shows a limit cycle. As initial values we use $G_\alpha^*(0) = 0.12$, PLC*(0) = 0.31, $\mathrm{Ca}_{\mathrm{cyt}}(0) = 0.0058$ and $\mathrm{Ca}_{\mathrm{er}}(0) = 4.3$. As sampling time domain we choose interval [0, 20] and the sampling interval is set to $\Delta t = 0.1$. This leads to 200 data points. We chose a biological reasonable noise model where the standard deviation of each observed variable is proportional to the concentration of its noise free state. This leads to an overall mean noise-to-signal ratio of 6.5% .

For comparing multiple shooting with single shooting, we aim to estimate the parameters $k_1, \ldots, k_{11}$. The initial guesses of these parameters are randomly selected from an uniform distribution over [0, 1]. Note, that some of the true parameters, for example k_5, k_6 and k_{10}, are far of this interval of initial guesses for the parameters, rendering the estimate to a difficult one. A snapshot of the initial trajectory, after the eighth multiple shooting iteration and the final trajectory is shown in Fig. 2. Here, 17 multiple shooting intervals are used, leading to a rather rough initial trajectory (Fig. 2*a*). These discontinuities are still present after eight iterations (Fig. 2*b*) and are completely removed at convergence, (Fig. 2*c*). The estimated 11 parameters are compatible with real parameter values stated earlier (data not shown). To compare the performance of multiple shooting to the initial value approach, a simulation study has been carried out. To achieve the most comparable results, a sample of 250 initial guesses are randomly selected. The performance for both multiple shooting and initial value approach is compared in terms of stability and computational load using the same initial guess for each sample. The results are summarised in Table 1. These results clearly support the statements about the superior stability of multiple shooting, since only 16% of fits converged for the initial value approach whereas 94% for multiple shooting. This picture does not change if only the fits to the global optimum are considered. Here, a significant drop of the percentage of converged fits is visible, 4% for single shooting and 49% for multiple shooting. This significant drop is a matter of fact that some of the initial guesses are more than two orders of magnitude away from the true parameters. If one compares the ratio of these values, it turns out that about twelve times as many convergent fits converged to the global optimum for multiple shooting than for the initial value approach. In terms of computational load, single shooting is substantially faster than multiple shooting if all convergent fits are considered. But if only convergent fits are taken into account, the computational effort is basically the same for both methods. Therefore for this particular problem, the condensation algorithm is highly efficient since the condensed problem is computationally equivalent to the initial value problem as discussed in Section 3.2. In addition, the high computational cost for converged fits to a local optimum in the case of multiple shooting indicates that the objective function around these minima is rather flat. This is due to the fact that the presented damping algorithm often hits the lower bound in these regions and therefore more iterations have to be taken into account. However, this property can be used to monitor the convergence of the algorithm. Moreover, the high standard deviation of the computational load for both methods indicates that the needed computational effort to find an optimum of the cost function highly depends on the used initial guess.

5.2 Example 2: measured data

So far, only simulated data have been 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

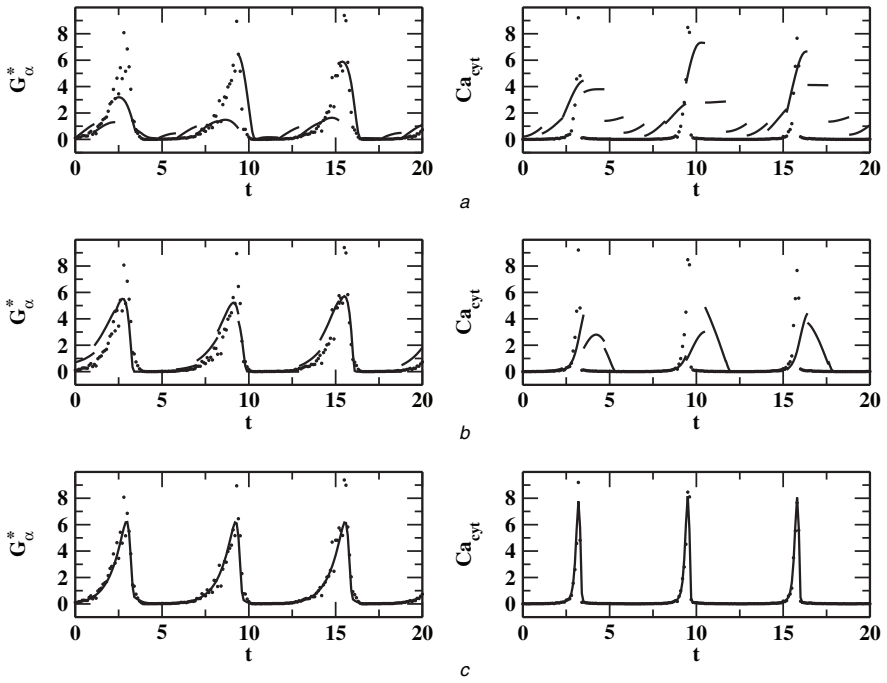

Fig. 2 *Identification of the presented calcium signalling pathway using multiple shooting*

For sake of clearness, three snapshots of the identification procedure is shown only for the state variables G^*_α and Ca_{cyt}
a Due to the large amount of multiple shooting intervals, the initial trajectory is highly discontinuous
b After 8 iterations the trajectory is significantly smoother
c Since the discontinuities are removed by the algorithm, the trajectory turns out to be continuous at convergence (38 iterations)

complicates the situation significantly. For the considered data, the model selection procedure is thoroughly described in the works of Swameye *et al.* and Timmer *et al.* [46, 48]. 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 STAT5 dimers enter the cell-nucleus and can trigger the transcription of target genes. After that, the dimer separates and the STAT5 molecule is dephosphorylated. Finally, these single STAT5 molecules are able to re-enter 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 approximated by an ODE. Since no in vivo measurements inside of the nucleus are possible, all nuclear processes are condensed into a single step containing 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 $\mathrm{EpoR}_A(t)$ and x_4 is the concentration of STAT5 molecules staying in the nucleus. Unfortunately, no concentration of the reaction components could 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 are accessible. For a given set of observations, the most simple identifiable system capturing all the properties stated earlier is

$$\begin{aligned}
\dot{x}_1 &= -k_1 x_1 \mathrm{EpoR}_A(t) + k_2 x_3(t-\tau) \\
\dot{x}_2 &= -x_2^2 + k_1 x_1 \mathrm{EpoR}_A(t) \\
\dot{x}_3 &= -k_2 x_3 + x_2^2 \\
\dot{x}_4 &= -k_2 x_3(t-\tau) + k_2 x_3
\end{aligned} \tag{25}$$

where k_1, k_2 are rate constants and τ is a delay parameter. Here, the rate constant of the x_2^2 term is set equal to one because it can be absorbed into the scaling parameter s_1; thus such a parameter would not be identifiable. Instead of using a 'hard' delay in (25), we decided to use a delay chain approach. A delay chain of length N is a linear ODE of type

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

Table 1: Comparison of multiple and single shooting (initial value approach) in terms of stability, convergence to the global optimum and computational load

	Single shooting	Multiple shooting
Convergent fits	16%	96%
Needed computational load	(31 ± 26) s	(102 ± 123) s
Fits converged to the global optimum	4%	49%
Needed computational load	(44 ± 16) s	(48 ± 58) s

The results are obtained from 250 runs using a randomly generated initial guess for each sample. For sake of comparability, the same initial guess is used for single and multiple shooting within the sample. For the simulations a computer with a 2.6-GHz Pentium 4 processor is used.

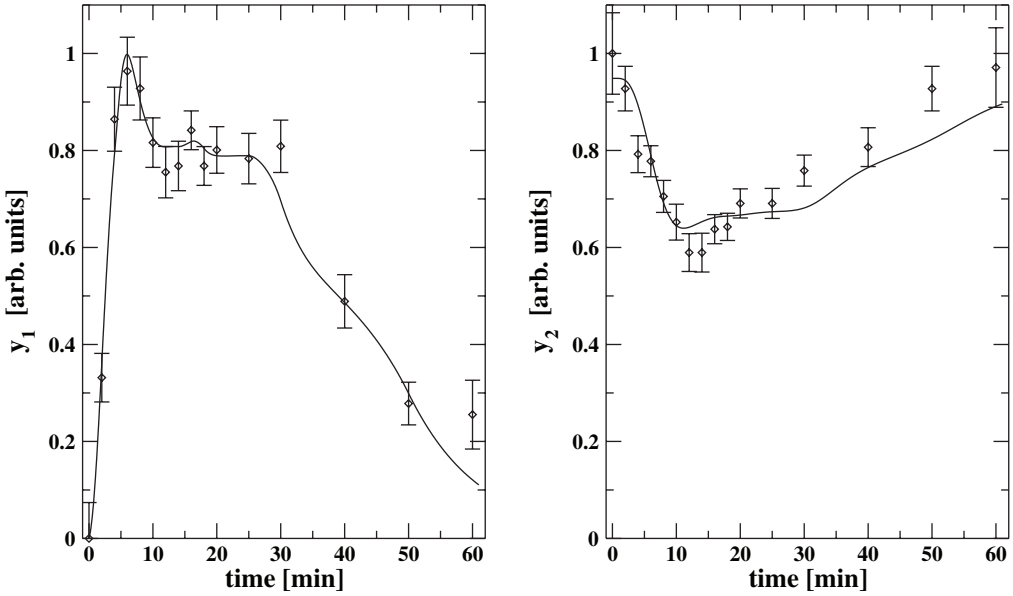

Fig. 3 *Total activated STAT5, y_1 and total STAT5, y_2, in the cytoplasm of the cell*
Trajectory of the best fit is indicated by the solid line

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 the 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. According to the experimental design outlined in the work of Swameye *et al*. [46], it is known that all state variables except x_1 are initially zero, these values are therefore kept fixed throughout the optimisation. In addition, the scaling parameters s_1, s_2 are not identifiable from a single experiment. Fixing them to $s_1 = 0.33$ and $s_2 = 0.26$, the remaining parameters as well as the initial value of x_1 are now identifiable. They turn out to be $k_1 = (2.12 \pm 0.22)\ \text{min}^{-1}\ \text{mol}^{-1}$, $k_2 = (0.109 \pm 0.015)\ \ \text{min}^{-1}\ \text{mol}^{-1}$, $\tau = (5.2 \pm 0.6)$ min and $x_1(0) = (3.71 \pm 0.07)$ mol. The corresponding trajectory is displayed in Fig. 3. As can be seen, the fitted model yields a good description of the data.

6 Summary

The parameter estimation procedure for ODEs, multiple shooting, is reviewed and described in detail. 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. Two examples have been provided to demonstrate the performance of multiple shooting.

Moreover, the extension of multiple shooting to partial differential equations is also possible [49, 50]. Additionally, the method can also be used to find an optimal experimental design [51, 52]. This broad applicability of the multiple shooting method marks the relevance of such a tool for a vast range of applied sciences and engineering. Especially for estimating parameters in complex reaction networks, as they often appear in systems biology, multiple shooting can substantially assist the modelling procedure. We hope that this article allows an easy re-implementation of the presented ideas and therefore propagating the availability of the method to a larger community.

Availability of the code: The code is available upon request.

7 Acknowledgements

M. P. received financial support from the German Research Foundation (DFG) and the German Federal Ministry of Education and Research (BMBF grant 0313074A). We also like to thank C. Fleck for his suggestions on the manuscript.

8 References

1 Ali, M., Storey, C., and Törn, A.: 'Applications of stochastic global optimization algorithms to practical problems', *J. Optim. Theory Appl.*, 1997, **95**, pp. 545–563
2 Zabinsky, Z.B., and Smith, R.L.: 'Pure adaptive search in global optimization', *Math. Prog.*, 1992, **53**, pp. 323–338
3 Banga, J.R., and Seider, W.D.: 'Global optimization of chemical processes using stochastic algorithms' in Floudas, C.A., and Pardalos, P.M. (Eds.) 'State of the art in global optimization' (Kluwer, Dordrecht, 1996), pp. 563–583
4 Törn, A., Ali, M., and Viitanen, S.: 'Stochastic global optimization: problem classes and solution techniques', *J. Glob. Optim.*, 1999, **14**, p. 437
5 Rinnooy-Kan, A.H.G., and Timmer, G.T.: 'Stochastic global optimization methods. Part I: Clustering methods', *Math. Prog.*, 1987, **39**, pp. 27–56
6 Holland, J.H.: 'Adaption in natural and artificial systems' (MIT Press, Cambridge, MA, 1992)

7 Banga, J.R., Moles, C.G., and Alonso, A.A.: 'Global optimization of bioprocesses using stochastic and hybrid methods' in Floudas, C.A., and Pardalos, P.M. (Eds.): 'Frontiers in global optimization' (Kluwer, 2003), pp. 45–70
8 Stoer, J., and Bulirsch, R.: 'Introduction to numerical analysis' (Springer, 1993)
9 Bock, H.G.: 'Numerical treatment of inverse problems in chemical reaction kinetics' in Ebert, K., Deuflhard, P., and Jäger, W. (Eds.): 'Modelling of chemical reaction systems' (Springer, 1981), pp. 102–125
10 Bock, H.G.: 'Recent advances in parameter identification techniques for ordinary differential equations' in Deuflhard, P., and Hairer, E. (Eds.): 'Numerical treatment of inverse problems in differential and integral equations' (Birkhäuser, 1983), pp. 95–121
11 Bock, H.G.: 'Randwertproblemmethoden zur parameteridenti fizierung in systemen nichtlinearer differentialgleichungen'. PhD Thesis, Universität Bonn, 1987
12 Cox, D.R., and Hinkley, D.V.: 'Theoretical statistics' (Chapman & Hall, London, 1994)
13 Milstein, J.: 'Fitting multiple trajectories simultaneously to a model of inducible enzyme synthesis', *Math. Biosci.*, 1978, **40**, pp. 175–184
14 Schittkowski, K.: 'Parameter estimation in systems of nonlinear equations', *Num. Math.*, 1995, **68**, pp. 129–142
15 Schittkowski, K.: 'Numerical data fitting in dynamical systems' (Kluwer, 2002)
16 Ardenghi, J.I., Maciel, M.C., and Verdiell, A.B.: 'A trust-region approach for solving a parameters estimation problem from the biotechnology area', *Appl. Num. Math.*, 2003, **47**, pp. 281–294
17 Richter, O., Nörtersheuser, P., and Pestemer, W.: 'Non-linear parameter estimation in pesticide degradation', *Sci. Total Environ.*, 1992, **123/124**, pp. 435–450
18 Timmer, J., Rust, H., Horbelt, W., and Voss, H.U.: 'Parameteric, nonparametric and parametric modelling of a chaotic circuit time series', *Phys. Lett. A*, 2000, **274**, pp. 123–134
19 Stribet, A.D., Rosenau, P., Ströder, A.C., and Strasser, R.J.: 'Parameter optimisation of fast chlorophyll fluorescence induction model', *Math. Comput. Simul.*, 2001, **56**, pp. 443–450
20 Horbelt, W., Timmer, J., Bünner, M., Meucci, R., and Ciofini, M.: 'Identifying physically properties of a CO_2 laser by dynamical modeling of measured time series', *Phys. Rev. E*, 2001, **64**, p. 016222
21 von Grünberg, H.H., Peifer, M., Timmer, J., and Kollmann, M.: 'Variations in substitution rate in human and mouse genomes', *Phys. Rev. Lett.*, 2004, **93**, p. 208102
22 Press, W.H., Flannery, B.P., Saul, S.A., and Vetterling, W.T.: 'Numerical recipes' (Cambridge University Press, Cambridge, 1992)
23 Kuhn, H., and Tucker, A.: 'Nonlinear programming'. Proc. 2nd Berkeley Symp. on Mathematical Statistics and Probabilistics, 1951, University of California Press, pp. 481–492
24 Leis, J.R., and Kramer, M.A.: 'The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations', *ACM Trans. Math. Softw.*, 1988, **14**, pp. 45–60
25 Leis, J.R., and Kramer, M.A.: 'ODESSA – an ordinary differential equation solver with explicit simultaneous sensitivity analysis', *ACM Trans. Math. Softw.*, 1988, **14**, pp. 61–67
26 Horbelt, W.: 'Maximum likelihood estimation in dynamical systems', PhD thesis, University of Freiburg, 2001. Available at http://webber.physik.uni-freiburg.de/ horbelt/diss
27 Horbelt, W., Timmer, J., and Voss, H.U.: 'Parameter estimation in nonlinear delayed feedback systems from noisy data', *Phys. Lett. A*, 2002, **299**, pp. 513–521
28 Voss, H.U., Peifer, M., Horbelt, W., Rust, H., and Timmer, J.: 'Identification of chaotic systems from experimental data' in Gousebet, G. (Ed.): 'Chaos and its reconstruction' (Nova Science Publishers Inc., New York, 2003), pp. 245–286
29 Hairer, E., Nørsett, S.P., and Wanner, G.: 'Solving ordinary differential equations. I: Nonstiff problems' (Springer, Berlin, 1993)
30 Bellen, A., and Zennaro, M.: 'Numerical methods for delay differential equations' (Oxford Science Publications, 2003)
31 Brenan, K.E., Campell, S.L., and Petzold, L.R.: 'The numerical solution of initial value problems in differential-algebraic equations' (North-Holland, New York, 1989)
32 Kelly, C.T.: 'Iterative methods for optimization' (SIAM, 1996)
33 Bischof, C.H., Carle, A., Corliss, G.F., Griewank, A., and Hovland, P.: 'ADIFOR: generating derivative code from Fortran programs', *Sci. Program.*, 1992, **1**, pp. 11–29
34 Griewank, A., Juedes, D., and Utke, J.: 'Algorithm 755: ADOL-C: a package for the automatic differentiation of algorithms written in C/C++', *ACM Trans. Math. Softw.*, 1996, **22**, pp. 131–167
35 Bauer, C., Frink, A., and Kreckel, R.: 'Introduction to the GiNaC framework for symbolic computation within the C++ programming language', *J. Symbolic Com.*, 2002, **33**, pp. 1–12
36 Hanson, R.J., and Haskell, K.H.: 'Algorithm 587: two algorithms for the linearly constrained least squares problem', *ACM Trans. Math. Softw.*, 1982, **8**, pp. 323–333
37 Stoer, J.: 'On the numerical solution of constrained least squares', *SIAM J. Numer. Anal.*, 1971, **382**, pp. 282–411
38 van der Vaart, A.W.: 'Asymptotic statistics' (Cambridge University Press, 1998)
39 Denis-Vidal, L., Joly-Blanchard, G., and Noiret, C.: 'Some effective approaches to check the identifiability of uncontrolled nonlinear systems', *Math. Comput. Simul.*, **57**, pp. 35–44
40 Godfray, K.R., and DiStefano, J.J.: 'Identifiability of model parameters', in Walter, E. (Ed.) 'Identification and system parameter estimation' (Pergamon Press, 1985), pp. 89–114
41 Ljung, L., and Glad, T.: 'On global identifiability for arbitrary model parameterization'', *Automatica*, 1994, **30**, pp. 265–276
42 Noykova, N., Müller, T.G., Gyllenberg, M., and Timmer, J.: 'Quantitative analysis of anaerobic wastewater treatment processes: identifiability and parameter estimation', *Biotech. Bioeng.*, 2002, **78**, pp. 89–103
43 Levenberg, K.A.: 'A method for the solution of certain nonlinear problems in least squares'', *Quart. Appl. Math.*, 1994, **2**, pp. 164–168
44 Marquardt, D.W.: 'An algorithm for least-squares-estimation of nonlinear problems in least squares', *SIAM J. Appl. Math*, 1963, **11**, pp. 431–441
45 Kummer, U., Olsen, L.F., Dixon, C.J., Green, A.K., Bornberg-Bauer, E., and Baier, G.: 'Switching from simple to complex oscillations in calcium signaling', *Biophys. J.*, 2000, **79**, pp. 1188–1195
46 Swameye, I., Müller, T.G., Timmer, J., Sandra, O., and Klingmüller, U.: 'Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by data-based modeling', *Proc. Natl. Acad. Sci.*, 2003, **100**, pp. 1028–1033
47 Schuster, S., Marhl, M., and Höfer, T.: 'Modelling of simple and complex calcium oscillations', *Eur. J. Biochem.*, 2002, **269**, pp. 1333–1355
48 Timmer, J., Müller, T.G., Swameye, I., Sandra, O., and Klingmüller, U.: 'Modelling the nonlinear dynamics of cellular signal transduction', *Int. J. Bif. Chaos*, 2004, **14**, pp. 2053–2060
49 Müller, T.G., and Timmer, J.: 'Fitting parameters in partial differential equations from partially observed noisy data', *Phys. D*, 2002, **171**, pp. 1–7
50 Müller, T.G., and Timmer, J.: 'Parameter identification techniques for partial differential equations', *Int. J. Bif. Chaos*, 2004, **14**, pp. 2053–2060
51 Lohmann, T., Bock, H.G., and Schlöder, J.P.: 'Numerical methods for parameter estimation and optimal experimental design in chemical reaction systems', *Ind. Eng. Chem. Res.*, 1992, **31**, pp. 54–57
52 Faller, D., Klingmüller, U., and Timmer, J.: 'Simulation methods for optimal experimental design in systems biology', *Simulation*, 2003, **79**, pp. 717–725

9 Appendices

9.1 Appendix A

In this section, an estimate for the natural level function T_N^l evaluated at $\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l$ for some $\lambda^l \in (0, 1]$ is derived. Provided that the second derivative $\mathrm{d}_\theta^2 \boldsymbol{R}^a$ for the vector of residuals $\boldsymbol{R}^a$ exists in a sufficiently large domain containing $\boldsymbol{\theta}^l$, the following estimate holds

$$T_N^l(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l) \leq \left(1 - \lambda^l + \frac{\lambda^{l^2}}{2}\omega(\boldsymbol{\theta}^l, \lambda^l)\right)^2 T_N^l(\boldsymbol{\theta}^l) \quad (26)$$

where for each $\lambda^l \in (0, 1]$, the function ω is given by

$$\omega(\boldsymbol{\theta}^l, \lambda^l) = \sup_{s \in (0, \lambda^l]} \left\{ \frac{\| G(\boldsymbol{\theta}^l)(\mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l + s\Delta\boldsymbol{\theta}^l) - \mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l))\Delta\boldsymbol{\theta}^l \|}{s\|\Delta\boldsymbol{\theta}^l\|^2} \right\} < \infty$$

Therefore on the basis of T_N^l a descendent step can always be found, if λ^l is correctly adjusted. In order show (26), consider the following estimates for $\alpha = \sqrt{(T_N^l(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l))} - (1 - \lambda^l)\sqrt{(T_N^l(\boldsymbol{\theta}^l))}$

$$\begin{aligned}\alpha &\leq \left| \sqrt{T_N^l(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l)} - (1-\lambda^l)\sqrt{T_N^l(\boldsymbol{\theta}^l)} \right| \\ &\leq \|G(\boldsymbol{\theta}^l)\boldsymbol{R}^a(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l) - (1-\lambda^l)G(\boldsymbol{\theta}^l)\boldsymbol{R}^a(\boldsymbol{\theta}^l)\| \quad (27)\end{aligned}$$

Since $G(\boldsymbol{\theta}^l)\boldsymbol{R}^a(\boldsymbol{\theta}^l) = \Delta\boldsymbol{\theta}^l$, and inserting $-G(\boldsymbol{\theta}^l)\mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l) = \mathbf{1}$ into (27), we arrive at

$$\begin{aligned}\alpha &\leq \left\| G(\boldsymbol{\theta}^l)[\boldsymbol{R}^a(\boldsymbol{\theta}^l + \lambda^l \Delta\boldsymbol{\theta}^l) - \boldsymbol{R}^a(\boldsymbol{\theta}^l) - \lambda^l \mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l)\Delta\boldsymbol{\theta}^l] \right\| \\ &= \left\| \int_0^{\lambda^l} \frac{G(\boldsymbol{\theta}^l)\{\mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l + s\Delta\boldsymbol{\theta}^l) - \mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l)\}\Delta\boldsymbol{\theta}^l}{s\|\Delta\boldsymbol{\theta}^l\|^2} \right. \\ &\quad \left. \times s\|\Delta\boldsymbol{\theta}^l\|\sqrt{T_N^l(\boldsymbol{\theta}^l)} \right\| \leq \frac{\lambda^{l^2}}{2}\omega(\boldsymbol{\theta}^l, \lambda^l)\|\Delta\boldsymbol{\theta}^l\|\sqrt{T_N^l(\boldsymbol{\theta}^l)}\end{aligned}$$

which proves (26).

9.2 Appendix B

In the following, the convergence of damping parameter λ^l, (Section 3.4) to $\lambda^l = 1$ whenever the method approaches the minimum is shown. Suppose that for all initial guesses $\boldsymbol{\theta}_0 \in D$, where D is a convex set, the undamped generalised quasi-Newton converges to $\boldsymbol{\theta}^*$ which minimises $\mathcal{L}$ (local convergence). Moreover, let the norm of the second derivative $\mathrm{d}_\theta^2 \boldsymbol{R}^a$ be bounded by $\tilde{\omega}$ on D. Then, for all $s \in [0, 1]$

$$\begin{aligned}&\|G(\boldsymbol{\theta}^l)(\mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}^l + s\Delta\boldsymbol{\theta}^l) - \mathrm{d}_\theta \boldsymbol{R}^a(\boldsymbol{\theta}))\Delta\boldsymbol{\theta}^l\| \\ &\quad = \|G(\boldsymbol{\theta}^l)\int_0^s \mathrm{d}_\theta^2 \boldsymbol{R}^a(\boldsymbol{\theta}^l + t\Delta\boldsymbol{\theta}^l)(\Delta\boldsymbol{\theta}^l, \Delta\boldsymbol{\theta}^l)\mathrm{d}t\| \\ &\quad \leq \|G(\boldsymbol{\theta}^l)\| \sup_{x \in D} \|\mathrm{d}_\theta^2 \boldsymbol{R}^a(x)\| s\|\Delta\boldsymbol{\theta}^l\|^2 \\ &\quad \leq \tilde{\omega} s\|\Delta\boldsymbol{\theta}^l\|^2\end{aligned}$$

by the continuity of G on D. According to (17), $\omega(\boldsymbol{\theta}^l, \lambda^l) \leq \tilde{\omega} < \infty$ and therefore $\omega(\boldsymbol{\theta}^l, \lambda^l)\|\Delta\boldsymbol{\theta}^l\| \leq \tilde{\omega}\|\Delta\boldsymbol{\theta}^l\| \to 0$ for $l \to \infty$. Due to (18), the maximal possible damping parameter leading to descending quasi-Newton steps converges to 1.