Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Systems biology

OXFORD

Systems Biology

Optimization and uncertainty analysis of ODE models using second order adjoint sensitivity analysis

Paul Stapor [1, 2], Fabian Fröhlich [1, 2] and Jan Hasenauer [1, 2*]

[1] Helmholtz Zentrum München - German Research Center for Environmental Health, Institute of Computational Biology, 85764 Neuherberg, Germany, and
[2] Technische Universität München, Center for Mathematics, Chair of Mathematical Modeling of Biological Systems, 85748 Garching, Germany.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Parameter estimation methods for ordinary differential equation (ODE) models of biological processes can exploit gradients and Hessians of objective functions to achieve convergence and computational efficiency. However, the computational complexity of established methods to evaluate the Hessian scales linearly with the number of state variables and quadratically with the number of parameters. This limits their application to low-dimensional problems.

Results: We introduce second order adjoint sensitivity analysis for the computation of Hessians and a hybrid optimization-integration based approach for profile likelihood computation. Second order adjoint sensitivity analysis scales linearly with the number of parameters and state variables. The Hessians are effectively exploited by the proposed profile likelihood computation approach. We evaluate our approaches on published biological models with real measurement data. Our study reveals an improved computational efficiency and robustness of optimization compared to established approaches, when using Hessians computed with adjoint sensitivity analysis. The hybrid computation method was more than two-fold faster than the best competitor. Thus, the proposed methods and implemented algorithms allow for the improvement of parameter estimation for medium and large scale ODE models.

Availability: The algorithms for second order adjoint sensitivity analysis are implemented in the Advance MATLAB Interface CVODES and IDAS (AMICI, https://github.com/ICB-DCM/AMICI/). The algorithm for hybrid profile likelihood computation is implemented in the parameter estimation toolbox (PESTO, https://github.com/ICB-DCM/PESTO/). Both toolboxes are freely available under the BSD license.

Contact: jan.hasenauer@helmholtz-muenchen.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

In systems and computational biology, ordinary differential equation (ODE) models are used to gain a holistic understanding of complex processes (Swameye *et al.*, 2003; Becker *et al.*, 2010). Unknown parameters of these ODE models, e.g., synthesis and degradation rates, have to be estimated from experimental data. This is achieved by optimizing an objective function, i.e. the likelihood or posterior probability of observing the given data (Raue *et al.*, 2013a). This optimization problem can be solved using multi-start local, global, or hybrid optimization methods (Raue *et al.*, 2013a; Villaverde *et al.*, 2015). Since experimental data are noise-corrupted and in most cases, only a subset of the state variables is observable, the inferred parameter estimates are subject to uncertainties. These uncertainties can be assessed using profile likelihood calculation (Raue *et al.*, 2009) and sampling (Girolami and Calderhead, 2011).

Many of the algorithms, which are applied in optimization or profile likelihood computation, exploit the gradient and the Hessian of the objective function or approximations thereof. These quantities can be used to determine search directions in optimization (Vassiliadis *et al.*, 1999; Balsa-Canto *et al.*, 2001) or to update the vector field in integration-based profile calculation (Chen and Jennrich, 1996). However, the evaluation of gradient and Hessian using standard approaches, i.e. finite differences or forward sensitivity analysis, is computationally demanding for high-dimensional ODE models. To accelerate the calculation of the objective function gradient, first order adjoint sensitivity analysis have been developed and applied (see (Fröhlich *et al.*, 2017c) and references therein). In engineering problems, similar concepts have been proposed for the calculation of the Hessian (Cacuci, 2015), but until now, these methods were never adapted to parameter estimation in biological applications.

In this manuscript, we provide a comprehensive formulation of second order adjoint sensitivity analysis for ODE constrained parameter estimation problems with discrete-time measurements. We outline the algorithmic evaluation of the Hessian and discuss the computational complexity. We include the functionality in the Advanced MATLAB Interface for CVODE and IDAS (AMICI). Furthermore, we introduce a hybrid approach for the calculation of profile likelihoods, which combines the ideas the two currently existing approaches and exploits the Hessian. We provide detailed comparisons of optimization and profile likelihood calculation of the proposed approaches and state-of-the-art methods based on published models of biological processes. Our analysis reveals that the robustness of optimization can be improved using Hessians. Moreover, we find that the hybrid method outperforms existing approaches for profile likelihood computation in terms of accuracy and computational efficiency when combined with second order adjoint sensitivity analysis.

2 Methods

2.1 Mathematical model

We consider ODE models of biological processes. The temporal evolution of a chemical concentration vector $x \in \mathbb{R}^{n_x}$ is given by a vector field f, depending on unknown parameters $\theta \in \mathbb{R}^{n_\theta}$ and time $t \in [t_0, t_{n_t}]$:

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

The initial state x_0 may be parameter dependent. As in most applications not all states can be observed directly, a set of observables $y \in \mathbb{R}^{n_y}$ is defined:

$$y(t,\theta) = h(x(t,\theta), t, \theta) \tag{2}$$

Measurements are usually noise-corrupted and this noise is modelled as normally distributed random variables with standard deviation σ_{ij} for observable $i = 1, \ldots, n_y$ and time point $j = 1, \ldots, n_t$:

$$\bar{y}_{ij} = y_i(x(t_j,\theta), t_j, \theta) + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2) \tag{3}$$

If the noise is unknown, it can be modelled as parameter dependent $\sigma_{ij} = \sigma_{ij}(\theta)$ and inferred from the data together with the other parameters. In the following, we will assume the σ_{ij} to be known. All derivations for parameter dependent σ_{ij} can be found in Section 1 of the Supplementary Information.

2.2 Parameter optimization

To infer the unknown parameters θ, we maximize the likelihood of observing the data given the parameter vector θ. Hence, the maximum likelihood estimate θ^* is defined as:

$$\theta^* = \mathrm{argmax}_{\theta \in \Omega}\, \mathcal{L}(\theta) \tag{4}$$

$\mathcal{L}(\theta)$ depends on the solution of the model and hence estimating θ^* is an ODE-constrained optimization problem. It must be solved numerically, since the considered ODEs rarely have closed form solutions. To improve numerical stability, we use the negative logarithm of the likelihood function as objective function, $\mathcal{J}(\theta) = -\mathcal{L}(\theta)$, for minimization:

$$\mathcal{J}(\theta) = \frac{1}{2}\sum_{j=1}^{n_t}\sum_{i=1}^{n_y}\left(\log(2\pi\sigma_{ij}^2) + \frac{(\bar{y}_{ij} - h_i(x(t_j,\theta), t, \theta))^2}{\sigma_{ij}^2}\right). \tag{5}$$

Typically, the considered optimization problems are non-convex and possess multiple local optima.

In this study, we solve the optimization problems using multi-start local optimization, an approach which has been shown to perform well in systems and computational biology (Raue *et al.*, 2013b). Initial points for local optimizations are drawn randomly from a biologically plausible region $\Omega \subset \mathrm{R}^{n_\theta}$ of the parameter space and the results of these optimizations are sorted by their final objective function value. Local optimization is carried out using either least-squares algorithms such as the Gauss-Newton-type methods combined with trust-region algorithms (Dennis *et al.*, 1981; Coleman and Li, 1996), or constraint optimization algorithms, which compute descent direction with (quasi-)Newton-type methods combined with interior-point or trust-region algorithms (Byrd *et al.*, 2000). Convergence of these methods can usually be improved, if the computed derivatives are accurate (Raue *et al.*, 2013b). Common least-squares algorithms such as the MATLAB function `lsqnonlin` only use first order derivatives of the residuals, whereas constraint optimization algorithms like the MATLAB function `fmincon` exploit first and second order derivatives of the objective function.

2.3 Profile likelihood calculation

Since experimental data are limited, parameter estimates are subject to uncertainties. Profile likelihoods (hereafter called profiles), introduced in (Raue *et al.*, 2009), are a common method to assess these uncertainties (Kreutz *et al.*, 2013). A profile is a maximum projection of the likelihood to a chosen parameter axis: for θ_k, $k \in \{1, \ldots, n_\theta\}$, the profile value at $\theta_k = c$ this given by

$$\mathrm{PL}_{\theta_k}(c) = \max_{\substack{\theta_k = c\\ \theta \in \Omega}} \mathcal{L}(D|\theta) \tag{6}$$

Profiles have to be computed separately for each parameter $\theta_k, k = 1, \ldots, n_\theta$, for which currently two approaches exist.

The optimization-based approaches (as implemented in (Raue *et al.*, 2015)) computes the profile for θ_k via a sequence of optimization problems (Raue *et al.*, 2009). In each step, all parameters besides θ_k are optimized and θ_k is fixed to a value c. For each new step, c either increased or decreased (depending on the profile calculation direction) and the new optimization is initialized based on the previously found parameter values. As long as the function $\mathrm{PL}_{\theta_k}(c)$ is smooth, this initial point will be close to the optimum and the optimization will converge within few iterations. Yet, as many optimizations have to be performed to obtain a full profile and usually all profiles have to be computed, this process is computationally demanding.

An efficient alternative to the optimization-based is the integration-based approach (Chen and Jennrich, 1996; Boiger *et al.*, 2016) (as implemented in (Kaschek *et al.*, 2016)), which circumvents the repeated optimization by using a dynamical system which evolves along the optimal path of the constraint optimization problem (6). For a constraint $g(\theta) = c$, in which $g\colon \Omega \longrightarrow \mathbb{R}$ is the constraint function (in our case $g(\theta) = \theta_k$), the dynamical system is obtained by differentiating the optimality condition

$$\nabla_\theta \mathcal{J}(\theta) + \lambda \nabla_\theta g(\theta) = 0, \tag{7}$$

with respect to the value of the constraint, c, where λ is a Lagrangian multiplier. This yields the differential equation

$$\begin{pmatrix} \nabla^2_\theta \mathcal{J} + \nabla^2_\theta g & \nabla_\theta g \\ \nabla_\theta g & 0 \end{pmatrix} \begin{pmatrix} \frac{d\theta}{dc} \\ \frac{d\lambda}{dc} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{8}$$

which can in principle be integrated with established differential equation solvers given the Hessian $\nabla^2_\theta \mathcal{J}$ or an approximation thereof (Chen and Jennrich, 2002). However, integrating the ODE in (8) is non-trivial, as the matrix on the left hand side can degenerate and the profile path may have discontinuities. This leads to small step sizes during ODE integration. Moreover, the trajectory of the ODE solver may deviate from the true profile path of Equation (6) due to numerical errors or approximations being used.

In this study, we introduce a hybrid optimization- and integration-based approach to handle discontinuities and to ensure optimality. Our hybrid approach employs by default the integration-based approach using a high-order Adams-Bashforth scheme (Shampine and Reichelt, 1997). A pseudo-inverse is used if the matrix in (8) is degenerated. If the step size falls below a previously defined threshold, integration will be stopped and a few optimization-based steps are carried out to circumvent numerical integration problems and to accelerate the calculation. Then, integration is reinitialized at the profile path. Moreover, the remaining gradient is monitored during profile integration. If it exceeds a certain value, an optimization will be started and integration reinitialized at the profile path.

2.4 Computation of objective function gradient and Hessian

Providing accurate derivative information is favourable for optimization and profile computation. Yet, due to the high computational complexity, gradients are sometimes not computed and Hessians even less frequently. In this section, we recapitulate available forward and adjoint sensitivity analysis methods to, subsequently, introduce second order adjoint sensitivity analysis for the efficient computation of the Hessian for ODE models. Remark: In the following, the dependencies of f, x, h and their derivatives on t, θ, and x are not stated explicitly. For a detailed mathematical description of all approaches, we refer to Supplementary Information, Section 1.

Computation of the objective function gradient

Many state-of-the-art toolboxes compute objective function gradients using forward sensitivity analysis. When differentiating Equation (5) with respect to θ_k, the gradient is obtained:

$$\frac{\partial \mathcal{J}}{\partial \theta_k} = \sum_{i=1}^{n_y} \sum_{j=1}^{n_t} \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma^2_{ij}} s^{y_i}_k \tag{9}$$

in which s^y_k denotes the sensitivity of observable y_i with respect to parameter θ_k. The observable sensitivities are calculated from the state sensitivities $s^x_k = \frac{\partial x}{\partial \theta_k}$ as

$$s^{y_i}_k = (\nabla_x h_i) s^x_k + \frac{\partial h_i}{\partial \theta_k} \tag{10}$$

The state sensitivities need to be computed by integrating the corresponding ODE, which is obtained from differentiating Equation (1):

$$\dot{s}^x_k = (\nabla_x f) s^x_k + \frac{\partial f}{\partial \theta_k} \tag{11}$$

In forward sensitivities analysis, the error in the state sensitivities can be controlled together with the error of the state variables when integrating both ODEs (11) together, which makes it possible to obtain accurate gradients (Fröhlich *et al.*, 2017c). However, using this method for a system with n_x state variables and n_θ parameters requires solving an ODE of the size $n_x(n_\theta + 1)$. First order forward sensitivity analysis hence scales linearly in the number of parameters and in the number of state variables, which is computationally demanding for large n_x and n_θ.

Adjoint sensitivity analysis circumvents the integration of the state sensitivities. In this approach, only the original ODE system (1) is integrated forward in time and subsequently the ODE for the adjoint state $p(t)$ is integrated backward in time, starting at t_{n_t}:

$$\dot{p} = -(\nabla_x f^T) p \tag{12}$$

$$p(t_{n_t}) = \sum_{i=1}^{n_y} \nabla_x h_i \frac{\bar{y}_{in_t} - h_i(t_{n_t})}{\sigma^2_{in_t}} \tag{13}$$

For time-discrete data, $p(t)$ has to be reinitialized for each measurement:

$$p(t_j) = \lim_{t \to t_j^+} p(t) + \sum_{i=1}^{n_y} \nabla_x h_i \frac{\bar{y}_{ij} - h_i(x(t_j, \theta), t_j)}{\sigma^2_{ij}}. \tag{14}$$

In the end, the gradient can be computed as

$$\frac{\partial \mathcal{J}}{\partial \theta_k} = -\sum_{i=1}^{n_y} \sum_{j=1}^{n_t} \frac{\bar{y}_{ij} - h_i}{\sigma^2_{ij}} \frac{\partial h_i}{\partial \theta_k} - \int_{t_0}^{t_{n_t}} p^T \frac{\partial f}{\partial \theta_k} dt - p(t_0)^T \frac{\partial x_0}{\partial \theta_k}. \tag{15}$$

where n_θ one dimensional quadratures have to be computed during the backward integration. In practice, these quadratures are typically computationally less expensive, so the linear dependence of the computation time on n_θ for adjoint sensitivity analysis can be considered to be weak, as pointed out in (Özyurt and Barton, 2005). This yields the gradient for little more than the cost of integrating two differential equations of the size n_x. As these scaling properties were shown to also hold true in practice (Fröhlich *et al.*, 2017c), adjoint sensitivity analysis is so far probably the most efficient method for the computation of gradients for large systems.

Computation and approximation of the objective function Hessian

In this study, we consider two approximations of the Hessian:

1. the Fisher Information Matrix (FIM) (Fisher, 1922)
2. the Broyden-Fletcher-Goldfarb-Shanno (BFGS) scheme (Goldfarb, 1970)

and employ three approaches to compute the Hessian itself

3. central finite differences, based on gradients from adjoint sensitivities (Andrei, 2009)
4. second order forward sensitivity analysis (Vassiliadis *et al.*, 1999)
5. second order adjoint sensitivity analysis

The FIM is related to the asymptotic covariance of maximum likelihood estimates (Swameye *et al.*, 2003) and provides an approximation to the Hessian of the negative log-likelihood function. The approximation converges quadratically in the size of the residuals $(\bar{y}_{ij} - h_i(t_j))/\sigma_{ij}$ (Raue, 2013). Although, the FIM provides only an approximation, it is used in optimization, as it can be computed using first order forward sensitivities:

$$\mathrm{FIM}_{k,\ell}(\theta) = \sum_{i=1} \sum_{j=1} \frac{1}{\sigma^2_{ij}} s^{y_i}_k(t_j) s^{y_i}_k(t_j)^T \tag{16}$$

The BFGS scheme is an algorithm, which computes a positive-definite approximation to the Hessian sequentially during an optimization process using gradients, which are computed in each optimization step. Different variants of this algorithm are implemented in many state-of-the-art optimization toolboxes, like e.g. (Wächter and Biegler, 2006).

Central finite differences compute the Hessian based on perturbations in each parameter direction by a small step δ:

$$\frac{\partial^2 \mathcal{J}(\theta)}{\partial\theta_k \partial\theta_\ell} \approx \frac{\frac{\partial\mathcal{J}(\theta+\delta e_\ell)}{\partial\theta_k} - \frac{\partial\mathcal{J}(\theta-\delta e_\ell)}{\partial\theta_k}}{2\delta} \tag{17}$$

where e_ℓ is the unit vector with 1 at the ℓ-th position. The accuracy of this method depends on the step size δ. Good choices of δ depend in turn on the error tolerances of the ODE solver and are thus not easy to determine (Hanke and Scherzer (2001) and the references therein).

Second order forward sensitivity analysis extends, similar to first order forward sensitivity analysis, the considered ODE system, now including first order and second order derivatives of the state variables (see Supplementary Information, Equation (9)). If the symmetry of the Hessian is exploited, This leads to an ODE system of the size $n_\theta(n_\theta+1)n_x/2$. Hence, the computational complexity of the problem scales quadratically in the number of parameters and linearly in the number of state variables, which limits this method to low-dimensional applications. Yet, second order forward sensitivity analysis yields accurate Hessians, since the error of the second order state sensitivities can be controlled during ODE integration.

Second order adjoint sensitivity analysis has so far never been applied in the field of systems and computational biology and we are not aware of any ready-to-use implementation thereof. Along the lines of first order adjoint sensitivity analysis, second order adjoint sensitivity analysis gives Hessians with better scaling properties than second order forward sensitivities. Again, the error of the Hessian can be controlled during ODE integration, yielding as accurate results as those from second order forward sensitivity analysis. To compute Hessians, the idea of the adjoint method is applied to (11) instead of (5). In a first step, the system defined by (11) is integrated forward in time. Subsequently, the corresponding adjoint system is integrated backwards in time, using the information from the forward simulation. This system consists of the original adjoint system plus the n_θ derivatives of p with respect to θ_k.

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial p}{\partial\theta_\ell}\right) &= -\left(\nabla_x f^T\right)\frac{\partial p}{\partial\theta_\ell} - \nabla_x\left(\frac{\partial f}{\partial\theta_\ell}\right)^T p \\ &\quad - \left(\left(s_\ell^x\right)^T \otimes \mathbf{1}_{1,n_\theta}\right)\left(\nabla_x \otimes \nabla_x f^T\right)p, \end{aligned}\tag{18}$$

$$\begin{aligned}\frac{\partial p(t_{n_t})}{\partial\theta_\ell} &= \sum_{i=1}^{n_y}\left(\frac{\bar{y}_{in_t} - h_i(t_{n_t})}{\sigma_{in_t}^2}\left(\nabla_x^T\nabla_x h_i(t_{n_t})s_\ell^x(t_{n_t})\right.\right. \\ &\quad \left. + \nabla_x\frac{\partial h_i(t_{n_t})}{\partial\theta_\ell}\right) + \frac{1}{\sigma_{in_t}^2}\left(\nabla_x^T h_i(t_{n_t})s_\ell^x(t_{n_t})\right. \\ &\quad \left.\left. + \frac{\partial h_i(t_{n_t})}{\partial\theta_\ell}\right)\nabla_x h_i(t_{n_t})\right)\end{aligned}\tag{19}$$

Again, the system must be reinitialized at every data time point:

$$\begin{aligned}\frac{\partial p(t_j)}{\partial\theta_\ell} &= \sum_{i=1}^{n_y}\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^2}\left(\nabla_x^T\nabla_x h_i(t_j)s_\ell^x(t_j) + \nabla_x\frac{\partial h_i(t_j)}{\partial\theta_\ell}\right) \\ &+ \lim_{t\to t_j^+}\frac{\partial p(t)}{\partial\theta_\ell} + \sum_{i=1}^{n_y}\frac{1}{\sigma_{ij}^2}\left(\nabla_x^T h_i(t_j)s_\ell^x(t_j) + \frac{\partial h_i(t_j)}{\partial\theta_\ell}\right)\nabla_x h_i(t_j)\end{aligned}\tag{20}$$

During this backward integration, n_θ^2 one-dimensional quadratures, which also depend on the forward trajectories of the state variables and their sensitivities, have to be calculated. Finally, the Hessian matrix can be assembled with the information coming from both ODE solves and these quadratures:

$$\begin{aligned}\frac{\partial^2\mathcal{J}}{\partial\theta_k\theta_\ell} &= \sum_{j=1}^{n_t}\sum_{i=1}^{n_y}\left(\frac{1}{\sigma_{ij}^2}\left(\nabla_x h_i(t_j)s_\ell^x(t_j) + \frac{\partial h_i(t_j)}{\partial\theta_\ell}\right)\frac{\partial h_i(t_j)}{\partial\theta_k}\right. \\ &\left. - \frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^2}\left(\frac{\partial\nabla_x h_i(t_j)}{\partial\theta_k}s_\ell^x(t_j)\frac{\partial^2 h_i(t_j)}{\partial\theta_\ell\partial\theta_k}\right)\right) \\ &- \frac{\partial p(t_0)^T}{\partial\theta_\ell}\frac{\partial x(t_0)}{\partial\theta_k} - p(t_0)^T\frac{\partial^2 x(t_0)}{\partial\theta_k\partial\theta_\ell} - \int_{t_0}^{t_{n_t}}\left(\frac{\partial p^T}{\partial\theta_\ell}\frac{\partial f}{\partial\theta_k}\right. \\ &\left. + p^T\frac{\partial^2 f}{\partial\theta_\ell\partial\theta_k} + p^T\frac{\partial\nabla_x^T f}{\partial\theta_k}s_\ell^x\right)dt.\end{aligned}\tag{21}$$

The computation of the Hessian by second order adjoint sensitivities requires solving two ODE systems of size $n_x(1+n_\theta)$ and n_θ^2 one dimensional quadratures. Again, these quadratures are fast to evaluate compared with the ODE systems. Hence, the scaling behaviour is expected to be almost linear in the number of state variables and the number of parameters.

3 Implementation and Results

To assess the potential of Hessian computation using second order adjoint sensitivities, we implemented the approach and we compared accuracy and computation time of the computed Hessians to to those of available methods. Furthermore, we evaluated parameter optimization and profile calculation methods using exact Hessian information for published models.

3.1 Implementation

The presented algorithms for the computation of gradients and Hessians by first and second order forward and adjoint sensitivity analysis were made applicable in the MATLAB and C++ based toolbox AMICI (Advanced Matlab Interface to CVODES and IDAS, Fröhlich *et al.* (2017a)), which uses the ODE solver CVODES (Serban and Hindmarsh, 2005) from the SUNDIALS package. The algorithm for hybrid profile calculation was implemented in the MATLAB toolbox PESTO (Parameter EStimation TOolbox, Stapor *et al.* (2017)).

3.2 Application examples

For the assessment of the methods, we considered five published models and corresponding datasets (M1 - M5). The models possess 3 to 26 state variables, 9 to 116 unknown parameters and a range of dataset sizes and identifiability properties. Four models describe signal transduction processes in mammalian cells, one describes the central carbon metabolism of E. Coli. An overview about the model properties is provided in Table 1 and a detailed description is included in the supplementary material.

3.3 Scalability

To verify the theoretical scaling of the discussed methods, we evaluated the computation times for the model with the largest number of state variables (M5). This evaluation revealed that the practical scaling rates are close to their theoretical predictions. (Figure 1A). Second order adjoint sensitivities, Fisher information matrix and finite differences based on first order adjoint sensitivities exhibited a roughly linear scaling with respect to the parameters. Second order forward sensitivities exhibited the predicted quadratic scaling. The Fisher information matrix showed the lowest computation time for all models. The proposed approach, second

Table 1. Overview of considered ODE models and their properties

ID	State variables	Parameters	Time points	Conditions	Data points	Modelled system	Reference
M1	6	9	16	1	46	Epo receptor signalling	Becker *et al.* (2010)
M2	3	28	7	3	72	RAF/MEK/ERK signalling	Fiedler *et al.* (2016)
M3	9	16	16	1	46	JAK/STAT signalling	Swameye *et al.* (2003)
M4	18	116	51	1	110	E.Coli carbon metabolism	Chassagnole *et al.* (2002)
M5	26	86	16	10	960	EGF & TNF signalling	MacNamara *et al.* (2012)

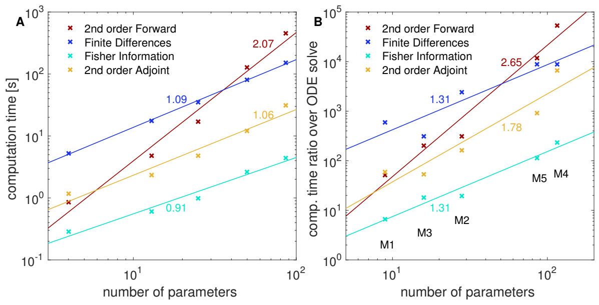

Fig. 1. Scaling of computation times of the four investigated methods to compute or approximate the Hessian, (at global optimum for each model)including linear fits and their slopes. All reported computation times were averaged over 10 runs. A) Model (M5) was taken and the number of parameters was fixed to different values. B) The ratio of the computation times for Hessians or its approximation over the computation time for solving the original ODE is given for the five models from Table 1.

order adjoint sensitivity analysis, was the fastest method to compute the exact Hessian, taking in average about 4 times as long to compute as the Fisher information matrix.

We also evaluated whether the same scaling holds across models (Figure 1B). Interestingly, we found similar but slightly higher slopes for all considered methods, although the number of state variables between models differs substantially. This suggests that in practice the number of parameters is indeed a dominating factor. Overall, second order adjoint sensitivity analysis was the most efficient method for the evaluation of the Hessian.

3.4 Accuracy

To assess the accuracy of Hessians and their approximations provided by the different methods, we compared the results at the global optimum. In general, we observed a good agreement of Hessians computed using second order adjoint and forward sensitivity analysis (Figure 2A). For the Hessian computed by finite difference, we found – as expected – numerical errors (Figure 2 B), which depended non-trivially on the combination of ODE solver accuracy and the step size of the finite differences. The Fisher information matrix usually differed substantially from the Hessians, even though this approximation is often considered to be good close to a local optimum (Figure 2 C).

In combination, our assessment of scaling and accuracy revealed that second order adjoint sensitivity analysis provides the most scalable approach to obtain accurate Hessian information. Rough approximations of the Hessian in terms of the FIM could however be computed at a lower computational cost.

3.5 Optimization

As our results revealed an trade-off between accuracy and computation time for computation Hessians, we investigated how this affects different optimization algorithms. To this end we compared Newton and quasi-Newton variants of the interior point algorithm and the trust region algorithm:

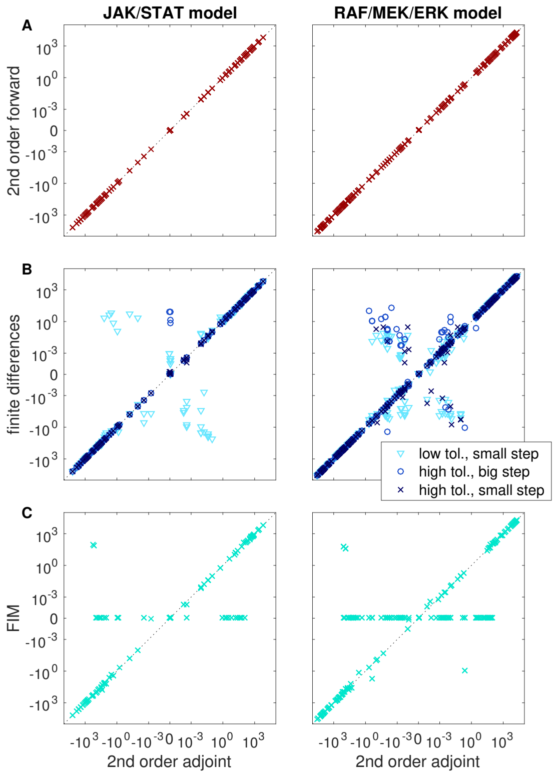

Fig. 2. Accuracy of different methods to compute or approximate the Hessian at the global optimum for the models M2 and M3. Each point represents the numerical value of one Hessian entry as computed by two different methods: A) second order forward analysis vs. second order adjoint analysis. B) finite differences (different finite difference step sizes and ODE solver tolerances were considered) vs. second order adjoint analysis. C) Fisher information matrix vs. second order adjoint analysis. All computations were carried out with relative and absolute tolerances set to 10^{-11} and 10^{-14}, respectively. For finite differences, lower accuracies of 10^{-7} and 10^{-10} were tested, together with the step sizes 10^{-5} and 10^{-2}.

- Residuals and their sensitivities were computed with first order forward sensitivity analysis and provided to the least-squares algorithm `lsqnonlin`, which used the trust-region-reflective algorithm.
- Gradient and FIM were computed using first order forward sensitivity analysis and provided to `fmincon`, which using the trust-region-reflective algorithm.
- Gradient and Hessian were computed with second order adjoint sensitivity analysis. A positive definite transformation of the Hessian was provided to `fmincon`, using the trust-region-reflective algorithm (which needs a positive definite Hessian to work).

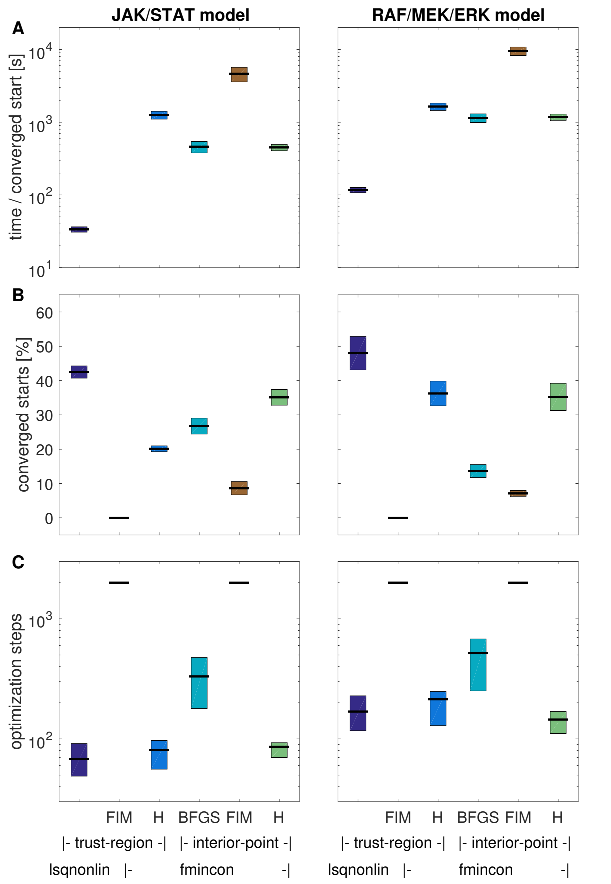

Fig. 3. Performance measures of different local optimization methods (`lsqnonlin` with trust-region algorithm and `fmincon` with trust-region and interior-point algorithm, using either Hessians (H), Fisher information matrix (FIM), or the BFGS scheme). The multi-start optimization was carried out multiple times using different starting points for the local optimizations. Mean and standard deviation for A) the ratio of computation time over converged optimization starts and B) the number of converged starts are shown. C) Median and standard deviation of the number of steps over all optimization runs.

- Gradients were calculated using first order forward sensitivity analysis and provided to `fmincon`, using the interior-point algorithm with BFGS approximation the Hessian.
- Gradient and FIM were computed with first order forward sensitivity analysis and provided to `fmincon`, using the interior-point algorithm.
- Gradients and Hessians were calculated with second order adjoint sensitivity analysis and provided to `fmincon`, using the interior-point algorithm.

The optimization study was carried out using the MATLAB toolbox PESTO for the models M2 and M3. For each of these local optimization methods, we performed four multi-start local optimizations with different initializations and 200 starting points each.

We considered the least-squares algorithm as gold standard for the considered problem class, as this method has previously been shown to be very efficient (Raue *et al.*, 2013b). Here, we studied the effect of using exact Hessians on the optimization algorithms trust-region-reflective and interior-point implemented in `fmincon`. As performance measure of the optimization methods, we considered the computation time per converged start (i.e. starts which reached the global optimum), the total number of converged starts and the number of optimization steps.

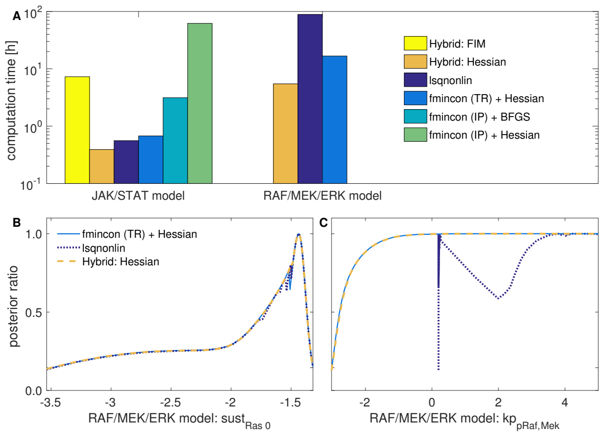

Fig. 4. Profile likelihood computation using either the optimization-based method (`lsqnonlin` or `fmincon` with trust-region (TR) or interior-point (IP) algorithm and Hessian or BFGS approximation), or the hybrid method with either FIM or Hessian. A) Computation time for all profiles of the considered models. Three methods failed to compute profiles for the RAF/MEK/ERK model. Thus, their computation times are not depicted. B) A profile of the JAK/STAT model, all methods in good agreement with each other. C) A profile of the RAF/MEK/ERK model, `lsqnonlin` failed to compute the profile. D) Another profile of the RAF/MEK/ERK model, three methods in good agreement with each other.

The least-square solver `lsqnonlin` outperformed, as expected, the constraint optimization method `fmincon` (Figure 3 and supplementary Figure 1). Among the constraint optimization methods, the methods using exact Hessians computed using the second order adjoint method, performed equal or better than the alternatives regarding overall computational efficiency (Figure 3 A). Indeed, the methods reached a higher percentage of converged starts (Figure 3 B and supplementary Figure 1) than `fmincon` using the FIM or the BFGS scheme. This is important, as convergence of the local optimizer is often the critical property (Raue *et al.*, 2013b). In addition, the number of necessary function evaluations was reduced (Figure 3 C). Furthermore, we found differences in convergence and computational efficiency for `fmincon`, depending on the chosen algorithm.

3.6 Profile Likelihood Calculation

To assess the benefits of Hessians in uncertainty analysis, we compared the performance of optimization- and integration-based profile calculation methods for the models M2 and M3. For the optimization based approach we employed the algorithm implemented in PESTO, which uses first order proposal with adaptive step-length selection (Boiger *et al.*, 2016). We compared the local optimization strategies described in Section 3.5 (omitting the methods based on the FIM, due to their poor performance). For the hybrid approach, we used MATLAB default tolerances for ODE integration. We compared the hybrid scheme using Hessians and the FIM. All profiles were computed to a confidence level of 95%.

The comparison of the profile likelihoods calculated using different approaches revealed substantial differences (Figure 4B and C). The optimization-based approaches worked fine for the JAK/STAT model but mostly failed for the RAF/MEK/ERK model (Figure 4A). For the RAF/MEK/ERK model, only `fmincon` with the trust-region-reflective algorithm and exact Hessians worked reliably among the optimization-based methods. Even `lsqnonlin` yielded inaccurate results for 11 out of 28 parameter profiles. A potential reason is that the tolerances – which were previously also used for optimization – were not sufficiently tight. Purely integration-based methods failed due to numerical problems, e.g. jumps in the profile paths. Even extensive manual tuning and the use of different established ODE solvers (including ode113, ode45, ode23, and ode15s) did not result in reasonable approximations for all profiles. In contrast, the hybrid approach provided accurate profiles for all parameters and all models, when provided with exact Hessians. When provided with the FIM, the hybrid approach failed, when it had to perform optimization, which could not rely on Hessians in this case.

In addition to the accuracy, also the computation time of the methods differed substantially. The hybrid method using exact Hessians was substantially faster than the remaining methods (see Figure 4A and Supplementary Information, Figure 6). The second fastest method was the optimization-based approach using the Hessian for optimization. lsqnonlin was slightly and fmincon using the interior-point algorithm substantially slower (for both, the BFGS scheme and Hessian), although they – as mention above – did not provide accurate profiles.

Overall, the proposed hybrid approach using exact Hessians outperformed all other methods. Compared to the best reliable competitor (optimization-based profile calculation using fmincon with the trust-region algorithm and exact Hessians), the computation time was reduced by more than a factor of two. This is substantial for such highly optimized routines and outlines the potential of exact Hessian information for uncertainty analysis.

4 Discussion

Mechanistic ODE models in systems and computational biology rely on parameter values, which are inferred from experimental data. In this manuscript, we showed that the efficiency of some of the most common methods in parameter estimation can be improved by providing exact second order derivatives. We presented second order adjoint sensitivity analysis, a method to compute accurate Hessians at low computational cost, i.e. the method scales linearly in the number of model parameters and state variables. We also provide a ready-to-use implementation thereof in the freely available toolbox AMICI.

We showed that second order adjoint sensitivity analysis possesses better scaling properties than common methods to compute Hessians while yielding accurate results, rendering it a promising alternative to existing techniques. Moreover, we demonstrated that state-of-the-art constraint optimization algorithms yield more robust results when using exact Hessians. For the computation of profile likelihoods, we demonstrated that Hessians can improve computation time and robustness of various state-of-the-art methods. Furthermore, we presented a hybrid method for profile computation, which can efficiently handle stiff and ill-conditioned problems. We also provided an implementation of this method in the parameter estimation toolbox PESTO. Although being a reliable tool in uncertainty analysis (Fröhlich *et al.*, 2014), profile likelihoods are often disregarded due to their high computational effort. The presented hybrid method based on exact Hessians is an approach the tackle this problem, as already the rudimentary implementation used in this study outperformed all established approaches.

The analysis of the optimizer performance revealed that least-squares algorithms (such as lsqnonlin), which exploit the problem structure are difficult to outperform. Many parameter estimation problems consider in systems biology do however not possess this structure. This is for instance the case for problems with additional constraints or applications considering the chemical master equation (Fröhlich *et al.*, 2016), or ODE-constrained mixture models (Hasenauer *et al.*, 2014). For these problem classes, the constraint trust-region and interior-point optimization algorithms as implemented in fmincon are the state-of-the-art methods. Additionally, new algorithms, which can exploit the additional curvature information, available through exact Hessian computation, in novel ways are steadily developed (see Fröhlich *et al.* (2017b)). Either directions of negative curvature can be used to escape saddle-points efficiently (Dauphin *et al.*, 2014), or third-order approximations of the objective functions are constructed iteratively from the Hessians along the trajectory of optimization to improve the convergence order (Martinez and Raydan, 2017). These approaches might outperform current optimization strategies, which are not designed to exploit directions of negative curvature that may be present in non-convex problems, and are therefore interesting subjects of further studies using the methods for Hessian computation introduced here.

While this study focused on the efficient calculation of the Hessian, second order adjoint sensitivity analysis can also be used to compute Hessian vector products. This information can be exploited by optimization methods such as truncated Newton (Nash, 1984) or accelerated conjugate gradient (Andrei, 2009) algorithms, which are suited for large-scale optimization problems. These are a few examples to illustrate how the presented results may pave the way for future improvements.

Funding

This work was supported by the German Research Foundation (DFG) through the Graduate School of Quantitative Biosciences Munich (QBM; F.F.) and through the European Union's Horizon 2020 research and innovation programme under grant agreement no. 686282 (J.H., P.S.).

References

Andrei, N. (2009). Accelerated conjugate gradient algorithm with finite difference hessian/vector product approximation for unconstrained optimization. *Journal of Computational and Applied Mathematics*, **230**(2), 570–582.

Balsa-Canto, E., Banga, J. R., Alonso, A. A., and Vassiliadis, V. S. (2001). Dynamic optimization of chemical and biochemical processes using restricted second-order information. *Comput. Chem. Eng.*, **25**(4), 539–546.

Becker, V., Schilling, M., Bachmann, J., Baumann, U., Raue, A., Maiwald, T., Timmer, J., and Klingmüller, U. (2010). Covering a broad dynamic range: information processing at the erythropoietin receptor. *Science*, **328**(5984), 1404–1408.

Boiger, R., Hasenauer, J., Hross, S., and Kaltenbacher, B. (2016). Integration based profile likelihood calculation for PDE constrained parameter estimation problems. *Inverse Prob.*, **32**(12), 125009.

Byrd, R. H., Gilbert, J. C., and Nocedal, J. (2000). A trust region method based on interior point techniques for nonlinear programming. *Math. Program.*, **89**(1), 149–185.

Cacuci, D. G. (2015). Second-order adjoint sensitivity analysis methodology (2nd-asam) for computing exactly and efficiently first- and second-order sensitivities in large-scale linear systems: Ii. illustrative application to a paradigm particle diffusion problem. *Journal of Computational Physics*, **284**(1), 700–717.

Chassagnole, C., Noisommit-Rizzi, N., Schmid, J. W., Mauch, K., and Reuss, M. (2002). Dynamic modeling of the central carbon metabolism of escherichia coli. *Biotechnol Bioeng*, **79**(1), 53–73.

Chen, J.-S. and Jennrich, R. I. (1996). The signed root deviance profile and confidence intervals in maximum likelihood analysis. *J. Am. Stat. Assoc.*, **91**(435), 993–998.

Chen, J.-S. and Jennrich, R. I. (2002). Simple accurate approximation of likelihood profiles. *J. Comput. Graphical Statist.*, **11**(3), 714–732.

Coleman, T. F. and Li, Y. (1996). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, **6**, 418–445.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., and Cho, K. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems 26 (NIPS 2014)*, pages 2933–2941.

Dennis, Jr., J. E., Gay, D. M., and Welsch, R. E. (1981). Algorithm 573: Nl2sol—an adaptive nonlinear least-squares algorithm. *ACM T. Math. Software.*, **7**(3), 369–383.

Fiedler, A., Raeth, S., Theis, F. J., Hausser, A., and Hasenauer, J. (2016). Tailored parameter optimization methods for ordinary differential equation models with steady-state constraints. *BMC Syst. Biol.*, **10**(80).

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philos. Trans. R. Soc. London, Ser. A*, **222**, 309–368.

Fröhlich, F., Theis, F. J., and Hasenauer, J. (2014). Uncertainty analysis for non-identifiable dynamical systems: Profile likelihoods, bootstrapping and more. In P. Mendes, J. O. Dada, and K. O. Smallbone, editors, *Proc. 12th Int. Conf. Comp. Meth. Syst. Biol.*, Lecture Notes in Bioinformatics, pages 61–72. Springer International Publishing Switzerland.

Fröhlich, F., Thomas, P., Kazeroonian, A., Theis, F. J., Grima, R., and Hasenauer, J. (2016). Inference for stochastic chemical kinetics using moment equations and system size expansion. *PLoS Comput. Biol.*, **12**(7).

Fröhlich, F., Weindl, D., Stapor, P., and Hasenauer, J. (2017a). Icb-dcm/amici: Amici 0.4.0 (version v0.4.0). Zenodo. http://doi.org/10.5281/zenodo.579891.

Fröhlich, F., Loos, C., and Hasenauer, J. (2017b). Scalable inference of ordinary differential equation models of biochemical processes. *arXiv preprint arXiv:1711.08079*.

Fröhlich, F., Kaltenbacher, B., Theis, F. J., and Hasenauer, J. (2017c). Scalable parameter estimation for genome-scale biochemical reaction networks. *PLoS Comput. Biol.*, **13**(1), e1005331.

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Statist. Soc. B*, **73**(2), 123–214.

Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Math Comp*, **24**(109), 23–26.

Hanke, M. and Scherzer, O. (2001). Inverse problems light: Numerical differentiation. *Am. Math. Mon.*, **108**(6), 512–521.

Hasenauer, J., Hasenauer, C., Hucho, T., and Theis, F. J. (2014). ODE constrained mixture modelling: A method for unraveling subpopulation structures and dynamics. *PLoS Comput. Biol.*, **10**(7), e1003686.

Kaschek, D., Mader, W., Fehling-Kaschek, M., Rosenblatt, M., and Timmer, J. (2016). Dynamic modeling, parameter estimation and uncertainty analysis in r. https://www.biorxiv.org/content/early/2016/11/02/085001.

Kreutz, C., Raue, A., Kaschek, D., and Timmer, J. (2013). Profile likelihood in systems biology. *FEBS J.*, **280**(11), 2564–2571.

MacNamara, A., Terfve, C., Henriques, D., Bernabé, B. P., and Saez-Rodriguez, J. (2012). State–time spectrum of signal transduction logic models. *Phys Biol*, **9**(4), 045003.

Martinez, J. M. and Raydan, M. (2017). Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization. *Journal of Global Optimization*, **68**(2), 367–385.

Nash, S. G. (1984). Newton-type minimization via the Lanczos method. *SIAM Journal on Numerical Analysis*, **21**(4), 770–788.

Özyurt, D. B. and Barton, P. I. (2005). Cheap second order directional derivatives of stiff ODE embedded functionals. *SIAM J. Sci. Comput.*, **26**(5), 1725–1743.

Raue, A. (2013). *Quantitative Dynamic Modeling: Theory and Application to Signal Transduction in the Erythropoietic System*. Phd. thesis, Albert-Ludwigs-Universität Freiburg im Breisgau.

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., and Timmer, J. (2009). Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, **25**(25), 1923–1929.

Raue, A., Kreutz, C., Theis, F. J., and Timmer, J. (2013a). Joining forces of Bayesian and frequentist methodology: A study for inference in the presence of non-identifiability. *Philos T Roy Soc A*, **371**(1984).

Raue, A., Schilling, M., Bachmann, J., Matteson, A., Schelke, M., Kaschek, D., Hug, S., Kreutz, C., Harms, B. D., Theis, F. J., Klingmüller, U., and Timmer, J. (2013b). Lessons learned from quantitative dynamical modeling in systems biology. *PLoS ONE*, **8**(9), e74335.

Raue, A., Steiert, B., Schelker, M., Kreutz, C., Maiwald, T., Hass, H., Vanlier, J., Tönsing, C., Adlung, L., Engesser, R., Mader, W., Heinemann, T., Hasenauer, J., Schilling, M., Höfer, T., Klipp, E., Theis, F. J., Klingmüller, U., Schöberl, B., and J.Timmer (2015). Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems. *Bioinformatics*, **31**(21), 3558–3560.

Serban, R. and Hindmarsh, A. C. (2005). CVODES: An ODE solver with sensitivity analysis capabilities. *ACM Math. Software*, **31**(3), 363–396.

Shampine, L. F. and Reichelt, M. W. (1997). The matlab ode suite. *SIAM J. Sci. Comput.*, **18**(1), 1–22.

Stapor, P., Weindl, D., Ballnus, B., Hug, S., Loos, C., Fiedler, A., Krause, S., Hross, S., Fröhlich, F., and Hasenauer, J. (2017). PESTO: Parameter EStimation TOolbox. *Bioinformatics*, **btx676**.

Swameye, I., Müller, T. G., Timmer, J., Sandra, O., and Klingmüller, U. (2003). Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling. *Proc. Natl. Acad. Sci. USA*, **100**(3), 1028–1033.

Vassiliadis, V. S., Canto, E. B., and Banga, J. R. (1999). Second-order sensitivities of general dynamic systems with application to optimal control problems. *Chem. Eng. Sci.*, **54**(17), 3851–3860.

Villaverde, A. F., Henriques, D., Smallbone, K., Bongard, S., Schmid, J., Cicin-Sain, D., Crombach, A., Saez-Rodriguez, J., Mauch, K., Balsa-Canto, E., Mendes, P., Jaeger, J., and Banga, J. R. (2015). BioPreDyn-bench: A suite of benchmark problems for dynamic modelling in systems biology. *BMC Syst. Biol.*, **9**(8).

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, **106**(1), 25–57.