OXFORD

Systems biology

Parameter estimation for dynamical systems with discrete events and logical operations

Fabian Fröhlich[1,2], Fabian J. Theis[1,2], Joachim O. Rädler[3] and Jan Hasenauer[1,2,*]

[1]Institute of Computational Biology, Helmholtz Zentrum München, Neuherberg 85764, Germany, [2]Center for Mathematics, Technische Universität München, Garching 85748, Germany and [3]Faculty of Physics, Ludwig-Maximilians-Universität, München 80539, Germany

*To whom correspondence should be addressed.

Abstract

Motivation: Ordinary differential equation (ODE) models are frequently used to describe the dynamic behaviour of biochemical processes. Such ODE models are often extended by events to describe the effect of fast latent processes on the process dynamics. To exploit the predictive power of ODE models, their parameters have to be inferred from experimental data. For models without events, gradient based optimization schemes perform well for parameter estimation, when sensitivity equations are used for gradient computation. Yet, sensitivity equations for models with parameter- and state-dependent events and event-triggered observations are not supported by existing toolboxes.

Results: In this manuscript, we describe the sensitivity equations for differential equation models with events and demonstrate how to estimate parameters from event-resolved data using event-triggered observations in parameter estimation. We consider a model for GFP expression after transfection and a model for spiking neurons and demonstrate that we can improve computational efficiency and robustness of parameter estimation by using sensitivity equations for systems with events. Moreover, we demonstrate that, by using event-outputs, it is possible to consider event-resolved data, such as time-to-event data, for parameter estimation with ODE models. By providing a user-friendly, modular implementation in the toolbox AMICI, the developed methods are made publicly available and can be integrated in other systems biology toolboxes.

Availability and Implementation: We implement the methods in the open-source toolbox Advanced MATLAB Interface for CVODES and IDAS (AMICI, https://github.com/ICB-DCM/AMICI).

Contact: jan.hasenauer@helmholtz-muenchen.de

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

1 Introduction

Systems and computational biology aim at a holistic understanding of biological processes. To achieve this, mechanistic mathematical models are developed which recast their essential properties (Kitano, 2002). These mathematical models—mostly ordinary differential equations (ODEs) (Klipp *et al.*, 2005)—describe the temporal evolution of states of biological processes by accounting for continuous changes (e.g. synthesis, interconversion and degradation) as well as discrete transitions and logical operations. Transitions can be triggered by internal or external events, e.g. bolus injections or fast processes. Exemplarily, in the firing of neurones, the event trigger is the membrane voltage reaching a threshold voltage, which results in an instantaneous decrease of the membrane voltage and an increase in the ion flux across the membrane (Izhikevich, 2003). Logical

Table 1. Sensitivity and event related features for a selection of toolboxes

	COPASI (Hoops et al., 2006)	PySCeS (Olivier et al., 2005)	libRoadRunner (Somogyi et al., 2015)	IQM Tools (Schmidt and Jirstrand, 2006)	SBML-SAT (Zi et al., 2008)	ByoDyn (de Lomana et al., 2007)	DEDiscover (de Lomana et al., 2007)	SensSB (Rodriguez-Fernandez and Banga, 2010)	SimBiology (Roberts, 2006)	SOSlib (Lu et al., 2008)	odeSD (Gonnet et al., 2012)	IBRENA (Liu and Neelamegham, 2008)	Data2Dynamics (Raue et al., 2015)	BIOPARKIN (Dierkes et al., 2011)	DOTcvpSB (Hirmajer et al., 2009)	SloppyCell (Myers et al., 2007)	AMICI
Events	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓
with time dependent trigger	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓
with state/parameter dependent trigger	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	×	✓	✓
with event observable	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✓	✓
with missing event penalization	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✓
Symbolic sensitivity equations	×	×	×	×	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
for time dependent trigger	×	×	×	×	×	×	×	×	×	×	×	×	×	✓	✓	✓	✓
for state/parameter dependent trigger	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✓	✓
for event observable	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✓
for missing event penalization	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	✓

operations appear in the presence of latent state and discrete decisions, e.g. in the context of coupled models of gene regulation and signal transduction (Le Novère, 2015; McAdams and Shapiro, 1995).

In this study, we consider ODE models with discrete events, i.e. event-triggered state transitions and observations, and logical operations. This broad model class becomes increasingly relevant as it allows for the coupling of biological processes operating on different time-scales (Le Novère, 2015) and involved observation processes (Geissen et al., 2016). While the Systems Biology Markup Language (SBML) (Hucka et al., 2003) facilitates the description of discrete events and logical operations, the functionality supported by software toolboxes is limited (Table 1). Many toolboxes are restricted to time-triggered events and do not support trigger functions which depend on the state variables or parameters of the ODE model. Of the toolboxes supporting parameter- and state-dependent trigger functions, only SloppyCell (Myers et al., 2007) allows for sensitivity analysis using symbolically derived forward sensitivity equations. However, no existing toolbox supports sensitivity analysis for event-triggered observations which yield event-resolved data, e.g. neuronal spike trains (Perkel et al., 1967). Currently, such event-triggered observations are rarely linked to mechanistic models, but instead described by statistical models (Geissen et al., 2016; Holford, 2013), which limits the understanding of the underlying processes.

The shortcomings of the available software toolboxes complicate the analysis of ODE models with discrete events and logical operations, especially for gradient based parameter estimation. Most ODE models possess unknown parameters, such as reaction rates, which have to be estimated from experimental data. The corresponding nonlinear optimization problems can be solved using local and global optimization schemes (Egea et al., 2014; Raue et al., 2015). Both approaches benefit from reliable gradient computation via forward sensitivity equations (Raue et al., 2013). For ODE models with events for which no sensitivity equations are available, numerical differentiation has to be employed to assess the gradient of the objective function with respect to the parameters. This results in

inferior performance. Beyond gradient computation, sensitivity equations can be used to gain insight into model properties (Dai et al., 2014; Schilling et al., 2009), to guide experimental design (Balsa-Canto et al., 2010; Raue et al., 2011; Vanlier et al., 2012) and to accelerate uncertainty analysis (Girolami and Calderhead, 2011; Raue et al., 2009).

In this manuscript we present the governing equations for state and output sensitivities for ODE models with discrete events and logical operations. The equations have been derived in the context of hybrid models by Barton et al. (1998) and Rozenvasser (1967), but were neither broadly adopted nor evaluated in the systems biology community. Furthermore, we introduce an objective function that allows the consideration of event-triggered observations through mechanistic models. All relevant methods are implemented in the open-source software toolbox Advanced Matlab Interface for CVODES and IDAS (AMICI, originally presented by Fröhlich et al., 2016). AMICI provides a user-friendly MATLAB interface to the SUNDIALS solvers CVODES (Serban and Hindmarsh, 2005) and IDAS (Hindmarsh et al., 2005) and automatically handles events, which are not natively supported by CVODES or IDAS. The methods and their implementation are evaluated using two examples: A model for GFP expression after transfection which includes the instantaneous release of mRNA molecules and a model for a spiking neuron, in which the after-spike reset of the membrane potential is instantaneous. For these models we evaluate the optimizer efficiency and convergence using sensitivity based gradients as well as finite difference based gradients. For the model of a spiking neuron, parameters are estimated solely from event-resolved data, in this case the time points of the after-spike resets.

2 Methods

In this section, we will introduce ODE models with discrete events and logical operations and formulate the respective sensitivity

equations. Furthermore, we will formulate the objective function for parameter estimation which allows the consideration of data which is not time-resolved, but event-resolved, such as time-to-event.

2.1 Biochemical processes with events

For biochemical processes, the continuous change of the concentration of different chemical species and other biological quantities is often formulated as system of ODEs,

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

where $x(t, \theta) \in \mathbb{R}^{n_x}$ denotes the state vector at time $t \in \mathbb{T} = [t_0, t_f]$ and $\theta \in \mathbb{R}^{n_\theta}$ denotes the parameter vector. Usually, The parameters of such a model are kinetic constants, initial conditions and scaling factors, but can also define event related information such as thresholds, bolus, or event-times. The function $x_0 : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_x}$ provides the, possibly parameter dependent, initial condition at time t_0. The vector field $f : \mathbb{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_x}$ models the temporal evolution of the states and can have a piecewise definition.

Every model can include multiple different events which we distinguish by specifying n_e different event types. The jth event type is defined by a trigger function $g^{(j)}(t, x, \theta) : \mathbb{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ and update function $v^{(j)}(t, x, \theta) : \mathbb{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_x}$. The time point of the ℓth occurrence $\tau_\ell^{(j)}$ of the jth event type is defined by the roots of the trigger function $g^{(j)}$:

$$\forall \ell, j : \quad g^{(j)}(\tau_\ell^{(j)}, x(\tau_\ell^{(j)}, \theta), \theta) = 0 \text{ with } t_0 < \tau_1^{(j)} < \tau_2^{(j)} < \dots.$$

For Boolean trigger functions $g^{(j)}$, output values true are mapped to positive values and output values false are mapped to negative values such that $g^{(j)}$ has a root at every change of the Boolean value (see Supplementary Information, Section 2). The changes which are induced at every event occurrence are defined by the update function $v^{(j)}$:

$$x(\tau_{\ell,+}^{(j)}, \theta) - x(\tau_\ell^{(j)}, \theta) = v^{(j)}(\tau_\ell^{(j)}, x(\tau_\ell^{(j)}, \theta), \theta), \tag{2}$$

where $\tau_{\ell,+}^{(j)} = \lim_{\epsilon \to 0} \tau_\ell^{(j)} + \epsilon, \epsilon > 0$. If multiple different events occur simultaneously the cumulative change in the states is computed by summing over all triggered updates. The special case $v^{(j)} \equiv 0$ can be used to treat discontinuities in the vector field f or define events that only produce model outputs.

Model outputs describe the observed quantities in the measurement process (see Fig. 1a–c). In this manuscript, we consider time-resolved outputs y (Fig. 1b) and event-resolved outputs z (Fig. 1c). Time-resolved outputs y model measurements which are collected at predefined time points. As in most applications only partial observations are available, we introduce the output function $h(t, x, \theta) : \mathbb{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_y}$, which defines the time-resolved output $y(t, \theta) \in \mathbb{R}^{n_y}$ at measurement time point t:

$$y(t, \theta) = h(t, x(t, \theta), \theta). \tag{3}$$

The event-resolved outputs z describe measurements which are triggered by events and ordered accordingly. To allow for arbitrary event-resolved outputs, we introduce the output functions $\lambda(t, x, \theta)^{(j)} : \mathbb{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_z^{(j)}}$, which defines the event-resolved output for the ℓth occurence of the jth event type

$$z_\ell^{(j)}(\theta) = \lambda^{(j)}(\tau_\ell^{(j)}, x(\tau_\ell^{(j)}, \theta), \theta). \tag{4}$$

In the following we will derive the forward sensitivity equation for states and outputs. To improve the readability, we will partially omit the arguments of f, h, x_0, g, v and their respective partial derivatives with respect to t, x and θ. Where necessary, we will denote a pointwise evaluation by $|_\tau$, i.e. $\lambda^{(j)}|_\tau = \lambda^{(j)}(\tau, x(\tau, \theta), \theta)$.
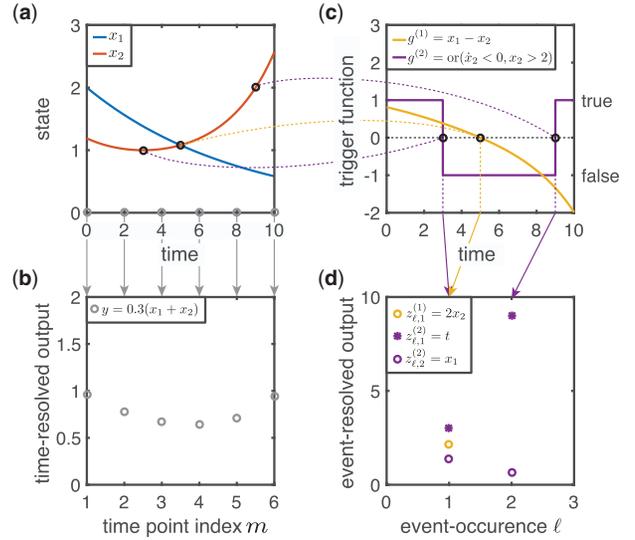

Fig. 1. Illustration of time-resolved and event-resolved outputs. (**a**) Time-dependent state of the process. (**b**) Time-resolved outputs pre-specified time points. (**c**) Trigger fuctions of the system. Events are triggered at the roots of the trigger functions. (**d**) Event-resolved outputs at event occurrences. The relation between elements of different subplots are indicated by arcs and arrows

2.2 Sensitivity equation for state and output variables

The dependence of the state variables on the parameter values is described by the sensitivity equations. The differential equations for the sensitivities $s_k^x(t, \theta) : \mathbb{T} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}^{n_x}$ of the states x to the parameter θ_k can be obtained by differentiating (1) with respect to θ_k and reordering the derivatives. This yields

$$\dot{s}_k^x = \frac{\partial f}{\partial x} s_k^x + \frac{\partial f}{\partial \theta_k}, \quad s_k^x(t_0, \theta) = \frac{\partial x_0}{\partial \theta_k}. \tag{5}$$

The events must also be taken into account as they can induce jumps in the solutions to the sensitivity equations. The formula for these jumps was originally derived by Rozenvasser (1967) and can be obtained by taking the derivative of (2) with respect to θ_k and reordering:

$$\begin{aligned}
s_k^x(\tau_{\ell,+}^{(j)}) - s_k^x(\tau_\ell^{(j)}) &= -(\dot{x}(\tau_{\ell,+}^{(j)}) - \dot{x}(\tau_\ell^{(j)})) s_k^{\tau_\ell^{(j)}} \\
&+ \frac{\partial v^{(j)}}{\partial x} \left(s_k^x + \dot{x} s_k^{\tau_\ell^{(j)}} \right) \bigg|_{\tau_\ell^{(j)}} + \frac{\partial v^{(j)}}{\partial t} s_k^{\tau_\ell^{(j)}} \bigg|_{\tau_\ell^{(j)}} + \frac{\partial v^{(j)}}{\partial \theta} \bigg|_{\tau_\ell^{(j)}},
\end{aligned}$$

where $s_k^{\tau_\ell^{(j)}}$ is the sensitivity of the event-time of the ℓth occurrence of the jth event type with respect to θ_k. The formula for this sensitivity follows from the implicit function theorem as:

$$s_k^{\tau_\ell^{(j)}} = \frac{\partial \tau_\ell^{(j)}}{\partial \theta_k} = -\left(\frac{\partial g^{(j)}}{\partial t} \right)^{-1} \frac{\partial g^{(j)}}{\partial \theta_k} \bigg|_{\tau_\ell^{(j)}}.$$

For the time-resolved outputs y as well as the event-resolved outputs z the sensitivities can be computed by applying the chain rule to Equations (3) and (4). The sensitivity of time-resolved output y_i, $i = 1, \dots, n_y$ at measurement time points t with respect to parameter θ_k is denoted by $s_k^{y_i}(t, \theta) : \mathbb{R} \times \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$. These sensitivities can be computed according to

$$s_k^{y_i} = \frac{\partial h_i}{\partial x} s_k^x + \frac{\partial h_i}{\partial \theta_k}.$$

The sensitivity of $z_{\ell,q}^{(j)}$, the qth out of $n_z^{(j)}$ event-resolved outputs of the ℓth occurrence of the jth event type with respect to θ_k, is denoted by $s_k^{z_{\ell,q}^{(j)}}(\theta) : \mathbb{R}^{n_\theta} \mapsto \mathbb{R}$ and can be computed according to

$$s_k^{z_{\ell,q}} = \frac{\partial \lambda_q^{(j)}}{\partial t} s_k^{\tau_\ell^{(j)}} + \frac{\partial \lambda_q^{(j)}}{\partial x} s_k^x + \frac{\partial \lambda_q^{(j)}}{\partial \theta_k} .$$

2.3 Objective function

We consider experimental data

$$\mathcal{D} = \left\{ \{t_m, \{\bar{y}_{m,i}\}_{i=1}^{n_y}\}_{m=1}^{n_t}, \{\bar{z}_{\ell,q}^{(j)}\}_{q=1,l=1,j=1}^{n_z^{(j)}, n_\tau^{(j)}, n_e} \right\} ,$$

q with n_t time points t_m at which n_y different measurements $\bar{y}_{m,i}$ are taken. Furthermore $\mathcal{D}$ includes the $n_z^{(j)}$ different outputs $\bar{z}_{q,j,l}$ for the $n_\tau^{(j)}$ occurrences of the jth out of n_e different event types. To estimate the parameters θ from $\mathcal{D}$, we use maximum likelihood estimation. Accordingly, we minimize the negative log-likelihood function $J(\theta)$, which provides a measure for the distance of the experimental data and the model simulation. In the following we will split the objective in two parts:

$$J(\theta) = J_y(\theta) + J_z(\theta) ,$$

with $J_y(\theta)$ to account for time-resolved data and $J_z(\theta)$ to account for event-resolved data.

We assume independent, normally distributed measurement noise with mean zero for time-resolved and event-resolved data. For time-resolved data, this yields the negative log-likelihood

$$J_y(\theta) = \frac{1}{2} \sum_{i=1}^{n_y} \sum_{m=1}^{n_t} \left(\frac{\bar{y}_{m,i} - y_i(t_m, \theta)}{\sigma_{m,i}^{(y)}(\theta)} \right)^2 + \log \left(2\pi \sigma_{m,i}^{(y)2}(\theta) \right) ,$$

with standard deviation of the time-resolved measurements $\sigma_{m,i}^{(y)}$. The negative log-likelihood function for event-resolved data is the sum of negative log-likelihood functions $\mathcal{J}_{\ell,q}^{z_{\ell,q}^{(j)}}$ for the individual measurements,

$$J_z(\theta) = \sum_{j=1}^{n_e} \sum_{q=1}^{n_z^{(j)}} \sum_{\ell=1}^{n_\tau^{(j)}} \mathcal{J}_{\ell,q}^{z_{\ell,q}^{(j)}} .$$

In accordance with the noise model, the obvious choice for $\mathcal{J}_{\ell,q}^{z_{\ell,q}^{(j)}}$ would be

$$\mathcal{J}_{\ell,q}^{z_{\ell,q}^{(j)}} = \frac{1}{2} \left(\frac{\bar{z}_{\ell,q}^{(j)} - z_{\ell,q}^{(j)}(\theta)}{\omega_{\ell,q}^{(j)}(\theta)} \right)^2 + \frac{1}{2} \log \left(2\pi \omega_{\ell,q}^{(j)2}(\theta) \right), \qquad (6)$$

with standard deviation of the event-resolved measurements $\omega_{\ell,q}^{(j)}$. The evaluation of $\mathcal{J}_{\ell,q}^{z_{\ell,q}^{(j)}}$, however, requires all event time points $\tau_\ell^{(j)}$ to be computed. For certain parameter values, these events might occur at very late time points $\tau_\ell^{(j)} \gg 1$ or not at all. To enable objective function evaluation for such parameter values, we set the maximum simulation time to a reasonable value $t_f > t_0$, at which all remaining event outputs are evaluated at t_f. As the gradient to the objective function for these remaining event outputs would be flat, we add an additional event penalization term based on the corresponding trigger function:

$$\mathcal{J}_{\ell,q}^{z_{\ell,q}^{(j)}} = \frac{1}{2} \left(\frac{\bar{z}_{\ell,q}^{(j)} - \lambda_q^{(j)}|_{\min(\tau_\ell^{(j)}, t_f)}}{\omega_{\ell,q}^{(j)}(\theta)} \right)^2 \\ + \frac{1}{2} \left(\frac{g^{(j)}|_{\min(\tau_\ell^{(j)}, t_f)}}{\omega_{q,q}^{(l)}(\theta)} \right)^2 + \frac{1}{2} \log \left(2\pi \omega_{\ell,q}^{(j)2}(\theta) \right). \qquad (7)$$

For $\tau_\ell^{(j)} > t_f$, $g^{(j)}(\min(\tau_\ell^{(j)}(\theta), t_f), x, \theta)$ acts as a proxy for the remaining time until event occurrence, while for $\tau_\ell^{(j)}(\theta) < t_f$ the penalty is

set to zero. This event penalization is only applied when the simulation yields less than the experimentally observed number of event occurrences n_τ^i. When the simulation yields more than the experimentally observed number of event occurrences n_τ^i, only the first n_τ^i are considered. The gradient of this objective function is provided in the Supplementary Information, Section 1.

2.4 Parameter optimization

For parameter estimation, the objective function $J(\theta)$ needs to be minimized. For this optimization we used the MATLAB function fmincon with the interior-point algorithm, a local optimizer. Other optimization schemes such as fmincon with the trust-region-reflective algorithm or lsqnonlin are also applicable, but require user-defined gradients. In contrast, the interior-point algorithm in MATLAB can compute finite difference gradients using a sophisticated method for automatic step-size selection and is thereby applicable to all settings considered in this manuscript. This facilitates a more consistent comparison.

For most applications, the minimization of the objective function is a non-convex optimization problem that possesses multiple local minima. As globalization strategy we initialized the optimization at 100 different starting points. Moreover, all parameters were log-transformed to increase optimization efficiency and ensure positivity of estimated parameters (Raue *et al.*, 2013). We allowed a maximum of 500 iterations and specified an objective function tolerance of 0 and an optimization variable tolerance of 10^{-8}.

2.5 Practical identifiability

Measurement data are limited and noise-corrupted, potentially rendering model parameter non-identifiable. In this study, we assess practical identifiability using profile likelihoods (Kreutz *et al.*, 2013), a method which is also applicable to ODE models with discrete events and logical operators. Profile likelihoods provide parameter confidence intervals by solving a sequence of constrained local optimization problems.

2.6 Implementation

We implemented the numerical simulation of ODE models with events in the MATLAB toolbox AMICI (Fröhlich *et al.*, 2016). AMICI uses CVODES for numerical simulation. In AMICI events can be directly defined in the vector field f. Several discontinuous or non-differentiable functions can be used in the model definition in AMICI (see Supplementary Information, Section 2), including Heaviside and Dirac delta distributions. These functions are automatically parsed by AMICI and transformed into appropriate event definitions. Additionally, events can also be specified in the event field of the model definition struct via the amievent(trigger,update,z) command or imported from SBML models via libSBML (Bornstein *et al.*, 2008).

3 Results

In the following, we illustrate and evaluate the accuracy of the numerical simulation of ODE models with events and the benefit of accurate sensitivities for parameter estimation. For this purpose, we consider one application and one simulation example.

3.1 Application example: mRNA transfection

In this section, we evaluate the numerical accuracy of the simulation for a model with a parameter-dependent state transition. Furthermore, we estimate the model parameters from published time-resolved data (Leonhardt *et al.*, 2014).

3.2 Model of mRNA transfection

We consider a model for the transfection of a cell with mRNA coding for GFP (Fig. 2a). The mRNA coding for GFP is encapsulated in lipoplexes and added to the extracellular medium. The lipoplexes are absorbed via endocytosis and dissolved, resulting in the release of GFP mRNA which becomes accessible for translation (Ligon et al., 2014). The release of mRNA into the cytoplasm is modeled as a parameter dependent event, where the parameters describe the time point of mRNA release and the amount of released mRNA. Both mRNA and protein are degraded. The process is described by the following ODE:

$$\frac{dx_1}{dt} = -\beta x_1, \qquad x_1(0) = 0,$$
$$\frac{dx_2}{dt} = k_2 x_1 - \gamma x_2, \qquad x_2(0) = 0,$$
$$g^{(1)} = t - t_r, \qquad v^{(1)} = [m_0, 0]^T,$$

where x_1 denotes the mRNA concentration, x_2 denotes the GFP concentration, β is the mRNA degradation rate, m_0 is the released amount of mRNA, t_r is the time point of release, k_2 is the translation rate and γ is the degradation rate of GFP. For optimization, the model is reparametrized such that all remaining parameters are structurally identifiable (see Supplementary Information, Section 3).

The GFP expression after mRNA transfection has been assessed using single cell time-lapse fluorescence microscopy (Leonhardt et al., 2014). These measurements are corrupted by background fluorescence b. Furthermore, as our analysis revealed that the measurement noise is additively normally distributed on the log-scale, we consider the output $y = \log(sx_2 + b)$.

3.3 Accuracy of numerical simulation and sensitivity analysis

The model for mRNA transfection is linear and can be solved analytically (see Supplementary Information, Section 3). To verify the implementation in AMICI, we compared the numerically computed sensitivities to their analytical counterparts. This revealed the high accuracy of the simulation results (see Fig. 2b).

3.4 Improvement of parameter optimization for time-resolved data using sensitivity equations

The parameters influencing mRNA transfection are not experimentally accessible. Here, we estimated those parameters from a representative single-cell trajectory reported by Leonhardt et al. (2014). To evaluate the importance of accurate gradient information, we performed multi-start local optimization using sensitivity equations and finite differences for gradient calculation. For both methods, local optimization was initialized at the same 100 starting points.

The assessment of the optimization results revealed that both methods yielded fits which are consistent with the experimental data (Fig. 3a). To evaluate the efficiency of the two approaches we compared the computation time per local optimization. We found that the sensitivity based approach required on average 0.67 s per optimization while the finite difference based approach required on average 1.22 s (Fig. 3b). To evaluate the robustness of the two approaches we compared the consistency with which the lowest objective function value was achieved. For both approaches we observe pronounced plateaus in the final objective function value (Fig. 3c), implying good convergence of both approaches. In a fixed timespan, the sensitivity based approach will yield 2.2 times as many converged starts as the finite differences based approach (Fig. 3d). These results suggest that already for small ODE models with events, the sensitivity based approach is more efficient than the finite difference based approach.

3.5 Simulation example: spiking neurons

In this section, we use the numerical simulation and optimization methods to study a system with state-dependent trigger function and event-resolved data. Such models are widely used in computational neurosciences. In this section we will consider a model for the membrane potential in neurons introduced by Izhikevich (2003). The model can describe a variety of spiking and bursting patterns of the membrane potential which are typical for all known types of cortical neurons. For the model of spiking neurons the after-spike reset is modeled as an event with a state-dependent trigger function (see Fig. 2c).
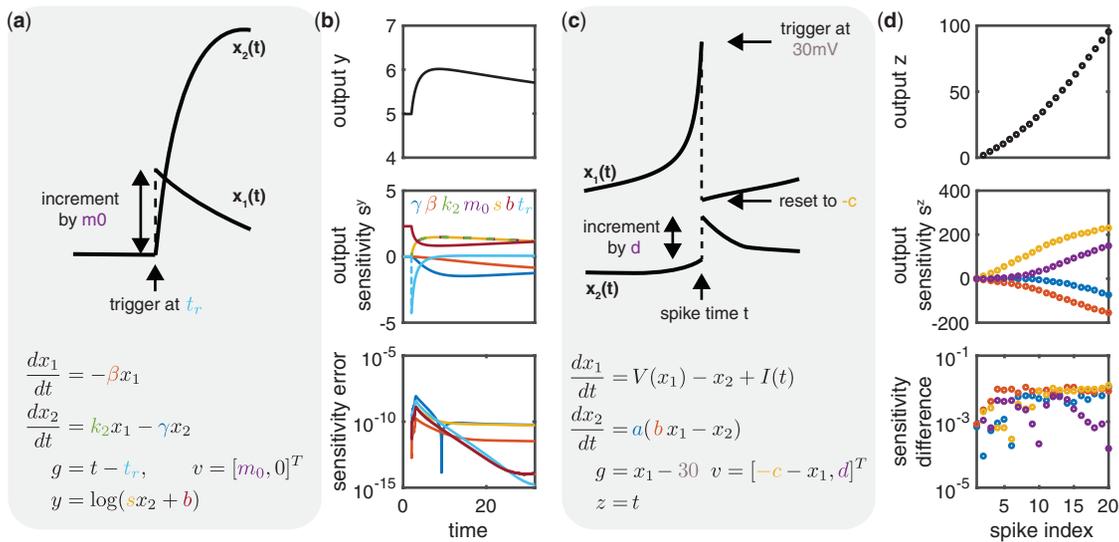

Fig. 2. Illustration and simulation for the model of mRNA transfection and Izhikevich model. (**a**) Event description and mathematical formulation for the model of mRNA transfection. (**b**) Time-resolved output and comparison of the numerically calculated sensitivities with the respective analytical results. (**c**) Event description and mathematical formulation for the Izhikevich model. (**d**) Event-resolved output and comparison of respective sensitivity equation based and finite differences based sensitivities. The numerical integration was carried out with an absolute tolerance of 10^{-8} and a relative tolerance of 10^{-8}. The step-size for finite differences was 10^{-5}
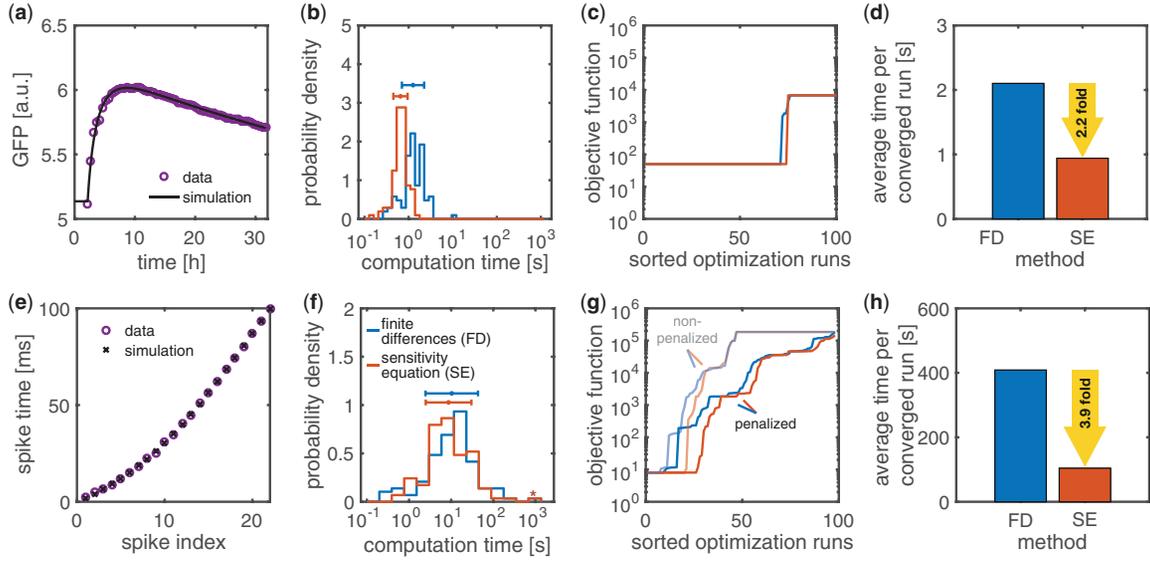
Fig. 3. Optimization results for the model of mRNA transfection and for the model of spiking neurons. (**a,e**) Fitted output and experimental data. (**b,f**) Comparison of optimization times for different initializations of the local optimizer. Outliers are indicated by stars and added to the respective outermost bin. (**c,g**) Comparison of sorted objective function values for different initializations of the local optimizer. (**d,h**) Comparison of effective time per converged optimization run obtained using the penalization of missing events

3.6 Model of spiking neurons

We consider the Izhikevich model for spiking neurons (Izhikevich, 2003). This model describes the membrane potential x_1 of a neuron, which is influenced via a negative feedback by the membrane recovery variable x_2. The ODEs are:

$$\dot{x}_1 = V(x_1) - x_1 + I(t)$$
$$\dot{x}_2 = a(b \cdot x_1 - x_2)$$
$$g^{(1)} = x_1 - 30$$
$$v^{(1)} = [-c - x_1, d]^T,$$

where $V(x) = 0.04x^2 + 5x + 140$ defines the spike initiation dynamics and $I(t) = 10 \cdot H(t - 1)$ describes the induction current. The parameter a defines the time-scale of the recovery variable x_2, b defines the sensitivity of the recovery variable on the membrane potential, c defines the after-spike reset value of the membrane potential v and d describes the after-spike reset of the recovery variable x_2. The event implements the after-spike reset when the potential x_1 exceeds the threshold 30 and resets the value of the membrane potential x_1 to c and increases the value of the recovery variable x_2 by d. In experiments usually the time points of spikes, so called spike trains are recorded. Therefore, we specify the time point of the event as event-output: $\lambda^{(1)} = t$. As for the considered process no analytical solution is known, we verified the correctness of sensitivity equations for this event-resolved output by comparing the simulation results to the sensitivities computed via finite differences (see Fig. 2d). As the finite differences results are noise-corrupted, this comparison does not yield an exact quantification of the error.

3.7 Improvement of parameter optimization for event-resolved data using sensitivity equations and event penalization

For the model of spiking neurons we generated synthetic data by simulating the model for $\theta = (a, b, c, d)$ with $a = 0.02$, $b = 0.3$, $c = 65$ and $d = 0.9$ and reporting the first 22 spike-times. The spike-times were perturbed by adding Gaussian noise with standard deviation $\sigma_\tau = 0.5$, yielding the artificial data.

Several toolboxes for fitting of spike-train data have been developed (e.g. Rossant *et al.*, 2011). None of these toolboxes exploit sensitivity equations for gradient computation although sensitivity based methods are known to be more efficient for a variety of applications in systems biology (Raue *et al.*, 2013). To assess whether the use of sensitivity equations for gradient calculation is also advantageous for time-to-event fitting we carried out parameter estimation on spike trains using both finite differences and sensitivity equations. The comparison of model simulation and experimental data (Fig. 3e) revealed that both methods yield a best fit which agrees well with the experimental data.

To evaluate the efficiency of the two approaches, we compared the computation time per optimization. We found that for the neuron model the sensitivity based approach required on average 8.2 s per local optimization while the finite difference based approach required on average 11.3 s (Fig. 3f). To evaluate the robustness of the two approaches we compared the consistency with which the lowest objective function value was achieved. For forward sensitivities three times as many optimization runs converged to the lowest objective value of 8.05 (Fig. 3g), implying better convergence for the sensitivity based approach. In a fixed timespan, the sensitivity based approach will yield 3.9 times as many converged starts as the finite differences based approach (Fig. 3h). When the event penalization (7) is not employed for the objective function, over 50% of the optimization runs stop after the first iteration as no event occurred and the gradient was evaluated to zero. These optimization runs resulted in a pronounced plateau in the objective function value at $1.85 \cdot 10^5$. This demonstrates that optimization using sensitivity equations for time-ordered data is more reliable and more efficient compared to finite differences and that event penalization is essential for optimization.

Assessment of practical identifiability for event-resolved data

To assess the identifiability of parameters for the model for spiking neurons from event-resolved data, we computed confidence intervals for all parameters using the profile likelihood method. This analysis
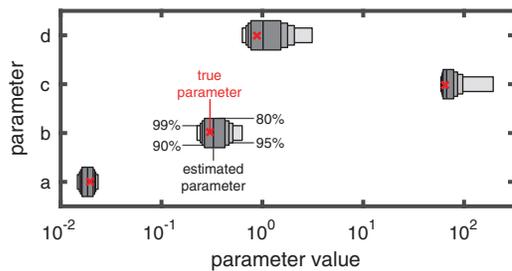
Fig. 4. Identifiability results for the neuron model. Profile likelihood confidence levels for the four parameters a, b, c and d of the model for spiking neuronsat four different confidence levels 0.99, 0.98, 0.9 and 0.8. The thickness of the bars corresponds to the confidence level, the width of the bars corresponds to the extent of the confidence interval. All parameters are identifiable

revealed that all parameters are practically identifiable (Fig. 4). The confidence intervals contain the parameter values used to generate the data. This emphasizes the reliability of the proposed method for handling event-resolved data.

4 Discussion

Discrete events and logical operators are frequently used in system biology to describe involved dynamics and are particular relevant in multi-scale modeling. For models with multiple time-scales, the fast processes can be described as Boolean model and linked to the slow processes via events, giving rise to hybrid models (Le Novère, 2015; McAdams and Shapiro, 1995). In this manuscript, we formulated the sensitivity equations for time-resolved and event-resolved outputs for ODE models with discrete events and logical operations. These sensitivity equations provide means to efficiently and accurately compute objective function gradients, which are essential for parameter estimation. All methods are provided to the systems and computational biology community in the easy-to-use software toolbox AMICI. AMICI provides full support for sensitivity calculation for models with discrete events and logical operations. In contrast to available toolboxes, event-triggered observations are considered and a penalization strategy for missing events is provided to facilitate observations. AMICI has been tested on Windows, Unix and Mac OS X based operating systems. Due to its modular implementation, AMICI can be easily integrated with other toolbox to provide simulation results and sensitivities to various systems biology toolboxes. The import of SBML models provides access to a large number of models stored in databases, such as BIOMODELS (Le Novère *et al.*, 2006) and allows for model editing using tools with graphical user interfaces, such as COPASI (Hoops *et al.*, 2006)).

We compared the performance of multi-start local optimization using sensitivity equations and finite differences for gradient evaluation for one simulation and one application example. For both examples we found that the optimization has better convergence and is faster when using sensitivity equations compared to finite differences. These results are in line with previous comparisons of sensitivity based and finite difference based parameter estimation in systems biology (Hross and Hasenauer, 2016; Raue *et al.*, 2013).

Optimization and practical identifiability analysis for the model of a spiking neuron revealed that estimation for event-resolved data is practically feasible and can yield robust parameter estimates. For the investigated model, all parameters were practically identifiably solely from event-resolved data. The direct examination of structural identifiability of parameters remains elusive as existing methods are not applicable.

The presented framework enables the consideration of event-resolved observations, such as event-times. Time to event measurements for cellular events such as transfection, differentiation, division or apoptosis could provide additional insight into the underlying process when described by mechanistic models. Moreover, by using events with Boolean trigger functions (i.e. derived from Signal Temporal Logic (Donzé and Maler, 2010) formulas), transitions in the qualitative behaviour (Toni *et al.*, 2011) of a system could be used for parameter estimation by considering the respective transition time-points as event-resolved observables.

In conclusion, the developed methods facilitate the simulation and sensitivity analysis for a more general class of ODE models and make event-resolved data accessible for parameter estimation. This will facilitate the analysis of more involved models and the mechanistic description of cellular events in a variety of novel applications.

References

Balsa-Canto,E. *et al.* (2010) An iterative identification procedure for dynamic modeling of biochemical networks. *BMC Syst. Biol.*, **4**,

Barton,P.I. *et al.* (1998) Dynamic optimization in a discontinuous world. *Ind. Eng. Chem. Res.*, **37**, 966–981.

Bornstein,B.J. *et al.* (2008) LibSBML: An API library for SBML. *Bioinformatics*, **24**, 880–881.

Dai,W. *et al.* (2014) Parameter set selection for signal transduction pathway models including uncertainties. In: *Proc. Of the 19th IFAC World Congress*, Cape Town, South Africa, pp. 815–820.

de Lomana,A.G. *et al.* (2007) User reference manual of byodyn version 4.1. Technical report, Barcelona Biomedical Research Park.

Dierkes,T. *et al.* (2011) BioPARKIN - Biology-related parameter identification in large kinetic networks. Technical Report 11-15, ZIB, Takustr.7, 14195 Berlin.

Donzé,A. and Maler,O. (2010). *Robust Satisfaction of Temporal Logic over Real-Valued Signals*, pp. 92–106. Springer, Berlin, Heidelberg.

Egea,J.A. *et al.* (2014) MEIGO: an open-source software suite based on meta-heuristics for global optimization in systems biology and bioinformatics. *BMC Bioinf.*, **15**, 136. (

Fröhlich,F. *et al.* (2016) Scalable parameter estimation for genome-scale biochemical reaction networks. *bioRxiv*, 089086.

Geissen,E.M. *et al.* (2016) MEMO – multi-experiment mixture model analysis of censored data. *Bioinformatics*, **32**, 2464–2472.

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

Gonnet,P. *et al.* (2012) A specialized ODE integrator for the efficient computation of parameter sensitivities. *BMC Syst. Biol.*, **6**, 46.

Hindmarsh,A.C. *et al.* (2005) SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM T. Math. Softw.*, **31**, 363–396.

Hirmajer,T. *et al.* (2009) Dotcvpsb, a software toolbox for dynamic optimization in systems biology. *BMC Bioinf.*, **10**, 1–14.

Holford,N. (2013) A time to event tutorial for pharmacometricians. *CPT Pharmacometrics Syst. Pharmacol.*, **2**, e43.

Hoops,S. *et al.* (2006) COPASI – a COmplex PAthway SImulator. *Bioinformatics*, **22**, 3067–3074.

Hross,S. and Hasenauer,J. (2016) Analysis of CFSE time-series data using division-, age- and label-structured population models. *Bioinformatics*, **32**, 2321–2329.

Hucka,M. *et al.* (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.

Izhikevich,E.M. (2003) Simple model of spiking neurons. *IEEE Trans. Neural Netw.*, **14**, 1569–1572.

Kitano,H. (2002) Computational systems biology. *Nature*, **420**, 206–210.

Klipp,E. *et al.* (2005). *Systems Biology in Practice*. Wiley-VCH, Weinheim.

Kreutz,C. *et al.* (2013) Profile likelihood in systems biology. *FEBS J.*, **280**, 2564–2571.

Le Novère,N. (2015) Quantitative and logic modelling of molecular and gene networks. *Nat. Rev. Genet. Genet.*, **16**, 146–158.

Le Novère,N. *et al.* (2006) BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, **34**, D689–D691.

Leonhardt,C. *et al.* (2014) Single-cell mRNA transfection studies: delivery, kinetics and statistics by numbers. *Nanomed. Nanotechnol. Biol. Med.*, **10**, 679–688.

Ligon,T.S. *et al.* (2014) Multi-level kinetic model of mRNA delivery via transfection of lipoplexes. *PLoS ONE*, **9**, 1–11.

Liu,G. and Neelamegham,S. (2008) In silico biochemical reaction network analysis (ibrena): a package for simulation and analysis of reaction networks. *Bioinformatics*, **24**, 1109–1111.

Lu,J. *et al.* (2008). SBML ODE solver library: Extensions for inverse analysis. In: *Proceedings of the Fifth International Workshop on Computational Systems Biology, WCSB*.

McAdams,H. and Shapiro,L. (1995) Circuit simulation of genetic networks. *Science*, **269**, 650–656.

Myers,C.R. *et al.* (2007) Python unleashed on systems biology. *Comput. Sci. Eng.*, **9**, 34–37.

Olivier,B.G. *et al.* (2005) Modelling cellular systems with pysces. *Bioinformatics*, **21**, 560–561.

Perkel,D.H. *et al.* (1967) Neuronal spike trains and stochastic point processes: I. the single spike train. *Biophys. J.*, **7**, 391–418.

Raue,A. *et al.* (2009) Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, **25**, 1923–1929.

Raue,A. *et al.* (2011) Addressing parameter identifiability by model-based experimentation. *IET. Syst. Biol.*, **5**, 120–130.

Raue,A. *et al.* (2013) Lessons learned from quantitative dynamical modeling in systems biology. *PLoS ONE*, **8**, e74335.

Raue,A. *et al.* (2015) Data2Dynamics: a modeling environment tailored to parameter estimation in dynamical systems. *Bioinformatics*, **31**, 3558–3560.

Roberts,S. (2006). Modelling and simulation of biochemical pathways. In: *Proceedings of the Fourth International Workshop on Computational Systems Biology, WCSB*, p. 11.

Rodriguez-Fernandez,M. and Banga,J.R. (2010) Senssb: a software toolbox for the development and sensitivity analysis of systems biology models. *Bioinformatics*, **26**, 1675–1676.

Rossant,C. *et al.* (2011) Fitting neuron models to spike trains. *Front. Neurosci.*, **5**, 9.

Rozenvasser,E.N. (1967) General sensitivity equations of discontinuous systems. *Avtomat I Telemekh*, **3**, 52–56.

Schilling,M. *et al.* (2009) Theoretical and experimental analysis links isoform-specific ERK signalling to cell fate decisions. *Mol. Syst. Biol.*, **5**, 334. (

Schmidt,H. and Jirstrand,M. (2006) Systems biology toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics*, **22**, 514–515.

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

Somogyi,E.T. *et al.* (2015) libRoadRunner: A high performance SBML simulation and analysis library. *Bioinformatics*, **31**, 3315–3321.

Toni,T. *et al.* (2011) From qualitative data to quantitative models: analysis of the phage shock protein stress response in *Escherichia coli. BMC Syst. Biol.*, **5**, 69. (

Vanlier,J. *et al.* (2012) A Bayesian approach to targeted experiment design. *Bioinformatics*, **28**, 1136–1142.

Zi,Z. *et al.* (2008) SBML-SAT: a systems biology markup language (SBML) based sensitivity analysis tool. *BMC Bioinf.*, **9**, 1–14.