Forward Sensitivity Equations in the Presence of Events

Daniel Kaschek* and Matthew Fidler†
*IntiQuan GmbH, Basel, Switzerland †Novartis, Fort Worth TX, USA

***Abstract*—Forward sensitivity equations are frequently used in optimization. Based on forward sensitivities, the gradient and Hessian of the least squares function can be derived which allow to use gradient-based optimization methods. In the presence of events, i.e. sudden changes of the state variables, the differential equations and corresponding sensitivity equations are structurally unchanged. However, the events on states need to be accounted for as events in the sensitivities. Here, we derive these necessary events and present them in a way that helps with the implementation in computational software. Finally, the sensitivity events are illustrated on a typical example from the field of pharmacometrics.**

I. Introduction

EVENTS frequently are used in ODE systems. These events include intervention events such as a dose or infusion, or process events like bile dumping and gastric emptying. These events change ordinary differential equations (ODE) by changing its states. Some of the more interesting system events like zero order release, gastric emptying and bioavailability are not known *a-priori*. Often these events should be estimated from available data.

Like many data-based estimation methods we may have an initial guess about when and how these events occur. But we want to find the best solution by optimization to the data. This most often performed while trying to estimate some other processes and parameters of the ODE system.

Like many optimization problems with initial conditions, the system needs to maximize the likelihood surface based on the next best step. The directions to the best location is provided by the gradient. With ODEs one way to calculate the is the forward sensitivity equations.

Often when forward sensitivity analysis is calculated it is done without considering the events that are estimated. This has been more often handled by simple but inaccurate numerical derivatives, but less often by a formal sensitivity analysis (Ref). A formal sensitivity analysis adds accuracy and often speeds up computation.

Exact forward sensitivities of these events have been calculated, called jump sensitivities. These jump sensitivities depend on other events making it more difficult to calculate the forward sensitivity for these events easily in optimization. In one optimization example, the next event time needs to be known before the event based sensitivities are calculated.

However, a simple linearization allows the event sensitivities to depend only on the event itself. This simplification adds minuscule to no loss in accuracy in the point derivative at the event time. Additionally, this will also speed up computational time because the next event times do not need to be calculated for optimization. The cost of this method is to introduce new events to the sensitivity states already calculated.

Because of these advantages, we would like to share this new method of calculating jump sensitivities.

II. Main Part

Let $\dot{x} = f(x, p, t)$ be a system of ordinary differential equations (ODEs) with the states $x(t) \in \mathbb{R}^n$, parameters $p \in \mathbb{R}^m$ and time $t \in \mathbb{R}$. The sensitivity equations corresponding to this dynamic system are

$$\begin{aligned} \frac{d}{dt}\frac{\partial x_i}{\partial p_j} &= \frac{d}{dp_j} f_i(x, p, t) \\ &= \frac{\partial f_i}{\partial x_k}\frac{\partial x_k}{\partial p_j} + \frac{\partial f_i}{\partial p_j}. \end{aligned} \tag{1}$$

Let us assume that an event occurs at time t_e that changes the current state vector $x(t_e)$ to the value $v_e \in \mathbb{R}^n$. Both t_e and v_e are assumed to be parameters for which sensitivities are to be determined.

Because changing $x(t_e)$ is a singular event in time, it does not change the structure of the ODEs. They are the same before and after the event time. Therefore, also the sensitivity equations, eq. (1), are structurally unchanged. However, the sensitivities themselves are affected by jumps at event time t_e, as being shown in this section.

The derivation is based on linearization of the ODE around $x(t_e) = x_e$, this is the value of $x(t)$ at the latest time point before the event occurs. With this choice, the linearized ODE reads

$$\dot{x} \doteq A(p, t)x + b(p, t) \tag{2}$$

with

$$\begin{aligned} A(p, t) &= \left.\frac{\partial f}{\partial x}\right|_{x_e}(p, t) \\ b(p, t) &= f(x_e, p, t) - \left.\frac{\partial f}{\partial x}\right|_{x_e}(p, t)x_e. \end{aligned}$$

The general solution of eq. (2) is

$$x(t) = \Phi(t)x_0 + \Phi(t)\int_0^t \Phi^{-1}(\tau)b(\tau)d\tau, \quad t \le t_e, \tag{3}$$

where $\Phi(t) = \big(\varphi_1(t), \ldots, \varphi_n(t)\big)$ is the matrix of linearly independent solutions $\varphi_i(t)$ of the homogeneous part of eq. (2), i.e., $\forall i:\ \dot{\varphi}_i = A(p, t)\varphi_i$ with initial condition $\varphi_i(0) = \mathrm{e}_i$, the i'th unit vector.

First, we note that the original and linearized ODE's have the same sensitivity equations. The jumps of the sensitivities that we derive based on eq. (2) are therefore valid for the sensitivities of the original ODE, too. Second, the solution after the event time t_e can be explicitly stated as

$$x(t) = \Phi(t - t_e)v_e + \Phi(t) \int_{t_e}^{t} \Phi^{-1}(\tau)b(\tau)d\tau, \quad t > t_e. \quad (4)$$

A. Sensitivities with respect to t_e

Based on the explicit solutions, eqs. (3) and (4), we find that

$$\left.\frac{\partial x}{\partial t_e}\right|_{t=t_e} = \begin{cases} 0 & \text{, for } t \nearrow t_e \\ -Av_e + \frac{\partial v_e}{\partial t_e} - b(t_e) & \text{, for } t \searrow t_e, \end{cases} \quad (5)$$

where we have used that $\Phi(0) = \mathbb{I}$ is the unit matrix. The sensitivities with respect to t_e jump at time point t_e. For $t > t_e$, the sensitivities propagate forward in time according to the sensitivity equations. The jump equation contains the derivative of v_e which depends on the kind of the event.

a) Replacement: The value of the i'th state, x_i, is set to a predefined value $v_{e,i}$. In that case, the derivative vanishes and the sensitivity at time t_e is

$$\lim_{t \searrow t_e} \frac{\partial x_i}{\partial t_e} = \left.\frac{\partial f_i}{\partial x}\right|_{x_e} \cdot (x_e - v_e) - f_i(x_e), \quad (6)$$

where we have used the definition of $b(t_e)$. In the equation, $\cdot : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ denotes the scalar product. While the state value x_i jumps, the other states x_j, $j \neq i$, are continued. Continuation means that $v_{e,j}$ is set to $x_{e,j}$ and we need eq. (3) evaluated at t_e to get $\frac{\partial v_e}{\partial t_e} = Ax_e + b(t_e)$. The sensitivities become

$$\lim_{t \searrow t_e} \frac{\partial x_j}{\partial t_e} = \left.\frac{\partial f_j}{\partial x}\right|_{x_e} \cdot (x_e - v_e). \quad (7)$$

The contributions from b cancel out. Due to the fact that all states except for x_i are continued, the scalar product reduces to a single term. However, we find that an event at time t_e affects sensitivities with respect to t_e of both the affected and unaffected states.

b) Additive: A constant Δx_i is added to the value of the i'th state, x_i. In that case, v_e is set to eq. (3) evaluated at t_e plus the constant Δx. Therefore, the same argumentation as in the continued case holds for both, the affected and unaffected states, and the sensitivities are as eq. (7) for states i and $j \neq i$.

c) Multiplicative: The value of the i'th state, x_i, is multiplied with a constant α_i. Eq. (3) is evaluated at t_e and multiplied by α to get v_e. Consequently, the derivative $\frac{\partial v_e}{\partial t_e}$ is computed and plugged into eq. (5), yielding

$$\lim_{t \searrow t_e} \frac{\partial x_i}{\partial t_e} = \left.\frac{\partial f_i}{\partial x}\right|_{x_e} \cdot (x_e - v_e) - (1 - \alpha_i)f_i(x_e). \quad (8)$$

Same as for replacement events, the sensitivity of the affected and the unaffected states need to be distinguished. The term $(1 - \alpha_i)f_i(x_e)$ only occurs for the affected state. Futhermore, all states except x_i are continued. The scalar product therefore reduces to the i'th contribution.

B. Sensitivities with respect to v_e

We use again the explicit solutions, eqs. (3) and (4), and find that

$$\left.\frac{\partial x}{\partial v_e}\right|_{t=t_e} = \begin{cases} 0 & \text{, for } t \nearrow t_e \\ \mathbb{I} & \text{, for } t \searrow t_e. \end{cases} \quad (9)$$

Again, the sensitivities jump. They are forward propagated according to their sensitivity equations. Based on eq. (9), we can construct sensitivities in case of additive events, where $v_e = x_e + \Delta x$ with $\Delta x \in \mathbb{R}^n$, and multiplicative events, where $v_{e,i} = \alpha_i x_{e,i}$ with $i = 1, \ldots, n$. This is:

$$\left.\frac{\partial x}{\partial \Delta x}\right|_{t=t_e} = \mathbb{I}, \qquad \left.\frac{\partial x}{\partial \alpha}\right|_{t=t_e} = \mathrm{diag}(x_e). \quad (10)$$

C. Sensitivities with respect to p

In the above sections we have derived expressions for the jumps of sensitivities with respect to event parameters. In this section we show that also sensitivities with respect to other parameters are affected by the events.

Let $S_j(t) = \frac{\partial x}{\partial p_j}(t) \in \mathbb{R}^n$ be the sensitivities with respect to p_j as derived from the solution for $t \leq t_e$, eq. (3). On the other hand, the right-sided limit $t \searrow t_e$ of $\frac{\partial x}{\partial p_j}$ as derived from eq. (4) yields

$$\lim_{t \searrow t_e} \frac{\partial x}{\partial p_j} = \frac{\partial v_e}{\partial p_j}. \quad (11)$$

Based on eq. (11) we construct sensitivities in case of replacement, additive, and multiplicative events. States i which are set to $v_{e,i}$ by an event have $\lim_{t \searrow t_e} \frac{\partial x_i}{\partial p_j} = 0$. States i affected by an additive event, $v_{e,i} = x_{e,i} + \Delta x_i$, have $\lim_{t \searrow t_e} \frac{\partial x_i}{\partial p_j} = \frac{\partial x_{e,i}}{\partial p_j} = S_{ij}(t_e)$, i.e., the sensitivities are continued. States i affected by a multiplicative event, $v_{e,i} = \alpha_i x_{e,i}$, have $\lim_{t \searrow t_e} \frac{\partial x_i}{\partial p_j} = \alpha_i S_{ij}(t_e)$, i.e., they are multiplied with the same constant α_i as the state value x_i.

III. Implementation

When working with events in numerical ODE solvers, events are typically specified by

var	time	value	method

where "var" denotes the state variable, "time" is the time point at which the event occurs, "value" is the event value and "method" is either replace, add or multiply.

Using the results from Section II, we are going to derive the set of events that apply for the sensitivities.

Without loss of generality, let "var" be x_1. The time and value parameters are denoted as τ and ξ, and the value be replaced. This event with the required additional events is shown in Table I. In the table, j ranges from 1 to m, J_{11} denotes the (1, 1)-element of the Jacobian $J_{ij} = \lim_{t \nearrow \tau} \frac{\partial f_i}{\partial x_j}(x(t), p, t)$ and f_1 denotes the first element of $f_i := \lim_{t \nearrow \tau} f_i(x(t), p, t)$. The differences $x_e - v_e$ are all zero except for the first state. Therefore, the scalar products in eqs. (6) and (7) reduce to one contribution. Note that, depending on the code implementation of the events, the symbolic expressions for J_{11} and f_1 can be

TABLE I
REPLACEMENT EVENT AND LIST OF REQUIRED ADDITIONAL SENSITIVITY EVENTS.

1	x_1	τ	ξ	replace
2	$\partial x_1/\partial p_j$	τ	0	replace
3	$\partial x_1/\partial \tau$	τ	$J_{11}(x_1 - \xi) - f_1$	add
4	$\partial x_k/\partial \tau$	τ	$J_{k1}(x_1 - \xi)$	add
5	$\partial x_1/\partial \xi$	τ	1	add

used and evaluated within the event function based on the state values at the event time **before** execution of the event.

Lines 3 and 4 have method = add because these contributions need to be accumulated when several events occur.

In a similar way, we derive the table for additive events, see Table II. In the table, δ denotes the value which is added to x_1 at time τ. The event in line 2 could be omitted. It is shown in the table to explicitly show that $\frac{\partial x_1}{\partial p_j}$ is not affected.

TABLE II
ADDITIVE EVENT AND LIST OF REQUIRED ADDITIONAL SENSITIVITY EVENTS.

1	x_1	τ	δ	add
2	$\partial x_1/\partial p_j$	τ	0	add
3	$\partial x_1/\partial \tau$	τ	$-J_{11}\delta$	add
4	$\partial x_k/\partial \tau$	τ	$-J_{k1}\delta$	add
5	$\partial x_1/\partial \delta$	τ	1	add

Finally, the table for multiplicative events is derived, see Table III. In the table, α denotes the value by which x_1 is multiplied at time τ.

TABLE III
MULTIPLICATIVE EVENT AND LIST OF REQUIRED ADDITIONAL SENSITIVITY EVENTS.

1	x_1	τ	α	multiply
2	$\partial x_1/\partial p_j$	τ	α	multiply
3	$\partial x_1/\partial \tau$	τ	$(1-\alpha)(J_{11}x_1 - f_1)$	add
4	$\partial x_k/\partial \tau$	τ	$(1-\alpha)J_{k1}x_1$	add
5	$\partial x_1/\partial \alpha$	τ	x_1	add

When implementing the additional events, one should keep in mind that:

1) States should be changed by the event after changing the sensitivities. Otherwise expressions like $x_1 - \xi$ can accidentally vanish.
2) Events occurring at the same time point should be implemented as if the occurred at different time points, i.e., changing sensitivities and states in alteration.

IV. EXAMPLE

The event sensitivities are illustrated on a simple pharmacokinetic-pharmacodynamic (PK/PD) model. The PK and PD are described by a one compartment first order absorption model and an inhibitory IMAX model, respectively. The equations are

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Ad} = \mathrm{Favail} \cdot \mathrm{Input} - \mathrm{KA} \cdot \mathrm{Ad} \tag{12}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Ac} = \mathrm{KA} \cdot \mathrm{Ad} - \frac{\mathrm{CL}}{\mathrm{V}}\mathrm{Ac} \tag{13}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Input} = 0 \tag{14}$$

$$\mathrm{Effect} = \mathrm{E0} \cdot \left(1 - \frac{\frac{\mathrm{Ac}}{\mathrm{V}} \cdot \mathrm{IMAX}}{\mathrm{IC50} + \frac{\mathrm{Ac}}{\mathrm{V}}}\right). \tag{15}$$

All states and parameters are characterized in Table IV. The input is switched on and off by events:

TABLE IV
OVERVIEW OF MODEL STATES AND PARAMETERS.

	Description	(Initial) value
Ad	Drug compartment	0
Ac	Central compartment	0
Input	Dosing input	0
Favail	Bioavailability	1
KA	Transfer rate	1
CL	Clearance rate	6
V	Central volume	60
E0	Baseline effect	15
IMAX	Maximal inhibition	1
IC50	Half maximal conc.	1

1	Input	τ_1	r_1	replace
2	Input	τ_2	0	replace

Typically, the dosing is parameterized in terms of the lag time (tlag), the duration of the infusion (tinf) and the drug dose (dose). The event parameters τ_1, τ_2 and r_1 are expressed as functions of the dosing parameters:

$$\tau_1 = \mathrm{tlag}, \quad \tau_2 = \mathrm{tlag} + \mathrm{tinf}, \quad r_1 = \frac{\mathrm{dose}}{\mathrm{tinf}}, \tag{16}$$

with $\mathrm{tlag} = 10$, $\mathrm{tinf} = 10$, $\mathrm{dose} = 200$. The model, eqs. (12–14), has been simulated between $t = 0$ and $t = 60$ alongside the sensitivity equations. Subsequently, the response in the effect compartment, eq. (15), was evaluated based on the solution of the ODE. To illustrate the event sensitivities, we computed derivatives of the Effect with respect to tlag, tinf and dose, where we used the chain rule to obtain expressions in terms of model sensitivities ($^{\mathrm{dAc}}/_{\mathrm{d}\tau_1}$, $^{\mathrm{dAc}}/_{\mathrm{d}\tau_2}$ and $^{\mathrm{dAc}}/_{\mathrm{d}r_1}$), and the Jacobian of the dosing parameterization, eq. (16).

The simulation outcome is shown in Fig. 1. Upon dosing, the effect state is inhibited and recovers after the dosing input switches back to zero, shown in Fig. 1A. According to the effect sensitivities, see Fig. 1B, higher doses lead to stronger inhibition. A longer infusion time will decrease the inhibition during the infusion and increase it afterwards. The same holds for an increased lag time. The impact of the lag time is five times higher than the impact of the infusion time. The reason is that the infusion time is connected to the infusion rate r_1, i.e., a longer infusion time means a lower infusion rate to keep the dose constant.

Typically, the input is unobserved and the lag and infusion time need to be estimated from the observed effect. The derivatives $\frac{\mathrm{dEffect}}{\mathrm{dtlag}}$ and $\frac{\mathrm{dEffect}}{\mathrm{dtinf}}$ at the observation time points

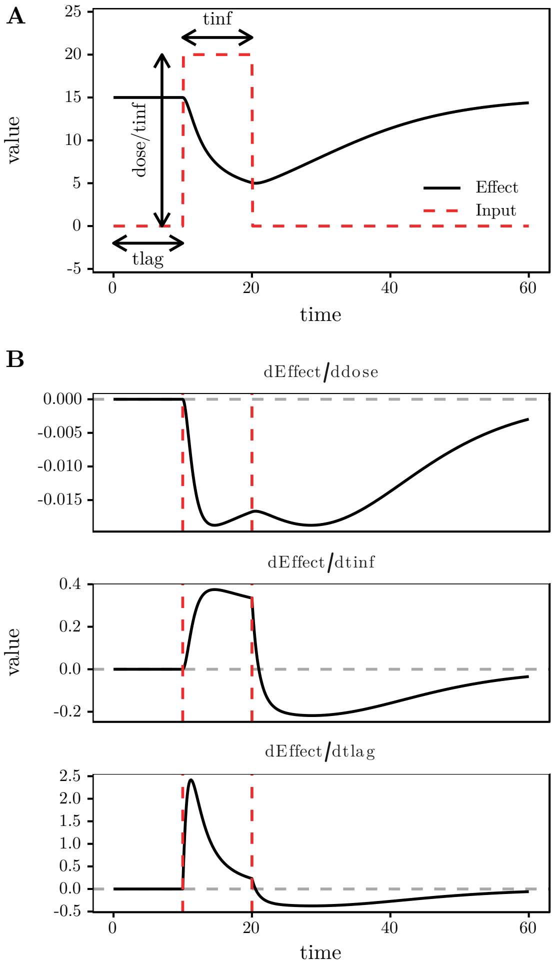

Fig. 1. Simulation of a PK/PD model. (A) The infusion input (red dashed line), parameterized by tlag, tinf and dose, provokes a response in the effect compartment (black continuous line). (B) The effect sensitivities with respect to the input parameters are shown. Start and end of the infusion are indicated by vertical dashed lines.

can be used to construct the gradient and Hessian of the least squares function. A typical application of the derivative $\frac{\mathrm{dEffect}}{\mathrm{ddose}}$ would be a sensitivity analysis.

V. Conclusion

The paper used linearization of ODE models to derive left- and right-sided limits of model sensitivities in the presence of different types of events: replacement, addition or multiplication of current state values with an event value at the event time. The expressions found for the jumps of the model sensitivities were interpreted as additional events on the sensitivity states themselves. Thereby, for each event type, a table of additional events to be applied to the sensitivity equations was derived. This makes the entire approach very transparent and easy to implement.

References

[1] P.I. Barton, R.J. Allgor, W.F. Feehery, and S. Galn. Dynamic optimization in a discontinuous world. *Industrial & Engineering Chemistry Research*, vol. 37, no. 3, pp. 966–981, Mar. 1998.

[2] F. Fröhlich, F. Theis, J.O. Rädler, and J. Hasenauer. Parameter estimation for dynamical systems with discrete events and logical operations. *Bioinformatics*, vol. 33, no. 7, pp. 1049–1056, Dec. 2016.