# Supplementary information to Optimization and uncertainty analysis of ODE models using 2nd 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.

January 2018

## Contents

[*] To whom correspondence should be addressed.

# 1 Derivation of the second order adjoint system

In the main manuscript, we gave the equations for forward and adjoint sensitivity analysis for the case of known measurement noise. However, the measurement noise is not always known and it is a common method to model it as parameter dependent quantity. This yields $\sigma_{ij} = \sigma_{ij}(\theta)$ for Equation (3) of the main manuscript. Here and in the rest of this section, we have $i = 1, \ldots, n_y$ the index of observables and $j = 1, \ldots, n_t$ the index of measurement time points. In the following, we show how the equations for first and second order forward and adjoint sensitivity analysis are derived. This derivation follows the ideas of (Özyurt & Barton (2005)), with some modification for the full Hessian and time-discrete measurement data.

## 1.1 First and second order forward sensitivity analysis

We start with the negative log-likelihood function

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

which describes the probability of observing a certain data set

$$\mathcal{D} = \{\bar{y}_{ij}\}_{\substack{i=1,\ldots,n_y \\ j=1,\ldots,n_t}} \tag{2}$$

given a parameter vector $\theta \in \Omega$. We use $\mathcal{J}(\theta)$ as objective function in the following. It is useful to define the contribution of each time point to the objective function as

$$\mathcal{J}_j(\theta) = \frac{1}{2} \sum_{i=1}^{n_y} \left( \log\left(2\pi\sigma_{ij}(\theta)^2\right) + \left( \frac{\bar{y}_{ij} - h_i(x(t_j, \theta), \theta)}{\sigma_{ij}(\theta)} \right)^2 \right) \tag{3}$$

Differentiation of Equation (1) yields the entries of the gradient:

$$\begin{aligned} \frac{\partial \mathcal{J}(\theta)}{\partial \theta_k} = & \sum_{j=1}^{n_t} \sum_{i=1}^{n_y} \left( \left( \frac{1}{\sigma_{ij}(\theta)} - \frac{(\bar{y}_{ij} - h_i(x(t_j, \theta), \theta))^2}{\sigma_{ij}(\theta)^3} \right) \frac{\partial \sigma_{ij}(\theta)}{\partial \theta_k} - \frac{\bar{y}_{ij} - h_i(x(t_j, \theta), \theta)}{\sigma_{ij}(\theta)^2} \right. \\ & \left. \cdot \left( \nabla_x h_i(x(t_j, \theta), \theta) s_k^x(t_j, \theta) + \frac{\partial h_i(x(t_j, \theta), \theta)}{\partial \theta_k} \right) \right) \end{aligned} \tag{4}$$

in which we used

$$s_k^x(t_j, \theta) = \frac{\partial x(t_j, \theta)}{\partial \theta_k} \tag{5}$$

to denote the first order state sensitivities. Thus, computing the gradient by forward sensitivity analysis requires computing these expression by integrating the corresponding ODE, which reads

$$\dot{s}_k^x(t_j, \theta) = \nabla_x f(x(t_j, \theta), t_j, \theta) s_k^x(t_j, \theta) + \frac{\partial f(x(t_j, \theta), t_j, \theta)}{\partial \theta_k} \tag{6}$$

where $f(x(t_j, \theta), t_j, \theta)$ denotes the right hand side of the original ODE, given in Equation (1) of the main manuscript, which was given by

$$\dot{x}(t, \theta) = f(x(t, \theta), t, \theta), \quad x(t_0, \theta) = x_0(\theta). \tag{7}$$

Another derivation with respect to $\theta_\ell$ gives the Hessian. For reasons of brevity, we omit all dependencies except those on $t_j$ in the notation:

$$\begin{aligned}
\frac{\partial^2 \mathcal{J}(\theta)}{\partial\theta_k \partial\theta_\ell} =& \sum_{j=1}^{n_t}\sum_{i=1}^{n_y} \left( -\frac{1}{\sigma_{ij}} \frac{\partial \sigma_{ij}}{\partial \theta_\ell} + 3\frac{(\bar{y}_{ij} - h_i(t_j))^2}{\sigma_{ij}^4} \frac{\partial \sigma_{ij}}{\partial \theta_\ell} + 2\frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^3} \left( \left(s_\ell^x(t_j)\right)^T \nabla_x h_i(t_j) + \frac{\partial h_i(t_j)}{\partial \theta_\ell} \right) \right) \frac{\partial \sigma_{ij}}{\partial \theta_k} \\
&+ \sum_{j=1}^{n_t}\sum_{i=1}^{n_y} \left( \left( \frac{1}{\sigma_{ij}} - \frac{(\bar{y}_{ij} - h_i(t_j))^2}{\sigma_{ij}^3} \right) \frac{\partial^2 \sigma_{ij}}{\partial\theta_k \partial\theta_\ell} - \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^2} \left( \left(s_\ell^x(t_j)\right)^T \frac{\partial \nabla_x h_i(t_j)}{\partial \theta_k} + \frac{\partial^2 h_i(t_j)}{\partial\theta_k \partial\theta_\ell} \right) \right) \\
&+ \sum_{j=1}^{n_t}\sum_{i=1}^{n_y} \left( \frac{1}{\sigma_{ij}^2} \left( \left(s_\ell^x(t_j)\right)^T \nabla_x h_i(t_j) + \frac{\partial h_i(t_j)}{\partial \theta_\ell} \right) + 2\frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^3} \frac{\partial \sigma_{ij}}{\partial \theta_\ell} \right) \frac{\partial h_i(t_j)}{\partial \theta_k} \\
&+ \sum_{j=1}^{n_t}\sum_{i=1}^{n_y} \left( \frac{1}{\sigma_{ij}^2} \left( \left(s_\ell^x(t_j)\right)^T \nabla_x h_i(t_j) + \frac{\partial h_i(t_j)}{\partial \theta_\ell} \right) + 2\frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^3} \frac{\partial \sigma_{ij}}{\partial \theta_\ell} \right) \left( \nabla_x^T h_i(t_j) s_k^x(t_j) \right) \\
&- \sum_{j=1}^{n_t}\sum_{i=1}^{n_y} \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^2} \left( \left( \left(s_\ell^x(t_j)\right)^T \nabla_x^T \nabla_x h_i(t_j) + \frac{\partial \nabla_x^T h_i(t_j)}{\partial \theta_\ell} \right) s_k^x(t_j) + \nabla_x^T h_i(t_j) s_{k,\ell}^x(t_j) \right) \qquad (8)
\end{aligned}$$

In this expression, the second order state sensitivities $s_{k,\ell}^x(t_j)$ show up. If the Hessian is to be calculated via second order forward sensitivity analysis, they have to be computed. Again, this is done by integrating the corresponding ODE:

$$\begin{aligned}
\dot{s}_{k,\ell}^x(t_j,\theta) =& \nabla_x^T f(x(t_j,\theta),t_j,\theta) s_{k,\ell}^x(t_j,\theta) + \left(\nabla_x^T \otimes \nabla_x^T\right) f(x(t_j,\theta),t_j,\theta) \left(s_k^x(t_j,\theta) \otimes s_\ell^x(t_j,\theta)\right) \\
&+ \frac{\partial \nabla_x^T f(x(t_j,\theta),t_j,\theta)}{\partial \theta_k} s_\ell^x(t_j,\theta) + \frac{\partial \nabla_x^T f(x(t_j,\theta),t_j,\theta)}{\partial \theta_\ell} s_k^x(t_j,\theta) + \frac{\partial^2 f(x(t_j,\theta),t_j,\theta)}{\partial\theta_k \partial\theta_\ell} \qquad (9)
\end{aligned}$$

Since Equation (9) is a system of $n_x n_\theta^2$ ODEs, it is common to approximate the Hessian under the assumption that the residuals $\bar{y}_{ij} - h_i(x(t_j,\theta),\theta)$ are small. This yields the Fisher information matrix (we omit again all dependencies but those on $t_j$):

$$\mathrm{FIM}_{k\ell}(\theta) = \sum_{j=1}^{n_t}\sum_{i=1}^{n_y} \frac{1}{\sigma_{ij}^2} \left( \left( \nabla_x^T h_i(t_j) s_k^x(t_j) + \frac{\partial h_i(t_j)}{\partial \theta_k} \right) \left( \nabla_x^T h_i(t_j) s_\ell^x(t_j) + \frac{\partial h_i(t_j)}{\partial \theta_\ell} \right) - \frac{\partial \sigma_{ij}}{\partial \theta_k} \frac{\partial \sigma_{ij}}{\partial \theta_\ell} \right) \qquad (10)$$

## 1.2 First order adjoint sensitivity analysis

In this part, we again omit all dependencies but those on $t_j$. We construct a term equal to zero. It comes from differentiating the original ODE with respect to $\theta_k$. We multiply it with a not further specified time-dependent vector $p(t) \in \mathbb{R}^{n_x}$, which will later be the adjoint state:

$$0 = \int_{t_j}^{t_{j+1}} p(t)^T \left( \dot{s}_k^x(t_j) - \frac{\partial f(t)}{\partial \theta_k} - \nabla_x^T f(t) s_k^x(t_j) \right) dt \qquad (11)$$

In order to derive the equations for the first order adjoint system, we take the contribution of the $j$-th time point to equation (4) and add Equation (11) to it:

$$\begin{aligned}\frac{\partial \mathcal{J}_j(\theta)}{\partial \theta_k} =& \sum_{i=1}^{n_y}\left(\left(\frac{1}{\sigma_{ij}} - \frac{\left(\bar{y}_{ij} - h_i(t_j)\right)^2}{\sigma_{ij}^3}\right)\frac{\partial \sigma_{ij}}{\partial \theta_k} - \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^2}\left(\nabla_x^T h_i(t_j) s_k^x(t_j) + \frac{\partial h_i(t_j)}{\partial \theta_k}\right)\right) \\ &+ \int_{t_j}^{t_{j+1}} p(t)^T \left(\dot{s}_k^x(t_j) - \frac{\partial f(t)}{\partial \theta_k} - \nabla_x^T f(t) s_k^x(t_j)\right) dt \qquad (12) \\ =& \sum_{i=1}^{n_y}\left(\left(\frac{1}{\sigma_{ij}} - \frac{\left(\bar{y}_{ij} - h_i(t_j)\right)^2}{\sigma_{ij}^3}\right)\frac{\partial \sigma_{ij}}{\partial \theta_k} - \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^2}\frac{\partial h_i(t_j)}{\partial \theta_k}\right) - \int_{t_j}^{t_{j+1}} p(t)^T \frac{\partial f(t)}{\partial \theta_k} dt \\ &- \left(\int_{t_j}^{t_{j+1}} \left(\dot{p}(t)^T + p(t)^T \nabla_x^T f(t)\right) dt + \sum_{i=1}^{n_y} \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^2} \nabla_x^T h_i(t_j)\right) s_k^x(t_j) \\ &+ \lim_{t \to t_{j+1}^-} p(t)^T s_k^x(t) - \lim_{t \to t_j^+} p(t)^T s_k^x(t) \qquad (13)\end{aligned}$$

In the last step, we used integration by parts in order to get the time derivative of the adjoint state and regrouped all expressions with state sensitivities together. To circumvent the computation of these state sensitivities, we can impose the following conditions on the adjoint state:

$$\dot{p}(t) = -\left(\nabla_x f(t)^T\right) p(t), \qquad \text{for t } \in (t_j, t_{j+1}) \qquad (14)$$

$$p(t_j) = \lim_{t \to t_{j+1}^-} p(t) = \lim_{t \to t_j^+} p(t) + \sum_{i=1}^{n_y} \nabla_x h_i(t_j) \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^2} \qquad (15)$$

$$p(t) = 0, \qquad \text{for t } > t_{n_t} \qquad (16)$$

Imposing those conditions on the adjoint state and summing over all time points $j = 1, \ldots, n_t$ yields the following expression for the gradient:

$$\frac{\partial \mathcal{J}(\theta)}{\partial \theta_k} = \sum_{i=1}^{n_y}\sum_{j=1}^{n_t}\left(\left(\frac{1}{\sigma_{ij}} - \frac{\left(\bar{y}_{ij} - h_i(t_j)\right)^2}{\sigma_{ij}^3}\right)\frac{\partial \sigma_{ij}}{\partial \theta_k} - \frac{\bar{y}_{ij} - h_i(t_j)}{\sigma_{ij}^2}\frac{\partial h_i(t_j)}{\partial \theta_k}\right) - \int_{t_0}^{t_{n_t}} p(t)^T \frac{\partial f(t)}{\partial \theta_k} dt - p(t_0)^T s_k^x(t_0) \qquad (17)$$

In Equation (17), all expression depending on the state sensitivities have vanished except those for the initial condition. However, those dependencies can usually be computed analytically.

### 1.3 Second order adjoint sensitivity analysis

To compute the Hessian, we differentiate Equation (13) and get

$$
\begin{aligned}
\frac{\partial^2 \mathcal{J}_j(\theta)}{\partial\theta_k\partial\theta_\ell} =& \sum_{i=1}^{n_y}\left(-\frac{1}{\sigma_{ij}}\frac{\partial\sigma_{ij}}{\partial\theta_\ell} + 3\frac{(\bar{y}_{ij}-h_i(t_j))^2}{\sigma_{ij}^4}\frac{\partial\sigma_{ij}}{\partial\theta_\ell} + 2\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^3}\left(\left(s_\ell^x(t_j)\right)^T\nabla_x h_i(t_j) + \frac{\partial h_i(t_j)}{\partial\theta_\ell}\right)\right)\frac{\partial\sigma_{ij}}{\partial\theta_k} \\
&+ \sum_{i=1}^{n_y}\left(\left(\frac{1}{\sigma_{ij}} - \frac{(\bar{y}_{ij}-h_i(t_j))^2}{\sigma_{ij}^3}\right)\frac{\partial^2\sigma_{ij}}{\partial\theta_k\partial\theta_\ell} - \frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^2}\left(\left(s_\ell^x(t_j)\right)^T\frac{\partial\nabla_x h_i(t_j)}{\partial\theta_k} + \frac{\partial^2 h_i(t_j)}{\partial\theta_k\partial\theta_\ell}\right)\right) \\
&+ \sum_{i=1}^{n_y}\left(\frac{1}{\sigma_{ij}^2}\left(\left(s_\ell^x(t_j)\right)^T\nabla_x h_i(t_j) + \frac{\partial h_i(t_j)}{\partial\theta_\ell}\right) + 2\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^3}\frac{\partial\sigma_{ij}}{\partial\theta_\ell}\right)\frac{\partial h_i(t_j)}{\partial\theta_k} \\
&- \int\limits_{t_j}^{t_{j+1}}\left(\frac{\partial p(t)^T}{\partial\theta_\ell}\frac{\partial f(t)}{\partial\theta_k} + p(t)^T\frac{\partial\nabla_x^T f(t)}{\partial\theta_k}s_\ell^x(t) + p(t)^T\frac{\partial^2 f(t)}{\partial\theta_k\partial\theta_\ell}\right)dt \\
&+ \left(\sum_{i=1}^{n_y}\left(\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) + 2\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^3}\frac{\partial\sigma_{ij}}{\partial\theta_\ell}\right)\nabla_x^T h_i(t_j)\right. \\
&- \sum_{i=1}^{n_y}\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^2}\left(\left(s_\ell^x(t_j)\right)^T\nabla_x^T\nabla_x h_i(t_j) + \frac{\partial\nabla_x^T h_i(t_j)}{\partial\theta_\ell}\right) \\
&\left.- \int\limits_{t_j}^{t_{j+1}}\left(\frac{\partial\dot{p}(t)^T}{\partial\theta_\ell} + \frac{\partial p(t)^T}{\partial\theta_\ell}\nabla_x^T f(t) + p(t)^T\left((\nabla_x^T\otimes\nabla_x^T)f(t)(\mathbb{I}_n\otimes s_\ell^x(t)) + \frac{\partial\nabla_x^T f(t)}{\partial\theta_\ell}\right)\right)\right)s_k^x(t_j) \\
&- \left(\int\limits_{t_j}^{t_{j+1}}\left(\dot{p}(t)^T + p(t)^T\nabla_x^T f(t)\right)dt + \sum_{i=1}^{n_y}\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^2}\nabla_x^T h_i(t_j)s_k^x(t_j)\right)s_{k,\ell}^x(t_j) \\
&+ \lim_{t\to t_{j+1}^-}p(t)^T s_{k,\ell}^x(t) - \lim_{t\to t_j^+}p(t)^T s_{k,\ell}^x(t) + \lim_{t\to t_{j+1}^-}\frac{\partial p(t)^T}{\partial\theta_\ell}s_k^x(t) - \lim_{t\to t_j^+}\frac{\partial p(t)^T}{\partial\theta_\ell}s_k^x(t) \qquad (18)
\end{aligned}
$$

We can use Equation (18) to define the equations for the second order adjoint state analogously to the first order adjoint state:

$$
\frac{\partial\dot{p}(t)}{\partial\theta_\ell} = -\frac{\partial p(t)}{\partial\theta_\ell}\nabla_x(f(t)^T) - \left((s_\ell^x(t)\otimes\mathbb{I}_n)(\nabla_x\otimes\nabla_x)(f(t)^T) + \frac{\partial\nabla_x(f(t)^T)}{\partial\theta_\ell}\right)p(t) \tag{19}
$$

$$
\begin{aligned}
\frac{\partial p(t_j)}{\partial\theta_\ell} =& \lim_{t\to t_j^+}\frac{\partial p(t)}{\partial\theta_\ell} + \sum_{i=1}^{n_y}\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^2}\left(\nabla_x^T\nabla_x h_i(t_j)s_\ell^x(t_j) + \frac{\partial\nabla_x h_i(t_j)}{\partial\theta_\ell}\right) \\
&- \sum_{i=1}^{n_y}\left(\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^3}\frac{\partial\sigma_{ij}}{\partial\theta_\ell}\nabla_x h_i(t_j) + \frac{1}{\sigma_{ij}^2}\left(\nabla_x^T h_i(t_j)s_\ell^x(t_j) + \frac{\partial h_i(t_j)}{\partial\theta_\ell}\right)\nabla_x h_i(t_j)\right) \qquad (20)
\end{aligned}
$$

$$
\frac{\partial p(t_{n_t})^T}{\partial\theta_\ell} = 0 \tag{21}
$$

using these equations, those from the adjoint state and summing up over all time points finally yields the Hessian:

$$
\begin{aligned}
\frac{\partial^2 \mathcal{J}_j(\theta)}{\partial\theta_k \partial\theta_\ell} =& \sum_{j=1}^{n_t}\sum_{i=1}^{n_y}\left(-\frac{1}{\sigma_{ij}}\frac{\partial\sigma_{ij}}{\partial\theta_\ell} + 3\frac{(\bar{y}_{ij}-h_i(t_j))^2}{\sigma_{ij}^4}\frac{\partial\sigma_{ij}}{\partial\theta_\ell} + 2\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^3}\left(\nabla_x^T h_i(t_j) s_\ell^x(t_j) + \frac{\partial h_i(t_j)}{\partial\theta_\ell}\right)\right)\frac{\partial\sigma_{ij}}{\partial\theta_k} \\
&+ \sum_{j=1}^{n_t}\sum_{i=1}^{n_y}\left(\left(\frac{1}{\sigma_{ij}} - \frac{(\bar{y}_{ij}-h_i(t_j))^2}{\sigma_{ij}^3}\right)\frac{\partial^2\sigma_{ij}}{\partial\theta_k\partial\theta_\ell} - \frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^2}\left(\frac{\partial\nabla_x^T h_i(t_j)}{\partial\theta_k} s_\ell^x(t_j) + \frac{\partial^2 h_i(t_j)}{\partial\theta_k\partial\theta_\ell}\right)\right) \\
&+ \sum_{j=1}^{n_t}\sum_{i=1}^{n_y}\left(\frac{1}{\sigma_{ij}^2}\left(\nabla_x^T h_i(t_j) s_\ell^x(t_j) + \frac{\partial h_i(t_j)}{\partial\theta_\ell}\right) + 2\frac{\bar{y}_{ij}-h_i(t_j)}{\sigma_{ij}^3}\frac{\partial\sigma_{ij}}{\partial\theta_\ell}\right)\frac{\partial h_i(t_j)}{\partial\theta_k} \\
&- \int_{t_0}^{t_{n_t}}\left(\frac{\partial p(t)^T}{\partial\theta_\ell}\frac{\partial f(t)}{\partial\theta_k} + p(t)^T\frac{\partial\nabla_x^T f(t)}{\partial\theta_k} s_\ell^x(t) + p(t)^T\frac{\partial^2 f(t)}{\partial\theta_k\partial\theta_\ell}\right)dt - p(t_0)^T s_{k,\ell}^x(t_0) - \frac{\partial p(t_0)^T}{\partial\theta_\ell} s_k^x(t_0)
\end{aligned}
\tag{22}
$$

## 2 Parameter optimization

The comparison of optimization methods we carried out included a four-fold repetition with different initial points of the multi-start optimization experiment, which we carried out for six different local optimization methods with 200 starts for each local optimization method. Mean and standard deviation depicted in Figure 4 of the main manuscript are computed over these four repetitions. For the numerical experiments, we provided the following tolerances to the local optimization algorithms:

- tolerance for step size in parameter: $10^{-10}$ for the JAK/STAT model, $10^{-12}$ for the RAF/MEK/ERK model
- tolerance for change in the objective function: $10^{-10}$ for both models
- tolerance for the remaining gradient: $10^{-6}$ for both models

Moreover, the following settings have been used for optimization:

- maximum number of optimization steps (per step at least one gradient evaluation): 2000
- maximum number of objective function evaluations: $1000 \cdot n_\theta$
- `PrecondBandWidth` = `inf`, i.e. the inner optimization problem was solved by factorization

For all other optimization options, the MATLAB default settings were used. The definition for a threshold of the objective function value to which an optimization was accepted as converged to the global optimum was based on the corresponding waterfall plot for each experiment (Figure 1).

## 3 Profile likelihood calculation

The quality of a profile is not easy to assess: A quadratic objective function leads to a Gaussian-shaped profile likelihood, therefore one would expect a similar shape for the profile of a well identifiable parameter. Kinks or wrinkled shapes usually indicate problems during profile calculation. However, as similar profile shapes may also occur naturally, a close inspection of the results from profile computation is necessary.

For the model M3 (JAK/STAT), all considered methods could compute the profiles well and the results are in good agreement (Figure 2). However, model M2 (RAF/MEK/ERK) showed a more complex behavior, as many profiles computed with the optimization-based method and `lsqnonlin` were truncated (Figure 3), which led to incorrect confidence intervals (Figure 4). As already mentioned, results were even worse for the profiles computed with `fmincon` and the interior-point method using either the Hessian or the BFGS approximation.

When circumventing optimization, it was necessary in both examples to employ the hybrid method, as the purely integration-based method started to use very small step sizes at discontinuities in the optimal path and got stuck (Figure 5). When the error tolerances of the ODE solvers for the profile ODE were sufficiently relaxed and/or the Fisher information matrix was used as approximation, this problem of too small step sizes could be avoided. However, the profiles computed in this manner were clearly suboptimal (see Figure 6). It is important to note that also the hybrid method using the FIM could not compute the profiles for the RAF/MEK/ERK model correctly, as the optimization, which had to be carried out at profile path discontinuities, did not converge for this model and hence the profiles were truncated.

# 4 Trust-region-reflective, interior-point, and Levenberg-Marquardt algorithm

We could see that the chosen optimization algorithm within `fmincon` (either trust-region-reflective or interior-point) had a substantial impact on the results. For multi-start local optimization, the interior-point algorithm yielded the better results (i.e., better convergence, see Figure 1 here and Figure 3 in the main manuscript), while the trust-region-reflective algorithm converges faster in the case of profile computation (Figure 7 here and Figure 4 in the main manuscript). We suppose these differences are due to the barrier function which is employed in the interior-point algorithm: It seems to have a positive effect on optimization when started far away from a local optimum, like this is usually the case in multi-start local optimization tasks. When started close to a local optimum, as it is done in profile calculation, this barrier function can lead to inaccuracies which obstruct the optimization process. Hence, each optimization task within a profile needs more steps to converge when using the interior-point algorithm (see also The MathWorks Inc. (2018)).

Generally, curvature information is particularly helpful for optimization when being close to a local optimum. This is reflected by the fact that the trust-region-reflective algorithm in `fmincon` takes fewer iterations to converge than trust-region-reflective algorithm in `lsqnonlin` when doing an optimization-based profile calculation. The algorithm in `fmincon` can rely on the exact Hessian matrix, whereas the algorithm in `lsqnonlin` only uses a first-order sensitivity based approximation of the Hessian.

In principle, also the Levenberg-Marquardt algorithm in `lsqnonlin` could be used for parameter estimation. However, the implementation does not allow to have box constraints on the parameters. As this kind of constraints is present in our application examples and many other ODE-models in systems biology, we had to disregard this algorithm for our study.

# 5 Implementation

We used the MATLAB toolbox AMICI (Advanced Matlab Interface to CVODES and IDAS) for ODE simulation, which is freely available at https://github.com/ICB-DCM/AMICI. The code version[1] used for simulation

[1] The toolboxes AMICI and PESTO are being steadily developed. In order to refer to the precise code versions used for this study, we rely on the platform Zenodo. It makes codes and code versions citable by providing DOIs (Digital Object Identifier) for them and by storing the respective program code.

can be found under *version Second_order_adjoint_submission*, see Fröhlich *et al.* (2018). We moreover used the MATLAB toolbox PESTO (Parameter EStimation TOolbox) for the optimization and profile calculation experiments, which is also freely available at https://github.com/ICB-DCM/PESTO. The code version which was used can be found under *version Second_order_adjoint_submission*, see Stapor *et al.* (2018). All numerical experiments were carried out using the Release 2017a of MATLAB.

The models for the numerical experiments can be found at https://github.com/ICB-DCM/PESTO. In the two models, for which we carried out the optimization and profile calculation experiments, we used the following error tolerances for ODE integration: relative error $10^{-10}$, absolute error $10^{-14}$, relative error in quadratures $10^{-4}$, absolute error in quadratures $10^{-6}$. We restricted the maximum number of steps of the ODE solver to $10^5$.

As stated in the main manuscript, ODE simulation (with or without sensitivities) is the computationally most expensive part during optimization and profile calculation. In order to have quantitative results, we carried out the first 5 out of the 200 runs for optimization with `lsqnonlin` on the model M3 (JAK/STAT) and assessed the computation time spent in different parts of the code.

- Computation time (total): $257,6$ seconds
- Computation time (C++ part): $239,6$ seconds ($93,01\%$ of total computation time)
- Computation time (MATLAB part): $18,0$ seconds ($6,99\%$ of total computation time)

For Hessian-based optimization, an even higher percentage of the total computation time is spent in the C++ part, since the integration of the corresponding ODE systems takes longer. This shows that for the optimization problems considered in this study, the time spent in the MATLAB part of the code can basically be neglected when comparing computation times for optimization and profile calculation.

# 6 Biological background

## 6.1 List of biological abbreviations

In the biological test models which are mentioned in the main manuscript and the Supplementary Information, we have used several biological abbreviations. They are listed in this subsection.

- Akt (or PKB): historical name, no abbreviation (or Protein Kinase B)
- EGF: Epidermal Growth Factor
- Epo: Erythropoietin
- ERK (or MAPK): Extracellular-signal Regulated Kinases (or Microtubule Associated Protein Kinase)
- JAK: Janus Kinase
- MEK: MAPK/ERK Kinase
- NF$\kappa$B: Nuclear Factor Kappa-light-chain-enhancer of activated B cells
- PI3K: Phosphatidylinositol-4, 5-bisphosphate 3-Kinase
- RAF: Rapidly Accelerated Fibrosarcoma
- RAS: Rat Sarcoma

- STAT: Signal Transducer and Activator of Transcription
- TNF: Tumor Necrosis Factor

## 6.2 Biological models used in the main manuscript

In this subsection, we provide a short description of the biological test models which we used for the benchmarking of our methods.

### Epo receptor model

The ODE-model on the Epo (Erythropoietin) receptor by Becker *et al.* (2010) describes the uptake of the hormone Epo in hematopoietic stem cells by the Epo receptor. It was used to infer the precise mechanism of Epo uptake among a series of potential candidates, since this process was known to be very sensitive to changes in the Epo concentrations over a large range of the latter, which makes it very interesting from a biological point of view.

The model used in our study relies on the published model and consists of

- 6 state variables ($n_x = 6$),
- 9 parameters ($n_\theta = 9$),
- 3 observables ($n_y = 3$), and includes
- 24 data points (biological measurement data) distributed on
- 8 time points ($n_t = 8$).

### JAK/STAT signaling model

The JAK/STAT signaling pathway involves the two proteins JAK2 (Janus kinase 2) and STAT5 (Signal Transducer and Activator of Transcription 5), which transmit the signal from the Erythropoietin receptor to the nucleus. This signaling pathway plays an important role in the process of haematopoiesis and is crucial for the survival decision of erythrocytes. The ODE model which we used in our study relies on the model published by Swameye *et al.* (2003), with additionally spline fitted input of Epo (see Schelker *et al.* (2012).

This model was also used in other studies for method benchmarking (see e.g. Maier *et al.* (2017)) since it is known to be a non-trivial optimization problem. It consists of

- 9 state variables ($n_x = 9$),
- 16 parameters ($n_\theta = 16$),
- 3 observables ($n_y = 3$), and includes
- 46 data points (biological measurement data) distributed on
- 16 time points ($n_t = 16$).

### RAF/MEK/ERK signaling model

The RAF/MEK/ERK pathway (also known as RAS/RAF/MEK/ERK or MAPK/ERK pathway) is a signaling cascade which transmits a signaling from the EGF receptor to the nucleus. Mutations in the involved proteins may lead to the development of cancer in mammalian cells, which is the reason why is pathway plays a key role cancer research and treatment.

The model and the data used in this manuscript are taken from Fiedler *et al.* (2016). It consists of

- 3 state variables ($n_x = 3$),
- 28 parameters ($n_\theta = 28$),
- 8 observables ($n_y = 8$), and includes
- 72 data points (biological measurement data) distributed on
- 7 time points ($n_t = 7$) and
- 3 different experimental conditions.

### Escherichia Coli carbon metabolism model

Escherichia Coli (E.Coli) is a species of gut bacteria which is often used for applications in metabolic engineering. Hence, a quantitative understanding of the central carbon metabolism of E.Coli is important for microbial production processes. The model presented by Chassagnole *et al.* (2002) gives such a quantitative description of many of these processes.

We used this model and the data from the same publication in our study, since has already been used to asses the computational efficiency of gradient computation by Fröhlich *et al.* (2017). It consists of

- 18 state variables ($n_x = 18$),
- 116 parameters ($n_\theta = 116$),
- 9 observables ($n_y = 9$), and includes
- 110 data points (biological measurement data) distributed on
- 51 time points ($n_t = 7$).

### EGF & TNF signaling model

This model is taken from (MacNamara *et al.*, 2012), who developed it as a reduced version in the model presented in (Saez-Rodriguez *et al.*, 2009). It describes a large-scale signaling process in mammalian cells, including different pathways such as the RAF/MEK/ERK signaling pathway (see above), the PI3K/Akt signaling pathway, and the NF$\kappa$B signaling pathway. Most of these proteins play a major role in research and treatment of cancer, as the corresponding singaling processes are altered in cancer cells.

This model was also used to asses the computational efficiency of gradient computation by Fröhlich *et al.* (2017). It consists of

- 26 state variables ($n_x = 26$),
- 86 parameters ($n_\theta = 86$),

- 6 observables ($n_y = 6$), and includes
- 960 data points (simulated in silico data) distributed on
- 16 time points ($n_t = 16$) and
- 10 different experimental conditions.

# References

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

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

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

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

Fröhlich, F., Weindl, D., Stapor, P., & Hasenauer, J. (2018). Icb-dcm/amici: Amici (version Second_order_adjoint_submission). Zenodo. Http://doi.org/10.5281/zenodo.1162326.

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

Maier, C., Loos, C., & Hasenauer, J. (2017). Robust parameter estimation for dynamical systems from outlier-corrupted data. *Bioinformatics*, *33*(5), 718–725.

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

Saez-Rodriguez, J., Alexopoulos, L. G., Epperlein, J., Samaga, R., Lauffenburger, D. A., Klamt, S., & Sorger, P. K. (2009). Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Molecular Systems Biology*, *5*(1).

Schelker, M., Raue, A., Timmer, J., & Kreutz, C. (2012). Comprehensive estimation of input signals and dynamics in biochemical reaction networks. *Bioinformatics*, *28*(18), i529–i534.

Stapor, P., Weindl, D., Ballnus, B., Hug, S., Loos, C., Fiedler, A., Krause, S., Hross, S., Fröhlich, F., & Hasenauer, J. (2018). Icb-dcm/pesto: Pesto (version Second_order_adjoint_submission). Zenodo. Http://doi.org/10.5281/zenodo.579891.

Swameye, I., Müller, T. G., Timmer, J., Sandra, O., & 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.

The MathWorks Inc. (2018). Matlab documentation, choosing the algorithm. Website.

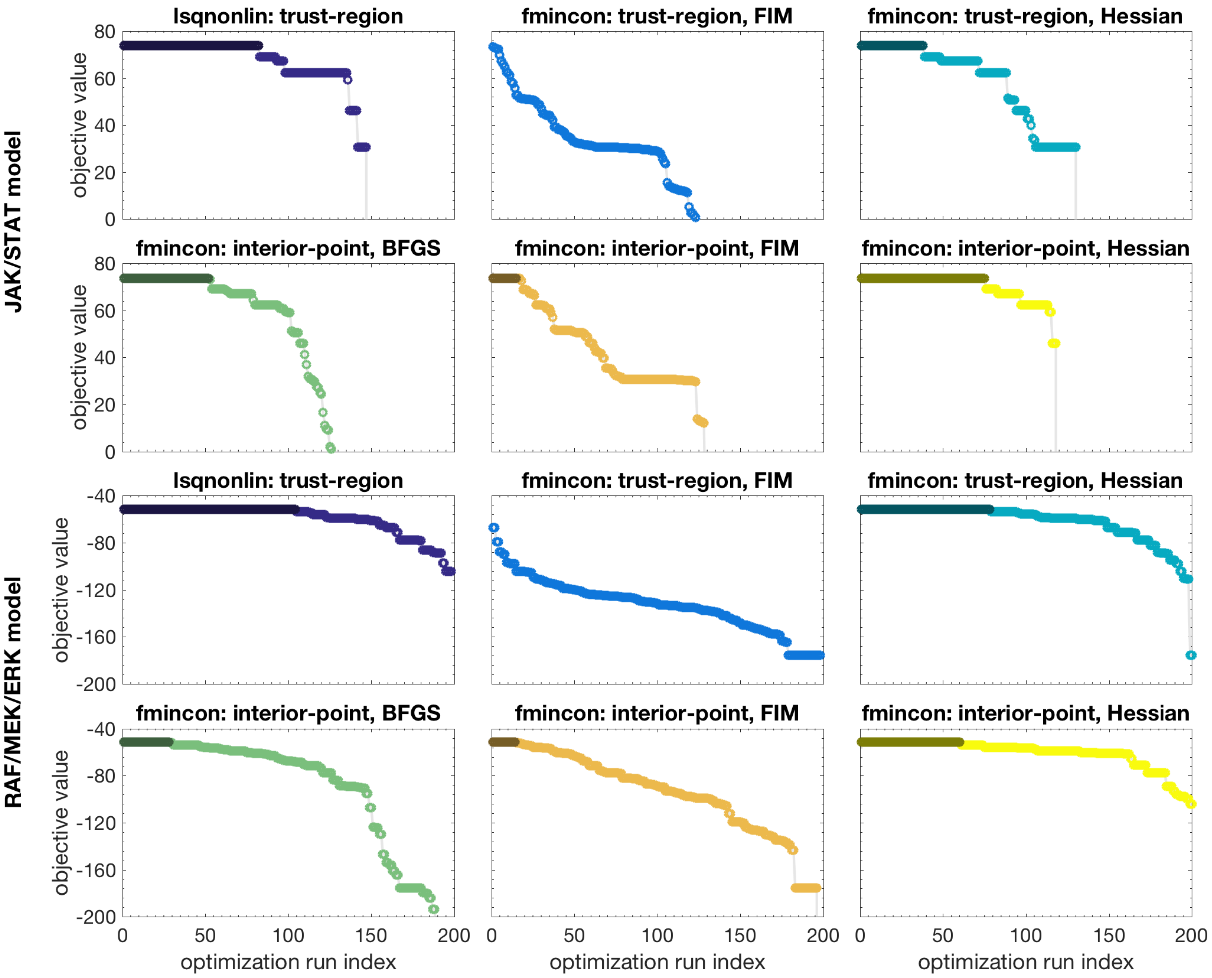

Figure 1: Waterfall plots of different local optimization methods. For the models of JAK/STAT and RAF/MEK/ERK signaling, four multi-start local optimizations (of which the first is shown here) with six different local optimization methods were carried out (using the least-squares optimization algorithm `lsqnonlin` with the trust-region-reflective algorithm, the constraint optimization algorithm `fmincon` with a trust-region-reflective algorithm provided with either the FIM or the Hessian computed with second order adjoint sensitivity analysis, and `fmincon` with an interior-point method with either a BFGS-approximation to the Hessian, the FIM, or the Hessian computed with second order adjoint sensitivity analysis). Each multi-start consisted of 200 local optimization runs, of which the final objective function values were sorted and depicted. Plateaus in the waterfall plot indicate local optima of the objective function.

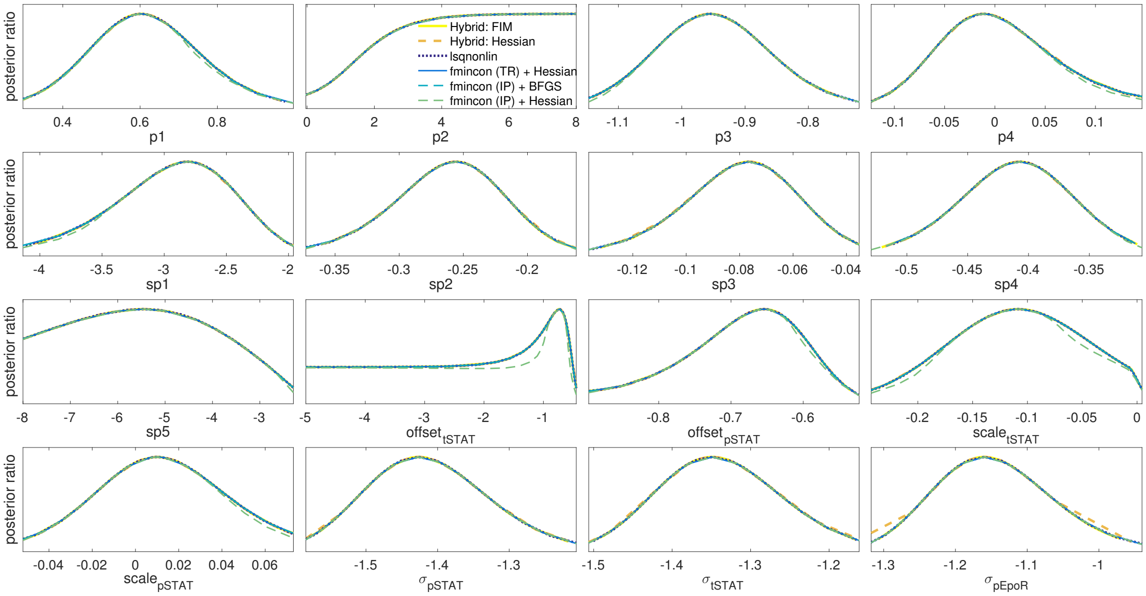

Figure 2: Profile likelihoods for the 16 parameters of the JAK/STAT example. The posterior ratio in the profile likelihood is depicted against the parameter value for each profile, profiles are cut off at a confidence level of 95%. All five discussed methods are depicted here (optimization based methods using `lsqnonlin`, `fmincon` with the trust-region-reflective algorithm and BFGS, `fmincon` with the trust-region-reflective algorithm and Hessian, hybrid approach with the Fisher information matrix and with the Hessian) using different colors. All profiles are in good agreement with each other.

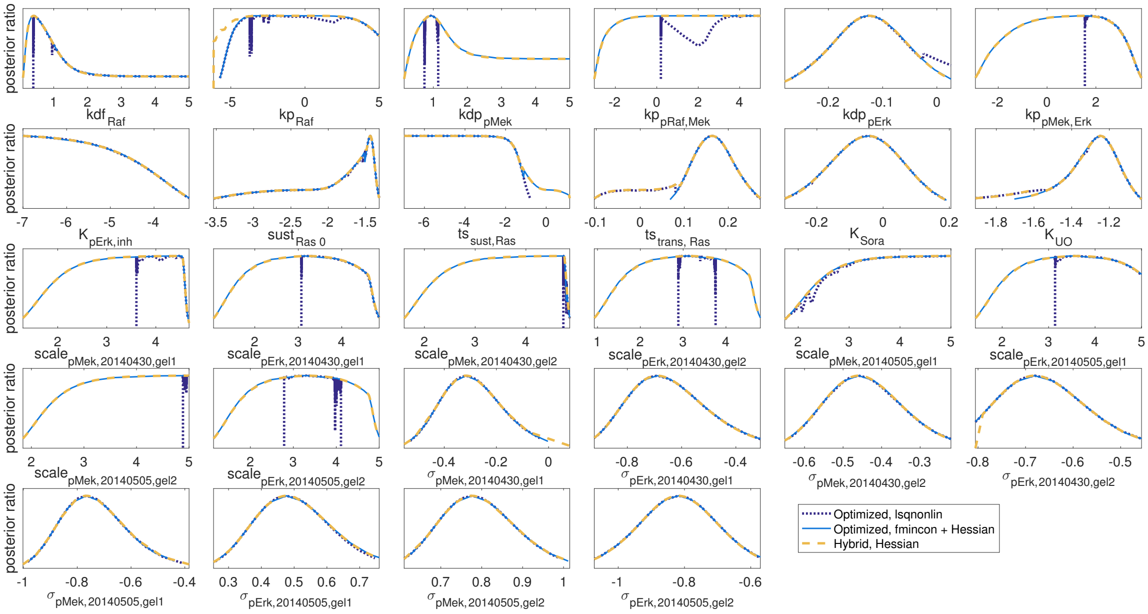

Figure 3: Profile likelihoods for the 28 parameters of the RAF/MEK/ERK example. The posterior ratio in the profile likelihood is depicted against the parameter value for each profile, profiles are cut off at a confidence level of 95%. The two methods using the Hessian (yellow colors, optimization based and hybrid) are in good agreement with each other. The optimization based profiles computed with `lsqnonlin` show an unfavorable behavior for 11 out of 28 profiles, as those profiles are not fully computed and cut off due to convergence problems during optimization.

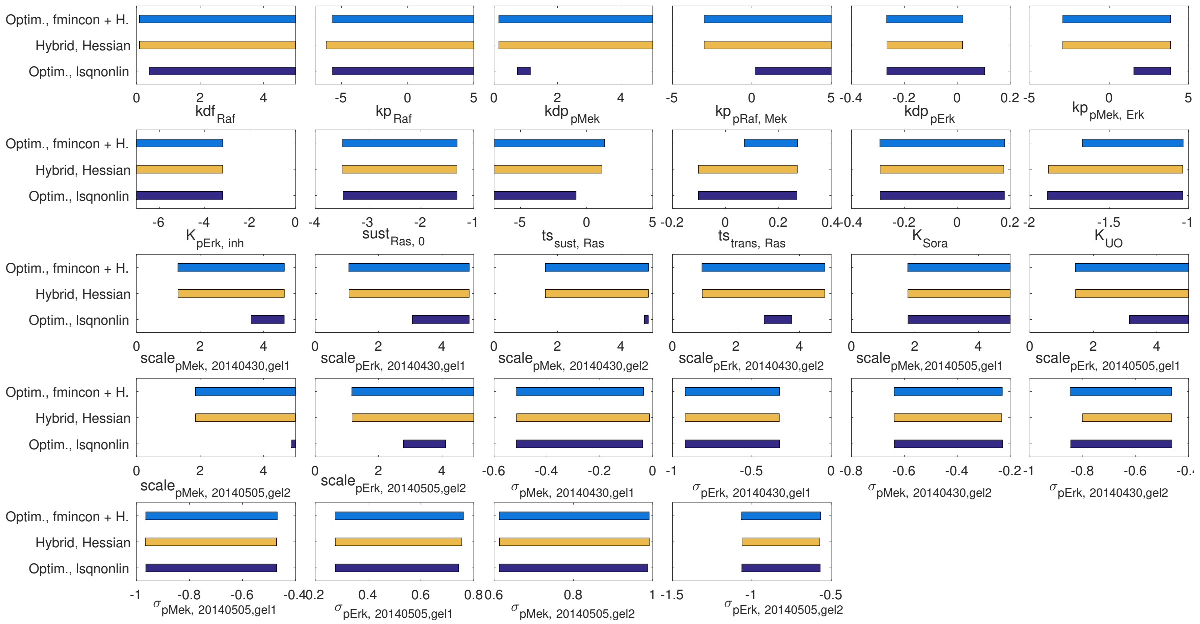

Figure 4: Confidence intervals computed for the RAF/MEK/ERK model, based on profiles computed with the optimization-based approach and `lsqnonlin` (dark blue), the optimization-based approach and `fmincon` using the trust-region-reflective algorithm and exact Hessians (light blue), and the hybrid approach using exact Hessians (orange).

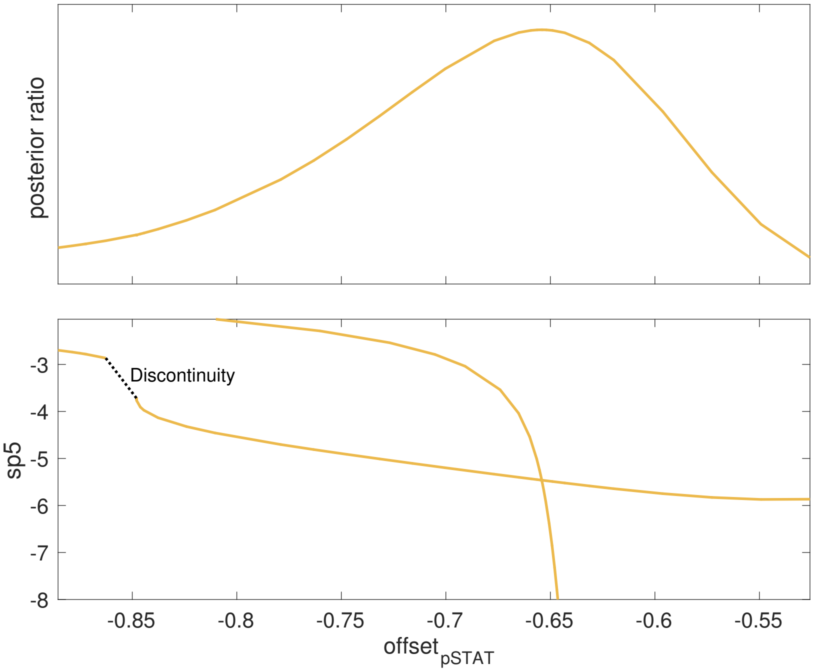

Figure 5: Discontinuity in the profile path. The profile of the parameter $\text{offset}_{\text{tSTAT}}$ in the JAK/STAT example is shown in the upper figure. The paths of this parameter and a second one (sp5) are projected to the subspace spanned by those parameters are depicted in the lower figure. The parameter values are depicted on the axes. The jump in the profile path of $\text{offset}_{\text{tSTAT}}$, which had to be computed using optimization steps, is shown as dashed line.

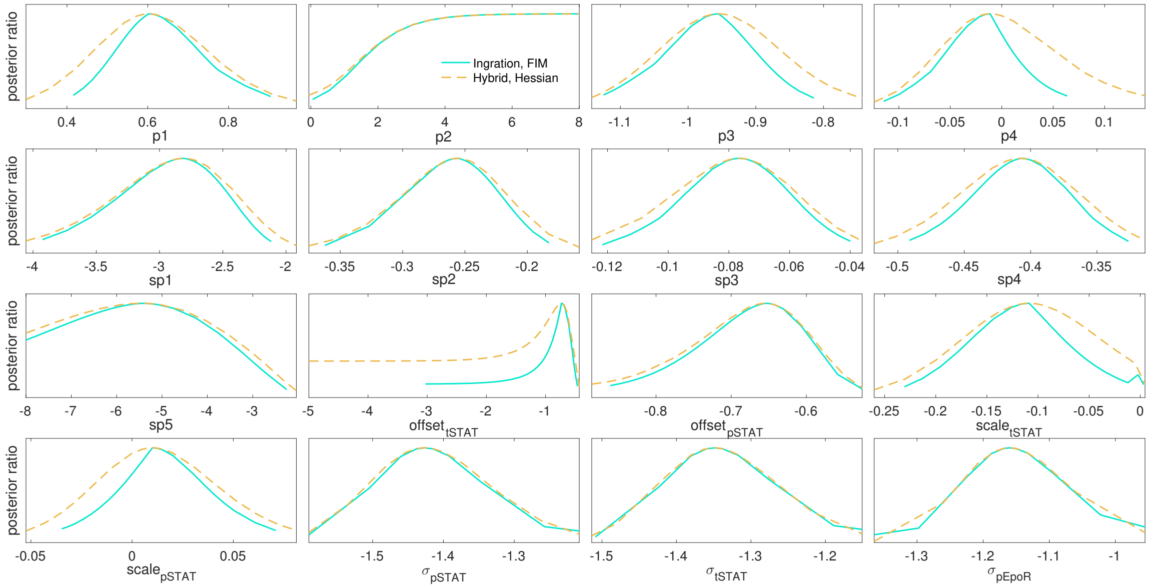

Figure 6: Profiles likelihoods using a purely integration based approach. The 16 parameter profiles of the JAK/STAT example were computed by integration using the Fisher information matrix as approximation. The trajectory was not optimized and reinitialized at the optimal path when a remaining gradient was observed. The correct profiles computed with the hybrid approach are depicted as dashed line for comparison.

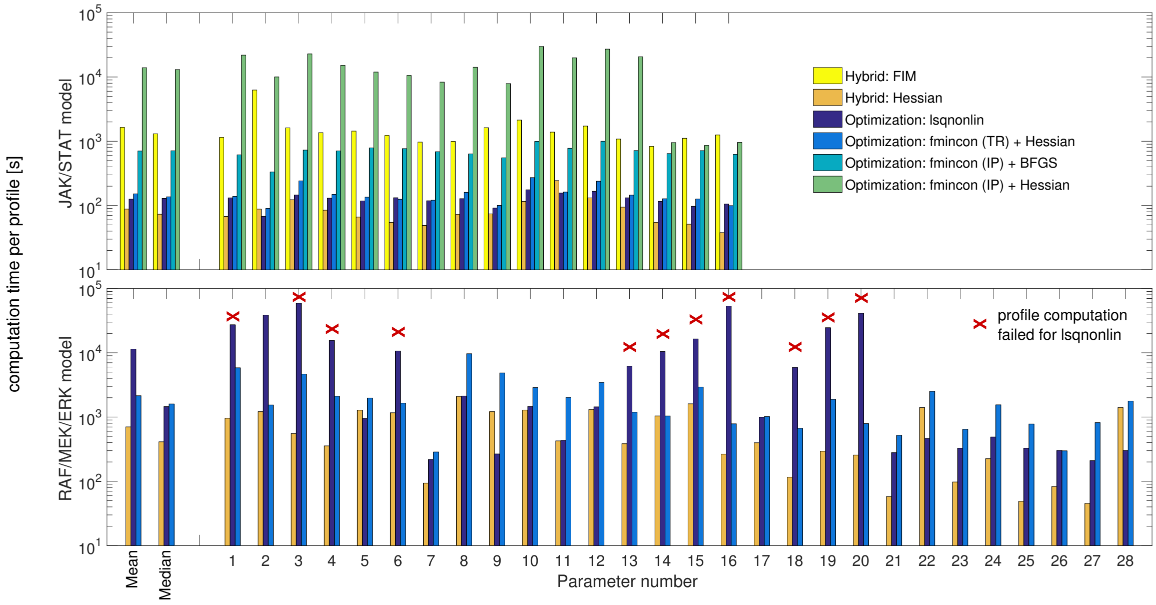

Figure 7: Computation time of profile likelihoods for the JAK/STAT and the RAF/MEK/ERK model using different methods.