



Customized Steady-State Constraints for Parameter Estimation in Non-Linear Ordinary Differential Equation Models

Marcus Rosenblatt^{1*}, Jens Timmer^{1,2,3} and Daniel Kaschek¹

¹ Institute of Physics, Albert Ludwig University of Freiburg, Freiburg, Germany, ² Freiburg Centre for Systems Biology, Albert Ludwig University of Freiburg, Freiburg, Germany, ³ BIOS Centre for Biological Signaling Studies, Albert Ludwig University of Freiburg, Freiburg, Germany

OPEN ACCESS

Edited by:

Julio Vera González,
University Hospital Erlangen, Germany

Reviewed by:

Irene Otero Muras,
Consejo Superior de Investigaciones
Científicas, Spain
Luis L. Fonseca,
Georgia Institute of Technology, USA

*Correspondence:

Marcus Rosenblatt
marcus.rosenblatt@fdm.uni-freiburg.de

Specialty section:

This article was submitted to
Systems Biology,
a section of the journal
Frontiers in Cell and Developmental
Biology

Received: 04 December 2015

Accepted: 21 April 2016

Published: 11 May 2016

Citation:

Rosenblatt M, Timmer J and
Kaschek D (2016) Customized
Steady-State Constraints for
Parameter Estimation in Non-Linear
Ordinary Differential Equation Models.
Front. Cell Dev. Biol. 4:41.
doi: 10.3389/fcell.2016.00041

Ordinary differential equation models have become a wide-spread approach to analyze dynamical systems and understand underlying mechanisms. Model parameters are often unknown and have to be estimated from experimental data, e.g., by maximum-likelihood estimation. In particular, models of biological systems contain a large number of parameters. To reduce the dimensionality of the parameter space, steady-state information is incorporated in the parameter estimation process. For non-linear models, analytical steady-state calculation typically leads to higher-order polynomial equations for which no closed-form solutions can be obtained. This can be circumvented by solving the steady-state equations for kinetic parameters, which results in a linear equation system with comparatively simple solutions. At the same time multiplicity of steady-state solutions is avoided, which otherwise is problematic for optimization. When solved for kinetic parameters, however, steady-state constraints tend to become negative for particular model specifications, thus, generating new types of optimization problems. Here, we present an algorithm based on graph theory that derives non-negative, analytical steady-state expressions by stepwise removal of cyclic dependencies between dynamical variables. The algorithm avoids multiple steady-state solutions by construction. We show that our method is applicable to most common classes of biochemical reaction networks containing inhibition terms, mass-action and Hill-type kinetic equations. Comparing the performance of parameter estimation for different analytical and numerical methods of incorporating steady-state information, we show that our approach is especially well-tailored to guarantee a high success rate of optimization.

Keywords: non-linear ODE models, parameter estimation, biochemical reaction networks, steady-state, positive solutions, multiplicity, multi-stability, success rate

1. INTRODUCTION

Dynamical systems are frequently modeled by systems of ordinary differential equations (ODEs). Homogeneously distributed molecules are treated as continuous quantities interacting with each other according to kinetic laws, e.g., mass-action or Michaelis-Menten kinetics.

A typical ODE system

$$\dot{x} = f(x, p, u(t)), \quad x(0) = x_0 \quad (1)$$

determines the time-evolution of an N -dimensional state vector $x(t)$. Here, $p \in \mathbb{R}_+^M$ denotes the M -dimensional vector of non-negative kinetic parameters. The vector $x_0 \in \mathbb{R}_{+,0}^N$, where $\mathbb{R}_{+,0} = \mathbb{R}_+ \cup \{0\}$, gives the set of initial conditions. The kinetic parameters and initial conditions together span the space of model parameters $\theta = (p, x_0)$. The explicit time-dependency via $u(t)$ corresponds to external driving forces, like drug stimuli in biological dynamic systems.

In many fields where ODE models are used, parameter values are not *a priori* known and have to be estimated from experimental data. Commonly, this is achieved by minimizing an objective function $g(\theta, D)$ that penalizes weighted differences between model prediction $x(t)$ and data D , e.g., by maximum-likelihood estimation. For the case of non-linear ODE systems, several local optima may exist. In order to find the global optimum, several optimization methods, e.g., particle swarm optimizers (Peng et al., 2010) or simulated annealing (Xiang and Gong, 2000), include stochasticity to escape local minima. Compared to that, deterministic algorithms may stick to local optima during optimization. On the other hand, gradient and Hessian information of the objective function can be incorporated, increasing the performance of optimization by a multiple. Combining the advantages of derivative-based optimization and random sampling, a multi-start deterministic optimization approach has proven to yield superior overall performance for our problem class (Raue et al., 2013). Throughout this work, we perform optimization by means of a trust-region optimizer from multiple starting positions.

Specially in models of biological systems, available data is sparse and parameters are often non-identifiable. Apart from that, the high-dimensional parameter space hampers parameter sampling. In order to reduce the number of parameters, the system is assumed to initially ($t = 0$) be in a steady-state which is determined by the constraint equation

$$f(x_0, p, 0) = 0. \quad (2)$$

As a standard approach, the steady-state constraint is solved for the initial values x_0 . Since Equation (2) is in general non-linear, this may lead to higher-order polynomial equations for which no general solution is available. Even for a rather simple case of quadratic or cubic equations, solutions are not unique and optimization would have to be performed for all possibilities. Another aspect of steady-state calculation are negative solutions for x_0 and p that appear for certain model specifications. Negative solutions are not only contradicting the biological setting with positively defined concentrations and kinetic parameters but also constitute a problem for optimization. Negative parameter values change the sign of damping terms of the ODE's right-hand side which might lead to rapidly growing solutions and an abort of the optimization before an optimum was reached.

In order to obtain a high convergence probability for the optimization of randomly chosen initial parameter samples, our

aim is to derive non-negative, analytical steady-state expressions, while multiple steady-state expressions are likewise circumvented by a proper choice of kinetic and initial value parameters for which Equation (2) is solved.

Over the last decades, steady-state analysis has been addressed by many algorithms and methods. In the following, we give an overview of existing approaches and summarize their applicability to different types of model equations with a special focus on parameter estimation in ODE models, see **Table 1**.

The earliest-proposed algorithm for deriving steady-states in enzyme-catalyzed systems being described by simple mass-action rules was developed by King and Altman (1956). In the original paper, however, interactions that do not involve the enzyme were not allowed which prohibits applicability to most of today's systems with proteins mediating the activation of other proteins without being part of the reaction. After Chemical Reaction Network Theory (CRNT) was formulated (Horn and Jackson, 1972; Feinberg, 1979), the method of King and Altman has been improved by graph theory (Chou, 1990) and extended to special subclasses of CRNs, e.g., layered signaling cascades (Feliu et al., 2012) and post-translational modification networks (Feliu and Wiuf, 2013). The same authors also published a more general approach for CRNs in Feliu and Wiuf (2012). Here, a set of core variables is introduced serving for a parametrization of the steady-states whereby non-negative solutions are guaranteed due to graph-theoretical arguments.

Another approach developed by Halasz et al. (2013), introduces bilinearities of the system as new variables leading to a linearized system solvable by application of Cramer's rule. The number of bilinearities, however, is restricted and negative steady-state solutions are not prevented.

All mentioned approaches deal with steady-state analysis for CRNs based on mass-action rules. However, modern modeling approaches often make use of special reaction types such as inhibition, Michaelis-Menten or Hill kinetics that cannot be included into standard CRNT without changing the model structure and introducing new dynamical variables. In the approach of Halasz et al. (2013), inhibition and Michaelis-Menten terms can easily be integrated by multiplying the corresponding steady-state equation by the denominator of the rate expression. However, since a state variable is contained in the denominator, this can increase the number of bilinearities significantly.

TABLE 1 | Covered reaction types and positivity for different methods of steady-state determination.

Method	CRN, Mass action, Production, Degradation	Inhibition, Michaelis-Menten	Hill Kinetic	Non-negative Solutions
King-Altman	Yes	No	No	Yes
Feliu and Wiuf	Yes	No	No	Yes
Halasz et al.	Yes	Yes	No	No
Loriaux et al. (<i>py-substitution</i>)	Yes	Yes	Yes	No
Proposed	Yes	Yes	Yes	Yes

In order to avoid problems of higher-order polynomial equations, steady-state equations can be solved not only for initial value- but also for kinetic parameters, which is done in the steady-state solver *py-substitution* developed by Loriaux et al. (2013). From N initial values and M kinetic parameters, a set of N variables is chosen that have to be fixed in Equation (2). In doing so, a lot of freedom is incorporated into the solution. In fact, *py-substitution* is able to solve the very most steady-state equation systems, since in principle N kinetic parameters could be chosen as fixed variables directly leading to a simple linear equation system.

Complementary to analytical approaches, steady-state information can be incorporated into the system by numerically computing the initial conditions during each optimization step. Even gradient information that is necessary for efficient optimization is available by means of the implicit function theorem. A numerical incorporation of steady-state information has the advantage that the complexity of the underlying equation system is in principal not restricted. Furthermore, the implementation remains untouched when model equations are changed. However, convergence of the numerical steady-state calculation is not guaranteed and issues of multiple steady-states cannot be controlled.

In the following section, we present a method to derive non-negative steady-state expressions for a large class of nonlinear ODE models that are based on biochemical reactions. Our approach picks up the idea of solving for kinetic parameters in order to derive unique and simple steady-state expressions. Due to the structure of the ODE system, solving for kinetic parameters often leads to potentially negative steady-state solutions, depending on the point of evaluation in parameter space. By introducing appropriate parameter transformations and exploiting the given model structure, our approach guarantees a non-negative solution space. In the Results section, we show how different steady-state parameterizations influence the optimization procedure and compare our approach to the

standard approach of solving for initial value parameters as well as to a numerical steady-state approach.

2. METHODS

2.1. Theoretical Background

Let us consider a *model* f as an N -dimensional ODE system $\dot{x} = f(x, p)$ with states x , parameters p and no external driving forces, i.e., the ODE is autonomous. We write f as a matrix product

$$f(x, p) = S \cdot F(x, p), \quad (3)$$

of the $N \times M$ -dimensional *stoichiometry matrix* S and the M -dimensional *flux vector* F which depends on states and parameters. For the entries of the flux vector, we allow rational functions of x and p including e.g., mass-action, inhibition, Michaelis-Menten and Hill-Type kinetics. **Table 2** gives an overview of the main reaction types covered by the presented steady-state approach.

We assume that each single flux F_l is proportional to some *flux parameter* k_l and can be written as

$$F_l = k_l \cdot G_l(x, q), \quad (4)$$

where the function G_l only depends on the states and a set of *additional parameters* q taken from the set of all model parameters p . The union of flux parameters k and additional model parameters q coincides with the parameter set p . Typically all reaction types described by CRNT only need one flux parameter and do not contribute to q , however, inhibition terms and Michaelis-Menten kinetics contain at least one additional parameter and Hill kinetics even two.

The signs of the entries of the stoichiometry matrix S determine whether a flux contributes as an *in-* or an *outflux* to the time evolution of the corresponding state. We assume that each outflux is at least linearly dependent on the corresponding

TABLE 2 | Examples of typical reaction types of ODE models.

#	Stoichiometry	Reaction type	Flux	Contribution to $\frac{dA}{dt}$
1	$\emptyset \rightarrow A$	Production	k	Positive/influx
2	$A \rightarrow \emptyset, B$	Degradation, transformation	$k \cdot A$	Negative/outflux
3	$A + A \rightarrow AA$	Dimerization	$k \cdot A^2$	Negative/outflux
4	$B \rightarrow A$	Transformation	$k \cdot B$	Positive/influx
5	$B + C \rightarrow A$	Binding	$k \cdot B \cdot C$	Positive/influx
6	$B \rightarrow A$	Inhibition by C	$k \cdot B \cdot \frac{1}{q+C}$	Positive/influx
7	$B \rightarrow A$	Michaelis-Menten	$k \cdot B \cdot \frac{1}{q+B}$	Positive/influx
8	$A + B \rightarrow C$	Binding	$k \cdot A \cdot B$	Negative/outflux
9	$A \rightarrow B$	Inhibition by C	$k \cdot A \cdot \frac{1}{q+C}$	Negative/outflux
10	$A \rightarrow B$	Michaelis-Menten	$k \cdot A \cdot \frac{1}{q+A}$	Negative/outflux
11	$B \rightarrow A$	Hill	$k \cdot \frac{B^{q_2}}{q_1^{q_2} + B^{q_2}}$	Positive/influx
12	$\emptyset \rightarrow A$	Self-activation	$k \cdot \frac{A^{q_2}}{q_1^{q_2} + A^{q_2}}$	Positive/influx
13	$B + C \rightarrow A$	Power-law	$k \cdot B^{q_1} \cdot C^{q_2}$	Positive/influx

All types are covered by our steady-state approach.

state, as being always the case for mass-action systems. By means of Equation (2), each initial value $x_{0,i} \equiv x_i$ is directly related with a steady-state equation of the form

$$0 = \sum \text{in}_i - x_i \cdot \sum \text{out}_i, \tag{5}$$

where in_i and out_i constitute functions of states and parameters. For a majority of reaction types used in ODE models, the fluxes in_i and out_i are independent of x_i , compare **Table 2**. In these cases, Equation (5) is linear in x_i and has the solution $x_i = \frac{\sum \text{in}_i}{\sum \text{out}_i}$. However, if the fluxes in_i or out_i still depend on x_i , e.g., reaction 12 in **Table 2** for the case of self-activation or reaction 3 with an outflux being quadratic in x_i , Equation (5) might be non-linear in x_i .

In order to solve the complete set of steady-state equations, we analyze their specific structure by means of graph theory. We therefore rewrite Equation (5) as

$$x_i = \frac{\sum \text{in}_i}{\sum \text{out}_i} \tag{6}$$

and summarize appearances of states on the right-hand side of Equation (6). Here, the set of states is defined by the set of dynamic variables x that we want to fix by the steady-state determination. Once a dynamic variable is fixed by a non-negative expression or treated as a free parameter, it is removed from the set of states.

Definition 1: A *head* of state x_i is a state x_j that appears on the right-hand side of Equation (6). By $h(x_i)$, we refer to the set of heads for a specific state x_i . In particular, x_i can itself be a part of $h(x_i)$.

Proposition 1: If non-negative steady-state solutions for all heads of x_i are known, a non-negative steady-state solution for x_i can directly be obtained by Equation (6). This holds especially, if the set $h(x_i)$ is empty.

Definition 2: The *adjacency matrix* $M(f)$ of an ODE model $f(x, p)$ with states x and parameters p is an $N \times N$ matrix with entries

$$M_{ji} = \begin{cases} 1, & \text{if } x_j \in h(x_i) \\ 0, & \text{else.} \end{cases}$$

Each d_M -dimensional adjacency matrix M defines a *directed graph* G_M with nodes x_1 to x_{d_M} which we call *steady-state graph*. Each non-zero entry of M corresponds to a directed edge (x_j, x_i) implying that x_j occurs in the steady-state expression of x_i , i.e., Equation (6). A non-zero diagonal entry M_{ii} reflects that the corresponding steady-state equation is non-linear in x_i .

2.2. Splitting Cycles

The specific structure of the steady-state graph enables to solve the steady-state equations step-by-step as is shown in the following.

Definition 3: A *cycle* of a steady-state graph is a path through the graph along its edges with equal starting and end point. Here, we allow cycles of length one arising from non-zero diagonal entries in the adjacency matrix M .

Definition 4: Graphs that do not contain cycles are called *tree-like*.

Proposition 2: If a steady-state graph of an N -dimensional model f is tree-like, non-negative steady-state solutions can be obtained for all x_i inside the graph.

Proof: For any tree-like steady-state graph, there exists at least one *root*, i.e., state without head, called x_r . Since $h(x_r) = \emptyset$, the corresponding steady-state expression can be obtained by *Proposition 1*. In doing so, x_r is removed from the steady-state graph and a new state serves as root for which *Proposition 1* again gives the corresponding steady-state expression. By iteratively applying *Proposition 1* for each of the N nodes, the complete steady-state solution is obtained.

Considering *Proposition 2*, it is clear that solving the steady-state constraint Equation (2) for the set of initial values only becomes intricate, if there are cycles inside the steady-state graph such that higher-order polynomial equations arise. The idea of our steady-state approach is to split all these cycles step-by-step such that *Proposition 2* can ultimately be applied to the remaining graph.

The simplest way of splitting a cycle is by means of a conserved quantity (CQ) of the system arising from the stoichiometry. A general introduction can be found in Loriaux et al. (2013) or Halasz et al. (2013). The following definition restricts to the properties being relevant for the presented approach.

Definition 5: A *conserved quantity* (CQ) of the model f is an expression of states and parameters which remains constant during the time-evolution of f . For each CQ, the number of independent steady-state equations is reduced by one implying that one state or flux parameter that appears in the CQ can be chosen freely. If all CQs can be derived from the stoichiometry matrix, the number of CQs is given by $n_{cq} = N - R_S$, with the model size N and the rank of the stoichiometry matrix R_S . The cases for which $n_{cq} > N - R_S$ are discussed in Section 2.5.

In order to split a cycle by a CQ, one of the states, x_c , that appears both inside the cycle and in the CQ is chosen freely. The corresponding steady-state equation is removed from Equation (2), whereby the number of independent steady-state equations remains constant. Since the state x_c is treated as a free variable, all edges originating from and leading to x_c can be removed from the steady-state graph and the considered cycle is split. Note, that each CQ can only be used once.

If no states inside the cycle appear in CQs, the cycle can be split by solving the steady-state equation of a specific cycle state x_i for a flux parameter k_l . By means of Equations (3) and (4), the steady-state expression of k_l holds

$$k_l = \frac{-1}{S_{il}G_l(x, q)} \sum_{j \neq l} S_{ij}k_jG_j(x, q). \tag{7}$$

Proposition 3: Let n_{k_l} be the number of appearances of the flux parameter k_l , see Equation (4), inside the steady-state constraint Equation (2). Then n_{k_l} coincides with the number of non-zero entries in the l -th column of the stoichiometry matrix S .

Unless k_l does not appear in other steady-state equations, i.e., $n_{k_l} = 1$, the considered cycle is removed from the steady-state graph without affecting other parts of the graph. However, if

$n_{k_l} > 1$, all further appearances have to be substituted by Equation (7) which creates new edges inside the steady-state graph and possibly even new cycles. In order to keep the structure as simple as possible, flux parameters with $n_{k_l} = 1$ play a special role.

2.3. Enforcing Positivity

Although solving for flux parameters implies linear equations and therefore structurally simple steady-state expressions, the solutions are often negative for certain model specifications. Here, we show how positivity of the expressions can be guaranteed by appropriate transformations.

The steady-state expression, Equation (7), of the flux parameter k_l was derived by means of the steady-state equation of x_i . The expression contains minus signs if and only if at least one of the stoichiometry entries $S_{i,j \neq l}$ has the same sign as S_{il} . In this case, at least one further flux contributes to x_i with the same sign as $F_l = k_l G_l$, namely as an in- or outflux.

Definition 6: For the steady-state equation of x_i , Equation (5), we define μ_i and ν_i as the number of in- and outfluxes, respectively. Furthermore, we define the *dimension* of the state x_i as the minimum $\dim(x_i) = \min(\mu_i, \nu_i)$.

If $\dim(x_i) = 1$, a non-negative steady-state expression is obtained by solving for the particular flux parameter being the only in- or outflux, compare first examples in **Table 3**.

If $\dim(x_i) > 1$, positivity can be enforced by performing an appropriate parameter transformation. In order to do so, we divide the fluxes contributing to x_i into influxes $F_{in,1} \dots F_{in,\mu_i}$ and outfluxes $F_{out,1} \dots F_{out,\nu_i}$. Then, Equation (5) reads

$$0 = \sum_{j=1}^{\mu_i} F_{in,j} - \sum_{l=1}^{\nu_i} F_{out,l} \tag{8}$$

Let us assume that we want to solve Equation (8) for the influx parameter $k_{in,1} = \frac{F_{in,1}}{G_{in,1}}$. We perform a variable transformation by defining the ratio between the remaining influxes and $F_{in,1}$ as

$$r_z = \frac{F_{in,z}}{F_{in,1}} = k_{in,z} \cdot \frac{G_{in,z}}{k_{in,1} G_{in,1}} \quad \text{for } z = 2, \dots, \mu_i, \tag{9}$$

TABLE 3 | Examples of solving steady-state equations for flux parameters.

Steady-state equation	Solve for	Type	Solution
$0 = k_1 A^2 - k_2 X$	k_1	1	$k_1 = \frac{k_2 X}{A^2}$
$0 = k_1 A^2 - k_2 X$	k_2	1	$k_2 = \frac{k_1 A^2}{X}$
$0 = k_1 A^2 - k_2 X - \frac{k_3 X B}{q_1 + C} - \frac{k_4 X}{q_1 + C}$	k_1	1	$k_1 = \frac{X}{A^2} \left(k_2 + k_3 B + \frac{k_4}{q_1 + C} \right)$
$0 = k_1 A^2 + k_5 B - k_2 X - \frac{k_3 X B}{q_1 + C} - \frac{k_4 X}{q_1 + C}$	k_1, k_5	2	$k_1 = \frac{1}{A^2} X \left(k_2 + k_3 B + \frac{k_4}{q_1 + C} \right) \cdot \frac{1}{1+r}$ $k_5 = \frac{1}{B} X \left(k_2 + k_3 B + \frac{k_4}{q_1 + C} \right) \cdot \frac{r}{1+r}$

where the r_z replace the kinetic parameters k_2 to k_{μ_i} . By means of Equations (8) and (9), we obtain

$$\sum_l F_{out,l} = F_{in,1} \cdot \left(1 + \sum_{j=2}^{\mu_i} r_{in,j} \right)$$

and therefore

$$k_{in,1} = \frac{1}{G_{in,1}} \sum_l F_{out,l} \cdot \frac{1}{1 + \sum_{j=2}^{\mu_i} r_j} \tag{10}$$

Since $G_{in,1}$ and $F_{out,l}$ are positive and Equation (10) is a sum of positive contributions, a non-negative steady-state expression for $k_{in,1}$ is guaranteed. By means of Equation (9), the remaining flux parameters have to be substituted by the non-negative expressions

$$k_{in,z} = \frac{1}{G_{in,z}} \sum_l F_{out,l} \cdot \frac{r_z}{1 + \sum_{j=2}^{\mu_i} r_j} \quad \text{for } z = 2, \dots, \mu_i. \tag{11}$$

For an outflux parameter, we analogously obtain

$$k_{out,1} = \frac{1}{G_{out,1}} \sum_j F_{in,j} \cdot \frac{1}{1 + \sum_{l=1}^{\nu_i} r_l} \quad \text{and} \tag{12}$$

$$k_{out,z} = \frac{1}{G_{out,z}} \sum_j F_{in,j} \cdot \frac{r_z}{1 + \sum_{l=1}^{\nu_i} r_l} \quad \text{for } z = 2, \dots, \nu_i. \tag{13}$$

2.4. Algorithm for Steady-State Determination

In the previous sections, we showed how simple steady-state expressions can be obtained (Section 2.2), while positivity is likewise guaranteed (Section 2.3). In order to split one cycle of the steady-state graph and solve for a flux parameter, a pair (x_i, k_j) of state and flux parameter has to be chosen, which is not unique. In the following, we suggest an algorithm based on a classification of such pairs.

According to Definitions 5, 6 and Proposition 3, we associate each pair with one of four different types:

$$(x_i, k_j) \equiv \begin{cases} \text{Type 0, if } x_i \text{ appears in a CQ} \\ \text{Type 1, } n_{k_j} = 1 \text{ and } \dim(x_i) = 1 \\ \text{Type 2, } n_{k_j} = 1 \text{ and } \dim(x_i) > 1 \\ \text{Type 3, else.} \end{cases}$$

Figure 1 shows a flowchart of the algorithm. At first, the set of CQs is computed for the ODE system serving as an input for the algorithm. If the graph is tree-like, the remaining equations are obtained according to *Proposition 2* and the complete set of steady-state equations is returned.

In the case of a pair of Type 0, the cycle can simply be removed by interpreting the corresponding state as a free variable. The CQ that is thereby used is removed from the set of CQs and cannot

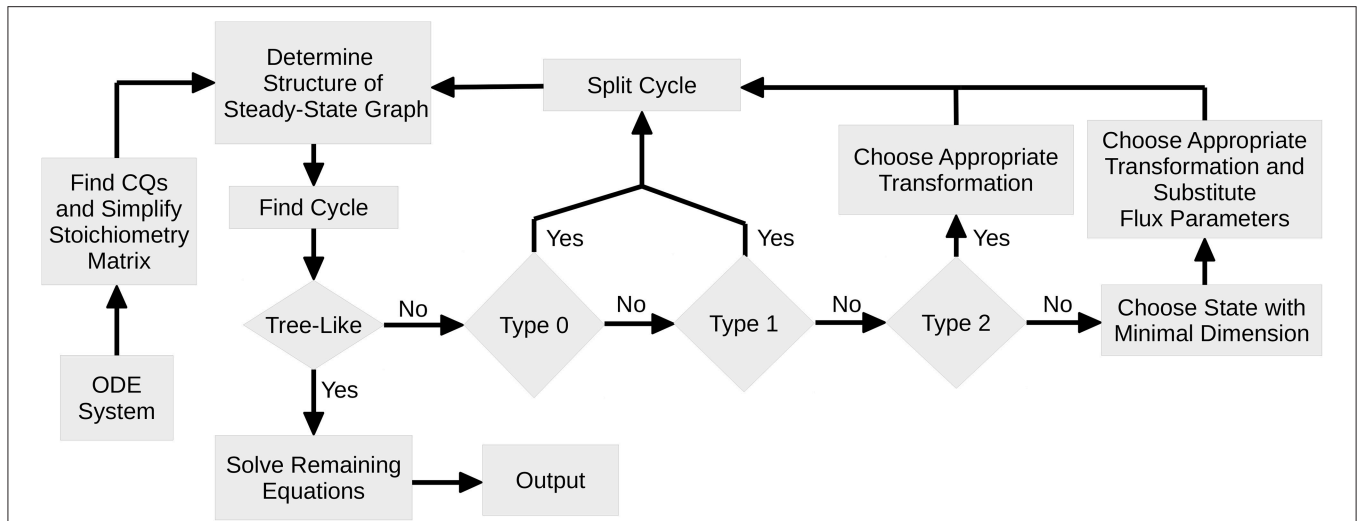


FIGURE 1 | Flowchart of Steady-State Determination: After identification of all CQs of the system, the algorithm performs a loop where in each pass one cycle of the steady-state graph is removed. Since for each cycle the nodes are analyzed with regard to the number of in- and outflux rates, the graph structure as well as the structure of the steady-state equations is kept as simple as possible. Once the steady-state graph is tree-like, the remaining equations are solved and equations are returned.

further contribute to the steady-state determination. Here, the flux parameters remain unaffected.

Unless the cycle cannot be directly split by means of a CQ, the corresponding steady-state equation, Equation (5), is solved for one of the $\mu_i + v_i$ flux parameters by use of a pair of Type 1,2 or 3. In order to keep the steady-state solution as simple as possible, pairs of Type 1 are preferred, since this enables to split the cycle both without substituting the flux parameter by its steady-state expression, Equation (7), and without introducing flux ratios as new parameters, Equation (9).

If no pairs of Type 1 are available, the algorithm scans the steady-state graph for pairs of Type 2. In this case a parameter transformation is necessary in order to guarantee positivity of the solution. However, the flux parameter does not appear any more in the system and therefore has not to be substituted. In all three cases, Type 0, 1, or 2, the number of cycles of the steady-state graph is reduced.

If pairs of Type 2 are also not available, all pairs are of Type 3. In this case, it is not *a priori* clear which pair is the best choice. As a simply revisable choice, the algorithm then solves the steady-state equation of the state with minimal dimension. Subsequently, all further appearances of flux parameters have to be replaced by their particular transformation, Equations (11) or (13).

2.5. Calculating the Conserved Quantities and Simplifying the Stoichiometry Matrix

In order to find CQs of the ODE system, linear combinations of rows of the stoichiometry matrix S can be analyzed. According to Equation (3), the N -dimensional ODE system can be written as

$$\dot{x} = S \cdot F(x, p). \tag{14}$$

Multiplication of Equation (14) by an $N \times N$ -matrix M yields

$$M \cdot \dot{x} = \tilde{S} \cdot F(x, p), \tag{15}$$

where the matrix $\tilde{S} = M \cdot S$ defines linear combinations of rows of S . For each row \tilde{S}_i that is equal to zero, the quantity $M_i \cdot x = \sum_j M_{ij}x_j$ is conserved.

In fact, each set of linearly dependent rows of S implies a CQ. For some ODE systems, however, not all CQs can be derived from S without accounting for the flux vector F . Equation (14) can be written

$$\dot{x} = C(p, x) \cdot x. \tag{16}$$

where $C(p, x)$ is an $N \times N$ -matrix dependent on the parameters and states. Analyzing linear dependencies of C , all CQs of the form

$$\sum_j a_j(x, p) \cdot x_j = \text{const.} \tag{17}$$

can be found, where the coefficients a_j might depend on states and parameters.

In order to determine symbolic expressions for the a_j , we transpose the matrix C and numerically search for linearly dependent columns. All parameters and states appearing in C^T are replaced by random values to obtain a numeric matrix C_{ran}^T for which a QR -decomposition is performed. The matrix R constitutes an upper triangular matrix, where the number of non-empty rows corresponds to the rank of C_{ran} . The first column R_ℓ with $R_{\ell\ell} = 0$ is a linear combination of the columns $R_{j<\ell}$. Therefore, also the column C_ℓ is a linear combination of the columns $C_{j<\ell}$ implying that the equation system $\sum_j a_j C_j = 0$

has a solution for the a_j with $a_{j>\ell} = 0$ which can be calculated symbolically. Thus, the quantity $a^T \cdot x$ is conserved. Once a CQ has been found, one of the corresponding linearly dependent rows of the stoichiometry matrix is removed and the procedure is repeated. In most ODE systems, all CQs of the system can be obtained in that way. For all other cases, our Python code provides a possibility to manually specify CQs.

The idea of taking linear combinations of the stoichiometry matrix S , see Equations (14) and (15), can be augmented to simplify S for the calculation of steady-state expressions. For each matrix M , the original steady-state constraint, $S \cdot F = 0$, is replaced by a new set of steady-state equations, $\tilde{S} \cdot F = 0$. With a clever choice of M , these new steady-state equations might be structurally simpler than the original ones. With respect to our proposed algorithm, the matrix M should (1) minimize the overall number of entries in the new stoichiometry matrix \tilde{S} and (2) prevent the creation of new cycles. In practice, the idea of linearly combining rows of the stoichiometry matrix can lead to structurally simpler steady-state expressions as we show by means of a small example in the Supplementary Material.

2.6. Numerically Computed Steady-States

Besides calculating steady-states analytically, roots of the steady-state constraint, Equation (2), can be computed numerically during each step of the optimization. Here, we perform Newton's method which is fast compared to the time of the ODE integration. The gradient information that is necessary within our deterministic optimization scheme is determined by the implicit function theorem, i.e., given the steady-state constraint

$$f(x_0(p), p) = 0,$$

we derive the equation with respect to p and obtain

$$0 = \frac{\partial f}{\partial x_0} \cdot \frac{\partial x_0}{\partial p} + \frac{\partial f}{\partial p} \quad \implies \quad \frac{\partial x_0}{\partial p} = - \left(\frac{\partial f}{\partial x_0} \right)^{-1} \frac{\partial f}{\partial p}.$$

2.7. Technical Remarks

The steady-state algorithm was implemented in Python by use of the libraries `numpy` and `sympy`. It can either be downloaded from the author's homepage as a Python code or can be used from within the R-packages `dMod/cOde` available from <https://github.com/dkaschek/>. Simulation of data and parameter estimation with analytical and numerical steady-states were performed in `dmod`.

3. RESULTS

When calculating steady-state expressions for parameter estimation of ODE systems, several aspects have to be considered simultaneously. Most importantly, the parameter space is to be reduced as far as possible. Therefore, all available steady-state constraints should be taken into account. Since solving for state variables often leads to higher-order equations for which solutions are difficult to obtain, one has at least partially to solve for kinetic parameters. In doing so, the steady-state expressions

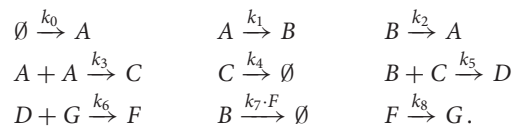
often lead to negative parameter values for certain model specifications.

Due to mass balance, outfluxes contribute with a minus sign to the time derivative of the corresponding state. Provided that outflux rates are proportional to positive powers of their states, they contribute damping terms to the time-evolution of the state. However, if for a certain model specification the corresponding flux parameter is negative, the sign of the outflux term becomes positive which leads to an exploding model trajectory for the state.

In Section 3.1, we show how our steady-state approach determines simple steady-state equations for systems that lead to higher-order equations when solved for the state variables. In Section 3.2, we show how steady-state expressions with negative realizations lead to optimization problems and a significantly lower success rate, i.e., the probability to converge to a local or the global optimum. Non-linear ODE systems often have several steady-state solutions, when the steady-state equations are solved for the state variables. For parameter estimation, multiple steady-states constitute a problem, since all possible realizations have in principle to be followed up. By solving for kinetic parameters, our steady-state approach likewise avoids multiple solutions and improves the optimization as we show in Section 3.3.

3.1. Determination of Non-Negative Steady-State Expressions

To show the applicability of the presented steady-state approach, we investigate a toy model with six state variables and nine reactions of the form



All reactions satisfy the law of mass action, the degradation of B is mediated by F . With these assumptions, one obtains the following ODE system

$$\begin{aligned} \dot{A} &= k_0 + k_2B - k_1A - k_3A^2 \\ \dot{B} &= k_1A - k_2B - k_5BC - k_7BF \\ \dot{C} &= k_3A^2 - k_5BC - k_4C \\ \dot{D} &= k_5BC - k_6DG \\ \dot{F} &= k_6DG - k_8F \\ \dot{G} &= k_8F - k_6DG. \end{aligned}$$

The system contains one conserved quantity $F + G = \text{const}$, reflecting that the steady-state equations of F and G are not independent from each other. Therefore, the number of variables that have to be fixed by the steady-state is five. In order to obtain the corresponding steady-state equations, all time-derivatives of the states are set to zero. Although the single equations of this system are of degree two or lower, solving for the states leads to a sixth order polynomial equation, see Supplementary Material, for which no closed-form solution is available.

TABLE 4 | Steps of steady-state determination for a model with six states and eight reactions.

Loop	Steady-state graph	Steady-state equations	Simplification
1		$A: 0 = k_0 + k_2B - k_1A - k_3A^2$ $B: 0 = k_1A - k_2B - k_5BC - k_7BF$ $C: 0 = k_3A^2 - k_5BC - k_4C$ $D: 0 = k_5BC - k_6DG$ $F: 0 = k_6DG - k_8F$ $G: 0 = k_8F - k_6DG$	Cycle: [A, A] Type 3 $k_0 = \frac{A^2k_3 + Ak_1}{r_1 + 1}$ $k_2 = r_1 \frac{k_0}{B} = r_1 \frac{A^2k_3 + Ak_1}{B(r_1 + 1)}$
2		$B: 0 = k_1A \frac{1}{r_1 + 1} - k_3A^2 \frac{r_1}{r_1 + 1} - k_5BC - k_7BF$ $C: 0 = k_3A^2 - k_5BC - k_4C$ $D: 0 = k_5BC - k_6DG$ $F: 0 = k_6DG - k_8F$ $G: 0 = k_8F - k_6DG$	Cycle: [B, B] Type 1 $k_1 = \frac{r_1 + 1}{A} [k_3A^2 \frac{r_1}{r_1 + 1} + k_5BC + k_7BF]$
3		$C: 0 = k_3A^2 - k_5BC - k_4C$ $D: 0 = k_5BC - k_6DG$ $F: 0 = k_6DG - k_8F$ $G: 0 = k_8F - k_6DG$	Cycle: [D, G, D] Type 0 G is part of CQ
4		$C: 0 = k_3A^2 - k_5BC - k_4C$ $D: 0 = k_5BC - k_6DG$ $F: 0 = k_6DG - k_8F$	Tree-like Solve for C, D and F

Table 4 summarizes how our steady-state solver determines a non-negative steady-state solution by partially solving for flux parameters. During the first loop, the cycle [A, A] of state A to itself is split. The pairs of A and its contributing flux parameters are all of Type 3, since there are two influx- and two outflux parameters of which at least one is appearing in the other steady-state equations, e.g., in the equation of B. The equation of A is solved for the influx parameter k_0 , whereby k_2 is transformed and replaced by the new free parameter $r_1 = k_2B/k_0$, see first loop in **Table 4**. The appearance of k_2 in the equation of B is substituted, whereas k_0 has no further appearances.

Whereas the state A is removed from the steady-state graph, the state B has become a new head of itself in consequence of the substitution. In the second loop, this new cycle [B, B] is split by solving the equation of B for the flux parameter k_1 . Here, the pair (B, k_1) is of Type 1, since k_1 is the only influx parameter and not appearing in the remaining equations.

In the next loop, the algorithm splits the cycle [D, G, D] by taking G as a free parameter, since it is part of the conserved quantity $F + G$. The remaining steady-state graph in the last loop is tree-like and therefore the steady-state equations can be derived according to *Proposition 2* starting with C which in this case serves as the root of the graph.

For simplification of writing, our steady-state solver outputs the equations in a specific order where fixed states or parameters may still appear in the equations below. In order to obtain a complete independent set of equations, one has to replace step by step. For the presented example, the ultimately obtained expressions are

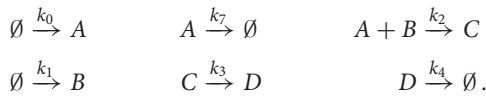
$$\begin{aligned}
 F &= A^2 \frac{k_3}{k_8} \frac{Bk_5}{Bk_5 + k_4} \\
 D &= A^2 \frac{k_3}{Gk_6} \frac{Bk_5}{Bk_5 + k_4} \\
 C &= A^2 \frac{k_3}{Bk_5 + k_4} \\
 k_1 &= Ak_3r_1 + \frac{r_1 + 1}{A} (Bck_5 + BFk_7) \\
 k_2 &= r_1 \frac{k_3A^2 + k_1A}{B(r_1 + 1)} \\
 k_0 &= \frac{k_3A^2 + k_1A}{r_1 + 1},
 \end{aligned}$$

where six parameters are fixed, while one additional parameter r_1 can be chosen freely.

3.2. Minus Signs Imply a Low Convergence Rate

For a given data set and a given ODE model, each parameter set determines the time-evolution of the states and its likelihood L can be computed based on the data. Here, parameter values are estimated by minimizing the negative log-likelihood function $-\log L$. For the case of non-linear ODE models, several local optima may exist. In order to find the global optimum, we perform multi-start optimization in combination with a trust-region optimizer. A powerful optimization approach should have a high probability to find a local or the global optimum.

Let us consider an ODE system with four state variables and six reactions of the form



The corresponding ODE system is given by

$$\begin{aligned} \dot{A} &= k_0 - k_7A - k_2AB \\ \dot{B} &= k_1 - k_2AB \\ \dot{C} &= k_2AB - k_3C \\ \dot{D} &= k_3C - k_4D. \end{aligned}$$

In order to test if negative steady-state expressions lead to optimization problems, we implemented four different steady-state parameterizations, see **Table 5**, and compared the success rate of parameter optimization. For each approach, six parameters are optimized as shown in **Table 5**. Besides the standard approach, i.e., exclusively solving for initial values, two other parameterizations were derived by solving the equation of state *B* for two different kinetic parameters, namely k_7 and k_0 . The latter guarantees a non-negative steady-state solution. Apart from that, a fourth parameterization was constructed by adding the equation $k_0 = k_1 + \Delta_{k_0}$ to the standard steady-state formulation. In doing so, k_0 is transformed such that $k_0 > k_1$, with the new free parameter Δ_{k_0} describing the difference between k_0 and k_1 . This approach likewise implies positivity.

TABLE 5 | Several steady-state representations for a model with four states and six flux parameters.

Steady-state method	Steady-state equations	Parameters to be estimated
Solved for initial values (Standard)	$A = k_0 / (k_2B + k_7)$ $B = k_1 k_7 / (k_2(k_0 - k_1))$ $C = k_1 / k_3$ $D = k_1 / k_4$	k_1, k_2, k_3, k_4 k_0, k_7
Equation of <i>B</i> solved for k_7	$A = k_0 / (k_2B + k_7)$ $k_7 = Bk_2(k_0 - k_1) / k_1$ $C = k_1 / k_3$ $D = k_1 / k_4$	k_1, k_2, k_3, k_4 k_0, B
Equation of <i>B</i> solved for k_0	$A = k_0 / (k_2B + k_7)$ $k_0 = k_1 k_7 / (Bk_2) + k_1$ $C = k_1 / k_3$ $D = k_1 / k_4$	k_1, k_2, k_3, k_4 B, k_7
Standard with additional parameter transformation	$k_0 = k_1 + \Delta_{k_0}$ $A = k_0 / (k_2B + k_7)$ $B = k_1 k_7 / (k_2(k_0 - k_1))$ $C = k_1 / k_3$ $D = k_1 / k_4$	k_1, k_2, k_3, k_4 Δ_{k_0}, k_7

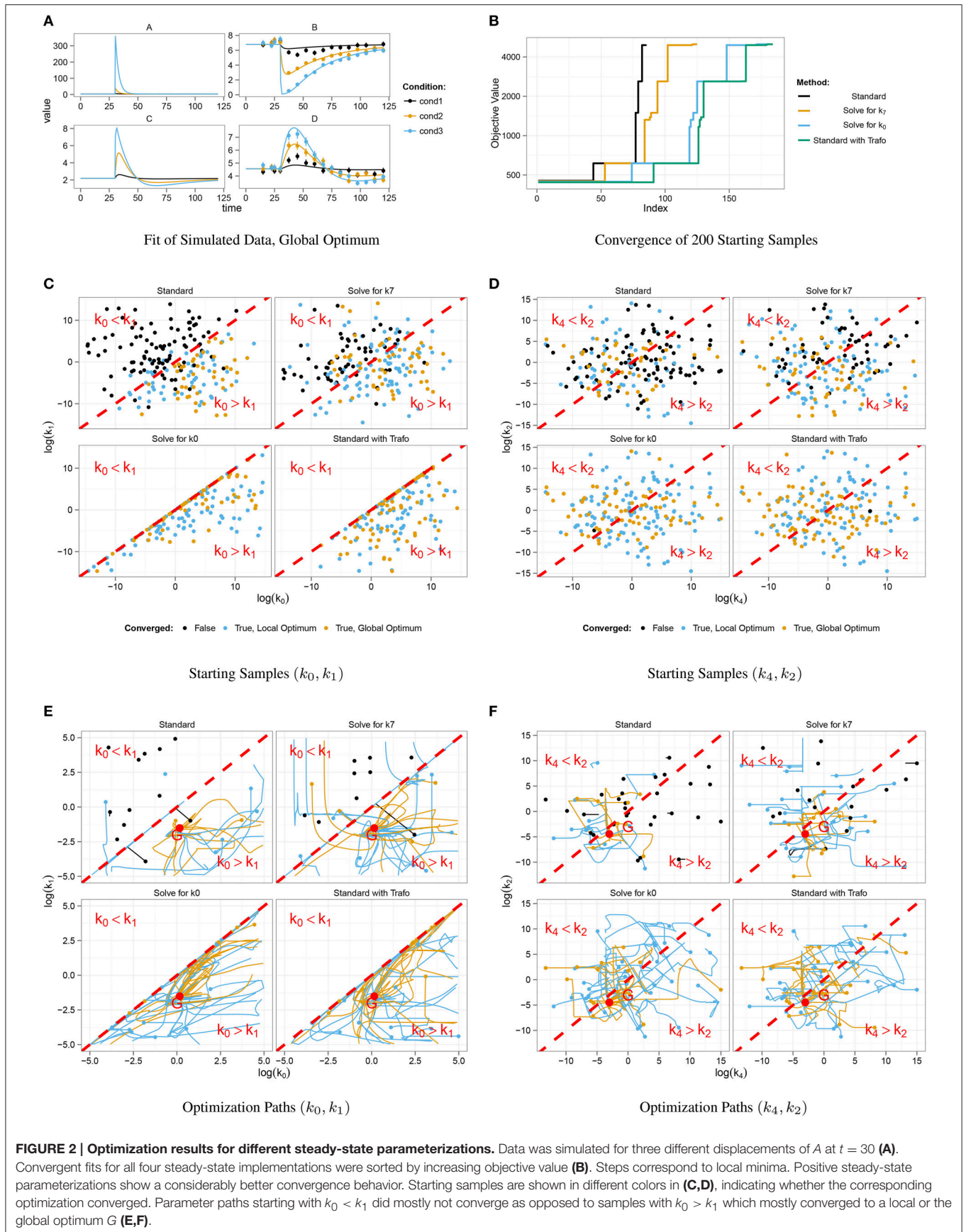
For simulation of data, we chose a set of kinetic parameters for ODE integration, initialized the system with its steady-state and excited it by displacement of *A* at time point $t = 30$. Data points were generated for 16 different time points by adding normally distributed noise to the model trajectories. In order to study a scenario with different experimental conditions, i.e., different stimulations, simulation was done for three different displacement values, compare *cond1*, *cond2*, and *cond3* in **Figure 2A**.

Figure 2A shows data points and trajectories of a model fit that reached the global optimum. For each steady-state parameterization, we optimized 200 different parameter samples and counted how often several optima were reached. In **Figure 2B**, all converged fits are shown in order of the objective value, in our case the negative log-likelihood value. Several steps corresponding to local optima appear for all steady-state parameterizations, the deepest step corresponds to the global optimum. It can be concluded that the two parameterizations without minus signs, i.e., *Solving for k_0* and *Standard with Trafo*, show a significantly better convergence than the other two. For example, in the parameterization with transformation, the global optimum was twice as often reached than in the standard approach.

In order to explain the convergence behavior of the different steady-state implementations, we analyzed the correlation between initial parameter guess and the success of optimization. The steady-state of the presented model is negative, if and only if $k_0 < k_1$. **Figure 2C** shows the starting samples along the parameter axes of k_0 and k_1 for all four steady-state parameterizations, colors indicate whether a sample did not converge (black) or did converge to a local (blue) or the global optimum (yellow). For comparison, **Figure 2D** shows starting samples along axes of parameters that do not affect the sign of the steady-state, namely k_2 and k_4 . The sample distribution shows that samples with $k_0 > k_1$ have a high probability to converge, while samples with $k_0 < k_1$ tend to abort. On the other hand, the relation of k_2 and k_4 does not have a significant impact on the convergence probability. Furthermore, **Figure 2C** shows that the reparameterized steady-states prohibit sampling in the region with $k_0 < k_1$.

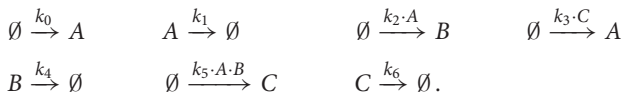
In addition to the starting samples, we analyzed the parameter paths during the optimization. **Figures 2E,F** show the paths for the first 50 starting samples with respect to the above used parameter axes. Parameter samples with $k_0 < k_1$ usually abort without any considerable steps in parameter space even though several samples cross the border $k_0 = k_1$ and proceed. In the opposite direction, some samples reach the border when started in the area with $k_0 > k_1$ and abort exactly at the border. The very most samples drawn with $k_0 > k_1$ converged to a local or the global optimum *G*. Again, **Figure 2F** underlines that the convergence behavior is unaffected by the relation of k_2 and k_4 .

We conclude that steady-state parameterizations that lead to negative parameter values for certain model specifications constitute a severe issue for optimization. Due to the formulation of our steady-state algorithm, negative solutions are automatically avoided in the obtained steady-state expressions.



3.3. Dealing with Multiplicity of Steady-States

Let us consider a system with three state variables and seven reactions of the form



The production of B is mediated by A , production of A is mediated by C and production of C is mediated by both A and B . The corresponding ODE system is given by

$$\begin{aligned} \dot{A} &= k_0 + k_3 C - k_1 A \\ \dot{B} &= k_2 A - k_4 B \\ \dot{C} &= k_5 AB - k_6 C. \end{aligned} \quad (18)$$

For this system, the steady-state can still be analytically solved for the states A , B and C . The solution reads

$$\begin{aligned} C_{1/2} &= \frac{1}{2k_2 k_3 k_5} \left(\frac{1}{k_1^2 k_4 k_6} \Delta + 2k_0 k_2 k_3 k_5 \pm \sqrt{\Delta} \right) \\ B_{1/2} &= \frac{1}{2k_1 k_3 k_4 k_5} \left(k_1^2 k_4 k_6 \pm \sqrt{\Delta} \right) \\ A_{1/2} &= \frac{1}{2k_1 k_2 k_3 k_5} \left(k_1^2 k_4 k_6 \pm \sqrt{\Delta} \right), \end{aligned} \quad (19)$$

with the discriminant $\Delta = k_1^2 k_4 k_6 \cdot (k_1^2 k_4 k_6 - 4k_0 k_2 k_3 k_5)$. For $\Delta > 0$, two positive steady-state solutions $S_1 = (A_1, B_1, C_1)$ and $S_2 = (A_2, B_2, C_2)$ are obtained, while the system has no real steady-state for $\Delta < 0$.

Linear stability analysis reveals that solution S_1 is unstable and solution S_2 is stable, see Supplementary Material. If several steady-state solutions exist, only one of them can be chosen for an optimization run at a time. Here, the stable solution was chosen.

For this system, the issue of multiplicity can easily be solved, however, for more complicated systems several stable solutions might exist. Stable solutions might even switch to unstable solutions along the optimization path, e.g., in case of a Hopf bifurcation. For higher-order equations, analytical solutions become unfeasible and numerical steady-state computation comes into play. However, since the numerical root finding is performed by means of Newton's method, the result depends on initial guesses for A , B and C . Consequently, it is not clear which of the solutions is obtained and stability of the retrieved steady-state is not guaranteed. As we will show, coexistence of stable and unstable steady-state solutions leads to a reduced convergence probability in the numerical approach.

Unlike solving the steady-state equations for A , B and C , the steady-state expressions obtained by our proposed approach are

$$C = A^2 \frac{k_2 k_5}{k_4 k_6} \quad B = A \frac{k_2}{k_4} \quad k_1 = A \frac{k_2 k_3 k_5}{k_4 k_6} + \frac{k_0}{A}, \quad (20)$$

where the kinetic parameter k_1 is fixed, while the initial value of A is taken as a free parameter. The obtained solution is unique, since the steady-state equations are linear in the parameters B , C and k_1 .

In general, our approach avoids multiple-steady-states by choosing a combination of parameters for which the steady-state equations are linear. In doing so, no solution is neglected as long as all steady-state equations are fulfilled. As an analogon, let us consider a single algebraic equation of the form $ab^3 + cb^2 + db + f = 0$, with the five parameters a , b , c , d and f . On the one hand this equation can be solved for b whereby multiple solutions are obtained. On the other hand it can be solved for one of the parameters a , c , d or f for which the equation is linear leading to a unique solution.

In the following, we compare the convergence behavior of three different steady-state implementations, namely *Standard*, i.e., analytically solved for the states, *Numeric*, i.e., numerically solved for the states, and *Proposed*, i.e., our steady-state approach with positive solutions. For the former two implementations, the seven kinetic parameters k_0 to k_6 are estimated, whereas for the proposed approach, the initial value of A is estimated instead of k_1 , compare Equation (20).

Since natural systems are always subject to external noise, unstable steady-states are never realized by the system. Therefore, data was simulated by means of the stable steady-state solution. Analogously to Section 3.2, three different displacements of the state A were triggered at time point $t = 30$ to excite the system. Here, data points were generated for eight different time points.

In order to test the convergence behavior, we started 200 fits from randomly chosen parameter samples. **Figure 3A** shows an example fit that converged to the global optimum. The optimization result of the three steady-state approaches is compared in **Figure 3B**. Steps correspond to local optima. In our approach, nearly half of the samples converged to the global optimum, whereas only about 10% of the fits converged in the standard approach and even less in the numeric approach.

Similar to Section 3.2, we analyzed the correlation between initial parameter guess and success in optimization. **Figure 3D** shows the distribution of starting samples with respect to the sign of the discriminant Δ . For $\Delta < 0$ the discriminant of the standard steady-state expression, Equation (19), becomes negative, and all starting samples drawn from this region did not converge. In addition, **Figure 3E** shows that the optimization of these samples directly aborted, since the path did not take any or at most a very small step in parameter space.

Furthermore, we analyzed the correlation between the coexistence of stable and unstable steady-state solutions and the success of the numerical approach. During each optimization step, the root of the ODE's right-hand side is computed for the current parameter values. Depending on the initial guess, either the stable or the unstable solution is obtained. In order to see, if the unstable solution causes optimization aborts, we chose state A as a representative and compared numerically and analytically calculated values at the end of the optimization

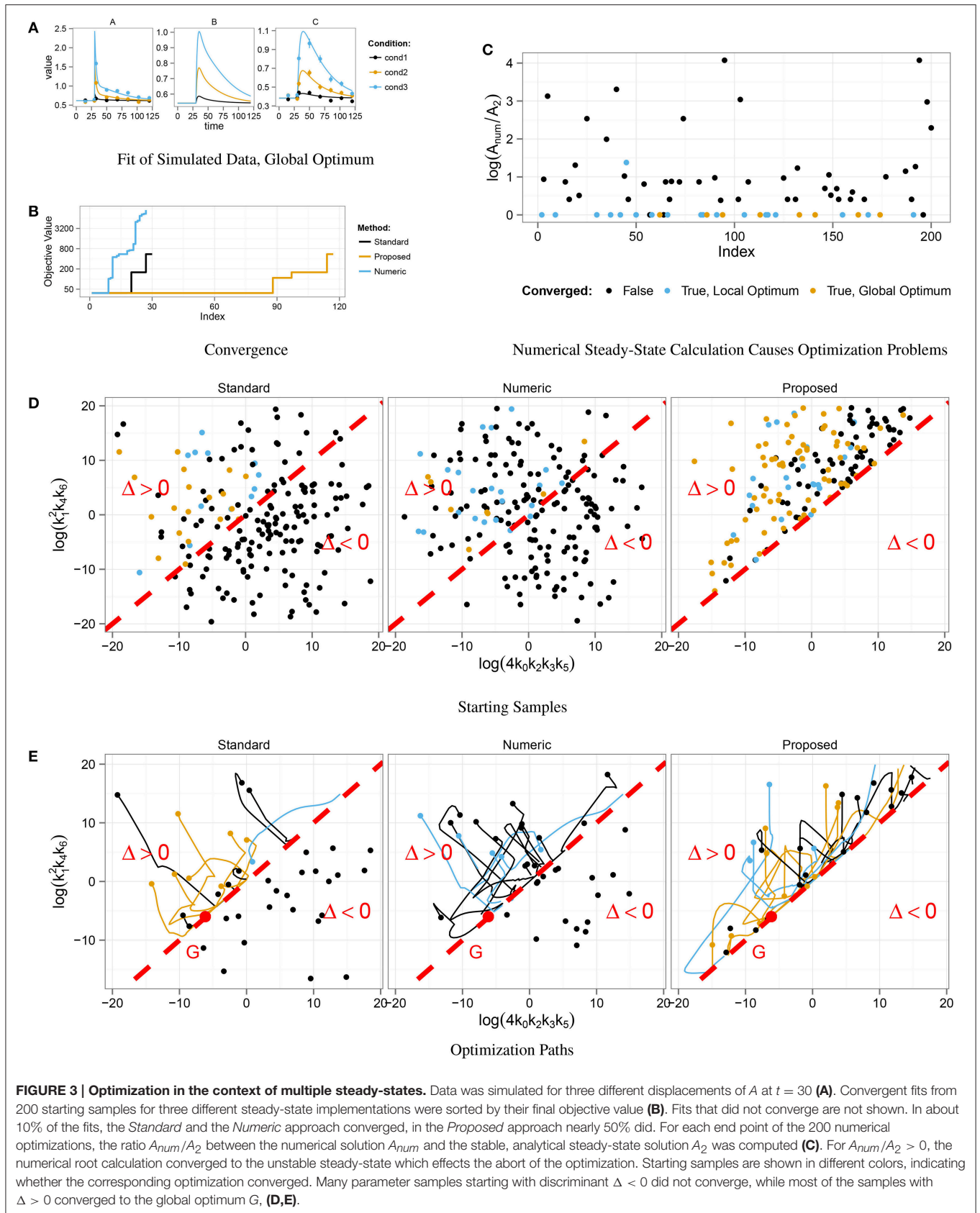


FIGURE 3 | Optimization in the context of multiple steady-states. Data was simulated for three different displacements of A at $t = 30$ (A). Convergent fits from 200 starting samples for three different steady-state implementations were sorted by their final objective value (B). Fits that did not converge are not shown. In about 10% of the fits, the *Standard* and the *Numeric* approach converged, in the *Proposed* approach nearly 50% did. For each end point of the 200 numerical optimizations, the ratio A_{num}/A_2 between the numerical solution A_{num} and the stable, analytical steady-state solution A_2 was computed (C). For $A_{num}/A_2 > 0$, the numerical root calculation converged to the unstable steady-state which effects the abort of the optimization. Starting samples are shown in different colors, indicating whether the corresponding optimization converged. Many parameter samples starting with discriminant $\Delta < 0$ did not converge, while most of the samples with $\Delta > 0$ converged to the global optimum G, (D,E).

path. The numerically calculated value A_{num} was taken from the root calculation by Newton's method and the value A_2 of the stable steady-state was calculated by means of Equation (19) with the corresponding parameter values. **Figure 3C** shows ratios A_{num}/A_2 for all fits. If $A_{num}/A_2 = 0$, the stable steady-state was obtained, while $A_{num}/A_2 > 0$ implies that the unstable solution was obtained. Since nearly all fits that reached the unstable solution did not converge, we conclude that the coexistence of a second unstable steady-state causes optimization aborts in the numerical approach of steady-state determination.

Both problems arising from the existence of multiple steady-states, i.e., negative discriminants and stable vs. unstable steady-states, are automatically circumvented by our steady-state algorithm resulting in a superior convergence rate during parameter estimation.

4. DISCUSSION AND CONCLUSION

Parameter estimation in non-linear ODE models of biological systems has to deal with several local optima and a high-dimensional parameter space. In order to reduce the number of parameters, steady-state constraints are taken into account. Deterministic algorithms search for the global optimum by performing the optimization with multiple starting samples. The way of implementing steady-states, i.e., the exact parameterization, has an impact on the convergence probability of a randomly chosen starting sample. If optimizations tend to abort before reaching an optimum, many starting samples are necessary to find the best possible fit. Since incorporation of steady-state information shifts parameter distributions and contributes to gradient information, the exact steady-state parametrization plays a crucial role in optimization.

For many systems, steady-state equations lead to higher-order polynomial equations when being solved for the state variables. To exploit the full steady-state information, equations can be partially solved for kinetic parameters. If the obtained steady-state expressions yield negative values for certain parameter specifications, those might lead to rapidly growing solutions for the ODE system. We showed that negative parameter values have a considerable, negative impact on the success of the optimization.

In many applications, multiplicity and multi-stability of the steady-state constitutes the relevant question. In the case of parameter estimation, however, multiple steady-states complicate the estimation process. For the standard approach of solving steady-state equations for the state variables, all solutions principally have to be considered and optimization has to be performed for all possibilities. For the numerical implementation of steady-states also unstable steady-state solutions constitute a problem, since the numerical root finding method might converge to the unstable solution. In our case, the convergence probability dropped by 80%.

In this work, we presented an algorithm that derives steady-state expressions and circumvents negative and multiple solutions by construction. The approach covers the most common classes of ODE models consisting of e.g., mass-action kinetics, inhibition terms, Michaelis-Menten or Hill-type equations. By means of graph theory, cyclic dependencies between dynamical variables, e.g., positive or negative feedbacks inside a signaling cascade that lead to polynomial equations of order two or higher are removed by solving for kinetic parameters for which the equations are linear. In order to guarantee positivity of all solutions, the algorithm performs appropriate parameter transformations replacing kinetic parameters by ratios of participating fluxes. Our approach experiences a major limitation if simultaneously, the size of the ODE model becomes large and combinations of several in- and outflux parameters contribute to multiple states. Then, the algorithm is not able to find a strictly positive solution for the system. Furthermore, since the algorithm solves for rate parameters, it might not be applicable if solving for parameters is not allowed due to other reasons, e.g., if rate parameters must take certain fixed values.

In summary, our approach enables steady-state calculation for models with many cyclic dependencies that lead to higher-order polynomial equations when solved for state variables. Multiplicity and multi-stability are avoided and positivity of the solution is guaranteed. The parameter space is reduced by the number of independent steady-state equations while the nice convergence behavior is preserved.

AUTHOR CONTRIBUTIONS

MR developed, implemented and tested the method. MR, JT, and DK together wrote the paper.

FUNDING

This work was supported by the Federal Ministry of Education and Research (BMBF), by the MIP-DILI project, Innovative Medicines Initiative Joint Undertaking under grant agreement No. 115336-2 and by the ReelinSys project, Systems biology of Reelin-associated neuropsychiatric disorders, under No. 0316174D.

ACKNOWLEDGMENTS

We thank all our colleagues who provided us with many test models that contributed to the development of the algorithm.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <http://journal.frontiersin.org/article/10.3389/fcell.2016.00041>

REFERENCES

- Chou, K.-C. (1990). Applications of graph theory to enzyme kinetics and protein folding kinetics. *Biophys. Chem.* 35, 1–24. doi: 10.1016/0301-4622(90)80056-D
- Feinberg, M. (1979). Complex balancing in general kinetic systems. *Arch. Rat. Mech. Anal.* 49, 187–194. doi: 10.1007/BF00255665
- Feliu, E., Knudsen, M., Andersen, L. N., and Wiuf, C. (2012). An algebraic approach to signaling cascades with n layers. *Bull. Math. Biol.* 74, 45–72. doi: 10.1007/s11538-011-9658-0
- Feliu, E., and Wiuf, C. (2012). Variable elimination in chemical reaction networks with mass-action kinetics. *SIAM J. Appl. Math.* 72, 959–981. doi: 10.1137/110847305
- Feliu, E., and Wiuf, C. (2013). Variable elimination in post-translational modification reaction networks with mass-action kinetics. *J. Math. Biol.* 66, 281–310. doi: 10.1007/s00285-012-0510-4
- Halász, A., Lai, H.-J., McCabe Pryor, M., Radhakrishnan, K., and Edwards, J. (2013). Analytical solution of steady-state equations for chemical reaction networks with bilinear rate laws. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 10, 957–969. doi: 10.1109/TCBB.2013.41
- Horn, F., and Jackson, R. (1972). General mass action kinetics. *Arch. Rat. Mech. Anal.* 47, 81–116. doi: 10.1007/BF00251225
- King, E. L., and Altman, C. (1956). A schematic method of deriving the rate laws for enzyme-catalyzed reactions. *J. Phys. Chem.* 60, 1375–1378. doi: 10.1021/j150544a010
- Loriaux, P. M., Tesler, G., and Hoffmann, A. (2013). Characterizing the relationship between steady state and response using analytical expressions for the steady states of mass action models. *PLoS Comput. Biol.* 9:e1002901. doi: 10.1371/journal.pcbi.1002901
- Peng, H., Li, L., Yang, Y., and Liu, F. (2010). Parameter estimation of dynamical systems via a chaotic ant swarm. *Phys. Rev. E* 81:016207. doi: 10.1103/PhysRevE.81.016207
- Raue, A., Schilling, M., Bachmann, J., Matteson, A., Schelker, M., Kaschek, D., et al. (2013). Lessons learned from quantitative dynamical modeling in systems biology. *PLoS ONE* 8:e74335. doi: 10.1371/journal.pone.0074335
- Xiang, Y., and Gong, X. G. (2000). Efficiency of generalized simulated annealing. *Phys. Rev. E* 62, 4473–4476. doi: 10.1103/PhysRevE.62.4473

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2016 Rosenblatt, Timmer and Kaschek. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.