



# Algorithmic Reduction of Biological Networks with Multiple Time Scales

Niclas Kruff · Christoph Lüders ·  
Ovidiu Radulescu · Thomas Sturm ·  
Sebastian Walcher

Received: 16 October 2020 / Revised: 2 March 2021 / Accepted: 16 March 2021 / Published online: 8 July 2021  
© The Author(s) 2021

**Abstract** We present a symbolic algorithmic approach that allows to compute invariant manifolds and corresponding reduced systems for differential equations modeling biological networks which comprise chemical reaction networks for cellular biochemistry, and compartmental models for pharmacology, epidemiology and ecology. Multiple time scales of a given network are obtained by scaling, based on tropical geometry. Our reduction is mathematically justified within a singular perturbation setting. The existence of invariant manifolds is subject to hyperbolicity conditions, for which we propose an algorithmic test based on Hurwitz criteria. We finally obtain a sequence of nested invariant manifolds and respective reduced systems on those manifolds. Our theoretical results are generally accompanied by rigorous algorithmic descriptions suitable for direct implementation based on existing off-the-shelf software systems, specifically symbolic computation libraries and Satisfiability Modulo Theories solvers. We present computational examples taken from the well-known BioModels database using our own prototypical implementations.

**Keywords** Chemical reaction network · Compartmental model · Dimension reduction · Invariant set · Logic computation · Multiple time scales · Polynomial differential equations · Real algebraic computation · Satisfiability modulo theories · Singular perturbation · Symbolic computation · Tropical geometry

**Mathematics Subject Classification** Primary 68W30 · Secondary 14P10 · 34E15 · 37D10 · 92C45

N. Kruff · S. Walcher  
RWTH Aachen University, Aachen, Germany  
e-mail: niclas.kruff@matha.rwth-aachen.de  
S. Walcher  
e-mail: walcher@matha.rwth-aachen.de

C. Lüders  
University of Bonn, Bonn, Germany  
e-mail: chris@cfos.de

O. Radulescu  
University of Montpellier and CNRS UMR5235 LPHI, Montpellier, France  
e-mail: ovidiu.radulescu@umontpellier.fr

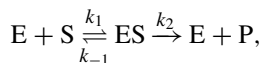
T. Sturm (✉)  
CNRS, Inria, and the University of Lorraine, Nancy, France  
e-mail: thomas.sturm@loria.fr

MPI Informatics and Saarland University, Saarbrücken, Germany  
e-mail: sturm@mpi-inf.mpg.de

## 1 Introduction

Biological network models describing elements in interaction are used in many areas of biology and medicine. Chemical reaction networks are used as models of cellular biochemistry, including gene regulatory networks, metabolic networks, and signaling networks. In epidemiology and ecology, compartmental models can be described as networks of interactions between compartments. Both in chemical reaction networks and in compartmental models, the probability that two elements interact is assumed proportional to their abundances. This property, called mass action law in biochemistry, leads to polynomial differential equations in the kinetics.

For differential equations that describe the development of such networks over time, a crucial question is concerned with reduction of dimension. We illustrate such a reduction and the steps involved for the classic Michaelis–Menten system, an archetype of enzymatic reactions. This system is described by the chemical reactions



where E, S, ES, and P are the enzyme, substrate, enzyme-substrate complex, and product, respectively. The mechanism has two conserved quantities  $[E] + [ES] = c_1$  and  $[S] + [ES] + [P] = c_2$ , where  $c_1$  and  $c_2$  are constant functions representing the respective total concentrations. Let us choose the concentration units such that  $c_2 = 1$ , and furthermore rename  $c_1 = \varepsilon$ . In the ordinary differential equations describing the kinetics of  $[E]$ ,  $[S]$ ,  $[ES]$ , and  $[P]$  according to [24, Sect. 2.1.2] we eliminate the variable  $[E] = \varepsilon - [ES]$ . This yields a reduced system made of variables  $y_1 = [S]$  and  $y_2 = [ES]$  as follows:

$$\begin{aligned} \dot{y}_1 &= -\varepsilon k_1 y_1 + (k_1 y_1 + k_{-1}) y_2 \\ \dot{y}_2 &= \varepsilon k_1 y_1 - (k_1 y_1 + k_{-1} + k_2) y_2. \end{aligned}$$

Notice that  $[\dot{P}]$  is already determined by the algebraic conservation constraint  $[P] = 1 - y_1 - y_2$ . The parameter  $\varepsilon$  represents the ratio of total concentrations  $c_1$  to  $c_2$ . The general idea is that  $\varepsilon$  is small.

In a first step toward reduction, a scaling transformation  $y_1 = x_1$  and  $y_2 = \varepsilon x_2$  yields

$$\begin{aligned} \dot{x}_1 &= \varepsilon(-k_1 x_1 + (k_1 x_1 + k_{-1}) x_2) \\ \dot{x}_2 &= k_1 x_1 - (k_1 x_1 + k_{-1} + k_2) x_2. \end{aligned}$$

In a second step, one uses singular perturbation theory to obtain the famous Michaelis–Menten equation. It consists of two components: First, we obtain a one dimensional invariant manifold given approximately by the quasi-steady state condition  $k_1 x_1 - (k_1 x_1 + k_{-1} + k_2) x_2 = 0$ . This considers the fast variable  $x_2$  to be at the steady state and lowers dimension from two to one. Second, we obtain a reduced system for the slow variable:

$$\dot{x}_1 = -\varepsilon \frac{k_1 k_2 x_1}{k_1 x_1 + k_{-1} + k_2}.$$

With our example, we paraphrased the approach in a seminal paper by Heineken et al. [34], which was the first one to rigorously discuss quasi-steady state from the perspective of singular perturbation theory. Realistic network models may have many species and differential equations. Considerable effort has been put into model order reduction, i.e., finding approximate models with a smaller number of species and equations, where the reduced model can be more easily analyzed than the full model [58].

The scaling of parameters and variables by a small parameter  $\varepsilon$  and the study of the limit  $\varepsilon \rightarrow 0$  is central in singular perturbation theory. It is rather obvious that arbitrary scaling transformations are unlikely to provide useful information about a given system. Successful scalings, in contrast, are typically related to the existence of nontrivial invariant manifolds. Applications of scaling rely on the observation that, loosely speaking, any result that holds asymptotically for  $\varepsilon \rightarrow 0$  remains valid for sufficiently small positive  $\varepsilon_*$ , provided some technical conditions are satisfied. To determine scalings of polynomial or rational vector fields that model biological networks, tropical

equilibration methods were introduced and developed in a series of papers by Noel et al. [55], Radulescu et al. [59], Samal et al. [62,63], and others. These methods open a feasible path for biological networks of high dimension. For a given system they provide a list of possible slow-fast systems, which may or may not yield invariant manifolds and reduced equations. Other methods due to Goeke et al. [31], and recently extended to multiple time scales by Kruff and Walcher [40], determine critical parameter values and manifolds for singular perturbation reductions.

The principal purpose of the present paper is to complement scaling with an algorithmic test for the existence of invariant manifolds and the computation of those manifolds along with corresponding reduced systems of differential equations. In the asymptotic limit, methods from singular perturbation theory, principally developed by Tikhonov [74] and Fenichel [25], are available. A recent extension to multiscale systems by Cardin and Teixeira [11] turns out to be a valuable tool for the systematic computation of reductions with nested invariant manifolds and allows an algorithmic approach.

In the language of dynamical systems, the behavior of systems with multiple time scales can be described as follows. All variables evolve towards the steady state of the ordinary differential equation that they follow, but not with the same speed, in other words, not within the same *time scale*. *Slow variables* correspond to long time scales, and *fast variables* correspond to short time scales. The steady state of a subset of the ordinary differential equations is called *quasi-steady state*, and the evolution of a variable or group of variables towards its quasi-steady state is called *relaxation*. At a given time scale, a group of variables relaxes towards one of their quasi-steady states. The set of all quasi-steady states of variables relaxing within the same time scale forms a *critical manifold*, which provides the lowest order approximation to the corresponding *invariant manifold*. In this article, we do not distinguish between critical and invariant manifolds, since we do not discuss higher order approximations. All slower variables can be considered fixed, and all faster variables have already relaxed and satisfy quasi-steady state conditions. As the number of relaxed variables increases and thus the set of quasi-steady state conditions grows, the respective invariant manifolds get nested so that later manifolds are contained in earlier ones. Local linear approximations of these manifolds were proposed by Valorani and Paolucci [75] using numerical methods based on the local Jacobian. However, to the best of our knowledge, constructive approaches providing nonlinear descriptions of these manifolds and reduced models are still missing.

From a computer science point of view, we propose a novel symbolic computation-based algorithmic workflow for the reduction process outlined above. This includes in particular the automatic verification of certain hyperbolicity conditions required for the validity of the reductions. We restrict ourselves to the case of polynomial differential equations that covers mass action chemical reaction networks and compartmental models. We present a series of algorithms that takes as input a system of polynomial autonomous ordinary differential equations together with numerical information related to the desired coarse graining of the scaling. As output one finally obtains a collection of nested invariant manifolds for the input system, associated with smaller dimensional systems that govern the dynamics on those manifolds. This output establishes the reduced systems discussed above.

The computationally hard parts of our methods are reduced to decision problems in interpreted first-order logic over various theories. It turns out that quantifier alternation can be entirely avoided, so that the Satisfiability Modulo Theories (SMT) framework by Nieuwenhuis et al. [52] can be applied. Several corresponding SMT solvers are freely available and professionally supported [1,13,17,51]. It is remarkable that we arrive with our comprehensive algorithmic work here at SMT sub-problems for several different logics, viz. linear integer arithmetic, linear real arithmetic, and non-linear real arithmetic. The algorithms presented here are suitable for straightforward implementation provided that a symbolic computation library, or computer algebra system, and an SMT solver are available. We created two independent prototypical realizations in software ourselves, one in Python using freely available libraries, and one in Maple.

The plan of the paper is as follows: In Sect. 2.1 we introduce an abstract scaling procedure, which assumes, for given  $0 < \varepsilon_* < 1$ , the existence of families of exponents  $c_{k,j}$  and  $d_k$  for scaling polynomial coefficients and variables, respectively. From the scaled system, higher order terms are truncated, and the obtained system is partitioned into several time scales, ordered from fastest to slowest. A corresponding generic algorithm uses black-box functions  $c$  and  $d$ . In Sect. 2.2, we make precise one possible way to realize  $c$  and  $d$ , based on tropical geometry. So far, our transformations are mostly of formal nature. On these grounds, we algorithmically determine in Sect. 3 invariant

manifolds and corresponding reduced systems, which makes the formal scaling meaningful in a mathematically precise way. In general, this is possible only for a certain number  $\ell$  of time scales, where  $\ell$  is explicitly found and—in contrast to existing alternative approaches—often larger than 2. Technically, we apply recent results by Cardin and Teixeira [11] based on Fenichel theory. In Sect. 4, we employ symbolic computation techniques, specifically Gröbner basis theory, to equivalently simplify our reduced systems, which are still scaled in terms of  $\varepsilon_*$ ,  $c$ , and  $d$ . In Sect. 5, we finally transform back to the principal scale of the original system while preserving the obtained multiple time scales and the structure of the corresponding reduced systems. In particular, the various time scale factors remain explicit. In Sect. 6, we summarize what we have gained from the overall procedure for our original input. At this point, the mathematical development of our framework has been accompanied by nine algorithms, and we give a tenth top-level algorithm, which makes precise how various modules are combined and interact with one another. In Sect. 7 we discuss computational examples with our prototypical software mentioned above. We consider models from the BioModels database, a repository of mathematical models of biological processes [57]. The focus is on successful reductions for biologically interesting examples. This is counterbalanced in Appendix A by further examples to support the understanding of our algorithms. This also provides some examples where we do not obtain meaningful reductions. In Sect. 8 we highlight some computational steps in our algorithms from the point of view of asymptotic worst-case complexity. In Sect. 9, we wrap up and point at possible future research directions.

## 2 Scaling of Polynomial Vector Fields

In what follows, we adopt a rather general scaling formalism that has been used recently in [54, 55, 58, 59, 63] and is recurrent in the literature on singular perturbations, see for instance [53, Sect. 3]. We use the convention that the natural numbers  $\mathbb{N}$  include 0.

### 2.1 An Abstract Scaling Procedure

Our starting point is a parameter dependent system  $S$  of polynomial differential equations

$$\dot{y}_k := \frac{dy_k}{dt} = \sum_J \gamma_{k,J} y^J, \quad 1 \leq k \leq n, \quad (1)$$

where the summation ranges over multi-indices  $J = (j_1, \dots, j_n) \in \mathbb{N}^n$ ,  $\gamma_{k,J} \in \mathbb{R}$ , and only finitely many  $\gamma_{k,J}$  are non-zero. We abbreviate  $y^J = y_1^{j_1} \dots y_n^{j_n}$ , as usual. In terms of network models,  $y_k$  represents the concentration of either a chemical species or a type of individual in a compartment. Note that we use positive integers as indices, instead of concrete names for species and compartments. The real coefficients  $\gamma_{k,J}$  describe actions of other species or individuals on the species or individual  $k$ . If these actions are activations one has  $\gamma_{k,J} > 0$ , whereas for repressions one has  $\gamma_{k,J} < 0$ . Several species may interact to produce an action on a given species  $k$ . This information is contained in the number of non-zero components of  $J$ , which is called the order of the action. This terminology is inspired from chemical reactions, where the order represents essentially the number of reactant species.

Throughout this paper, we require that positive  $y_k$  remain positive as time progresses. In other words, the positive first orthant  $\mathcal{U} = (0, \infty)^n \subseteq \mathbb{R}^n$  is positively invariant for system (1), which is the case, e.g., in chemical reaction networks when  $\gamma_{k,J} y^J \geq 0$  on all intersections of hyperplanes  $\{(y_1, \dots, y_n) \in \mathbb{R}^n \mid y_k = 0\}$  with  $\bar{\mathcal{U}}$ .

We fix some small  $\varepsilon_* \in (0, 1)$ , and we impose that

$$\gamma_{k,J} = \varepsilon_*^{c_{k,J}} \bar{\gamma}_{k,J}, \quad (2)$$

with rational numbers  $c_{k,J}$ . The tacit understanding is that only nonzero  $\gamma_{k,J}$  are being considered. The intuitive idea, which will be made more precise in Sect. 2.2, is that the  $\bar{\gamma}_{k,J}$  are close to one. Moreover, we introduce a positive parameter  $\varepsilon$  and consider the system

$$\dot{y}_k = \sum_J \varepsilon^{c_{k,J}} \bar{\gamma}_{k,J} y^J, \quad 1 \leq k \leq n, \quad (3)$$

with  $\varepsilon$ -dependent coefficients. Notice that (3) matches (1) at  $\varepsilon = \varepsilon_*$ . By renormalizing  $y_k = \varepsilon^{d_k} x_k$ ,  $d_k \in \mathbb{Q}$ , one obtains a system in scaled variables

$$\dot{x}_k = \sum_J \varepsilon^{c_{k,J} + \langle D, J \rangle - d_k} \bar{\gamma}_{k,J} x^J, \quad 1 \leq k \leq n, \quad (4)$$

with  $D = (d_1, \dots, d_n)$  and the dot product in  $\mathbb{R}^n$  denoted by  $\langle \cdot, \cdot \rangle$ . This transformation preserves the positive invariance of  $\mathcal{U}$ . The scaling comes with the implicit assumption that for  $i, j \in \{1, \dots, n\}$ , the relative order of  $y_i$  with respect to  $y_j$  is bounded by  $y_i/y_j = \Theta(\varepsilon^{d_i - d_j})$  for  $\varepsilon \rightarrow 0$ , so that all  $x_k$  get the same order of magnitude. Continuing, we set  $v_k = \min\{c_{k,J} + \langle D, J \rangle - d_k \mid \bar{\gamma}_{k,J} \neq 0\}$  to obtain

$$\dot{x}_k = \varepsilon^{v_k} \sum_J \varepsilon^{c_{k,J} + \langle D, J \rangle - d_k - v_k} \bar{\gamma}_{k,J} x^J, \quad 1 \leq k \leq n, \quad (5)$$

where now all exponents of  $\varepsilon$  inside the sums are nonnegative. Finally one may perform a preliminary time scaling  $\tau = \varepsilon^\mu t$ ,  $\mu = \min\{v_1, \dots, v_n\}$  to arrive at

$$x'_k := \frac{dx_k}{d\tau} = \varepsilon^{v_k - \mu} \sum_J \varepsilon^{c_{k,J} + \langle D, J \rangle - d_k - v_k} \bar{\gamma}_{k,J} x^J, \quad 1 \leq k \leq n, \quad (6)$$

with all exponents nonnegative. We are interested in system (6) for variable  $\varepsilon > 0$ , in the asymptotic limit  $\varepsilon \rightarrow 0$ .

We restructure (6) by collecting all variables with equal  $v_i - \mu$  in vectors  $z_1, \dots, z_m$ , where  $z_k \in \mathbb{R}^{n_k}$  for  $k \in \{1, \dots, m\}$ , in ascending order of exponents and such that  $n_1 + \dots + n_m = n$ . We obtain a system of the form

$$z'_k = \varepsilon^{a_k} \tilde{f}_k(z, \varepsilon) = \varepsilon^{a_k} \left( \tilde{f}_k(z, 0) + \varepsilon^{a'_{k,2}} p_{k,2} + \dots + \varepsilon^{a'_{k,w_k}} p_{k,w_k} \right) = \varepsilon^{a_k} \left( \tilde{f}_k(z, 0) + o(1) \right), \quad 1 \leq k \leq m, \quad (7)$$

where  $a_k, a'_{k,j} \in \mathbb{Q}$ ,  $0 = a_1 < a_2 < \dots < a_m$ ,  $0 < a'_{k,j}$ , and  $p_{k,j}$  are multivariate polynomials in  $z$  for  $1 \leq k \leq m$  and  $2 \leq j \leq w_k$ . Note that the case  $m = 1$  is not excluded. By substituting  $\delta := \varepsilon^{1/q}$  with a sufficiently large positive integer  $q$ , one ensures that only nonnegative integer powers of  $\delta$  appear:

$$z'_k = \delta^{b_k} \hat{f}_k(z, \delta) = \delta^{b_k} \left( \hat{f}_k(z, 0) + \delta^{b'_{k,2}} p_{k,2} + \dots + \delta^{b'_{k,w_k}} p_{k,w_k} \right) = \delta^{b_k} \left( \hat{f}_k(z, 0) + o(1) \right), \quad 1 \leq k \leq m, \quad (8)$$

where  $b_k, b'_{k,j} \in \mathbb{N}$ ,  $0 = b_1 < b_2 < \dots < b_m$ ,  $0 < b'_{k,j}$  for  $1 \leq k \leq m$  and  $2 \leq j \leq w_k$ .

Our idea is that the indices  $k$  correspond to different time scales  $\delta^{b_k} \tau$ . For  $m > 1$ , system (8), as  $\delta \rightarrow 0$ , may be thought of as separating fast variables from increasingly slow ones. It will turn out in Sect. 3 that the exact number of time scales finally obtained by our overall approach can actually be smaller than  $m$ .

Given certain conditions, which will be made explicit in Theorem 1 and with its application in Sect. 3.1, we may formally truncate the right hand sides of (8) and keep only terms of lowest order in  $\delta$ :

$$z'_k = \delta^{b_k} \hat{f}_k(z, 0), \quad 1 \leq k \leq m. \quad (9)$$

In the following, we refer to the transformation process from (1) to (8) as *scaling*. Strictly speaking, this comprises scaling in combination with *partitioning*. We refer to the step from (8) to (9) as *truncating*.

Algorithm 1 reflects our discussions so far. It takes as input a list  $S$  of differential equations representing system (1)

---

### Algorithm 1 ScaleAndTruncate

---

**Input:** 1. A list  $S = [\frac{dy_1}{dt} = f_1, \dots, \frac{dy_n}{dt} = f_n]$  of autonomous first-order ordinary differential equations where  $f_1, \dots, f_n \in \mathbb{Q}[y_1, \dots, y_n]$ ;  
 2.  $c : \{1, \dots, n\} \times \{1, \dots, n\}^n \rightarrow \mathbb{Q}$ ;  
 3.  $d : () \rightarrow \mathbb{Q}^n \cup \{\perp\}$ ;  
 4.  $\varepsilon_* \in (0, 1) \cap \mathbb{Q}$

**Output:** 1. A list  $[T_1, \dots, T_m]$  where, abbreviating  $\frac{d}{d\tau}$  by a prime,  $T_k = (z'_k = \delta^{b_k} f'_k)$  with  $z'_k \subseteq [x'_1, \dots, x'_n]$ ,  $\bigcup_k z'_k = [x'_1, \dots, x'_n]$ ,  $z'_1, \dots, z'_m$  pairwise disjoint,  $b_1 < \dots < b_m \in \mathbb{N}$ , and  $f'_k \subseteq \mathbb{Q}[x_1, \dots, x_n]$ , or the empty list;  
 2. A list  $[P_1, \dots, P_m]$  of lists with  $P_k \subseteq \mathbb{Q}[x_1, \dots, x_n][\delta]$  and  $|P_k| = |T_k|$  for  $k \in \{1, \dots, m\}$ ;  
 3. A substitution  $\sigma$  for  $x_1, \dots, x_n, \tau, \delta$ , and  $\varepsilon$

The first output  $[T_1, \dots, T_m]$  contains differential equations  $z'_k = \delta^{b_k} \widehat{f}'_k(z, 0)$  for  $k \in \{1, \dots, m\}$  in terms of system (8). The second output  $[P_1, \dots, P_m]$  contains the higher order terms in (8) as polynomials  $p_k = \delta^{b_k+b'_{k,2}} p_{k,2} + \dots + \delta^{b_k+b'_{k,w_k}} p_{k,w_k}$ . The last output is a substitution that undoes all substitutions applied for obtaining (8) from (1).

This gives the following invariant: Denote  $\widetilde{S} = (\bigcup_{k=1}^m T_k \oplus P_k)\sigma$ , where  $(x' = g) \oplus p$  stands for  $x' = g + p$  and is applied elementwise. Then  $\widetilde{S}$  is equal to  $S$  up to multiplication of the differential equation  $\dot{y}_i = \sum_J \gamma_{i,J} y^J$  in  $S$  with a positive scalar factor  $1/\varepsilon_*^{\mu+d_i}$ .

For  $q \in \mathbb{Q}[x_1, \dots, x_n](\delta)$  we use  $\deg_\delta(q)$  for the univariate degree of  $q$  in  $\delta$ . Similarly,  $\text{tmon}_\delta(q)$  is the trailing monomial in  $\delta$ .

```

1: if  $d() = \perp$  then
2:   return [], [], []
3: end if
4:  $\mu := \infty$ 
5:  $q := 1$ 
6:  $(d_1, \dots, d_n) := d()$   $\in \mathbb{Q}^n$ 
7: for  $k := 1$  to  $n$  do
8:    $h_k := 0$ 
9:   for all monomials  $\gamma y^J$  in  $f_k$  do
10:     $\bar{\gamma} := \gamma / \varepsilon_*^{c(k,J)}$   $\in \overline{\mathbb{Q}}$ 
11:     $\eta := c(k, J) + \langle (d_1, \dots, d_n), J \rangle - d_k$   $\in \mathbb{Q}$ 
12:     $\mu := \min(\mu, \eta)$   $\in \mathbb{Q}$ 
13:     $q := \text{lcm}(q, \text{denom } \eta)$   $\in \mathbb{N} \setminus \{0\}$ 
14:     $h_k := h_k + \varepsilon^\eta \bar{\gamma} x^J$ 
15:   end for
16: end for
17: for  $k := 1$  to  $n$  do
18:    $h_k := h_k / \varepsilon^\mu$ 
19:    $h_k := h_k[\varepsilon \leftarrow \delta^q]$   $\in \overline{\mathbb{Q}}[x_1, \dots, x_n][\delta]$ 
20:    $g_k := \text{tmon}_\delta h_k$ 
21:    $p_k := h_k - g_k$ 
22: end for
23:  $L := [\frac{dx_1}{d\tau} = g_1, \dots, \frac{dx_n}{d\tau} = g_n]$ 
24:  $[b_1, \dots, b_m] := \text{sort}(\deg_\delta g_1, \dots, \deg_\delta g_n)$ , ascending and removing duplicates
25: for  $k := 1$  to  $m$  do
26:    $T_k := [\frac{dx}{d\tau} = g \in L \mid \deg_\delta g = b_k]$ 
27:    $P_k := [p_j \in \{p_1, \dots, p_n\} \mid \deg_\delta g_j = b_k]$ 
28: end for
29:  $\sigma := [x_1 \leftarrow y_1 / \varepsilon^{d_1}, \dots, x_n \leftarrow y_n / \varepsilon^{d_n}] \circ [\tau \leftarrow \varepsilon^\mu t] \circ [\delta \leftarrow \varepsilon^{1/q}] \circ [\varepsilon \leftarrow \varepsilon_*]$ 
30: return  $[T_1, \dots, T_m], [P_1, \dots, P_m], \sigma$ 

```

---

and a choice of  $0 < \varepsilon_* < 1$  for (2). For our practical purposes, the polynomial coefficients in  $S$  as well as  $\varepsilon_*$  are taken from  $\mathbb{Q}$ . Our algorithm is furthermore parameterized with a function  $c$  mapping suitable indices to rational

numbers and a constant function  $d$  yielding either a tuple  $D = (d_1, \dots, d_n)$  of rational numbers or  $\perp$ . The black-box functions  $c$  and  $d$  reflect the mathematical assumptions around (2) and (4) that suitable  $c_{k,J}$  and  $d_k$  exist, respectively. Suitable instantiations for the parameters  $c$  and  $d$  can be realized, e.g., using tropical geometry, which will be the topic of the next section. It will turn out that instantiations of  $d$  can fail on the given combination of  $S$  and  $\varepsilon_*$ , which is signaled by the return value  $\perp$  of  $d$ , and checked right away in l.1 of Algorithm 1.

## 2.2 Scaling via Tropical Geometry

So far, the above transformations leading to (4) are a formal exercise. No particular strategy was applied for choosing  $\varepsilon_* \in (0, 1)$ . Early model reduction studies used dimensional analysis to obtain  $\varepsilon_*$  as a power product in model parameters [34,66].

Here we discuss a different approach, based on tropical geometry, also called max-plus or idempotent algebra. This is a relatively recent field of mathematics that draws its origins from fields as diverse as algebraic geometry, optimization, and physics [46]. In all these fields, tropical geometry appears as a technique to simplify non-linear objects. Polynomials are replaced by piecewise-linear functions, and geometric problems are transformed into combinatorial problems [50].

Tropical geometry is natural for any computation with orders of magnitude. In physics, it occurs as Litvinov–Maslov dequantization of real numbers leading to degeneration of complex algebraic varieties into tropical varieties [45,77]. The name dequantization originates from the formal analogy between this limiting process and Schrödinger’s dequantization that turns quantum physics into classical physics when the Planck constant is considered a small parameter  $\varepsilon$  that tends to zero. Closer, in the physical world, to Litvinov–Maslov dequantization are the vanishing viscosity phenomena, well known in problems of wave propagation. In mathematics, tropical varieties and prevarieties establish a modern tool in the theory of Puiseux series [4].

In contrast to the dimensional analysis approach mentioned above, the value  $\varepsilon_*$  is now not dictated by physico-chemistry. Instead, it is freely chosen to provide “power” parametric descriptions of all the quantities occurring in the differential equations (parameters, monomials, time scales), in a similar way to describing curves by continuously varying real parameters.

Next, we explain how to obtain the orders  $c_{k,J}$  and  $D$  introduced in the previous section with (2) and (4), respectively. The orders  $c_{k,J}$  are computed from  $\varepsilon_* \in (0, 1)$  and  $\gamma_{k,J}$  as

$$c_{k,J} = \frac{\text{round}(p \log_{\varepsilon_*} |\gamma_{k,J}|)}{p}. \quad (10)$$

The function  $\text{round} : \mathbb{R} \rightarrow \mathbb{Z}$  rounds to nearest, ties to even, in the sense of IEEE 754 [37]. The positive integer  $p$  controls the precision of the rounding step. Using  $\tilde{\gamma}_{k,J} = \gamma_{k,J}/\varepsilon_*^{c_{k,J}}$  as defined in (2), our definition satisfies the constraint  $\varepsilon_*^{1/(2p)} \leq |\tilde{\gamma}_{k,J}| \leq \varepsilon_*^{-1/(2p)}$ . The orders  $D = (d_1, \dots, d_n)$  satisfy certain constraints as well. Those constraints result heuristically from the idea of compensation of dominant monomials [54]. Slow dynamics is possible if for each dominant, i.e., much larger than the other, monomial on the right hand side of (6), there is at least one other monomial of the same order but with opposite sign. This condition, named tropical equilibration condition [54,55,58,59,62,63], reads

$$\min_{\gamma_{k,J} > 0} (c_{k,J} + \langle D, J \rangle) = \min_{\gamma_{k,J'} < 0} (c_{k,J'} + \langle D, J' \rangle). \quad (11)$$

On these grounds, given system (1), the choice of  $\varepsilon_*$  boils down to defining orders of magnitude. Model parameters are coarse-grained and transformed to orders of magnitude in order to apply tropical scaling. The result depends on which parameters are close and which are very different as dictated by the coarse-graining procedure, i.e., by the choice of  $\varepsilon_*$ . Decreasing  $\varepsilon_*$  destroys details, and parameters tend to have the same order of magnitude. Increasing  $\varepsilon_*$  refines details, and parameters range over several orders of magnitude. For instance, using (10) and  $p = 1$ , parameters  $k_1 = 0.1$  and  $k_2 = 0.01$  have orders  $c_1 = 1$  and  $c_2 = 2$  for  $\varepsilon_* = 1/10$  but  $c_1 = c_2 = 1$  for  $\varepsilon_* = 1/50$ . This is the perspective taken in [54,55,63].

On the one hand, we have just seen that smaller choices of  $\varepsilon_*$  possibly hide details. On the other hand, in the following section we are going to review singular perturbation methods, which provide asymptotic results as a small parameter  $\delta$  approaches zero. Following the construction in Sect. 2.1, small choices of  $\varepsilon_*$  lead to small  $\delta$ , which gives a heuristic argument for choosing  $\varepsilon_*$  rather small. Thus in practice one has to reconcile two competing requirements, which unfortunately, still requires some human intuition.

We are now ready to instantiate the black-box functions  $c$  and  $d$  in our generic Algorithm 1 with tropical versions as given in Algorithm 2 and Algorithm 3, respectively. Algorithm 2 explicitly uses, besides the parameters  $k$  and  $J$  specified for  $c$  in Algorithm 1, also the right hand sides of the input system (1) and the choice of  $\varepsilon_*$ . As yet another parameter it takes the desired precision  $p$  for rounding in (10). Notice that the use of this extra information is compatible with the abstract scaling procedure in Sect. 2.1. Currying [18] allows to use Algorithm 2 in place of  $c$  in a formally clean manner.

---

### Algorithm 2 TropicalC

---

**Input:** 1.  $k \in \{1, \dots, n\}$ ;  
 2.  $J \in \{1, \dots, n\}^n$ ;  
 3. A list  $S = [\dot{y}_1 = f_1, \dots, \dot{y}_n = f_n]$  of autonomous first-order ordinary differential equations where  $f_1, \dots, f_n \in \mathbb{Q}[y_1, \dots, y_n]$ ;  
 4.  $\varepsilon_* \in (0, 1) \cap \mathbb{Q}$ .  
 5.  $p \in \mathbb{N} \setminus \{0\}$

**Output:**  $c \in \mathbb{Q}$

1:  $\gamma := \text{coeff}(f_k, y^J)$   $\in \mathbb{Q}$   
 2:  $c := \text{round}(p \log_{\varepsilon_*} |\gamma|) / p$   $\in \mathbb{Q}$   
 3: **return**  $c$

---



---

### Algorithm 3 TropicalD

---

**Input:** 1. A list  $S = [\dot{y}_1 = f_1, \dots, \dot{y}_n = f_n]$  of autonomous first-order ordinary differential equations where  $f_1, \dots, f_n \in \mathbb{Q}[y_1, \dots, y_n]$ ;  
 2.  $\varepsilon_* \in (0, 1) \cap \mathbb{Q}$ .  
 3.  $p \in \mathbb{N} \setminus \{0\}$

**Output:**  $(d_1, \dots, d_n) \in \mathbb{Q}^n \cup \{\perp\}$

1:  $\Pi(a_1, \dots, a_n) := \text{TropicalEquilibration}(S, \varepsilon_*, p)$   
 2: **if not**  $\mathbb{R} \models \exists a_1 \dots \exists a_n \Pi$  **then**  
 3:   **return**  $\perp$   
 4: **end if**  
 5:  $(d_1, \dots, d_n) :=$  one possible choice for  $a_1, \dots, a_n$   
 6: **return**  $(d_1, \dots, d_n)$

---

Similarly, Algorithm 3 takes parameters  $\varepsilon_*$  and  $p$ , while  $d$  is specified in Algorithm 1 to have no parameters at all. In l.1 we use Algorithm 4 as a subalgorithm for tropical equilibration. One obtains a disjunctive normal form  $\Pi$ , which explicitly describes a set  $\mathcal{P} = \{p \in \mathbb{Q}^n \mid \Pi(p)\}$  as a finite union of convex polyhedra, as known from tropical geometry. Every  $(d_1, \dots, d_n) \in \mathcal{P}$  satisfies (11). The satisfiability condition in l.2 tests whether  $\mathcal{P} \neq \emptyset$ . We



**Algorithm 4** TropicalEquilibration

**Input:** 1. A list  $S = [y_1 = f_1, \dots, y_n = f_n]$  of autonomous first-order ordinary differential equations where  $f_1, \dots, f_n \in \mathbb{Q}[y_1, \dots, y_n]$ ;  
 2.  $\varepsilon_* \in (0, 1) \cap \mathbb{Q}$ ;  
 3.  $p \in \mathbb{N} \setminus \{0\}$ .

**Output:** A formula  $\Pi(a_1, \dots, a_n)$  describing a finite union of convex polyhedra in  $\mathbb{R}^n$ .

```

1:  $A_0 := (1, a_1, \dots, a_n)$   $\in \mathbb{Q}[a_1, \dots, a_n]^{n+1}$ 
2: for  $j := 1$  to  $n$  do
3:    $c := 0$ 
4:   for all monomials  $\gamma y_1^{\alpha_1} \dots y_n^{\alpha_n}$  in  $f_j$  do  $\in \mathbb{Q}$ 
5:      $\alpha_0 := \text{round}(p \log_{\varepsilon_*} |\gamma|) / p$   $\in \mathbb{Q}$ 
6:      $c := c + 1$ 
7:      $\Sigma_c := \text{sgn } \gamma$   $\in \{-1, 1\}$ 
8:      $A_c := (\alpha_0, \alpha_1, \dots, \alpha_n)$   $\in \mathbb{Q} \times \mathbb{Z}^n \subseteq \mathbb{Q}^{n+1}$ 
9:   end for
10:   $B_j := \emptyset$ 
11:  for  $k := 1$  to  $c$  do
12:    for  $\ell := k + 1$  to  $c$  do
13:      if  $\Sigma_k \Sigma_\ell < 0$  then
14:         $P := \{(A_k - A_\ell, A_0) = 0\}$   $(A_k - A_\ell, A_0) \in \mathbb{Q}[a_1, \dots, a_n]$ 
15:        for  $m := 1$  to  $c$  do
16:           $P := P \cup \{(A_m - A_k, A_0) \geq 0\}$   $(A_m - A_k, A_0) \in \mathbb{Q}[a_1, \dots, a_n]$ 
17:        end for
18:         $B_j := B_j \cup \{P\}$  set of sets of constraints
19:      end if
20:    end for
21:  end for
22: end for
23:  $\Pi := \text{DisjunctiveNormalForm}(\bigwedge_{j=1}^n \bigvee_{P \in B_j} \bigwedge P)$ 
24: return  $\Pi$ 

```

employ *Satisfiability Modulo Theories (SMT)* solving [52] using the logic  $\text{QF\_LRA}$  [2] for quantifier-free linear real arithmetic. The set  $\mathcal{P}$  can get empty, e.g. when all monomials on the right hand side of some differential equation have the same sign. Such an exceptional situation is signaled with a return value  $\perp$  in l.3. In the regular case  $\mathcal{P} \neq \emptyset$ , the choice  $(d_1, \dots, d_n)$  in l.5 is provided by the SMT solver. From a practical point of view, the disjunctive normal form computation in Algorithm 4 is a possible bottleneck and requires good heuristic strategies [48].

With applications in the natural sciences one often wants to make in l.5 an adequate choice for  $(d_1, \dots, d_n)$  lying in a specific convex polyhedron  $P \subseteq \mathcal{P}$ , which technically corresponds to one conjunction in  $\Pi$ . Such choices are subtle and typically require human interaction. For instance, when the chain of reduced dynamical systems ends with a steady state, it is interesting to consider the polyhedron  $P$  that is closest to that steady state. Such strategies are not covered by our algorithms presented here.

At this stage we have obtained a scaled system as defined in Sect. 2.1, including partitioning. The focus of the next section is to utilize this scaling for analytically substantiated reductions.

### 3 Singular Perturbation Methods

The theory of singular perturbations is used to compute and justify theoretically the limit of system (8) when  $\delta \rightarrow 0$ . There are several types of results in this theory. The results of Tikhonov, further improved by Hoppensteadt, show the convergence of the solution of system (8) to the solution of a differential-algebraic system in which the slowest variables  $z_m$  follow differential equations, and the remaining fast variables follow algebraic equations [35, 74]. The results of Fenichel are known under the name of *geometric singular perturbations*. He showed that the algebraic equations in Tikhonov's theory define a slow invariant manifold that is persistent for  $\delta > 0$  [25]. For geometric singular perturbations, differentiability in  $\delta$  is needed in system (8).

Samal et al. have noted that Tikhonov's theorem is applicable to tropically scaled systems [63]. For instance, with  $\delta_1 = \delta^{b_2}$ , system (8) may be rewritten as

$$z'_1 = \widehat{g}_1(z, \delta_1), \quad z'_2 = \delta_1 \widehat{g}_2(z, \delta_1), \quad \dots, \quad z'_m = \delta_1 \widehat{g}_m(z, \delta_1). \quad (12)$$

However, this approach comes with certain limitations. To start with, it allows only two time scales. Furthermore, in case  $b_2 > 1$ , there may be differentiability issues with respect to  $\delta_1$ , and some care has to be taken when one tries to apply to (12) also Fenichel's results [25]. In this section, we are going to generalize geometric singular perturbations, and compute invariant manifolds and reduced models for more than two time scales, introducing further  $\delta_2, \dots, \delta_{\ell-1}$ . Our generalization is based on a recent paper by Cardin and Teixeira [11].

Section 3.1 presents relevant results from [11] adapted to our purposes here and applied to our system (8). In contrast to the original article, which is based on a series of hyperbolicity conditions, we introduce the notion of hyperbolic attractivity, which is stronger but still adequate for our purposes. In Sect. 3.2 we describe efficient algorithmic tests for hyperbolic attractivity. Section 3.3 gives sufficient algorithmic criteria addressing the above-mentioned differentiability issues.

### 3.1 Application of a Fenichel Theory for Multiple Time Scales

We consider our system (8) over the positive first orthant  $\mathcal{U} = (0, \infty)^n \subseteq \mathbb{R}^n$ . A recent paper by Cardin and Teixeira [11] generalizes Fenichel's theory to provide a solid foundation to obtain more than one nontrivial invariant manifold. This allows, in particular, the reduction of multi-time scale systems such as system (8). Technically, the approach considers a multi-parameter system using time scale factors  $\delta_1, \delta_1 \delta_2, \dots$  instead of increasing powers of one single  $\delta$ .

We let  $\ell \in \{2, \dots, m\}$  and define

$$\beta_1 = b_2 - b_1 = b_2, \quad \dots, \quad \beta_{\ell-1} = b_\ell - b_{\ell-1}, \quad (13)$$

and furthermore  $\delta_1 = \delta^{\beta_1}, \dots, \delta_{\ell-1} = \delta^{\beta_{\ell-1}}$ , and  $\bar{\delta} = (\delta_1, \dots, \delta_{\ell-1})$ .

These definitions allow us to express also all  $\delta^{b'_{k,j}}$  occurring in (8) as products of powers of  $\delta_1, \dots, \delta_{\ell-1}$ , with nonnegative but possibly non-integer rational exponents, via expressing each  $b'_{k,j}$  as a nonnegative rational linear combination of  $\beta_1, \dots, \beta_{\ell-1}$ . This yields

$$\widehat{g}_k(z, \delta_1, \dots, \delta_{\ell-1}) = \widehat{f}_k(z, \bar{\delta}), \quad 1 \leq k \leq m. \quad (14)$$

Moreover, we express  $\delta^{b_{\ell+1}} = \delta_1 \cdots \delta_{\ell-1} \cdot \eta_{\ell+1}, \dots, \delta^{b_m} = \delta_1 \cdots \delta_{\ell-1} \cdot \eta_m$ , via

$$\eta_k(\delta_1, \dots, \delta_{\ell-1}) = \delta^{b_k - b_\ell}, \quad \ell + 1 \leq k \leq m, \quad (15)$$

which is obtained by writing each  $b_k - b_\ell$  as a nonnegative rational linear combination of  $\beta_1, \dots, \beta_{\ell-1}$ . In these terms our system (8) translates to

$$\begin{aligned}
z'_1 &= \widehat{g}_1(z, \bar{\delta}) \\
z'_2 &= \delta_1 \widehat{g}_2(z, \bar{\delta}) \\
&\vdots \\
z'_\ell &= \delta_1 \cdots \delta_{\ell-1} \widehat{g}_\ell(z, \bar{\delta}) \\
z'_{\ell+1} &= \delta_1 \cdots \delta_{\ell-1} \eta_{\ell+1}(\bar{\delta}) \widehat{g}_{\ell+1}(z, \bar{\delta}) \\
&\vdots \\
z'_m &= \delta_1 \cdots \delta_{\ell-1} \eta_m(\bar{\delta}) \widehat{g}_m(z, \bar{\delta}).
\end{aligned} \tag{16}$$

In terms of the right hand sides of (16) the application of relevant results in [11] requires that  $\widehat{g}_1, \dots, \widehat{g}_\ell$  and  $\eta_{\ell+1} \widehat{g}_{\ell+1}, \dots, \eta_m \widehat{g}_m$  are smooth on an open neighborhood of  $\mathcal{U} \times [0, \vartheta_1) \times \cdots \times [0, \vartheta_{\ell-1})$  with  $\vartheta_1 > 0, \dots, \vartheta_{\ell-1} > 0$ . We are going to tacitly assume such smoothness here and address this issue from an algorithmic point of view in Sect. 3.3.

We are now ready to transform our system into  $\ell$  time scales as follows, where possibly  $\ell > 2$ :

$$\tau_1 = \tau, \quad \tau_2 = \delta_1 \tau, \quad \dots, \quad \tau_\ell = \delta_1 \cdots \delta_{\ell-1} \tau.$$

In time scale  $\tau_k$ , with  $1 \leq k \leq \ell$ , system (16) then becomes

$$\begin{aligned}
\delta_1 \cdots \delta_{k-1} \frac{dz_1}{d\tau_k} &= \widehat{g}_1(z, \bar{\delta}) \\
&\vdots \\
\delta_{k-1} \frac{dz_{k-1}}{d\tau_k} &= \widehat{g}_{k-1}(z, \bar{\delta}) \\
\frac{dz_k}{d\tau_k} &= \widehat{g}_k(z, \bar{\delta}) \\
\frac{dz_{k+1}}{d\tau_k} &= \delta_k \widehat{g}_{k+1}(z, \bar{\delta}) \\
&\vdots \\
\frac{dz_\ell}{d\tau_k} &= \delta_k \cdots \delta_{\ell-1} \widehat{g}_\ell(z, \bar{\delta}) \\
\frac{dz_{\ell+1}}{d\tau_k} &= \delta_k \cdots \delta_{\ell-1} \eta_{\ell+1}(\bar{\delta}) \widehat{g}_{\ell+1}(z, \bar{\delta}) \\
&\vdots \\
\frac{dz_m}{d\tau_k} &= \delta_k \cdots \delta_{\ell-1} \eta_m(\bar{\delta}) \widehat{g}_m(z, \bar{\delta}).
\end{aligned} \tag{17}$$

For  $k = 1$  and  $k = \ell$  we obtain empty products, which yield the neutral element 1, as usual.

Similarly to Sect. 2.1, we are interested in the asymptotic behavior for  $\bar{\delta} \rightarrow 0$ , which is approximated by the elimination of higher order terms. We are now going to introduce a construction required for a justification of this approximation, which also clarifies the greatest possible choice for  $\ell \leq m$  above. Define  $F_0 = 0$  and

$$Z_k = \begin{pmatrix} z_1 \\ \vdots \\ z_k \end{pmatrix}, \quad F_k(z, \delta) = \begin{pmatrix} \widehat{f}_1(z, \delta) \\ \vdots \\ \widehat{f}_k(z, \delta) \end{pmatrix}, \quad 1 \leq k \leq m.$$

With system (8) in mind, we are going to use  $\widehat{f}_k(z, 0)$  in favor of  $\widehat{g}_k(z, 0, \dots, 0)$ . It is easy to see that both are equal. We define furthermore

$$M_k = (F_k(z^*, 0) = 0), \quad \mathcal{M}_k = \{z^* \in \mathcal{U} \mid F_k(z^*, 0) = 0\}, \quad 0 \leq k \leq m. \quad (18)$$

The sets  $\mathcal{M}_k$  are obtained from varieties defined by the systems  $M_k$  via intersection with the first orthant. Furthermore,

$$\mathcal{U} = \mathcal{M}_0 \supseteq \mathcal{M}_1 \supseteq \dots \supseteq \mathcal{M}_m \quad (19)$$

establishes a chain of nested subvarieties, again intersected with the first orthant.

We define that  $\mathcal{M}_1$  is *hyperbolically attractive* on  $\mathcal{M}_0$ , if  $\mathcal{M}_1 \neq \emptyset$ , and for all  $z \in \mathcal{M}_1$  all eigenvalues of the Jacobian  $D_{z_1} \widehat{f}_1(z, 0)$  have negative real parts. Therefore  $\mathcal{M}_1$  is a manifold. For  $k \in \{2, \dots, m\}$ ,  $\mathcal{M}_k$  is hyperbolically attractive on  $\mathcal{M}_{k-1}$ , if  $\mathcal{M}_k \neq \emptyset$ , and the following holds. Recall that using the defining polynomials  $F_{k-1}$  of  $\mathcal{M}_{k-1}$ , the implicit function theorem yields a unique local resolution of  $Z_{k-1}$  as functions of  $z_k, \dots, z_m$ , provided that  $D_{Z_{k-1}} F_{k-1}$  has no zero eigenvalues. We thus obtain

$$\widehat{f}_k(z, 0) = \widehat{f}_k^*(z_k, \dots, z_m, 0) \quad \text{on } \mathcal{M}_{k-1}.$$

For our definition we now require that for all  $z \in \mathcal{M}_k$  all eigenvalues of  $D_{z_k} \widehat{f}_k^*(z_k, \dots, z_m, 0)$  have negative real parts. Again,  $\mathcal{M}_k$  is a manifold. When  $\mathcal{M}_k$  is hyperbolically attractive on  $\mathcal{M}_{k-1}$  we write  $\mathcal{M}_{k-1} \triangleright \mathcal{M}_k$ , where  $Z_k$  will be clear from the context.

If we find for some  $\ell \in \{1, \dots, m\}$  that  $\mathcal{M}_0 \triangleright \mathcal{M}_1, \mathcal{M}_1 \triangleright \mathcal{M}_2, \dots, \mathcal{M}_{\ell-1} \triangleright \mathcal{M}_\ell$ , then we simply write  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$ , and call this a *hyperbolically attractive  $\ell$ -chain*. Such a chain is called *maximal* if either  $\ell = m$  or  $\mathcal{M}_\ell \not\triangleright \mathcal{M}_{\ell+1}$ .

Let  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  be a hyperbolically attractive  $\ell$ -chain. Consider for each  $k \in \{1, \dots, \ell\}$  the following differential-algebraic system:

$$0 = F_{k-1}(z, 0), \quad \frac{dz_k}{d\tau_k} = \widehat{f}_k(z, 0), \quad \frac{dz_{k+1}}{d\tau_k} = 0, \quad \dots, \quad \frac{dz_m}{d\tau_k} = 0. \quad (20)$$

In the limiting case  $\bar{\delta} = 0$ , this corresponds to system (17). Recall that

$$\tau_k = \delta_1 \cdots \delta_{k-1} \tau = \delta^{b_2 - b_1} \cdots \delta^{b_k - b_{k-1}} \tau = \delta^{b_k - b_1} \tau = \delta^{b_k} \tau,$$

and equivalently rewrite (20) as a triplet  $(M_{k-1}, T_k, R_k)$  with entries as follows:

$$F_{k-1}(z, 0) = 0, \quad \frac{dz_k}{d\tau} = \delta^{b_k} \widehat{f}_k(z, 0), \quad \frac{dz_{k+1}}{d\tau} = \dots = \frac{dz_m}{d\tau} = 0. \quad (21)$$

For a given index  $k$ , we call  $(M_{k-1}, T_k, R_k)$  a *reduced system* on  $\mathcal{M}_{k-1}$ , where the relevant hyperbolic attractivity relation is  $\mathcal{M}_{k-1} \triangleright \mathcal{M}_k$ . In order to indicate the relevance of  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  we write  $(M_0, T_1, R_1) \triangleright \dots \triangleright (M_{\ell-1}, T_\ell, R_\ell)$  also for reduced systems, where  $\mathcal{M}_\ell$  is not made explicit but relevant for the last triplet. Slightly abusing language, we speak of a hyperbolically attractive  $\ell$ -chain of reduced systems, which is maximal if  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  is.

The following theorem is a consequence of [11, Theorem A and Corollary A], specialized to the situation at hand.

**Theorem 1** *Let  $\ell \geq 2$ . Assume that  $(M_0, T_1, R_1) \triangleright \dots \triangleright (M_{\ell-1}, T_\ell, R_\ell)$  is a hyperbolically attractive  $\ell$ -chain of reduced systems for system (16). Let  $K \subseteq \mathcal{U}$  be compact. Then for sufficiently small  $\bar{\delta}$  and all  $k \in \{1, \dots, \ell\}$ , system (16) admits invariant manifolds  $\mathcal{N}_{k-1}$  that depend on  $\bar{\delta}$  and are  $(\delta_1 + \dots + \delta_{k-1})$ -close to  $\mathcal{M}_{k-1} \cap K$  with respect to the Hausdorff distance. Moreover, there exists  $T > 0$  such that solutions of system (16) on  $\mathcal{N}_{k-1}$  in time scale  $\tau_k$  converge to solutions of  $(M_{k-1}, T_k, R_k)$ , uniformly on any closed subinterval of  $(0, T)$ , as  $\bar{\delta} \rightarrow 0$ .  $\square$*

For  $k \in \{1, \dots, \ell\}$ , the  $\mathcal{M}_{k-1}$  are critical manifolds, which contain only stationary points. The systems  $(T_k, R_k)$  of ordinary differential equations on  $\mathcal{M}_{k-1}$  approximate invariant manifolds  $\mathcal{N}_{k-1}$  in the sense of the theorem. They furthermore approximate solutions in time scale  $\tau_k$  of system (16), which is equivalent to our system (8). In other words, system (8) admits a succession of invariant manifolds, on which the behavior in the appropriate time scale is approximated by the respective reduced equations (20) and, equivalently, (21). Note that only the  $\delta^{b_k} \widehat{f}_k(z, 0)$  without the higher order terms enter the reduced systems  $(M_{k-1}, T_k, R_k)$ .

Algorithm 5 now starts with the output  $[T_1, \dots, T_m]$  of Algorithm 1, which represents the scaled system (9). Notice that each  $T_k$  already meets the specification in (21). In 1.1 we define  $U$  to contain defining inequalities of the first orthant  $\mathcal{U}$ . Starting with  $k = 1$ , the for-loop in 1.4–15 successively constructs  $M_k$  and  $R_k$  such that in combination with  $T_k$  from the input,  $(M_{k-1}, T_k, R_k)$  forms a reduced system as in (21). The loop stops when either  $k = m + 1$  or a test for hyperbolic attractivity in 1.8 finds that  $\mathcal{M}_{k-1} \not\triangleright \mathcal{M}_k$ . We are going to discuss this test in detail in the next section. Note that we maintain a matrix  $A$  for storing information between the subsequent calls of our test. In either case we arrive at a maximal hyperbolically attractive  $(k - 1)$ -chain of reduced systems given as a list  $[(M_0, T_1, R_1), \dots, (M_{k-2}, T_{k-1}, R_{k-1})]$ . Following the notational convention used throughout this section we set  $\ell$  to  $k - 1$  in 1.17. The test in 1.18–20 reflects the choice of  $\ell \in \{2, \dots, m\}$  at the beginning of this section. Finally, 1.21 uses the second input  $[P_1, \dots, P_m]$  of the algorithm to address the smoothness requirements for system (16). We are going to discuss the corresponding procedure in detail in Sect. 3.3. It will turn out that this procedure provides only a sufficient test. Therefore we issue in case of failure only a warning, allowing the user to verify smoothness a posteriori, using alternative algorithms or human intelligence. One might mention that it is actually sufficient to consider weaker, finite differentiability conditions instead of smoothness, which can be seen by inspection of the proofs in [11].

From an application point of view, attracting invariant manifolds are relevant in the context of biological networks, and our notion of hyperbolic attractivity holds for large classes of such networks [24]. This is our principal motivation for using hyperbolic attractivity here. From a computational perspective, hyperbolic attractivity can be tested based on Hurwitz criteria, as we are going to make explicit in the next section.

The relevant results in [11], in contrast, are based on a series of hyperbolicity conditions, which are somewhat weaker than hyperbolic attractivity. Hyperbolicity can be tested algorithmically as well, albeit with more effort. For approaches based on Routh’s work see, e.g., [27, Chapter V, §4], which checks the number of purely imaginary eigenvalues of a real polynomial via the Cauchy index of a related rational function.

### 3.2 Verification of Hyperbolic Attractivity

Our definition of hyperbolic attractivity  $\mathcal{M}_{k-1} \triangleright \mathcal{M}_k$  refers to the eigenvalues of the Jacobians of the  $\widehat{f}_k^*$ , which cannot be directly obtained from the Jacobians of the  $\widehat{f}_k$  [11, 12]. Generalizing work on systems with three time scales [40], we take in this section a linear algebra approach to obtain the relevant eigenvalues without computing the  $\widehat{f}_k^*$ .

To start with, recall the well-known *Hurwitz criterion* [36]:

**Theorem 2** (Hurwitz, 1895). Consider  $f = a_0x^n + a_1x^{n-1} + \dots + a_n \in \mathbb{R}[x]$ ,  $a_0 > 0$ . For  $i \in \{1, \dots, n\}$  define

$$H_i = \begin{pmatrix} a_1 & a_3 & a_5 & \dots & a_{2i-1} \\ a_0 & a_2 & a_4 & \dots & a_{2i-2} \\ 0 & a_1 & a_3 & \dots & a_{2i-3} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \dots & a_i \end{pmatrix}, \quad \Delta_i = \det H_i.$$

Then all complex zeros of  $f$  have negative real parts if and only if  $\Delta_1 > 0, \dots, \Delta_n > 0$ . Notice that  $\Delta_n = a_n \Delta_{n-1}$ , and therefore  $\Delta_n > 0$  can be equivalently replaced with  $a_n > 0$ .

We call  $H_n$  the *Hurwitz matrix* and  $\Delta_i$  the *i-th Hurwitz determinant* of  $f$ . Furthermore, we refer to  $\Gamma = (\Delta_1 > 0 \wedge \dots \wedge \Delta_{n-1} > 0 \wedge a_n > 0)$  as the *Hurwitz conditions* for  $f$ .

**Algorithm 5** ComputeReducedSystems**Input:** Output of Algorithm 1:

1.  $[T_1, \dots, T_m]$ , a list of lists  $z'_k = \delta^{b_k} f_k$ ;
2.  $[P_1, \dots, P_m]$ , a list of lists of polynomials in  $\mathbb{Q}[x_1, \dots, x_n][\delta]$ ;

We denote  $\xi_k := |T_k|$ ,  $\Xi_k := \sum_{i=1}^k \xi_i$ , and  $X = (x_1, \dots, x_n)$ .

**Output:** A list  $[(M_0, T_1, R_1), \dots, (M_{\ell-1}, T_\ell, R_\ell)]$  of triplets where  $\ell \in \{2, \dots, m\}$ , or the empty list. For  $k \in \{1, \dots, \ell\}$ ,  $M_{k-1}$  is a list of real constraints defining  $\mathcal{M}_{k-1} \subseteq \mathbb{R}^n$ ;  $T_k$  is a list of differential equations;  $R_k$  is a list of trivial differential equations  $x' = 0$  for all differential variables from  $T_{k+1}, \dots, T_m$ .

The triplets  $(M_{k-1}, T_k, R_k)$  represent reduced systems according to (21).

```

1:  $U := [x_1 > 0, \dots, x_n > 0]$ 
2:  $M_0, Z, F := []$ 
3:  $A := ()$ 
4: for  $k := 1$  to  $m$  do
5:    $z := [x \mid x' = \delta^{b_k} g \in T_k]$ 
6:    $f := [g \mid x' = \delta^{b_k} g \in T_k]$ 
7:    $M_k := M_{k-1} \circ [f = 0]$ 
8:    $\varphi, A := \text{IsHyperbolicallyAttractive}(U \circ M_k, Z, z, F, f, k, A)$ 
9:   if not  $\varphi$  then
10:    break
11:  end if
12:   $R_k := [x' = 0 \mid x' = h \in T_{k+1} \cup \dots \cup T_m]$ 
13:   $Z := Z \circ z$ 
14:   $F := F \circ f$ 
15: end for
16: # We either broke in line 10 preserving  $k$ , or we have  $k = m + 1$ .
17:  $\ell := k - 1$ 
18: if  $\ell < 2$  then
19:  return  $()$ 
20: end if
21: if  $\text{TestSmoothness}([T_1, \dots, T_m], [P_1, \dots, P_m], \ell) = \text{failed}$  then
22:  print "Warning: differentiability requires further verification"
23: end if
24: return  $[(M_0, T_1, R_1), \dots, (M_{\ell-1}, T_\ell, R_\ell)]$ 

```

$$\begin{aligned} &\subseteq X, |z| = \xi_k \\ &= \hat{f}_k(z, 0) \in \mathbb{Q}[X]^{\xi_k} \\ &= M_0 \circ [F = 0] \circ [f = 0] \end{aligned}$$

$$\begin{aligned} &\Xi_{k-1} + \xi_k + |R_k| = n \\ &\subseteq X, |Z| = \Xi_k \\ &\in \mathbb{Q}[X]^{\Xi_k} \end{aligned}$$

Our first result generalizes [40, Proposition 1 (ii)]. The proof is straightforward by induction using the argument in [40, Lemma 3] and its proof.

**Lemma 3** For  $k \in \{1, \dots, m\}$  define

$$J_k = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \varrho_1 \cdots \varrho_{k-1} \end{pmatrix} \cdot D_{Z_k} F_k(z, 0) = \begin{pmatrix} D_{z_1} \hat{f}_1(z, 0) & \dots & D_{z_k} \hat{f}_1(z, 0) \\ \varrho_1 D_{z_1} \hat{f}_2(z, 0) & \dots & \varrho_1 D_{z_k} \hat{f}_2(z, 0) \\ \dots & \dots & \dots \\ \varrho_1 \cdots \varrho_{k-1} D_{z_1} \hat{f}_k(z, 0) & \dots & \varrho_1 \cdots \varrho_{k-1} D_{z_k} \hat{f}_k(z, 0) \end{pmatrix}.$$

Let  $\ell \in \{1, \dots, m\}$ . Then  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  if and only if  $\mathcal{M}_\ell \neq \emptyset$  and for all  $k \in \{1, \dots, \ell\}$ , all sufficiently small  $\varrho_1^* > 0, \dots, \varrho_{k-1}^* > 0$ , and all  $z^* \in \mathcal{M}_k$ , all eigenvalues of  $J_k(\varrho_1^*, \dots, \varrho_{k-1}^*, z^*)$  have negative real parts.

In particular, one can choose  $\varrho_1^* = \dots = \varrho_{k-1}^* = \varrho^*$  with sufficiently small  $\varrho^*$  and consider  $J'_k = \text{diag}(1, \dots, \varrho^{k-1}) \cdot D_{Z_k} F_k(z, 0)$ .

Let  $\Gamma_k$  denote the Hurwitz conditions for the characteristic polynomial of  $J'_k$ . Then Lemma 3 allows to state hyperbolic attractivity  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  as a first-order formula over the reals as follows:

$$\left( \exists (0 < z) : F_\ell(z, 0) = 0 \right) \wedge \left( \bigwedge_{k=1}^{\ell} \exists (0 < \sigma) \forall (0 < \varrho < \sigma) \forall (0 < z) : F_k(z, 0) = 0 \Rightarrow \Gamma_k(\varrho, z) \right). \quad (22)$$

On these grounds, any real decision procedure [14, 73, 80] provides an effective test for hyperbolic attractivity. However, our formulation (22) uses a quantifier alternation  $\exists\sigma\forall\varrho$  in its second part. We would like to use this in order to use SMT solving over a quantifier-free logic. Our next result allows a suitable first-order formulation without quantifier alternation. Its proof combines [40, Lemma 3] with our Lemma 3.

**Proposition 4** (Effective Characterization of Hyperbolically Attractive  $\ell$ -Chains). *Define  $A_1 = D_{z_1} \widehat{f}_1(z, 0)$ . For  $k \in \{2, \dots, m\}$  define*

$$\begin{pmatrix} A_{k-1} & B_k \\ C_k & V_k \end{pmatrix} = \begin{pmatrix} D_{Z_{k-1}} F_{k-1}(z, 0) & D_{z_k} F_{k-1}(z, 0) \\ D_{Z_{k-1}} \widehat{f}_k(z, 0) & D_{z_k} \widehat{f}_k(z, 0) \end{pmatrix},$$

and note that  $\begin{pmatrix} A_{k-1} & B_k \\ C_k & V_k \end{pmatrix} = A_k$ .

Let  $\ell \in \{1, \dots, m\}$ . Then  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  if and only if

- (i)  $\mathcal{M}_\ell \neq \emptyset$ ,
- (ii) For all  $z^* \in \mathcal{M}_1$  all eigenvalues of  $W_1(z^*)$ , where  $W_1 = A_1$ , have negative real parts,
- (iii) For all  $k \in \{2, \dots, \ell\}$  and all  $z^* \in \mathcal{M}_k$ ,  $A_{k-1}(z^*)$  is regular and all eigenvalues of  $W_k(z^*)$ , where  $W_k = V_k - C_k A_{k-1}^{-1} B_k$ , have negative real parts.

*Proof* Assume  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$ . By Lemma 3 we have  $\mathcal{M}_\ell \neq \emptyset$ . For all  $z^* \in \mathcal{M}_1$ , all eigenvalues of the Jacobian  $W_1(z^*)$  have negative real parts by the definition of hyperbolic attractivity. Let now  $k \in \{2, \dots, \ell\}$ ,  $z^* \in \mathcal{M}_k$ , and define  $P = \text{diag}(1, \dots, \varrho^{k-2})$ . Using Lemma 3 we fix  $0 < \tau^* < 1$  such that for all  $0 < \varrho^* < \tau^*$  all eigenvalues of  $J'_{k-1}(\varrho^*, z^*) = P(\varrho^*) A_{k-1}(z^*)$  have negative real parts. It follows that  $P(\varrho^*) A_{k-1}(z^*)$ ,  $P(\varrho^*)$ , and  $A_{k-1}(z^*)$  are all regular. Next, consider

$$J'_k = \begin{pmatrix} P A_{k-1} & P B_k \\ \varrho^{k-1} C_k & \varrho^{k-1} V_k \end{pmatrix}.$$

Using Lemma 3 once more, we find  $0 < \sigma^* < \tau^*$  such that for all  $0 < \varrho^* < \sigma^*$  also all eigenvalues of  $J'_k(\varrho^*, z^*)$  have negative real parts. Now  $J'_k(\varrho^*, z^*)$  satisfies condition (ii) of [40, Lemma 3] with  $\delta = \sigma^*$  and  $\varepsilon = (\varrho^*)^{k-1}$ , which allows us to conclude that all eigenvalues of  $(V_k - C_k (P A_{k-1})^{-1} P B_k)(\varrho^*, z^*) = (V_k - C_k A_{k-1}^{-1} P^{-1} P B_k)(\varrho^*, z^*) = W_k(z^*)$  have negative real parts as well.

Assume, vice versa, that (i)–(iii) hold. We use induction on  $k$  to show  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_k$  for  $1 \leq k \leq \ell$ . For  $k = 1$  we have  $\mathcal{M}_0 \triangleright \mathcal{M}_1$  by definition of hyperbolic attractivity. Assume that  $2 \leq k \leq \ell$  and  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_{k-1}$ . By Lemma 3 there exists  $0 < \tau^*$  such that for all  $0 < \sigma^* < \tau^*$  and all  $z^* \in \mathcal{M}_{k-1}$  all eigenvalues of  $P(\sigma^*) A_{k-1}(z^*)$  have negative real parts, where  $P = \text{diag}(1, \dots, (\sigma^*)^{k-2})$ . We rewrite  $W_k = V_k - C_k (P A_{k-1})^{-1} P B_k$ . Then  $W_k(z^*)$  satisfies condition (i) of [40, Lemma 3] with  $A = P(\sigma^*) A_{k-1}(z^*)$ ,  $B = P(\sigma^*) B_k(z^*)$ ,  $C = C_k(z^*)$  and  $D = V_k(z^*)$ . Thus there exists  $0 < \delta$  such that for all  $0 < \varepsilon < \delta$  all eigenvalues of

$$\begin{pmatrix} P(\sigma^*) A_{k-1}(z^*) & P(\sigma^*) B_k(z^*) \\ \varepsilon C_k(z^*) & \varepsilon V_k(z^*) \end{pmatrix}$$

have negative real parts. Choosing  $\varrho^* = \min\{\sigma^*, \varepsilon^{-1/\sqrt{\varepsilon}}\}$  in Lemma 3 yields  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_k$ .  $\square$

From now on let  $\Gamma_k$  denote the Hurwitz conditions for the characteristic polynomial of  $W_k$ , which—in contrast to the ones used in (22)—do not depend on  $\varrho$  anymore.

**Corollary 5** (Logic-Based Test for Hyperbolically Attractive  $\ell$ -Chains). *For  $k \in \{1, \dots, m\}$  define*

$$\begin{aligned} \varphi_k &= (\exists(0 < z) : F_k(z, 0) = 0), \\ \psi_k &= (\forall(0 < z) : F_k(z, 0) = 0 \Rightarrow \Gamma_k(z)). \end{aligned}$$

Let  $\ell \in \{1, \dots, m\}$ . Then  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  if and only if  $\mathbb{R} \models \varphi_\ell \wedge \bigwedge_{k=1}^{\ell} \psi_k$ .

*Proof* Assume  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$ . Then Proposition 4 yields its conditions (i)–(iii). Now,  $\varphi_\ell$  holds as a formalization of (i). Furthermore,  $\psi_1$  holds as a formalization of (ii), and the validity of  $\psi_2, \dots, \psi_\ell$  follows directly from (iii). Hence  $\mathbb{R} \models \varphi_\ell \wedge \bigwedge_{k=1}^\ell \psi_k$ .

Assume, vice versa, that  $\mathbb{R} \models \varphi_\ell \wedge \bigwedge_{k=1}^\ell \psi_k$ . We show  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$  by induction on  $\ell$ . If  $\ell = 1$ , then  $\varphi_1$  formalizes (i) and  $\psi_1$  formalizes (ii) in Proposition 4, and we obtain  $\mathcal{M}_0 \triangleright \mathcal{M}_1$ . Let now  $\ell > 1$ . Then  $\varphi_\ell$  formalizes Proposition 4(i). Our induction hypothesis yields  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_{\ell-1}$ . By Lemma 3 there exists  $0 < \tau^*$  such that for all  $0 < \sigma^* < \tau^*$  and all  $z^* \in \mathcal{M}_{\ell-1} \supseteq \mathcal{M}_\ell$  all eigenvalues of  $P(\sigma^*)A_{\ell-1}(z^*)$ , where  $P = \text{diag}(1, \dots, (\sigma^*)^{\ell-2})$ , have negative real parts. In particular,  $P(\sigma^*)A_{\ell-1}(z^*)$  is regular and so is  $A_{\ell-1}(z^*)$ . Furthermore, the Hurwitz conditions in  $\psi_\ell$  guarantee for all  $z^* \in \mathcal{M}_\ell$  that all eigenvalues of  $W_\ell(z^*)$  have negative real parts. Taking these observations together, Proposition 4(iii) is satisfied, hence  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_\ell$ .  $\square$

In contrast to (22), our first-order characterization

$$\left( \exists(0 < z) : F_\ell(z, 0) = 0 \right) \wedge \left( \bigwedge_{k=1}^\ell \forall(0 < z) : F_k(z, 0) = 0 \Rightarrow \Gamma_k(z) \right) \quad (23)$$

in Corollary 5 has no quantifier alternation. Note that the two top-level components of (23) establish two independent decision problems, addressing non-emptiness of the manifold and our requirement on the eigenvalues, respectively.

It is easy to see that for all  $\ell \in \{1, \dots, m\}$  and all  $k \in \{1, \dots, \ell-1\}$ ,  $\varphi_\ell$  entails  $\varphi_k$ . Thus (23) can be equivalently rewritten as  $\bigwedge_{k=1}^\ell (\varphi_k \wedge \psi_k)$ , explicitly:

$$\bigwedge_{k=1}^\ell \left( \exists(0 < z) : F_k(z, 0) = 0 \wedge \forall(0 < z) : F_k(z, 0) = 0 \Rightarrow \Gamma_k(z) \right). \quad (24)$$

Our approach tests the conjunction in (24) using a for-loop over  $k$  in Algorithm 5. Technically, this construction ensures with the test for  $\mathcal{M}_{k-1} \triangleright \mathcal{M}_k$  in Algorithm 6 that  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_{k-1}$  already holds, and exploits the fact that  $\psi_k$  and  $\varphi_k$  do not refer to smaller indices than  $k$ .

In 1.1–3 we test the validity of  $\varphi_k$ . Using from the input the defining inequalities and equations  $M = U \circ M_k$  of  $\mathcal{M}_k$  along with  $Z = Z_{k-1}$ ,  $z = z_k$ ,  $F = F_{k-1}$ ,  $f = f_k$ , and  $A = A_{k-1}$ , we construct in 1.4–13  $A' = A_k$  as noted in Proposition 4. In 1.14–19 we construct the Hurwitz conditions  $\Gamma = \Gamma_k$  according to Theorem 2. On the grounds of the validity of  $\varphi_k$  tested in 1.1, we finally test in 1.20 the validity of  $\psi_k$  and return a corresponding Boolean value. We additionally return  $A' = A_k$  for reuse with the next iteration. The validity tests for  $\varphi_k$  and  $\psi_k$  in 1.1 and 1.20, respectively, again amount to SMT solving, this time using the logic QF\_NRA [2] for quantifier-free nonlinear real arithmetic. Recall the positive integer parameter  $p$  used for the precision with both Algorithm 2 and Algorithm 3. For  $p > 1$  symbolic computation possibly yields fractional powers of numbers in the defining equations for manifolds as well as in the vector fields of the differential equations. Such expressions are not covered by QF\_NRA. When this happens, we catch the corresponding error from the SMT solver and restart with floats.

### 3.3 Sufficient Smoothness Criteria

Let us get back to the requirement in Sect. 3.1 that  $\widehat{g}_1, \dots, \widehat{g}_\ell$  and  $\eta_{\ell+1}\widehat{g}_{\ell+1}, \dots, \eta_m\widehat{g}_m$  occurring on the right hand sides of system (16) are all smooth on an open neighborhood of  $\mathcal{U} \times [0, \vartheta_1) \times \dots \times [0, \vartheta_{\ell-1})$  with  $\vartheta_1 > 0, \dots, \vartheta_{\ell-1} > 0$ . A first sufficient criterion for smoothness is that all those expressions are polynomials in  $z$  and  $\bar{\delta}$ .

Recall the definitions of  $\widehat{g}_k$  for  $k \in \{1, \dots, m\}$  in (14) and of  $\eta_k$  for  $k \in \{\ell+1, \dots, m\}$  in (15). For  $k \in \{1, \dots, m\}$  and  $j \in \{1, \dots, w_k\}$  one finds nonnegative  $r_1, \dots, r_{\ell-1} \in \mathbb{Q}$  such that

$$\langle (\beta_1, \dots, \beta_{\ell-1}), (r_1, \dots, r_{\ell-1}) \rangle = b'_{k,j},$$

and for  $k \in \{\ell+1, \dots, m\}$  one finds nonnegative  $r_1, \dots, r_{\ell-1} \in \mathbb{Q}$  such that

$$\langle (\beta_1, \dots, \beta_{\ell-1}), (r_1, \dots, r_{\ell-1}) \rangle = b_k - b_\ell.$$



**Algorithm 6** IsHyperbolicallyAttractive**Input:** 1.  $M$ , 2.  $Z$ , 3.  $z$ , 4.  $F$ , 5.  $f$ , 6.  $k$ , 7.  $A$ , as in the calling Algorithm 5Knowing that  $\mathcal{M}_0 \triangleright \dots \triangleright \mathcal{M}_{k-1}$ , we check here whether also  $\mathcal{M}_{k-1} \triangleright \mathcal{M}_k$ . We denote  $\xi := |f| = |z|$ ,  $\Xi := |F| = |Z|$ , and  $X = \{x_1, \dots, x_n\}$ . In these terms,  $A \in \mathbb{Q}[X]^{\Xi \times \Xi}$ .**Output:** 1. Boolean, 2.  $A' \in \mathbb{Q}[X]^{(\Xi+\xi) \times (\Xi+\xi)}$ 

```

1: if not  $\mathbb{R} \models \exists \wedge M$  then
2:   return false, ()
3: end if
4:  $V := \text{Jacobian}(f, z)$   $\in \mathbb{Q}[X]^{\xi \times \xi}$ 
5: if  $k = 1$  then
6:    $W := V$ 
7:    $A' := V$ 
8: else
9:    $B := \text{Jacobian}(F, z)$   $\in \mathbb{Q}[X]^{\Xi \times \xi}$ 
10:   $C := \text{Jacobian}(f, Z)$   $\in \mathbb{Q}[X]^{\xi \times \Xi}$ 
11:   $W := V - CA^{-1}B$   $\in \mathbb{Q}[X]^{\xi \times \xi}$ 
12:   $A' := \begin{pmatrix} A & B \\ C & V \end{pmatrix}$   $\in \mathbb{Q}[X]^{(\Xi+\xi) \times (\Xi+\xi)}$ 
13: end if
14:  $\chi := \lambda^\xi + \dots + a_\xi := \text{CharacteristicPolynomial}(W)$   $\in \mathbb{Q}[X][\lambda]$ 
15:  $H := \text{HurwitzMatrix}(\chi)$   $\in \mathbb{Q}[X]^{\xi \times \xi}$ 
16: for  $j := 1$  to  $\xi - 1$  do
17:    $\Delta_j := \det(H_{r,s})_{1 \leq r,s \leq j}$   $\in \mathbb{Q}[X]$ 
18: end for
19:  $\Gamma := \{\Delta_1 > 0, \dots, \Delta_{\xi-1} > 0, a_\xi > 0\}$ 
20: return  $\mathbb{R} \models \forall (\wedge M \rightarrow \wedge \Gamma), A'$ 

```

Such representations always exist but are not unique in general. If one even finds suitable nonnegative integers  $r_1, \dots, r_{\ell-1} \in \mathbb{N}$ , which do not always exist, then one obtains  $\widehat{g}_1, \dots, \widehat{g}_m$  as polynomials in  $z$  and  $\bar{\delta}$ , and  $\eta_{\ell+1}, \dots, \eta_m$  as polynomials in  $\bar{\delta}$ , which is sufficient for our first criterion above.

An improved but still only sufficient criterion uses similar constructions to directly verify the existence of polynomial representations of the products  $\eta_{\ell+1}\widehat{g}_{\ell+1}, \dots, \eta_m\widehat{g}_m$ , in contrast to considering the factors independently. From an algorithmic point of view, we furthermore have to take into account that  $P_1, \dots, P_m$  obtained in Algorithm 1 do not contain  $b'_{k,j}$  but  $b_k + b'_{k,j}$ . For  $k \in \{1, \dots, \ell\}$  and  $j \in \{1, \dots, w_k\}$  we try to find  $r_1, \dots, r_{\ell-1} \in \mathbb{N}$  such that

$$\langle (\beta_1, \dots, \beta_{\ell-1}), (r_1, \dots, r_{\ell-1}) \rangle = b'_{k,j} = (b_k + b'_{k,j}) - b_k > 0, \quad (25)$$

and for  $k \in \{\ell + 1, \dots, m\}$  we try to find  $r_1, \dots, r_{\ell-1} \in \mathbb{N}$  such that

$$\langle (\beta_1, \dots, \beta_{\ell-1}), (r_1, \dots, r_{\ell-1}) \rangle = (b_k - b_\ell) + b'_{k,j} = (b_k + b'_{k,j}) - b_\ell > 1. \quad (26)$$

Notably, such representations exist whenever  $1 \in \{\beta_1, \dots, \beta_{\ell-1}\}$ .

On these grounds, we introduce Algorithm 7,

which specifies the sufficient test applied in 1.21 of Algorithm 5. The first two parameters  $[T_1, \dots, T_m]$  and  $[P_1, \dots, P_m]$  originate from Algorithm 1, while the last parameter  $\ell$  originates from the calling Algorithm 5.

In 1.1–8 of Algorithm 7 we compute  $\beta_1, \dots, \beta_{\ell-1}$  as defined in (13) and simultaneously obtain  $b_1, \dots, b_\ell$ . In 1.9–14 we compute the right hand sides of the conditions in (25) or (26), depending on the current index  $k$ . For checking those conditions in 1.16 we once more employ SMT solving, this time using the adequate logic `QF_LIA` [2] for quantifier-free linear integer arithmetic. Since we are aiming at nonnegative integer solutions, we introduce explicit non-negativity conditions  $r_1 \geq 0, \dots, r_{\ell-1} \geq 0$ . In case of unsatisfiability Algorithm 7 returns “failed” in 1.17. Recall that in this case the calling Algorithm 5 issues a warning but continues. In case of satisfiability, in contrast, smoothness is guaranteed, we reach 1.20, and return “true.” We remark that the computation time spent

**Algorithm 7** TestSmoothness

**Input:**  $[T_1, \dots, T_m], [P_1, \dots, P_m], \ell$  as in the calling Algorithm 5:

1.  $[T_1, \dots, T_m]$ , a list of lists  $z'_k = \delta^{b_k} f_k$ ;
2.  $[P_1, \dots, P_m]$ , a list of lists of polynomials in  $\mathbb{Q}[x_1, \dots, x_n][\delta]$ ;
3.  $\ell \in \mathbb{N}, \ell \geq 2$ ;

We check here a sufficient criterion for smoothness as required for (16).

**Output:** “true” or “failed” in terms of a 3-valued logic;

```

1:  $b_1 := 0$ 
2: for  $k := 2$  to  $\ell$  do
3:    $b_k :=$  the unique exponent of  $\delta$  in  $T_k$ 
4:    $\beta_{k-1} := b_k - b_{k-1}$ 
5:   if  $\beta_{k-1} = 1$  then
6:     return true
7:   end if
8: end for
9:  $E := \emptyset$ 
10: for  $k = 1$  to  $m$  do
11:   for all  $p$  in  $P_k$  do
12:      $E := E \cup \{ \deg_\delta m - b_{\min(k, \ell)} \mid m \text{ monomial of } p \}$ 
13:   end for
14: end for
15: for all  $e \in E$  do
16:   if not  $\mathbb{Z} \models \exists r_1 \dots \exists r_{\ell-1} (r_1 \geq 0 \wedge \dots \wedge r_{\ell-1} \geq 0 \wedge \langle (\beta_1, \dots, \beta_{\ell-1}), (r_1, \dots, r_{\ell-1}) \rangle = e)$  then
17:     return failed
18:   end if
19: end for
20: return true

```

on  $E$  is negligible compared to the SMT solving later on. The construction of the entire set  $E$  beforehand avoids duplicate SMT instances.

#### 4 Algebraic Simplification of Reduced Systems

In the output  $(M_0, T_1, R_1), \dots, (M_{\ell-1}, T_\ell, R_\ell)$  of Algorithm 5, the  $T_k$  are taken literally from the input, and the  $M_{k-1}$  and  $R_k$  are obtained via quite straightforward rewriting of the input. As a matter of fact, the computationally hard part of Algorithm 5 consists in the computation of the upper index  $\ell$ . We now want to rewrite the triplets  $(M_{k-1}, T_k, R_k)$  once more, aiming at less straightforward but simpler and, hopefully, more intuitive representations. The principal idea is to heuristically eliminate on the right hand side of the differential equations in  $T_k$  those variables whose derivatives have already occurred as left hand sides in one of the  $T_1, \dots, T_{k-1}$ . Of course, our simplifications will preserve all relevant properties of  $(M_0, T_1, R_1), \dots, (M_{\ell-1}, T_\ell, R_\ell)$ , such as hyperbolic attractivity and sufficient differentiability. Technically, our next Algorithm 8 employs Gröbner basis techniques [3, 10].

Recall that  $z_k$  are the variables occurring on the left hand sides of differential equations in  $T_k$ , and  $Z_{k-1} = (z_1, \dots, z_{k-1})$ . In 1.1–5 we construct a block term order  $\omega$  on all variables  $\{x_1, \dots, x_n\}$  so that variables from  $Z_{k-1}$  are larger than variables from  $z_k$ . This ensures that all multivariate polynomial reductions with respect to  $\omega$  throughout our algorithm will eliminate variables from  $Z_{k-1}$  in favor of variables from  $z_k$  rather than vice versa. Prominent examples for such block orders are pure lexicographic orders, but ordering by total degree inside the  $z_1, \dots, z_\ell, y$  will heuristically give more efficient computations.

Recall that the radical ideal  $\sqrt{\langle F \rangle}$  is the infinite set of *all* polynomials with the same common complex roots as  $F$ . In 1.8, we compute a finite reduced Gröbner basis  $G$  with respect to  $\omega$  of that radical. If radical computation is not available on the software side, then the algorithm remains correct with a Gröbner basis of the ideal  $\langle F \rangle$  instead of the radical ideal, but might miss some simplifications.

**Algorithm 8** SimplifyReducedSystems

**Input:** A list  $[(M_0, T_1, R_1), \dots, (M_{\ell-1}, T_\ell, R_\ell)]$ , the output of Algorithm 5, with entries corresponding to (21)

**Output:** A list  $[(M'_0, T'_1, R_1), \dots, (M'_{\ell-1}, T'_\ell, R_\ell)]$ ;  $M'_{k-1}$  describes the same manifold as  $M_{k-1}$  in a canonical form; the system  $T'_k$  is equivalent to  $T_k$  modulo  $M'_{k-1}$ , its right hand sides are in a canonical normal form modulo  $M'_{k-1}$ , possibly with fewer different differential variables than  $T_k$

```

1: for  $k := 1$  to  $\ell$  do
2:    $z_k := \{x \mid x' = g \in T_k\}$ 
3: end for
4:  $y := \{x \mid x' = 0 \in R_\ell\}$ 
5:  $\omega :=$  a block term order with  $z_1 \gg \dots \gg z_\ell \gg y$ 
6: for  $k := 1$  to  $\ell$  do
7:    $F := [f \mid f = 0 \in M_{k-1}]$ 
8:    $G := \text{GroebnerBasis}(\text{Radical}(F), \omega)$ 
9:    $M'_{k-1} := [g = 0 \mid g \in G]$ 
10:   $T'_k := []$ 
11:  for  $x' = g$  in  $T_k$  do
12:     $T'_k := T'_k \circ [x' = h]$  where  $g \xrightarrow*_G h$  and  $h$  is irreducible mod  $G$ 
13:  end for
14: end for
15: return  $[(M'_0, T'_1, R_1), \dots, (M'_{\ell-1}, T'_\ell, R_\ell)]$ 

```

In 1.9, the polynomials in  $G$  equivalently replace the left hand side polynomials of the equations in  $M_{k-1}$ . In 1.12, reduction with respect to  $\omega$ , which comes with heuristic elimination of variables, applies once more to the reduction results  $h$  obtained from right hand sides  $g$  of differential equations in  $T_k$ . Since  $G$  is a Gröbner basis, the reduction in 1.11–13 furthermore produces unique normal forms with the following property: if two polynomials  $g_1, g_2$  coincide on the manifold  $\mathcal{M}_{k-1}$  defined by  $M_{k-1}$ , then they reduce to the same normal form  $h$ . In particular, if  $g_1$  vanishes on  $\mathcal{M}_{k-1}$ , then it reduces to 0. We call the output of Algorithm 8 *simplified reduced systems*.

**5 Back-Transformation of Reduced Systems**

Let  $\ell \in \{2, \dots, m\}$  and  $k \in \{1, \dots, \ell\}$ . Recall that a triplet  $(M_{k-1}, T_k, R_k)$  obtained from Algorithm 5 describes a reduced system according to (21):

$$F_{k-1}(z, 0) = 0, \quad \frac{dz_k}{d\tau} = \delta^{b_k} \widehat{f}_k(z, 0), \quad \frac{dz_{k+1}}{d\tau} = \dots = \frac{dz_m}{d\tau} = 0.$$

A corresponding simplified system  $(M'_{k-1}, T'_k, R_k)$  can be obtained from Algorithm 8 via an equivalence transformation on the set of equations  $M_{k-1}$  and further equivalence transformations modulo  $M_{k-1}$  on the right hand sides of the differential equations in  $T_k$ , while the left hand sides of those differential equations remain untouched. It is not hard to see that for both these outputs scaling can be reversed using the substitution

$$\sigma = [x_1 \leftarrow y_1/\varepsilon^{d_1}, \dots, x_n \leftarrow y_n/\varepsilon^{d_n}] \circ [\tau \leftarrow \varepsilon^\mu t] \circ [\delta \leftarrow \varepsilon^{1/q}] \circ [\varepsilon \leftarrow \varepsilon_*]$$

obtained with Algorithm 1.

Using names  $M_{k-1}, T_k, R_k$  as in the unsimplified system, this yields a raw back-transformation  $(M_{k-1}\sigma, T_k\sigma, R_k\sigma)$ . We define  $M_{k-1}^* = M_{k-1}\sigma$ . The system  $T_k\sigma$  can be written as

$$\frac{dy_j}{d\varepsilon_*^{\mu+d_j} t} = \varepsilon_*^{b_k/q} (\widehat{f}_k(z, 0)_{j\sigma}), \quad x_j \in z_k.$$

We multiply by  $\varepsilon_*^{\mu+d_j}$  in order to arrive at differential equations in  $\frac{dy_j}{dt}$ . Furthermore, recall that the explicit factor  $\delta^{b_k}$  in the original  $T_k$  corresponds to a time scale  $\delta^{b_k} \tau$ . The corresponding time scale in  $t$  is given by  $(\delta^{b_k} \tau)\sigma = \varepsilon_*^{b_k/q+\mu} t$ ,

which we make explicit by equivalently rewriting  $T_k\sigma$  as  $T_k^*$  as follows:

$$\frac{dy_j}{dt} = \varepsilon_*^{b_k/q+\mu} (\varepsilon_*^{d_j} \widehat{f}_k(z, 0)_j \sigma), \quad x_j \in z_k.$$

Similarly,  $R_k\sigma$  can be rewritten as  $R_k^*$  as follows:

$$\frac{dy_j}{dt} = 0, \quad x_j \in z_{k+1} \cup \dots \cup z_m.$$

Recall that  $M_{k-1}$  describes a manifold  $\mathcal{M}_{k-1}$  over the positive first orthant  $\mathcal{U}$ , which is defined by inequalities  $U = \{x_1 > 0, \dots, x_n > 0\}$  not explicit in  $M_{k-1}$ . Following our construction above, this translates into  $U\sigma = \{y_1/\varepsilon_*^{d_1} > 0, \dots, y_n/\varepsilon_*^{d_n} > 0\}$ , which describes again the positive first orthant. Furthermore, the nestedness (19) is preserved:

$$\mathcal{U} = \mathcal{M}_0^* \supseteq \mathcal{M}_1^* \supseteq \dots \supseteq \mathcal{M}_{\ell-1}^*.$$

Finally, the system  $(T_k^*, R_k^*)$  defines differential equations on  $\mathcal{M}_{k-1}^*$ .

We call  $(M_0^*, T_1^*, R_1^*), \dots, (M_{\ell-1}^*, T_\ell^*, R_\ell^*)$  *back-transformed reduced systems*. In terms of the definitions after (9) in Sect. 2.1 we have reverted the scaling but not the partitioning and not the truncating. Furthermore, we have preserved all information obtained with the computation of the reduced systems in Sect. 3, where we keep the time scale factors explicit, and with their algebraic simplification in Sect. 4.

Our back-transformation is realized in Algorithm 9. In 1.3 we compute the time scale factor  $\varepsilon_*^{(b_k/q)+\mu}$  for  $T_k^*$  as described above, and in 1.6 we compute its co-factor  $\varepsilon_*^{d_j} f\sigma$  as  $(y_j f/x_j)\sigma$ .

---

#### Algorithm 9 TransformBack

---

**Input:** 1.  $[(M_0, T_1, R_1), \dots, (M_{\ell-1}, T_\ell, R_\ell)]$ , the output of either Algorithm 5 or Algorithm 8;  
2.  $\sigma$ , the output of Algorithm 1

**Output:** A list  $[(M_0^*, T_1^*, R_1^*), \dots, (M_{\ell-1}^*, T_\ell^*, R_\ell^*)]$ .

```

1: for  $k := 1$  to  $\ell$  do
2:    $M_{k-1}^* := M_{k-1}\sigma$ 
3:    $v := ((\delta^{b_k} \tau)\sigma)/t$ , extracting  $\delta^{b_k}$  from  $T_k$ 
4:    $T_k^* := []$ 
5:   for all  $x'_j = \delta^{b_k} f_j \in T_k$  do
6:      $h := (y_j f_j/x_j)\sigma$ 
7:      $T_k^* := T_k^* \circ [y'_j = vh]$ 
8:   end for
9:    $R_k^* := [y'_j = 0 \mid x'_j = 0 \in R_k]$ 
10: end for
11: return  $[(M_0^*, T_1^*, R_1^*), \dots, (M_{\ell-1}^*, T_\ell^*, R_\ell^*)]$ 

```

## 6 The Big Picture

Let us discuss what has been gained in  $(M_0^*, T_1^*, R_1^*), \dots, (M_{\ell-1}^*, T_\ell^*, R_\ell^*)$  for our original system  $S$  in (1). We are faced with a discrepancy. On the one hand, we fix  $\varepsilon = \varepsilon_*$ . On the other hand, the requirement that  $\bar{\delta}$  be sufficiently small in Theorem 1 entails that  $\varepsilon$  be sufficiently small. It is of crucial importance whether invariant manifolds of (3), which do exist for sufficiently small  $\varepsilon$ , persist at  $\varepsilon = \varepsilon_*$ . We are not aware of any algorithmic results addressing

this question. In particular, singular perturbation theory is typically concerned with asymptotic results, which are not helpful here.

In case of persistence, there exist nested invariant manifolds  $\mathcal{N}_{k-1}^*$  which are Hausdorff-close to  $\mathcal{M}_{k-1}^*$  for system (1). Moreover, the differential equations  $T_k^*$  associated with  $\mathcal{M}_{k-1}^*$  correspond to the  $k$ th level in a hierarchy of time scales and approximate the flow on  $\mathcal{N}_{k-1}^*$ . We have achieved a decomposition of (1) into  $\ell$  systems of smaller dimension. At the very least, one obtains a well-educated guess about possible candidates for invariant manifolds and reductions. For the investigation of those candidates one may check the  $\mathcal{N}_k^*$  for approximate invariance using, e.g., numerical methods, or by applying criteria proposed in [56].

Algorithm 10 provides a wrapper combining all our algorithms to decompose input systems like (1) into several time scales. The underlying tropicalization is not made explicit, and the result is presented on the original scale. Figure 1 explains the functional dependencies and principal data flow between our algorithms graphically.

---

### Algorithm 10 TropicalMultiReduce

---

**Input:** 1. A list  $S = [y_1 = f_1, \dots, y_n = f_n]$  of autonomous first-order ordinary differential equations where  $f_1, \dots, f_n \in \mathbb{Q}[y_1, \dots, y_n]$ ;  
 2.  $\varepsilon_* \in (0, 1) \cap \mathbb{Q}$ ;  
 3.  $p \in \mathbb{N} \setminus \{0\}$

**Output:** A list  $[(M_0^*, T_1^*, R_1^*), \dots, (M_{\ell-1}^*, T_\ell^*, R_\ell^*)]$  of triplets where  $\ell \in \{2, \dots, m\}$ , or the empty list. For  $k \in \{1, \dots, \ell\}$ ,  $M_{k-1}^*$  is a list of real constraints defining  $\mathcal{M}_{k-1}^* \subseteq \mathbb{R}^n$ ;  $T_k^*$  is a list of differential equations;  $R_k^*$  is a list of trivial differential equations  $\dot{y} = 0$  for all differential variables from  $T_{k+1}^*, \dots, T_m^*$ .

The relevance of the output in terms of the input is discussed in Sect. 5.

1:  $\text{TropicalC}_{S, \varepsilon_*, p} := \text{curry}(\text{TropicalC}, S, \varepsilon_*, p)$   
 2:  $\text{TropicalD}_{S, \varepsilon_*, p} := \text{curry}(\text{TropicalD}, S, \varepsilon_*, p)$   
 3:  $T, P, \sigma := \text{ScaleAndTruncate}(S, \text{TropicalC}_{S, \varepsilon_*, p}, \text{TropicalD}_{S, \varepsilon_*, p}, \varepsilon_*)$   
 4:  $\Sigma := \text{ComputeReducedSystems}(T, P)$   
 5:  $\Sigma' := \text{SimplifyReducedSystems}(\Sigma)$   
 6:  $\Sigma^* := \text{TransformBack}(\Sigma', \sigma)$   
 7: **return**  $\Sigma^*$

$\text{TropicalC}_{S, \varepsilon_*, p}$  is a binary function  
 $\text{TropicalD}_{S, \varepsilon_*, p}$  is a constant function  
 $= [(M_0, T_1, R_1), \dots, (M_{\ell-1}, T_\ell, R_\ell)]$   
 $= [(M'_0, T'_1, R_1), \dots, (M'_{\ell-1}, T'_\ell, R_\ell)]$   
 $= [(M_0^*, T_1^*, R_1^*), \dots, (M_{\ell-1}^*, T_\ell^*, R_\ell^*)]$

---

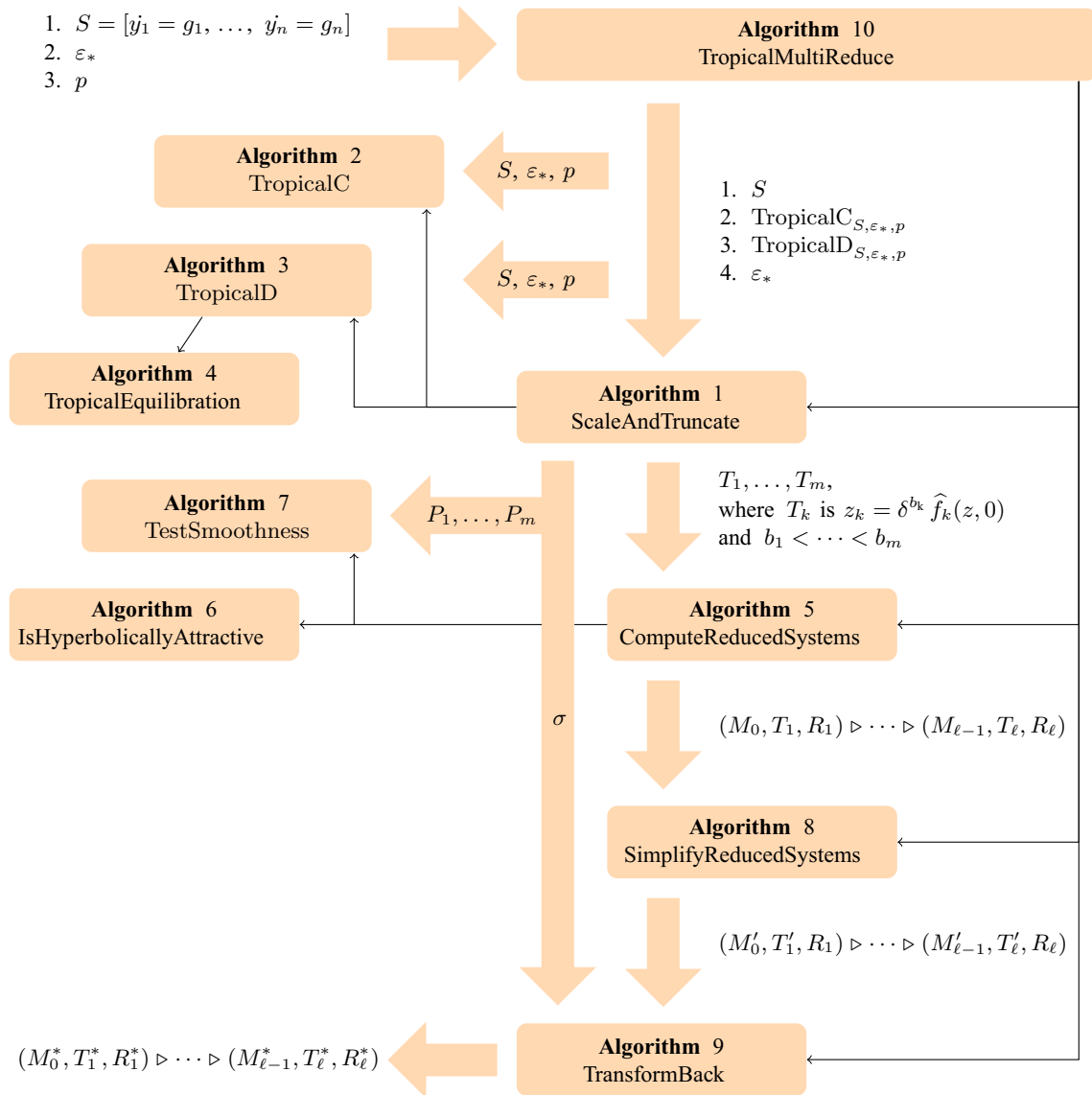
## 7 Computational Examples

Based on our explicit algorithms in the present work, we have developed two independent software prototypes realizing all methods described here. The first one is in Python using SymPy [49] for symbolic computation, pySMT [28] as an interface to the SMT solver MathSAT5 [13], and SMTcut for the computation of tropical equilibrations [48]. The second one is a Maple package, which makes use of Maple's built-in SMTLIB package [26] for using the SMT solver Z3 [51]. For our computations here we have used our Python code. Computation results are identical with both systems, and timings are similar. We have conducted our computations on a standard desktop computer with a 3.3 GHz 6-core Intel 5820K CPU and 16 GB of main memory. Computation times listed are CPU times.

In the next section, we discuss in detail the computations for one specific biological input system from the BioModels database, a repository of mathematical models of biological processes [57]. The subsequent sections showcase several further such examples in a more concise style. The focus here is on biological results. For an illustration of our algorithms, we discuss in Appendix A examples where reduction stops at  $\ell < m$  for various reasons.

### 7.1 An Epidemic Model of Avian Influenza H5N6

We consider BioModel 716, which is related to the transmission dynamics of subtype H5N6 of the influenza A virus in the Philippines in August 2017 [42]. The model specifies four species: *Susceptible birds* ( $S_b$ ), *Infected birds*



**Fig. 1** Functional dependencies (thin arrows) and principal data flow (thick arrows) between our algorithms

(I\_b), *Susceptible humans* (S\_h), and *Infected humans* (I\_a), the concentrations of which over time we map to differential variables  $y_1, \dots, y_4$ , respectively. The input system is given by

$$S = \left[ \begin{aligned} \frac{d}{dt} y_1 &= -\frac{9137}{2635182} y_1 y_2 - \frac{1}{730} y_1 + \frac{412}{73}, \\ \frac{d}{dt} y_2 &= \frac{9137}{2635182} y_1 y_2 - \frac{4652377}{961841430} y_2, \\ \frac{d}{dt} y_3 &= -\frac{1}{6159375000} y_2 y_3 - \frac{1}{25258} y_3 + \frac{40758549}{3650000}, \\ \frac{d}{dt} y_4 &= \frac{1}{6159375000} y_2 y_3 - \frac{112500173}{2841525000000} y_4 \end{aligned} \right].$$

We choose  $\varepsilon_* = \frac{1}{5}$ ,  $p = 1$ , and Algorithm 3 non-deterministically selects  $D = (-1, -4, -7, -3)$  from the tropical equilibration. Algorithm 1 then yields the following scaled and truncated system with three time scales:

$$\begin{aligned} T_1 &= \left[ \frac{d}{d\tau} x_1 = 1 \cdot \left( -\frac{5710625}{2635182} x_1 x_2 + \frac{412}{365} \right) \right], \\ T_2 &= \left[ \frac{d}{d\tau} x_2 = \delta^3 \cdot \left( \frac{5710625}{2635182} x_1 x_2 - \frac{116309425}{192368286} x_2 \right) \right], \\ T_3 &= \left[ \frac{d}{d\tau} x_3 = \delta^6 \cdot \left( -\frac{15625}{25258} x_3 + \frac{40758549}{18250000} \right), \right. \\ &\quad \left. \frac{d}{d\tau} x_4 = \delta^6 \cdot \left( \frac{15625}{15768} x_2 x_3 - \frac{112500173}{181857600} x_4 \right) \right]. \end{aligned}$$

Notice that the lexicographic order of the differential variables is coincidence. From this input, Algorithm 5 produces the following reduced systems:

$$\begin{aligned} M_0 &= [ ], & T_1 &= \left[ \frac{d}{d\tau} x_1 = 1 \cdot \left( -\frac{5710625}{2635182} x_1 x_2 + \frac{412}{365} \right) \right], \\ & & R_1 &= \left[ \frac{d}{d\tau} x_2 = 0, \right. \\ & & &\quad \frac{d}{d\tau} x_3 = 0, \\ & & &\quad \left. \frac{d}{d\tau} x_4 = 0 \right], \\ M_1 &= [2084378125 x_1 x_2 - 1085694984 = 0], & T_2 &= \left[ \frac{d}{d\tau} x_2 = \delta^3 \cdot \left( \frac{5710625}{2635182} x_1 x_2 - \frac{116309425}{192368286} x_2 \right) \right], \\ & & R_2 &= \left[ \frac{d}{d\tau} x_3 = 0, \right. \\ & & &\quad \left. \frac{d}{d\tau} x_4 = 0 \right], \\ M_2 &= [2084378125 x_1 x_2 - 1085694984 = 0, \\ &\quad 16675025 x_1 x_2 - 4652377 x_2 = 0], & T_3 &= \left[ \frac{d}{d\tau} x_3 = \delta^6 \cdot \left( -\frac{15625}{25258} x_3 + \frac{40758549}{18250000} \right), \right. \\ & & &\quad \left. \frac{d}{d\tau} x_4 = \delta^6 \cdot \left( \frac{15625}{15768} x_2 x_3 - \frac{112500173}{181857600} x_4 \right) \right], \\ & & R_3 &= [ ]. \end{aligned}$$

In that course, Algorithm 6 confirms hyperbolic attractivity according to Sect. 3.2 for all three scaled systems. Furthermore, Algorithm 7 applies the sufficient smoothness test from Sect. 3.3 with

$$\ell = 3, \quad b_1 = 3, \quad b_2 = 3, \quad P_1 = 1 \cdot (-\delta^4 \cdot \frac{125}{146} x_1), \quad P_2 = \delta^6 \cdot (-\delta^4 \cdot \frac{15625}{15768} x_2 x_3).$$

This yields  $E = \{4\}$ , where 4 cannot be expressed as an integer multiple of 3. Thus the test fails, which causes a warning in Algorithm 5.

Algebraic simplification through Algorithm 8 yields the simplified reduced systems

$$\begin{aligned} M'_0 &= [ ], & T'_1 &= \left[ \frac{d}{d\tau} x_1 = 1 \cdot \left( -\frac{5710625}{2635182} x_1 x_2 + \frac{412}{365} \right) \right], \\ & & R'_1 &= \left[ \frac{d}{d\tau} x_2 = 0, \right. \\ & & &\quad \frac{d}{d\tau} x_3 = 0, \\ & & &\quad \left. \frac{d}{d\tau} x_4 = 0 \right], \\ M'_1 &= [x_1 x_2 = \frac{1085694984}{2084378125}], & T'_2 &= \left[ \frac{d}{d\tau} x_2 = \delta^3 \cdot \left( -\frac{116309425}{192368286} x_2 + \frac{412}{365} \right) \right], \\ & & R'_2 &= \left[ \frac{d}{d\tau} x_3 = 0, \right. \\ & & &\quad \left. \frac{d}{d\tau} x_4 = 0 \right], \\ M'_2 &= [x_1 = \frac{4652377}{16675025}], & T'_3 &= \left[ \frac{d}{d\tau} x_3 = \delta^6 \cdot \left( -\frac{15625}{25258} x_3 + \frac{40758549}{18250000} \right) \right], \end{aligned}$$

$$x_2 = \frac{1085694984}{581547125}], \quad \frac{d}{dt}x_4 = \delta^6 \cdot \left( \frac{1884887125}{1018870563}x_3 - \frac{112500173}{181857600}x_4 \right)],$$

$$R'_3 = [ ].$$

Notice that our implementations conveniently rewrite equational constraints as monomial equations with numerical right hand sides when possible. This supports readability but is not essential for the simplifications applied here, which are based on Gröbner basis theory. Comparing  $T'_2$  with  $T_2$ , we see that the equation for  $x_1x_2$  in  $M'_1$  is plugged in. Similarly,  $M_2$  is simplified to  $M'_2$ , which is in turn used to reduce  $T_3$  to  $T'_3$ .

The back-transformed reduced systems as computed by Algorithm 9 read as follows:

$$M_0^* = [ ], \quad T_1^* = \left[ \frac{d}{dt}y_1 = 1 \cdot \left( -\frac{9137}{2635182}y_1y_2 + \frac{412}{73} \right) \right],$$

$$R_1^* = \left[ \frac{d}{dt}y_2 = 0, \right. \\ \left. \frac{d}{dt}y_3 = 0, \right. \\ \left. \frac{d}{dt}y_4 = 0 \right],$$

$$M_1^* = [y_1y_2 = \frac{1085694984}{667001}], \quad T_2^* = \left[ \frac{d}{dt}y_2 = \frac{1}{125} \cdot \left( -\frac{116309425}{192368286}y_2 + \frac{51500}{73} \right) \right],$$

$$R_2^* = \left[ \frac{d}{dt}y_3 = 0, \right. \\ \left. \frac{d}{dt}y_4 = 0 \right],$$

$$M_2^* = \left[ y_1 = \frac{4652377}{3335005}, \right. \\ \left. y_2 = \frac{5428474920}{4652377} \right], \quad T_3^* = \left[ \frac{d}{dt}y_3 = \frac{1}{15625} \cdot \left( -\frac{15625}{25258}y_3 + \frac{203792745}{1168} \right), \right. \\ \left. \frac{d}{dt}y_4 = \frac{1}{15625} \cdot \left( \frac{15079097}{5094352815}y_3 - \frac{112500173}{181857600}y_4 \right) \right],$$

$$R_3^* = [ ].$$

We compare  $T_1^*, \dots, T_3^*$  to the input system  $S$ : In the equation for  $\frac{d}{dt}y_1$ , the monomial in  $y_1$  is identified as a higher order term with respect to  $\delta$  and discarded by Algorithm 1. In the equation for  $\frac{d}{dt}y_2$ , the monomial in  $y_1y_2$  has been Gröbner-reduced to a constant modulo the defining equation in  $M'_1$ . Similarly, the equation for  $\frac{d}{dt}y_3$  loses its monomial in  $y_2y_3$  by truncation of higher order terms, and in the equation for  $\frac{d}{dt}y_4$ , the monomial in  $y_2y_3$  is Gröbner-reduced to a monomial in  $y_3$ .

Notice the explicit constant factors on the right hand sides of the differential equations in  $T_1^*, \dots, T_3^*$ . They originate from factors  $\delta^{b_k}$  in the respective scaled systems  $T_1, \dots, T_3$ , corresponding to (8). They are left explicit to make the time scale of the differential equations apparent. We see that the system  $T_2^* \circ R_2^*$  is 125 times slower than  $T_1^* \circ R_1^*$ , and  $T_3^* \circ R_3^*$  is another 125 times slower.

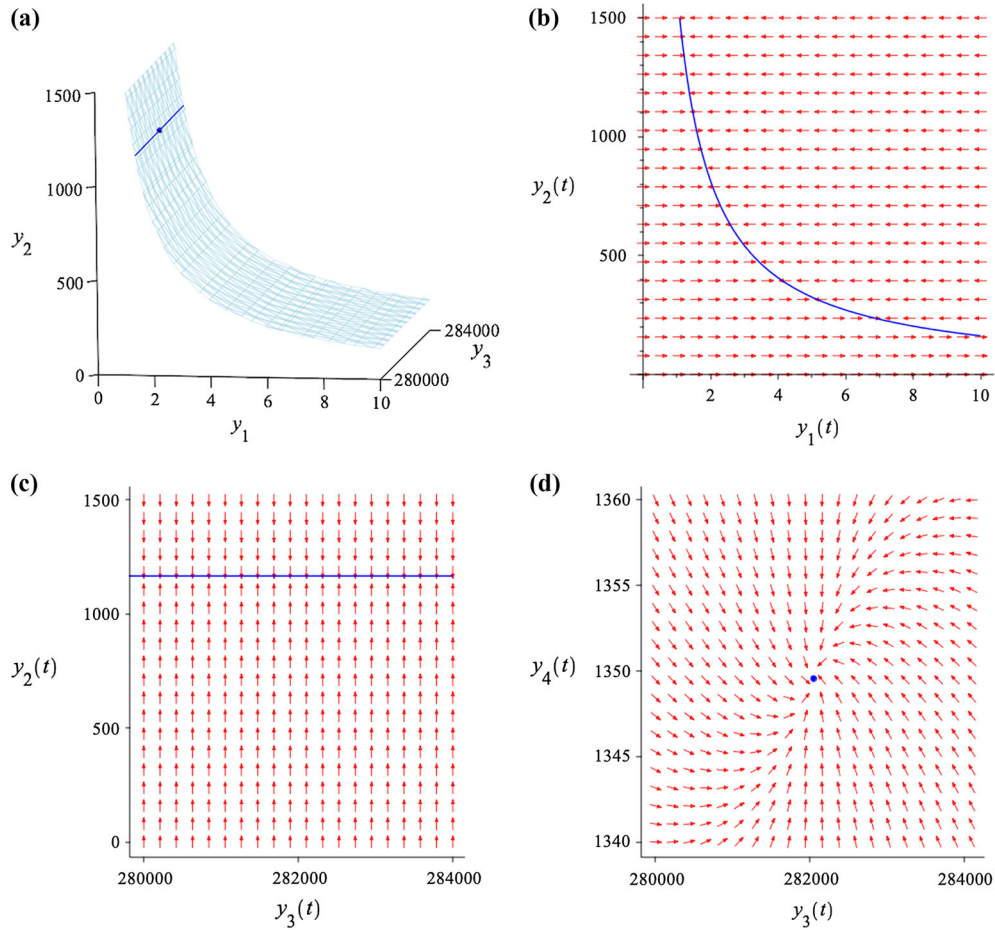
Figure 2 visualizes the direction fields of  $T_1^* \circ R_1^*, \dots, T_3^* \circ R_3^*$  on their respective manifolds  $\mathcal{M}_0^*, \dots, \mathcal{M}_2^*$  along with their respective critical manifolds  $\mathcal{M}_1^*, \dots, \mathcal{M}_3^*$ , where  $\mathcal{M}_3^*$  can be derived from  $\mathcal{M}_2^*$  by additionally equating the vector field of  $T_3^* \circ R_3^*$  to zero:

$$M_3^* = \left[ y_1 = \frac{4652377}{3335005}, \quad y_2 = \frac{5428474920}{4652377}, \quad y_3 = \frac{7051228977}{25000}, \quad y_4 = \frac{441466240042010928888}{327120760850763125} \right].$$

This list  $M_3^*$  does not explicitly occur in the output. However, its preimage  $M_3$  is constructed in Algorithm 5 and justifies the presence of  $(M_2, T_3, R_3)$  in the output there. The total computation time was 0.906 s.

This multiple-time scale reduction of the bird flu model emphasizes a cascade of successive relaxations of model variables. First, the population of susceptible birds relaxes. This relaxation is illustrated in Fig. 2(b). Then, the population of infected birds relaxes as shown in Fig. 2(c). Finally, the populations of susceptible and infected humans relax to a stable steady state as shown in Fig. 2(d), following a reduced dynamics described by  $T_3^*$ .





**Fig. 2** Critical manifolds and direction fields of our reductions of BioModel 716. (a) The surface is the critical manifold  $\mathcal{M}_1^* \subseteq \mathcal{M}_0^* = \mathcal{U}$  projected from  $\mathbb{R}^4$  into real  $(y_1, y_2, y_3)$ -space. The line located at  $(y_1, y_2) \approx (1.4, 1166.8)$  is the critical submanifold  $\mathcal{M}_2^* \subseteq \mathcal{M}_1^*$ . The dot located at  $(y_1, y_2, y_3) \approx (1.4, 1166.8, 282049.2)$  is the critical submanifold  $\mathcal{M}_3^* \subseteq \mathcal{M}_2^*$ . Both  $\mathcal{M}_1^*$  and  $\mathcal{M}_2^*$  extend to  $\pm\infty$  in both  $y_3$  and  $y_4$  direction, and  $\mathcal{M}_3^*$  is located near  $(1.4, 1166.8, 282049.2, 1349.6)$ . (b) The direction field of  $T_1^* \circ R_1^*$  on  $\mathcal{M}_0^* = \mathcal{U}$  projected from  $\mathbb{R}^4$  into real  $(y_1, y_2)$ -space. The curve is the critical submanifold  $\mathcal{M}_1^* \subseteq \mathcal{M}_0^*$ . (c) The direction field of  $T_2^* \circ R_2^*$  on  $\mathcal{M}_1^*$  projected from  $\mathbb{R}^4$  into real  $(y_3, y_2)$ -space. The line is the critical submanifold  $\mathcal{M}_2^* \subseteq \mathcal{M}_1^*$ . The system here is slower than the one in (b) by a factor of 125. (d) The direction field of  $T_3^* \circ R_3^*$  on  $\mathcal{M}_2^*$  projected from  $\mathbb{R}^4$  into real  $(y_3, y_4)$ -space. The dot is the critical submanifold  $\mathcal{M}_3^* \subseteq \mathcal{M}_2^*$ . The system here is slower than the one in (c) by another factor of 125

## 7.2 TGF- $\beta$ Pathway

BioModel 101 is a simple representation of the TGF- $\beta$  signaling pathway that plays a central role in tissue homeostasis and morphogenesis, as well as in numerous diseases such as fibrosis and cancer [76]. Concentrations over time of species *Receptor 1* (RI), *Receptor 2* (RII), *Ligand receptor complex-plasma membrane* (LRIRII), *Ligand receptor complex-endosome* (LRIRII\_endo), *Receptor 1 endosome* (RI\_endo), and *Receptor 2 endosome* (RII\_endo), are mapped to differential variables  $y_1, \dots, y_6$ , respectively. The original BioModel 101 has a change of ligand concentration at time  $t = 2500$ . For our computation here, we ignore this discrete event. The input system is given by

$$S = \left[ \begin{aligned} \frac{d}{dt} y_1 &= -\frac{1}{100} y_1 y_2 - \frac{90277}{250000} y_1 + \frac{33333}{1000000} y_4 + \frac{33333}{1000000} y_5 + 8, \\ \frac{d}{dt} y_2 &= -\frac{1}{100} y_1 y_2 - \frac{90277}{250000} y_2 + \frac{33333}{1000000} y_4 + \frac{33333}{1000000} y_6 + 4, \end{aligned} \right.$$

$$\begin{aligned}\frac{d}{dt}y_3 &= \frac{1}{100}y_1y_2 - \frac{152777}{250000}y_3, \\ \frac{d}{dt}y_4 &= \frac{33333}{100000}y_3 - \frac{33333}{1000000}y_4, \\ \frac{d}{dt}y_5 &= \frac{33333}{100000}y_1 - \frac{33333}{1000000}y_5, \\ \frac{d}{dt}y_6 &= \frac{33333}{100000}y_2 - \frac{33333}{1000000}y_6.\end{aligned}$$

We choose  $\varepsilon_* = \frac{1}{5}$ ,  $p = 1$ , and select  $D = (0, -4, -1, -2, -1, -5)$  from the tropical equilibrium. Our back-transformed reduced systems read as follows:

$$\begin{aligned}M_0^* &= [\ ], & T_1^* &= \left[ \frac{d}{dt}y_1 = 5 \cdot \left( -\frac{1}{500}y_1y_2 + \frac{8}{5} \right) \right], \\ & & R_1^* &= \left[ \frac{d}{dt}y_2 = 0, \quad \frac{d}{dt}y_3 = 0, \quad \frac{d}{dt}y_4 = 0, \quad \frac{d}{dt}y_5 = 0, \right. \\ & & & \quad \left. \frac{d}{dt}y_6 = 0 \right], \\ M_1^* &= [y_1y_2 = 800], & T_2^* &= \left[ \frac{d}{dt}y_3 = 1 \cdot \left( -\frac{152777}{250000}y_3 + 8 \right) \right], \\ & & R_2^* &= \left[ \frac{d}{dt}y_2 = 0, \quad \frac{d}{dt}y_4 = 0, \quad \frac{d}{dt}y_5 = 0, \quad \frac{d}{dt}y_6 = 0 \right], \\ M_2^* &= [y_1y_2 = 800, \\ & \quad y_3 = \frac{2000000}{152777}], & T_3^* &= \left[ \frac{d}{dt}y_2 = \frac{1}{5} \cdot \left( -\frac{90277}{50000}y_2 + \frac{33333}{200000}y_6 \right) \right], \\ & & R_3^* &= \left[ \frac{d}{dt}y_4 = 0, \quad \frac{d}{dt}y_5 = 0, \quad \frac{d}{dt}y_6 = 0 \right].\end{aligned}$$

The total computation time was 0.906 s.

The multiple-time scale reduction of the TGF- $\beta$  model emphasizes a cascade of successive relaxations of concentrations of different species. First, the concentration of receptor 1 relaxes rapidly. Then follows the membrane complex, and, even slower, the relaxation of receptor 2.

### 7.3 Caspase Activation Pathway

BioModel 102 is a quantitative kinetic model that examines the intrinsic pathway of caspase activation that is essential for apoptosis induction by various stimuli including cytotoxic stress [43]. Species concentrations over time are mapped to differential variables  $y_1, \dots, y_{13}$  as described in Table 1.

The input system is given by

$$\begin{aligned}S &= \left[ \begin{aligned}\frac{d}{dt}y_1 &= -\frac{1}{500}y_1y_2 - \frac{1}{500}y_1y_3 - \frac{1}{500}y_1y_{10} - \frac{1}{500}y_1y_{11} - \frac{1}{1000}y_1 + \frac{1}{10}y_5 + \frac{1}{10}y_6 + \frac{1}{10}y_{12} \\ &+ \frac{1}{10}y_{13} + \frac{1}{50}, \\ \frac{d}{dt}y_2 &= -\frac{1}{500}y_1y_2 - \frac{1}{1000}y_2y_4 - \frac{1}{5000}y_2y_8 - \frac{1}{1000}y_2 + \frac{1}{1000}y_3 + \frac{1}{10}y_6 + \frac{1}{50}, \\ \frac{d}{dt}y_3 &= -\frac{1}{500}y_1y_3 + \frac{1}{1000}y_2y_4 - \frac{1}{500}y_3 + \frac{1}{10}y_5, \\ \frac{d}{dt}y_4 &= -\frac{1}{1000}y_2y_4 + \frac{1}{1000}y_3 - \frac{1}{1000}y_4y_6 - \frac{3}{1000}y_4y_8 - \frac{1}{1000}y_4y_{11} - \frac{1}{1000}y_4y_{12} - \frac{1}{1000}y_4 \\ &+ \frac{1}{1000}y_5 + \frac{1}{1000}y_9 + \frac{1}{1000}y_{10} + \frac{1}{1000}y_{13} + \frac{1}{25}, \\ \frac{d}{dt}y_5 &= \frac{1}{500}y_1y_3 + \frac{1}{1000}y_4y_6 - \frac{51}{500}y_5, \\ \frac{d}{dt}y_6 &= \frac{1}{500}y_1y_2 - \frac{1}{1000}y_4y_6 + \frac{1}{1000}y_5 - \frac{1}{5000}y_6y_8 - \frac{101}{1000}y_6, \\ \frac{d}{dt}y_7 &= -\frac{1}{200000}y_2y_7 - \frac{7}{20000}y_6y_7 - \frac{1}{20000}y_7y_{11} - \frac{7}{2000}y_7y_{12} - \frac{1}{1000}y_7 + \frac{1}{5}, \\ \frac{d}{dt}y_8 &= \frac{1}{200000}y_2y_7 - \frac{3}{1000}y_4y_8 + \frac{7}{20000}y_6y_7 + \frac{1}{20000}y_7y_{11} + \frac{7}{2000}y_7y_{12} - \frac{1}{1000}y_8 + \frac{1}{1000}y_9, \\ \frac{d}{dt}y_9 &= \frac{3}{1000}y_4y_8 - \frac{1}{500}y_9, \\ \frac{d}{dt}y_{10} &= -\frac{1}{500}y_1y_{10} + \frac{1}{1000}y_4y_{11} - \frac{1}{500}y_{10} + \frac{1}{10}y_{13},\end{aligned}\right.\end{aligned}$$

**Table 1** Mapping of species concentrations to differential variables for BioModel 102

Species	Species variable	Differential variable
APAF-1	A	$y_1$
Caspase 9	C9	$y_2$
Caspase 9-XIAP complex	C9X	$y_3$
XIAP	X	$y_4$
APAF-1-Caspase 9-XIAP complex	AC9X	$y_5$
APAF-1-Caspase 9 complex	AC9	$y_6$
Caspase 3	C3	$y_7$
Caspase 3 cleaved	C3_star	$y_8$
Caspase 3 cleaved-XIAP complex	C3_starX	$y_9$
Caspase 9 cleaved-XIAP complex	C9_starX	$y_{10}$
Caspase 9 cleaved	C9_star	$y_{11}$
APAF-1-Caspase 9 cleaved complex	AC9_star	$y_{12}$
APAF-1-Caspase 9 cleaved-XIAP complex	AC9_starX	$y_{13}$

$$\begin{aligned} \frac{d}{dt}y_{11} &= -\frac{1}{500}y_1y_{11} + \frac{1}{5000}y_2y_8 - \frac{1}{1000}y_4y_{11} + \frac{1}{1000}y_{10} - \frac{1}{1000}y_{11} + \frac{1}{10}y_{12}, \\ \frac{d}{dt}y_{12} &= \frac{1}{500}y_1y_{11} - \frac{1}{1000}y_4y_{12} + \frac{1}{5000}y_6y_8 - \frac{101}{1000}y_{12} + \frac{1}{1000}y_{13}, \\ \frac{d}{dt}y_{13} &= \frac{1}{500}y_1y_{10} + \frac{1}{1000}y_4y_{12} - \frac{51}{500}y_{13}. \end{aligned}$$

We choose  $\varepsilon_* = \frac{1}{2}$ ,  $p = 1$ , and select  $D = (-4, 2, 3, 5, 5, 4, -6, -8, -4, -2, -2, 0, 0)$  from the tropical equilibration. Our back-transformed reduced systems read as follows:

$$\begin{aligned} M_0^* &= [\ ], & T_1^* &= \left[ \frac{d}{dt}y_4 = 1 \cdot \left( -\frac{3}{1000}y_4y_8 + \frac{1}{25} \right) \right], \\ & & R_1^* &= \left[ \frac{d}{dt}y_5 = 0, \quad \frac{d}{dt}y_6 = 0, \quad \frac{d}{dt}y_{12} = 0, \quad \frac{d}{dt}y_{13} = 0, \right. \\ & & & \quad \frac{d}{dt}y_2 = 0, \quad \frac{d}{dt}y_3 = 0, \quad \frac{d}{dt}y_{10} = 0, \quad \frac{d}{dt}y_{11} = 0, \\ & & & \quad \left. \frac{d}{dt}y_1 = 0, \quad \frac{d}{dt}y_7 = 0, \quad \frac{d}{dt}y_9 = 0, \quad \frac{d}{dt}y_8 = 0 \right], \\ M_1^* &= \left[ y_4y_8 = \frac{40}{3} \right], & T_2^* &= \left[ \frac{d}{dt}y_5 = \frac{1}{8} \cdot \left( \frac{2}{125}y_1y_3 - \frac{102}{125}y_5 \right), \right. \\ & & & \quad \frac{d}{dt}y_6 = \frac{1}{8} \cdot \left( \frac{2}{125}y_1y_2 - \frac{101}{125}y_6 \right), \\ & & & \quad \frac{d}{dt}y_{12} = \frac{1}{8} \cdot \left( \frac{2}{125}y_1y_{11} - \frac{101}{125}y_{12} \right), \\ & & & \quad \left. \frac{d}{dt}y_{13} = \frac{1}{8} \cdot \left( \frac{2}{125}y_1y_{10} - \frac{102}{125}y_{13} \right) \right], \\ & & R_2^* &= \left[ \frac{d}{dt}y_2 = 0, \quad \frac{d}{dt}y_3 = 0, \quad \frac{d}{dt}y_{10} = 0, \quad \frac{d}{dt}y_{11} = 0, \right. \\ & & & \quad \left. \frac{d}{dt}y_1 = 0, \quad \frac{d}{dt}y_7 = 0, \quad \frac{d}{dt}y_9 = 0, \quad \frac{d}{dt}y_8 = 0 \right], \\ M_2^* &= \left[ y_4y_8 = \frac{40}{3}, \right. \\ & & & \quad y_1y_3 - 51y_5 = 0, \\ & & & \quad 2y_1y_2 - 101y_6 = 0, \\ & & & \quad 2y_1y_{11} - 101y_{12} = 0, \\ & & & \quad \left. y_1y_{10} - 51y_{13} = 0 \right], \\ & & T_3^* &= \left[ \frac{d}{dt}y_2 = \frac{1}{16} \cdot \left( -\frac{2}{625}y_2y_8 + \frac{8}{25} \right) \right], \\ & & R_3^* &= \left[ \frac{d}{dt}y_3 = 0, \quad \frac{d}{dt}y_{10} = 0, \quad \frac{d}{dt}y_{11} = 0, \quad \frac{d}{dt}y_1 = 0, \right. \\ & & & \quad \left. \frac{d}{dt}y_7 = 0, \quad \frac{d}{dt}y_9 = 0, \quad \frac{d}{dt}y_8 = 0 \right]. \end{aligned}$$

The total computation time was 8.547 s, of which Algorithm 4 took 6.188 s.

The multiple-time scale reduction of the caspase activation model emphasizes a cascade of successive relaxations. First, the inhibitor of apoptosis XIAP binds rapidly to the cleaved caspase. Then, the four APAF complexes are formed. Finally, the Caspase 9 is recruited to the apoptosome.

#### 7.4 Avian Influenza Bird-to-Human Transmission

BioModel 709 describes bird-to-human transmission of different strains of avian influenza A viruses, such as H5N1 and H7N9 [47]. Species concentrations over time of *Susceptible avians* ( $S_a$ ), *Infected avians* ( $I_a$ ), *Susceptible humans* ( $S_h$ ), *Infected humans* ( $I_h$ ), and *Recovered humans* ( $R_h$ ) are mapped to differential variables  $y_1, \dots, y_5$ , respectively. The input system is given by

$$S = \left[ \begin{aligned} \frac{d}{dt} y_1 &= -\frac{1}{8000000000} y_1^3 + \frac{127}{20000000} y_1^2 - \frac{9}{500000000} y_1 y_2 - \frac{1}{200} y_1, \\ \frac{d}{dt} y_2 &= \frac{9}{500000000} y_1 y_2 - \frac{37123}{50000000} y_2, \\ \frac{d}{dt} y_3 &= -\frac{3}{500000000} y_2 y_3 - \frac{391}{10000000} y_3 + 30, \\ \frac{d}{dt} y_4 &= \frac{3}{500000000} y_2 y_3 - \frac{4445391}{10000000} y_4, \\ \frac{d}{dt} y_5 &= \frac{1}{10} y_4 - \frac{391}{10000000} y_5 \end{aligned} \right].$$

We choose  $\varepsilon_* = \frac{1}{5}$ ,  $p = 1$ , and select  $D = (-7, 0, -8, 3, -2)$  from the tropical equilibration. Our back-transformed reduced systems read as follows:

$$\begin{aligned} M_0^* &= [ ], & T_1^* &= \left[ \frac{d}{dt} y_1 = 1 \cdot \left( -\frac{1}{8000000000} y_1^3 + \frac{127}{20000000} y_1^2 \right) \right], \\ M_1^* &= [y_1^3 - 50800 y_1^2 = 0], & R_1^* &= \left[ \frac{d}{dt} y_4 = 0, \quad \frac{d}{dt} y_2 = 0, \quad \frac{d}{dt} y_3 = 0, \quad \frac{d}{dt} y_5 = 0 \right], \\ & & T_2^* &= \left[ \frac{d}{dt} y_4 = \frac{1}{5} \cdot \left( \frac{3}{100000000} y_2 y_3 - \frac{4445391}{2000000} y_4 \right) \right], \\ & & R_2^* &= \left[ \frac{d}{dt} y_2 = 0, \quad \frac{d}{dt} y_3 = 0, \quad \frac{d}{dt} y_5 = 0 \right]. \end{aligned}$$

The total computation time was 0.578 s.

The multiple-time scale reduction of this avian influenza model emphasizes a cascade of successive relaxations of different model variables. First, the susceptible bird population relaxes rapidly. The reduced equation  $T_1$  and manifold  $M_1$  suggest that the bird population dynamics is of the Allee type and evolves toward the stable extinct state. It follows the relaxation of infected human population that also evolves toward the extinct state, the end of the epidemics.

## 8 Some Remarks on Complexity

A detailed complexity analysis of our approach is beyond the scope of this article. We collect some remarks on the asymptotic worst-case complexity of various computational steps required by our algorithms:

- (i) The existential decision problem over real closed fields, for which we use SMT solving over QF\_NRA in Algorithm 6;
- (ii) Linear programming, for which we used SMT solving over QF\_LRA in Algorithm 3;
- (iii) Integer linear programming, for which we used SMT solving over QF\_LIA in Algorithm 7.

Problem (i) is in single exponential time [32]. Problem (ii) is in polynomial [38] time. Problem (iii) is NP-complete; a proof can be found in [65, Theorem 18.1], where it is essentially attributed to [16]. With all these problems, the dominant complexity parameter is the number of variables, which in our context corresponds to the number  $n$  of

differential variables. Biological models impose reasonable upper bounds on  $n$ , which corresponds to the number of species occurring there. When  $n$  is bounded, problem (i) becomes polynomial [32]. The same holds for problem (iii); a proof based on Lenstra's algorithm [44] can be found in [65, Corollary 18.7a].

When characterizing the decision in 1.2 of Algorithm 3 as linear programming in (ii) above, we tacitly assume that it is not applied to the tropical equilibration  $\Pi$  as a whole but to each contained polyhedron independently. The number of contained polyhedra is in turn exponential in  $n$ , in the worst case. The motivation for the computation of a disjunctive normal form in Algorithm 4 is to support the choice of a suitable point  $(d_1, \dots, d_n)$  in Algorithm 3 by providing information on the geometry of the tropical equilibration as a union of polyhedra. It is possible to omit this and accordingly drop the computation of the disjunctive normal form in 1.23 of Algorithm 4 altogether. In that case, solving in 1.2 of Algorithm 3 is applied to the formula  $\bigwedge \bigvee \bigwedge P$  instead of  $\Pi$ . This is a linear decision problem over ordered fields, which is NP-complete [29], and solutions can be found in single exponential time [79].

Efficient implementations do not necessarily use theoretically optimal algorithms. At the time of writing, SMT solving and corresponding decision procedures are a very active field of research and thus subject to constant change. We deliberately refrain from going into details about the current state of the art at this point.

Beyond the examples and computation times discussed here, we currently have no systematic empirical data that would allow substantial statements on the practical performance and limits of our implementations.

## 9 Concluding Remarks

We provided a symbolic method for automated model reduction of biological networks described by ordinary differential equations with multiple time scales. Our method is applicable to systems with two time scales or more, superseding traditional slow-fast reduction methods that can cope with only two time scales. We also proposed, for the first time, the automatic verification of hyperbolicity conditions required for the validity of the reduction. Our theoretical framework is accompanied by rigorous algorithms and prototypical implementations, which we successfully applied to real-world problems from the BioModels database [57].

We would like to list some open points and possible extensions of our research here. Our reduction algorithm is based on a fixed scaling (8) leading to a fixed ordering of the time scales of different variables. In our reduction scheme, different variables relax hierarchically, first the fastest ones, then the second fastest, and finally the slowest ones, which justifies our geometric picture of nested invariant manifolds. However, there are situations, e.g. in models of relaxation oscillations, when the ordering of time scales changes with time: variables that were fast can become slow at a later time, and vice versa. In order to cope with such situations, one would like to use different scalings for different time segments. One attempt to implement such a procedure has been provided in [68].

Although our proposed method identifies the full hierarchy of time scales, the subsequent reduction may stop early in this hierarchy when hyperbolic attractivity is not satisfied at some stage. One possible reason is the presence of conservation laws, also known as first integrals, at the given reduction stage. Such conservation laws force an eigenvalue zero for the Jacobian. A theorem by Schneider and Wilhelm [64] can be employed to reduce such a setting to the hyperbolically attractive case. As for the behavior of first integrals when proceeding to the reduced system, see the discussion of the non-standard case in [41] for two time scales; an extension to multiple time scales should be straightforward. Work in progress is concerned with the introduction of new slow variables, one for each independent conservation law of the fast subsystem. This is applied to networks with multiple time scales and approximate linear and polynomial conservation laws.

More generally, it is of interest to consider cases when hyperbolic attractivity fails but hyperbolicity still holds: In such cases, Cardin and Teixeira show there still exist invariant manifolds [11]. Testing for hyperbolicity is more involved than testing for hyperbolic attractivity, but in theory it is well understood, and there exists an algorithmic approach due to Routh [27]. In the case of hyperbolicity, but not attractivity, the ensuing global dynamics may be quite interesting; for instance slow-fast cycles may appear.

Concerning differentiability requirements, we check in Sect. 3.3 for smoothness of the full system. However, Fenichel's results, and in principle also those by Cardin and Teixeira, require only sufficient finite differentiability.

Therefore, given a differential equation system and a scaling, invariant manifolds and corresponding reduced systems exist for  $C^p$  functions with fixed  $p < \infty$ . Going through the details will involve intricate analysis that is left to future work.

In the introduction we sketched a Michaelis–Menten system abstracting from the known numerical values for the reaction rate constants  $k_1, k_{-1}, k_2$ . It would be indeed interesting to work on such parametric data. In the presence of parameters, one would consider effective quantifier elimination over real closed fields [15, 20, 39, 69, 70, 80] as a generalization of SMT solving. Robust implementations are freely available [9, 19] and well supported. They have been successfully applied to problems in chemical reaction network theory during the past decade [7, 8, 21–23, 33, 67, 71, 72, 78]. Such a generalization is not quite straightforward. With the tropical scaling in Sect. 2.2, Algorithm 2 would introduce logarithms of polynomials in the parametric coefficients, which is not compatible with the logic framework used here. Similar tropicalization methods, which are unfortunately not compatible with our abstract view on scaling in Sect. 2.1, require only logarithms of individual parametric coefficients [63]. Such a more special form would allow the use of abstraction in the logic engine.

From a point of view of user-oriented software, it would be most desirable to develop automatic strategies for determining good values for  $\varepsilon_*$  and for choices of  $(d_1, \dots, d_n)$  from the tropical equilibration in Algorithm 3.

**Acknowledgements** We are grateful to the anonymous referees for their thorough reviews and numerous helpful comments. This work has been supported by the interdisciplinary bilateral project ANR-17-CE40-0036 and DFG-391322026 SYMBIONT [5, 6]. The SYMBIONT project owes much to the interdisciplinary perspective, the scientific vision, and the enthusiasm of our colleague Andreas Weber, who unexpectedly passed away in March 2020. Andreas was also a major proponent of and contributor to tropical methods for the reduction of reaction networks, which are continued and extended in the present work. We miss him as an extraordinary scientist and person.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix A. Further Computational Examples

Recall that in Sect. 7 we have discussed computations for several systems from the BioModels database [57]. While the focus there was on biological results, we discuss here examples where reduction stops at  $\ell < m$  for various reasons. Again, we have conducted our computations on a standard desktop computer with a 3.3 GHz 6-core Intel 5820K CPU and 16 GB of main memory. Computation times listed are CPU times.

### A.1. Hypertoxicity of a Painkiller

BioModel 609 describes the metabolism and the related hepatotoxicity of acetaminophen, a pain killer [60]. The species concentrations over time of *Sulphate PAPS*, *GSH*, *NAPQI*, *Paracetamol APAP*, and *Protein adducts* are mapped to differential variables  $y_1, \dots, y_5$ , respectively. The input system is given by

$$S = \left[ \begin{aligned} \frac{d}{dt} y_1 &= -2260000000000000 y_1 y_4 - 2y_1 + \frac{53}{2000000000000000}, \\ \frac{d}{dt} y_2 &= -160000000000000000 y_2 y_3 - 2y_2 + \frac{687}{50000000000000000}, \\ \frac{d}{dt} y_3 &= -160000000000000000 y_2 y_3 - \frac{220063}{2000} y_3 + \frac{63}{200} y_4, \end{aligned} \right.$$

**Table 2** Mapping of species concentrations to differential variables for BioModel 726

Species	Species variable	Differential variable
Susceptible dogs	S_d	y1
Exposed dogs	E_d	y2
Infectious dogs	I_d	y3
Recovered dogs	R_d	y4
Susceptible humans	S_h	y5
Exposed humans	E_h	y6
Infectious humans	I_h	y7
Recovered humans	R_h	y8

$$\begin{aligned} \frac{d}{dt}y_4 &= -22600000000000y_1y_4 + \frac{63}{2000}y_3 - \frac{661}{200}y_4, \\ \frac{d}{dt}y_5 &= 110y_3]. \end{aligned}$$

Since there is only one monomial on the right hand side of the equation for  $\frac{d}{dt}y_5$ , equilibration is impossible. This causes Algorithm 4 to return in l.23 a disjunctive normal form  $\Pi$  equivalent to “false,” which describes the empty set. Hence Algorithm 3 returns  $\perp$ , and Algorithm 1 returns the empty list. The total computation time was 0.006 s.

### A.2. Transmission Dynamics of Rabies

BioModel 726 examines the transmission dynamics of rabies between dogs and humans [61]. Species concentrations over time are mapped to differential variables  $y_1, \dots, y_8$  as described in Table 2.

The input system is given by

$$\begin{aligned} S = \left[ \begin{aligned} \frac{d}{dt}y_1 &= -\frac{79}{500000000}y_1y_3 - \frac{17}{100}y_1 + \frac{18}{5}y_2 + y_4 + 3000000, \\ \frac{d}{dt}y_2 &= \frac{79}{500000000}y_1y_3 - \frac{617}{100}y_2, \\ \frac{d}{dt}y_3 &= \frac{12}{5}y_2 - \frac{27}{25}y_3, \\ \frac{d}{dt}y_4 &= \frac{9}{100}y_1 + \frac{9}{100}y_2 - \frac{27}{25}y_4, \\ \frac{d}{dt}y_5 &= -\frac{229}{100000000000000}y_3y_5 - \frac{3}{1000}y_5 + \frac{18}{5}y_6 + y_8 + 15400000, \\ \frac{d}{dt}y_6 &= \frac{229}{100000000000000}y_3y_5 - \frac{6543}{1000}y_6, \\ \frac{d}{dt}y_7 &= \frac{12}{5}y_6 - \frac{1343}{1000}y_7, \\ \frac{d}{dt}y_8 &= \frac{27}{50}y_6 - \frac{1003}{1000}y_8]. \end{aligned} \right. \end{aligned}$$

We choose  $\varepsilon_* = \frac{1}{5}$ ,  $p = 1$ , and Algorithm 3 selects  $D = (-10, -10, -11, -9, -14, -7, -8, -7)$  from the tropical equilibration. Algorithm 1 then yields the following scaled and truncated system with three time scales:

$$\begin{aligned} T_1 &= \left[ \begin{aligned} \frac{d}{d\tau}x_1 &= 1 \cdot \left( -\frac{395}{256}x_1x_3 + \frac{18}{25}x_2 \right), \\ \frac{d}{d\tau}x_2 &= 1 \cdot \left( \frac{395}{256}x_1x_3 - \frac{617}{500}x_2 \right), \\ \frac{d}{d\tau}x_6 &= 1 \cdot \left( \frac{28625}{16384}x_3x_5 - \frac{6543}{5000}x_6 \right), \end{aligned} \right] \\ T_2 &= \left[ \frac{d}{d\tau}x_3 = \delta \cdot \left( \frac{12}{25}x_2 - \frac{27}{25}x_3 \right), \right. \end{aligned}$$

$$\begin{aligned} \frac{d}{d\tau}x_4 &= \delta \cdot \left( \frac{9}{20}x_1 + \frac{9}{20}x_2 - \frac{27}{25}x_4 \right), \\ \frac{d}{d\tau}x_7 &= \delta \cdot \left( \frac{12}{25}x_6 - \frac{1343}{1000}x_7 \right), \\ \frac{d}{d\tau}x_8 &= \delta \cdot \left( \frac{27}{50}x_6 - \frac{1003}{1000}x_8 \right), \\ T_3 &= \left[ \frac{d}{d\tau}x_5 = \delta^5 \cdot \left( \frac{4928}{3125} - \frac{15}{8}x_5 \right) \right]. \end{aligned}$$

Equating the right hand sides  $F_1$  of the differential equations in  $T_1$  to zero equivalently yields

$$M_1 = [-9875x_1x_3 + 4608x_2 = 0, \quad 49375x_1x_3 - 39488x_2 = 0, \quad 17890625x_3x_5 - 13400064x_6 = 0].$$

In 1.1 of Algorithm 6,  $U \circ M_1$  is tested for satisfiability. This fails, which means that the corresponding manifold  $\mathcal{M}_1$  is empty over the positive first orthant. Consequently, Algorithms 5, 8, and 9 return empty lists. The total computation time was 0.921 s.

### A.3. Negative Feedback Loop Between Tumor Suppressor and Oncogene

BioModel 156 describes the dynamics of a negative feedback loop between the tumor suppressor protein p53 and the oncogene protein Mdm2 in human cells [30]. The species concentrations over time for *P53* ( $x$ ), *Mdm2* ( $y$ ), and *Precursor Mdm2* ( $y_0$ ) are mapped to differential variables  $y_1, \dots, y_3$ , respectively. The input system is given by

$$\begin{aligned} S &= \left[ \frac{d}{d\tau}y_1 = -\frac{37}{10}y_1y_2 + 2y_1, \right. \\ &\quad \frac{d}{d\tau}y_2 = -\frac{9}{10}y_2 + \frac{11}{10}y_3, \\ &\quad \left. \frac{d}{d\tau}y_3 = \frac{3}{2}y_1 - \frac{11}{10}y_3 \right]. \end{aligned}$$

We choose  $\varepsilon_* = \frac{1}{2}$ ,  $p = 1$ , and select  $D = (2, 1, 1)$ . Algorithm 1 then yields the following scaled and truncated system with two time scales:

$$\begin{aligned} T_1 &= \left[ \frac{d}{d\tau}x_1 = 1 \cdot \left( -\frac{37}{40}x_1x_2 + x_1 \right) \right], \\ T_2 &= \left[ \frac{d}{d\tau}x_2 = \delta \cdot \left( -\frac{9}{10}x_2 + \frac{11}{10}x_3 \right), \right. \\ &\quad \left. \frac{d}{d\tau}x_3 = \delta \cdot \left( \frac{3}{4}x_1 - \frac{11}{10}x_3 \right) \right]. \end{aligned}$$

Analogously to the previous section we obtain  $M_1 = [-37x_1x_2 + 40x_1 = 0]$ , for which we find  $\mathcal{M}_1$  to be non-empty over the positive first orthant in 1.1 of Algorithm 6. However in 1.20, the test for hyperbolic attractivity fails with  $M_1$  and the Hurwitz conditions

$$\Gamma = \left\{ \frac{37}{40}x_2 - 1 > 0 \right\},$$

so that “false” is returned. Therefore, Algorithm 5 breaks the for-loop in 1.10 with  $k = 1$  and returns the empty list in 1.19. Obviously, the simplified and back-translated systems are empty lists as well. The total computation time was 0.453 s.

### A.4. CD4 T-Cells in the Spread of HIV

BioModel 663 describes how CD4 T-cells can influence the spread of the HIV infection [81]. Species concentrations over time for *Infected T-cells* ( $x$ ), *Uninfected T-cells* ( $y$ ), and *Free viruses* ( $v$ ) are mapped to variables  $y_1, \dots, y_3$ , respectively. The input system is given by

$$S = \left[ \frac{d}{d\tau}y_1 = -\frac{1}{10}y_1^2y_3 - \frac{1}{10}y_1y_2y_3 + \frac{4}{5}y_1y_3 - \frac{1}{10}y_1, \right.$$



$$\begin{aligned} \frac{d}{dt}y_2 &= -\frac{1}{10}y_1y_2y_3 + \frac{1}{5}y_1y_3 - \frac{1}{10}y_2^2y_3 + y_2y_3 - \frac{1}{5}y_2, \\ \frac{d}{dt}y_3 &= y_2 - \frac{1}{2}y_3. \end{aligned}$$

We choose  $\varepsilon_* = \frac{1}{2}$ ,  $p = 5$ , and  $D = (1, 4, 3)$ . The choice of  $p = 5$  causes fractional powers of numbers in the scaled and truncated system

$$\begin{aligned} T_1 &= \left[ \frac{d}{d\tau}x_3 = 1 \cdot (x_2 - x_3) \right], \\ T_2 &= \left[ \frac{d}{d\tau}x_2 = \delta^7 \cdot \left( \frac{4}{5}\sqrt[5]{4}x_1x_3 - \frac{4}{5}\sqrt[5]{4}x_2 \right) \right], \\ T_3 &= \left[ \frac{d}{d\tau}x_1 = \delta^{12} \cdot \left( \frac{4}{5}\sqrt[5]{4}x_1x_3 - \frac{4}{5}\sqrt[5]{4}x_1 \right) \right]. \end{aligned}$$

However, such input is not accepted with the SMT logic `QF_NRA` used in Algorithm 6. As discussed in Sect. 3.2, we catch the corresponding error from the SMT solver, convert to floats, and restart, which solves the problem.

Similarly to the previous example, the Hurwitz test in 1.20 of Algorithm 6 succeeds for  $k = 1$  but fails for  $k = 2$  in Algorithm 5. Since there are fewer than two reduced systems, we return the empty list. Consequently, the list of simplified reduced systems and the corresponding list of back-transformed systems are empty as well. The total computation time was 0.390 s.

## References

1. Barrett, C., Conway, C.L., Deters, M., Hadarean, L., Jovanović, D., King, T., Reynolds, A., Tinelli, C.: CVC4. In: Gopalakrishnan, G., Qadeer, S. (eds.) Proc. CAV 2011. LNCS, vol. 6806, pp. 171–177. Springer (2011). [https://doi.org/10.1007/978-3-642-22110-1\\_14](https://doi.org/10.1007/978-3-642-22110-1_14)
2. Barrett, C., Fontaine, P., Tinelli, C.: The SMT-LIB standard: version 2.6. Technical report, Department of Computer Science, The University of Iowa (2017)
3. Becker, T., Weispfenning, V., Kredel, H.: Gröbner Bases, a Computational Approach to Commutative Algebra. Graduate Texts in Mathematics, vol. 141. Springer, Berlin (1993). <https://doi.org/10.1007/978-1-4612-0913-3>
4. Bogart, T., Nedergaard Jensen, A., Speyer, D., Sturmfels, B., Thomas, R.R.: Computing tropical varieties. *J. Symb. Comput.* **42**(1–2), 54–73 (2007). <https://doi.org/10.1016/j.jsc.2006.02.004>
5. Boulrier, F., Fages, F., Radulescu, O., Samal, S.S., Schuppert, A., Seiler, W.M., Sturm, T., Walcher, S., Weber, A.: The SYMBIONT project: symbolic methods for biological networks. *F1000Research* **7**(1341), 67–70 (2018). <https://doi.org/10.7490/f1000research.1115995.1>
6. Boulrier, F., Fages, F., Radulescu, O., Samal, S.S., Schuppert, A., Seiler, W.M., Sturm, T., Walcher, S., Weber, A.: The SYMBIONT project: symbolic methods for biological networks. *ACM Commun. Comput. Algebra* **52**(3), 67–70 (2018). <https://doi.org/10.1145/3313880.3313885>
7. Bradford, R., Davenport, J.H., England, M., Errami, H., Gerdt, V., Grigoriev, D., Hoyt, C., Košta, M., Radulescu, O., Sturm, T., Weber, A.: A case study on the parametric occurrence of multiple steady states. In: Burr, M. (ed.) Proc. ISSAC 2017, pp. 45–52. ACM (2017), <https://doi.org/10.1145/3087604.3087622>
8. Bradford, R., Davenport, J.H., England, M., Errami, H., Gerdt, V., Grigoriev, D., Hoyt, C., Košta, M., Radulescu, O., Sturm, T., Weber, A.: Identifying the parametric occurrence of multiple steady states for some biological networks. *J. Symb. Comput.* **98**, 84–119 (2020). <https://doi.org/10.1016/j.jsc.2019.07.008>
9. Brown, C.W.: QEPCAD B: a program for computing with semi-algebraic sets using CADs. *ACM SIGSAM Bull.* **37**(4), 97–108 (2003). <https://doi.org/10.1145/968708.968710>
10. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Doctoral dissertation, Mathematical Institute, University of Innsbruck, Innsbruck, Austria (1965)
11. Cardin, P.T., Teixeira, M.A.: Fenichel theory for multiple time scale singular perturbation problems. *SIAM J. Appl. Dyn. Syst.* **16**(3), 1425–1452 (2017). <https://doi.org/10.1137/16m1067202>
12. Cardin, P.T., Teixeira, M.A.: Corrigendum: Fenichel theory for multiple time scale singular perturbation problems. *SIAM J. Appl. Dyn. Syst.* **18**(2), 1223 (2019). <https://doi.org/10.1137/19M1241660>
13. Cimatti, A., Griggio, A., Schaafsma, B., Sebastiani, R.: The MathSAT5 SMT solver. In: Piterman, N., Smolka, S.A. (eds.) Proc. TACAS 2013. LNCS, vol. 7795, pp. 93–107. Springer (2013). [https://doi.org/10.1007/978-3-642-36742-7\\_7](https://doi.org/10.1007/978-3-642-36742-7_7)
14. Collins, G.E.: Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In: Brakhage, H. (ed.) Automata Theory and Formal Languages. 2nd GI Conference. LNCS, vol. 33, pp. 134–183. Springer (1975). [https://doi.org/10.1007/3-540-07407-4\\_17](https://doi.org/10.1007/3-540-07407-4_17)

15. Collins, G.E., Hong, H.: Partial cylindrical algebraic decomposition for quantifier elimination. *J. Symb. Comput.* **12**(3), 299–328 (1991). [https://doi.org/10.1016/S0747-7171\(08\)80152-6](https://doi.org/10.1016/S0747-7171(08)80152-6)
16. Cook, S.A.: The complexity of theorem-proving procedures. In: Proc. STOC '71, pp. 151–158. ACM Press, New York, NY (1971). <https://doi.org/10.1145/800157.805047>
17. Corzilius, F., Kremer, G., Junges, S., Schupp, S., Ábrahám, E.: SMT-RAT: an open source C++ toolbox for strategic and parallel SMT solving. In: Heule, M., Weaver, S. (eds.) Proc. SAT 2015. LNCS, vol. 9340, pp. 360–368. Springer (2015). [https://doi.org/10.1007/978-3-319-24318-4\\_26](https://doi.org/10.1007/978-3-319-24318-4_26)
18. Curry, H.B., Feys, R.: *Combinatory Logic. Studies in Logic and the Foundations of Mathematics*, vol. I. North Holland Publishing Company, Amsterdam (1958)
19. Dolzmann, A., Sturm, T.: Redlog: computer algebra meets computer logic. *ACM SIGSAM Bull.* **31**(2), 2–9 (1997). <https://doi.org/10.1145/261320.261324>
20. Dolzmann, A., Sturm, T., Weispfenning, V.: Real quantifier elimination in practice. In: Matzat, B.H., Greuel, G.-M., Hiss, G. (eds.) *Algorithmic Algebra and Number Theory*, pp. 221–247. Springer, Berlin (1998). [https://doi.org/10.1007/978-3-642-59932-3\\_11](https://doi.org/10.1007/978-3-642-59932-3_11)
21. England, M., Errami, H., Grigoriev, D., Radulescu, O., Sturm, T., Weber, A.: Symbolic versus numerical computation and visualization of parameter regions for multistationarity of biological networks. In: Gerdt, V., Koepf, W., Seiler, W., Vorozhtsov, E. (eds.) Proc. CASC 2017. LNCS, vol. 10490, pp. 93–108. Springer (2017). [https://doi.org/10.1007/978-3-319-66320-3\\_8](https://doi.org/10.1007/978-3-319-66320-3_8)
22. Errami, H., Eiswirth, M., Grigoriev, D., Seiler, W.M., Sturm, T., Weber, A.: Detection of Hopf bifurcations in chemical reaction networks using convex coordinates. *J. Comput. Phys.* **291**, 279–302 (2015). <https://doi.org/10.1016/j.jcp.2015.02.050>
23. Errami, H., Seiler, W.M., Sturm, T., Weber, A.: On Muldowney’s criteria for polynomial vector fields with constraints. In: Gerdt, V.P., Koepf, W., Mayr, E.W., Vorozhtsov, E.V. (eds.) Proc. CASC 2011. LNCS, vol. 6885, pp. 135–143. Springer (2011). [https://doi.org/10.1007/978-3-642-23568-9\\_11](https://doi.org/10.1007/978-3-642-23568-9_11)
24. Feinberg, M.: *Foundations of Chemical Reaction Network Theory. Applied Mathematical Sciences*, vol. 202. Springer, Berlin (2019). <https://doi.org/10.1007/978-3-030-03858-8>
25. Fenichel, N.: Geometric singular perturbation theory for ordinary differential equations. *J. Differ. Equ.* **31**(1), 53–98 (1979). [https://doi.org/10.1016/0022-0396\(79\)90152-9](https://doi.org/10.1016/0022-0396(79)90152-9)
26. Forrest, S.: Integration of SMT-LIB support into Maple. In: England, M., Ganesh, V. (eds.) Proc. Satisfiability Checking and Symbolic Computation 2017. CEUR Workshop Proceedings, vol. 1974. CEUR-WS, Kaiserslautern, Germany (2017)
27. Gantmacher, F.R.: *The Theory of Matrices. Vol. 2. Translation from the Russian by K. A. Hirsch. Reprint of the 1959 translation edition.* AMS Chelsea Publishing, Providence, RI (1998)
28. Gario, M., Micheli, A.: PySMT: a solver-agnostic library for fast prototyping of SMT-based algorithms. In: SMT Workshop 2015. 13th International Workshop on Satisfiability Modulo Theories, Affiliated With the 27th International Conference on Computer Aided Verification, San Francisco, CA (2015)
29. von zur Gathen, J., Sieveking, M.: Weitere zum Erfüllungsproblem polynomial äquivalente kombinatorische Aufgaben. In: Strassen, V. (ed.) *Komplexität von Entscheidungsproblemen.* LNCS, vol. 43, chapter 4, pp. 49–71. Springer (1976). [https://doi.org/10.1007/3-540-07805-3\\_5](https://doi.org/10.1007/3-540-07805-3_5)
30. Geva-Zatorsky, N., Rosenfeld, N., Itzkovitz, S., Milo, R., Sigal, A., Dekel, E., Yarnitzky, T., Liron, Y., Polak, P., Lahav, G., Alon, U.: Oscillations and variability in the p53 system. *Mol. Syst. Biol.* **2**(1), 2006–0033 (2006). <https://doi.org/10.1038/msb4100068>
31. Goeke, A., Walcher, S., Zerz, E.: Determining “small parameters” for quasi-steady state. *J. Differ. Equ.* **259**(3), 1149–1180 (2015). <https://doi.org/10.1016/j.jde.2015.02.038>
32. Grigoriev, D.: Complexity of deciding Tarski algebra. *J. Symb. Comput.* **5**(1–2), 65–108 (1988). [https://doi.org/10.1016/S0747-7171\(88\)80006-3](https://doi.org/10.1016/S0747-7171(88)80006-3)
33. Grigoriev, D., Iosif, A., Rahkooy, H., Sturm, T., Weber, A.: Efficiently and effectively recognizing toricity of steady state varieties. *Math. Comput. Sci.* **15**(2), 199–232 (2021). <https://doi.org/10.1007/s11786-020-00479-9>
34. Heineken, F.G., Tsuchiya, H.M., Aris, R.: On the mathematical status of the pseudo-steady state hypothesis of biochemical kinetics. *Math. Biosci.* **1**(1), 95–113 (1967). [https://doi.org/10.1016/0025-5564\(67\)90029-6](https://doi.org/10.1016/0025-5564(67)90029-6)
35. Hoppensteadt, F.: On systems of ordinary differential equations with several parameters multiplying the derivatives. *J. Differ. Equ.* **5**(1), 106–116 (1969). [https://doi.org/10.1016/0022-0396\(69\)90106-5](https://doi.org/10.1016/0022-0396(69)90106-5)
36. Hurwitz, A.: Ueber die Bedingungen, unter welchen eine Gleichung nur Wurzeln mit negativen reellen Theilen besitzt. *Math. Ann.* **46**, 273–284 (1895). <https://doi.org/10.1007/BF01446812>
37. IEEE Std. 754-2019: IEEE Standard for Floating-Point Arithmetic (2019). <https://doi.org/10.1109/IEEESTD.2019.8766229>
38. Khachiyan, L.G.: A polynomial algorithm in linear programming. *Soviet Math. Dokl.* **20**(1), 191–194 (1979)
39. Košta, M.: New concepts for real quantifier elimination by virtual substitution. Doctoral dissertation. Saarland University, Germany (2016). <https://doi.org/10.22028/D291-26679>
40. Kruff, N., Walcher, S.: Coordinate-independent singular perturbation reduction for systems with three time scales. *Math. Biosci. Eng.* **16**(5), 5062–5091 (2019). <https://doi.org/10.3934/mbe.2019255>
41. Lax, C., Walcher, S.: Singular perturbations and scaling. *Discrete Cont. Dyn.-B* **25**(1), 1–29 (2020). <https://doi.org/10.3934/dcdsb.2019170>
42. Lee, H., Lao, A.: Transmission dynamics and control strategies assessment of avian influenza A (H5N6) in the Philippines. *Infect. Dis. Model.* **3**, 35–59 (2018). <https://doi.org/10.1016/j.idm.2018.03.004>
43. Legewie, S., Blüthgen, N., Herzog, H.: Mathematical modeling identifies inhibitors of apoptosis as mediators of positive feedback and bistability. *PLoS Comput. Biol.* **2**(9), e120 (2006). <https://doi.org/10.1371/journal.pcbi.0020120>

44. Lenstra Jr., H.W.: Integer programming with a fixed number of variables. *Math. Oper. Res.* **8**(4), 538–548 (1983). <https://doi.org/10.1287/moor.8.4.538>
45. Litvinov, G.L.: Maslov dequantization, idempotent and tropical mathematics: a brief introduction. *J. Math. Sci.* **140**(3), 426–444 (2007). <https://doi.org/10.1007/s10958-007-0450-5>
46. Litvinov, G.L., Sergeev, S.N.: Tropical and Idempotent Mathematics: International Workshop TROPICAL-07, Tropical and Idempotent Mathematics, August 25–30, 2007, Independent University of Moscow and Laboratory J.-V. Poncelet. *Contemporary Mathematics*, vol. 495. AMS (2009)
47. Liu, S., Ruan, S., Zhang, X.: Nonlinear dynamics of avian influenza epidemic models. *Math. Biosci.* **283**, 118–135 (2017). <https://doi.org/10.1016/j.mbs.2016.11.014>
48. Lüders, C.: Computing tropical prevarieties with satisfiability modulo theories (SMT) solvers. In: Fontaine, P., Korovin, K., Kotsireas, I.S., Rümmer, P., Tourret, S. (eds.) *Proc. SC-Square 2020, Co-Located With IJCAR 2020. CEUR Workshop Proceedings*, vol. 2752, pp. 189–203. CEUR-WS, June–July (2020)
49. Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J.K., Singh, S., Rathnayake, T., Vig, S., Granger, B.E., Muller, R.P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M.J., Terrel, A.R., Roučka, Š., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., Scopatz, A.: SymPy: symbolic computing in Python. *PeerJ Comput. Sci.* **3**, e103 (2017). <https://doi.org/10.7717/peerj-cs.103>
50. Mikhalkin, G.: Enumerative tropical algebraic geometry in  $\mathbb{R}^2$ . *J. Am. Math. Soc.* **18**(2), 313–377 (2005). <https://doi.org/10.1090/S0894-0347-05-00477-7>
51. de Moura, L., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) *Proc. TACAS 2008. LNCS*, vol. 4963, pp. 337–340. Springer (2008). [https://doi.org/10.1007/978-3-540-78800-3\\_24](https://doi.org/10.1007/978-3-540-78800-3_24)
52. Nieuwenhuis, R., Oliveras, A., Tinelli, C.: Solving SAT and SAT modulo theories: from an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T). *J. ACM* **53**(6), 937–977 (2006). <https://doi.org/10.1145/1217856.1217859>
53. Nipp, K.: An algorithmic approach for solving singularly perturbed initial value problems. In: Kirchgraber, U., Walther, H. (eds.) *Dynamics Reported*, vol. 1, pp. 173–263. John Wiley & Sons and B. G. Teubner, Hoboken (1988), [https://doi.org/10.1007/978-3-322-96656-8\\_4](https://doi.org/10.1007/978-3-322-96656-8_4)
54. Noel, V., Grigoriev, D., Vakulenko, S., Radulescu, O.: Tropical geometries and dynamics of biochemical networks application to hybrid cell cycle models. In: Feret, J., Levchenko, A. (eds.) *Proc. SASB 2011. ENTCS*, vol. 284, pp. 75–91. Elsevier, Amsterdam (2012), <https://doi.org/10.1016/j.entcs.2012.05.016>
55. Noel, V., Grigoriev, D., Vakulenko, S., Radulescu, O.: Tropicalization and tropical equilibration of chemical reactions. In: Litvinov, G.L., Sergeev, S.N. (eds.) *Tropical and Idempotent Mathematics and Applications. Contemporary Mathematics*, vol. 616, pp. 261–277. AMS, Providence (2014), <https://doi.org/10.1090/conm/616/12316>
56. Noethen, L., Walcher, S.: Quasi-steady state and nearly invariant sets. *SIAM J. Appl. Math.* **70**(4), 1341–1363 (2009). <https://doi.org/10.1137/090758180>
57. Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., Snoep, J.L., Hucka, M.: BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.* **34**(supp\_1), D689–D691 (2006). <https://doi.org/10.1093/nar/gkj092>
58. Radulescu, O., Gorban, A.N., Zinovyev, A., Noel, V.: Reduction of dynamical biochemical reactions networks in computational biology. *Front. Genet.* **3**, 131 (2012). <https://doi.org/10.3389/fgene.2012.00131>
59. Radulescu, O., Vakulenko, S., Grigoriev, D.: Model reduction of biochemical reactions networks by tropical analysis methods. *Math. Model. Nat. Phenom.* **10**(3), 124–138 (2015). <https://doi.org/10.1051/mmnp/201510310>
60. Reddyhoff, D., Ward, J., Williams, D., Regan, S., Webb, S.: Timescale analysis of a mathematical model of acetaminophen metabolism and toxicity. *J. Theor. Biol.* **386**, 132–146 (2015). <https://doi.org/10.1016/j.jtbi.2015.08.021>
61. Ruan, S.: Modeling the transmission dynamics and control of rabies in China. *Math. Biosci.* **286**, 65–93 (2017). <https://doi.org/10.1016/j.mbs.2017.02.005>
62. Samal, S.S., Grigoriev, D., Fröhlich, H., Radulescu, O.: Analysis of reaction network systems using tropical geometry. In: Gerdt, V., Koepf, W., Seiler, W., Vorozhtsov, E. (eds.) *Proc. CASC 2015. LNCS*, vol. 9301, pp. 424–439. Springer (2015), [https://doi.org/10.1007/978-3-319-24021-3\\_31](https://doi.org/10.1007/978-3-319-24021-3_31)
63. Samal, S.S., Grigoriev, D., Fröhlich, H., Weber, A., Radulescu, O.: A geometric method for model reduction of biochemical networks with polynomial rate functions. *Bull. Math. Biol.* **77**(12), 2180–2211 (2015). <https://doi.org/10.1007/s11538-015-0118-0>
64. Schneider, K.R., Wilhelm, T.: Model reduction by extended quasi-steady-state approximation. *J. Math. Biol.* **40**(5), 443–450 (2000). <https://doi.org/10.1007/s002850000026>
65. Schrijver, A.: *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester (1998). <https://doi.org/10.1002/oca.4660100108>
66. Segel, L.A., Slemrod, M.: The quasi-steady-state assumption: a case study in perturbation. *SIAM Rev.* **31**(3), 446–477 (1989). <https://doi.org/10.1137/1031091>
67. Seiler, W.M., Seif, M., Sturm, T.: A logic based approach to finding real singularities of implicit ordinary differential equations. *Math. Comput. Sci.* **15**(2), 333–352 (2021). <https://doi.org/10.1007/s11786-020-00485-x>
68. Sommer-Simpson, J., Reinitz, J., Fridlyand, L., Philipson, L., Radulescu, O.: Hybrid reductions of computational models of ion channels coupled to cellular biochemistry. In: Bartocci, E., Lio, P., Paoletti, N. (eds.) *Proc. CMSB 2016. LNCS*, vol. 9859, p. 273. Springer (2016), [https://doi.org/10.1007/978-3-319-45177-0\\_17](https://doi.org/10.1007/978-3-319-45177-0_17)

69. Sturm, T.: A survey of some methods for real quantifier elimination, decision, and satisfiability and their applications. *Math. Comput. Sci.* **11**(3–4), 483–502 (2017). <https://doi.org/10.1007/s11786-017-0319-z>
70. Sturm, T.: Thirty years of virtual substitution: Foundations, techniques, applications. In: Arreche, C. (ed.) *Proc. ISSAC 2018*, pp. 11–16. ACM (2018), <https://doi.org/10.1145/3208976.3209030>
71. Sturm, T., Weber, A.: Investigating generic methods to solve Hopf bifurcation problems in algebraic biology. In: Horimoto, K. (ed.) *Proc. Algebraic Biology 2008*. LNCS, vol. 5147, pp. 200–215. Springer (2008), [https://doi.org/10.1007/978-3-540-85101-1\\_15](https://doi.org/10.1007/978-3-540-85101-1_15)
72. Sturm, T., Weber, A., Abdel-Rahman, E.O., El Kahoui, M.: Investigating algebraic and logical algorithms to solve Hopf bifurcation problems in algebraic biology. *Math. Comput. Sci.* **2**(3), 493–515 (2009). <https://doi.org/10.1007/s11786-008-0067-1>
73. Tarski, A.: A decision method for elementary algebra and geometry. Prepared for publication by J.C.C. McKinsey. R109, August 1, 1948, Revised May 1951, Second Edition. RAND Report. RAND, Santa Monica, CA (1957)
74. Tikhonov, A.N.: Systems of differential equations containing small parameters in the derivatives. *Mat. Sb. (N. S.)* **73**(3), 575–586 (1952)
75. Valorani, M., Paolucci, S.: The G-scheme: a framework for multi-scale adaptive model reduction. *J. Comput. Phys.* **228**(13), 4665–4701 (2009). <https://doi.org/10.1016/j.jcp.2009.03.011>
76. Vilar, J.M.G., Jansen, R., Sander, C.: Signal processing in the TGF- $\beta$  superfamily ligand-receptor network. *PLoS Comput. Biol.* **2**(1), e3 (2006). <https://doi.org/10.1371/journal.pcbi.0020003>
77. Viro, O.: Dequantization of real algebraic geometry on logarithmic paper. In: Casacuberta, C., Miró-Roig, R.M., Verdera, J., Xambó-Descamps, S. (eds.) *European Congress of Mathematics, Progress in Mathematics*, vol. 201, pp. 135–146. Springer, Berlin (2001), [https://doi.org/10.1007/978-3-0348-8268-2\\_8](https://doi.org/10.1007/978-3-0348-8268-2_8)
78. Weber, A., Sturm, T., Abdel-Rahman, E.O.: Algorithmic global criteria for excluding oscillations. *Bull. Math. Biol.* **73**(4), 899–916 (2011). <https://doi.org/10.1007/s11538-010-9618-0>
79. Weispfenning, V.: The complexity of linear problems in fields. *J. Symb. Comput.* **5**(1–2), 3–27 (1988). [https://doi.org/10.1016/S0747-7171\(88\)80003-8](https://doi.org/10.1016/S0747-7171(88)80003-8)
80. Weispfenning, V.: Quantifier elimination for real algebra—the quadratic case and beyond. *Appl. Algebr. Eng. Commun.* **8**(2), 85–101 (1997). <https://doi.org/10.1007/s002000050055>
81. Wodarz, D., Hamer, D.H.: Infection dynamics in HIV-specific CD4 T cells: does a CD4 T cell boost benefit the host or the virus? *Math. Biosci.* **209**(1), 14–29 (2007). <https://doi.org/10.1016/j.mbs.2007.01.007>