



# An Algorithm for the Formal Reduction of Differential Equations as Over-Approximations

Giuseppe Squillace<sup>1(✉)</sup>, Mirco Tribastone<sup>1</sup>, Max Tschaikowski<sup>2</sup>,  
and Andrea Vandin<sup>3,4</sup>

<sup>1</sup> IMT School for Advanced Studies Lucca, Lucca, Italy

[giuseppe.squillace@imtlucca.it](mailto:giuseppe.squillace@imtlucca.it)

<sup>2</sup> Aalborg University, Aalborg, Denmark

<sup>3</sup> Sant'Anna School of Advanced Studies, Pisa, Italy

<sup>4</sup> DTU Technical University of Denmark, Lyngby, Denmark

**Abstract.** Models of complex systems often consist of state variables with structurally similar dynamics that differ in the specific values of some parameters. Examples are multi-class epidemiological models, chemical reaction networks describing multiple components (e.g., binding sites) with equivalent functional behavior, and models of electric circuits with replicated designs. In these cases, the analysis may be expensive due to the model size. Here we consider models defined as systems of polynomial ordinary differential equations (ODEs) with positive solutions. We present an algorithm to reduce the computational cost by transforming the original ODE model into one for which we can compute an appropriate over-approximation on a smaller set of state variables. The algorithm is based on the theory of differential inequalities and consists of two steps. The first step computes a differential hull, which is an ODE system providing lower and upper bounds for each state variable. The hull is constructed such that variables with structurally similar dynamics but originally different parameters may now be represented by the same lower and upper bounds. Based on this, the second step exploits already developed notions of exact model reduction for ODEs to lump such variables. The algorithm is showcased on several case studies and its results are favourably compared against CORA, a well-known tool for reachability analysis of dynamical systems.

**Keywords:** Ordinary differential equations · Model reduction · Reachability analysis

## 1 Introduction

Ordinary differential equations (ODEs) are a fundamental model to describe the behavior of dynamical systems. In many cases, they represent classes of entities governed by structurally similar laws governed by different parameters. Such heterogeneous parameters may encode different dynamical behavior of the same

underlying phenomenon, such as age- or location-dependent rates for the contagion of a disease [4], class-dependent service demands in a queuing system [3], and conformation-dependent binding affinities in protein interaction networks [13].

When there is a large degree of heterogeneity, intended as the number of classes used in the model, the analysis becomes increasingly more complex. This problem justifies the quest for reduction methods that simplify the description whilst retaining some formal relationship with the original models. Here we tackle this issue by designing an algorithm that aims to homogenize class-dependent behavior into representative equations that suitably summarize the difference in parameters. The idea builds on an earlier approach that yields so-called *differential hulls* for heterogeneous ODE systems [27]. Differential hulls provide lower- and upper-bounds on the original equations by relying on the theory of differential inequalities [23, 24, 26] which can be traced back to the seminal work of Müller [21]. Here, equations with different parameters are lower- and upper-bounded with the same differential inequality by taking appropriate minimum and maximum values across the parameters. Thus, the resulting system of differential inequalities is independent from the number of aforementioned classes and its solution give an envelope within which the original trajectories live, effectively constituting a formal over-approximation of the original model.

The main limitation of [27] is that no algorithm is given to build differential hulls; that is, the method requires a priori knowledge of the existence of “equivalent” dynamical equations up to the choice of the parameters. The main contribution of this paper is an algorithm that builds differential hulls for ODEs.

We focus on polynomial ODEs with positive solutions. This is already a general class to which other forms of nonlinearity can be reduced [19], and it essentially covers many dynamical models of systems where the state variables are physical quantities such as concentration of molecules and populations of agents. Our algorithm takes as input a tolerance  $\varepsilon$  that, informally, defines the amount of heterogeneity allowed in the model parameters. The procedure consists of two steps. First, the algorithm splits the set of parameters into blocks so long as the difference between the values of the minimum and the maximum elements in each block is less than  $\varepsilon$ . The differential hull is then built by doubling the number of variables in the system (e.g., from the original  $n$  to  $2n$ ), replacing each variable with a pair of new ones representing its upper and lower bounds. This is obtained by appropriately substituting in the equations of each pair of new variables the original parameters with either of the extremal elements in the block it belongs to. If the original system consisted of structurally similar equations with different parameters (and these parameters are at most  $\varepsilon$  away), the intended output of this first step is to have replicated equations that have the same dynamical behavior due to the consistent choice of the parameter bounds. Since the algorithm is agnostic to the form of the original model, the second step performs an automated discovery of the replicated behavior. This is done with backward differential equivalence (BDE) [5, 6, 8], a reduction algorithm that exactly lumps ODE variables that have the same solution when starting from the same initial condition. Overall, the procedure returns a reduced differential hull that still bounds the original dynamics while using fewer than  $2n$  variables.

We use case studies from engineering, biology, biochemistry, organic chemistry, and epidemiology to compare against our method against CORA [1], a state-of-the-art tool for over-approximation/reachability analysis. The comparison is justified by the fact that the proposed approach can be tied to over-approximation. Indeed, the first step of our algorithm splits the parameters into blocks where the difference between the maximum and the minimum is less than  $\varepsilon$ . Afterwards, the algorithm substitutes each parameter with the maximum or the minimum of the block it belongs to. These extremal elements of each block define the admissible parameter values that over-approximation techniques such as CORA take into account to compute the reachable sets. The investigation reveals that CORA computes in general tighter bounds than the proposed approach, but at higher time and space requirements.

*Further Related Work.* Many common over-approximation techniques rely on Lyapunov-like functions [12, 20] known from stability theory of ODEs. However, the automatic computation of Lyapunov-like functions remains a challenging task in case of nonlinearity [14]. Instead, approaches such as CORA or Flow\* approximate the nonlinear model by a multivariate polynomial or an affine system, see [2, 10] and references therein. The research on approximate quotients of ODE systems, instead, can be traced to the 1960s [17] where the authors over-approximated the dynamics of mono-molecular reaction networks. Li and Rabitz extend this approximate lumping to general CRNs [18], but an explicit error bound was not given. In a similar vein, approximate quotients in ecology have been studied from the point of view of finding a reduced ODE system whose derivatives are as close as possible (in norm) to the derivatives of the original ODE system [15]. This is also exploited in  $\varepsilon$ -BDE [9], a reduction technique that is based on a partition-refinement algorithm of BDE [6] and aims to lump ODE variables with nearby trajectories, essentially by relaxing the requirement of exact symmetry imposed by the BDE approach used in this paper. Using a case study from [9], we show that our method can provide bounds for larger differences in the model parameters than  $\varepsilon$ -BDE.

## 2 Background

*Backward Differential Equivalence.* Let us consider a polynomial ODE system composed of a set of variables  $\mathcal{V} = \{x_1, \dots, x_n\}$ . The dynamics of variable  $x_i$  is in the form  $\dot{x}_i = q_i, 1 \leq x_i \leq n$ , where  $q_i$  is a multivariate polynomial over  $\mathcal{V}$ . We say that  $q_i$  is in normal form when each monomial  $x^\alpha \equiv \prod_{x_i \in \mathcal{V}} x_i^{\alpha_{x_i}}$ , where  $\alpha \in \mathbb{N}_0^{\mathcal{V}}$  is a multi-index, appears in  $q_i$  at most once. In this way, we can define  $c(q_i, x^\alpha)$  as the coefficient of the monomial  $x^\alpha$  in a normal form polynomial  $q_i$ .

The notion of BDE [6, 9] relates variables that have the same solutions at all time points if they start from the same initial conditions. In the polynomial ODE systems that we consider, this technique makes pairwise comparisons between the coefficients of any two variables in the same equivalence class.

**Definition 1 (Backward differential equivalence (BDE)).** Fix a polynomial ODE, a partition  $\mathcal{H}$  of  $\mathcal{V}$  and write  $x_i \sim_{\mathcal{H}}^B x_j$  if all coefficients of the following polynomial are zero,

$$q_{i,j}^H := (q_i - q_j)[x_{H',1}/x_{H'}, \dots, x_{H',|H'|}/x_{H'} : H' \in \mathcal{H}]$$

i.e., when

$$\sum_{\alpha \in \mathbb{N}_0^V} |c(q_{i,j}^H, x^\alpha)| = 0. \tag{1}$$

A partition  $\mathcal{H}$  is a BDE if  $\mathcal{H} = \mathcal{V} / (\sim_{\mathcal{H}}^{B*} \cap \sim_{\mathcal{H}})$ .

Following the definition, a partition is a BDE partition if the differences between the coefficients on the same monomials are zero for any two variables in the same block.

*Differential Hulls.* We use the notation  $x \leq y$  for the vectors  $x = (x_1, \dots, x_n)$  and  $y = (y_1, \dots, y_n)$  in  $\mathbb{R}^n$  if and only if  $x_i \leq y_i$  for all  $1 \leq i \leq n$ . The strict inequality,  $x < y$ , is defined similarly. The differential hull is a vector field with  $2n$  variables that provides upper and lower bounds for the dynamics of the original ODE system defined on the set of variables  $\mathcal{V} = \{x_1, \dots, x_n\}$ .

**Definition 2 (Differential Hull [27]).** We call  $(g_{\underline{1}}, \dots, g_{\underline{n}}, g_{\overline{1}}, \dots, g_{\overline{n}}) : \mathbb{R}_{>0}^{2n} \rightarrow \mathbb{R}^{2n}$  a differential hull of the polynomial ODE system  $(q_1, \dots, q_n) : \mathbb{R}_{>0}^n \rightarrow \mathbb{R}^n$  when, for all  $1 \leq i \leq n$   $g_{\underline{i}}, g_{\overline{i}}$  are polynomials and for any  $\underline{x} \leq x \leq \overline{x}$ ,

$$\underline{x}_i = x_i \implies g_{\underline{i}}(\underline{x}, \overline{x}) \leq q_i(x) \quad \text{and} \quad x_i = \overline{x}_i \implies q_i(x) \leq g_{\overline{i}}(\underline{x}, \overline{x})$$

The previous definition is very general because the only condition a differential hull should satisfy is that it should over-approximate the dynamics of a polynomial vector field  $q$ .

**Theorem 1.** Let  $g$  be a differential hull of  $q$ . Then, if the solution of the polynomial ODE system  $(\dot{\underline{x}}, \dot{\overline{x}}) = g(\underline{x}, \overline{x})$  subject to  $0 < \underline{x}(0) \leq x(0) \leq \overline{x}(0)$  exists and is positive on  $[0; T]$ , where  $T > 0$ , then the solution of  $\dot{x} = q(x)$  exists on  $[0; T]$  as well and satisfies  $\underline{x}(t) \leq x(t) \leq \overline{x}(t)$  for all  $0 \leq t \leq T$ .

### 3 Computing Differential Hulls

Algorithm 1 takes as input a tolerance value  $\varepsilon > 0$  and a polynomial ODE system  $O$ , given by  $\dot{x}_i = q_i(x)$  with  $1 \leq i \leq n$ . Line 2 sorts all coefficients  $\{(i, \alpha, c(q_i, x^\alpha)) \in O \mid 1 \leq i \leq n, \alpha \in \mathbb{N}_0^n\}$  of  $O$  in increasing order and splits them into blocks whose members are within distance  $\varepsilon$ . More in detail, we start from the minimum parameter and add the next one in the same block until the difference between the first and last inserted is not greater than  $\varepsilon$ . Blocks are collected in the resulting partition,  $P$ . Lines 4–5 define two new equations  $\overline{x}_i$  and  $\underline{x}_i$ , respectively the lower and upper bound of  $x_i$ . In lines 6–11, the algorithm

---

**Algorithm 1.** computeDifferentialHull

---

**Require:** An ODE system  $O$ , a tolerance  $\varepsilon$ .

```

1:  $DHull = \{\}$ 
2:  $P = \text{groupParameters}(O, \varepsilon)$ 
3: for each  $x_i$  in  $O$  do
4:    $\dot{\bar{x}}_i = []$ 
5:    $\dot{\underline{x}}_i = []$ 
6:   for each monomial  $M$  in  $O$  do
7:      $\bar{M} = \text{upperBound}(M, P, x_i)$ 
8:      $\underline{M} = \text{lowerBound}(M, P, x_i)$ 
9:      $\text{append}(\dot{\bar{x}}_i, \bar{M})$ 
10:     $\text{append}(\dot{\underline{x}}_i, \underline{M})$ 
11:   end for
12:    $\text{add}(DHull, \dot{\bar{x}}_i)$ 
13:    $\text{add}(DHull, \dot{\underline{x}}_i)$ 
14: end for
15: return  $DHull$ 

```

---

considers the monomials  $M$  in equation  $x_i$ . It computes the lower and upper bound for each of them and appends these results to  $\dot{\bar{x}}$  and  $\dot{\underline{x}}$ , respectively.

The procedure to compute the upper bound is shown in Algorithm 2. It requires a monomial  $M$ , the coefficients partition  $P$  already calculated by Algorithm 1, and variable  $x_i$ . In lines 2–3, the procedure retrieves the coefficient and the variables associated with the monomial  $M$ . In lines 4–8, the algorithm substitutes the original parameter of the monomial. If the coefficient of the monomial  $p$  is positive, the computation picks the maximum parameter in the block  $p$  belongs to (line 5), otherwise the minimum (line 7). In lines 9–15, the method takes care of the variables  $x_j$ . The idea is similar. The method picks the upper or lower bound of  $x_j$  depending on the value of  $p$ . The first condition in line 10 represents the case where the variable  $x_j$  is the same variable as  $x_i$ . We are computing  $\dot{\bar{x}}_i$  and we find  $x_j$  equals to  $x_i$  in  $q_i$ , in this case, since the variable defines itself, the algorithm will pick  $\bar{x}_j$  no matter what is the value of  $p$ .

We omit the algorithm for the lower bound, called in line 9 of Algorithm 1, because it is similar to Algorithm 2. In lines 12–13 Algorithm 1 composes the new equations to the differential hull and returns it.

**Theorem 2.** *The time and space complexity of Algorithm 1 and Algorithm 2 is polynomial in the size of the ODE model.*

*Running Example.* Let us take the simple polynomial ODE system:

$$\dot{x}_1 = -k_2x_1, \quad \dot{x}_2 = k_1x_1 - k_3x_2, \quad \dot{x}_3 = k_2x_1 - k_3x_3$$

with  $k_1 = 1.0$ ,  $k_2 = 1.1$ , and  $k_3 = 1.2$  and initial conditions all equal to 1.

We now consider the application of the Algorithm 1 with a tolerance parameter  $\varepsilon = 0.2$ . In the first step, the procedure splits the parameters in a single block

**Algorithm 2.** upperBound

---

**Require:** A monomial  $M$ , the parameters partition  $P$ , variable  $\dot{x}_i$ .

```

1:  $\bar{M} = \{\}$ 
2:  $(\cdot, \cdot, p) = \text{getParameter}(M)$ 
3:  $V = \text{getVariables}(M)$ 
4: if  $p > 0$  then
5:    $\text{add}(\bar{M}, \text{getMax}(p, P))$ 
6: else
7:    $\text{add}(\bar{M}, \text{getMin}(p, P))$ 
8: end if
9: for each  $x_j$  in  $V$  do
10:  if  $x_j == x_i$  or  $p > 0$  then
11:     $\text{add}(\bar{M}, \bar{x}_j)$ 
12:  else
13:     $\text{add}(\bar{M}, \underline{x}_j)$ 
14:  end if
15: end for
16: return  $\bar{M}$ 

```

---

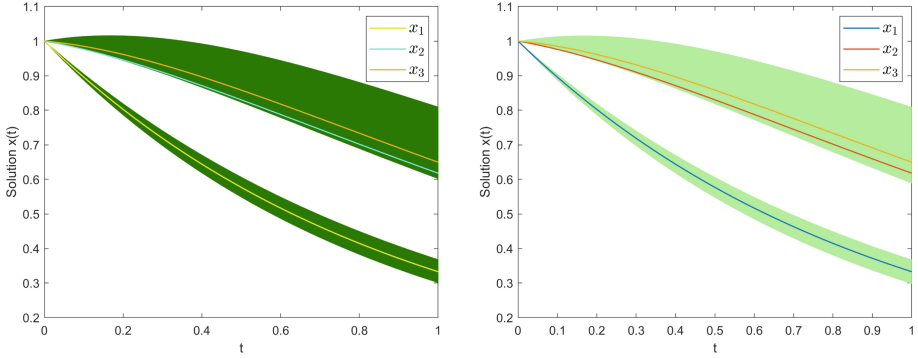
$B_1$  where the tolerance is exactly 0.2, corresponding to the difference  $k_3 - k_1$ . We now discuss the detailed process to compute the upper bound  $\bar{x}_2$ . In line 6, Algorithm 1 considers every monomial in the dynamics  $\dot{x}_2$  of ODE system  $O$ . For the first term  $k_1 x_1$ , since  $k_1$  is positive, line 5 of Algorithm 2 picks  $k_3$ , the maximum parameter for this block. Similarly, in line 11, the maximum value that  $x_1$  could assume is  $\bar{x}_1$ , which is the upper bound of  $x_1$ . In this way, the algorithm provides the first term  $k_3 \bar{x}_1$  of  $\bar{x}_2$ . The computation proceeds with the maximization of the second terms  $-k_3 x_2$ . Since  $-k_3$  is negative, the algorithm takes the parameter  $k_1$ . Moreover, we fall in the case where the condition  $x_j == x_i$  in line 10 is true; for this reason Algorithm 2 replaces  $x_2$  with  $\bar{x}_2$  rather than  $\underline{x}_2$ . Summing up all the steps, the algorithm computes the upper bound of  $\dot{x}_2$  with the equation  $\bar{x}_2 = k_3 \bar{x}_1 - k_1 \bar{x}_2$ . The lower bound is computed similarly and, for this reason, is omitted.

Overall, the differential hull for the system reads:

$$\begin{array}{lll} \dot{\underline{x}}_1 = -k_3 \underline{x}_1 & \dot{\underline{x}}_2 = k_1 \underline{x}_1 - k_3 \underline{x}_2 & \dot{\underline{x}}_3 = k_1 \underline{x}_1 - k_3 \underline{x}_3 \\ \dot{\bar{x}}_1 = -k_1 \bar{x}_1 & \dot{\bar{x}}_2 = k_3 \bar{x}_1 - k_1 \bar{x}_2 & \dot{\bar{x}}_3 = k_3 \bar{x}_1 - k_1 \bar{x}_3 \end{array}$$

In Fig. 1 (left), we plot both the dynamics of the differential hull and the original system when all initial conditions are equal to 1. Every trajectory  $x_i$  falls in a band bounded by the two equations  $\underline{x}_i$  and  $\bar{x}_i$ . Importantly, we notice that, due to the choice of initial conditions, the solutions for  $\underline{x}_2$  and  $\underline{x}_3$  coincide, and so do the solutions for  $\bar{x}_2$  and  $\bar{x}_3$ . This is due to the fact that the partition of variables  $\{\{\underline{x}_1\}, \{\bar{x}_1\}, \{\underline{x}_2, \underline{x}_3\}, \{\bar{x}_2, \bar{x}_3\}\}$  satisfies the BDE criterion in Eq. 1. This gives the following BDE-reduced differential hull where variables  $\underline{x}_2$  and  $\bar{x}_2$  are taken as the representatives of their respective blocks.

$$\dot{\underline{x}}_1 = -k_3 \underline{x}_1, \quad \dot{\bar{x}}_1 = -k_1 \bar{x}_1, \quad \dot{\underline{x}}_2 = k_1 \underline{x}_1 - k_3 \underline{x}_2, \quad \dot{\bar{x}}_2 = k_3 \bar{x}_1 - k_1 \bar{x}_2.$$



**Fig. 1.** (left) Over-approximation by means of differential hulls for the running example. (right) CORA over-approximation of the running example.

It is important to notice that the bounds computed over-approximate not only the dynamics for the parameters under study. Indeed, any set of parameters giving rise to the same differential hull will be over-approximated by the hull. Specifically, the following can be shown.

**Theorem 3.** *Let  $O$  be an ODE system over variables  $x_1, \dots, x_n$  and let  $P$  be the partition as computed by Algorithm 1 and Algorithm 2. Assume that all blocks of  $P$  have common signs (i.e., for any  $B \in P$  and  $(\cdot, \cdot, p_1), (\cdot, \cdot, p_2) \in B$ , it holds that  $p_1 \cdot p_2 \geq 0$ ). Then, an ODE system  $O'$  over  $x_1, \dots, x_n$  gives rise to the same differential hull as  $O$  when*

- $O'$  has no more monomials than  $O$ , that is, if  $(j, \beta, \cdot) \notin B$  for each  $B \in P$ , then  $c(q'_j, x^\beta) = 0$  and;
- the parameters of  $O'$  yield the same minima and maxima over partition  $P$ , i.e., for all  $(j, \beta, \cdot) \in B$  and all  $B \in P$  we have that

$$\min\{c(q_i, x^\alpha) \mid (i, \alpha, \cdot) \in B\} \leq c(q'_j, x^\beta) \leq \max\{c(q_i, x^\alpha) \mid (i, \alpha, \cdot) \in B\},$$

where  $c(q'_j, x^\beta)$  denotes the coefficient of monomial  $x^\beta$  in  $q'_j$  of  $O'$ .

*Remark 1.* The assumption on  $P$  having blocks with common signs can be always enforced by means of a prepartitioning. This being said, we wish to point out that all models considered in the evaluation section did not require a prepartitioning, i.e., Theorem 3 could be applied directly.

The foregoing result ties differential hulls to reachability analysis, where an amount of perturbation is considered among the grouped parameter. This justifies the comparison against CORA in the next section. For completeness, we next show the application of CORA to our running example.

CORA requires choosing how to represent the reachability set and the amount of perturbation in the parameters. In this case, we decided to represent the sets with the zonotopes. We set up the parameters to their average values, that is 1.1,

allowing an amount of perturbation equal to 0.1. In this way, we consider the following range of uncertainty [1.0; 1.2], that represent the set of all the possible parameters considered by the differential hull. In Fig. 1 (right) we show the bounds computed by CORA. In this example, the two techniques provide almost the same bounds. In the next section, we will present several models from different fields to compare the bounds provided by our approach and the ones by CORA. It can be noted that the two techniques provide almost identical bounds. We will see in the next section that CORA tends to give better bounds compared to our approach, while requiring significantly more time and space.

## 4 Case Studies

In this section, we consider a number of case studies. The CORA implementation was carried out in Matlab, while the BDE reductions of Algorithm 1 were performed by invoking ERODE [7].

### 4.1 SIR Model

The SIR model describes the spread of an infection in a population composed of three main actors: infected (I), susceptible (S), and the recovered individuals (R) [16]. The infected individuals are the ones that could infect the susceptibles; the recovered obtained a permanent immunization from infection because they already got the disease. The model has two types of parameters:  $\beta$ , the infection rate, and  $\gamma$ , the recovery rate. In this context, we consider the following multiclass SIR model of individuals with class-specific infection and recovery rates:

$$\dot{S}_i = \sum_{j=1}^N -S_i \beta_{i,j} I_j, \quad \dot{I}_i = -\gamma_i I_i + \sum_{j=1}^N S_i \beta_{i,j} I_j, \quad \dot{R}_i = \gamma_i I_i,$$

where the parameters  $\beta_{i,j}$  represent cross-class infection rates. For consistency across all number of classes, the parameters were chosen using the same level of heterogeneity, as follows:

$$\theta_{\text{SIR}} = \left| \max_{i,j} \beta_{i,j} - \min_{i,j} \beta_{i,j} \right| + \left| \max_i \gamma_i - \min_i \gamma_i \right| = 0.2$$

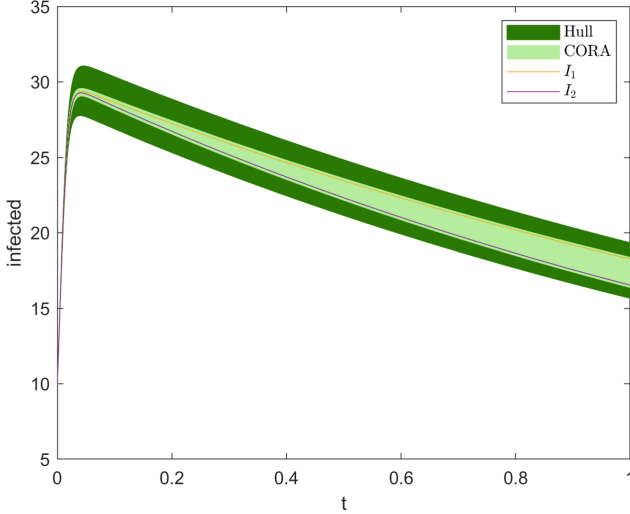
All parameter values and the initial conditions are provided in the Appendix.

We computed the differential hull running our algorithm with the tolerance  $\varepsilon$  equal to  $\theta_{\text{SIR}}$ , then we reduced it with BDE. The reduced differential hull is an SIR model where all the lower and the upper bounds for each class collapse into one, so that the reduction achieved by BDE is:  $\{\{\underline{S}_1, \dots, \underline{S}_N\}, \{\bar{S}_1, \dots, \bar{S}_N\}, \{\underline{I}_1, \dots, \underline{I}_N\}, \{\bar{I}_1, \dots, \bar{I}_N\}, \{\underline{R}_1, \dots, \underline{R}_N\}, \{\bar{R}_1, \dots, \bar{R}_N\}\}$ .

In Fig. 2, we show the comparison between CORA and differential hulls for the SIR model with two different classes; the bounds computed considering an higher number of classes are similar. CORA has tighter bounds, but it is more time consuming. Indeed, Table 1, which lists the CORA runtimes, shows a fast



increase with respect to the number of classes, issuing out of memory errors for 8. Our algorithm instead required less than 1 s in all cases. This is an expected result because, as stated in Theorem 2, the cost of the algorithm is polynomial and is based on the substitution of parameters and variables.



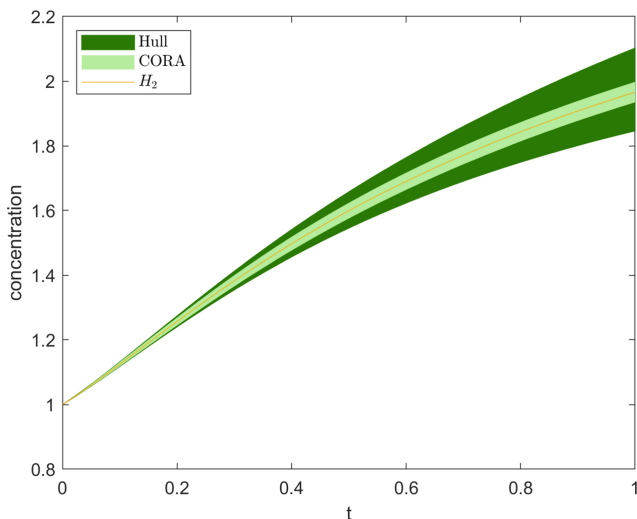
**Fig. 2.** Bounds of the infected individuals computed by our algorithm against CORA.

**Table 1.** CORA running times for the SIR model.

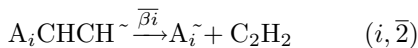
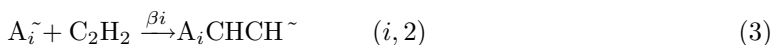
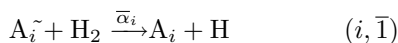
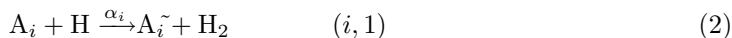
Number of classes	2	4	6	8
CORA runtime	12.98 s	43.43 s	162.96 s	Out of memory

## 4.2 Polymerization

In chemistry, polymerization is the process by which monomers react to form longer chains. We consider next the polymerization model presented in [28] which describes the formation of polycyclic aromatic hydrocarbons in flame combustion. The underlying system of polynomial ODEs is induced by the law of mass action [29]. Let us consider, for instance, the reaction  $A_i + H \xrightarrow{\alpha_i} A_i^{\sim} + H_2$ . The terms on the left side are called reagents, while those on the right are called products. An instance of each reagent is *consumed* when the reaction occurs, and one of each product is *produced*. The kinetic reaction rate is  $\alpha_i$ , instead. The reaction occurs at speed  $\alpha_i A_i H$ , where the variables denote the current concentration (the current amount) of the corresponding species. Consequently, the monomial  $\alpha_i A_i H$  will appear with negative sign in the ODEs of the reagents ( $A_i$  and  $H$ ), and with positive sign in those of the products ( $A_i^{\sim}$  and  $H_2$ ).



**Fig. 3.** Bounds of the molecule  $H_2$  computed by Algorithm 1 against CORA.



Here  $A_i\tilde{\phantom{A}}$  is an aromatic radical formed by H abstraction from  $A_i$ , and  $A_iCHCH\tilde{\phantom{A}}$  is a radical formed by adding  $C_2H_2$  to  $A_i\tilde{\phantom{A}}$ . We enumerate the reactions  $(i, 1)$  and their reverse versions  $(i, \bar{1})$ . The reverse reaction is a reaction where the products became the reagents and vice versa. Since the reaction network is infinite we restrict our analysis to truncated version of this model, where we consider the dynamics of polymers up to length  $N$  (i.e., with  $i \in \{1, \dots, N\}$ ). To do this we redirect the flux to  $A_{i+1}$ , when  $i+1 > N$  to  $A_1$  in order to mimic the fact that polymers longer than  $N$  are unstable. Similarly to the previous case study, let us define the following level of heterogeneity:

$$\theta_{\text{Poly}} = |\max_i \alpha_i - \min_i \alpha_i| + |\max_i \beta_i - \min_i \beta_i| + |\max_i \gamma_i - \min_i \gamma_i|$$

For the omitted parameters, the difference between the maximum and the minimum is zero. This keeps a level of heterogeneity equal to 0.2 for each model. For simplicity, only a part of the parameters was subject to perturbation; the respective values and the initial conditions can be found in the Appendix. We ran Algorithm 1 with  $\varepsilon = 0.2$ , obtaining the reduced differential hull through BDE. The variables are lumped according to the following partition:

**Table 2.** CORA running times of the polymerization model. Similarly to the SIR model, the running times of differential hulls were within one second.

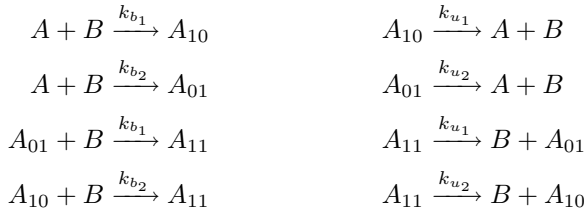
$N$	4	8	12	16
CORA runtime	69.73 s	232.90 s	671.10 s	Out of memory

$\{\{\underline{A}_1, \dots, \underline{A}_N\}, \{\overline{A}_1, \dots, \overline{A}_N\}, \{\underline{A}_1, \dots, \underline{A}_N\}, \{\overline{A}_1, \dots, \overline{A}_N\}, \{\underline{H}\}, \{\overline{H}\}, \{\underline{H}_2\}, \{\overline{H}_2\},$   
 $\{\underline{C}_2\underline{H}_2\}, \{\overline{C}_2\overline{H}_2\}, \{\underline{A}_1\underline{CHCH}, \dots, \underline{A}_N\underline{CHCH}\}, \{\overline{A}_1\overline{CHCH}, \dots, \overline{A}_N\overline{CHCH}\}\}.$

It can be noted the lower and upper bounds of each molecule-family were lumped together. Figure 3 shows the over-approximations of  $H_2$  obtained by CORA and differential hulls. Also in this case study, the plot show the results only for  $N = 2$ , but the results are similar also for bigger models. As shown in Table 2, CORA provides tighter over-approximations but becomes computationally challenging as the number of molecules grows.

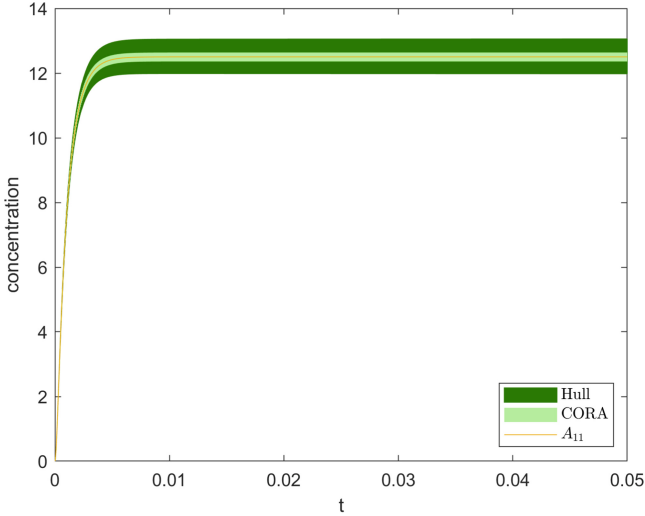
### 4.3 Protein Interaction Network

We next consider a prototypical model from systems biology where molecule  $A$  has multiple binding sites to which a molecule  $B$  can bind reversibly [11]. Since the number of reactions grows exponentially with the number of the binding sites, we only show the case for two binding sites. We indicate with  $A_{10}$  and  $A_{01}$  the complex obtained when  $A$  and  $B$  are bound via the first or second binding site, respectively. We denote with  $A_{11}$  when  $A$  is bounded with two molecules of  $B$ . The following reaction network describes this model:



The parameters  $k_b$  and  $k_u$  represent, respectively, the rate for binding and unbinding of molecules  $B$  to/from  $A$ . We define the level of heterogeneity as  $\theta_{Protein} = |k_{b1} - k_{b2}|$ . Without loss of generality, the heterogeneity was only applied to the binding parameters, with the specific parameters being reported in the Appendix. We applied our algorithm with a tolerance equal to 0.2 and computed the reduced differential hull. The reduction computed by BDE was

$$\{\{\underline{A}\}, \{\overline{A}\}, \{\underline{B}\}, \{\overline{B}\}, \{\underline{A}_{01}, \underline{A}_{10}\}, \{\overline{A}_{01}, \overline{A}_{10}\}, \{\underline{A}_{11}\}, \{\overline{A}_{11}\}\}$$



**Fig. 4.** Bounds of the molecule  $A_{11}$ .

**Table 3.** CORA running times of the protein interaction network.

$N$	2	4	6
CORA runtime	12.51 s	376.77 s	Out of memory

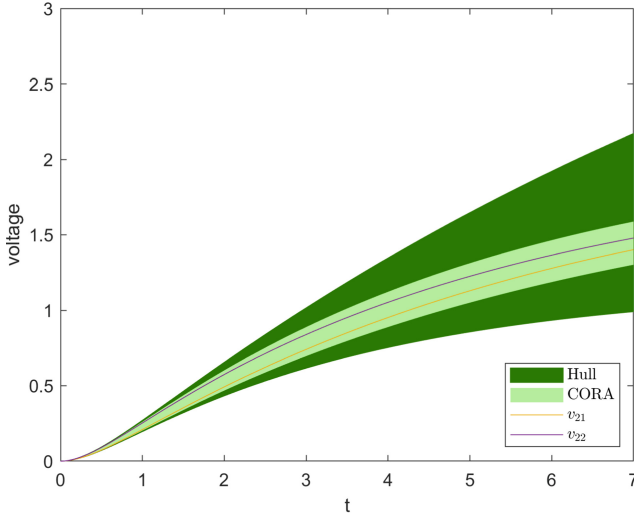
It can be noted that all molecules with the same amount of occupied binding site were lumped together. This yields an exponential reduction because the size of the original model increases exponentially in  $N$  (i.e.,  $2^N + 1$ ), while that of the reduced one polynomially (i.e.,  $N + 2$ ). We report in Fig. 4 the bounds computed with our technique and CORA; instead, Table 3 reports the computation times of CORA.

#### 4.4 Electrical Network

We consider a simplified (inductance free) version of a power distribution electrical network enjoying a so-called H-tree topology [25]. In this setting, let us denote with  $N$  the depth of the tree and let the resistance and the capacitance at depth  $i$  be denoted by  $R_{i,k}$  and  $C_{i,k}$ , respectively. We consider a constant source voltage  $v_s$  equal to 2.0 V. Denoting the voltage at  $C_{i,k}$  by  $v_{i,k}$ , we then obtain the following affine ODE system

$$\dot{v}_{1,1} = \frac{v_s - v_{1,1}}{R_{1,1}C_{1,1}} - \frac{v_{1,1} - v_{2,1}}{R_{2,1}C_{1,1}} - \frac{v_{1,1} - v_{2,2}}{R_{2,2}C_{1,1}}, \quad \dot{v}_{i,k} = \frac{v_{i-1,l} - v_{i,k}}{R_{i,k}C_{i,k}},$$

where  $1 \leq i \leq N$ ,  $k = 1, \dots, 2^{i-1}$ , and  $l = \lceil k/2 \rceil$ , with  $\lceil \cdot \rceil$  denoting the ceil function. As a baseline, we considered a network with depth  $N = 2$ . For the sake of



**Fig. 5.** Bounds of the voltages in the second level of the H-tree.

**Table 4.** CORA running times of the H-tree circuit model.

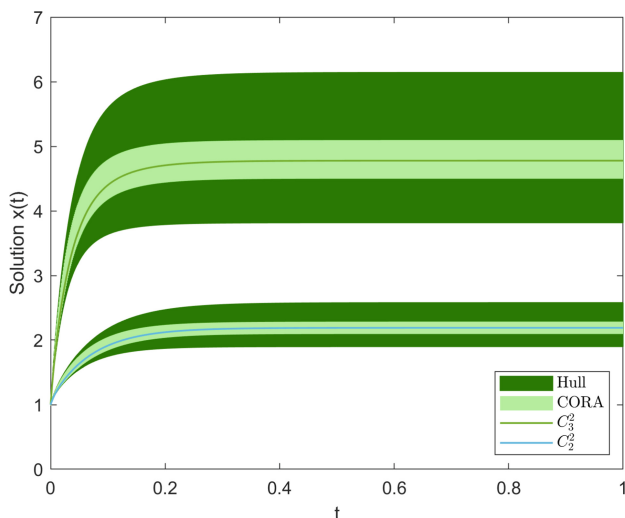
$N$	2	4	6
CORA runtime	53.99 s	231.56 s	Out of memory

simplicity, we define the associated ODE system with the following set of parameters  $\mathcal{P} = \{b_2 = 1/(R_{2,1}C_{1,1}), b_3 = 1/(R_{2,2}C_{1,1}), a_{1,1} = 1/(R_{1,1}C_{1,1}), a_{2,1} = 1/(R_{2,1}C_{2,1}), a_{2,2} = 1/(R_{2,2}C_{2,2})\}$ . We defined the following level of heterogeneity by

$$\theta_{Htree} = |b_2 - b_3| + |a_{2,1} - a_{2,2}|.$$

Similarly to the foregoing case studies, the differential hull was computed through Algorithm 1 and reduced afterwards via the BDE technique. The values of parameter and initial conditions can be found in the Appendix. The following variables were lumped:  $\{\{v_{1,1}\}, \{\bar{v}_{1,1}\}, \{v_{2,1}, v_{2,2}\}, \{\bar{v}_{2,1}, \bar{v}_{2,2}\}\}$ . As expected, the voltages of the same level are lumped together. The bounds for the voltages at the second level in case of a heterogeneity equal to 0.2 can be found in Fig. 5. We considered larger models by increasing the height  $N$  of the H-Tree. Table 4 reports the computational times required to calculate the respective over-approximations.

*Remark 2 ( $\varepsilon$ -BDE).* This model was already studied in [9], where it was reduced through  $\varepsilon$ -BDE, an approximate version of the BDE reduction. As anticipated earlier, we next discuss the bounds computed by the differential hull with one guaranteed by  $\varepsilon$ -BDE. Indeed in [9] a theorem states that, under certain conditions,  $\varepsilon$ -BDE assures a formal bound error between the original model and



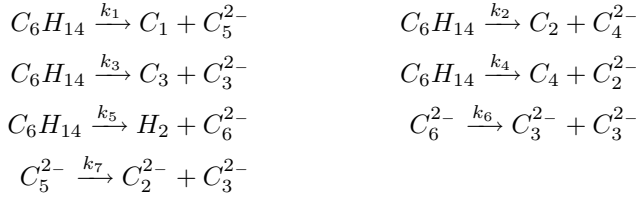
**Fig. 6.** Two largest over-approximations in the n-Hexane model (these of  $C_3^2$  and  $C_2^2$ , respectively). CORA provided tighter bounds but required around 10 s, while the proposed technique less than one second.

the reduced one. Unfortunately, the applicability of the aforementioned theorem hinges on restrictive assumptions and allows only for small heterogeneity in the parameters in practice. Instead, the differential hull always succeeds in computing error bounds for approximate lumpable trajectories. This case study is an example where the heterogeneity expressed by the parameters is too large to apply the theorem. Instead, as we can in Fig. 5, the differential hull approach is able to provide formal bounds.

#### 4.5 Conversion of Light Alkanes over H-ZSM-5

Catalytic conversions of light alkanes into industrial chemicals, such as olefins, aromatics, oxygenates, and organic nitriles, are promising candidates for traditional petroleum-based or coal-based producing routes. We consider the conversion of n-alkanes over H-ZSM5, which is commonly used in converting methanol to gasoline and diesel. In [22], the authors considered three n-alkanes: the n-Butane, the n-Pentane, and the n-Hexane. They investigated the three different conversions reporting the entire reaction networks for each n-alkanes.

We applied our framework to the n-Hexane conversion of H-ZSM5 for the original parameters from [22]. The heterogeneity parameter was set to  $\varepsilon = 15$ , while the reactions were



Similarly to before, the parameters and the initial conditions are reported in the Appendix. Likewise, the BDE algorithm was used to reduced the differential hull, giving rise to the following partition of the variables:

$$\begin{aligned}
 & \{ \{ \underline{C}_6 \underline{H}_{14} \}, \{ \overline{C}_6 \overline{H}_{14} \}, \{ \underline{C}_1, \underline{C}_4 \}, \{ \overline{C}_1 \overline{C}_4 \}, \{ \underline{C}_5^2 \}, \{ \overline{C}_5^2 \}, \{ \underline{C}_2, \underline{C}_4^2 \}, \\
 & \{ \overline{C}_2, \overline{C}_4^2 \}, \{ \underline{C}_3, \underline{H}_2 \}, \{ \overline{C}_3, \overline{H}_2 \}, \{ \underline{C}_3^2 \}, \{ \overline{C}_3^2 \}, \{ \underline{C}_2^2 \}, \{ \overline{C}_2^2 \}, \{ \underline{C}_6^2 \}, \{ \overline{C}_6^2 \} \}
 \end{aligned}$$

We compare our approach against CORA. In Fig. 6, we show the bounds computed for the molecules with the largest differential hull bounds,  $C_3^2$  and  $C_2^2$ . The CORA bounds are tighter, as expected. At the same time, CORA’s running time is around 10s, while our approach remains under 1s. Unlike to the other case studies, the computational advantage of differential hulls cannot be exploited on larger models instances.

## 5 Conclusion

Despite major efforts, the over-approximation of nonlinear models given in terms of ordinary differential equations (ODEs) remains computationally challenging. This work proposes an efficient algorithmic approach for the over-approximation of nonlinear ODE models by combining results from the theory of differential inequalities and nonlinear model reduction. More specifically, by enforcing a homogeneity across model parameters in dependence on a given numerical threshold parameter, the algorithm constructs a system of differential inequalities that a) is guaranteed to over-approximate the original ODE system in presence of uncertain/noisy parameters and; b) can be often reduced, thanks to homogeneous parameters, while preserving the aforementioned over-approximation. The applicability of the approach was demonstrated by complementing the established over-approximation tool CORA on models from epidemiology, (bio)chemistry and electrical engineering. Future work will integrate the approach into the software tool ERODE [7].

**Acknowledgments.** This work was supported in part by DFF project REDUCTO 9040-00224B, the Poul Due Jensen Grant 883901, the Villum Investigator Grant S4OS, and the PRIN project SEDUCE 2017TWRCNB.

## A Appendix

### A.1 Proofs

*Proof (Theorem 2).* Trivial.

*Proof (Theorem 3).* The only nontrivial fact to be aware about is that a parameter block with different signs will give rise a different differential hull because of the if-statements in algorithms `upperBound` and `lowerBound` will be evaluated differently.

### A.2 Experiments

We next report the parameter values and the initial conditions.

**SIR.** Here we provide the parameters and runtimes for the SIR model considered in Sect. 4.1 (Tables 5 and 6).

**Table 5.** Parameters of the SIR model.

Parameters	$\beta_{1,1}$	$\beta_{1,2}$	$\beta_{2,1}$	$\beta_{2,2}$	$\gamma_1$	$\gamma_2$
Actual values	2.46	2.45	2.53	2.55	0.5	0.6

**Table 6.** Initial conditions of the SIR model.

Variables	$S_1$	$S_2$	$I_1$	$I_2$	$R_1$	$R_2$
Initial conditions	20	20	10	10	0	0

**Polymerization.** Here we provide the parameters and runtimes for the polymerization model considered in Sect. 4.2 (Tables 7 and 8).

**Table 7.** Parameters of the Polymerization model.

Parameters	$\alpha_1$	$\alpha_2$	$\bar{\alpha}_1$	$\bar{\alpha}_2$	$\beta_1$	$\beta_2$	$\bar{\beta}_2$	$\bar{\beta}_2$	$\gamma_2$	$\gamma_2$
Actual values	0.55	0.60	1.95	2.00	1.5	1.6	0.01	0.01	0.25	0.25

**Table 8.** Initial conditions of the Polymerization model.

Variables	$A_1$	$A_2$	$A_1^{\sim}$	$A_2^{\sim}$	$H$	$H_2$	$C_2H_2$	$A_1CHCH$	$A_2CHCH$
Initial conditions	1	1	1	1	1	1	1	1	1

**Protein Interaction Network.** Here we provide the parameters and runtimes for the model considered in Sect. 4.3 (Tables 9 and 10).





## References

1. Althoff, M., Kochdumper, N.: CORA 2016 manual. TU Munich 85748 (2016)
2. Asarin, E., Dang, T., Girard, A.: Reachability analysis of nonlinear systems using conservative approximation. In: HSCC (2003)
3. Bortolussi, L., Gast, N.: Mean field approximation of uncertain stochastic models. In: DSN, pp. 287–298 (2016)
4. Cardelli, L., et al.: Exact maximal reduction of stochastic reaction networks by species lumping. *Bioinformatics* **37**(15), 2175–2182 (2021)
5. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Forward and backward bisimulations for chemical reaction networks. In: CONCUR, pp. 226–239 (2015)
6. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Symbolic computation of differential equivalences. In: POPL, pp. 137–150 (2016)
7. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: ERODE: a tool for the evaluation and reduction of ordinary differential equations. In: TACAS (2017)
8. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Maximal aggregation of polynomial dynamical systems. *Proc. Natl. Acad. Sci.* **114**(38), 10029–10034 (2017)
9. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Guaranteed error bounds on approximate model abstractions through reachability analysis. In: QEST, pp. 104–121 (2018)
10. Chen, X., Abraham, E., Sankaranarayanan, S.: Flow\*: an analyzer for non-linear hybrid systems. In: CAV, pp. 258–263 (2013)
11. Conzelmann, H., Fey, D., Gilles, E.D.: Exact model reduction of combinatorial reaction networks. *BMC Syst. Biol.* **2**(1), 1–25 (2008)
12. Duggirala, P.S., Mitra, S., Viswanathan, M.: Verification of annotated models from executions. In: EMSOFT, pp. 26:1–26:10. IEEE Press (2013)
13. Feret, J., Danos, V., Krivine, J., Harmer, R., Fontana, W.: Internal coarse-graining of molecular systems. *Proc. Natl. Acad. Sci.* **106**(16), 6453–6458 (2009)
14. Girard, A., Pappas, G.J.: Approximate bisimulations for nonlinear dynamical systems. In: IEEE Conference on Decision and Control and European Control Conference (2005)
15. Iwasa, Y., Levin, S.A., Andreasen, V.: Aggregation in model ecosystems II. Approximate aggregation. *Math. Med. Biol.* **6**(1), 1–23 (1989)
16. Kermack, W.O., McKendrick, A.G.: A contribution to the mathematical theory of epidemics. *Proc. R. Soc. London. Ser. A Contain. Papers. Math. Phys. Character.* **115**(772), 700–721 (1927)
17. Kuo, J.C.W., Wei, J.: Lumping analysis in monomolecular reaction systems. Analysis of approximately lumpable system. *Indus. Eng. Chem. Fundam.* **8**(1), 124–133 (1969)
18. Li, G., Rabitz, H.: A general analysis of approximate lumping in chemical kinetics. *Chem. Eng. Sci.* **45**(4), 977–1002 (1990)
19. Liu, J., Zhan, N., Zhao, H., Zou, L.: Abstraction of elementary hybrid systems by variable transformation. In: Bjørner, N., de Boer, F. (eds.) FM 2015. LNCS, vol. 9109, pp. 360–377. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-19249-9\\_23](https://doi.org/10.1007/978-3-319-19249-9_23)
20. Majumdar, R., Zamani, M.: Approximately bisimilar symbolic models for digital control systems. In: CAV, pp. 362–377 (2012)
21. Müller, M.: Über das Fundamentaltheorem in der Theorie der gewöhnlichen Differentialgleichungen. *Mathematische Zeitschrift* **26**, 619–645 (1927)

22. Narbeshuber, T., Vinek, H., Lercher, J.: Monomolecular conversion of light alkanes over H-ZSM-5. *J. Catal.* **157**(2), 388–395 (1995)
23. Ramdani, N., Meslem, N., Candau, Y.: Reachability of uncertain nonlinear systems using a nonlinear hybridization. In: HSCC, pp. 415–428 (2008)
24. Ramdani, N., Meslem, N., Candau, Y.: Computing reachable sets for uncertain nonlinear monotone systems. *Nonlinear Anal. Hybrid Syst.* **4**(2), 263–278 (2010)
25. Rosenfeld, J., Friedman, E.G.: Design methodology for global resonant H-tree clock distribution networks. *IEEE Trans. VLSI Syst.* **15**(2), 135–148 (2007)
26. Scott, J.K., Barton, P.I.: Bounds on the reachable sets of nonlinear control systems. *Automatica* **49**(1), 93–100 (2013)
27. Tschaikowski, M., Tribastone, M.: Approximate reduction of heterogeneous nonlinear models with differential hulls. *IEEE Trans. Autom. Contr.* **61**(4), 1099–1104 (2016)
28. Turányi, T., Tomlin, A.S.: Reduction of reaction mechanisms. In: *Analysis of Kinetic Reaction Mechanisms*, pp. 183–312. Springer, Heidelberg (2014). [https://doi.org/10.1007/978-3-662-44562-4\\_7](https://doi.org/10.1007/978-3-662-44562-4_7)
29. Voit, E.O.: Biochemical systems theory: a review. *ISRN Biomath.* **2013**, 53 (2013)