IMT School for Advanced Studies, Lucca
Lucca, Italy

Dynamical systems reduction through approximate lumping techniques

PhD Program in System Science - Track in Computer Science and Systems Engineering (CSSE)

XXXV Cycle

By

Giuseppe Squillace

2024

The dissertation of Giuseppe Squillace is approved.

PhD Program Coordinator: Prof. Alberto Bemporad, IMT School for Advanced Studies Lucca

Advisor: Prof. Mirco Tribastone, IMT School for Advanced Studies Lucca

Co-Advisor: Prof. Max Tschaikowski, Aalborg University

Co-Advisor: Prof. Andrea Vandin, School of Advanced Studies Sant'Anna

The dissertation of Giuseppe Squillace has been reviewed by:

Prof. Sabina Rossi, University Ca' Foscari of Venice

Prof. Tatjana Petrov, University of Konstanz

IMT School for Advanced Studies Lucca
2024

Contents

List of Figures viii

List of Tables x

Vita and Publications xii

Abstract xiv

1 Introduction 1

1.1 Related works 9

1.1.1 Reduction of dynamical systems 9

1.1.2 Regular Equivalences 11

2 Background 14

2.1 Ordinary Differential Equations 14

2.2 Reaction Networks 15

2.3 Exact reduction: backward and forward differential equivalence 16

2.4 CORA 20

3 Approximate BDE and FDE 22

3.1 Approximate differential equivalences 23

3.2 Error Bounds 31

3.3 Experiments 34

3.3.1 Electrical Network 34

3.3.2 Polymerization model 37

3.3.3 Protein interaction networks 40
3.3.4 Mobile virus model 43

4 Approximate Reduction through Differential Hulls 45
4.1 Differential Hull 47
4.2 Experiments 52
4.2.1 SIR Model 52
4.2.2 Polymerization 53
4.2.3 Protein interaction network 56
4.2.4 Electrical Network 56
4.2.5 Conversion of light alkanes over H-ZSM-5 58

5 Iterative ε-BDE for Approximate Regular Equivalences 61
5.1 Regular Equivalence 63
5.1.1 ε-BDE 64
5.2 Iterative ε-BDE 66
5.2.1 Asymptotics for BA networks 70
5.3 Experiments 74
5.3.1 Experimental Set-up 75
5.3.2 Results 77

6 Conclusion 81

A Appendix Chapter 3 84
A.1 Proofs 84

B Appendix Chapter 4 91
B.1 Proofs 91
B.1.1 Proof of Theorem 11 91
B.1.2 Proof of Theorem 12 91
B.2 Experiments 91
B.2.1 SIR 91
B.2.2 Polymerization 92
B.2.3 Protein interaction network 93
B.2.4 Electrical Network 93
B.2.5 n-Hexane model 93

C Appendix Chapter 5 **95**
C.1 Proofs 95

List of Figures

1 Reachability set from the CORA Manual [52]. 20

2 Given a PIVP $\mathcal{P}$, a partition $\mathcal{G}$ of $\mathcal{S}$, and an $\varepsilon > 0$, the coarsest ε-FDE/BDE partition $\mathcal{H}$ that refines $\mathcal{G}$ is constructed. Afterwards, the solution $\hat{\sigma}_*$ of the optimization problem (3.6) is computed in Fig. 1(a). This allows to compute the ε-FDE/BDE quotient $\hat{\mathcal{P}}(\hat{\sigma}_*)$ of $\mathcal{H}$. With this, λ and δ from Theorem 8 are calculated. In the case the distance between $\hat{\sigma}_0$ and $\hat{\sigma}_*$ does not exceed δ, the tight bounds of Theorem 8 can be applied and relate the trajectories of $\hat{\mathcal{P}}(\hat{\sigma}_*)$ and $\hat{\mathcal{P}}(\hat{\sigma}_0) = \mathcal{P}(\sigma)$, as depicted in Fig. 1(b). 30

3 H-tree network adapted from [133]. 35

4 (left) Over-approximation by means of differential hulls for the running example. (right) CORA over-approximation of the running example. 51

5 Bounds of the infected individuals computed by our algorithm against CORA. 54

6 Bounds of the molecule H_2 computed by Algorithm 2 against CORA. 55

7 Bounds of the molecule A_{11}. 57

8 Bounds of the voltages in the second level of the H-tree. . 58

9 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 seconds, while the proposed technique was less than one second. 59

10 Example of a network where BDE of A and A^T does not imply regular equivalence 65

11 Example of a network where BDE fails while ε-BDE find the regular equivalence $\{\{1\}, \{2, 3\}, \{4, 5, 6, 7, 8\}\}$ imposing ε equal to 1. 66

List of Tables

1 Binding model: parameters and reduction results. 2

2 Nominal parameters of electronic components at different depths i. 36

3 H-tree model results. The *Bound* column in the ε-BDE side refers to the quantity $\lambda \|\cdot\|$. 37

4 Polymerization models results.The *Bound* column in the ε-FDE side refers to the quantity $\lambda \|\cdot\|$. 40

5 Binding model: parameters and reduction results. 42

6 Results of ε-BDE and CORA for the model reported in table 5. The horizon τ is equal to 8E-4. The *Bound* column in the ε-BDE side refers to the quantity $\lambda \|\cdot\|$. 42

7 Results of ε-BDE and CORA for the multiclass SI models. The *Bound* column in the ε-BDE side refers to the quantity $\lambda \|\cdot\|$. 44

8 CORA running times for the SIR model. 53

9 CORA running times of the polymerization model. Similarly to the SIR model, the running times of differential hulls were within one second. 54

10 CORA running times of the protein interaction network. . 56

11 CORA running times of the H-tree circuit model. 57

12 Parameters and results for iterative ε-BE. 76

13 Comparison on weighted (left) and binary (right) networks. Best results in bold; methods that timed out are not listed. 78

14 Parameters of the SIR model. 92
15 Initial conditions of the SIR model. 92
16 Parameters of the Polymerization model. 92
17 Initial conditions of the Polymerization model. 92
18 Parameters of the Protein interaction network. 93
19 Initial conditions of the Protein interaction network. . . . 93
20 Parameters of the Electrical network. 93
21 Initial conditions of the Electrical network. 93
22 Parameters of the n-Hexane model. 94
23 Initial conditions of the n-Hexane model. 94

Vita

March 28, 1992	Born, Catanzaro, Italy
2016	Bachelor Degree in Computer Science University of Turin, Italy
2019	Master Degree in Computer Science University of Turin, Italy

Publications

1. G. Squillace, M. Tribastone, M. Tschaikowski, and A. Vandin, 2022, September. An Algorithm for the Formal Reduction of Differential Equations as Over-Approximations. In International Conference on Quantitative Evaluation of Systems (pp. 173-191).
2. L. Cardelli, G. Squillace, M. Tribastone, M. Tschaikowski, and A. Vandin, 2023. Formal lumping of polynomial differential equations through approximate equivalences. Journal of Logical and Algebraic Methods in Programming, 134, p.100876.

Abstract

Model reduction is a fundamental technique utilized across various disciplines, such as engineering, physics, and computational sciences, to simplify complex mathematical models while retaining essential dynamics.

This thesis introduces two novel approaches for model reduction, particularly focusing on dynamical systems described by polynomial ordinary differential equations (ODEs). The proposed techniques aim to reduce ODE systems while providing formal error bounds for the resultant reduced models.

The first approach, based on backward and forward differential equivalence (BDE/FDE), partitions the set of variables in an ODE system to construct a reduced model, incorporating a tolerance parameter ε to capture perturbations in polynomial coefficients. In the second approach, we present an algorithm to transform an ODE system into a so-called *differential hull*. This is a construction whereby variables with structurally similar dynamics but originally different parameters may be represented by the same lower and upper bounds and reduced through the backward differential equivalence.

Furthermore, the thesis explores the application of these techniques in discovering regular equivalences on networks. An iterative scheme, called iterative ε-BDE, is introduced to compute regular equivalences, allowing for the analysis of roles in networks.

Experimental evaluations demonstrate the effectiveness and efficiency of the proposed approaches compared to existing methods in the literature.

Chapter 1

Introduction

Model reduction is a fundamental technique widely utilized across diverse fields, such as engineering, physics, and computational sciences, to simplify complex mathematical models while preserving their essential dynamics [77, 1, 59, 117, 17]. The exponential increase in available data in recent years has led to systems becoming increasingly complex and computationally intensive to manage [9]. In response to this challenge, model reduction has emerged as a powerful strategy to address computational bottlenecks and facilitate analysis [9].

At its core, model reduction aims to transform detailed, high-dimensio nal models into compact representations that capture the fundamental behavior of the system. Model reduction enables faster computations, enhanced understanding, and improved control of complex systems by eliminating redundant information and focusing on the most significant variables and relationships.

The need for model reduction arises in many practical scenarios where the detailed models become computationally infeasible due to their high dimensionality and high computational costs. For example, in engineering design, the simulation of large-scale structures or systems often requires extensive computational resources [28], making real-time analysis or optimization impractical. By reducing the model complexity, engineers can achieve significant speedup in simulations, enabling efficient design itera-

	Rates of bindings				*Runtime (s)*		
N	k_{b_1}	k_{b_2}	k_{b_3}	k_{b_4}	0.8-FDE	0.4-BDE	$\lvert\mathcal{H}\rvert$
2	10.0748	9.9864	—	—	0.003	0.001	4
3	9.9174	10.0575	9.9740	—	0.010	0.001	5
4	10.0886	10.0226	9.9418	9.9505	0.078	0.002	6

Table 1: Binding model: parameters and reduction results.

tions and faster decision-making.

To provide a concrete example, consider a biological model discussed later in the experimental sections of chapters 2 and 3. In this model, a protein interaction network involves molecule A with N independent binding sites to which molecule B can bind reversibly. We focus on the scenario where $N = 2$. Here, we encounter four distinct configurations: A binding with B in the first binding site, A binding with B in the second binding site, A binding with two molecules of B, and A with all binding sites unoccupied. Each configuration entails unique dynamics that must be tracked. Notably, for $N = 2$, we observe an exponential increase in the number of dynamics, prompting the application of our dimensionality reduction approaches. Table 1 showcases the reduction achieved for the model across varying numbers of binding sites.

In the table, the first column denotes the number of binding sites, N, while the last column indicates $|\mathcal{H}|$, representing the number of dynamics in the reduced model. Our approach effectively addresses the exponential growth inherent in the original model, providing a reduced model where the number of equations scales linearly with N. This allows practitioners to simulate the system efficiently on a smaller scale.

However, it's essential to acknowledge that the reduced model introduces an approximation error in capturing the original system dynamics. In this thesis, we propose a formal quantification of this deviation, a task that embodies one of the most challenging aspects of model reduction. While constructing a reduced model may seem straightforward, minimizing errors in replicating the original system's dynamics presents a significant challenge.

Ordinary differential equations (ODEs) are a fundamental formalism to describe dynamical systems across many branches of science and engineering. In this thesis, we consider ODEs with polynomial right-hand sides. This does not impose any limitations on our findings, as polynomial ODE systems encompass a wide array of applications. They are particularly relevant for investigating nonlinear interactions occurring within populations under the assumption of opportunistic contacts, a scenario prevalent in fields such as biology [110], chemistry [112], and ecology [80]. Moreover, polynomial ODEs find utility in diverse domains, including the encoding of electric circuits [35] and the design of control systems [159]. In addition, we consider an equivalent way to represent polynomial ODE systems by means of the so-called reaction networks (RN) [8]. This formalism is strictly related to the ODEs system and represents a more compact way to represent dynamical systems where reactions and compounds substitute the equations.

It is well-known that closed-form solutions of initial value problems with polynomial ODEs are available only in special cases, a major problem when dealing with complex models regards the computational cost of the analysis, which generally is conducted by means of numerical integration. This problem has spurred a considerable amount of cross-disciplinary research on model reduction (e.g., [9, 139]). The model reduction techniques can be categorized into two main classes: exact and approximate.

Exact methods yield a reduced model that preserves the projected dynamics without introducing any error (e.g., as seen in the early work by Aoki [10]). However, achieving exact reduction necessitates matching the reduced dynamics with precisely equal parameters, which proves challenging given the complexity and uncertainty inherent in model parameters. Hence, exact techniques may not always be feasible. In contrast, approximate techniques offer a means to further reduce model dimensionality at the expense of introducing some error in the reduction process. Depending on the specific approach, these errors may be quantified with formal bounds (e.g., as demonstrated in [126] and related literature).

The most common model-reduction methods fall into one of these categories: time-scale separation [87, 89], singular value decomposition,

balanced truncation [9, 112, 139], and lumping [90].

Time-scale separation, in essence, hinges on the distinction between slow and fast variables [127]. Slow variables evolve over relatively large timescales, while fast variables undergo rapid variations over much smaller timescales. The central notion revolves around constructing reduced models that effectively capture the interplay between these variable types. The choice of slow and fast variables dictates whether the reduced model precisely or approximately replicates the original model's dynamics. Commonly employed approaches include the Singular Perturbation Method [143], Quasi-Steady State Methods [29], and Quasi-Equilibrium methods [104, 83] in their various forms [125, 155, 135]. The major drawback of these approaches is that their effectiveness relies on the identification of the timescale separation. The choice of method and its parameters should be tailored to the specific characteristics and requirements of the system under consideration.

Singular value decomposition is a matrix factorization technique that can be employed to reduce dynamical systems. By decomposing a matrix A into the product of three matrices $U\Sigma V$, SVD yields a diagonal matrix where the elements represent the importance of associated dynamics in the system. This enables the selection of crucial dynamics while discarding others to construct a reduced model. Balanced truncation, a similar approach, builds upon the work of Morrie in control theory [108]. Instead of directly manipulating the system matrix A, balanced truncation involves constructing controllability and observability Gramians, followed by finding a balancing transformation that equalizes and diagonalizes these Gramians. The subsequent truncation phase involves discarding states and outputs of the balanced model, corresponding to less significant modes. This method is particularly suited for approximate reductions, where the dimension of the reduced model depends on the discarded components.

These methods offer accurate results, and some techniques can establish a priori bounds on the error [9]. However, a notable drawback lies in the transformation applied to the state variables, rendering the interpretation of new state variables challenging. Consequently, balanced truncation

may be regarded as a black-box approach to model reduction [139].

Lumping techniques involve constructing the reduced model by defining variables as a mapping of those in the original model. This approach was proposed in [158]. Nowadays, it is a very popular approach with a lot of applications in different fields such as performance engineering [105, 100, 76, 150], machine learning [132], biology [121, 11, 55, 94] and control theory [144, 148, 6]. The lumping scheme can be categorized as linear or non-linear, depending on whether each reduced variable represents a linear combination of the original ones. Furthermore, the scheme can project the original dynamics either exactly or approximately, termed exact or approximate lumping, respectively. This framework is versatile and can be seamlessly integrated with previous techniques [55, 85, 142].

One of the strengths of lumping approaches lies in the flexibility of defining the lumping scheme. The reduced model remains interpretable since the mapping from the original variables to the new ones is well-defined. However, a significant drawback is the challenge of identifying the appropriate lumping scheme. If the scheme is ineffective, the system may be oversimplified, potentially compromising the accuracy of the reduced-order model.

In this thesis, we address the challenge of model reduction by proposing two distinct approximate lumping techniques. This decision stems from the recognition that when dealing with real-world systems, exact reductions often prove too fragile [70, 73, 79]. This fragility arises due to imprecision and uncertainties in model parameters, structure, and initial conditions, which frequently violate the criteria for exact lumping. Moreover, the techniques presented here offer formal estimations of the error computed by the reduced model, enhancing their utility and reliability.

The first approach proposed is related to a specific class of lumping induced by a so-called *differential equivalence* [34]. It consists of a partition of the set of variables in a given ODE system such that each macro-variable represents the sum of the variables in a partition block [41]. Specifically, we consider both *backward* and *forward* differential equivalence (abbreviated BDE and FDE, respectively). In the former case, variables in the same block have exactly the same solution if starting from the same ini-

tial condition; in the latter case, each macro-variable exactly represents the sum of the original variables. We introduce approximate variants of differential equivalence tailored for polynomial ODE systems. The core concept involves incorporating a threshold parameter $\varepsilon \geq 0$, which intuitively captures perturbations in polynomial coefficients. This enables the establishment of relationships between ODE variables within the same partition block that would otherwise remain distinct. The parameter ε governs the degree of similarity among lumped variable blocks. The partition obtained is employed to build the reduced model. To make the approach more robust, we equipped the reduced model with a formal bound that relates the reduced model to the original model.

The second approach reduces the model, building an augmented model [140], the so-called *differential hulls* for heterogeneous ODE systems [148]. Differential hulls provide lower- and upper-bounds on the original equations by relying on the theory of differential inequalities [129, 128, 136] which can be traced back to the seminal work of Müller [109]. We present an algorithm to compute the differential hull of a generic ODE system with positive solutions. In many cases, these systems 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 [33], class-dependent service demands in a queuing system [26], and conformation-dependent binding affinities in protein interaction networks [64]. The algorithm aims to homogenize class-dependent behavior into representative equations that suitably summarize the difference in similar parameters. The algorithm considers a level of perturbation in the parameters ε. Then, it builds the differential hull. If the original system consisted of structurally similar equations with different parameters (and these parameters are at most ε away), the intended output of this first step is to have replicated equations that have the same dynamical behavior by taking appropriate minimum and maximum values across the parameters. As a result, the resulting model becomes amenable to exact reduction via BDE, enabling the lumping together of variables with identical dynamics. This approach is orthog-

onal to the previous one. Instead of introducing an approximate technique on the original model, we apply an exact reduction on the differential hull, which is an overapproximation of the original model.

Among the wide range of applicability of these techniques, we show how the approximate BDE can be used to discover node equivalences on networks. Rooted in the social sciences, notions of node equivalences are useful tools to uncover and understand roles and relations in networks across a variety of domains, including biology [98], economics [137], and management [114]. Several definitions of node equivalence are proposed in the literature. Structural equivalence identifies nodes that are connected to the same neighbors [97]. In automorphic equivalence [22], nodes are related through graph-theoretic properties such as in-/out-degree and centralities. This is relaxed by regular equivalence, which aims to identify nodes that play the same role in the network even if they do not share neighbors by requiring that any two regularly equivalent nodes are both connected to nodes that are themselves regularly equivalent [160]. We focus on regular equivalence because it represents an advance in capturing key features of the relational role concept [24]. We consider the ε-BDE technique exploiting the well-understood interpretation of regular equivalence as a bisimulation [101]. More in detail, we relate approximate regular equivalence to an approximate bisimulation computed by means of ε-BDE. Our intuition is to associate the adjacency matrix of network A with a linear system of differential equations $\dot{x} = Ax$ and establish a formal correspondence between regular equivalence in the former and backward equivalence on the latter. More in detail, we provide an iterative framework to compute the regular equivalences on a network called iterative ε-BDE. The iterative scheme allows the computing of regular equivalence on the network considering increasing values of ε in order to capture roles with different levels of uncertainty. In addition, we exploit the partition refinement nature of the approach to specify an initial partition of the nodes. The initial partition is suitable to avoid excessive aggregation that can arise, especially on binary, as experienced by other methods in literature [23]. For this reason, we provide an initial partition for the relevant class of Barabasi-Albert (BA) networks [16], which are

well-known to fit real-world datasets appropriately. We prove formally how this partition corresponds, on average, to an ε-BDE partition for sufficiently large BA networks.

The thesis organization Chapter 2 introduces the background necessary for the rest of the thesis. Chapters 3, 4 present the new lumping techniques proposed for the reduction of dynamical systems. Chapter 5 presents the application of ε-BDE in finding regular equivalences in a network. Each of these chapters consists of a small introduction, an exposition of the theory, and an experimental analysis. The last chapter presents a final discussion with the conclusions. We decided to move the detailed proofs within the chapter to their respective appendices in order to improve the overall flow and coherence of the exposition. Part of this thesis, including Chapter 1, Chapter 2, Chapter 3, Chapter 4 and appendices A and B are based on the following publications [140, 34]:

- G. Squillace, M. Tribastone, M. Tschaikowski, and A. Vandin, 2022, September. An Algorithm for the Formal Reduction of Differential Equations as Over-Approximations. In International Conference on Quantitative Evaluation of Systems (pp. 173-191).

- L. Cardelli, G. Squillace, M. Tribastone, M. Tschaikowski, and A. Vandin, 2023. Formal lumping of polynomial differential equations through approximate equivalences. Journal of Logical and Algebraic Methods in Programming, 134, p.100876.

where the last one is a journal extension of the following paper [39]:

- L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Guaranteed Error Bounds on Approximate Model Abstractions Through Reachability Analysis". In: Quantitative Evaluation of Systems. Ed. by Annabelle McIver and Andras Horvath. Cham: Springer International Publishing, 2018, pp. 104– 121.

1.1 Related works

The proposed approaches offer the potential for reducing dynamical systems and establishing formal error bounds for the resultant reduced models. We divided the related work into two sections. The first one reports a comprehensive survey of the literature, which encompasses not only lumping techniques but also methodologies capable of computing bounds associated with the system's dynamics. The second one presents a meticulous literature review of the methods for computing the regular equivalences.

1.1.1 Reduction of dynamical systems

Classic approximation approaches relying on Lyapunov-like functions [99, 60] may provide tight bounds, but their automatic computation remains a challenging task, especially in case of nonlinearity [72]. A restriction to special classes of Lyapunov-like functions (e.g., sum-of-squares polynomials [72]), instead, leads to efficient construction algorithms, which may provide tight bounds, but existence is not guaranteed. Conversely, the approximate differential equivalences and the differential hull can efficiently compute tight bounds.

On the other side, the method based on *abstraction* like CORA locally approximates the nonlinear model by a multivariate polynomial or an affine system, see [12, 47] and references therein. The reachable set can also be over-approximated by geometrically convenient objects such as zonotopes [69, 2]. In that spirit, the approximate backward equivalences linearize across a reference trajectory, a concept also known from gain scheduling [63]. A closer approach to ours is discussed in [62]. It combines local Lyapunov-like functions and techniques based on sensitivity analysis [91]. Our bound is, however, different because the nonlinear part is bounded analytically by restricting to polynomial derivatives. On the other side, our differential hull approach always provides an approximate dynamical system that bounds the dynamics without any linearization.

In [19], the authors propose a reduction technique by overapproximating trajectories through the tropicalization of the differential equations [127]. This method leverages the concept of max-plus algebra, where the

maximum term can replace the sum of well-separated positive terms. By imposing a dominance definition, they compute upper and lower bounds for the original system. The idea is similar to the differential hull, although we do not tropicalize the equations. The approach is applied mainly to biological systems while we show the effectiveness of the differential hull in different fields.

Recently, researchers have explored deep learning techniques for model reduction [27, 141, 74, 5, 131, 67, 75, 93]. These approaches aim to reduce the computational burden of simulating large systems while providing reduced models. For instance, in [27], a Mixture Density Network (MDN) is employed to estimate the probability distribution of the model, enabling abstraction by projecting onto a subset of species. Despite yielding promising outcomes, deep learning approaches lack of full explainability, and determining the most suitable architecture for the specific problem is not straightforward. In [131], the authors introduced an automated method to identify the optimal architecture for neural networks. However, they emphasize that their objective is not to outperform any conceivable human-designed architecture, which remains the preferred choice.

Lumping techniques applied to dynamical systems have already been investigated in the literature. In chemistry, it can be traced back to Kuo and Wei [90]. They studied monomolecular reaction networks, which give rise to affine ODE systems. The approximation consists in *nearly exact lumping*, i.e., a linear transformation of the state space that would be exact up to a perturbation of the parameters. The approximation, however, only applies when the transition matrix underlying the linear system is diagonalizable. In our reduction techniques, we harness the same underlying principle, with the tolerance parameter denoted as ε, signifying the degree of perturbation in the system's parameters. Notably, our techniques offer a broader scope, eliminating the need for system diagonalization and making them applicable to a more extensive range of scenarios. Li and Rabitz extend approximate lumping to general CRNs [95], but an explicit error bound is 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, where the 0-distance induces the exact quotient [81]. The justification that variables underlying similar ODEs have nearby solutions is grounded on Gronwall's inequality, which is also at the basis of more recent quotient constructions [147, 78], which, however, are not algorithmic.

1.1.2 Regular Equivalences

In the literature, the approaches for discovering regular equivalences on a network are categorized into direct and indirect methods. The indirect approach first computes a similarity matrix among the nodes, which is then partitioned using hierarchical clustering [160, 24]. Direct approaches do not require a similarity matrix. Deterministic blockmodeling sorts the adjacency matrix in blocks [58], each identifying a cluster of nodes in the network and specifying the links from one cluster to another through an optimization problem. Stochastic blockmodeling is a generative model based on statistical properties of random graphs [66, 119].

Indirect methods REGE is the first indirect method for regular equivalence based on an iterative point-scoring procedure that builds a similarity matrix for both binary and weighted networks [160]. Its time complexity is $\mathcal{O}(n^5)$, where n is the number of nodes. CATREGE is an improved version for binary and categorical networks with time complexity $\mathcal{O}(n^3)$ [24], which, moreover, allows specifying an initial partition of the nodes to constrain the solution and improve the results. Despite this, the current implementation of CATREGE limits its use to networks with at most a few hundred nodes [25]. To build a partition of approximately regularly equivalent nodes with an indirect approach, typical hierarchical clustering techniques are based on single-link, complete-link, and Ward's method [162].

Direct methods While stochastic blockmodeling is based on a generative model, deterministic blockmodeling is more similar to indirect approaches in that it performs clustering. Usually, the number of clusters is a parameter set beforehand, and an optimization problem is set

up to minimize a certain objective function of discrepancy with respect to ideal blocks by allowing permuting rows and the columns of the adjacency matrix [58]. In binary networks, binary blockmodeling defines the discrepancy in terms of ideal blocks specified as 0-1 patterns [58]. For weighted networks, dichotomization is used; given a threshold, it sets all values in the matrix under (resp., over) the threshold with 0 (resp. 1), so that blockmodeling for binary networks can be used. Since dichotomization may lead to loss of information, in [164] a new approach, the valued blockmodeling, is proposed trying to define ideal blocks in terms of weighted networks (*f-regular* equivalence). This is shown to be more robust than dichotomization, but it requires the choice of a parameter that depends on the strength of the links. Often, the estimation is based on previous knowledge of the network [164]; some general estimations are possible, like the median or the mode of the weights, but no guarantees can be provided [102]. A different approach is the homogeneity blockmodeling proposed in [164]. Its aim is to create blocks where a measure of the variability of the links is minimal.

Since exact methods that guarantee globally optimal solutions are computationally expensive [30], heuristic methods of local search are generally employed [58], which, however, lack guarantees of optimality [58]. As shown also by our numerical experiments, such an optimization-based approach cannot scale, in practice, to networks larger than a few hundred nodes. Our algorithm can be considered a direct method in that it does not compute a similarity matrix. Different from blockmodeling, however, it does not require the user to choose the number of clusters. Instead, it is parameterized by a tolerance ε that, roughly speaking, determines the degree to which nodes can be deemed approximately regularly equivalent. In terms of efficiency, our experimental results, presented in Chapter 5, demonstrate that our approach consistently outperforms both direct and indirect methods found in the existing literature.

We mention the works [123, 124], where the authors employed the exact and approximate BDE reduction to reduce multilayer networks. More in detail, they propose a PIVP to encode the iterative scheme to compute the eigenvalue centrality on multilayer networks. Consequently, they com-

pute equivalences on this PIVP in order to find exact and approximate role assignments. The novelty of our approach is the definition of the iterative scheme that avoids aggressive aggregation on single-level networks. The scheme can be applied straightforwardly to the proposed PIVP for multilayer networks.

Chapter 2

Background

This chapter provides essential background information that serves as the foundation for the subsequent discussions in this thesis. Part of this chapter is based on the published work [34].

2.1 Ordinary Differential Equations

In our study, we focus on polynomial ordinary differential equations (ODEs) as effective models for dynamical systems.

We begin by addressing polynomial initial value problems (PIVPs) over the set of ODE variables $\mathcal{S} = \{x_1, \ldots, x_n\}$. We define $x_i(t)$ as the unique solution for variable x_i at time point t. We denote the initial conditions with $x(0)$ as the value of $x(t)$ at time t equal to 0. We assume to work with PIVPs that do not exhibit explosion in finite time, ensuring solutions on arbitrary long time intervals. A PIVP comprises ODEs in the form $\dot{x}_i = q_i$, where $1 \leq i \leq n$ and q_i is a multivariate polynomial over $\mathcal{S}$. A PIVP satisfies the *normal form* when each monomial $x^\alpha \equiv \prod_{x_i \in \mathcal{S}} x_i^{\alpha_{x_i}}$, where $\alpha \in \mathbb{N}_0^{\mathcal{S}}$ is a multi-index, appears in q_i at most once. Without loss of generality, we assume that the polynomials q_i in the PIVPs are in normal form, and we indicate this with the notation $\mathcal{N}(q_i)$. We denote with $c(q_i, x^\alpha)$ the coefficient of the monomial x^α in a normal form polynomial q_i with variables in $\mathcal{S}$, where $\alpha \in \mathbb{N}_0^{\mathcal{S}}$.

2.2 Reaction Networks

Here, we introduce the reaction network formalism as an alternative formalism to define a dynamical system. A reaction network is composed of a set of species $S = \{s_1, ..., s_n\}$ specifying the compounds that participate in the chemical reactions and a set of reactions $R = \{r_1, ..., r_m\}$. Each reaction is associated with a time-invariant kinetic parameter k, determining the frequency of occurrence. In a reaction, certain species, called reactants, combine to produce a specified number of other species, called products. We indicate with $A + 2B \xrightarrow{k} C$ a reaction where A and B are the reactants, C is the product, and the paramater k is the reaction rate. Each species involved in a reaction presents a stoichiometry coefficient that represents the quantity of this species involved in the reaction. In our reaction, one molecule of A fires between two molecules of B and produces a molecule of C. More in general, we can define a reaction i as

$$\sum_{j \in S} \alpha_{ij} s_j \rightarrow \sum_{j \in S} \beta_{ij} s_j$$

Here, the left side of the reaction equation represents the summation of reactants, each multiplied by their respective stoichiometry coefficients, α_{ij}. On the right side, we show the resulting compound of the reaction, with their stoichiometry coefficients denoted as β_{ij}.

We store the coefficients in the so-called stoichiometry matrix $\Gamma \in R^{nxm}$. A stoichiometry matrix is a mathematical representation of a chemical reaction network. The entry Γ_{ij} represents the net value of the stoichiometric coefficients (product minus reactant) of the i-th species in the j-th reaction. The sign of Γ_{ij} indicates the role of the species in the reaction: positive for a product, negative for a reactant, and zero if the concentration of the compound remains unaffected by the reaction.

A chemical reaction network can be transformed into an associated system of ordinary differential equations (ODEs) to keep track of the concentrations over time. We define $x_i(t) : S \rightarrow R^n_{\geq 0}$ as the concentration vector a time t. A common assumption is that the change in the concentrations is governed by the Law of Mass Action [45]. It states that

the rate of a chemical reaction is proportional to the product of the concentrations of the reactants, each raised to the power of their respective stoichiometric coefficients. We can define for each reaction the rate vector

$$r_i(x(t)) = k_i \prod_{j=1}^{n} x_j^{\alpha_{ij}}$$

where n is the number of species. This represents the function of the concentration and the kinetic parameters. Consequently, we can define the following system of ODE:

$$\dot{x}(t) = \Gamma r(x(t))$$

Example 1. *Let us consider a reaction network with* $S = \{A, B, C\}$ *and two reactions* $R = \{A + B \xrightarrow{k_1} 2C, C + B \xrightarrow{k_2} A\}$ *The corresponding ODE system will be:*

$$\dot{x}_A = -k_1 x_A x_B + k_2 x_B x_C$$
$$\dot{x}_B = -k_1 x_A x_B - k_2 x_C x_B \tag{2.1}$$
$$\dot{x}_C = -k_2 x_C x_B + 2k_1 x_A x_B \tag{2.2}$$

A polynomial initial value problem (PIVP) can be formulated to track species concentrations over time by specifying appropriate initial conditions.

2.3 Exact reduction: backward and forward differential equivalence

The major contribution of this thesis lies in developing approximate lumping techniques. These novel approaches either extend existing exact methods or integrate them into the proposed framework. For this reason, we present the exact techniques here. In order to explain these methods, let us consider the following running example:

Example 2. *We use the following ODE system, with variables* $\mathcal{S} = \{x_1, x_2, x_3\}$*, as a running example.*

$$\dot{x}_1 = -4.00x_1 + x_2 + x_3$$
$$\dot{x}_2 = 2.00x_1 - x_2 \tag{2.3}$$
$$\dot{x}_3 = 2.00x_1 - x_3 \tag{2.4}$$

The backward differential equivalence (BDE) and the forward differential equivalence (FDE) were originally provided in [41] for a class of nonlinear ODE systems covering derivatives more general than polynomials. Since the focus of this thesis is on polynomials, we restate these notions for a PIVP. (The proofs for this correspondence are straightforward hence we omit them.)

BDE assures that variables in the same equivalence class have the same solutions at all time points. These relations are built by making pairwise comparisons between the coefficients of the polynomials related to any two variables in the same equivalence class.

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

$$\wp^{\mathcal{H}}_{i,j} := (q_i - q_j)\big[x_{\langle H',1\rangle}/x_{H'}, \ldots, x_{\langle H',|H'|\rangle}/x_{H'} : H' \in \mathcal{H}\big]$$

i.e., when

$$\sum_{\alpha \in \mathbb{N}_0^{\mathcal{S}}} |c(\wp^{\mathcal{H}}_{i,j}, x^\alpha)| = 0. \tag{2.5}$$

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

The definition states that a candidate partition is a BDE if, for any two variables in the same block, the differences between the coefficients on the same monomials are zero. The correctness of this approach can be checked by replacing each variable in a block with a representative one. After the substitution, all the dynamics BDE equivalents have exactly the same derivatives.

Example 3. *In our running example, let us consider the partition of variables* $\mathcal{H} = \{H_1, H_2\}$*, with* $H_1 = \{x_1\}$ *and* $H_2 = \{x_2, x_3\}$*. Then* $\mathcal{H}$ *is*

a BDE partition. We can manually check the correctness of this reduction. We pick x_1 and x_2 as the representative variable for the blocks H_1 and H_2, respectively. The new PIVP is the following:

$$\dot{x}_1 = -4.00x_1 + x_2 + x_2$$
$$\dot{x}_2 = 2.00x_1 - x_2 \tag{2.6}$$
$$\dot{x}_3 = 2.00x_1 - x_2 \tag{2.7}$$

We compute $\wp^{\mathcal{H}}_{2,3} = 2.00x_1 - x_2 - (2.00x_1 - x_2)$. It consists of the difference of the equation q_2 and q_3 after the replacement of the representative variables. Now we check that the sum of coefficients in $\wp^{\mathcal{H}}_{2,3}$ is equal to 0, i.e, $\sum_{\alpha \in \mathbb{N}_0^{\mathcal{S}}} |c(\wp^{\mathcal{H}}_{i,j}, x^\alpha)| = 0$. The equation is satisfied because $\wp^{\mathcal{H}}_{2,3} = 0$. Therefore $\dot{x}_2$ and $\dot{x}_3$ are BDE. Indeed, the same equations ensure the same solutions at all-time points if they start from the same initial conditions.

FDE relates sums of variables. More in detail, it identifies a partition that induces a quotient ODE that tracks sums of variables in each equivalence class. This can be done by replacing any two variables in the same equivalence class with their sum.

Definition 2 (Forward differential equivalence (FDE))**.** *Fix a PIVP, a partition $\mathcal{H}$ of $\mathcal{S}$ and write $x_i \sim^F_{\mathcal{H}} x_j$ if all coefficients of the polynomial $\sum_{H \in \mathcal{H}} \wp^H_{i,j}$ are zero, where*

$$\wp^H_{i,j} := \sum_{x_k \in H} q_k - \sum_{x_k \in H} q_k[x_i/s(x_i + x_j), x_j/(1-s)(x_i + x_j)]$$

and s is an auxiliary variable that does not denote any state. That is, when

$$\sum_{k=1}^{m} \sum_{\alpha \in \mathbb{N}_0^{\mathcal{S} \cup \{s\}}} |c(\wp^{H_k}_{i,j}, x^\alpha)| = 0. \tag{2.8}$$

$\mathcal{H} = \{H_1, \ldots, H_m\}$ is an FDE when $\mathcal{H} = \mathcal{S}/(\sim^{F^*}_{\mathcal{H}} \cap \sim_{\mathcal{H}})$.

For a PIVP, FDE can be checked by requiring that the evaluation of the polynomial that represents the quotient derivative for an equivalence class is invariant with respect to a redistribution of the values of any two variables within that equivalence class.

Example 4. *Let us consider the partition* $\mathcal{H} = \{\{x_1\}, \{x_2, x_3\}\}$ *in our example. It is an FDE partition because we can find the following system for* $x_2 + x_3$*:*

$$\begin{aligned} \dot{x}_1 &= -4.00x_1 + (x_2 + x_3) \\ \dot{(x_2 + x_3)} &= 4.00x_1 - (x_2 + x_3) \end{aligned}$$

Replacing the variable x_{23} *with* $x_2 + x_3$ *we obtain the quotient ODE*

$$\begin{aligned} \dot{x}_1 &= -4.00x_1 + x_{23} \\ \dot{x}_{23} &= 4.00x_1 - x_{23} \end{aligned}$$

Again, we can check that the partition is FDE. We consider, as an example, the case where $H = \{x_2, x_3\}$*. We must check that the definition (2.8) holds. To illustrate this, we compute* $\wp^H_{2,3}$*:*

$$\begin{aligned} \wp^H_{2,3} = \quad & 2x_1 - x_2 + 2x_1 - x_3 \\ & - [2x_1 - s(x_2 + x_3) + 2x_1 - (1 - s)(x_2 + x_3)] \end{aligned}$$

It consists of the sum of the equation q_2 *and* q_3 *minus their formulation after redistributing the variables in the block. Upon simplification, we can ensure the* $\wp^H_{i,j}$ *is equal to 0, and consequently, the equation (2.8) is satisfied. Thus we can conclude that the solution satisfies* $x_{23}(t) = x_2(t) + x_3(t)$ *for all times* t *if this holds for the initial condition, i.e.,* $x_{23}(0) = x_2(0) + x_3(0)$*.*

In general, differential equivalences are related but not comparable to the notion of bisimulation for differential systems [72, 79] since it partitions ODE variables rather than the state space. Likewise, it complements [21] in that it captures nonlinear relations between ODE variables but does not enjoy a polynomial time algorithm. Instead, for both equivalences, one can prove the existence of a maximal exact reduction (or equivalently, of the coarsest partition) [41], which can be computed by building on fundamental algorithmic results in computer science related to partition-refinement algorithms [115]. In particular, restricting to polynomial derivatives enables the computation of differential equivalences on a suitable encoding of the ODEs into a hyper-graph akin to a formal chemical reaction network [36]. This finitary encoding ultimately allows

for algorithms that efficiently run in polynomial time and space with respect to the size of the original ODE system, related to the number of variables and of monomials appearing in the right-hand sides [40]. (This is in contrast to significantly more expensive symbolic checks for a more general class of nonlinear ODE systems studied in [41, 42].)

2.4 CORA

In this section, we present the state-of-art tool for reachability analysis CORA. The COntinuous Reachability Analyzer (CORA) [2, 3] is a MATLAB toolbox for reachability analysis. Reachability analysis is a well-known technique [48] in the domain of formal methods for system verification and validation, offering a systematic approach to exploring the dynamic behavior of complex systems. It aims to determine whether a particular state or set of states within a system can be reached from an initial state through a sequence of transitions defined by the system's dynamics. CORA requires specifying initial reachable sets that can be expressed with different formalisms. For the purposes of this thesis, we utilize zonotopes [103], which are default representations provided by CORA. Informally speaking, zonotopes are compact representations of sets in high-dimensional space.

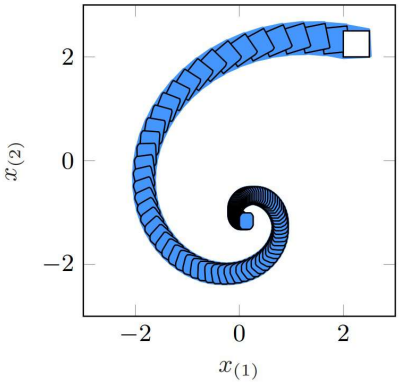

Figure 1: Reachability set from the CORA Manual [52].

They are useful in various fields, such as control theory [92] and robotics [145, 120], for describing sets of possible states or trajectories in high-dimensional spaces. Once the initial set is defined, CORA approximates the trajectories of the system, moving the zonotopes according to the dynamics of the ODE system. We report the reachability set of an example using zonotopes in Figure 1. The axes represent different variables, illustrating their movement within the space. The blue region denotes the states reached by the trajectory. Given that one of the main objectives of this thesis is to provide formal bounds, we have chosen CORA as a benchmark tool to compare computational speed and the tightness of our bounds. Through the thesis, we demonstrate that CORA yields tight bounds but requires more computational effort compared to our approaches.

Chapter 3

Approximate BDE and FDE

This chapter is based on the published work [34], which is a journal extension of [39].

Although exact reduction methods prove effective in a variety of models (e.g., performance engineering [149, 151, 146, 150], biochemistry [79, 33, 43] or physical engineering [21, 4]), especially when they have "structural" symmetries that are not dependent on specific values of the parameters, they may not be effective in applications domains where parameter uncertainty, error tolerances, and calibration from finite-precision measurements are common, such as in biology or engineering [139, 53]. To cope with this challenge, in this chapter, we introduce approximate variants of differential equivalence for polynomial ODE systems. This allows relating ODE variables in the same partition block considering a threshold parameter $\varepsilon \geq 0$, which intuitively captures perturbations in polynomial coefficients. The extension is conservative in the sense that the case $\varepsilon = 0$ corresponds to an exact differential equivalence. In addition to defining criteria for approximate differential equivalences, we provide an algorithm for obtaining the maximal one, still in a partition-refinement framework.

The reduction is represented as a *reference model*, obtained through a perturbation of the coefficients of the original model, which makes the

given approximate differential equivalence an exact one. By considering a metric (the Euclidean norm) to measure the degree of perturbation, the reference model is the one that minimizes such perturbation. This can be done efficiently by solving an optimization problem that runs polynomially with the size of the ODE system [86]. This approach is analogous to optimal approximate lumping for Markov chains (e.g., [61]), although our theory can be applied to other choices of reference models.

The bound of the error produced by the reference model with respect to the original system can be computed by studying the reachable set of the reference model from an uncertain set of initial conditions that covers the applied perturbation. Therefore, this reachable set becomes a formal bound that relates the reference model to the original model. Our bound is given in terms of an ε-δ argument (similar in spirit to the ones routinely used in calculus). Informally, it states the following: for any choice of the tolerance ε, there exists a degree of perturbation δ and an *amplifier* λ such that, for any ODE system obtained by applying a perturbation to the reference model of at most δ, at all time points the difference between the solution of the reference model and the perturbed one is at most λ times the perturbation. We show that our bounding technique can complement a state-of-the-art overapproximation technique, CORA [2, 3], in that it can scale to larger systems while being more conservative in the size of the initial uncertain set that it supports.

3.1 Approximate differential equivalences

In this section, we present two different approximate versions of BDE and FDE. In an approximate differential equivalence, we allow the conditions (2.5)-(2.8) to be satisfied with a certain amount of tolerance.

Definition 3 (Approximate BDE)**.** *Fix a PIVP, a partition* $\mathcal{H} = \{H_1, \ldots, H_m\}$ *of* $\mathcal{S}$*, and* $\varepsilon \geq 0$*. We write* $x_i \sim^B_{\mathcal{H},\varepsilon} x_j$ *if* $\sum_{\alpha \in \mathbb{N}_0^{\mathcal{S}}} |c(\wp^{\mathcal{H}}_{i,j}, x^\alpha)| \leq \varepsilon$*, where* $\wp^{\mathcal{H}}_{i,j}$ *is as in Definition 1. A partition* $\mathcal{H}$ *is an* ε*-BDE if* $\mathcal{H} = \mathcal{S}/(\sim^{B^*}_{\mathcal{H},\varepsilon} \cap \sim_{\mathcal{H}})$*.*

Definition 4 (Approximate FDE)**.** *Fix a PIVP, a partition* $\mathcal{H} = \{H_1, \ldots, H_m\}$ *of* $\mathcal{S}$*, and* $\varepsilon \geq 0$*. We write* $x_i \sim^F_{\mathcal{H},\varepsilon} x_j$ *if* $\sum_{k=1}^{m} \sum_{\alpha \in \mathbb{N}_0^{\mathcal{S} \cup \{s\}}} |c(\wp^{H_k}_{i,j}, x^\alpha)|$

$\leq \varepsilon$, *where* $\wp^H_{i,j}$ *is as in Definition 2. A partition* $\mathcal{H}$ *is an* ε*-FDE when* $\mathcal{H} = \mathcal{S}/(\sim^{F^*}_{\mathcal{H},\varepsilon} \cap \sim_{\mathcal{H}})$.

In the definitions above $*$ represents the transitive closure of this relation while B and F stand, respectively, for BDE and FDE. Setting $\varepsilon = 0$ recovers the exact counterparts in both cases. That is, $\mathcal{H}$ is a BDE (resp., FDE) partition if and only if $\mathcal{H}$ is a 0-BDE (resp., 0-FDE) partition. The two approximate differential equivalences are not comparable since their exact counterparts are not [41]. Since these two notions have similar structures, in the example of this tractation, we will illustrate only approximate BDE. Instead, both notions will be discussed in more detail for the numerical evaluation at the end of the chapter.

Example 5. *Let us consider the following running example.*

$$\dot{x}_1 = -4.00x_1 + x_2 + x_3$$
$$\dot{x}_2 = 1.99x_1 - x_2 \tag{3.1}$$
$$\dot{x}_3 = 2.01x_1 - x_3 \tag{3.2}$$

$\mathcal{H} = \big\{\{x_1\}, \{x_2, x_3\}\big\}$ *cannot be a BDE partition because* $c(\wp^{\mathcal{H}}_{2,3}, x_1) = -0.02 \neq 0$. *Of course, it is clear that in this context, the exact BDE is too fragile because a small perturbation of the parameter brings to miss the detection of a strong association. The approximate counterpart is able to cope with this problem. Indeed, if we set up* $\varepsilon = 0.02$, *the partition* $\mathcal{H}$ *is a* 0.02*-BDE partition.*

The next two theorems are concerned with providing an algorithm to compute an approximate differential equivalence. Theorem 1 shows the existence of the largest approximate differential equivalence. Theorem 2 proves the correctness of the partition-refinement algorithm to compute it as the coarsest refinement of a given initial partition of variables.

Theorem 1. *Fix a PIVP, a partition* $\mathcal{G}$ *of* $\mathcal{S}$, *and* $\varepsilon \geq 0$. *Then, there exists a unique coarsest* ε*-FDE (*ε*-BDE) partition refining* $\mathcal{G}$.

The following lemma will be needed in the proof of Theorem 2.

Lemma 1. *Let* $\mathcal{G}, \mathcal{H}$ *be two partitions of* $\mathcal{S}$. *Then, for any* $\varepsilon > 0$, *the following can be shown.*

Algorithm 1 Template partition refinement algorithm for the computation of the coarsest ε-FDE/ε-BDE partition that refines a given initial partition $\mathcal{G}$.

Require: A PIVP over variables $\mathcal{S}$, a partition $\mathcal{G}$ of $\mathcal{S}$, a threshold $\varepsilon \geq 0$, and a mode $\chi \in \{F, B\}$.
  $\mathcal{H} \longleftarrow \mathcal{G}$
  while true do
    $\mathcal{H}' \longleftarrow S/(\sim^{\chi^*}_{\mathcal{H},\varepsilon} \cap \sim_{\mathcal{H}})$
    if $\mathcal{H}' = \mathcal{H}$ **then**
      return $\mathcal{H}$
    else
      $\mathcal{H} \longleftarrow \mathcal{H}'$
    end if
  end while

(i) $x_i \sim^{F^*}_{\mathcal{H},\varepsilon} x_j$ *implies* $x_i \sim^{F^*}_{\mathcal{G},\varepsilon} x_j$ *if* $\mathcal{H}$ *is a refinement of* $\mathcal{G}$.

(ii) $x_i \sim^{B^*}_{\mathcal{H},\varepsilon} x_j$ *implies* $x_i \sim^{B^*}_{\mathcal{G},\varepsilon} x_j$ *if* $\mathcal{H}$ *is a refinement of* $\mathcal{G}$.

Theorem 2. *Fix a PIVP, a partition* $\mathcal{G}$ *of* $\mathcal{S}$*, and* $\varepsilon \geq 0$*. Then, Algorithm 1 computes the coarsest* ε*-FDE (*ε*-BDE) that refines* $\mathcal{G}$ *if* $\chi = F$ *(*$\chi = B$*).*

Example 6. *We decided to apply the Algorithm 1 on the running example. For this purpose we set an* $\varepsilon = 0.02$ *and the initial partition* $\mathcal{G} = \{x_1, x_2, x_3\}$*. In the first iteration, the partition was split into two blocks* $\mathcal{H} = \{\{x_1\}, \{x_2, x_3\}\}$*. In the next iteration,* $\mathcal{H}$ *can not be split anymore, and the algorithm returns it as the* 0.02*-BDE partition.*

The importance of these two theorems lies in the fact that, generally, the largest and unique approximate equivalence does not exist for approximate equivalence, as noted in [105]. In their work, the authors introduced the concept of approximate strong equivalence inspired by quasi-lumpability [106], where elements within the same block display a difference in the sum of rates towards equivalent classes that is less than ε. This criterion aligns with our operator $\sim^{B}_{\mathcal{H}}$. But differently, by enforcing the transitive closure of the operator to compute the partition, we establish the basis for defining the unique and coarsest ε-BDE partition.

These properties are fundamental for the definition of Algorithm 1 and assure the convergency to the resulting partition. On the other hand, the transitive closure necessary for these properties can lead to aggressive aggregation. In Chapter 5, we tackle this problem, proposing an iterative scheme for the ε-BDE reduction.

We now study how efficiently the conditions for approximate differential equivalence can be computed. Since the reduction techniques are concerned with the coefficients of the polynomials, we define the complexity in terms of $\wp_{i,j}^{H}$ and $\wp_{i,j}^{\mathcal{H}}$. In the case of ε-FDE, we estimate an exponential complexity due to term replacement. Suppose to have the PIVP $\dot{x}_1 = x_2^k, \dot{x}_2 = x_1^k$, for some $k > 0$, then, for $\mathcal{H} = \{\{x_1, x_2\}\}$, the term $q_1[x_1/s(x_1 + x_2), x_2/(1 - s)(x_1 + x_2)]$ will be of size $\mathcal{O}(2^k)$.

The ε-BDE case is different because the conditions involve a difference between polynomial terms with no term rewritings. This discussion can be formalized as follows.

Theorem 3. *There exists a polynomial* Π *such that, under the assumptions of Theorem 2, the number of steps done by Algorithm 1 is* $\mathcal{O}\big(\Pi(2^d \cdot p)\big)$ *if* $\chi = F$ *and* $\mathcal{O}\big(\Pi(p)\big)$ *if* $\chi = B$*, respectively, where* d *is the maximum degree of the polynomial and* p *is the number of monomials present in the PIVP.*

Proof. Follows from the proof of Theorem 2. □

As already mentioned, the approximate differential equivalences are a relaxation of the exact counterparts; for this reason, the above results provide a complexity bound for a subclass of ODE considered in [41].

In practice, d is not large. Indeed, in the numerical evaluations in Section 3.3, d was no larger than two. For instance, in a PIVP of a chemical reaction network with mass-action kinetics, one typically has $d = 2$ because, in nature, at most, two species interact in a given reaction [71].

For the sake of completeness, we take into account another complexity measure. Another natural definition could be the maximal distance between derivatives "semantically", i.e., under all possible evaluations within a given domain of interest. For example, consider the PIVP $\dot{x}_1 = x_1^3 - x_2$, $\dot{x}_2 = x_1 - x_2^3$. Establishing that $\{\{x_1, x_2\}\}$ is an ε-BDE would require checking that the difference between the derivatives satisfies

$$|\dot{x}_1 - \dot{x}_2| = |x_1^3 - x_1 + x_2^3 - x_2| \leq \varepsilon, \tag{3.3}$$

for all $0 \leq x_1, x_2 \leq C$ and for some finite bound $C > 0$. Since this question is in general equivalent to solving a non-convex optimization problem, we infer that the problem is NP-hard [116].

Despite this, if a partition $\mathcal{H}$ satisfies constraints like (3.3) with respect to some $\varepsilon > 0$, then we can prove that $\mathcal{H}$ is an $\mathcal{O}(\varepsilon)$-FDE/BDE, and vice versa. The basic idea is to observe that a polynomial is the zero function if and only if its coefficients are all zero. In this sense, our techniques defined through the coefficients of the polynomials correspond to check if (3.3) holds.

Given a partition of variables that represents an approximate differential equivalence, we construct a *reference PIVP* by finding a "perturbation" of the original PIVP, i.e., a modification of the initial conditions and the coefficients present in $q_1, \ldots, q_n$, which ensures that that partition becomes an exact differential equivalence. Here the initial conditions are denoted by the function $\sigma : \mathcal{S} \to \mathbb{R}$ such that $x_i(0) = \sigma(x_i)$. On this reference PIVP one can use the quotienting algorithms for FDE/BDE. Therefore, the as-obtained quotient represents an approximate reduction of the original PIVP. We obtain the desired perturbation by treating the original initial conditions and polynomial coefficients uniformly as initial conditions on an *extended* PIVP where every coefficient is parameterized and turned into a new ODE variable.

The perturbation to use in order to achieve this result can be found by considering an *extended* PIVP where every coefficient and initial condition is parameterized and turned into a new ODE variable.

Definition 5. *The parameterization of a polynomial q_i in normal form with variables $\mathcal{S}$ is denoted by $\hat{q}_i$ and arises from q_i by replacing, for each $\alpha \in \mathbb{N}_0^{\mathcal{S}}$, the constant $c(q_i, x^\alpha)$ with the parameter $\mathfrak{c}(\hat{q}_i, x^\alpha)$.*

Example 7. *The polynomials $q_2 = 1.99x_1 - x_2$ and $q_3 = 2.01x_1 - x_3$ from Example 5 give rise to the parameterized polynomials $\hat{q}_2 = \mathfrak{c}(\hat{q}_2, x_1)x_1 + \mathfrak{c}(\hat{q}_2, x_2)x_2$ and $\hat{q}_3 = \mathfrak{c}(\hat{q}_3, x_1)x_1 + \mathfrak{c}(\hat{q}_2, x_3)x_3$, respectively.*

Definition 6 (Extended PIVP)**.** *For a PIVP $\mathcal{P}$ with variables $\mathcal{S}$, set $\Theta = \{\mathfrak{c}(\hat{q}_i, x^\alpha) \mid 1 \leq i \leq n, \alpha \in \mathbb{N}_0^{\mathcal{S}}\}$. Its extended version $\hat{\mathcal{P}}$ has variables*

$\mathcal{S}\cup\Theta$ *and is given by* $\dot{x}_i = \hat{q}_i$ *and* $\dot{\mathfrak{c}}(\hat{q}_i, x^\alpha) = 0$*, where* $x_i \in \mathcal{S}$ *and* $\alpha \in \mathbb{N}_0^{\mathcal{S}}$*. For a given* $\hat{\sigma} \in \mathcal{R}^{\mathcal{S}\cup\Theta}$*, let* $\hat{\mathcal{P}}(\hat{\sigma})$ *denote the PIVP which arises from* $\hat{\mathcal{P}}$ *by replacing each* $v \in \mathcal{S}\cup\Theta$ *by the corresponding real value* $\sigma(v) \in \mathcal{R}$ *in* $\hat{\mathcal{P}}$*. In particular, let* $\hat{\sigma}_0 \in \mathcal{R}^{\mathcal{S}\cup\Theta}$ *be such that* $\mathcal{P}(\sigma) = \hat{\mathcal{P}}(\hat{\sigma}_0)$*. In particular, let* $\hat{\sigma}_0 \in \mathcal{R}^{\mathcal{S}\cup\Theta}$ *be such that* $\mathcal{P}(\sigma) = \hat{\mathcal{P}}(\hat{\sigma}_0)$*.*

Example 8. *If* $\mathcal{P}$ *is the PIVP from Example 5, its extended version* $\hat{\mathcal{P}}$ *is*

$$\begin{aligned}
\dot{x}_1 &= \mathfrak{c}(\hat{q}_1, x_1)x_1 + \mathfrak{c}(\hat{q}_1, x_2)x_2 + \mathfrak{c}(\hat{q}_1, x_3)x_3,\\
\dot{\mathfrak{c}}(\hat{q}_1, x_i) &= 0, \quad i = 1, 2, 3,\\
\dot{x}_2 &= \mathfrak{c}(\hat{q}_2, x_1)x_1 + \mathfrak{c}(\hat{q}_2, x_2)x_2,\\
\dot{\mathfrak{c}}(\hat{q}_2, x_i) &= 0, \quad i = 1, 2, 3,\\
\dot{x}_3 &= \mathfrak{c}(\hat{q}_3, x_1)x_1 + \mathfrak{c}(\hat{q}_3, x_2)x_2,\\
\dot{\mathfrak{c}}(\hat{q}_3, x_i) &= 0, \quad i = 1, 2, 3.
\end{aligned}$$

The corresponding $\hat{\sigma}_0$ *satisfies* $\hat{\sigma}_0(x_i) = \sigma(x_i)$ *for* $1 \le i \le 3$ *and*

$$\begin{aligned}
&\hat{\sigma}_0\big(\mathfrak{c}(\hat{q}_1, x_1)\big) = -4.00, && \hat{\sigma}_0\big(\mathfrak{c}(\hat{q}_1, x_2)\big) = 1.00,\\
&\hat{\sigma}_0\big(\mathfrak{c}(\hat{q}_1, x_3)\big) = 1.00, && \hat{\sigma}_0\big(\mathfrak{c}(\hat{q}_2, x_1)\big) = 1.99,\\
&\hat{\sigma}_0\big(\mathfrak{c}(\hat{q}_2, x_2)\big) = -1.00, && \hat{\sigma}_0\big(\mathfrak{c}(\hat{q}_3, x_1)\big) = 2.01,\\
&\hat{\sigma}_0\big(\mathfrak{c}(\hat{q}_3, x_2)\big) = -1.00.
\end{aligned}$$

The following is needed for the definition of the reference PIVP.

Definition 7. *Given constant free polynomial* $\hat{\wp}$ *(i.e., such that* $\hat{\wp}(0) = 0$*) and* $\Xi \subseteq \mathcal{S}\cup\Theta\cup\{s\}$*, let* $\mathfrak{t}(\hat{\wp}, x^\alpha, \Xi)$ *denote the coefficient term of* x^α *in* $\mathcal{N}(\hat{\wp}, \Xi)$*, where* $\alpha \in \mathbb{N}_0^{\Xi}$ *and* $\mathcal{N}(\hat{\wp}, \Xi)$ *is the normal form of* $\hat{\wp}$ *where variables outside* Ξ *are treated as parameters.*

Example 9. *With* $\hat{q}_2$ *and* $\hat{q}_3$ *as in Example 7 and* $\Xi = \{x_1, x_2, x_3\}$*, the normal form* $\mathcal{N}(\hat{q}_2 - \hat{q}_3, \Xi)$ *is given by* $(\mathfrak{c}(\hat{q}_2, x_1) - \mathfrak{c}(\hat{q}_3, x_1))x_1 + (\mathfrak{c}(\hat{q}_2, x_2) - \mathfrak{c}(\hat{q}_3, x_2))x_2$*, while* $\mathfrak{t}(\hat{q}_2 - \hat{q}_3, x_1, \Xi) = \mathfrak{c}(\hat{q}_2, x_1) - \mathfrak{c}(\hat{q}_3, x_1)$*.*

For an $\mathcal{H}$ to be an FDE/BDE, the coefficients of certain polynomials need to coincide or, alternatively, the corresponding differences to be zero. These constraints can be described by linear equations introduced next.

Definition 8. *Given a PIVP with variables* $\mathcal{S}$ *and an* ε*-FDE partition* $\mathcal{H}$ *of* $\mathcal{S}$*, the set of linear constraints of* $\mathcal{H}$ *is given by*

$$\big\{\mathfrak{t}(\tilde{\wp}_{i,j}^H, x^\alpha, \mathcal{S}\cup\{s\}) = 0 \mid \alpha \in \mathbb{N}_0^{\mathcal{S}\cup\{s\}}, H \in \mathcal{H} \text{ and } x_i \sim_{\mathcal{H}} x_j\big\} \tag{3.4}$$

with $\tilde{\varphi}^H_{i,j} = \sum_{x_k \in H} \hat{q}_k - \sum_{x_k \in H} \hat{q}_k[x_i/s(x_i + x_j), x_j/(1-s)(x_i + x_j)]$. *Instead, if* $\mathcal{H}$ *is an* ε*-BDE partition of* $\mathcal{S}$*, the corresponding set of linear constraints is*

$$\big\{\mathfrak{t}(\tilde{\varphi}^{\mathcal{H}}_{i,j}, x^\alpha, \mathcal{S}) = 0 \mid \alpha \in \mathbb{N}_0^{\mathcal{S}}, x_i \sim_{\mathcal{H}} x_j\big\} \cup \big\{x_{i_j} - x_{i_{j+1}} = 0 \mid 1 \leq j \leq k-1 \text{ and } \{x_{i_1}, \ldots, x_{i_k}\} \in \mathcal{S}/\!\sim_{\mathcal{H}}\big\}, \quad (3.5)$$

where $\tilde{\varphi}^{\mathcal{H}}_{i,j} = (\hat{q}_i - \hat{q}_j)\big[x_{H',1}/x_{H'}, \ldots, x_{H',|H'|}/x_{H'} : H' \in \mathcal{H}\big]$.

Next, we showcase the linear equations in our example.

Example 10. *From Example 3, we know that* $\mathcal{H} = \{\{x_1\}, \{x_2, x_3\}\}$ *is a* 0.02*-BDE partition of the PIVP (3.1). The set of linear constraints underlying* $\mathcal{H}$ *is given by* $\mathfrak{c}(\hat{q}_2, x_1) - \mathfrak{c}(\hat{q}_3, x_1) = 0$ *and* $x_2 - x_3 = 0$.

Remark 1. *In line with its exact counterpart, an* ε*-BDE is "useful" under the further constraint that related variables have the same initial conditions in the reference model as a necessary condition for having equal solutions at all time points. This translates into adding the constraints in (3.5) that perturbed initial conditions of related variables are equal. This leads, for instance, to the constraint* $x_2 - x_3 = 0$ *in the running example. For* ε*-FDE, instead, only constraints on the parameters* Θ *are made.*

Theorem 4. *Given a PIVP* $\mathcal{P}$ *with variables* $\mathcal{S}$*, an* ε*-FDE/BDE partition* $\mathcal{H}$ *and a configuration* $\hat{\sigma} \in \mathcal{R}^{\mathcal{S} \cup \Theta}$ *that satisfies (3.4)/(3.5), it holds that* $\mathcal{H}$ *is an FDE/BDE of* $\hat{\mathcal{P}}(\hat{\sigma})$.

In general, an approximate quotient is not unique; indeed, the linear system of constraints from Theorem 4 is underdetermined. Here, we fix one candidate perturbation by assuming that nearby initial conditions yield nearby trajectories. This fact is asymptotically true due to Gronwall's inequality, as mentioned in the related works section. For this purpose, we set the Euclidian norm as the objective function in a linear system with the constraints defined by 4.

$$\hat{\sigma}_* = \underset{\hat{\sigma}:\text{Eq. (3.4)/(3.5) holds}}{\operatorname{argmin}} \|\hat{\sigma} - \hat{\sigma}_0\|_2 \quad (3.6)$$

This yields a convex quadratic program that can be solved in polynomial time [86].

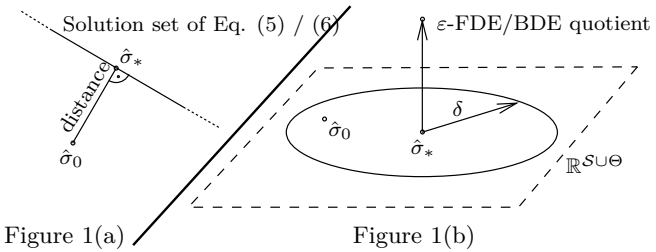

Figure 1(a) Figure 1(b)

Figure 2: Given a PIVP $\mathcal{P}$, a partition $\mathcal{G}$ of $\mathcal{S}$, and an $\varepsilon > 0$, the coarsest ε-FDE/BDE partition $\mathcal{H}$ that refines $\mathcal{G}$ is constructed. Afterwards, the solution $\hat{\sigma}_*$ of the optimization problem (3.6) is computed in Fig. 1(a). This allows to compute the ε-FDE/BDE quotient $\hat{\mathcal{P}}(\hat{\sigma}_*)$ of $\mathcal{H}$. With this, λ and δ from Theorem 8 are calculated. In the case the distance between $\hat{\sigma}_0$ and $\hat{\sigma}_*$ does not exceed δ, the tight bounds of Theorem 8 can be applied and relate the trajectories of $\hat{\mathcal{P}}(\hat{\sigma}_*)$ and $\hat{\mathcal{P}}(\hat{\sigma}_0) = \mathcal{P}(\sigma)$, as depicted in Fig. 1(b).

Example 11. *Let us continue Example 10 and assume that* $\sigma(x_2) = \sigma(x_3)$*. In such a case, it can be easily seen that* $\hat{\sigma}_*$ *and* $\hat{\sigma}_0$ *satisfy*

$$\hat{\sigma}_*(\mathfrak{c}(\hat{q}_2, x_1)) = \hat{\sigma}_*(\mathfrak{c}(\hat{q}_3, x_1)) = \big(\hat{\sigma}_0(\mathfrak{c}(\hat{q}_2, x_1)) + \hat{\sigma}_0(\mathfrak{c}(\hat{q}_3, x_1))\big)/2 = 2.00$$

and coincide on all other parameters. In other words, the closest PIVP that enjoys an exact BDE relating x_2 *and* x_3 *is given, as expected, by perturbing the coefficients* 1.99 *and* 2.01 *of (3.1) to their average value, yielding:*

$$\begin{aligned}
\dot{x}_1 &= -4.00x_1 + x_2 + x_3 \\
\dot{x}_2 &= 2.00x_1 - x_2 \\
\dot{x}_3 &= 2.00x_1 - x_3
\end{aligned}$$

The above discussions are summarized in the following.

Theorem 5. *Given a PIVP,* $\varepsilon \geq 0$*, and an* ε*-FDE/BDE partition* $\mathcal{H}$*, the solution of (3.6) exists and can be computed in polynomial time.*

Proof. Follows from Theorem 3, Theorem 4 and [86]. □

The solution of the optimization problem (3.6) stated in Theorem 5 is informally depicted in Fig. 2a.

The reference PIVP is the extended, exactly reducible PIVP with the optimum initial condition $\hat{\sigma}_*$, i.e., $\hat{\mathcal{P}}(\hat{\sigma}_*)$. Its ODE solution is called the *reference trajectory*.

3.2 Error Bounds

The objective of this section is to provide a formal bound between the solution of the original PIVP and the reference one. For this, we consider two parameters $\delta > 0$ and $\lambda > 0$. The former corresponds to the size of a ball around the initial condition $\hat{\sigma}_*$ of the reference PIVP; the latter is an *amplifier* that relates the maximum distance between trajectories to the distance between the initial conditions. Every time the initial condition of the original PIVP $\hat{\mathcal{P}}(\hat{\sigma}_0)$ stays inside the δ ball, it is possible to show that the maximum error is limited by a formal bound affected by λ. This idea is visualized in Fig. 2(b).

The next definitions and theorems build the theory necessary to formalize this idea. We start recalling the notion of Jacobian matrix.

Definition 9. *Given an extended PIVP with variables $\mathcal{S} \cup \Theta$, the entries of the Jacobian matrix $A = (A_{i,j})_{x_i, x_j \in \mathcal{S}\cup\Theta}$ are given by $A_{i,j} = \partial_{x_j} \hat{q}_i$, where ∂_x denotes the partial derivative with respect to x.*

Let $A(t) \in \mathcal{R}^{\mathcal{S}\cup\Theta \times \mathcal{S}\cup\Theta}$ denote the Jacobian obtained by plugging in the reference trajectory $x^{\hat{\sigma}_*}(t)$. We will need the following result from the theory of ODEs.

Theorem 6. *There exists a family of matrices $\Lambda(t_0, t_1)$, with $0 \leq t_0 \leq t_1 \leq \hat{\tau}$, such that the solution of $\dot{y}(t) = A(t)y(t) + u(t)$, where $y(t_0) = y_0$ and u is continuous, is given by $y(t) = \Lambda(t_0, t)y_0 + \int_{t_0}^{t} \Lambda(s, t)u(s)ds$ for all $0 \leq t_0 \leq t \leq \hat{\tau}$.*

Fix some arbitrary $\hat{\sigma}_1 \in \mathbb{R}^{\mathcal{S}\cup\Theta}$ and let $\hat{\mathcal{P}}$ be given by $\dot{x} = \hat{q}(x)$. Using Taylor's expansion of $\hat{q}$ at point $x^{\hat{\sigma}_*}(s)$, it holds that $\hat{q}(x^{\hat{\sigma}_1}(s)) = \hat{q}(x^{\hat{\sigma}_*}(s)) + A(s)(x^{\hat{\sigma}_1}(s) - x^{\hat{\sigma}_*}(s)) + r(s, x^{\hat{\sigma}_1}(s) - x^{\hat{\sigma}_*}(s))$, where r is the *remainder function* which accounts for the higher order terms of $\hat{q}$ at point $x^{\hat{\sigma}_*}(s)$. This implies that

$$(\dot{x}^{\hat{\sigma}_1}(t) - \dot{x}^{\hat{\sigma}_*}(t)) = A(t)(x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)) + r(s, x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)),$$

meaning that $x^{\hat{\sigma}_1} - x^{\hat{\sigma}_*}$ can be interpreted as a solution of the linear ODE system from Theorem 6 with input function $u(s) = r(s, x^{\hat{\sigma}_1}(s) - x^{\hat{\sigma}_*}(s))$. Let Δx denote the solution of $\Delta\dot{x}(t) = A(t)\Delta x(t)$ with $\Delta x(0) = x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)$. The above discussion and Theorem 6 then ensure that the auxiliary function $z = x^{\hat{\sigma}_1} - x^{\hat{\sigma}_*} - \Delta x$ satisfies $z(t) = \Lambda(0,t)z(0) + \int_0^t \Lambda(s,t)r(s, x^{\hat{\sigma}_1}(s) - x^{\hat{\sigma}_*}(s))ds$. With $z(0) = 0$, we thus get the following.

Theorem 7. *With the notation from above, it holds that* $\|x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)\| \leq \|\Delta x(t)\| + \|\int_0^t \Lambda(s,t) \cdot \big(r(s, x^{\hat{\sigma}_1}(s) - x^{\hat{\sigma}_*}(s))\big)ds\|$.

Since $\Delta x \equiv x^{\hat{\sigma}_1} - x^{\hat{\sigma}_*}$ only if the remainder function r is zero (that is, only if the original ODE system is linear), we call Δx the linearization of the true difference function $x^{\hat{\sigma}_1} - x^{\hat{\sigma}_*}$. With this, we are in a position to show Theorem 8 and 9.

Theorem 8. *Consider an extended PIVP* $\hat{\mathcal{P}}$ *with variables* $\mathcal{S} \cup \Theta$ *and define* $\lambda_0 = \max_{0 \leq t \leq \hat{\tau}} \|\Lambda(0,t)\|$ *and* $\lambda_1 = \max_{0 \leq t_0 \leq t_1 \leq \hat{\tau}} \|\Lambda(t_0,t_1)\|$. *Further, define the remainder function* $r : [0;\hat{\tau}] \times \mathbb{R}^{\mathcal{S}\cup\Theta} \to \mathbb{R}^{\mathcal{S}\cup\Theta}$ *via*

$$r(t, x - x^{\hat{\sigma}_*}(t)) = \hat{q}(x) - \hat{q}(x^{\hat{\sigma}_*}(t)) - A(t)(x - x^{\hat{\sigma}_*}(t))$$

and let $0 \leq d_2, d_3, \ldots$ *be such that* $\|r(t,y)\| \leq \sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k \|y\|^k$ *for all* $y \in \mathbb{R}^{\mathcal{S}\cup\Theta}$ *and* $0 \leq t \leq \hat{\tau}$. *Then, with* $\lambda = 2\lambda_0$, *for any* $x^{\hat{\sigma}_1}(0) \in \mathbb{R}^{\mathcal{S}\cup\Theta}$, *it holds that*

$$\|x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)\| \leq \delta \;\Rightarrow\; \max_{0 \leq t \leq \hat{\tau}} \|x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)\| \leq$$

$$\lambda \|x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)\|$$

whenever $\delta > 0$ *satisfies* $\sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k (2\lambda_0\delta)^{k-1} \leq (2\lambda_1\hat{\tau})^{-1}$.

Through the theorem 8, we show that there exists a formal bound proportional to λ in terms of the initial perturbation if this is smaller than δ.

We wish to point out that the maximal δ satisfying $\sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k (2\lambda_0\delta)^{k-1} \leq (2\lambda_1\hat{\tau})^{-1}$ is a root of a polynomial in one variable and thus can be efficiently approximated from below via Newton's method. Instead, the assumption $\|r(s,y)\| \leq \sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k \|y\|^k$ on the remainder function r states

essentially that, for any $k \geq 2$, the sum of all k-th order derivatives of r are bounded by d_k along the reference trajectory $x^{\hat{\sigma}_*}$.

The previous Theorem could be more precise in the special case of linear systems (i.e., $\deg(\hat{\mathcal{P}}) = 1$) as discussed in [56]. In the next result, we consider this context, and we show that the amplifier could be halved. This is because Theorem 9 need not estimate nonlinear terms present in remainder function r. More importantly, Theorem 9 shows that the amplifier of Theorem 8 cannot be substantially improved.

Theorem 9. *If an extended PIVP $\hat{\mathcal{P}}$ satisfies $\deg(\hat{\mathcal{P}}) = 1$ and $\lambda = 2\lambda_0$, it holds that*

$$\max_{0 \leq t \leq \hat{\tau}} \|x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)\| \leq \frac{\lambda}{2} \|x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)\|$$

for any $x^{\hat{\sigma}_1}(0) \in \mathbb{R}^{\mathcal{S} \cup \Theta}$. The bound is tight in the sense that there exist $0 \leq t \leq \hat{\tau}$ and $x^{\hat{\sigma}_1}(0) \in \mathbb{R}^{\mathcal{S} \cup \Theta}$ such that $\|x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)\| = \frac{\lambda}{2} \|x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)\|$.

Remark 2. *We note that λ_0, λ_1 can be estimated efficiently. Indeed, let $e_{x_i} \in \mathcal{R}^{\mathcal{S} \cup \Theta}$ be the x_i-th unit vector in $\mathbb{R}^{\mathcal{S} \cup \Theta}$, i.e., $e_{x_i}(x_j) = \delta_{i,j}$ where $\delta_{i,j}$ is the Kronecker delta. Then, if $y(t_0) = e_{x_i}$, Theorem 6 implies $y(t_1) = \Lambda(t_0, t_1) e_{x_i}$. Since $\Lambda(0, t_1) e_{x_i}$ is the x_i-column of $\Lambda(0, t_1)$ and $\Lambda(t_0, t_1) = \Lambda(0, t_1)\Lambda(0, t_0)^{-1}$, this shows that the matrices $\Lambda(t_0, t_1)$ can be computed by solving $|\mathcal{S} \cup \Theta|$ instances of the linear ODE system from Theorem 6 up to time $\hat{\tau}$.*

By calculating a bound $L > 0$ on $\max_{0 \leq t \leq \hat{\tau}} \|A(t)\|$ and by computing the matrices $\Lambda(t_i, t_j)$ for all time points t_k underlying a fixed discretization step $\Delta t > 0$ of $[0; \hat{\tau}]$, the following can be shown.

Lemma 2. *Together with $\lambda_0^+ = \max_i \|\Lambda(0, t_i)\|$ and $\lambda_1^+ = \max_{i \leq j} \|\Lambda(t_i, t_j)\|$, it holds that $\lambda_0 \leq \lambda_0^+ e^{L\Delta t}$ and $\lambda_1 \leq \lambda_1^+ [1 + L\Delta t (e^{L\Delta t} + 1)]$.*

The next result simplifies the constraints on δ from Theorem 8 if $\deg(\hat{\mathcal{P}}) \leq 3$.

Lemma 3. *In the case where $\deg(\hat{\mathcal{P}}) \leq 3$, the constraint on δ of Theorem 8 simplifies to $\delta \leq \left[2\hat{\tau}\lambda_0\lambda_1 \left(d_2 + \sqrt{d_2^2 + \frac{2d_3}{\lambda_1 \hat{\tau}}} \right) \right]^{-1}$.*

The above lemma applies, for instance, to most biochemical systems, as discussed in Section 3.1. The next result, instead, allows for an efficient estimation of d_k, with $2 \leq k \leq \deg(\hat{\mathcal{P}})$.

Lemma 4. *Given an extended PIVP $\hat{\mathcal{P}}$ with variables $\mathcal{S} \cup \Theta$, let $\#_k(\hat{q}_i)$ be the number of degree k monomials in $\mathcal{N}(\hat{q}_i)$ and $D(\hat{q}_i, \hat{\sigma})$ the largest coefficient of $\mathcal{N}(\hat{q}_i)$ for configuration $\hat{\sigma} \in \mathcal{R}^{\mathcal{S} \cup \Theta}$. With $C = \max_{0 \le t \le \hat{\tau}} \|x^{\hat{\sigma}_*}(t)\|$, $M = \max_{x_i \in \mathcal{S}} \max_{k \ge 2} \#_k(\hat{q}_i)$ and $D = \max_{x_i \in \mathcal{S}} D(\hat{q}_i, \hat{\sigma}_*)$, it suffices to set d_k in Theorem 8 to $C^{\deg(\hat{\mathcal{P}})-k} MD$.*

For the case of linear systems whose parameters are subject to perturbation, the following lemma can be applied. It provides a sharper estimate on d_2 but comes at the price of more involved computation.

Lemma 5. *Given an extended PIVP $\hat{\mathcal{P}}$ with variables $\mathcal{S} \cup \Theta$, the Hessian matrix $H^k = (H^k_{i,j})_{x_i, x_j}$ of $\hat{q}_k$ is given by $H^k_{i,j} = \partial_{x_i} \partial_{x_j} \hat{q}_k$. With this, d_2 can be chosen as $d_2 = \frac{1}{2} \cdot \max_{x_i \in \mathcal{S} \cup \Theta} \max_{0 \le t \le \hat{\tau}} \|H^i(x^{\hat{\sigma}_*}(t))\|$.*

Example 12. *Since $\deg(\hat{\mathcal{P}}) = 2$ in Example 8, coefficients $d_3, d_4, \ldots$ are zero and we only need to bound d_2. Moreover, the constraint in Theorem 8 simplifies to $\delta \le (4\hat{\tau}\lambda_0\lambda_1 d_2)^{-1}$ thanks to Lemma 3. By applying Lemma 4, instead, we see that it suffices to choose $d_2 = 3.00$ because $M = 3.00$ and $D = 1.00$. In the case of $\hat{\tau} = 2.00$, we thus get $\lambda_0 = \lambda_1 = 1.40$ which yields $\delta \le 0.02$.*

3.3 Experiments

In this section, let us show the application of ε-BDE (FDE) on real case studies. More in detail, we take into account four different examples: an electrical network, a protein interaction network, a polymerization model, and a mobile virus infection model. For every model, we provide the results achieved by applying the reductions implemented in the ERODE tool [37], and we compare them against CORA. The results reported were computed on a laptop with Windows 10 64 bits over an actual Intel Core i7 machine with 16 GB of RAM.

3.3.1 Electrical Network

We consider a simplified (inductance-free) version of a power distribution electrical network from [133], arranged as a tree called *H-tree* (Figure 3). We let N be the depth of the tree and denote the resistances and the

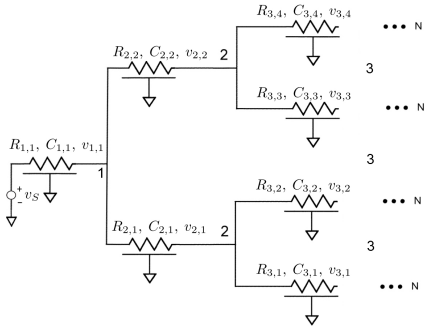

Figure 3: H-tree network adapted from [133].

capacitances at depth i by $R_{i,k}$ and $C_{i,k}$, respectively. The source voltage is v_s, here assumed to be constant, $v_s = 2.0$V. Then, the voltage across $C_{i,k}$, denoted by $v_{i,k}$, obeys the affine ODE

$$\begin{aligned} \dot{v}_{1,1} &= \frac{v_S - v_{1,1}}{R_{1,1}C_{1,1}} - \frac{v_{1,1} - v_{2,1}}{R_{2,1}C_{1,1}} - \frac{v_{1,1} - v_{2,2}}{R_{2,2}C_{1,1}}, \\ \dot{v}_{i,k} &= \frac{v_{i-1,l} - v_{i,k}}{R_{i,k}C_{i,k}}, \end{aligned} \tag{3.7}$$

where $1 \leq i \leq N$, $k = 1, \ldots, 2^{i-1}$, and $l = \lceil k/2 \rceil$, where $\lceil \cdot \rceil$ denotes the ceil function. Here, we considered networks with depth up to $N = 6$. For depths, $i \leq 4$, the nominal parameter values R_i^* and C_i^* were taken from [133]; for $i \geq 5$, instead, we extrapolated them. The parameters are summarized in Table 2.

In [133] the authors show that when the values of resistors and capacitors of any depth are equal, i.e., $R_{i,\cdot} \equiv R_i^*$ and $C_{i,\cdot} \equiv C_i^*$ then the network is *symmetric*. That is, the voltages at the capacitors at any level are equal at all time points. Indeed, $\big\{\{v_{i,k} \mid 1 \leq k \leq 2^{i-1}\} \mid 1 \leq i \leq N\big\}$ is an exact BDE partition (with N equivalence classes).

We now study the robustness of the symmetry under the realistic assumption that resistances and capacitances are only approximately equal. In particular, we test whether it is possible to explain quasi-symmetries when the parameters have tolerance $\eta = 0.01\%$. This corresponds to a practical situation when components or measurement parameters enjoy

i	R_i^* (mΩ)	C_i^* (fF)
1	3.19	0.280
2	6.37	0.300
3	12.75	0.130
4	25.50	0.140
5	50.00	0.070
6	100.00	0.070

Table 2: Nominal parameters of electronic components at different depths i.

high accuracy.

We considered networks of different sizes by varying the maximum depth N from 2 to 6. For each size, we built an ODE model by sampling values for $R_{i,k}$ and $C_{i,k}$ uniformly at random within η percent from their nominal values.

To each model, we applied the ε-BDE reduction algorithm; choosing ε = 6.00E-4, it returned a quotient corresponding to a perfectly symmetrical case. We computed the values of δ and λ over a time horizon equal to 7. This was chosen as a representative time point for any N, of the transient state of the network (to account for the fact that, typically, circuits are analyzed in the time domain for transient analysis only).

The presence of uncertain parameters required us to transform the originally affine system (3.7) into a polynomial system of degree two (by substituting each $1/(R_{i,k}C_{j,l})$ with a corresponding new state variable) with 2^{N+1} states. Since this required nonlinear over-approximation techniques, it ruled out standard over-approximation approaches for linear systems.

In Table 3 we present the results for the random models generated. The runtimes (second column) refer to the computation cost of the λ-δ pair. In all cases, δ turned out to be larger than the distance between the original model and its quotient $\|\hat{\sigma}_0 - \hat{\sigma}_*\| = \|x^{\hat{\sigma}_0}(0) - x^{\hat{\sigma}_*}(0)\|$ (shown in column $\|\cdot\|$).

This demonstrates that the 0.01% tolerance can be formally explained

	ϵ-*BDE*				*CORA*	
N	*Time (s)*	δ	$\|\cdot\|$	*Bound*	*Time (s)*	*Bound*
2	7.06E-1	7.95E–4	6.64E–5	3.59E–4	2.05E+1	3.90E–3
3	1.33E+0	6.34E–4	6.70E–5	4.20E–4	5.60E+1	3.00E–3
4	2.62E+0	4.71E–4	8.79E–5	6.84E–4	3.04E+2	2.30E–3
5	9.20E+0	4.71E–4	1.27E–4	9.88E–4	4.02E+3	1.60E–3
6	2.57E+1	4.71E–4	1.58E–4	1.23E–3	—	

Table 3: H-tree model results. The *Bound* column in the ε-BDE side refers to the quantity $\lambda \, \|\cdot\|$.

by approximate differential equivalence. Indeed, the preconditions of Theorem 8 hold, and we can provide a formal bound.

For CORA, we set the time step equal to 0.01 as this led to tight enough bounds. For our approach, instead, we used time step 0.023 because this ensured tight approximations of λ_0 and λ_1 via Lemma 2. In all cases, the time-out was set to 3 hours.

The comparison results are also reported in Table 3. For a network of depth N, the over-approximations for CORA are reported as the maximal diameter of the flowpipe underlying $v_{N,1}$ across all time points. As such, it can be compared to the product $\lambda \cdot \|\hat{\sigma}_0 - \hat{\sigma}_*\|$ given by our bound, which is also explicitly reported in the table for the sake of easy comparability (column *Bound* in Table 3).

CORA reported tight bounds, but ours are better, approximately 1 order of magnitude less. CORA failed to compute the symbolic Jacobian matrices for $N > 5$. Our approach, instead, also reaches good performance for $N = 6$. Not only that, but we also wish to point out that our algorithm naturally applies to parallelization. Indeed, its bottleneck is in the computation of the set of linear ODE systems discussed in Remark 2, which can be trivially solved independently from each other.

3.3.2 Polymerization model

In chemistry, polymerization is the process by which basic compounds, called *monomers*, react and bind to form chains of several units. A

prototypical CRN for modeling such a situation in the case of *homo-polymerization* (when monomers are of the same species), may be as follows: $A + A \to AA$, $AA + A \to AAA$, and so on until the polymer grows so large that other phenomena will cause instability, preventing further binding of monomers. To allow for an exact quotient, the kinetic rates are typically set equally in all reactions. However, this assumption is challenged by measurements, explained by the fact that the geometrical conformation of polymers of different lengths affects reactivity [153].

Here, we consider the polymerization scenario taken from [153, Chapter 7] illustrating the formation of polycyclic aromatic hydrocarbons in flame combustion. The CRN describes the growth of a molecule with i aromatic rings, denoted by the formal chemical species A_i, according to the infinite reaction scheme.

$$\begin{aligned}
\mathrm{A}_i + \mathrm{H} &\xrightarrow{\alpha_i} \mathrm{A}_i\tilde{} + \mathrm{H}_2 && (i,1) && && (3.8)\\
\mathrm{A}_i\tilde{} + \mathrm{H}_2 &\xrightarrow{\overline{\alpha}_i} \mathrm{A}_i + \mathrm{H} && (i,\overline{1}) && &&\\
\mathrm{A}_i\tilde{} + \mathrm{C_2H_2} &\xrightarrow{\beta_i} \mathrm{A}_i\mathrm{CHCH}\tilde{} && (i,2) && && (3.9)\\
\mathrm{A}_i\mathrm{CHCH}\tilde{} &\xrightarrow{\overline{\beta}_i} \mathrm{A}_i\tilde{} + \mathrm{C_2H_2} && (i,\overline{2}) && &&\\
\mathrm{A}_i\mathrm{CHCH}\tilde{} + \mathrm{C_2H_2} &\xrightarrow{\gamma_i} \mathrm{A}_{i+1} + \mathrm{H} && (i,3) && \dots && (i+1,1)
\end{aligned}$$

Here $\mathrm{A}_i\tilde{}$ is an aromatic radical formed by H abstraction from A_i, and $\mathrm{A}_i\mathrm{CHCH}\tilde{}$ is a radical formed by adding $\mathrm{C_2H_2}$ to $\mathrm{A}_i\tilde{}$. The reactions $(i,1)$ and $(i,\overline{1})$, and similarly $(i,2)$ and $(i,\overline{2})$, model reversible mechanisms. In this section, we consider the following finite truncations of the previous

model:

$$\mathrm{A}_1 + \mathrm{H} \xrightarrow{\alpha_1} \mathrm{A}_1\tilde{} + \mathrm{H}_2 \qquad (1,1) \tag{3.10}$$

$$\mathrm{A}_2 + \mathrm{H} \xrightarrow{\alpha_2} \mathrm{A}_2\tilde{} + \mathrm{H}_2 \qquad (2,1)$$

$$\mathrm{A}_1\tilde{} + \mathrm{H}_2 \xrightarrow{\overline{\alpha}_1} \mathrm{A}_1 + \mathrm{H} \qquad (1,\overline{1}) \tag{3.11}$$

$$\mathrm{A}_2\tilde{} + \mathrm{H}_2 \xrightarrow{\overline{\alpha}_2} \mathrm{A}_2 + \mathrm{H} \qquad (2,\overline{1})$$

$$\mathrm{A}_1\tilde{} + \mathrm{C}_2\mathrm{H}_2 \xrightarrow{\beta_1} \mathrm{A}_1\mathrm{CHCH}\tilde{} \qquad (1,2)$$

$$\mathrm{A}_2\tilde{} + \mathrm{C}_2\mathrm{H}_2 \xrightarrow{\beta_2} \mathrm{A}_2\mathrm{CHCH}\tilde{} \qquad (2,2) \tag{3.12}$$

$$\mathrm{A}_1\mathrm{CHCH}\tilde{} \xrightarrow{\overline{\beta}_1} \mathrm{A}_1\tilde{} + \mathrm{C}_2\mathrm{H}_2 \qquad (1,\overline{2})$$

$$\mathrm{A}_2\mathrm{CHCH}\tilde{} \xrightarrow{\overline{\beta}_2} \mathrm{A}_2\tilde{} + \mathrm{C}_2\mathrm{H}_2 \qquad (2,\overline{2}) \tag{3.13}$$

$$\mathrm{A}_1\mathrm{CHCH}\tilde{} + \mathrm{C}_2\mathrm{H}_2 \xrightarrow{\gamma_1} \mathrm{A}_2 + \mathrm{H} \qquad (1,3) \tag{3.14}$$

$$\mathrm{A}_2\mathrm{CHCH}\tilde{} + \mathrm{C}_2\mathrm{H}_2 \xrightarrow{\gamma_2} \mathrm{A}_1 + \mathrm{H} \qquad (2,3)$$

In this case, we restrict only to the dynamics of polymers up to length 2 (i.e., with $i \in \{1,2\}$), and redirect the flux originally going in A_3 to A_1, see reaction $(2,3)$. Intuitively, this mimics the fact that polymers of length 3 become unstable due to their length. Let's define N as the length of the polymers. In this section, we take into account three different truncation with N from 2 to 6. For the sake of simplicity, the set-up of the experiments is explained below only for the model with N equal to 2. The discussion easily extends to the other models. A model with polymers of length N has $3 \cdot (N+1)$ ODEs.

In [153], it is assumed that the dynamics of polymers do not depend on their length. In our truncated model, this corresponds to setting perfectly symmetric values for rates, i.e., $\alpha_1 = \alpha_2$, $\beta_1 = \beta_2$, $\gamma_1 = \gamma_2$, $\overline{\alpha}_1 = \overline{\alpha}_2$, $\overline{\beta}_1 = \overline{\beta}_2$. Then, it can be shown (similarly to [153]) that x_{A_1} and x_{A_2}, $x_{\mathrm{A}_1\tilde{}}$ and $x_{\mathrm{A}_2\tilde{}}$, and $x_{\mathrm{A}_1\mathrm{CHCH}\tilde{}}$ and $x_{\mathrm{A}_2\mathrm{CHCH}\tilde{}}$ are related by FDE.

We now consider a variant with approximately equal rates by perturbing the parameters of 0.1%, while we kept the reversed rates equal, i.e., $\overline{\alpha}_1 = \overline{\alpha}_2 = 0.1$ and $\overline{\beta}_1 = \overline{\beta}_2 = 0.2$. The perturbation was made around the values 1.0, 2.0, and 3.0, respectively, for α, β, and γ.

To study the approximation error, we set the initial concentrations of

	ε-FDE				CORA	
N	*Time (s)*	δ	$\|\cdot\|$	*Bound*	*Time (s)*	*Bound*
2	4.80	4.21E–2	1.50E–3	3.02E–3	21.54	1.50E–2
3	6.75	2.14E–2	4.00E–3	8.07E–3	48.60	1.50E–2
4	9.80	1.29E–2	4.20E–3	8.44E–3	81.27	1.50E–2
5	11.97	8.60E–3	4.60E–3	9.32E–3	231.81	1.50E–2
6	14.10	6.20E–3	5.80E–3	1.17E–2	432.47	1.50E–2

Table 4: Polymerization models results.The *Bound* column in the ε-FDE side refers to the quantity $\lambda \|\cdot\|$.

H, H_2, C_2H_2 and A_1 to 5, with a 0.06 time horizon, which ensures a good part of the models' dynamic.

The CORA bound corresponds to the maximal flowpipe diameter of molecule A_1.

The experimental results are reported in Table 4. For each model, the norms $\|\cdot\|$ is less than δ, which implies the existence of the formal bound $\lambda \|\cdot\|$ computed in column *Bound*. The running times show that ε-FDE reduction is quicker than CORA, not only, also our formal bounds outperform the ones provided by CORA.

3.3.3 Protein interaction networks

We tested approximate differential equivalence on another example of polynomial ODE systems from computational biochemistry. Here, a recurring case is the dynamics of complexes such as receptors and scaffold proteins, which have multiple binding domains (e.g., [50, 32]). Let us consider a prototypical situation where a molecule A has two independent binding sites to which molecule B can bind reversibly. We denote by A_{10} and A_{01} the species obtained when A and B are bound via the first or second binding site of A, respectively, while the other binding site is free. Instead, A_{11} denotes the complex obtained when A is bound to two molecules of B. This situation can be described using the following

mass-action CRN:

$$\begin{aligned} A + B &\xrightarrow{k_{b_1}} A_{10} & \qquad A_{10} &\xrightarrow{k_{u_1}} A + B \\ A + B &\xrightarrow{k_{b_2}} A_{01} & \qquad A_{01} &\xrightarrow{k_{u_2}} A + B \\ A_{01} + B &\xrightarrow{k_{b_1}} A_{11} & \qquad A_{11} &\xrightarrow{k_{u_1}} B + A_{01} \\ A_{10} + B &\xrightarrow{k_{b_2}} A_{11} & \qquad A_{11} &\xrightarrow{k_{u_2}} B + A_{10} \end{aligned}$$

Many models in the literature assume perfectly symmetric binding sites, where the kinetic constants do not depend on which particular binding site is involved in the interaction (e.g., [138, 107]). Mathematically, this is translated into assuming that $k_{b_1} = k_{b_2}$ and $k_{u_1} = k_{u_2}$. At the ODE level, these perfect symmetries between the species A_{10} and A_{01} are captured by both FDE and BDE [38, 41], as well as by domain-specific techniques [51, 50, 32]. However, it is well understood that, in reality, distinct binding sites will have different kinetic rates to account for their different conformation or just for the ineluctable uncertainty introduced by difficulties in measuring such rates. Thus, the assumption of perfect symmetry can be seen as a mathematical convenience to simplify the description of a more heterogeneous real system. With approximate differential equivalences, we can study *how strong* the assumption of perfect symmetry actually is.

We considered variants of this binding model by increasing the number of sites N of molecule A from 2 to 4, each site being involved in binding events with its own rate denoted by k_{b_i}; for simplicity, instead, we assume that unbinding events have the same kinetic constant $k_{u_i} = 0.1$ for every site i. The model with N binding sites has $2^N + 1$ ODEs. Similarly to the H-tree case study, the values of k_{b_i} were sampled uniformly in an interval around an arbitrarily fixed value of 10. The interval considers a 1% perturbation of the fixed value. In this way, the original model was always contained in the δ neighborhood around the reference PIVP, where all binding rates then become equal to 10 by construction. In Table 5 let's show the values for the model taken into account for experimental results.

Since distinct binding rates are used, none of the models can be reduced by FDE or BDE. Instead, we set $\varepsilon = 0.8$ and $\varepsilon = 0.4$ for ε-FDE and

	Rates of bindings				*Runtime (s)*		
N	k_{b_1}	k_{b_2}	k_{b_3}	k_{b_4}	0.8-FDE	0.4-BDE	$\lvert\mathcal{H}\rvert$
2	10.0748	9.9864	—	—	0.003	0.001	4
3	9.9174	10.0575	9.9740	—	0.010	0.001	5
4	10.0886	10.0226	9.9418	9.9505	0.078	0.002	6

Table 5: Binding model: parameters and reduction results.

	ε-BDE				CORA	
N	*Time (s)*	δ	$\lVert\cdot\rVert$	*Bound*	*Time (s)*	*Bound*
2	4.02	2.00E–0	6.25E–2	1.30E–1	5.80	1.13E–1
3	6.69	5.19E–1	9.97E–2	2.00E–1	14.25	1.34E–1
4	11.35	1.22E–1	1.19E–1	2.40E–1	—	

Table 6: Results of ε-BDE and CORA for the model reported in table 5. The horizon τ is equal to 8E-4. The *Bound* column in the ε-BDE side refers to the quantity $\lambda \lVert\cdot\rVert$.

ε-BDE, respectively. With these, all models for $2 \leq N \leq 4$ feature the same ε-FDE and ε-BDE coarsest partition (column $\lvert\mathcal{H}\rvert$ gives the number of blocks). This is similar to the exact counterpart, where all the binding rates are equal.

The fact that ε-BDE has much better runtime performance than ε-FDE backs the complexity result in Theorem 3.

We computed the bounds for the ε-BDE only. We set up the time horizon $\hat{\tau}$ equals to 8.0E–4, starting from an initial condition where the concentrations were always set to 20 for the free molecules A and B, and to zero for the other species. With this setup, the values of δ were computed using Lemma 3. The results, together with the runtimes for these computations, are reported in Table 6. We find that δ decreases as the number of asymmetric binding sites increases;

The results confirm that the δ neighborhood does contain the original model in all cases. Consequently also in this case, we can provide a formal bound for each model.

The comparison with CORA was performed similarly to the H-tree

case studies. In particular, in all cases, we used the time horizons $\hat{\tau}$, and we set the time step equal to $\hat{\tau}/100$ for CORA. For our approach, instead, in order to ensure tight approximations of λ_0 and λ_1 via Lemma 2 we used time steps equal to 1E–6. The CORA bound was computed as the maximal flow pipe diameter for the value of species A across the time interval. In the models with N equal to 2 and 3 the CORA bound is tighter. Despite this, the ε-BDE provides good formal bounds in a smaller amount of time. In the last model, CORA did not terminate its computation within the 3-hour timeout, instead, our tool returns a good formal bound.

3.3.4 Mobile virus model

We consider a simplified mobile virus propagation model inspired by [156]. Here, a region is divided into cells where the mobile virus can spread following an SI model [7]. The SI model describes the spread of an infection in a population composed of two kinds of individuals: susceptible (S) and infected (I). The former are the ones that can contract the virus, while the latter are the ones already infected. The proportion of infected grows according to the differential equation $\dot{I} = \beta SI/N$, where the parameter β is called infection rate.

More in detail, we consider a multiclass SI model where each cell represents a class of the model. In this case, the individuals are mobile phones that infect each other through message exchange. We can organize the cells in a network where nodes represent an SI model with S_i susceptible and I_i infected phones. An edge from a node i to a node j has a weight that represents the cross-class infection rate β_{ij}. The self-loops represent the spreading of the disease within the cell.

We generate three fully connected networks with 4, 8, and 12 nodes. We split the nodes into two groups with different parameters. We set up the infection rate β_{ij} equal to 1 for node i in the first group and equal to 2 for node i in the second group. In line with other case studies, we perturb these parameters of 0.1% around their fixed value. We expect to get a reduced model where the infected and susceptible belonging to the same

		ε-BDE				CORA	
N	$\hat{\tau}$	*Time (s)*	δ	$\|\cdot\|$	*Bound*	*Time (s)*	*Bound*
4	1.50	5.70E–1	3.80E–3	8.65E–4	1.89E–3	3.69E+1	1.68E–2
8	0.80	2.13E+0	2.80E–3	1.90E–3	4.52E–3	1.24E+3	4.17E–2
12	0.50	6.63E+0	2.40E–3	2.00E–3	5.18E–3	—	

Table 7: Results of ε-BDE and CORA for the multiclass SI models. The *Bound* column in the ε-BDE side refers to the quantity $\lambda \|\cdot\|$.

group are lumped together. This corresponds to the following partition

$$\big\{\{S_1, ..., S_{\frac{N}{2}}\}, \{S_{\frac{N}{2}+1}, ..., S_N\}, \{I_1, ..., I_{\frac{N}{2}}\}, \{I_{\frac{N}{2}+1}, ..., I_N\}\big\}.$$

As initial conditions, we pick a proportion of the susceptible equal to 0.9 for the first group and 1.0 for the second. We set up different time horizons $\hat{\tau}$ for each network to ensure the system's steady state. We chose a time step equal to 0.01 for both CORA and ε-BDE. We reduce the model with $\varepsilon = 5\text{E}{-3}$. The CORA bound was computed as the maximal flowpipe diameter for the value of species S_1 across the time interval.

In Table 7, we report the results for each network considered. Our approach provides *Bounds* one order of magnitude smaller than CORA. Increasing the dimension of the model, the number of parameters grows quickly. CORA goes out of memory for the model with 12 nodes, while we can compute the *Bound* in a small amount of time. The reduced model aggregates variables that are generated from the same group, showing that our approach can lump similar equations considering a perturbation of their parameters.

Chapter 4

Approximate Reduction through Differential Hulls

This chapter is based on the published work [140].

In some cases, dynamical systems describe classes of entities governed by structurally similar laws governed by different parameters. These parameters may encode different behaviors of the same phenomenon, such as age- or location-dependent rates for the contagion of a disease [33], class-dependent service demands in a queuing system [26], and conformation-dependent binding affinities in protein interaction networks [64]. When there is a large degree of heterogeneity, intended as the number of classes used in the model, the analysis becomes increasingly more complex. 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 using the notion of differential hull [148]. The differential hull is a dynamical system that provides upper and lower bounds for the dynamics of the original ODE system. The main limitation of [148] 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 presented in this chapter is to propose an algorithm to build the differential hull that is able to homogenize similar

parameters. We focus on polynomial ODEs with positive solutions. This is already a general class to which other forms of nonlinearity can be reduced [96], 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 ε that defines the amount of heterogeneity allowed in the model parameters and then computes the associated differential hull. Let us take the simple polynomial ODE system:

$$\dot{x}_1 = -k_2 x_1, \qquad \dot{x}_2 = k_1 x_1 - k_3 x_2, \qquad \dot{x}_3 = k_2 x_1 - k_3 x_3$$

with $k_1 = 1.0$, $k_2 = 1.1$, and $k_3 = 1.2$ and initial conditions all equal to 1. It is clear that the dynamics $\dot{x}_2$ and $\dot{x}_3$ presents structurally similar equations with different parameters.

The procedure consists of two steps. In the first phase, it builds the differential hull considering the perturbation in the parameter ε. In this case, the parameters suggest setting up ε equal to 0.2. The resulting differential hull is an ODE system with double the number of equations.

$$\begin{aligned} \dot{\underline{x}}_1 &= -k_3 \underline{x}_1 & \dot{\underline{x}}_2 &= k_1 \underline{x}_1 - k_3 \underline{x}_2 & \dot{\underline{x}}_3 &= k_1 \underline{x}_1 - k_3 \underline{x}_3 \\ \dot{\overline{x}}_1 &= -k_1 \overline{x}_1 & \dot{\overline{x}}_2 &= k_3 \overline{x}_1 - k_1 \overline{x}_2 & \dot{\overline{x}}_3 &= k_3 \overline{x}_1 - k_1 \overline{x}_3 \end{aligned}$$

The equations represent the lower bound $\dot{\underline{x}}_i$ and the upper bound $\dot{\overline{x}}_i$ for each original dynamic $\dot{x}_i$. The differential hull overapproximates the original system, replacing each parameter for the maximum or the minimum. The structurally similar equations in the original ODE system now present lower and upper bounds that are backward equivalent in the differential hull. Indeed, the second phase performs an automated discovery of the replicated behavior. This is done with backward differential equivalence (BDE) presented in Chapter 2. Similarly to the previous chapter, we compare the efficiency and the tightness of our bounds against the one computed by CORA.

4.1 Differential Hull

In this section, we introduce the notion of differential hull and the algorithm to compute it. We use the notation $x \leq y$ for the vectors $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ in $\mathbb{R}$ 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 provide upper and lower bounds for the dynamics of the original ODE system defined on the set of variables $\mathcal{V} = \{x_1, ..., x_n\}$.

Definition 10 (Differential Hull [148])**.** *We call* $(g_{\underline{1}}, ..., g_{\underline{n}}, g_{\overline{1}}, ..., g_{\overline{n}}) : \mathbb{R}^{2n}_{>0} \to \mathbb{R}^{2n}$ *a differential hull of the polynomial ODE system* $(q_1, ..., q_n) : \mathbb{R}^{n}_{>0} \to \mathbb{R}^{n}$ *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. Indeed, the main contribution of this chapter relies on the definition of an algorithm to compute the differential hull of a generic ODE system.

Theorem 10. *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* $0 \leq t \leq T$*.*

We show the pseudocode of this procedure in Algorithm 2. It 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 $\{(i, \alpha, c(q_i, x^\alpha)) \in O \mid 1 \leq i \leq n, \alpha \in \mathbb{N}^n_0\}$ of O in increasing order and splits them into blocks whose members are within distance ε. 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 ε. 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 considers the monomials M in equation x_i. It

computes the lower and upper bound for each of them and appends these results to $\dot{\overline{x}}$ and $\dot{\underline{x}}$, respectively.

The procedure to compute the upper bound is shown in Algorithm 3. It requires a monomial M, the coefficients partition P already calculated by Algorithm 2, 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 $\dot{x}_i$. We are computing $\dot{\overline{x}}_i$ and we find x_j equals to x_i in q_i, in this case, since the variable defines itself, the algorithm will pick $\overline{x}_j$ no matter what is the value of p.

We omit the algorithm for the lower bound, called in line 9 of Algorithm 2, because it is similar to Algorithm 3. In lines 12-13, Algorithm 2 composes the new equations to the differential hull and returns it.

Theorem 11. *The time and space complexity of Algorithm 2 and Algorithm 3 is polynomial in the size of the ODE model.*

For the sake of simplicity, let us consider the simple model presented at the beginning of this chapter as a running example.

Running example. Let us take the simple polynomial ODE system:

$$\dot{x}_1 = -k_2 x_1, \qquad \dot{x}_2 = k_1 x_1 - k_3 x_2, \qquad \dot{x}_3 = k_2 x_1 - k_3 x_3$$

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 2 with a tolerance parameter $\varepsilon = 0.2$. In the first step, the procedure splits the parameters in a single block 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 $\dot{\overline{x}}_2$. In line 6, Algorithm 2 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

Algorithm 2 computeDifferentialHull

Require: An ODE system O, a tolerance ε .

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

Algorithm 3 upperBound

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

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

positive, line 5 of Algorithm 3 picks k_3, the maximum parameter for this block. Similarly, in line 11, the maximum value that x_1 could assume is $\overline{x}_1$, which is the upper bound of x_1. In this way, the algorithm provides the first term $k_3\overline{x}_1$ of $\dot{\overline{x}}_2$. The computation proceeds with the maximization of the second term $-k_3x_2$. 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 3 replaces x_2 with $\overline{x}_2$ rather than $\underline{x}_2$. Summing up all the steps, the algorithm computes the upper bound of $\dot{x}_2$ with the equation $\dot{\overline{x}}_2 = k_3\overline{x}_1 - k_1\overline{x}_2$. The lower bound is computed similarly and, for this reason, is omitted.

Overall, the differential hull for the system reads:

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

In Figure 4 (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 $\overline{x}_i$. 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 $\overline{x}_2$ and $\overline{x}_3$. This is due to the fact that the partition of variables $\big\{\{\underline{x}_1\}, \{\overline{x}_1\}, \{\underline{x}_2, \underline{x}_3\}, \{\overline{x}_2, \overline{x}_3\}\big\}$ satisfies the BDE criterion in Eq. 2.5. This gives the following BDE-reduced differential hull where variables $\underline{x}_2$ and $\overline{x}_2$ are taken as the representatives of their respective blocks.

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

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 12. *Let O be an ODE system over variables $x_1, \ldots, x_n$ and let P be the partition as computed by Algorithm 2 and Algorithm 3. 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 $x_1, \ldots, x_n$ gives rise to the same differential hull as O when*

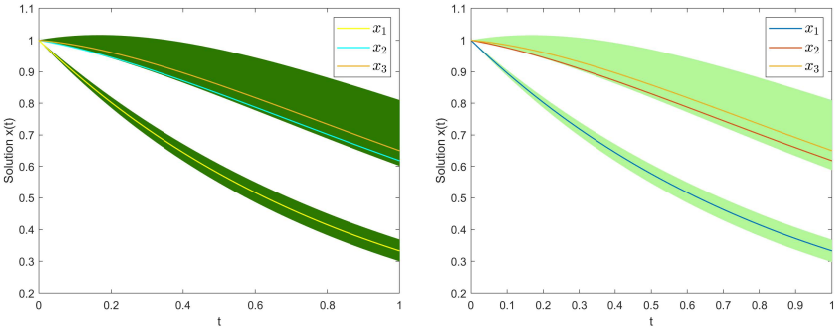

Figure 4: (left) Over-approximation by means of differential hulls for the running example. (right) CORA over-approximation of the running example.

- *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^β in q'_j of O'.

Remark 3. *The assumption on P having blocks with common signs can always be 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 12 could be applied directly.*

The foregoing result ties differential hulls to reachability analysis, where an amount of perturbation is considered among the grouped parameters. 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 $[1.0; 1.2]$, which represents the set of all the possible parameters considered by the differential hull. In Figure 4 (right) we show the bounds computed 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.

4.2 Experiments

In this section, we consider a number of case studies. The CORA implementation was carried out in Matlab, while the BDE reductions of Algorithm 2 were performed by invoking ERODE [37]. Since our approach exploits the exact reduction BDE, we consider here some models already considered in the previous chapter, where the reduction was carried out with the ε-BDE technique. All parameter values and the initial conditions are provided in the Appendix B.

4.2.1 SIR Model

The SIR model describes the spread of an infection in a population composed of three main actors: infected (I), susceptible (S), and recovered individuals (R) [84]. The infected individuals are the ones that could infect the susceptibles; the recovered obtained permanent immunization from infection because they already got the disease. The model has two types of parameters: β, the infection rate, and γ, the recovery rate. In this context, we consider the following multiclass SIR model of individuals with class-specific infection and recovery rates:

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

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

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

Table 8: CORA running times for the SIR model.

the same level of heterogeneity, as follows:

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

We computed the differential hull running our algorithm with the tolerance ε equal to θ_{SIR}, and then we reduced it with BDE. The reduced differential hull is a SIR model where all the lower and the upper bounds for each class collapse into one so that the partition achieved by BDE is:

$$P = \{\{\underline{S}_1, ..., \underline{S}_N\}, \{\overline{S}_1, ..., \overline{S}_N\}, \{\underline{I}_1, ..., \underline{I}_N\}, \{\overline{I}_1, ..., \overline{I}_N\}, \{\underline{R}_1, ..., \underline{R}_N\}, \{\overline{R}_1, ..., \overline{R}_N\}\}$$

In Figure 5, we show the comparison between CORA and differential hulls for the SIR model with two different classes; the bounds computed considering a higher number of classes are similar. CORA has tighter bounds, but it is more time-consuming. Indeed, Table 8, 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 11, the cost of the algorithm is polynomial and is based on the substitution of parameters and variables.

4.2.2 Polymerization

Polymerization is the chemical process already studied in Chapter 3. We recall the reaction network of the polymerization process as follows.

$$\begin{aligned}
\mathrm{A}_i + \mathrm{H} &\xrightarrow{\alpha_i} \mathrm{A}_i\tilde{} + \mathrm{H}_2 && (i,1) && (4.1)\\
\mathrm{A}_i\tilde{} + \mathrm{H}_2 &\xrightarrow{\overline{\alpha}_i} \mathrm{A}_i + \mathrm{H} && (i,\overline{1}) &&\\
\mathrm{A}_i\tilde{} + \mathrm{C}_2\mathrm{H}_2 &\xrightarrow{\beta i} \mathrm{A}_i\mathrm{CHCH}\tilde{} && (i,2) && (4.2)\\
\mathrm{A}_i\mathrm{CHCH}\tilde{} &\xrightarrow{\overline{\beta i}} \mathrm{A}_i\tilde{} + \mathrm{C}_2\mathrm{H}_2 && (i,\overline{2}) &&\\
\mathrm{A}_i\mathrm{CHCH}\tilde{} + \mathrm{C}_2\mathrm{H}_2 &\xrightarrow{\gamma_i} \mathrm{A}_{i+1} + \mathrm{H} && (i,3) \quad \dots \quad && (i+1,1)
\end{aligned}$$

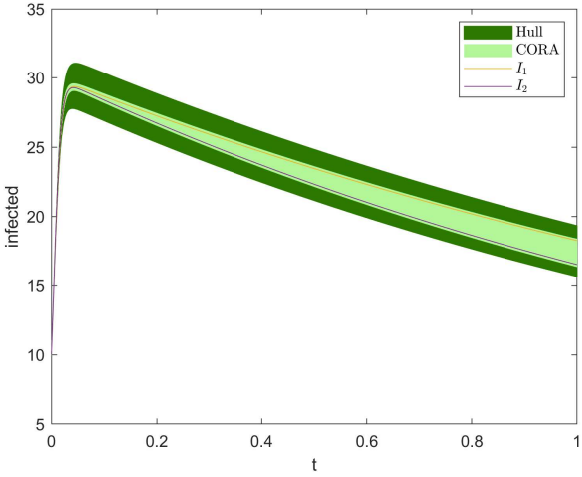

Figure 5: Bounds of the infected individuals computed by our algorithm against CORA.

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

Table 9: CORA running times of the polymerization model. Similarly to the SIR model, the running times of differential hulls were within one second.

Also, in this case, since the reaction network is infinite, we restrict our analysis to a truncated version of this model, where we consider the dynamics of polymers up to length N (i.e., with $i \in \{1, ..., N\}$). 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 SIR model, we define the following level of heterogeneity:

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

The difference between the maximum and the minimum is zero for the omitted parameters. This keeps a level of heterogeneity equal to 0.2 for

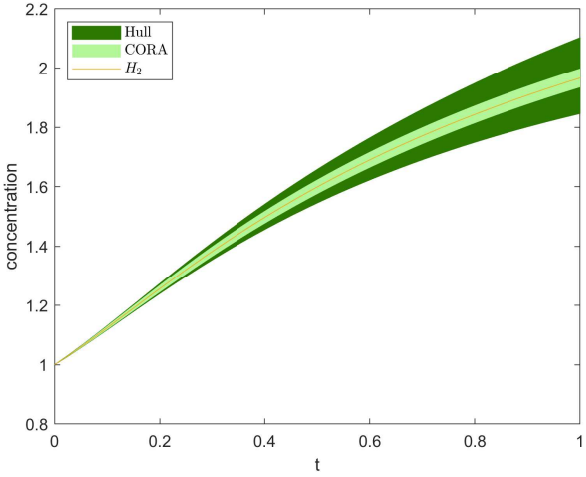

Figure 6: Bounds of the molecule H_2 computed by Algorithm 2 against CORA.

each model. For simplicity, only a part of the parameters was subject to perturbation. We ran Algorithm 2 with $\varepsilon = 0.2$, obtaining the reduced differential hull through BDE. The variables are lumped according to the following partition:

$$\begin{aligned} P = &\big\{\{\underline{A}_1, ..., \underline{A}_N\}, \{\overline{A}_1, ..., \overline{A}_N\}, \{\underline{A_1}\tilde{}, ..., \underline{A_N}\tilde{}\}, \{\overline{A_1}\tilde{}, ..., \overline{A_N}\tilde{}\}, \\ &\{\underline{H}\}, \{\overline{H}\}, \{\underline{H}_2\}, \{\overline{H}_2\}, \{\underline{C}_2\underline{H}_2\}, \{\overline{C}_2\overline{H}_2\} \\ &\{\underline{A}_1\underline{CHCH}\tilde{}, ..., \underline{A}_N\underline{CHCH}\tilde{}\}, \{\overline{A}_1\overline{CHCH}\tilde{}, ..., \overline{A}_N\overline{CHCH}\tilde{}\}\big\} \end{aligned}$$

It can be noted the lower and upper bounds of each molecule family were lumped together. Figure 6 shows the over-approximations of H_2 obtained by CORA and differential hulls. Also in this case study, the plot shows the results only for $N = 2$, but the results are similar also for bigger models. As shown in Table 9, CORA provides tighter over-approximations but becomes computationally challenging as the number of molecules grows.

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

Table 10: CORA running times of the protein interaction network.

4.2.3 Protein interaction network

We next consider the protein interaction network presented in Chapter 3. Again, we report the reaction network of this model as follows in order to facilitate the exposition of the experimental analysis.

$$\begin{aligned} A + B &\xrightarrow{k_{b_1}} A_{10} & A_{10} &\xrightarrow{k_{u_1}} A + B \\ A + B &\xrightarrow{k_{b_2}} A_{01} & A_{01} &\xrightarrow{k_{u_2}} A + B \\ A_{01} + B &\xrightarrow{k_{b_1}} A_{11} & A_{11} &\xrightarrow{k_{u_1}} B + A_{01} \\ A_{10} + B &\xrightarrow{k_{b_2}} A_{11} & A_{11} &\xrightarrow{k_{u_2}} B + A_{10} \end{aligned}$$

We applied our algorithm with a tolerance equal to 0.2 and computed the reduced differential hull. The reduction computed by BDE was

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

It can be noted that all molecules with the same amount of occupied binding sites 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 Figure 7 the bounds computed with our technique and CORA; instead, Table 10 reports the computation times of CORA.

4.2.4 Electrical Network

We consider as a further case study the Electrical network presented in Chapter 3. We report here the affine ODE system where $v_{i,k}$ denotes the voltage at $C_{i,k}$

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

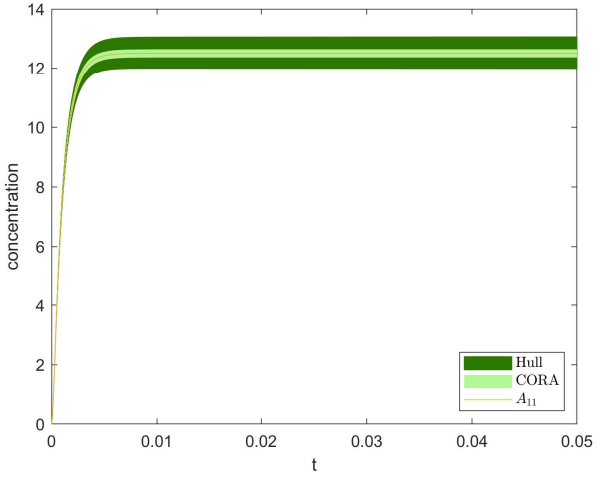

Figure 7: Bounds of the molecule A_{11}.

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

Table 11: CORA running times of the H-tree circuit model.

where $1 \leq i \leq N$, $k = 1, ..., 2^{i-1}$, 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 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 2 and reduced afterward via the BDE technique. The following variables were lumped:

$$\{\{\underline{v}_{1,1}\}, \{\overline{v}_{1,1}\}, \{\underline{v}_{2,1}, \underline{v}_{2,2}\}, \{\overline{v}_{2,1}, \overline{v}_{2,2}\}\}$$

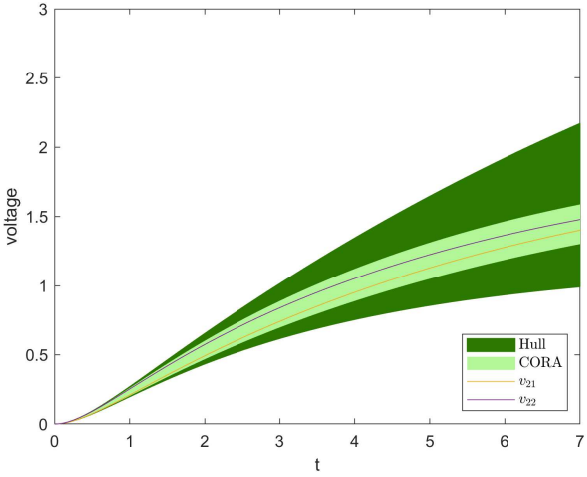

Figure 8: Bounds of the voltages in the second level of the H-tree.

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 Figure 8. We considered larger models by increasing the height N of the H-Tree. Table 11 reports the computational times required to calculate the respective over-approximations.

4.2.5 Conversion of light alkanes over H-ZSM-5

Catalytic conversions of light alkanes into industrial chemicals, such as olefins, aromatics, oxygenates, and organic nitrides, 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 [111], 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 [111]. The heterogeneity parameter was set

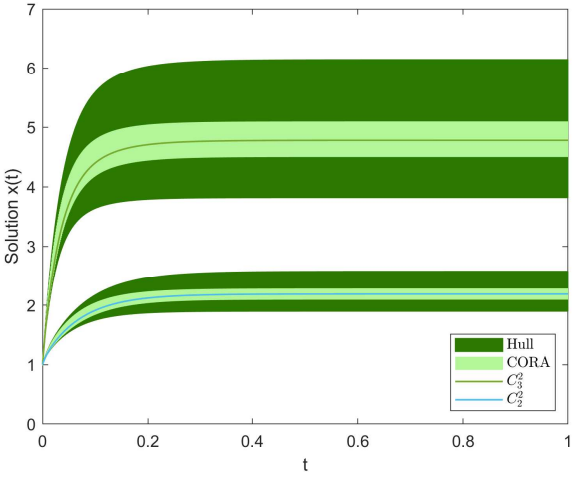

Figure 9: Two largest over-approximations in the n-Hexane model (these of C_3^2 and C_2^2, respectively). CORA provided tighter bounds but required around 10 seconds, while the proposed technique was less than one second.

to $\varepsilon = 15$, while the reactions were

$$C_6H_{14} \xrightarrow{k_1} C_1 + C_5^{2-} \qquad C_6H_{14} \xrightarrow{k_2} C_2 + C_4^{2-}$$
$$C_6H_{14} \xrightarrow{k_3} C_3 + C_3^{2-} \qquad C_6H_{14} \xrightarrow{k_4} C_4 + C_2^{2-}$$
$$C_6H_{14} \xrightarrow{k_5} H_2 + C_6^{2-} \qquad C_6^{2-} \xrightarrow{k_6} C_3^{2-} + C_3^{2-}$$
$$C_5^{2-} \xrightarrow{k_7} C_2^{2-} + C_3^{2-}$$

Likewise, the BDE algorithm was used to reduce the differential hull, giving rise to the following partition of the variables:

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

We compare our approach against CORA. In Figure 9, 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 10 seconds, while our approach remains under 1 second. Unlike the other case studies, the computational advantage of differential hulls cannot be exploited on larger model instances.

Chapter 5

Iterative ε-BDE for Approximate Regular Equivalences

Node equivalences play a vital role in simplifying complex network structures while preserving essential structural and functional properties [134, 57]. These notions involve identifying groups of nodes that can be treated as equivalent or indistinguishable in terms of their behavior or impact on the network dynamics. Node equivalences can be defined for both binary and weighted networks. Among many different notions, we consider the so-called regular equivalences [160]. Formally, two nodes are regularly equivalent if they are equally related to equivalent others. That is, regular equivalence sets are composed of nodes that have similar relations to members of other regular equivalence sets.

The regular equivalence definition, especially for weighted networks, could be too strict to discover useful relations in practice, for example, under the presence of noisy data (e.g., [130]). This has motivated the development of *approximate* relations that relax the assumptions on when two nodes can be deemed equivalent. Based on the fact that they can be related to the notion of bisimulation [101], we established a connection with ε-BDE. In the first instance, this can be done considering a net-

work denoted by an adjacency matrix A as a linear system of differential equations $\dot{x} = Ax$. Unfortunately, ε-BDE cannot be directly reused for two reasons. The first is of a mathematical nature and concerns the fact that regular equivalence is related to backward equivalence on both the network A and its transpose A^T. The second reason is that the algorithm constructs equivalence classes through a transitive closure of nodes that are pairwise ε-similar, which may lead to aggressive aggregation in the output (cf. Example 13). A similar phenomenon occurs in indirect methods, especially with binary networks. They could fail the analysis by identifying all nodes as approximately regularly equivalent [23].

To address these challenges, we propose an iterative approach wherein the ε-BDE algorithm is executed iteratively on both A and A^T, with each iteration considering progressively larger values of ε to prevent excessive aggregation. Additionally, the algorithm allows users to specify the initial partition to be refined, offering flexibility to encode specific requirements or prior knowledge. In general, the largest equivalence (i.e., the coarsest partition) is computed by initializing the algorithm with a singleton partition where all nodes are in the same block. However, users may choose an arbitrary initial partition to suit their needs, such as isolating a node or prepartitioning nodes based on predefined roles.

We exploit this feature by providing an initial candidate partition for the relevant class of Barabasi-Albert (BA) networks, which are well-known for fitting real-world datasets appropriately. Intuitively, such networks are particularly challenging for our algorithm because their power-law distributed degrees may lead to relatively low values of ε to collapse many low-degree nodes, with the risk of losing much information in the resulting equivalence. The initial BA partition avoids aggressive reduction and fits our framework because we show it is, on average, an ε-BDE partition for sufficiently large BA networks.

We conduct an experimental evaluation of our algorithm on binary and weighted networks from the literature. Our approach provides more accurate partitions than both direct and indirect methods using the same level of granularity (number of clusters), as indicated by statistics on the centralities of approximately regular equivalents. Furthermore, our ap-

proach demonstrates efficiency and scalability, particularly outperforming direct and indirect methods on larger networks. Additionally, the proposed asymptotic BA partition offers a solution to potential issues of excessive clustering. It also serves as a suitable pre-partition for indirect methods, such as those outlined in [23], which are designed to identify all nodes within the same block.

5.1 Regular Equivalence

We define a network with n nodes by its adjacency matrix $A = (a_{i,j}) \in \mathbb{R}^{nxn}$ where each component a_{ij} denotes the weight of the link from node i to node j; as usual, we call a network *binary* if $a_{i,j} \in \{0, 1\}$ and *undirected* if A is symmetric. We indicate with L the number of distinct weights in a weighted network. Nodes are labeled 1, 2, ..., n. Intuitively, regular equivalence relates nodes equivalent whenever these have identical links to and from regularly equivalent nodes [160]. For the purposes of this thesis, it is convenient to express it via the classic notion of bisimulation [101], as recalled next.

Definition 11. *For an adjacency matrix $A \in \{0,1\}^{n\times n}$, we write $i \to j$ whenever $a_{i,j} = 1$.*

- *A equivalence relation $\mathcal{R}$ is a bisimulation of A if for any $(i, j) \in \mathcal{R}$ and link $i \to i'$, there exists a link $j \to j'$ such that $(i', j') \in \mathcal{R}$*
- *A relation $\mathcal{R}$ is a regular equivalence of A whenever $\mathcal{R}$ is a bisimulation of A and A^T.*
- *We set $\mathcal{H}_{\mathcal{R}} = \{1, \ldots, n\}/\mathcal{R}$ for any equivalence relation $\mathcal{R}$.*

The definition naturally extends to weighted networks by, essentially, treating every distinct weight as a categorical label and requiring regular equivalences on all such labels (e.g., [162]).

Definition 12. *Let $A \in \mathcal{R}^{n\times n}$ be a weighted adjacency matrix with L distinct weights such that $A = \sum_{l=1}^{L} w_l A^l$, where $w^l \in \mathcal{R}$ and $A^l \in \{0,1\}^{n\times n}$. Then, $\mathcal{R}$ is a regular equivalence of A if $\mathcal{R}$ is a regular equivalence of $A^1, \ldots, A^L$.*

Regular equivalence allows the same link of a node to match more than one link of a regularly equivalent one.

We want to establish a connection between regular equivalences on networks and the BDE reduction on ODE systems. To do this, given a matrix $A \in \mathcal{R}^{n\times n}$ one considers an associated linear system of ordinary differential equations (ODEs) in the form $\dot{x} = Ax$, where $\dot{x}$ denotes the time derivative of the solution x. Here, we state the BDE definition tailored for a linear dynamical system associated with a network denoted by an adjacency matrix A.

Definition 13. *For an adjacency matrix $A \in \mathcal{R}^{n\times n}$, an equivalence relation $\mathcal{R}$ is called backward equivalence (BDE) when*

$$\sum_{H'\in\mathcal{H}_\mathcal{R}} |\sum_{k\in H'} a_{k,i} - \sum_{k\in H'} a_{k,j}| = 0$$

for all $H, H' \in \mathcal{H}_\mathcal{R}$ and $i, j \in H$.

Let us observe that BDE matches cumulative in-degrees towards equivalence classes (hence the term *backward*). A BDE relation for the transpose adjacency matrix A^T corresponds to a *forward* equivalence that matches out-degrees [14, 154, 13, 31]. Differently from regular equivalences, the BDE technique is based on the concept of bisimulation for dynamical systems that, informally, relate nodes that have the *same cumulative degree* with respect to blocks of nodes in the same equivalence class.

5.1.1 ε-BDE

We start with a simple yet crucial statement that relates regular equivalence with BDE.

Theorem 13. *Given $A = \sum_{l=1}^{L} w^l A^l$ with $A^l \in \{0,1\}^{n\times n}$, assume that $\mathcal{R}$ is a BDE of A^l and $(A^l)^T$, for all $1 \le l \le L$. Then, $\mathcal{R}$ is a regular equivalence of A and each $A^1, \ldots, A^L$.*

The above statement provides us with a sufficient condition for regular equivalence. Unfortunately, its assumption cannot be relaxed to $\mathcal{R}$ being

the BDE of A and A^T only. Indeed, partition $\{\{1,2,3\},\{4,5,6\}\}$ of the network depicted in Figure 10 can be shown to be a BDE of A and A^T, where $A = A^1 + \ldots + A^4$. At the same time, however, it is not a regular equivalence of A because it is not a regular equivalence of A^1. Indeed, nodes 1 and 3 have black links, while node 2 has no black links. It can also be noted that, in contrast to regular equivalence, BDE requires the rather strict assumption regarding equal degrees of related nodes.

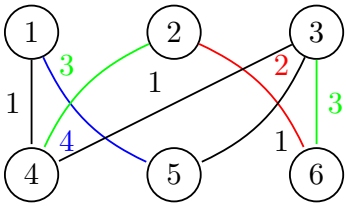

Figure 10: Example of a network where BDE of A and A^T does not imply regular equivalence

For this reason, we decided to consider ε-BDE, whereby the equality between the degrees of related nodes in Definition 14 are relaxed by inequalities up to a given tolerance ε. Since approximately related nodes will not have equal in- and out-degrees in general, ε-BDE becomes an alternative method to compute an approximate regular equivalence.

In the following, we recast the notion of ε-BDE to the linear case related to an adjacency matrix.

Definition 14 (ε-BDE)**.** *For an $A \in \mathcal{R}^{n\times n}$ and a partition $\mathcal{H}$, we write $i \sim_{A,\mathcal{H},\varepsilon} j$ whenever there exists an $H \in \mathcal{H}$ with $i, j \in H$ such*

$$\sum_{H'\in\mathcal{H}} |\sum_{k\in H'} a_{k,i} - \sum_{k\in H'} a_{k,j}| \leq \varepsilon.$$

A partition $\mathcal{H}$ is called ε-BDE if $\mathcal{H} = \{1,\ldots,n\}/\sim^*_{A,\mathcal{H},\varepsilon}$. Here, the asterisk denotes the equivalence closure of a relation.

The relaxed definition enables the identification of regular equivalences that elude detection by the exact BDE method. To illustrate this capability, consider the example depicted in Figure 11. Here, we consider all the

edges with the same unitary weight, and for this reason, they are omitted. The target regular equivalence is $\{\{1\},\{2,3\},\{4,5,6,7,8\}\}$. In this context, the exact BDE method fails to recognize this partition because node 2 has two connections to the group $\{4,5,6,7,8\}$, while node 3 has three connections. In contrast, the ε-BDE approach successfully identifies the regular equivalence by setting ε to 1.

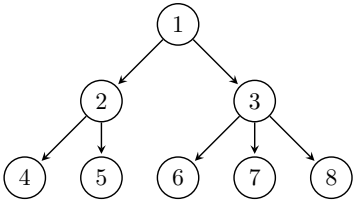

Figure 11: Example of a network where BDE fails while ε-BDE find the regular equivalence $\{\{1\},\{2,3\},\{4,5,6,7,8\}\}$ imposing ε equal to 1.

5.2 Iterative ε-BDE

As discussed in Chapter 3, an attractive feature of BDE, both in its exact and approximate variant, is that it can be computed by partition refinement [40, 39], where a given candidate initial partition of nodes is iteratively refined until the BDE criteria are satisfied. In ε-BDE, we obtain equivalence relations by closing under transitivity pairs of variables satisfying Definition 14. This transitive closure could lead to inconvenient associations. Indeed, if both pairs of nodes i, j and j, k are ε-BDE equivalent, their pairwise difference is less than ε; however, this does not mean that the difference between i and k is less than ε. To show this, let us consider the following example.

Example 13. *Pick* $\varepsilon = 0.3$ *and consider the adjacency matrix*

$$A = \begin{pmatrix} 0.4 & 0.1 & 0.5 & 0.7 \\ 0.1 & 0.5 & 0.6 & 0.7 \\ 0.5 & 0.6 & 0.3 & 0.8 \\ 0.7 & 0.7 & 0.8 & 0.3 \end{pmatrix}$$

Then, $3 \sim_{A,\mathcal{H},\varepsilon} 4$ *for* $\mathcal{H} = \{\{1,2,3,4\}\}$ *because*

$$|0.5 + 0.6 + 0.3 + 0.8 - (0.7 + 0.7 + 0.8 + 0.3)| = 0.3$$

Instead, $1 \not\sim_{A,\mathcal{H},\varepsilon} 3$ *for* $\mathcal{H} = \{\{1,2,3,4\}\}$ *since*

$$|0.4 + 0.1 + 0.5 + 0.7 - (0.5 + 0.6 + 0.3 + 0.8)| = 0.5$$

At the same time, $1 \sim^*_{A,\mathcal{H},\varepsilon} 3$ *because* $1 \sim_{A,\mathcal{H},\varepsilon} 2$ *and* $2 \sim_{A,\mathcal{H},\varepsilon} 3$*. In particular, we infer that* $\{1,\ldots,4\}/\sim^*_{A,\mathcal{H},\varepsilon} = \mathcal{H}$*, showcasing that* ε*-BDE can aggregate too much.*

To cope with this problem, we propose an iterative scheme where nodes are related by invoking the ε-BDE algorithm with increasingly larger values of ε, using an appropriate choice of initial partitions at each iteration. We call this approach *iterative* ε*-BDE* (Iε-BDE).

Let us discuss its pseudocode shown in Algorithm 4. It requires an adjacency matrix A, an initial partition $\mathcal{H}_{in}$ to keep track of the blocks discovered in the previous iterations, an initial tolerance ε_0, a step size δ, and a maximum tolerance Δ. At every iteration, the ε-BDE algorithm is invoked with the current tolerance, starting from ε_0. The algorithm refines the current partition with respect to A and A^T (lines 6-7) until no refinement is possible. The result is a partition satisfying Definition 14 on A and A^T according to the current ε. Afterward, the algorithm joins all trivial blocks of size one present in $\mathcal{H}_{\varepsilon'}$ (line 9). The intuition is to attempt node aggregation for smaller values of ε first. If that fails, i.e., nodes are eventually outputted as singleton blocks, the merging of such nodes is used to attempt aggregation for the larger ε values in the next iterations. The so-obtained partition $\mathcal{H}_{\varepsilon}$ is then used to refine the original input partition $\mathcal{H}_{in}$. Specifically, line 10 computes the coarsest partition that refines both $\mathcal{H}_{\varepsilon'}$ and $\mathcal{H}_{in}$, That is a partition with a minimal number of blocks such that each of its blocks is a subset of a block in $\mathcal{H}_{\varepsilon'}$ and a block in $\mathcal{H}_{in}$. Thereafter, line 11 and 12 update and increase, respectively, $\mathcal{H}_{\varepsilon'}$ and ε. The algorithm then iterates until the user-defined maximum tolerance Δ is reached.

Complexity. The repeat until loop of Algorithm 4 requires at most $\mathcal{O}(n^5)$ steps. Additionally, the total number of steps performed by Algo-

rithm 4 is at most $\mathcal{O}(\lceil \Delta/\delta \rceil n^5)$. We begin by noting that the while loop performs $\lceil \Delta/\delta \rceil$ iterations. Instead, each repeat until loop has at most n iterations because any partition may have at most n refinements. It thus suffices to show that Algorithm 5, inspired by the (non-approximative) forward equivalence algorithm [154], runs in at most $\mathcal{O}(n^4)$. To see this, we first note that lines 5-11 of Algorithm 5 can be computed in $\mathcal{O}(n^2)$. Then, Algorithm 5 computes in lines 13-19 the equivalence relation $\sim_{\mathcal{H},A,\varepsilon}$, where D_i is the list of nodes that are ε-BDE equivalent to node i. This portion of the code can be computed in $\mathcal{O}(n^3)$. This complexity arises due to the necessity of verifying the ε-BDE condition for every pair (i, j), requiring $\mathcal{O}(n^2)$ comparisons. The verification is performed in line 15 of Algorithm 5, involving the evaluation of differences. The cost of computing these differences is proportional to the number of blocks that, in the worst case, is $\mathcal{O}(n)$. In line 20 Algorithm 5 computes the blocks of the partition $\mathcal{H}_\varepsilon$ induced by the transitive closure $\sim^*_{\mathcal{H},A,\varepsilon}$. To do this, we consider the undirected graph induced by D_i, where node i is connected to node j if and only if i and j are ε-BDE equivalent. The blocks of the partition $\mathcal{H}_\varepsilon$ correspond to the strongly connected components of the graph and thus can be computed in $\mathcal{O}(n + m)$, where m is the number of edges. In the worst case, this implies that $\sim^*_{\mathcal{H},A,\varepsilon}$ can be computed in $\mathcal{O}(n^2)$. Finally, the main loop is repeated until no more refinement is possible. Since the partition is composed of n nodes, the number of possible refinements is bounded by n. To summarize, the overall computational complexity of Algorithm 5 is characterized by at most n iterations, thus giving rise to $\mathcal{O}(nn^3) = \mathcal{O}(n^4)$.

Remark 4. *In all experiments from Section 5.3, the repeat-until loop of Algorithm 4 executed at most twice.*

Example 14. *We next present Algorithm 4 on Example 13 with* $\varepsilon_0 = 0$, $\Delta = 0.3$, $\delta = 0.1$ *and* $\mathcal{H}_{in} = \big\{\{1, 2, 3, 4\}\big\}$. *For* $\varepsilon = 0$, *the repeat loop returns* $\mathcal{H}_{\varepsilon'} = \big\{\{1\}, \{2\}, \{3\}, \{4\}\big\}$, *making* `joinSingletons` *return* $\mathcal{H}_{in}$ *in line 9. The second iteration of the while loop with* $\varepsilon = 0.1$ *is therefore computed for the original* $\mathcal{H}_{in}$. *A similar statement can be made about* $\varepsilon = 0.1$, *meaning that the algorithm initiates its third while loop iteration with* $\varepsilon = 0.2$ *and the original* $\mathcal{H}_{in}$. *This time, nodes* 1 *and* 2 *are aggregated*

Algorithm 4 Iterative ε-BDE.

Require: Adjacency matrix A, initial partition of nodes $\mathcal{H}_{in}$, an initial tolerance ε_0, step size $\delta \geq 0$, maximum tolerance $\Delta \geq 0$.

1: $\varepsilon \longleftarrow \varepsilon_0$
2: $\mathcal{H}_{\varepsilon'} \longleftarrow \mathcal{H}_{in}$
3: **while** $\varepsilon \leq \Delta$ **do**
4: **repeat**
5: $\mathcal{H}_{\varepsilon} \longleftarrow \mathcal{H}_{\varepsilon'}$
6: $\mathcal{H}_{\varepsilon} \longleftarrow \{1, \ldots, n\}/\sim^*_{A,\mathcal{H}_{\varepsilon},\varepsilon}$ via Algorithm 5
7: $\mathcal{H}_{\varepsilon'} \longleftarrow \{1, \ldots, n\}/\sim^*_{A^T,\mathcal{H}_{\varepsilon},\varepsilon}$ via Algorithm 5
8: **until** $\mathcal{H}_{\varepsilon} = \mathcal{H}_{\varepsilon'}$
9: $\mathcal{H}_{\varepsilon'} \longleftarrow$ joinSingletons($\mathcal{H}_{\varepsilon'}$)
10: $\mathcal{H}_{in} \longleftarrow$ coarsestRefinement($\mathcal{H}_{in}$,$\mathcal{H}_{\varepsilon'}$)
11: $\mathcal{H}_{\varepsilon'} \longleftarrow \mathcal{H}_{in}$
12: $\varepsilon \longleftarrow \varepsilon + \delta$
13: **end while**
14: **return** $\mathcal{H}_{\varepsilon}$

during the repeat loop, giving rise to $\mathcal{H}_{\varepsilon'} = \big\{\{1,2\},\{3\},\{4\}\big\}$. *At this point, it is worth noting that block* $\{1,2\}$ *will not be split in any future iteration of the algorithm because the aggregation of nodes is monotonic in* ε. *The algorithm then executes* `joinSingletons` *in line 9 which yields* $\mathcal{H}_{\varepsilon'} = \big\{\{1,2\},\{3,4\}\big\}$. *Since* $\mathcal{H}_{\varepsilon'}$ *is a refinement of* $\mathcal{H}_{in}$, `coarsestRefinement` *in line 10 does not change* $\mathcal{H}_{\varepsilon'}$. *After setting* $\mathcal{H}_{in}$ *to* $\mathcal{H}_{\varepsilon'}$, *the algorithm begins its final while loop with* $\varepsilon = 0.3$ *and* $\mathcal{H}_{in} = \big\{\{1,2\},\{3,4\}\big\}$. *In it, the repeat loop aggregates nodes* 3 *and* 4 *and returns* $\mathcal{H}_{\varepsilon} = \big\{\{1,2\},\{3,4\}\big\}$, *the final result of the algorithm.*

The above example showcases why Algorithm 4 helps avoid unnecessary aggregations. Indeed, as discussed in Example 13, 0.3-BDE returns one block, while Algorithm 4 returns two blocks as outlined in Example 14. This is because aggregations arising for smaller ε-values are separated from these, which require larger ε-values (since the latter arise from singleton blocks obtained for smaller ε-values). Intuitively, δ accounts for the granularity with which Algorithm 4 aggregates nodes. A large δ may result in over-aggressive aggregations, while a small value may lead to prohibitively small blocks. In the experimental section, we evaluate

Algorithm 5 Routine for computing $\{1,\ldots,n\}/\sim^*_{\mathcal{H},A,\varepsilon}$

Require: Adjacency matrix A, partition of nodes $\mathcal{H}$, tolerance ε

$\mathcal{H}_\varepsilon \longleftarrow \mathcal{H}$
repeat
  $\mathcal{H}' \longleftarrow \mathcal{H}_\varepsilon$
  $w[i,H] \longleftarrow 0$ for all $H \in \mathcal{H}'$, $1 \le i \le n$
  for all $H \in \mathcal{H}'$ **do**
    for all $i \in H$ **do**
      for all j with $a_{i,j} \neq 0$ **do**
        $w[j,H] \longleftarrow w[j,H] + a_{i,j}$
      end for
    end for
  end for
  $D_i \longleftarrow \emptyset$ for $1 \le i \le n$
  for all $1 \le i \le n$ **do**
    for all $1 \le j \le n$ **do**
      if $\sum_{H\in\mathcal{H}'} |w[i,H] - w[j,H]| \le \varepsilon$ **then**
        insert j in D_i
      end if
    end for
  end for
  $\mathcal{H}_\varepsilon \longleftarrow$ **refine** $\mathcal{H}'$ using D to compute $\sim^*_{\mathcal{H},A,\varepsilon}$
until $\mathcal{H}_\varepsilon = \mathcal{H}'$
return $\mathcal{H}'$

Algorithm 4 on a number of benchmark networks from different fields.

5.2.1 Asymptotics for BA networks

In this section, we introduce a partition for a BA network of size $\mathcal{O}(n)$, which is approximately BDE equivalent. We begin by recalling the BA network [15].

Definition 15 (BA Model)**.** *For a given size n, the BA model is described by the stochastic process* $(G^t)_{0\le t\le n}$*, where* G^t *describes an undirected graph with nodes* $\{1,\ldots,n\}$*. Given* G^{t-1}*, we form* G^t *by adding*

node t and link node t to node i, where i is chosen randomly with

$$\mathbb{P}(i=j)=\begin{cases}\dfrac{d_{G^{t-1}}(j)}{2t-1} & ,\ 1\le j\le t-1\\ \dfrac{1}{2t-1} & ,\ j=t\end{cases}$$

Here, $d_{G^{t-1}}(j)$ denotes the degree of node j in graph G^{t-1}.

We are now ready to introduce the *BA partition*, which, inspired the theory of the BA model, divides the nodes into two groups: *celebrities* and *followers*.

Definition 16 (BA partition)**.** *Fix a partition threshold $0<\xi<1$ and generate a sample run of the stochastic process $(G^t)_{0\le t\le n}$. From this sample run, partition the set of graph nodes $\{1,\dots,n\}$ into blocks of* celebrities *$\mathcal{C}$ and* followers *$\mathcal{F}$, that is, let*

$$\mathcal{H}=\mathcal{C}\cup\mathcal{F}=\{C_1,\dots,C_\kappa\}\cup\{F_1,\dots,F_\kappa\},$$

where $\kappa=1/\xi$ and:[1]

- *celebrity nodes comprise $1,\dots,\sqrt{n}$, that is, we have:*

$$C_1\cup\ldots\cup C_\kappa=\{1,\dots,\sqrt{n}\};$$

- *follower nodes constitute $\sqrt{n}+1,\dots,n$, meaning that*

$$F_1\cup\ldots\cup F_\kappa=\{\sqrt{n}+1,\dots,n\};$$

- *$|C_\nu|=\xi\sqrt{n}$ and $C_\nu\le C_{\nu'}$ elementwise for $\nu\le\nu'$;*
- *$|F_\nu|=\xi(n-\sqrt{n})$ and $F_\nu\le F_{\nu'}$ elementwise for $\nu\le\nu'$.*

For the BA process, computing the partition is straightforward because the node identifier corresponds to its age. In the case of an instance of a scale-free network, without access to its underlying generative process, we can estimate the node's age by considering its degree and exploiting that older nodes have, on average, higher degrees than younger ones [20, 15].

[1] To simplify presentation, we assume that n is a square and that $\xi\sqrt{n}, 1/\xi\in\mathbb{N}$. The assumption can be dropped by rounding up.

For this reason, we can sort the nodes in decreasing order of their degrees and consider the first $\sqrt{N}$ as celebrities and the remaining $N - \sqrt{N}$ as followers. Then, these two groups of nodes will be split following the definition.

We next formally justify the choice of the partition from Definition 16. To this end, we recall the big-$\mathcal{O}$, big-Ω, and big-Θ notations.

Definition 17. *For two functions $f, g : \mathbb{N}_0 \to \mathcal{R}_{\geq 0}$, we define*

$$\begin{aligned} f = \mathcal{O}(g) &:\Longleftrightarrow \limsup_{n\to\infty} \tfrac{f(n)}{g(n)} < \infty \\ f = \Omega(g) &:\Longleftrightarrow \liminf_{n\to\infty} \tfrac{f(n)}{g(n)} > 0 \\ f = \Theta(g) &:\Longleftrightarrow f = \mathcal{O}(g) \text{ and } f = \Omega(g) \end{aligned}$$

The following key auxiliary result estimates the probability of finding links between nodes of a BA model and sharpens [20] by studying the error terms in the proof of [20, Lemma 2].

Lemma 6. *In a BA model $(G^t)_{0\leq t\leq n}$, let g_j with $j \leq n$ denote the node to which node j connects to. Then for all $i < j < k$, we have*

$$\begin{aligned} \mathbb{P}(g_j = i) &= \frac{1}{2}\frac{1}{\sqrt{ij}} + \mathcal{O}\Big(\frac{1}{ij}\Big) \\ \mathbb{P}(g_j = i, g_k = i) &= \frac{1}{2}\frac{1}{i\sqrt{jk}} + \mathcal{O}\Big(\frac{1}{i\sqrt{ijk}}\Big) \end{aligned}$$

Moreover, it holds that

$$\mathbb{P}(g_j = i, g_k = i) - \mathbb{P}(g_j = i)\mathbb{P}(g_k = i) = \frac{1}{4i\sqrt{jk}} + \mathcal{O}(\frac{1}{i\sqrt{ijk}})$$

For a block H of some given partition of nodes $\mathcal{H}$, we shall next study the number of H-in-degree of a node i. Formally, this is captured by the random variable $\sum_{j\in H} X^{i,j}$, where

$$X^{i,j} = \begin{cases} 1, & g_j = i \\ 0, & \text{otherwise} \end{cases} \tag{5.1}$$

Revisiting Lemma 6, we note that the in-degree of a node can be approximated by a sequence of Bernoulli trials. However, while the first identity

of Lemma 6 suggests to approximate $X^{i,j}$ by a Bernoulli variable with success probability $1/2\sqrt{ij}$, the third statement shows that variables $X^{i,j}$ and $X^{i,k}$ are positively correlated, hence stochastically dependent. This prevents direct applications of the law of large numbers or the central limit theorem.

The next result studies the number of links between the blocks of $\mathcal{H}$.

Theorem 14. *Let $(G^t)_{0\le t\le n}$ be a BA model and $\mathcal{H}$ as in Definition 16. Then, the number of links between celebrities and followers is of order $\sqrt{\xi}\sqrt[4]{n}$, that is*

$$\mathbb{E}[\deg(i,H)] = \Theta(\sqrt{\xi}\sqrt[4]{n}), \quad H \in \mathcal{F},\ i = \gamma\sqrt{n} \in C_\nu,\ C_\nu \in \mathcal{C}$$

for some $0 < \gamma < 1$; in all other cases, the number of links is $\mathcal{O}(1)$, i.e., negligible because it does not grow with n.

After studying the connectivity between the blocks of $\mathcal{H}$, we are ready to state our main result.

Theorem 15. *Let $G = (G^t)_{0\le t\le n}$ be a BA model and $\mathcal{H}$ as in Definition 16. Then, for large n, partition $\mathcal{H}$ is on average a $\sqrt{\xi}$-BDE of the scaled network $G/\sqrt[4]{n}$. Specifically, for any $0 < \gamma, \gamma' < 1$ and $H \in \mathcal{H}$, for large $n \ge 1$, we have*

$$\frac{1}{\sqrt[4]{n}}|\mathbb{E}[\deg(i,H)] - \mathbb{E}[\deg(i',H)]| = \mathcal{O}(\sqrt{\xi}|\gamma' - \gamma|),$$

where

- *either $i = \gamma\sqrt{n}$ and $i' = \gamma'\sqrt{n}$ such that $i, i' \in \bar{H} \in \mathcal{C}$;*
- *or $i = \gamma n$ and $i' = \gamma' n$ such that $i, i' \in \bar{H} \in \mathcal{F}$.*

A strengthening of the above result which would ensure that the difference of degrees (rather than their expected values) is with high probability of order $\mathcal{O}(|\gamma - \gamma'|)$ is difficult. Indeed, as stated in the next result, the variance of the difference $(\deg(i,H) - \deg(i',H))/\sqrt[4]{n}$ does not vanish as n increases.

Theorem 16. *In a BA model $(G^t)_{0\leq t\leq n}$. Then, for any $H \in \mathcal{H}$ and $i \neq i'$, it holds that:*

$$\mathbb{V}[\deg(i,H) - \deg(i',H)] \geq \mathbb{V}[\deg(i,H)] + \mathbb{V}[\deg(i',H)]$$

and

$$\mathbb{V}[\deg(i,H)] = \begin{cases} \mathcal{O}(1), & i \in C_\nu,\ H \in \mathcal{C} \\ \Theta(\sqrt{n}), & i \in C_\nu,\ H \in \mathcal{F} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{C} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{F} \end{cases}$$

We proved that the BA partition is asymptotically an ε-BDE partition for binary scale-free networks. By identifying important (celebrity) and less important nodes (followers) in scale-free networks, practitioners can use this result as a starting point in the analysis of the network. In particular, the BA partition can be used to avoid an aggressive reduction of the network, a well-known issue for regular equivalence on binary networks [23]. In the experimental section, we show how Iε-BDE and CATREGE can employ the BA partition to improve the results on binary networks.

5.3 Experiments

In this section, we show the effectiveness of our approach against the state of the art. For indirect approaches, we consider REGE and CATREGE; for direct approaches, we consider different variants of blockmodeling. To make a fair comparison, we analyzed the algorithms by keeping the same level of granularity for all. This is controlled by incomparable parameters δ/Δ for Iε-BDE and the number of clusters for the competing techniques. Thus, we first fixed Iε-BDE parameters in a uniform manner across all networks, as detailed next; then we chose the number of clusters of direct and indirect approaches equal to the number of partition blocks returned by Iε-BDE.

5.3.1 Experimental Set-up

We consider benchmark networks of different sizes as listed in Table 12. The networks are divided into binary and weighted networks.

Iterative ε-BDE We implemented the Iε-BDE in a prototype that uses the implementation of ε-BDE already available in the software tool ERODE [37]. The two parameters for Iε-BDE are the maximum tolerance Δ and the step size δ. For an unbiased, model-independent choice of these parameters, we considered the following heuristic: since ε-BDE relates nodes with similar row-sums of the adjacency matrix A, we set up Δ by picking a value roughly equal to the average sum of the rows of A. For δ, instead, we took a value of one order of magnitude smaller than the average value of the non-zero entries of A, considering only the weights greater than 0. When this value is less than the minimum non-zero entry in A, we set up δ equal to this minimum. The values of δ and Δ are listed in Table 12. With this choice, Iε-BDE was run using $\varepsilon_0 = 0$ and, unless otherwise stated, with the singleton initial partition considering all nodes. We use this output to compare against the other approaches.

REGE For weighted networks, we compare against the indirect method of REGE as implemented in the R package presented in [102]. Following [163], we set up a number of iterations for every model equal to 100. Then, we employ the REGE's similarity matrix to compute a partition with the same number of blocks returned by Iε-BDE. To do this, we use the dendrogram associated with hierarchical clustering, which was computed with the scikit-learn library [118]. In this case, following the literature on this subject [162], we consider two linkages for hierarchical clustering, i.e., single and complete links [162] (denoted by *REGE+SL* and *REGE+CL* in the forthcoming tables, respectively).

CATREGE For binary networks, we considered the indirect method of CATREGE, since it achieves superior performance with respect to REGE [24]. We used the implementation in UCINET [25]. CATREGE allows specifying an initial partition. In the first iteration, it divides the

Weighted networks						*Iε-BDE*		
Model	*Ref.*	*n*	*REGE*	0-*BDE*	*Size*	δ	Δ	*Ratio*
EIES	[65]	32	32	32	19	10^0	300	0.59
Windsurfers	[88]	43	43	43	20	10^0	60	0.47
Ecosystem	[88]	128	128	128	37	10^{-1}	30	0.29
FaoTrade	[54]	214	214	214	170	10^2	$2 \cdot 10^4$	0.79
WTN	[68]	226	226	226	171	10^4	10^7	0.75
CElegans	[157]	306	286	286	207	10^0	40	0.67
USairport	[49]	500	500	500	223	10^4	10^6	0.45
FB	[113]	1899	—	1857	1310	10	1000	0.69
							Average ratio:	**0.59**

Binary networks						*Iε-BDE*				Iε-BDE / BA
Model	*Ref.*	*n*	*CATREGE*	0-*BDE*	*Size*	δ	Δ	*Ratio*	*Size*	*Ratio*
Karate	[88]	34	30	30	16	10^0	4	0.47	24	0.71
GD	[18]	73	54	54	17	10^0	2	0.23	34	0.47
Revolution	[88]	136	56	56	11	10^0	2	0.11	46	0.34
Email	[88]	167	166	166	14	10^0	30	0.08	45	0.27
Physician	[88]	241	241	241	3	10^0	5	0.01	195	0.80
FilmTrust	[88]	874	—	673	238	10^0	2	0.27	371	0.42
BlogCatalog	[161]	10312	—	10106	5455	10^0	60	0.53	5490	0.53
Youtube	[161]	15088	—	12691	4760	10^0	10	0.32	6766	0.45
							Average ratios:	**0.25**		**0.50**

Table 12: Parameters and results for iterative ε-BE.

nodes following the given partition instead of considering all of them in the same block. The nodes in different blocks cannot be associated in the following iterations. We opted to utilize the BA partition for all binary networks to prevent CATREGE from erroneously identifying all nodes as equivalent, as highlighted in [24]. Similar to the approach taken with REGE, we applied hierarchical clustering to the resultant similarity matrix. This strategic adjustment ensures a more accurate and reliable analysis, mitigating the risk of aggressive aggregation.

Blockmodeling For direct approaches, we compare against the binary, valued, and homogeneity blockmodeling approaches [164], using the R package by Ziberna et al. [102]. Since these techniques are based on a local optimization algorithm, we set 1000 repetitions/different starting partitions to check. Valued and homogeneity blockmodeling identifies for f-regular equivalence, where the function f was set to max, which is the common setting for regular equivalence [102]. Valued blockmodeling re-

quires specifying a parameter m that distinguishes between prominent and non-prominent weights. The best way to determine m is to have prior knowledge about the network. In the absence of this, it is possible to choose the value of m considering the distribution of the links. Following [102, 164], we set m equal to the median and the mode of the nonzero weights.

Error metric In evaluating the precision of various methods for identifying approximately equivalent nodes, we adopt a similarity measure based on PageRank, as suggested by prior research [152]. We compute the maximum PageRank difference across all node pairs within each approximately regular equivalent block. Subsequently, we derive the minimum, average, and maximum difference across all blocks. Thus, lower values of such indices correspond to more similar (in terms of regular equivalence) blocks.

Timeout Throughout all experiments, we set a 3-hour timeout for each analysis.

5.3.2 Results

Preliminary analysis We conducted an analysis to determine the number of regularly equivalent node blocks using both REGE and CATREGE. These blocks comprise nodes that strictly satisfy the regular equivalence definition. Our findings reveal that regularly equivalent nodes are rare, particularly in the case of weighted networks. Table 12 provides an overview of the quantities of regularly equivalent nodes identified. Notably, in all weighted networks except CElegans, we observed the absence of nontrivial blocks, indicating the absence of regularly equivalent nodes. However, for CElegans, REGE identified 286 blocks. It's worth mentioning that CATREGE yielded a notable number of equivalent nodes; however, due to limitations, larger networks could not be analyzed, as CATREGE supports networks with a maximum of 256 nodes. For comparison purposes, we also explored regular equivalences using Iε-BDE by

setting $\Delta = 0$ (denoted as 0-BDE in the table column). This configuration aligns with the condition outlined in Theorem 13. Interestingly, although Theorem 13 presents a sufficient condition for regular equivalence, our analysis indicates that the two notions are indistinguishable across the analyzed networks, as both algorithms returned the same number of blocks.

Weighted Networks				
	Errors			
Method	*Min*	*Avg*	*Max*	*Times (s)*
EIES				
Iε-BDE	**2.23E–4**	**3.31E–3**	**9.34E–3**	2.62
REGE+CL	1.05E–3	8.56E–3	3.15E–2	**0.23**
REGE+SL	1.50E–3	1.26E–2	3.72E–2	**0.23**
Blockmod. Hom.	1.50E–3	5.30E–3	8.27E–3	319.86
Blockmod. Val. median	3.15E–2	4.65E–2	6.15E–2	142.74
Blockmod. Val. mode	7.12E–4	2.06E–2	8.30E–2	115.23
Windsurfers				
Iε-BDE	**5.66E–6**	**4.46E–3**	1.50E–2	1.47
REGE+CL	1.31E–4	6.36E–3	3.85E–2	**0.46**
REGE+SL	2.10E–4	7.41E–3	3.85E–2	**0.46**
Blockmod. Hom.	1.93E–3	5.67E–3	**1.41E–2**	532.80
Blockmod. Val. median	1.99E–3	3.78E–2	7.83E–2	291.55
Blockmod. Val. mode	1.31E–3	1.86E–2	3.85E–2	300.61
Ecosystem				
Iε-BDE	**6.89E–8**	**7.61E–4**	**2.89E–3**	**18.13**
REGE+CL	6.89E–8	1.27E–2	2.18E–1	18.20
REGE+SL	6.89E–8	1.70E–3	8.16E–3	18.20
FaoTrade				
Iε-BDE	**0.0**	**3.47E–4**	**1.38E–3**	**23.53**
REGE+CL	0.0	2.68E–3	2.89E–2	503.58
REGE+SL	0.0	5.23E–4	2.89E–2	503.58
WTN				
Iε-BDE	**2.42E–7**	**7.55E–5**	**4.35E–4**	**154.95**
REGE+CL	2.42E–7	2.44E–3	3.42E–2	857.40
REGE+SL	2.96E–5	3.69E–3	5.3E–2	857.40
CElegans				
Iε-BDE	**0.0**	**5.61E–4**	3.46E–3	**3.54**
REGE+complete link	0.0	6.47E–4	**2.06E–3**	75.30
REGE+single link	0.0	7.56E–4	3.40E–3	75.30
USairport				
Iε-BDE	**1.25E–7**	**1.92E–4**	**3.45E–3**	**13.73**
REGE+CL	1.32E–6	1.30E–3	3.27E–2	408.60
REGE+SL	1.52E–7	1.64E–3	4.02E–2	408.60
FB				
Iε-BDE	**0.0**	**3.21E–5**	**5.25E–4**	**47.46**

Binary Networks				
	Errors			
Method	*Min*	*Avg*	*Max*	*Times (s)*
Karate				
BA partition	2.37E–5	2.90E–3	7.62E–3	—
Iε-BDE	**0.0**	**1.17E–3**	**5.48E–3**	**0.28**
CATREGE+CL	0.0	1.53E–3	5.48E–3	1.00
CATREGE+SL	0.0	2.41E–3	7.62E–2	1.00
Blockmodeling	0.0	4.40E–3	1.23E–2	84.6
GD				
BA partition	0.0	3.09E–3	6.06E–3	—
Iε-BDE	**0.0**	**1.04E–3**	**4.89E–3**	**0.12**
CATREGE+CL	0.0	2.34E–3	6.06E–3	1.00
CATREGE+SL	0.0	2.12E–3	6.06E–3	1.00
Blockmodeling	4.54E–3	2.18E–2	9.92E–2	1441.8
Revolution				
BA partition	1.63E–4	1.38E–3	1.01E–2	—
Iε-BDE	**0.0**	**1.02E–4**	2.63E–3	**0.32**
CATREGE+CL	0.0	5.71E–4	1.01E–2	1.00
CATREGE+SL	0.0	5.57E–4	**1.01E–3**	1.00
Email				
BA partition	0.0	1.23E–3	2.96E–3	—
Iε-BDE	**0.0**	**6.71E–4**	**2.96E–3**	2.11
CATREGE+CL	0.0	9.99E–4	2.96E–3	**1.00**
CATREGE+SL	0.0	9.93E–4	2.96E–3	**1.00**
Physician				
BA partition	0.0	3.95E–3	1.38E–2	—
Iε-BE	**0.0**	**4.92E–4**	**3.28E–3**	**0.77**
CATREGE+CL	0.0	3.12E–3	1.38E–2	1.00
CATREGE+SL	0.0	3.32E–3	6.93E–3	1.00
FilmTrust				
BA partition	0.0	3.25E–3	1.19E–2	—
Iε-BDE	**0.0**	**1.79E–4**	**3.41E–3**	**1.46**
BlogCatalog				
BA partition	1.33E–5	4.47E–4	2.29E–3	—
Iε-BDE	**0.0**	**2.86E–6**	**2.50E–5**	**1250.0**
Youtube				
BA partition	5.83E–5	5.05E–4	1.83E–3	—
Iε-BDE	**0.0**	**5.77E–6**	**1.21E–4**	**114.04**

Table 13: Comparison on weighted (left) and binary (right) networks. Best results in bold; methods that timed out are not listed.

Weighted networks Table 12 shows the number of approximately equivalent blocks of nodes identified by Iε-BDE with the chosen parameters, highlighting that it halves, on average, the network size. The comparison against REGE and blockmodeling for weighted networks is reported in

Table 13 (left). Iε-BDE proved generally more accurate and can yield errors up to one order of magnitude smaller. We also observe that homogeneity blockmodeling performs better than REGE and valued blockmodeling. Blockmodeling requires a considerable amount of time, making it applicable in these examples to networks up to 43 nodes within the given threshold. REGE's implementation is faster than Iε-BDE for small networks. We remark that FaoTrade and WTN, despite being similar in size, are characterized by considerably different runtimes. This is attributed to the density of the network. For larger networks, Iε-BDE proved faster, justifying the differences in asymptotic cost complexity of the two algorithms; in practice, REGE could not analyze the FB network within the timeout.

Binary networks With the settings of Table 12, Iε-BDE identifies nodes more aggressively, on average, in binary networks than in weighted networks (ratios 0.25 and 0.59, respectively). We now consider Iε-BDE initialized with the BA partition, following Definition 16. Here, we set $\xi = 0.1$ for all networks, always leading to 20 blocks. With this setting, and using the same values of δ and Δ, the average ratio using the BA partition becomes comparable to that of weighted networks (last two columns of Table 12).

Table 13 (right) shows the comparison for binary networks. These results were obtained when initializing Iε-BDE with the BA partition for a fair analysis against CATREGE. For reference, we also report the error statistics directly computed on the BA partition. Since, in all cases, both the CATREGE and the Iε-BDE results refine the BA partition, their error metrics are consistently improved. However, we remark that, for small networks, the BA partition already provides errors within the same order of magnitude as the iterative algorithms; for larger networks, the error statistics of the BA pre-partition obviously deteriorate because, by fixing the number of blocks, it clusters increasingly more nodes.

Overall, in the networks where the comparison is possible, Iε-BE yields superior precision than CATREGE and blockmodeling. CATREGE may be faster than Iε-BDE in some cases, but it does not support networks

larger than 256 nodes, as discussed. The blockmodeling results confirm the scalability issues observed in weighted networks, timing out already with 136 nodes.

Chapter 6

Conclusion

In this thesis, we proposed two novel techniques to reduce nonlinear dynamical systems.

We developed notions of equivalence as a relaxation of the exact counterparts, allowing the derivatives of related ODE variables to vary up to a desired tolerance. Our algorithmic approach can be useful to systematically discover quasi-symmetries in different case studies. It is able to reduce the model, imposing a certain error within the dynamics. We also propose a formal bound that is able to estimate the maximum error in the reference model. In this context, our approach serves as a complement to well-established over-approximation methods such as CORA. Our method performs better with small uncertainties on larger models, while CORA is more effective with larger uncertainties on smaller models. As a result, our approach is more computationally efficient and capable of producing satisfactory results in a reasonable time. The main limitation is that the bounds account only for small perturbations of the parameters. In future work, we have to overcome this weakness in order to provide tighter bounds for larger perturbations.

In the second work, we proposed 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 homogeneity across model parameters in

dependence on a given numerical threshold parameter, the algorithm constructs a system of differential inequalities. The differential hull is guaranteed to over-approximate the original ODE system in the presence of uncertain/noisy parameters and can be reduced by exploiting the BDE technique. Also, in these cases, we tested the approach against CORA in different case studies. Similarly to the previous case, the results highlight a trade-off between the quality of the approximation and the computational cost of computing it. This approach also represents a solution with respect to the problem that arises in the first approach. Indeed, it is able to provide formal bounds for larger perturbation with respect to ε-BDE. The main limitation is that the algorithm proposed is limited to ODE systems with positive solutions. As part of future work, we intend to extend the algorithm for negative solutions.

Finally, we presented the Iε-BDE, a new method to compute approximate regular equivalences for networks based on a partition refinement algorithm. In most examples, it showed superior precision and performance with respect to the state of the art, enabling the analysis of networks that are beyond the scope of applicability of currently available direct and indirect approaches. The asymptotic result for scale-free (binary) networks provides a pre-partitioning heuristic for practical models that avoid the problem of aggressive clustering of nodes. A possible extension would be to develop a similar result for other classes of distributions, starting from scale-free weighted networks. Building on recent works on network embedding [82], it will be also interesting to study whether the characterization of approximate regular equivalence as a quantitative bisimulation paves new ways of encoding role relationships in lower-dimensional spaces.

Addressing the reduction of nonlinear dynamical systems has proven to be a great challenge, especially nowadays, where the models have become more complex and their analysis is computationally expensive. Our research has primarily centered on three key aspects: the accuracy of the reduced models, the efficiency of the reduction algorithms, and ensuring a broad range of applications. Notably, all the methods presented within this thesis offer provable formal bounds. The approximate differential equivalences are equipped with a formal bound while the reduced differ-

ential hull approach represents the bounds as a dynamical system. This is an extremely useful result, especially in applications where stringent control over the trajectory error of the reduced model is a fundamental requirement. Regarding computational efficiency, almost all the approaches proposed present a polynomial complexity. The only exception is ε-FDE, although as explained in Chapter 3 is in practice efficient. We tested the proposed techniques on several case studies spreading in different fields of research. Practically every problem that can be expressed by means of nonlinear differential equations is suitable for the application of our approaches. In this direction, we interpret the problem of finding regular equivalences on a network as reduction of a dynamical systems. We show how ε-BDE can be tailored to discover regular equivalences on networks. The results show the effectiveness and scalability of our approach with respect to the methods in the literature.

In conclusion, this thesis achieves the proposed objectives and represents a step forward in the challenging field of nonlinear dynamical system reduction.

Appendix A

Appendix Chapter 3

The appendix is taken from the published work [34].

A.1 Proofs

This appendix collects all the formal proofs of the results stated in the paper.

Proof of Theorem 1.

Proof. Assume that $\mathcal{H}_1, \ldots, \mathcal{H}_n$ are ε-FDE partitions of $\mathcal{S}$ and define $\sim_l := \sim^F_{\mathcal{H}_l,\varepsilon} \cap \sim_{\mathcal{H}_l}$ and $\sim := \sim_{\mathcal{H}}$, where $\mathcal{H} := \mathcal{S}/\big(\bigcup_{l=1}^n \sim_l\big)^*$. Note that the definition of ε-FDE implies that $\mathcal{H}_l = \mathcal{S}/\sim_l^*$ for all $1 \leq l \leq n$. Let us fix arbitrary $1 \leq l \leq n$ and $x_i \sim_l x_j$. It can be easily seen that for any $H \in \mathcal{H}$ there exist unique blocks $G^H_1, \ldots, G^H_{m_H} \in \mathcal{H}_l$ such that

$\biguplus_{k=1}^{m_H} G_k^H = H$. With this, it holds that

$$\begin{aligned}\sum_{H\in\mathcal{H}}\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}}|c(\wp_{i,j}^H,x^\alpha)| &= \sum_{H\in\mathcal{H}}\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}}|c(\sum_{k=1}^{m_H}\wp_{i,j}^{G_k^H},x^\alpha)|\\ &\le \sum_{H\in\mathcal{H}}\sum_{k=1}^{m_H}\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}}|c(\wp_{i,j}^{G_k^H},x^\alpha)|\\ &= \sum_{G\in\mathcal{H}_l}\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}}|c(\wp_{i,j}^{G},x^\alpha)|\\ &\le \varepsilon,\end{aligned}$$

where the first estimation follows from the triangle inequality, while the second estimation is thanks to the definition of $\sim_l$. The above readily implies that $\mathcal{H}$ is an ε-FDE partition.

We now turn to the case of ε-BDE. Similarly to the ε-FDE case, we assume that $\mathcal{H}_1,\dots,\mathcal{H}_n$ are ε-BDE partitions of $\mathcal{S}$ and $\sim_l$, $\sim:=\sim_{\mathcal{H}}$ and $\mathcal{H}$ are as above. Note that the definition of ε-BDE implies that $\mathcal{H}_l=\mathcal{S}/\sim_l^*$ for all $1\le l\le n$. For arbitrary $1\le l\le n$ and $x_i\sim_l x_j$, it holds that

$$\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}}}|c(\wp_{i,j}^{\mathcal{H}},x^\alpha)|\le\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}}}|c(\wp_{i,j}^{\mathcal{H}_l},x^\alpha)|$$

since $\mathcal{H}_l$ refines $\mathcal{H}$. The above implies that $\mathcal{H}$ is an ε-BDE partition.

So far, we have shown that the coarsening $\mathcal{S}/\big(\bigcup_{l=1}^n\sim_l\big)^*$ of ε-FDE/BDE partitions $\mathcal{H}_1,\dots,\mathcal{H}_n$ is again an ε-FDE/BDE partition. The claim follows by noting that Lemma 26 in [38] ensures that $\mathcal{S}/\big(\bigcup_{l=1}^n\sim_l\big)^*$ is a refinement of $\mathcal{G}$ if each $\mathcal{S}/\sim_l$ is a refinement of $\mathcal{G}$. □

Proof of Lemma 1.

Proof. Let us assume that $x_i\sim^F_{\mathcal{H},\varepsilon}x_j$, which is equivalent to

$$\sum_{H\in\mathcal{H}}\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}}|c(\wp_{i,j}^H,x^\alpha)|\le\varepsilon$$

Since $\mathcal{H}$ is a refinement of $\mathcal{G}$, for any $G \in \mathcal{G}$ there exist unique blocks $H_1^G, \ldots, H_{m_G}^G \in \mathcal{H}$ such that $\biguplus_{k=1}^{m_G} H_k^G = G$. With this, it holds that

$$\begin{aligned}
\sum_{G\in\mathcal{G}} \sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}} |c(\wp_{i,j}^G, x^\alpha)| &= \sum_{G\in\mathcal{G}} \sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}} |c(\sum_{k=1}^{m_G} \wp_{i,j}^{H_k^G}, x^\alpha)| \\
&\leq \sum_{G\in\mathcal{G}} \sum_{k=1}^{m_G} \sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}} |c(\wp_{i,j}^{H_k^G}, x^\alpha)| \\
&= \sum_{H\in\mathcal{H}} \sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}\dot\cup\{s\}}} |c(\wp_{i,j}^{H}, x^\alpha)| \\
&\leq \varepsilon,
\end{aligned}$$

thus showing $x_i \sim_{\mathcal{G},\varepsilon}^F x_j$. This yields the first statement. Let us now assume that $x_i \sim_{\mathcal{H},\varepsilon}^B x_j$ which corresponds by definition to $\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}}} |c(\wp_{i,j}^{\mathcal{H}}, x^\alpha)| \leq \varepsilon$. Moreover,

$$\sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}}} |c(\wp_{i,j}^{\mathcal{G}}, x^\alpha)| \leq \sum_{\alpha\in\mathbb{N}_0^{\mathcal{S}}} |c(\wp_{i,j}^{\mathcal{H}}, x^\alpha)|$$

because $\mathcal{H}$ is a refinement of $\mathcal{G}$. Hence, we infer that $x_i \sim_{\mathcal{G},\varepsilon}^B x_j$. This readily implies the second statement. □

Proof of Theorem 2.

Proof. Let $\mathcal{G}'$ denote the coarsest ε-FDE (ε-BDE) partition that refines $\mathcal{H}_0 := \mathcal{G}$ and set $\mathcal{H}_{k+1} := \mathcal{S}/(\sim_{\mathcal{H}_k,\varepsilon}^{\chi^*} \cap \sim_{\mathcal{H}_k})$ for all $k \geq 0$. Then, the sequence $(\mathcal{H}_k)_{k\geq 0}$ is such that $\mathcal{G}'$ is a refinement of $\mathcal{H}_k$ for all $k \geq 1$. We prove this by induction on k.

- $k = 1$: Since $\mathcal{G}'$ is a refinement of $\mathcal{H}_0$, Lemma 1 ensures the first claim.
- $k \to k+1$: Thanks to the fact that $\mathcal{G}'$ is a refinement of $\mathcal{H}_k$ by induction, Lemma 1 ensures the first claim.

From the fact that $\mathcal{G}'$ is a refinement of any $\mathcal{H}_k$, we conclude that $\mathcal{G}' = \mathcal{H}_k$ whenever $\mathcal{H}_k$ is an ε-FDE (ε-BDE) partition. Since $\mathcal{H}_k$ is a refinement of $\mathcal{H}_{k-1}$ for all $k \geq 1$ and $\mathcal{S}$ is finite, we can fix the smallest $k \geq 1$ such that $\mathcal{H}_k = \mathcal{H}_{k-1}$. This, in turn, implies that $\mathcal{H}_{k-1} = \mathcal{H}_k = \mathcal{S}/(\sim_{\mathcal{H}_{k-1},\varepsilon}^{\chi^*} \cap \sim_{\mathcal{H}_{k-1}})$. □

Proof of Theorem 4.

Proof. Fix a partition $\mathcal{H}$ and consider the system of equations (3.4)

$$\left\{\mathfrak{t}(\tilde{\wp}_{i,j}^{H}, x^{\alpha}, \mathcal{S} \cup \{s\}) = 0 \mid \alpha \in \mathbb{N}_0^{\mathcal{S}\cup\{s\}}, H \in \mathcal{H} \text{ and } x_i \sim_{\mathcal{H}} x_j\right\}$$

By definition, $\mathfrak{t}(\tilde{\wp}_{i,j}^{H}, x^{\alpha}, \mathcal{S} \cup \{s\}) = 0$ states that the coefficient of monomial x^{α} in polynomial $\tilde{\wp}_{i,j}^{H}$ is zero. Likewise, by definition, $\mathcal{H}$ is an FDE if and only if all polynomials $\tilde{\wp}_{i,j}^{H}$, where $x_i \sim_{\mathcal{H}} x_j$, are zero. Since a polynomial is zero if and only if all its coefficients are zero, we thus obtain that any choice of parameters satisfying the above set of equations yields an FDE. The proof for BDE proceeds along the same lines. □

Proof of Theorem 6.

Proof. See [122, Theorem 1, Section 1.10]. □

Proof of Theorem 8.

Proof. It can be efficiently checked whether $\mathcal{P}(\hat{\sigma}_*)$ exhibits singularities on $[0;T]$ by using a numerical ODE solver. Hence, if $\mathcal{P}(\hat{\sigma}_*)$ is singularity free, the bound which is to be proven next ensures that $\mathcal{P}(\hat{\sigma}_1)$ is singularity free as long as $\|x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)\| \leq \delta$.

Let $\delta = \|x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)\|$ satisfy $\sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k \delta_+^{k-1} < (2\lambda_1\hat{\tau})^{-1}$ for $\delta_+ = 2\lambda_0\delta$. Since $\delta < \delta_+$, it holds that

$$0 < \tau(\delta) = \inf\{0 \leq t \leq \hat{\tau} \mid \|x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)\| \geq \delta_+\},$$

where $\inf \emptyset := \infty$ as usual. With this, it holds that

$$\left\| \int_0^t \Lambda(s,t) r(s, x^{\hat{\sigma}_1}(s) - x^{\hat{\sigma}_*}(s)) ds \right\| \leq \lambda_1 t \sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k \delta_+^k$$

for all $t \leq \min\{\tau(\delta), \hat{\tau}\}$. Hence, Theorem 9 and 7 yield

$$\|x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)\| \leq \lambda_0\delta + \lambda_1\hat{\tau} \sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k \delta_+^k =$$

$$\frac{\delta_+}{2} + \lambda_1\hat{\tau} \sum_{k=2}^{\deg(\hat{\mathcal{P}})} d_k \delta_+^k < \delta_+$$

for all $t \leq \min\{\tau(\delta), \hat{\tau}\}$, where the last inequality follows from straightforward algebraic manipulation. This implies that $\tau(\delta) = \infty$, thus showing that $\|x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t)\| < 2\lambda_0 \|x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0)\|$ for all $0 \leq t \leq \hat{\tau}$ because $\delta_+ = 2\lambda_0 \delta$. □

Proof of Theorem 9.

Proof. As pointed out above, in the case of $\deg(\hat{\mathcal{P}}) = 1$, it holds that $r \equiv 0$. This and Theorem 6 imply $x^{\hat{\sigma}_1}(t) - x^{\hat{\sigma}_*}(t) = \Delta x(t) = \Lambda(0,t)\Delta x(0) = \Lambda(0,t)(x^{\hat{\sigma}_1}(0) - x^{\hat{\sigma}_*}(0))$ for any $x^{\hat{\sigma}_1}(0) \in \mathbb{R}^{\mathcal{S}\cup\Theta}$ and $0 \leq t \leq \hat{\tau}$, thus yielding the first claim. The second claim, instead, follows by noting that

$$\max_{0\leq t\leq\hat{\tau}} \max_{\|\Delta x(0)\|=1} \|\Delta x(t)\| = \max_{0\leq t\leq\hat{\tau}} \max_{\|\Delta x(0)\|=1} \|\Lambda(0,t)\Delta x(0)\| = \max_{0\leq t\leq\hat{\tau}} \|\Lambda(0,t)\| = \lambda_0$$

□

Proof of Lemma 2.

Proof. Fix $0 \leq s \leq t \leq \hat{\tau}$ and assume that $|s-t| \leq \Delta$. Since $(\partial_t \Lambda)(s,t) = A(t)\Lambda(s,t)$, it can be shown (see Lemma 1 and 2 in [78]) that $\|\Lambda(s,t)\| \leq e^{L(t-s)} \leq e^{L\Delta t}$ and the claim holds true. Using this, it can also be seen that

$$\|\Lambda(0,t)\| = \|\Lambda(t_i,t)\Lambda(0,t_i)\| \leq \lambda_0^+ e^{L\Delta t}$$

for some $t_i \leq t$ such that $t - t_i \leq \Delta t$. Let us now assume that $|s-t| > \Delta t$. Then there exist $t_i \leq t_j$ such that $s \leq t_i \leq t_j \leq t$ with $t_i - s, t - t_j \leq \Delta$. With this, it holds that

$$\|\Lambda(t_i,t_j) - \Lambda(s,t)\| \leq \|\Lambda(t_i,t_j) - \Lambda(t_i,t)\| + \|\Lambda(t_i,t) - \Lambda(s,t)\|$$

The bound on λ_1 then follows by noting that

$$\begin{aligned}\|\Lambda(t_i,t_j) - \Lambda(t_i,t)\| &= \|\Lambda(t_i,t_j) - \Lambda(t_j,t)\Lambda(t_i,t_j)\| \\ &= \|(\Lambda(t_j,t_j) - \Lambda(t_j,t))\Lambda(t_i,t_j)\| \\ &\leq \lambda_1^+ L\Delta t\end{aligned}$$

and

$$\begin{aligned}\|\Lambda(t_i,t) - \Lambda(s,t)\| &= \|\Lambda(t_i,t) - \Lambda(t_i,t)\Lambda(s,t_i)\| \\ &= \|\Lambda(t_i,t)(\Lambda(s,s) - \Lambda(s,t_i))\| \\ &= \|\Lambda(t_j,t)\Lambda(t_i,t_j)(\Lambda(s,s) - \Lambda(s,t_i))\| \\ &\leq e^{L\Delta t}\lambda_1^+ L\Delta t.\end{aligned}$$

□

Proof of Lemma 3.

Proof. For $\deg(\hat{\mathcal{P}}) = 3$, the constraint writes as $d_2(2\lambda_0\delta)^2 + d_3(2\lambda_0\delta)^1 \leq (2\lambda_1\hat{\tau})^{-1}$. Since the left-hand side is monotonic increasing in δ, it thus suffices to solve the quadratic equation $d_2(2\lambda_0\delta)^2 + d_3(2\lambda_0\delta)^1 = (2\lambda_1\hat{\tau})^{-1}$. The following formula, known as Muller's variant, can be readily checked to solve a quadratic equation $ax^2 + bx + c$:

$$x = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

Matching the coefficients gives the claim. The claim for $\deg(\hat{\mathcal{P}}) = 2$, instead, is trivial. □

Proof of Lemma 4.

Proof. Fix some $x_i \in \mathcal{S} \cup \Theta$ and $0 \leq s \leq \hat{\tau}$. Then, the multidimensional Taylor expansion of $\hat{q}_i$ at point $x_s := x^{\hat{\sigma}_*}(s)$ is given by

$$\hat{q}_i(x) = \sum_{|\alpha| \leq \deg(\hat{\mathcal{P}})} \frac{(D^\alpha \hat{q}_i)(x_s)}{\alpha!}(x - x_s)^\alpha,$$

where $D^\alpha = \frac{\partial^{|\alpha|}}{\prod_{x_i \in \mathcal{S}} \partial x_i^{\alpha_{x_i}}}$ is the standard compact notation of the partial derivative underlying the multi-index $\alpha \in \mathbb{N}_0^{\mathcal{S}\cup\Theta}$ with $\alpha! = \prod_{x_i \in \mathcal{S}\cup\Theta} \alpha_{x_i}!$ and $|\alpha| = \sum_{x_i \in \mathcal{S}} \alpha_{x_i}$. Using the concept of Jacobi matrix, the above formula rewrites to

$$\hat{q}_i(x) = \hat{q}_i(x_s) + e_{x_i}^T A(x_s)(x - x_s) + \sum_{2 \leq |\alpha| \leq \deg(\hat{\mathcal{P}})} \frac{(D^\alpha \hat{q}_i)(x_s)}{\alpha!}(x - x_s)^\alpha \tag{A.1}$$

Since this shows that

$$r_i(s, y) = \sum_{2 \leq |\alpha| \leq \deg(\hat{\mathcal{P}})} \frac{(D^\alpha \hat{q}_i)(x_s)}{\alpha!} y^\alpha,$$

a straightforward estimation of the terms yields the claim. □

Proof of Lemma 5.

Proof. Using the concept of Hessian matrix, (A.1) from the proof of Lemma 4 can be rewritten into

$$\hat{q}_i(x) = \hat{q}_i(x_s) + e_{x_i}^T A(x_s)(x - x_s) + \tfrac{1}{2}(x - x_s)^T H^i(x_s)(x - x_s) + \sum_{|\alpha|=3} \frac{(D^\alpha \hat{q}_i)(x_s)}{\alpha!}(x - x_s)^\alpha,$$

thus readily implying the claim. □

Appendix B

Appendix Chapter 4

The appendix is taken from the published work [140].

B.1 Proofs

B.1.1 Proof of Theorem 11

Proof. Trivial. □

B.1.2 Proof of Theorem 12

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

B.2 Experiments

We next report the parameter values and the initial conditions.

B.2.1 SIR

Here we provide the parameters and runtimes for the SIR model considered in Section 4.2.1.

Parameters	$\beta_{1,1}$	$\beta_{1,2}$	$\beta_{2,1}$	$\beta_{2,2}$	γ_1	γ_2
Actual values	2.46	2.45	2.53	2.55	0.5	0.6

Table 14: Parameters 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

Table 15: Initial conditions of the SIR model.

B.2.2 Polymerization

Here we provide the parameters and runtimes for the polymerization model considered in Section 4.2.2.

Parameters	α_1	α_2	$\overline{\alpha}_1$	$\overline{\alpha}_2$	β_1	β_2	$\overline{\beta}_2$	$\overline{\beta}_2$	γ_2	γ_2
Actual values	0.55	0.60	1.95	2.00	1.5	1.6	0.01	0.01	0.25	0.25

Table 16: Parameters of the Polymerization model.

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

Table 17: Initial conditions of the Polymerization model.

B.2.3 Protein interaction network

Here we provide the parameters and runtimes for the model considered in Section 4.2.3.

Parameters	k_{b_1}	k_{b_2}	k_{u_1}	k_{u_2}
Actual values	20.10	19.90	0.1	0.1

Table 18: Parameters of the Protein interaction network.

Variables	A	B	A_{10}	A_{01}	A_{11}
Initial conditions	50	50	0	0	0

Table 19: Initial conditions of the Protein interaction network.

B.2.4 Electrical Network

Here we provide the parameters and runtimes for the model considered in Section 4.2.4.

Parameters	b_2	b_3	$a_{1,1}$	$a_{2,1}$	$a_{2,2}$
Actual values	0.56	0.66	1.12	0.40	0.50

Table 20: Parameters of the Electrical network.

Variables	$v_{1,1}$	$v_{2,1}$	$v_{2,2}$
Initial conditions	0.56	0.66	1.12

Table 21: Initial conditions of the Electrical network.

B.2.5 n-Hexane model

Here we provide the parameters and runtimes for the model considered in Section 4.2.5.

Parameters	k_1	k_2	k_3	k_4	k_5	k_6	k_7
Actual values	17	54	42	13	32	32	14

Table 22: Parameters of the n-Hexane model.

Variables	C_6H_{14}	C_1	C_5^2	C_2	C_4^2	C_3	C_3^2	C_4	C_2^2	H_2	C_6^2
Initial conditions	1	1	1	1	1	1	1	1	1	1	1

Table 23: Initial conditions of the n-Hexane model.

Appendix C

Appendix Chapter 5

C.1 Proofs

This appendix collects all the formal proofs of the results stated in the Chapter 5.

Proof of Theorem 13

Proof. Follows via the if-then direction of [46, Lemma 1], applied on each A^l. □

Proof of Lemma 6

Lemma 7. *For any $0 < s < t$, it holds that*

$$\prod_{i=s}^{t}(1+\frac{1}{2i-1}) = \sqrt{\frac{t}{s}} + \mathcal{O}(s^{-1})$$

Proof. As suggested in [20, Lemma 2], we approximate the product by applying the logarithm

$$\log\Big(\prod_{i=s}^{t}(1+\frac{1}{2i-1})\Big) = \sum_{i=s}^{t}\log\big(1+\frac{1}{2i-1}\big) = \sum_{i=s}^{t}\log\big(\frac{2i}{2i-1}\big)$$

and noting that the integral convergence test ensures

$$\Big|\sum_{i=s}^{t}\log\big(\frac{2i}{2i-1}\big)-\int_{s}^{t}\log\big(\frac{2x}{2x-1}\big)dx\Big|\leq\log\big(\frac{2s}{2s-1}\big)$$

Moreover, $\log(2x/(2x-1))=\log(2x)-\log(2x-1)$, while integration by substitution yields

$$\int_{s}^{t}\log(2x)dx=\tfrac{1}{2}\int_{2s}^{2t}\log(x)dx$$
$$\int_{s}^{t}\log(2x-1)dx=\tfrac{1}{2}\int_{2s-1}^{2t-1}\log(x)dx$$

Hence, by Taylor's theorem, there exist $\xi,\xi'\in(-1;0)$ with

$$\begin{aligned}\int_{s}^{t}2\log\big(\frac{2x}{2x-1}\big)dx&=\Big[\int_{2t-1}^{2t}\log(x)dx-\int_{2s-1}^{2s}\log(x)dx\Big]\\&=\log(2t)+\frac{1}{2t}\cdot\xi'-\log(2s)-\frac{1}{2s}\xi'\\&=\log(t/s)+\mathcal{O}(s^{-1})\end{aligned}$$

Recalling that $\frac{1}{2}\log(a)=\log(\sqrt{a})$, we thus obtain

$$\begin{aligned}\Big|\prod_{i=s}^{t}\big(1+\frac{1}{2i-1}\big)-\sqrt{\frac{t}{s}}\Big|&\leq\frac{2s}{2s-1}\cdot\exp(\mathcal{O}(s^{-1}))\\&\leq\big(1+\mathcal{O}(1/s)\big)(1+\mathcal{O}(1/s)),\end{aligned}$$

yielding the claim. □

We next prove Lemma 6.

Proof of Lemma 6. In the proof of [20, Lemma 2], the authors show that

$$\mathbb{P}(g_j=i)=\frac{1}{2j-1}\prod_{k=i}^{j-1}\big(1+\frac{1}{2k-1}\big),$$

With this, Lemma 7 implies

$$\mathbb{P}(g_j=i)=\frac{1}{2j-1}\Big(\frac{(j-1)^{1/2}}{i^{1/2}}+\mathcal{O}(i^{-1})\Big)=\Big(\frac{1}{2j}+\mathcal{O}(j^{-2})\Big)\Big(\frac{j^{1/2}}{i^{1/2}}+\mathcal{O}(i^{-1})\Big)$$

because $|\sqrt{j}-\sqrt{j-1}| \leq |\frac{1}{2}(j-1)^{-1/2}| \cdot 1 \leq i^{-1/2}$ by the mean value theorem. For the second and third statement, we note using the notation from [20, Lemma 2] that

$$\mathbb{E}(d_{k,i}I_{g_j=i} \mid G^{k-1}) = d_{k-1}I_{g_j=i} + \frac{d_{k-1}}{2k-1}I_{g_j=i} = (1+\frac{1}{2k-1})d_{k-1,i}I_{g_j=i},$$

which in turn implies

$$\mathbb{E}(d_{k,i}I_{g_j=i}) = \prod_{s=j}^{k}\big(1+\frac{1}{2s-1}\big)\mathbb{E}(d_{j,i}I_{g_j=i})$$

as postulated in the proof of [20, Lemma 2]. With this, the discussion from the aforementioned lemma ensures that

$$\begin{aligned}\mathbb{P}(g_j=i, g_k=i) &= \frac{1}{2k-1}\prod_{s=j}^{k-1}\Big(1+\frac{1}{2s-1}\Big)\frac{1}{2j-1}\frac{2j-1}{2i+1}\frac{4i+2}{2i-1}\\ &= \frac{1}{2k-1}\frac{1}{2i-1}\prod_{s=j}^{k-1}\Big(1+\frac{1}{2s-1}\Big)\cdot 2,\end{aligned}$$

where, thanks to [44] and $\Gamma(x+1)=x\Gamma(x)$ for any $x>0$, we have that

$$\begin{aligned}\frac{2j-1}{2i+1}\frac{4i+2}{2i-1} = \mu^{(2)}_{j-1,i} &= \prod_{s=i+1}^{j-1}\Big(1+\frac{1}{2s-1}\Big)\mu^{(2)}_{i,i}\\ &= \frac{\Gamma(i+\frac{1}{2})}{\Gamma(j-\frac{1}{2})}\frac{\Gamma(j+\frac{1}{2})}{\Gamma(i+\frac{3}{2})}\mu^{(2)}_{i,i}\\ &= \frac{j-\frac{1}{2}}{i+\frac{1}{2}}\mu^{(2)}_{i,i}\\ &= \frac{2j-1}{2i+1}\mu^{(2)}_{i,i}\end{aligned}$$

and

$$\mu^{(2)}_{i,i} = \frac{2i-2}{2i-1}\cdot(1^2+1)+\frac{1}{2i-1}\cdot(2^2+2) = \frac{4i+2}{2i-1}$$

This, in turn, allows us to conclude that

$$\begin{aligned}
&\mathbb{P}(g_j = i, g_k = i) - \mathbb{P}(g_j = i)\mathbb{P}(g_k = i) = \\
&= \frac{1}{2k-1}\frac{1}{2i-1}\prod_{s=j}^{k-1}\Big(1+\frac{1}{2s-1}\Big)\cdot 2- \\
&\frac{1}{2k-1}\prod_{s=i}^{k-1}\big(1+\frac{1}{2s-1}\big)\cdot\frac{1}{2j-1}\prod_{s=i}^{j-1}\big(1+\frac{1}{2s-1}\big) \\
&= \frac{1}{2k-1}\prod_{s=j}^{k-1}\big(1+\frac{1}{2s-1}\big)\Big[\frac{2}{2i-1}-\frac{1}{2j-1}\prod_{s=i}^{j-1}\big(1+\frac{1}{2s-1}\big)^2\Big] \\
&= \frac{1}{4i\sqrt{jk}}+\mathcal{O}(\frac{1}{i\sqrt{ijk}}),
\end{aligned}$$

where the last identity follows by invoking several times Lemma 7. This establishes the third statement which, together with the first statement, implies the second statement. □

Proof of Theorem 14

Proof. Pick $H \in \mathcal{H}$ with $H = \{a, a+1, \ldots, b-1, b\}$, an $i \in \{\gamma\sqrt{n}, \gamma n\}$ such that $i \leq a$ and let $X^{i,j}$ be as in (5.1). Then

$$\begin{aligned}
\mathbb{E}[\sum_{j\in H} X^{i,j}] &= \sum_{j\in H}\mathbb{E}[X^{i,j}] \\
&= \sum_{j\in H}\frac{1}{2\sqrt{ij}}+\sum_{j\in H}\mathcal{O}(1/ij) \\
&= \sum_{j\in H}\frac{1}{2\sqrt{ij}}+\mathcal{O}((\log(b)+1)/i) \\
&= \int_a^b \frac{1}{2\sqrt{ij}}dj+\mathcal{O}(\log(b)/i+1/i+1/\sqrt{ia}) \\
&= \sqrt{b/i}-\sqrt{a/i}+\mathcal{O}(\log(b)/i+1/i+1/\sqrt{ia}),
\end{aligned}$$

where the first identity is due to the linearity of the expected value, the second due to the first identity of Lemma 6 and the logarithmic growth of harmonic numbers, the third due to the integral convergence test, and

the forth due follows via integration. This implies

$$\mathbb{E}[\sum_{j\in H} X^{i,j}] = \begin{cases} \mathcal{O}(1), & i \in C_\nu, H \in \mathcal{C} \\ \Theta(\sqrt{\xi}\sqrt[4]{n}), & i \in C_\nu, H \in \mathcal{F} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{C} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{F} \end{cases}$$

The statement follows because $|\deg(i,H) - \sum_{j\in H} X^{i,j}| \leq 1$. □

Proof of Theorem 15

Proof. From the proof of Theorem 14, we infer for $i < i'$:

$$\begin{aligned}\mathbb{E}[\sum_{j\in H} X^{i,j}] - \mathbb{E}[\sum_{j\in H} X^{i',j}] \\ = (\sqrt{b} - \sqrt{a})\Big(\frac{1}{\sqrt{i}} - \frac{1}{\sqrt{i'}}\Big) + \mathcal{O}(\log(b)/i + 1/i + 1/\sqrt{ia})\end{aligned}$$

We consider the case $i, i' \in C_\nu$ and $H \in \mathcal{F}$ as the other cases are straightforward. By Taylor's theorem, we infer for $0 < x_0 < x_1$:

$$|x_0^{-1/2} - x_1^{-1/2}| \leq |\tfrac{1}{2}x_0^{-3/2}| \cdot |x_1 - x_0|$$

Setting $x_1 = \gamma_1\sqrt[2]{n}$ and $x_0 = \gamma_0\sqrt[2]{n}$, the above formula yields

$$|\frac{1}{\sqrt{x_0}} - \frac{1}{\sqrt{x_1}}| \leq |\gamma_1 - \gamma_0| / \sqrt[4]{n}\gamma_0^{3/2}$$

Noting that $\sqrt{b} - \sqrt{a} = \mathcal{O}(\sqrt{\xi n})$, we obtain

$$|(\sqrt{b} - \sqrt{a})\Big(\frac{1}{\sqrt{i}} - \frac{1}{\sqrt{i'}}\Big)| \leq \sqrt{\xi}\sqrt[4]{n}\frac{|\gamma_1 - \gamma_0|}{\gamma_0^{3/2}}$$

The statement follows because $|\deg(i,H) - \sum_{j\in H} X^{i,j}| \leq 1$. □

Proof of Theorem 16

Proof of Theorem 16. Letting j, j' ranging over H, we first note that

$$\begin{aligned}
&\mathbb{V}[\sum_j X^{i,j} - \sum_j X^{i',j}] \\
&= \mathbb{V}[\sum_j X^{i,j}] + \mathbb{V}[\sum_j X^{i',j}] - 2\,Cov[\sum_j X^{i,j}, \sum_j X^{i',j}] \\
&= \mathbb{V}[\sum_j X^{i,j}] + \mathbb{V}[\sum_j X^{i',j}] - 2\sum_j \sum_{j'} Cov[X^{i,j}, X^{i',j'}]
\end{aligned}$$

The first statement follows by noting that [20, Lemma 3] ensures for any $i \neq i'$ and $j, j' \in H$ that

$$Cov[X^{i,j}, X^{i',j'}] = \mathbb{E}[X^{i,j}X^{i',j'}] - \mathbb{E}[X^{i,j}]\mathbb{E}[X^{i',j'}] \leq 0.$$

To see the second statement, we note that Lemma 6 implies for $j, j' \in H$ with $j \neq j'$:

$$\begin{aligned}
Cov(X^{i,j}, X^{i,j'}) &= \mathbb{E}[X^{i,j}X^{i,j'}] - \mathbb{E}[X^{i,j}]\mathbb{E}[X^{i,j'}] \\
&= \mathbb{P}(g_j = i, g_k = i) - \mathbb{P}(g_j = i)\mathbb{P}(g_k = i) \\
&= \frac{1}{4i\sqrt{jk}} + \mathcal{O}(\frac{1}{i\sqrt{ijk}})
\end{aligned}$$

With this, we obtain

$$\begin{aligned}
\mathcal{E} &:= \sum_{j \in H} \sum_{j' \in H \setminus \{j\}} \big(\mathbb{E}[X^{i,j}X^{i,j'}] - \mathbb{E}[X^{i,j}]\mathbb{E}[X^{i,j'}]\big) \\
&= \sum_{j \in H} \sum_{j' \in H \setminus \{j\}} \Big(\frac{1}{4i\sqrt{jj'}} + \mathcal{O}(\frac{1}{i\sqrt{ijj'}})\Big) \\
&= \begin{cases} \mathcal{O}(1), & i \in C_\nu,\ H \in \mathcal{C} \\ \Theta(\sqrt{n}), & i \in C_\nu,\ H \in \mathcal{F} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{C} \\ \mathcal{O}(1), & i \in F_\nu,\ H \in \mathcal{F} \end{cases}
\end{aligned}$$

Moreover, one can observe

$$\begin{aligned}\mathbb{V}\big(n^{-1/4}\sum_{j\in H} X^{i,j}\big) &= n^{-1/2}\big(\sum_{j\in H}\mathbb{V}[X^{i,j}] + \mathcal{E}\big)\\ &= n^{-1/2}\sum_{j\in H}\mathbb{V}[X^{i,j}] + n^{-1/2}\mathcal{E}\\ &\leq n^{-1/2}\sum_{j\in H}\big[\frac{1}{2\sqrt{ij}} - \frac{1}{4ij}\big] + n^{-1/2}\mathcal{E}\\ &= \mathcal{O}(n^{-1/2}\mathbb{E}[\sum_{j\in H} X^{i,j}]) + n^{-1/2}\mathcal{E}\end{aligned}$$

The statement follows because $|\deg(i,H) - \sum_{j\in H} X^{i,j}| \leq 1$. □

Bibliography

[1] Stefano Allesina and Si Tang. "Stability criteria for complex ecosystems". In: *Nature* 483.7388 (2012), pp. 205–208.

[2] Matthias Althoff. "An Introduction to CORA 2015". In: *Workshop on Applied Verification for Continuous and Hybrid Systems*. 2015.

[3] Matthias Althoff. "Reachability Analysis of Nonlinear Systems using Conservative Polynomialization and Non-Convex Sets". In: *International conference on Hybrid systems: computation and control*. 2013, pp. 173–182.

[4] Matthias Althoff, Colas Le Guernic, and Bruce H. Krogh. "Reachable set computation for uncertain time-varying linear systems". In: *International conference on Hybrid systems: computation and control*. Ed. by Marco Caccamo, Emilio Frazzoli, and Radu Grosu. ACM, 2011, pp. 93–102.

[5] *American Revolution network dataset – KONECT*. Oct. 2017. URL: `http://konect.cc/networks/brunson_revolution`.

[6] BDO Anderson and PC Parks. "Lumped approximation of distributed systems and controllability questions". In: *IEE Proceedings D (Control Theory and Applications)*. Vol. 132. 3. IET. 1985, pp. 89–94.

[7] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

[8] David Angeli. "A tutorial on Chemical Reaction Networks dynamics". In: *2009 European Control Conference (ECC)*. IEEE. 2009, pp. 649–657.

[9] Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[10] Masanao Aoki. “Control of large-scale dynamic systems by aggregation”. In: *IEEE Trans. Autom. Control* 13.3 (1968), pp. 246–253. ISSN: 0018-9286.

[11] Georgios Argyris, Alberto Lluch Lafuente, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “An Extension of ERODE to Reduce Boolean Networks By Backward Boolean Equivalence”. In: *Computational Methods in Systems Biology*. Ed. by Ion Petre and Andrei Păun. Cham: Springer International Publishing, 2022, pp. 294–301.

[12] Eugene Asarin, Thao Dang, and Antoine Girard. “Reachability Analysis of Nonlinear Systems Using Conservative Approximation”. In: *International conference on Hybrid Systems: Computation and Control*. 2003.

[13] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Efficient Local Computation of Differential Bisimulations via Coupling and Up-to Methods”. In: *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 2021, pp. 1–14.

[14] Christel Baier, Bettina Engelen, and Mila Majster-Cederbaum. “Deciding Bisimilarity and Similarity for Probabilistic Processes”. In: *Journal of Computer and System Sciences* 60.1 (2000), pp. 187–231.

[15] Albert-László Barabási. “The science of networks”. In: *Cambridge MA: Perseus* (2012).

[16] Albert-László Barabási and Eric Bonabeau. “Scale-free networks”. In: *Scientific american* 288.5 (2003), pp. 60–69.

[17] Stephan Barthel and Christian Isendahl. “Urban gardens, agriculture, and water management: Sources of resilience for long-term food security in cities”. In: *Ecological economics* 86 (2013), pp. 224–234.

[18] Vladimir Batagelj and Andrej Mrvar. *Pajek datasets*. 2006.

[19] Andreea Beica, Jérôme Feret, and Tatjana Petrov. “Tropical Abstraction of Biochemical Reaction Networks with Guarantees”. In: *Electronic Notes in Theoretical Computer Science* 350 (2020). Proceedings of SASB 2018, the Ninth International Workshop on Static Analysis and Systems Biology, Freiburg, Germany - August 28th, 2018, pp. 3–32. ISSN: 1571-0661. DOI: https://doi.org/10.1016/

j.entcs.2020.06.002. URL: https://www.sciencedirect.com/science/article/pii/S1571066120300293.

[20] Béla Bollobás and Oliver Riordan. "The diameter of a scale-free random graph". In: *Combinatorica* 24.1 (2004), pp. 5–34.

[21] Michele Boreale. "Algebra, Coalgebra, and Minimization in Polynomial Differential Equations". In: *International Conference on Foundations of Software Science and Computation Structures*. 2017, pp. 71–87.

[22] Stephen P Borgatti and Martin G Everett. "Notions of position in social network analysis". In: *Sociological methodology* (1992), pp. 1–35.

[23] Stephen P Borgatti and Martin G Everett. "The class of all regular equivalences: Algebraic structure and computation". In: *Social networks* 11.1 (1989), pp. 65–88.

[24] Stephen P Borgatti and Martin G Everett. "Two algorithms for computing regular equivalence". In: *Social networks* 15.4 (1993), pp. 361–376.

[25] Stephen P Borgatti, Martin G Everett, and Linton C Freeman. "Ucinet for Windows: Software for social network analysis". In: *Harvard, MA: analytic technologies* 6 (2002), pp. 12–15.

[26] Luca Bortolussi and Nicolas Gast. "Mean Field Approximation of Uncertain Stochastic Models". In: *International Conference on Dependable Systems and Networks*. 2016, pp. 287–298.

[27] Luca Bortolussi and Luca Palmieri. "Deep Abstractions of Chemical Reaction Networks". In: *Computational Methods in Systems Biology*. Ed. by Milan Češka and David Šafránek. Cham: Springer International Publishing, 2018, pp. 21–38.

[28] Paul PJ van den Bosch and Alexander C van der Klauw. *Modeling, identification and simulation of dynamical systems*. crc Press, 2020.

[29] George Edward Briggs and John Burdon Sanderson Haldane. "A note on the kinetics of enzyme action". In: *Biochemical journal* 19.2 (1925), p. 338.

[30] Peter Brucker. "On the complexity of clustering problems". In: *Optimization and Operations Research: Proceedings of a Workshop Held at the University of Bonn, October 2–8, 1977*. Springer. 1978, pp. 45–54.

[31] Peter Buchholz. “Exact and Ordinary Lumpability in Finite Markov Chains”. In: *Journal of Applied Probability* 31.1 (1994), pp. 59–75. ISSN: 00219002.

[32] Ferdinanda Camporesi and Jérôme Feret. “Formal Reduction for Rule-based Models”. In: *Electronic Notes in Theoretical Computer Science* 276 (2011), pp. 29–59.

[33] Luca Cardelli, Isabel Cristina Pérez-Verona, Mirco Tribastone, Max Tschaikowski, Andrea Vandin, and Tabea Waizmann. “Exact maximal reduction of stochastic reaction networks by species lumping”. In: *Bioinformatics* 37.15 (2021), pp. 2175–2182.

[34] Luca Cardelli, Giuseppe Squillace, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Formal lumping of polynomial differential equations through approximate equivalences”. In: *Journal of Logical and Algebraic Methods in Programming* 134 (2023), p. 100876. ISSN: 2352-2208. DOI: `https://doi.org/10.1016/j.jlamp.2023.100876`. URL: `https://www.sciencedirect.com/science/article/pii/S2352220823000305`.

[35] Luca Cardelli, Mirco Tribastone, and Max Tschaikowski. “From electric circuits to chemical networks”. In: *Natural Computing* 19.1 (2020), pp. 237–248.

[36] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Efficient Syntax-driven Lumping of Differential Equations”. In: *Tools and Algorithms for the Construction and Analysis of Systems — 21st International Conference, TACAS*. 2016.

[37] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “ERODE: A Tool for the Evaluation and Reduction of Ordinary Differential Equations”. In: *TACAS*. 2017.

[38] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Forward and Backward Bisimulations for Chemical Reaction Networks”. In: *International Conference on Concurrency Theory*. 2015.

[39] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Guaranteed Error Bounds on Approximate Model Abstractions Through Reachability Analysis”. In: *Quantitative Evaluation of Systems*. Ed. by Annabelle McIver and Andras Horvath. Cham: Springer International Publishing, 2018, pp. 104–121. ISBN: 978-3-319-99154-2.

[40] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Maximal aggregation of polynomial dynamical systems”. In: *Proceedings of the National Academy of Sciences* 114.38 (2017), pp. 10029–10034.

[41] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Symbolic computation of differential equivalences”. In: *POPL*. 2016, pp. 137–150.

[42] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Symbolic computation of differential equivalences”. In: *Theoretical Computer Science* 777 (2019). In memory of Maurice Nivat, a founding father of Theoretical Computer Science - Part I, pp. 132–154. ISSN: 0304-3975.

[43] Luca Cardelli, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Syntactic Markovian Bisimulation for Chemical Reaction Networks”. In: *Models, Algorithms, Logics and Tools - Essays Dedicated to Kim Guldstrand Larsen on the Occasion of His 60th Birthday*. Ed. by Luca Aceto, Giorgio Bacci, Giovanni Bacci, Anna Ingólfsdóttir, Axel Legay, and Radu Mardare. Vol. 10460. 2017, pp. 466–483.

[44] Marc Chamberland and Armin Straub. “On gamma quotients and infinite products”. In: *Advances in Applied Mathematics* 51.5 (2013), pp. 546–562. ISSN: 0196-8858.

[45] Vijaysekhar Chellaboina, Sanjay P. Bhat, Wassim M. Haddad, and Dennis S. Bernstein. “Modeling and analysis of mass-action kinetics”. In: *IEEE Control Systems Magazine* 29.4 (2009), pp. 60–78. DOI: 10.1109/MCS.2009.932926.

[46] Di Chen, Franck van Breugel, and James Worrell. “On the complexity of computing probabilistic bisimilarity”. In: *Foundations of Software Science and Computational Structures: 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24–April 1, 2012. Proceedings 15*. Springer. 2012, pp. 437–451.

[47] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. “Flow*: An Analyzer for Non-linear Hybrid Systems”. In: *International Conference on Computer Aided Verification*. 2013, pp. 258–263.

[48] Xin Chen and Sriram Sankaranarayanan. “Reachability Analysis for Cyber-Physical Systems: Are We There Yet?” In: *NASA Formal Methods Symposium*. Springer. 2022, pp. 109–130.

[49] Vittoria Colizza, Romualdo Pastor-Satorras, and Alessandro Vespignani. “Reaction–diffusion processes and metapopulation models in heterogeneous networks”. In: *Nature Physics* 3.4 (2007), pp. 276–282.

[50] Holger Conzelmann, Dirk Fey, and Ernst Gilles. “Exact model reduction of combinatorial reaction networks”. In: *BMC Systems Biology* 2.1 (2008), p. 78.

[51] Holger Conzelmann, Julio Saez-Rodriguez, Thomas Sauter, Boris Kholodenko, and Ernst Gilles. “A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks”. In: *BMC Bioinformatics* 7.1 (2006), p. 34.

[52] *CORA Manual 2024*. 2024. URL: `https://tumcps.github.io/CORA/manual/index.html`.

[53] Anup Das, Geoff V. Merrett, Mirco Tribastone, and Bashir M. Al-Hashimi. “Workload Change Point Detection for Runtime Thermal Management of Embedded Systems”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.8 (2016), pp. 1358–1371.

[54] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. “Structural reducibility of multilayer networks”. In: *Nature communications* 6.1 (2015), pp. 1–9.

[55] Aristides Dokoumetzidis and Leon Aarons. “Proper lumping in systems biology models”. In: *IET systems biology* 3.1 (2009), pp. 40–51.

[56] Alexandre Donzé and Oded Maler. “Systematic Simulation Using Sensitivity Analysis”. In: *International conference on Hybrid Systems: Computation and Control*. Springer, 2007, pp. 174–189.

[57] Patrick Doreian. “Equivalence in a social network”. In: *The Journal of Mathematical Sociology* 13.3 (1988), pp. 243–281.

[58] Patrick Doreian, Vladimir Batagelj, and Anuska Ferligoj. *Generalized blockmodeling*. 25. Cambridge university press, 2005.

[59] C Ronnie Drever, Garry Peterson, Christian Messier, Yves Bergeron, and Mike Flannigan. “Can forest management based on natural disturbances maintain ecological resilience?” In: *Canadian Journal of Forest Research* 36.9 (2006), pp. 2285–2299.

[60] Parasara Sridhar Duggirala, Sayan Mitra, and Mahesh Viswanathan. “Verification of Annotated Models from Executions”. In: *International Conference on Embedded Software*. IEEE Press, 2013.

[61] Weinan E, Tiejun Li, and Eric Vanden-Eijnden. “Optimal partition and effective dynamics of complex networks”. In: *Proceedings of the National Academy of Sciences* 105.23 (June 2008), pp. 7907–12.

[62] Chuchu Fan, Bolun Qi, Sayan Mitra, Mahesh Viswanathan, and Parasara Sridhar Duggirala. “Automatic Reachability Analysis for Nonlinear Hybrid Models with C2E2”. In: *International Conference on Computer Aided Verification*. 2016, pp. 531–538.

[63] Jay A. Farrell and Marios M. Polycarpou. *Adaptive Approximation Based Control*. Wiley-Interscience, 2006. ISBN: 0471727881.

[64] Jérôme Feret, Vincent Danos, Jean Krivine, Russ Harmer, and Walter Fontana. “Internal coarse-graining of molecular systems”. In: *Proceedings of the National Academy of Sciences* 106.16 (2009), pp. 6453–6458.

[65] Sue C Freeman and Linton C Freeman. *The networkers network: A study of the impact of a new communications medium on sociometric structure*. School of Social Sciences University of Calif., 1979.

[66] Thorben Funke and Till Becker. “Stochastic block models: A comparison of variants and inference methods”. In: *PloS one* 14.4 (2019), e0215296.

[67] Han Gao, Jian-Xun Wang, and Matthew J. Zahr. “Non-intrusive model reduction of large-scale, nonlinear dynamical systems using deep learning”. In: *Physica D: Nonlinear Phenomena* 412 (2020), p. 132614. ISSN: 0167-2789. DOI: https://doi.org/10.1016/j.physd.2020.132614. URL: https://www.sciencedirect.com/science/article/pii/S0167278919305573.

[68] Guillaume Gaulier and Soledad Zignago. *BACI: International Trade Database at the Product-Level. The 1994-2007 Version*. Working Papers 2010-23. CEPII, 2010. URL: http://www.cepii.fr/CEPII/fr/publications/wp/abstract.asp?NoDoc=2726.

[69] Khalil Ghorbal, Eric Goubault, and Sylvie Putot. “The Zonotope Abstract Domain Taylor1+”. In: *International Conference on Computer Aided Verification*. Ed. by Ahmed Bouajjani and Oded Maler. 2009, pp. 627–633.

[70] Alessandro Giacalone, Chi-Chang Jou, and Scott A. Smolka. “Algebraic Reasoning for Probabilistic Concurrent Games”. In: *IFIP WG 2.2/2.3*. 1990, pp. 443–458.

[71] Daniel Thomas Gillespie. “Exact Stochastic Simulation of Coupled Chemical Reactions”. In: *Journal of Physical Chemistry* 81.25 (1977), pp. 2340–2361.

[72] Antoine Girard and George J. Pappas. “Approximate Bisimulations for Nonlinear Dynamical Systems”. In: *IEEE Conference on Decision and Control and European Control Conference*. 2005.

[73] Antoine Girard and George J. Pappas. “Approximation Metrics for Discrete and Continuous Systems”. In: *IEEE Transactions on Automatic Control* 52.5 (2007), pp. 782–798.

[74] Ankit Gupta, Christoph Schwab, and Mustafa Khammash. “DeepCME: A deep learning framework for solving the Chemical Master Equation”. In: (2021). DOI: 10.1101/2021.06.05.447033.

[75] David Hartman and Lalit K Mestha. “A deep learning framework for model reduction of dynamical systems”. In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE. 2017, pp. 1917–1922.

[76] Jane Hillston, Carla Piazza, Andrea Marin, and Sabina Rossi. “Contextual lumpability”. In: *ValueTools 2013–7th International Conference on Performance Evaluation Methodologies and Tools*. ICST. 2013.

[77] Sui Huang, Gabriel Eichler, Yaneer Bar-Yam, and Donald E. Ingber. “Cell Fates as High-Dimensional Attractor States of a Complex Gene Regulatory Network”. In: *Physical Review Letters* 94 (12 Apr. 2005), p. 128701.

[78] Giulio Iacobelli and Mirco Tribastone. “Lumpability of fluid models with heterogeneous agent types”. In: *International Conference on Dependable Systems and Networks*. 2013.

[79] Md. Ariful Islam, Abhishek Murthy, Ezio Bartocci, Elizabeth Cherry, Flavio H. Fenton, James Glimm, Scott A. Smolka, and Radu Grosu. "Model-order reduction of ion channel dynamics using approximate bisimulation". In: *Theoretical Computer Science* 599 (2015), pp. 34–46.

[80] Yoh Iwasa, Viggo Andreasen, and Simon Levin. "Aggregation in model ecosystems. I. Perfect aggregation". In: *Ecological Modelling* 37.3-4 (1987), pp. 287–302.

[81] Yoh Iwasa, Simon A. Levin, and Viggo Andreasen. "Aggregation in Model Ecosystems II. Approximate Aggregation". In: *Mathematical Medicine and Biology* 6.1 (1989), pp. 1–23.

[82] Pengfei Jiao, Xuan Guo, Ting Pan, Wang Zhang, Yulong Pei, and Lin Pan. "A survey on role-oriented network embedding". In: *IEEE Transactions on Big Data* 8.4 (2021), pp. 933–952.

[83] JP Keener and James Sneyd. *Mathematical physiology 1: Cellular physiology*. 2009.

[84] William Ogilvy Kermack and Anderson G McKendrick. "A contribution to the mathematical theory of epidemics". In: *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character* 115.772 (1927), pp. 700–721.

[85] Markus Koschorreck, Holger Conzelmann, Sybille Ebert, Michael Ederer, and Ernst Dieter Gilles. "Reduced modeling of signal transduction–a modular approach". In: *BMC bioinformatics* 8.1 (2007), pp. 1–24.

[86] M.K. Kozlov, S.P. Tarasov, and L.G. Khachiyan. "The polynomial solvability of convex quadratic programming". In: *USSR Computational Mathematics and Mathematical Physics* 20.5 (1980), pp. 223–228.

[87] Niclas Kruff, Christoph Lüders, Ovidiu Radulescu, Thomas Sturm, and Sebastian Walcher. "Algorithmic Reduction of Biological Networks with Multiple Time Scales". In: (2021).

[88] Jérôme Kunegis. "Konect: the koblenz network collection". In: *Proceedings of the 22nd international conference on world wide web*. 2013, pp. 1343–1350.

[89] Juan Kuntz, Diego Oyarzún, and Guy-Bart Stan. “Model reduction of genetic-metabolic networks via time scale separation”. In: *A systems theoretic approach to systems and synthetic biology I: models and system characterizations* (2014), pp. 181–210.

[90] J. C. W. Kuo and James Wei. “Lumping Analysis in Monomolecular Reaction Systems. Analysis of Approximately Lumpable System”. In: *Industrial & Engineering Chemistry Fundamentals* 8.1 (1969), pp. 124–133.

[91] Ratan Lal and Pavithra Prabhakar. “Bounded error flowpipe computation of parameterized linear systems”. In: *International Conference on Embedded Software*. 2015, pp. 237–246.

[92] Vu Tuan Hieu Le, Cristina Stoica, Teodoro Alamo, Eduardo F Camacho, and Didier Dumur. *Zonotopes: From guaranteed state-estimation to control*. John Wiley & Sons, 2013.

[93] Kookjin Lee and Kevin T Carlberg. “Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders”. In: *Journal of Computational Physics* 404 (2020), p. 108973.

[94] Alexander Leguizamon-Robayo, Antonio Jiménez-Pastor, Micro Tribastone, Max Tschaikowski, and Andrea Vandin. “Approximate Constrained Lumping of Polynomial Differential Equations”. In: *International Conference on Computational Methods in Systems Biology*. Springer. 2023, pp. 106–123.

[95] Genyuan Li and Herschel Rabitz. “A general analysis of approximate lumping in chemical kinetics”. In: *Chemical Engineering Science* 45.4 (1990), pp. 977–1002. ISSN: 0009-2509.

[96] Jiang Liu, Naijun Zhan, Hengjun Zhao, and Liang Zou. “Abstraction of Elementary Hybrid Systems by Variable Transformation”. In: *International Symposium on Formal Methods*. Ed. by Nikolaj S. Bjørner and Frank S. de Boer. Vol. 9109. 2015, pp. 360–377.

[97] Francois Lorrain and Harrison C White. “Structural equivalence of individuals in social networks”. In: *The Journal of mathematical sociology* 1.1 (1971), pp. 49–80.

[98] Joseph J Luczkovich, Stephen P Borgatti, Jeffrey C Johnson, and Martin G Everett. “Defining and measuring trophic role similarity in food webs using regular equivalence”. In: *Journal of Theoretical Biology* 220.3 (2003), pp. 303–321.

[99] Rupak Majumdar and Majid Zamani. “Approximately Bisimilar Symbolic Models for Digital Control Systems”. In: *International Conference on Computer Aided Verification*. 2012, pp. 362–377.

[100] Andrea Marin, Carla Piazza, and Sabina Rossi. “Proportional Lumpability”. In: *Formal Modeling and Analysis of Timed Systems*. Ed. by Étienne André and Mariëlle Stoelinga. Cham: Springer International Publishing, 2019, pp. 265–281.

[101] Maarten Marx and Michael Masuch. “Regular equivalence and dynamic logic”. In: *Social Networks* 25.1 (2003), pp. 51–65.

[102] Miha Matjaŝiĉ, Marjan Cugmas, and Aleŝ Žiberna. “blockmodeling: An R package for generalized blockmodeling”. In: *Advances in Methodology and Statistics* 17.2 (2020), pp. 49–66.

[103] Peter McMullen. “On zonotopes”. In: *Transactions of the American Mathematical Society* 159 (1971), pp. 91–109.

[104] Leonor Michaelis, Maud L Menten, et al. “Die kinetik der invertinwirkung”. In: *Biochem. z* 49.333-369 (1913), p. 352.

[105] Dimitrios Milios and Stephen Gilmore. “Component aggregation for PEPA models: An approach based on approximate strong equivalence”. In: *Performance Evaluation* 94 (2015), pp. 43–71.

[106] Dimitrios Milios and Stephen Gilmore. “Compositional approximate Markov chain aggregation for PEPA models”. In: *Computer Performance Engineering: 9th European Workshop, EPEW 2012, Munich, Germany, July 30, 2012, and 28th UK Workshop, UKPEW 2012, Edinburgh, UK, July 2, 2012, Revised Selected Papers 9*. Springer. 2013, pp. 96–110.

[107] Michael I. Monine, Richard G. Posner, Paul B. Savage, James R. Faeder, and William S. Hlavacek. “Modeling Multivalent Ligand-Receptor Interactions with Steric Constraints on Configurations of Cell-Surface Receptor Aggregates”. In: *Biophysical Journal* 98.1 (2010), pp. 48–56.

[108] Bruce Moore. “Principal component analysis in linear systems: Controllability, observability, and model reduction”. In: *IEEE transactions on automatic control* 26.1 (1981), pp. 17–32.

[109] Max Müller. “Über das Fundamentaltheorem in der Theorie der gewöhnlichen Differentialgleichungen”. In: *Mathematische Zeitschrift* 26 (1927), pp. 619–645.

[110] James D. Murray. *Mathematical Biology I: An Introduction*. 3rd. Springer, 2002.

[111] T.F. Narbeshuber, H. Vinek, and J.A. Lercher. "Monomolecular Conversion of Light Alkanes over H-ZSM-5". In: *Journal of Catalysis* 157.2 (1995), pp. 388–395. ISSN: 0021-9517.

[112] M. S. Okino and M. L. Mavrovouniotis. "Simplification of Mathematical Models of Chemical Reaction Systems". In: *Chemical Reviews* 2.98 (1998), pp. 391–408.

[113] Tore Opsahl and Pietro Panzarasa. "Clustering in weighted networks". In: *Social networks* 31.2 (2009), pp. 155–163.

[114] Luigi Orsenigo, Fabio Pammolli, Massimo Riccaboni, Andrea Bonaccorsi, and Giuseppe Turchetti. "The evolution of knowledge and the dynamics of an industry network". In: *Journal of Management & Governance* 1 (1997), pp. 147–175.

[115] Robert Paige and Robert E. Tarjan. "Three Partition Refinement Algorithms". In: *SIAM Journal on Computing* 16.6 (1987), pp. 973–989.

[116] Panos M. Pardalos and Stephen A. Vavasis. "Quadratic programming with one negative eigenvalue is NP-hard". In: *Journal of Global Optimization* 1.1 (1991), pp. 15–22.

[117] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. "Epidemic processes in complex networks". In: *Reviews of modern physics* 87.3 (2015), p. 925.

[118] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[119] Tiago P Peixoto. "Bayesian stochastic blockmodeling". In: *Advances in network clustering and blockmodeling* (2019), pp. 289–332.

[120] Aaron Pereira and Matthias Althoff. "Safety control of robots under Computed Torque control using reachable sets". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 331–338. DOI: 10.1109/ICRA.2015.7139020.

[121] Isabel Cristina Perez-Verona, Mirco Tribastone, and Andrea Vandin. "A large-scale assessment of exact lumping of quantitative models in the BioModels repository". In: *Theoretical Computer Science* 893 (2021), pp. 41–59. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2021.06.026. URL: https://www.sciencedirect.com/science/article/pii/S0304397521003716.

[122] Lawrence Perko. *Differential Equations and Dynamical Systems*. New York: Springer, 1991.

[123] Tatjana Petrov and Stefano Tognazzi. "Centrality-Preserving Exact Reductions of Multi-Layer Networks". In: *Leveraging Applications of Formal Methods, Verification and Validation: Engineering Principles*. Ed. by Tiziana Margaria and Bernhard Steffen. Cham: Springer International Publishing, 2020, pp. 397–415.

[124] Tatjana Petrov and Stefano Tognazzi. "Exact and approximate role assignment for multi-layer networks". In: *Journal of Complex Networks* 9.5 (Oct. 2021), cnab027.

[125] V Petrov, Elena Nikolova, and Olaf Wolkenhauer. "Reduction of nonlinear dynamic systems with an application to signal transduction pathways". In: *IET systems biology* 1.1 (2007), pp. 2–9.

[126] Thomas P. Prescott and Antonis Papachristodoulou. "Guaranteed error bounds for structured complexity reduction of biochemical networks". In: *Journal of Theoretical Biology* 304.0 (2012), pp. 172–182. ISSN: 0022-5193.

[127] Ovidiu Radulescu, Alexander N Gorban, Andrei Zinovyev, and Vincent Noel. "Reduction of dynamical biochemical reactions networks in computational biology". In: *Frontiers in genetics* 3 (2012), p. 131.

[128] Nacim Ramdani, Nacim Meslem, and Yves Candau. "Computing reachable sets for uncertain nonlinear monotone systems". In: *Nonlinear Analysis: Hybrid Systems* 4.2 (2010). IFAC World Congress 2008, pp. 263–278.

[129] Nacim Ramdani, Nacim Meslem, and Yves Candau. "Reachability of Uncertain Nonlinear Systems Using a Nonlinear Hybridization". In: *HSCC*. 2008, pp. 415–428.

[130] Jörg Reichardt and Douglas R White. "Role models for complex networks". In: *The European Physical Journal B* 60 (2007), pp. 217–224.

[131] Denis Repin and Tatjana Petrov. “Automated deep abstractions for stochastic chemical reaction networks”. In: *Information and Computation* 281 (2021), p. 104788. ISSN: 0890-5401. DOI: https://doi.org/10.1016/j.ic.2021.104788. URL: https://www.sciencedirect.com/science/article/pii/S0890540121001048.

[132] Dalila Ressi, Riccardo Romanello, Carla Piazza, and Sabina Rossi. “Neural Networks Reduction via Lumping”. In: *AIxIA 2022 – Advances in Artificial Intelligence*. Ed. by Agostino Dovier, Angelo Montanari, and Andrea Orlandini. Cham: Springer International Publishing, 2023, pp. 75–90.

[133] Jonathan Rosenfeld and Eby G. Friedman. “Design Methodology for Global Resonant H-Tree Clock Distribution Networks”. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 15.2 (2007), pp. 135–148.

[134] Ryan A. Rossi and Nesreen K. Ahmed. “Role Discovery in Networks”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.4 (2015), pp. 1112–1131. DOI: 10.1109/TKDE.2014.2349913.

[135] Klaus R Schneider and Thomas Wilhelm. “Model reduction by extended quasi-steady-state approximation”. In: *Journal of mathematical biology* 40.5 (2000), pp. 443–450.

[136] Joseph K. Scott and Paul I. Barton. “Bounds on the reachable sets of nonlinear control systems”. In: *Automatica* 49.1 (2013), pp. 93–100.

[137] David A Smith and Douglas R White. “Structure and dynamics of the global economy: network analysis of international trade 1965–1980”. In: *Social forces* 70.4 (1992), pp. 857–893.

[138] Michael W. Sneddon, James R. Faeder, and Thierry Emonet. “Efficient modeling, simulation and coarse-graining of biological complexity with NFsim”. In: *Nature Methods* 8.2 (2011), pp. 177–183.

[139] Thomas J. Snowden, Piet H. van der Graaf, and Marcus J. Tindall. “Methods of Model Reduction for Large-Scale Biological Systems: A Survey of Current Methods and Trends”. In: *Bulletin of Mathematical Biology* 79.7 (2017), pp. 1449–1486.

[140] Giuseppe Squillace, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “An Algorithm for the Formal Reduction of Differential Equations as Over-Approximations”. In: *Quantitative Evaluation of Systems*. Ed. by Erika Ábrahám and Marco Paolieri. Cham: Springer International Publishing, 2022, pp. 173–191. ISBN: 978-3-031-16336-4.

[141] Augustinas Sukys, Kaan Öcal, and Ramon Grima. “Approximating solutions of the Chemical Master equation using neural networks”. In: *iScience* 25.9 (2022), p. 105010. ISSN: 2589-0042. DOI: https://doi.org/10.1016/j.isci.2022.105010. URL: https://www.sciencedirect.com/science/article/pii/S2589004222012822.

[142] Mikael Sunnåker, Gunnar Cedersund, and Mats Jirstrand. “A method for zooming of nonlinear models of biochemical systems”. In: *BMC systems biology* 5.1 (2011), pp. 1–21.

[143] Andrei Nikolaevich Tikhonov. “Systems of differential equations containing small parameters in the derivatives”. In: *Matematicheskii Sbornik. Novaya Seriya* 31.73 (1952), pp. 575–586.

[144] Stefano Tognazzi, Mirco Tribastone, Max Tschaikowski, and Andrea Vandin. “Differential Equivalence for Linear Differential Algebraic Equations”. In: *IEEE Transactions on Automatic Control* 67.7 (2022), pp. 3484–3493. DOI: 10.1109/TAC.2021.3108530.

[145] Bertrand Tondu. “A zonotope-based approach for manipulability study of redundant robot limbs”. In: *International Journal of Humanoid Robotics* 10.03 (2013), p. 1350023.

[146] Mirco Tribastone. “Behavioral relations in a process algebra for variants”. In: *Proceedings of the 18th International Software Product Line Conference*. Ed. by Stefania Gnesi, Alessandro Fantechi, Patrick Heymans, Julia Rubin, Krzysztof Czarnecki, and Deepak Dhungana. ACM, 2014, pp. 82–91.

[147] Max Tschaikowski and Mirco Tribastone. “A unified framework for differential aggregations in Markovian process algebra”. In: *Journal of Logical and Algebraic Methods in Programming* 84.2 (2015), pp. 238–258.

[148] Max Tschaikowski and Mirco Tribastone. “Approximate Reduction of Heterogenous Nonlinear Models With Differential Hulls”. In: *IEEE Transaction on Automatic Control* 61.4 (2016), pp. 1099–1104.

[149] Max Tschaikowski and Mirco Tribastone. “Exact fluid lumpability for Markovian process algebra”. In: *International Conference on Concurrency Theory*. LNCS. 2012, pp. 380–394.

[150] Max Tschaikowski and Mirco Tribastone. “Spatial fluid limits for stochastic mobile networks”. In: *Performance Evaluation* 109 (2017), pp. 52–76.

[151] Max Tschaikowski and Mirco Tribastone. “Tackling continuous state-space explosion in a Markovian process algebra”. In: *Theoretic Computer Science* 517 (2014), pp. 1–33.

[152] Ke Tu, Peng Cui, Xiao Wang, Philip S Yu, and Wenwu Zhu. “Deep recursive network embedding with regular equivalence”. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 2357–2366.

[153] Tamás Turányi and Alison S. Tomlin. “Reduction of Reaction Mechanisms”. In: *Analysis of Kinetic Reaction Mechanisms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 183–312.

[154] Antti Valmari and Giuliana Franceschinis. “Simple $O(m \log n)$ Time Markov Chain Lumping”. In: *TACAS*. 2010.

[155] Tomáš Vejchodskỳ, Radek Erban, and Philip K Maini. “Reduction of chemical systems by delayed quasi-steady state assumptions”. In: *arXiv preprint arXiv:1406.4424* (2014).

[156] Pu Wang, Marta C González, César A Hidalgo, and Albert-László Barabási. “Understanding the spreading patterns of mobile phone viruses”. In: *Science* 324.5930 (2009), pp. 1071–1076.

[157] Duncan J Watts and Steven H Strogatz. “Collective dynamics of ‘small-world’networks”. In: *nature* 393.6684 (1998), pp. 440–442.

[158] James Wei and James CW Kuo. “Lumping analysis in monomolecular reaction systems. analysis of the exactly lumpable system”. In: *Industrial & Engineering chemistry fundamentals* 8.1 (1969), pp. 114–123.

[159] Max Whitby, Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, Mirco Tribastone, and Max Tschaikowski. “PID Control of Biochemical Reaction Networks”. In: *IEEE Transaction on Automatic Control*. 67.2 (2022), pp. 1023–1030.

[160] Douglas R White and Karl P Reitz. “Graph and semigroup homomorphisms on networks of relations”. In: *Social Networks* 5.2 (1983), pp. 193–234.

[161] R. Zafarani and H. Liu. *Social Computing Data Repository at ASU*. 2009. URL: `http://socialcomputing.asu.edu`.

[162] Aleš Ziberna. "Direct and Indirect Approaches to Blockmodeling of Valued Networks in Terms of Regular Equivalence". In: *The Journal of Mathematical Sociology* 32.1 (2008), pp. 57–84.

[163] Aleš Žiberna. "Evaluation of direct and indirect blockmodeling of regular equivalence in valued networks by simulations". In: *Advances in Methodology and Statistics* 6.2 (2009), pp. 99–134.

[164] Aleš Žiberna. "Generalized blockmodeling of valued networks". In: *Social networks* 29.1 (2007), pp. 105–126.