INSTITUT POLYTECHNIQUE DE PARIS

Thèse de Habilitation à Diriger des Recherches

ÉCOLE POLYTECHNIQUE

IP PARIS

Symbolic Transformations of Dynamical Models

Habilitation à Diriger des Recherches de l'Institut Polytechnique de Paris
préparée à École Polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat: Informatique, données, intelligence artificielle

Thèse présentée et soutenue à Palaiseau, le 05/12/2024, par

Gleb Pogudin

Composition du Jury :

François Boulier Professeur, Université de Lille	Rapporteur
François Fages Directeur de Recherche, INRIA	Rapporteur
Joris van der Hoeven Directeur de Recherche, LIX, CNRS	Examinateur
Françoise Point Professeur, Université de Mons	Examinatrice
Daniel Robertz Professor, RWTH Aachen University	Examinateur
Bernd Sturmfels Scientific director, Max Planck Institute for Mathematics in the Sciences	Rapporteur
Jacques-Arthur Weil Professeur, Université de Limoges	Examinateur

Abstract

This thesis is focused on symbolic algorithms for dynamical models defined by differential or difference equations. Such algorithms aim at complementing traditional numerical tools, they are exact and often operate on the level of symbolic expressions.
In the context of differential and difference equations, perhaps the most well-known symbolic algorithms are the ones for finding closed-form solutions which are available in many scientific software packages. However, a large portion (if not majority) of the equations appearing in the modeling literature do not admit such solutions. This fact does not render the symbolic methods useless. On the contrary, there is a number of ways to transform a model on the symbolic level to facilitate its further analysis. In this thesis, we discuss the following problems of this type:

- *eliminating* a subset of the variables (for example, the latent ones) from a model;
- assessing *structural identifiability* of parameters, that is, checking if the parameter values can be inferred uniquely from input-output data, and transforming a model into a one with better identifiability properties;
- performing *exact model reduction*, that is mapping a model into a one of lower dimension without introducing approximation errors;
- *quadratizing* a model, that is embedding the model into a one with at most quadratic nonlinearities.

The results we present for these problems range from theorems and theoretical algorithms to practical software implementations and case studies.

Keywords
symbolic computation, differential equations, difference equations, differential algebra, elimination, exact model reduction, parameter identifiability, quadratization

Acknowledgements

I would like to thank François Boulier, François Fages, Joris van der Hoeven, Françoise Point, Daniel Robertz, Bernd Sturmfels, and Jacques-Arthur Weil for kindly accepting to serve as jury members and taking time to evaluate my work. In particular, I am grateful to the rapporteurs, François Boulier, François Fages, and Bernd Sturmfels, for their detailed and encouraging reports. I would also like to thank Grégoire Lecerf and Marc Mezzarobba for their valuable feedback on the draft of the manuscript. I am grateful to Benjamin Doerr and Sergio Mover for their help and advice on the formal and administrative details of the HDR process.

The manuscript covers research results obtained during my three years of postdoc and five more years at the Laboratoire d'Informatique de l'École Polytechnique. In the first part of the journey, I had an incredible opportunity to benefit from support and guidance of my postdoc mentors: Manuel Kauers, Hoon Hong, Alexey Ovchinnikov, and Chee Yap. The subsequent years at LIX would be definitely not as productive and enjoyable without welcoming and intellectually stimulating atmosphere created by my colleagues at the MAX team. While condicting the research described in this mémoire, I had a privilege to work with and learn from my collaborators: S. Abramov, E. Amzallag, M. Bessonov, M. Buchacher, E.S. Golod, C. Goodbrake, R. Gustavson, H.A. Harrington, I. Ilmer, O. Issan, A. Jiménez-Pastor, B. Kramer, T. Konstantinova, I. Koswara, R. Laubenbacher, W. Li, A. Minchenko, T. Novikova, I. Pérez Verona, A.Pillay, J. C. Ramella-Roman, T. Scanlon, S. Selivanova, P.Soto, M. Sun, A. Szanto, P. Thompson, M. Tribastone, T.N. Vo, M. Wibmer, M. Ziegler; I am very grateful to all of them. I would like to thank separately my student collaborators: Rida Ait El Manssour, Foyez Alauddin, Andrei Bychkov, Yubo Cai, Ruiwen Dong, Natali Gogishvili, Joshua P. Jacob, Sasha Demin, Liza Demitraki, Dima Pavlov, Eli Paul, William Qin, Xingjian Zhang. Working with them has always been a source of inspiration, fresh viewpoints, and fun.

I would never get even to the starting point of this work without my math teachers: Svetlana Trepakova and Vladimir Sharich. Retrospectively, I think that my interest in formal and algebraic approach to analysis started with unforgettable lectures in calculus given by Evgenii Shchepin. And, of course, I would like to express a deep gratitude to my Ph.D. advisor, Yuri Pitirimovich Razmyslow, the research momentum acquired during my work with him remains one of the main forces moving me forward.

Last but not least, my very special thanks go to my family, especially my parents, my wife Katia and my children Fedor, Agatha, Érica, and Véra (the last two joined the team during the writing of the present manuscript). Nothing of what I have done would be possible without their support and the light they bring to my life; and nothing would make much sense without them.

Table of Contents

1. Introduction

Dynamical models formulated in terms of differential or difference equations is one of the key tools for representing real-world processes and phenomena in continuous and discrete time, respectively. Nature (almost likely) has structure. Models, especially good ones, can inherit it. Analyzing and revealing this structure are important steps towards understanding and successfully applying dynamical models.

Two fundamental examples of this paradigm that go back to the 19th century and are centered around the symmetries of differential equations are Lie group analysis of differential equations [91] and differential Galois theory [101]. Starting with the works of Ritt [104, 105] and Kolchin [67], the latter has been developed into a more general theory, *differential algebra*, aiming at providing a framework for studying the structure of nonlinear differential equations beyond their symmetry properties. The general philosophy behind many of these works was to extend commutative algebra and algebraic geometry, traditionally applied to polynomial equations, to the differential (and, later, difference [26]) ones.

While being originally developed inside mathematics, the language and tools from differential/difference algebra made their way to the sciences and engineering with one of the pioneering application areas being control theory in the 1980s-1990s [32, 41, 54, 81, 86, 89, 124]. The availability of dedicated software tools was very limited at the time, so these earlier applications were mostly restricted to relatively small models. The next important chapter in this story begins in the 1990s-2000s when computational tools for differential algebra started to be developed systematically [6, 15, 17, 59, 123]. These algorithmic and software advances allowed easier dissemination of the language and methodology leading to applications of constructive differential algebra also in mathematical biology [10, 83], cyber-physical systems [45, 114], and scientific machine learning [29, 70].

Theoretical, computational, and applied aspects of differential algebra mentioned above provide the background and serve as a staring point to the research described in the present thesis. Furthermore, many of the contributions presented here can be naturally positioned with respect to these three axes. On the other hand, from a higher-level perspective, the binding theme of the present manuscript is *revealing and exploiting the structure of a model through a model transformation*, and this general way to state the task calls for a more holistic choice of mathematical and computational tools to be used. In particular, depending on a specific problem, methods of differential algebra are augmented with tools from algebraic geometry, mathematical logic, combinatorics, and representation theory, and classical symbolic computation algorithms are being combined with discrete optimization and numerical computing. To be more explicit, let us review the results presented in this thesis chapter by chapter.

Chapter 2 describes some key concepts from differential and difference algebra relevant for this thesis as well and some of our theoretical results in this domain. In particular, we discuss that the rings of analytic functions and numerical sequences typically used in modeling of continuous- and discrete-time systems, respectively, are universal solution spaces in the sense of differential/difference algebra. This indicates that the chosen theoretical tools are indeed adequate for the target application problems.

Using a geometric analogy, the transformations discussed in this thesis can be informally classified into *projections* and *liftings*. Perhaps, the most natural appearance of the projection motive in the context of models defined by differential and difference equations is the *elimination problem* which is discussed in Chapter 3. Elimination problem asks, for a given system of equations, to derive relations between a chosen subset of variables (in applications, the variables of interest or the variables for which data is available). The results presented in Chapter 3 include effective bounds reducing differential/difference elimination to the classical polynomial elimination and practical algorithmic results. The lifting theme appears episodically in Section 3.5 where we study a problem of representing a nonlinear differential equation as a projection of a rational dynamical model (also known as *realization problem* in control theory). We finish Chapter 3 with analogues of classical algebraic results of elimination flavor: differential Noether normalization lemma and primitive theorem for difference-differential fields.

The projection theme continues further into Chapter 4 which is devoted to *structural parameter identifiability*. For a model defined by parametric differential equations, a parameter is called structurally identifiable if its value can be inferred uniquely from the input-output data assuming the absence of noise and sufficiently exciting inputs. The problem is naturally connected to elimination since it can be viewed as a study of the relation involving only the parameter of interest and input and output variables of the model. The intuitive notion of identifiability may be formalized in several ways, and not all of them are equivalent. This has caused significant amount of confusion in the literature, and the first contribution presented in Chapter 4 is a series of results establishing the exact relation between several popular definitions of identifiability and even giving new semantics to some of them. The second contribution described in Chapter 4 consists of two new algorithms for assessing structural parameter identifiability. The corresponding software tools allowed tackling problems which were out of reach before. The algorithms are very different but both have elimination techniques of different flavor under the hood. The corresponding software packages, `SIAN` and `StructuralIdentifiability.jl`, are both being used in the modeling literature from mathematical biology [30, 120] to cyber-security [24], and a recent survey [103] of tools for structural identifiability recommends them as the best choices for assessing global identifiability. Chapter 4 is concluded by reporting work in progress on the problem of transforming a non-identifiable model into an identifiable one.

We take a different point of view on projecting dynamical models in Chapter 5 about *exact model reduction*. For a model described by ordinary differential equations (ODEs), the problem consists of finding a map to a lower-dimensional ODE model such that the map respects the differential structure. Therefore, compared to the elimination problem, the focus is shifted from computing a prescribed projection to *finding convenient direction to project along*. Most of Chapter 5 is focused on linear exact reduction. In this case, the projection one looks for is a linear projection of the state space of the model. We present several algorithmic results for this problem including first practical algorithm allowing searching among all linear reductions. The key theoretical tool here is the structure theory of finite-dimensional algebras which, being combined with sparse linear algebra, allows performing computations for sophisticated models coming from systems biology.

We conclude Chapter 5 with some recent progress on computing nonlinear exact reductions. This task turns out to be closely related to the problem of transforming a non-identifiable model to an identifiable one from the end of Chapter 4.

Quadratization problem is the main topic of Chapter 6 in which the *lifting* idea manifests itself most explicitly. While the transformations considered in the preceding chapters ultimately aim at reducing model's "dimension" (in the sense of the number of variables of the order of the model), here we are interested in reducing the *degree* of the nonlinearities in the model. More precisely, quadratization of an ODE model is its embedding into a model with at most quadratic dynamics or, in other words, a *lifting* of a model to a quadratic one. We present the first algorithm for finding an optimal (of lowest possible dimension) quadratization for the case when the lifting map is defined by monomials and extend this approach to models with control and to stability-preserving quadratizations. Then we show how to apply the developed methodology to high-dimensional models arising as semi-discretizations of partial differential equations (PDEs), and showcase the resulting algorithm on a model of solar wind.

The last chapter of the thesis, Chapter 7, offers an outlook to promising directions for future research discussing possible ways to circumvent existing computational bottlenecks, tighten the connections between the developed symbolic and algebraic methods with applications, and explore the potential of the developed point of view in new application domains.

Let us conclude the introduction with an account of the *chronology* of the results presented in this manuscript and their position *with respect to my PhD thesis* in Mathematics entitled "Prime differential algebras and Lie algebras associated to them" defended in 2016 [98]. Significant part of my PhD research was devoted to obtaining differential analogues of classical algebraic results, most notably the primitive element theorem [95]. Computation was not among the main themes of my PhD thesis although I used computer calculations and Gröbner bases in proofs [97, 99]. I have returned to these theoretical questions during the subsequent years [G2, G3, G37, G38, G39] partly motivated by my research on computational and applied aspects of differential algebra.

My first big research direction after receiving PhD, differential and difference elimination discussed in Chapter 3, follows the same pattern as my doctoral thesis: I worked on differential and difference analogues of classical problem in computational algebra. In particular, effective bounds from Sections 3.1, 3.2, and 3.3 can be viewed as a continuation of the tradition of effective Nullstellensätze [63, 69, 73]. However, when further diving into more applied problems (in particular, in control theory) related to differential elimination such as realizability (Section 3.5) and identifiability (Chapter 4), I have found out that "typical" differential and difference equations appearing in applications in many aspects are very different from "typical" polynomial systems. Because of this, direct analogies with the classical algebraic equations do not give much insight into the dynamical situation anymore. Addressing this challenge opened a new chapter in my research when the starting point of an investigation was not a natural question inspired by existing theory but an applied problem with strong structural/algebraic flavor. My more recent research projects, on exact model reduction (Chapter 5) and on quadratization (Chapter 6), emerged this way. In particular, their connection to differential algebra, which was my main field of research during the PhD studies and shortly after, is weaker, and other mathematical tools like representation theory (for exact model reduction) and combinatorics (for quadratization) come to the forefront.

2. Differential and Difference Algebra

Most of dynamical models discussed in the thesis will be defined by algebraic differential or difference equations. Differential/difference algebra provides us a convenient language to operate with such equations. The goal of this chapter will be to set up the notation from differential/difference algebra which will be used throughout the text and to survey some basic facts from differential/difference algebra relevant to our discussion.

2.1 Differential/difference rings and fields

Solutions of differential and difference equations can be sought in various structures including smooth functions, power series, trigonometric series, etc. If one wants to plug an element of some structure into an algebraic differential/difference equation, such a structure has to be equipped with arithmetic operations and derivation/shift. This motivates the notions of differential/difference rings.

Definition 2.1.1 — Differential rings and fields.

- Let R be a commutative ring. An additive map $\delta\colon R \longrightarrow R$ is called *a derivation* if it satisfies the Leibniz rule:

$$\delta(ab) = \delta(a)b + a\delta(b) \qquad \text{for every } a, b \in R.$$

  We will typically denote $\delta(a)$ by a' and, for $n \geqslant 0$, $\delta^n(a)$ by $a^{(n)}$.
- A commutative ring equipped with a derivation is called *a differential ring*. If a differential ring is a field, it is called *a differential field*.
- A commutative ring equipped with several pairwise commuting derivations is called *partial differential ring*. If the set of derivations is denoted by $\Delta = \{\partial_1, \ldots, \partial_m\}$, such a ring will also be called a Δ*-ring*.
- For a differential ring, a subring closed under the derivation is called *a differential subring*. Differential subfields are defined in the same way. A differential ring which is an algebra over its differential subfield, is called *a differential algebra*. Δ-subring, Δ-subfield, and Δ-algebra are defined in the same way.

▪ **Example 2.1 — Differential rings.**

- Any ring can be considered as a differential ring with respect to the zero derivation or even as a partial differential ring with respect to several zero derivations.
- Consider the ring $\mathbb{C}[x]$ and the field $\mathbb{C}(x)$. They are a differential ring and field with respect

to $\frac{\mathrm{d}}{\mathrm{dx}}$, respectively. Moreover, they are differential algebras over the constant field $\mathbb{C}$.
- Let $D \subset \mathbb{C}$ be a domain in the complex plane. By $\mathrm{Hol}(D)$ and $\mathrm{Mer}(D)$ we denote the set of all holomorphic and meromorphic functions in D, respectively. They are a differential ring and a differential field with respect to $\frac{\mathrm{d}}{\mathrm{dz}}$, respectively.
- The examples above naturally extend to the partial differential case, for example, the field $\mathbb{C}(x_1,\dots,x_m)$ of rational functions is a Δ-field for $\Delta = \{\partial_1,\dots,\partial_m\}$ if we take $\partial_i = \frac{\mathrm{d}}{\mathrm{dx}_i}$ for $1 \leqslant i \leqslant m$. ■

Definition 2.1.2 — Difference rings.
- Let R be a commutative ring. An additive map $\sigma\colon R \to R$ is called *an endomorphism* if $\sigma(1) = 1$ and

$$\sigma(ab) = \sigma(a)\sigma(b) \quad \text{for every } a,b \in R.$$

- A commutative ring equipped with an endomorphism is *a difference ring*. If a difference ring is a field, it is called *a difference field*.
- For a difference ring, a subring closed under the endomorphism is called *a difference subring*. Difference subfields are defined in the same way. A difference ring which is an algebra over its difference subfield, is called *a difference algebra*.

One can naturally extend Definition 2.1.2 to the setting with several commuting endomorphisms (thus defining *partial difference rings*). Such objects will be mentioned only tangentially in this text, so we will not formally define them here.

■ **Example 2.2 — Difference rings.**
- Any ring can be considered as a difference ring with respect to the identity endomorphism.
- Consider the ring $\mathbb{C}[x]$ and the field $\mathbb{C}(x)$. They are a difference ring and field with respect to $\sigma(f(x)) = f(x+1)$, respectively. They are also difference algebras over the constant field $\mathbb{C}$.
- The ring or meromorphic functions on a complex plane is a difference ring with respect to a shift $f(z) \to f(z+1)$ or a dilation $f(z) \to f(qz)$ (where $q \in \mathbb{C}$ is a fixed constant).
- Let R be any ring. Then the ring of two-sided infinite sequences $R^{\mathbb{Z}}$ is a difference ring with respect to the shift operator, that is, $\sigma(\{a_i\}_{i\in\mathbb{Z}}) = \{a_{i+1}\}_{i\in\mathbb{Z}}$. ■

2.2 Differential/difference polynomials

Typical algebraic differential/difference equations are polynomial expressions in unknown functions and their derivatives/shifts. This idea can be formalized using the notion of differential/difference polynomial.

Notation 2.1. *Let x be an element of a differential ring and h be a nonnegative integer. Then we introduce*

$$x^{(<h)} := (x, x', \dots, x^{(h-1)}), \quad x^{(\infty)} := (x, x', x'', \dots).$$

Notation $x^{(\leqslant h)}$ is defined analogously. This notation can be extended to tuples $\mathbf{x} = (x_1,\dots,x_n)$ as

$$\mathbf{x}^{(<h)} := (x_1^{(<h)},\dots,x_n^{(<h)}), \quad \mathbf{x}^{(\infty)} := (x_1^{(\infty)},\dots,x_n^{\infty}).$$

Definition 2.2.1 — Differential polynomials. Let R be a differential ring. Consider a ring of polynomials in infinitely many variables

$$R[x^{(\infty)}] := R[x, x', x'', x^{(3)}, \ldots]$$

and extend the derivation from R to this ring by $(x^{(j)})' := x^{(j+1)}$. The resulting differential ring is called *the ring of differential polynomials in x over R*. The ring of differential polynomials in several variables is defined by iterating this construction.

▪ **Example 2.3** Weierstrass's elliptic function $\wp(z)$ satisfies the following differential equation: $(\wp'(z))^2 = 4\wp(z)^3 - g_2\wp(z) - g_3$. This equation can be written as the following differential polynomial over a constant field $\mathbb{Q}(g_2, g_3)$: $(x')^2 - 4x^3 - g_2 x - g_3 \in \mathbb{Q}(g_2, g_3)[x^{(\infty)}]$. ▪

In order to extend this definition to the partial differential case, we fix $\Delta = \{\partial_1, \ldots, \partial_m\}$.

Notation 2.2. *Let x be an element of a Δ-ring and $\mathbf{h} = (h_1, \ldots, h_m)$ be a tuple of elements of $\mathbb{Z}_{\geqslant 0} \cup \{\infty\}$. We introduce*

$$x^{(<\mathbf{h})} := \{\partial_1^{e_1} \ldots \partial_m^{e_m} x \mid \forall\, 1 \leqslant i \leqslant m\colon 0 \leqslant e_i < h_i\}.$$

Notation $x^{(\leqslant \mathbf{h})}$ is defined analogously. If we denote $\boldsymbol{\infty} := (\underbrace{\infty, \ldots, \infty}_{m \text{ times}})$, then $x^{(\boldsymbol{\infty})}$ will be the set of all partial derivatives of x. Similarly to Notation 2.1, this notation can be extended to tuples $\mathbf{x} = (x_1, \ldots, x_n)$.

Definition 2.2.2 — Partial differential polynomials. Consider a Δ-ring R. Then the Δ-ring structure can be extended to the infinitely-dimensional polynomial ring

$$R[x^{(\boldsymbol{\infty})}] := R[\partial_1^{e_1} \ldots \partial_m^{e_m}(x) \mid \forall\, 1 \leqslant i \leqslant m\colon e_m \in \mathbb{Z}_{\geqslant 0}]$$

by setting $\partial_i(\partial_1^{e_1} \ldots \partial_i^{e_i} \ldots \partial_m^{e_m}(x)) := \partial_1^{e_1} \ldots \partial_i^{e_i+1} \ldots \partial_m^{e_m}(x)$. The resulting Δ-ring is called *the ring of Δ-polynomials in x over R*. The ring of Δ-polynomials in several variables is defined by iterating this construction.

▪ **Example 2.4** Let $m = 2$ and R be a Δ-ring. Then the Jacobian of x_1, x_2, that is

$$\begin{vmatrix} \partial_1(x_1) & \partial_2(x_2) \\ \partial_1(x_2) & \partial_2(x_2) \end{vmatrix} = \partial_1(x_1)\partial_2(x_2) - \partial_1(x_2)\partial_2(x_1),$$

is an element of the bivariate Δ-polynomial ring $R[x_1^{(\boldsymbol{\infty})}, x_2^{(\boldsymbol{\infty})}]$. ▪

Notation 2.3. *Let x be an element of a difference ring and h be a nonnegative integer. Then we introduce*

$$\sigma^{<h}(x) := (x, \sigma(x), \ldots, \sigma^{h-1}(x)), \quad \sigma^{\infty}(x) := (x, \sigma(x), \sigma^2(x), \ldots).$$

Notation $\sigma^{\leqslant h}(x)$ is defined analogously. As Notation 2.1, this notation can be naturally extended to tuples $\mathbf{x} = (x_1, \ldots, x_n)$.

Definition 2.2.3 — Difference polynomials. Let R be a difference ring. Consider a ring of polynomials in infinitely many variables

$$R[\sigma^\infty(x)] := R[x, \sigma(x), \sigma^2(x), \sigma^3(x), \ldots]$$

and extend the endomorphism σ from R to this ring by $\sigma(\sigma^j(x)) := \sigma^{j+1}(x)$. The resulting difference ring is called *the ring of difference polynomials in x over R*. The ring of difference polynomials in several variables is defined by iterating this construction.

▪ **Example 2.5** Fibonacci sequence $\{f_i\}_{i\in\mathbb{Z}} = (\ldots, 1, 1, 2, 3, 5, \ldots)$ can be viewed as an element of the difference ring $\mathbb{Q}^\mathbb{Z}$ with respect to shift (see Example 2.2). Then the recurrence $f_{n+2} = f_{n+1} + f_n$ can be expressed as a difference polynomial $\sigma^2(x) - \sigma(x) - x \in \mathbb{Q}[\sigma^\infty(x)]$. ▪

2.3 Differential ideals and their solutions

If a tuple of functions satisfies several algebraic differential equations, that is, the tuple is a zero of several differential polynomials, then it is also a zero of any combination of the derivatives of these polynomials. This idea is formalized using the notion of differential ideal.

Definition 2.3.1 — Differential ideals. Let R be a differential ring (resp., Δ-ring) and $I \subset R$ be an ideal. Then I is called a *differential ideal* (resp., Δ-*ideal*) if $a' \in I$ for every $a \in I$ (resp., $\partial_i(a) \in I$ for every $1 \leqslant i \leqslant m$).

For every elements $a_1, \ldots, a_\ell$, the smallest differential ideal (resp., Δ-ideal) containing $a_1, \ldots, a_\ell$ is the ideal generated by $a_1^{(\infty)}, \ldots, a_\ell^{(\infty)}$ (resp., $a_1^{(\boldsymbol{\infty})}, \ldots, a_\ell^{(\boldsymbol{\infty})}$). We will denote it by $\langle a_1, \ldots, a_\ell\rangle^{(\infty)}$ (resp., $\langle a_1, \ldots, a_\ell\rangle^{(\boldsymbol{\infty})}$).

▪ **Example 2.6** Consider a differential polynomial $x' - x$. Its solutions in the differential ring of formal power series $\mathbb{C}[[t]]$ (with the standard derivation $\frac{\mathrm{d}}{\mathrm{d}t}$) are exactly power series of the form ae^t for $a \in \mathbb{C}$. Clearly, these series are also zeros of $(x' - x)^{(i)} = x^{(i+1)} - x^{(i)}$ for every i as well as any other element of $\langle x' - x\rangle^{(\infty)}$. ▪

▪ **Example 2.7** We now extend the differential ideal from the previous example with one more generator $x'' - 1$. The solutions of $x'' - 1$ in $\mathbb{C}[[t]]$ are the quadratic functions of the form $\frac{1}{2}t^2 + at + b$, so the system $x' - x = x'' - 1 = 0$ does not have a solution in this differential ring. This can be also seen as follows. The ideal contains 1 because

$$(x'' - 1)' - (x'' - 1) - (x' - x)'' = 1,$$

so existence of a zero in any differential ring would imply the impossible identity $1 = 0$. ▪

As the following theorem (following from [64, Theorem 2.1]) shows, Example 2.7 is prototypical in the sense that the presence of 1 in the ideal is equivalent to the inconsistency. We will formulate it for Δ-fields, and this will include ordinary differential fields for $|\Delta| = 1$.

Theorem 2.3.1 — Differential Nullstellensatz, abstract version. Let k be a Δ-field and let $I \subset k[\mathbf{x}^{(\boldsymbol{\infty})}]$ be a Δ-ideal. Then there exists a Δ-field $K \supset k$ such that I has a solution in K if and only if $1 \notin I$.

For more specific choice of k, one can say more about the sufficient K (folklore result, see [96, Proposition 2.4] for a proof).

Theorem 2.3.2 — Differential Nullstellensatz, power series. Let $I \subset \mathbb{C}[\mathbf{x}^{(\infty)}]$ be a differential ideal with $1 \notin I$. Then I has a solution in the power series ring $\mathbb{C}[[t_1, \dots, t_m]]$.

▪ **Example 2.8 — Constant coefficients are essential.** Consider a differential ideal $\langle x^2 - t \rangle^{(\infty)} \subset \mathbb{C}[[t]][x^{(\infty)}]$. Since there is no power series $x(t)$ with $x^2(t) = t$, the ideal does not have a solution in $\mathbb{C}[[t]]$. On the other hand, the ideal does not contain 1 by Theorem 2.3.1 because it has a solution $x = \sqrt{t}$ in $\mathbb{C}[[\sqrt{t}]]\left[\frac{1}{\sqrt{t}}\right] \supset \mathbb{C}[[t]]$. ▪

The limitation illustrated by Example 2.8 can be avoided if one considers meromorphic functions as coefficient and solution space. The following Ritt's theorem of zeros and a related Seidenberg's embedding theorem [112] are the core facts relating differential-algebraic considerations with the realm of analysis.

Theorem 2.3.3 — Ritt's theorem of zeroes [104, p. 176]. Let $\mathcal{D} \subset \mathbb{C}^m$ be a domain, and let R be the Δ-field of meromorphic functions on this domain. Consider a differential ideal $I \subset R[\mathbf{x}^{(\infty)}]$ such that $1 \notin I$. Then there exists a domain $\mathcal{D}_0 \subset \mathcal{D}$ such that I has a solution in the field of meromorphic functions in $\mathcal{D}_0$.

R *Old theorems, new proofs.* While Theorems 2.3.1 and 2.3.2 are quite straightforward to prove, the classical proofs of Ritt's theorem and a closely related Seidenberg's theorem were relying on the Riquier theory from PDEs and the machinery of characteristics sets scattered over the book by Ritt [104] making them accessible only for the experts. With Yu.P. Razmyslow and our student D. Pavlov, we have proposed a new short and elementary proof of both results relying only on basic algebra and the standard Cauchy-Kovalevskaya theorem [G37].

R *Non-radical ideals.* As long as one is interested in solutions in integral domains (like power series ring above), one can always replace an ideal with its radical. In a more general case, non-radical ideals come into play, and even the simplest ones like $\langle x^m \rangle^{(\infty)}$ have rich combinatorial structure explored in our works [G2, G3] with Rida Ait El Manssour (a PhD student).

2.4 Difference ideals and their solutions

Definition 2.4.1 — Difference ideals. Let R be a difference ring and $I \subset R$ be an ideal. Then I is called a *difference ideal* if $\sigma(a) \in I$ for every $a \in I$.

For every elements $a_1, \dots, a_\ell$, the smallest difference ideal containing $a_1, \dots, a_\ell$ is the ideal generated by $\sigma^\infty(a_1), \dots, \sigma^\infty(a_\ell)$.

We have seen in Theorem 2.3.2 that, for constant coefficient algebraic differential equations, formal power series play a role of a "universal solution domain". The following theorem shows that, in the difference setting, the ring of sequences (see Example 2.2) plays the same role.

Theorem 2.4.1 — A. Ovchinnikov, G. Pogudin, T. Scanlon [G32, Theorem 7.1]. Let $\mathbb{K}$ be a constant algebraically closed difference field, and consider a difference ideal $I \subset \mathbb{K}[\sigma^\infty(\mathbf{x})]$. Then I has a solution in the ring of sequences $\mathbb{K}^{\mathbb{Z}}$ if and only if $1 \notin I$.

While the statements of Theorems 2.4.1 and 2.3.2 look quite similar, the former is a much harder to prove and involves establishing effective difference Nulstellensatz and some tools from model theory.

Theorem 2.4.1 in the paper [G32] is given for not necessarily constant field $\mathbb{K}$ but the statement is slightly more technical — one has to "twist" σ on the ring of sequences. Under additional assumption that the cardinality of $\mathbb{K}$ is large enough (typically, uncountable), Theorem 2.4.1 was generalized in a joint paper with T. Scanlon and M. Wibmer to the partial difference case (and even further) [G42, Theorem 3.1]. We also demonstrate a striking difference between the difference and partial difference case which does not occur in the differential world: while the existence of a sequence solution can be algorithmically checked for difference equations [G32, Theorem 3.1], it is undecidable in the partial difference case [G42, Proposition 3.9].

3. Differential/Difference Elimination

Let us consider a system of equations (for example, linear, polynomial, differential, ...)

$$f_1(\mathbf{x},\mathbf{y}) = \ldots = f_n(\mathbf{x},\mathbf{y}) = 0$$

in two groups of unknowns $\mathbf{x} = (x_1,\ldots,x_s)$ and $\mathbf{y} = (y_1,\ldots,y_\ell)$. Then the *elimination problem* is to find a nontrivial equation $g(\mathbf{y}) = 0$ in $\mathbf{y}$ only, which holds for every solution of the system, or establish that there is no such equation. A more refined version of the problem would be to describe the set of all such equations. The most prominent examples of elimination methods include Gaussian elimination for linear equations and resultants for polynomial elimination going back to Cayley, Sylvester, and Macaulay.

R The elimination problem often includes, as a special case, the *consistency problem*: if one takes $\mathbf{y}$ to be empty, then the existence of nontrivial equation $g(\mathbf{y}) = 0$ is equivalent to the fact that the original system implies an impossible equality $1 = 0$, so the system does not have any solutions. On the other hand, if such equality does not follow from the original equations, one can often prove the existence of solutions in suitable domains (see Sections 2.3 and 2.4 for differential and difference equations).

3.1 Differential elimination via characteristic sets

In the context of differential equations, the elimination problem was already considered by Ritt, the founder of differential algebra, in the 1930-s, see [105]. He developed the foundations of the *characteristic set* approach, which has been made fully constructive by Seidenberg [113]. The algorithmic aspect of this research culminated in the Rosenfeld-Gröbner algorithm [17, 61] implemented in the BLAD library [16] (available through Maple). Further developments include EPSILON library [123] and differential Thomas decomposition [6, 106].

These methods use characteristic sets or their variations in order to perform the computation and represent the result of elimination. The definition is a bit technical (we refer to [61] for details) but the idea is the same as in the case of triangular sets used for polynomial computation [60] which, in turn, can be viewed as a far-reaching generalization of the row echelon form of a matrix. More precisely, the variables are ordered and the equations are transformed into triangular shape allowing to eliminate variables one by one.

Practical algorithms based on these ideas, most notably the aforementioned Rosenfeld-Gröbner algorithm, could be successfully applied to systems of moderate size but their scalability has been known to be limited. On the theoretical side, one of the most natural complexity measures is

the maximal order of the differential polynomial appearing during the computation or, in other words, how large the polynomial rings to consider may get. In the case of a single derivation, such a bound for the Rosenfeld-Gröbner algorithm was given in [48]: the order may grow at most by the factor $(n-1)!$, where n is the total number of variables. The first order bound for the Rosenfeld-Gröbner algorithm in the PDE case was given in our joint papers with R. Gustavson and A. Ovchinnikov [G14, G15]: we have found an explicit function $F(m,h,n)$[1] depending on the number of derivations m, the order of the input h, and the number of variables n such that the total order during the computation may grow at most by a factor of $F(m,h,n)$.

The question of bounding the size of the output of the Rosenfeld-Gröbner algorithm was later revisited in our paper with W. Li, A. Ovchinnikov, and T. Scanlon [G25]. By combining methods from model theory and theoretical computer science, we have shown that the *mere existence* of an algorithm such as Rosenfeld-Gröbner implies the existence of computable bounds on the output of the algorithm. We used this result to show that there is a computable bound for the number of prime components of a Δ-ideal in terms of the combinatorial data of its generators [G25, Theorem 3.1] which we then used to give a computable reduction of elimination for delay-PDE equations to polynomial elimination [G25, Theorem 3.2] in the spirit of Sections 3.2 and 3.3.

3.2 Bounds for effective differential elimination

Another approach to the differential elimination question, which can also be traced back to Ritt [105, p. 118], is to *reduce the differential elimination problem to polynomial elimination*. An attractive feature of this point of view is that one can take advantage of the existing powerful theoretical and practical tools for polynomial elimination. More precisely, this question in the ODE case can be stated as follows:

Problem 3.2.1 For a system of differential polynomials $f_1,\dots,f_\ell \in \mathbb{K}[\mathbf{x}^{(\infty)},\mathbf{y}^{(\infty)}]$, compute an *a priori* bound B such that

$$\langle f_1,\dots,f_\ell\rangle^{(\infty)} \cap \mathbb{K}[\mathbf{y}^{(\infty)}] \neq \{0\} \iff \langle f_1^{(\leqslant B)},\dots,f_\ell^{(\leqslant B)}\rangle \cap \mathbb{K}[\mathbf{y}^{(\infty)}] \neq \{0\}.$$

R Although the latter intersection belongs to a polynomial ring $\mathbb{K}[\mathbf{y}^{(\infty)}]$ in infinitely many variables, the ideal is generated by finitely many polynomials, so one can focus on a finitely generated subring containing the generators. Thus, solving Problem 3.2.1 gives an algorithm to check if a differential elimination ideal is nonzero.

First results towards Problem 3.2.1 were obtained for the case of $\mathbf{y}$ being empty, that is, for the *consistency problem*. The first bound was given in the general PDE case in [47] based on careful analysis of the Rosenfeld-Gröbner algorithm and involved very fast-growing functions. This result was substantially improved in [28] for the constant-coefficient ODE case: a doubly-exponential bound involving the degrees, orders, and number of variables was given. This result was based on novel ideas to work in certain truncated differential rings which allowed using methods from constructive algebraic geometry. This approach was further extended to nonconstant coefficients and PDEs in [53]. Under additional assumptions on the input system, differential resultants can offer another point of view on Problem 3.2.1, see [79, 108].

[1] The actual formula is complicated but, curiously, it involves Fibonacci numbers.

In the joint work [G28] with A. Ovchinnikov and T. Vo, we extended the results of [28] in two directions. First, we performed a more refined analysis of how the geometry of algebraic ideals in differential rings interacts with differentiation[2] and used it to derive a fully explicit bound sufficient to perform practical computations for some systems. Second, we have observed that being able to work with nonconstant coefficients (using techniques from [53]) allows to establish a bound for the elimination problem, not only for consistency checking. Here is a simplified version of the main result of the paper:

Theorem 3.2.2 — **A. Ovchinnikov, G. Pogudin, T. Vo [G28, Theorem 1].** Consider differential polynomials $f_1, \dots, f_\ell \in \mathbb{K}[\mathbf{x}^{(<h)}, \mathbf{y}^{(\infty)}]$ over a differential field $\mathbb{K}$ with the total degree with respect to $\mathbf{x}^{(<h)}$ at most d. Let m be the dimension of the (algebraic, not differential) ideal generated by $f_1, \dots, f_\ell$ in $\mathbb{K}(\mathbf{y}^{(\infty)})[\mathbf{x}^{(<h)}]$. Then

$$\langle f_1, \dots, f_\ell \rangle^{(\infty)} \cap \mathbb{K}[\mathbf{y}^{(\infty)}] \neq \{0\} \iff \langle f_1^{(\leqslant B)}, \dots, f_\ell^{(\leqslant B)} \rangle \cap \mathbb{K}[\mathbf{y}^{(\infty)}] \neq \{0\},$$

where $B = d^{(h|\mathbf{x}|+1-m)2^{m+1}}$ if $d > 1$ and $B = m+1$ if $d = 1$.

We give a tighter bound for the case when $f_1, \dots, f_\ell$ generate a radical algebraic ideal [G28, Theorem 1] and a bound allowing to compute any truncation $\langle f_1, \dots, f_\ell \rangle^{(\infty)} \cap \mathbb{K}[\mathbf{y}^{(<H)}]$ of the elimination ideal [G28, Theorem 3].

▪ **Example 3.1** — **[G28, Example 1].** Consider the predator-prey model

$$\begin{cases} x' = ax - bxy, \\ y' = -cy + dxy \end{cases} \tag{3.1}$$

defined by two polynomials $f_1 = x' - ax + bxy$ and $f_2 = y' + cy - dxy$ in $\mathbb{Q}(a,b,c,d)[x,x',y,y']$ over the constant field $\mathbb{Q}(a,b,c,d)$. In the notation of the theorem, we have $h = 2$ and $d = 1$ as we take the degree with respect to x *only*. For computing m, we consider an algebraic ideal generated by f_1, f_2 in a bivariate polynomial ring $\mathbb{Q}(a,b,c,d,y,y')[x,x']$. Since the f_1 and f_2 are coprime, the ideal is zero-dimensional, so $m = 0$. Theorem 3.2.2 yields a bound $B = 1$, and, indeed, after one differentiation we find

$$acy^2 + ayy' - bcy^3 - by^2y' - yy'' + (y')^2 \in \langle f_1, f_1', f_2, f_2' \rangle. \tag{3.2}$$

▪

3.3 Effective difference elimination

The elimination and consistency problems in the difference case exist in two substantially different setups:

- the solutions can be sought in the space of *sequences* (which is a "universal solution space" for difference equations by Theorem 2.4.1), or
- the solutions can be sought in *difference fields* (the ring of sequences is not a field and cannot be embedded into a field as it has zero divisors).

[2] I have further extended this analysis in my paper [G41]. The problem turned out to be connected to questions about jet schemes coming from the singularity theory.

In the latter case, the corresponding first-order theory is "well-behaving" (is decidable and admits quantifier elimination) [23], so in theory the elimination and consistency problems are decidable as well. Furthermore, there are some explicit algorithmic results in this direction [44, 78].

However, if one is interested in *sequence solutions* (for example, for discrete-time dynamical systems), which provide a "universal solution space" thanks to Theorem 2.4.1, then the corresponding first-order theory is undecidable [58], so there is no simple way to establish even the decidability of the elimination problem in the sequence case. Together with A. Ovchinnikov and T. Scanlon, we have established the first effective bound which, similarly to Theorem 3.2.2, allowed reducing difference elimination to polynomial.

Theorem 3.3.1 — **A. Ovchinnikov, G. Pogudin, T. Scanlon [G32, Theorem 3.4].** Consider difference polynomials $f_1, \dots, f_\ell \in \mathbb{K}[\sigma^{<h}(\mathbf{x}), \sigma^\infty(\mathbf{y})]$ over a difference field $\mathbb{K}$. Let m and D be the dimension and the degree of the (algebraic, not difference) ideal generated by $f_1, \dots, f_\ell$ in $\mathbb{K}(\sigma^\infty(\mathbf{y}))[\sigma^{<h}(\mathbf{x})]$. Then

$$\langle \sigma^\infty(f_1), \dots, \sigma^\infty(f_\ell)\rangle \cap \mathbb{K}[\sigma^\infty(\mathbf{y})] \neq \{0\} \iff \langle \sigma^{<B(m,D)}(f_1), \dots, \sigma^{<B(m,D)}(f_\ell)\rangle \cap \mathbb{K}[\sigma^\infty(\mathbf{y})] \neq \{0\},$$

where $B(0,D) = D+1$, $B(1,D) = \frac{D^3}{6} + \frac{D^2}{2} + \frac{4D}{3} + 1$, and $B(m,D) = B(m-1,D) + D^{B(m-1,D)}$ for $m > 1$.

▪ **Example 3.2** Let F_n be the n-th Fibonacci number. It turns out [34, p. 856] that the sequence $A_n := F_{2^n}$ satisfies a nonlinear difference equation. We can find it as follows. We introduce $B_n := F_{2^n+1}$. Then standard identities $F_{2k} = F_k(2F_{k+1} - F_k)$ and $F_{2k+1} = F_{k+1}^2 + F_k^2$ yield the following system of difference equations

$$A_{n+1} = A_n(2B_n - A_n), \quad B_{n+1} = A_n^2 + B_n^2. \tag{3.3}$$

Considered as polynomial equations in B_n and B_{n+1}, system (3.3) defines an affine variety of dimension zero and degree two over $\mathbb{Q}(A_n, A_{n+1})$. Theorem 3.3.1 implies that it is sufficient to consider system (3.3) and two of its shifts to eliminate B. Performing this computation, we indeed find a difference equation $5F_{2^n}^4 F_{2^{n+1}} - 2F_{2^n}^2 F_{2^{n+2}} + F_{2^{n+1}}^3 = 0$. ▪

In the case of the consistency problem (i.e., $|\mathbf{y}| = 0$), the theorem gives the first proof that the problem is decidable. Furthermore, it implies that, if there is a finite sequence of elements in $\mathbb{K}$ satisfying the system of length greater than $B(m,D)$, then there exists a full infinite sequence solution [G32, Corollary 3.2].

Corollary 3.3.2 The problem of checking, for given $f_1, \dots, f_\ell \in \mathbb{K}[\sigma^\infty(\mathbf{x})]$, the existence of solutions of $f_1 = \dots = f_\ell = 0$ in the ring $\mathbb{K}^\mathbb{Z}$ of sequence is decidable for an algebraically closed field $\mathbb{K}$.

Interestingly, as we have shown with T. Scanlon and M. Wibmer in [G42], a number of reasonable generalizations of the consistency problem are not decidable.

Theorem 3.3.3 — **G. Pogudin, T. Scanlon, M. Wibmer [G42].** The following problems are undecidable:

1. Given $f_1, \dots, f_\ell \in \mathbb{R}[\sigma^\infty(\mathbf{x})]$, check the existence of solutions of $f_1 = \dots = f_\ell = 0$ in $\mathbb{R}^\mathbb{Z}$.
2. Given $f_1, \dots, f_\ell, g \in \mathbb{C}[\sigma^\infty(\mathbf{x})]$, check the existence of solutions of $f_1 = \dots = f_\ell = 0$ & $g \neq 0$.
3. Check the existence of solutions for a system with two commuting automorphisms in the

ring of bivariate sequences.

The proof technique used to establish Theorem 3.3.1 was then successfully applied in [G26] by W. Li, A. Ovchinnikov, T. Scanlon, and myself to give an analogous bound for the elimination and consistency problems for delay-differential equations. We extended the result further to delay-PDEs in [G25, Theorem 3.2].

3.4 Practical differential elimination for ODE models

As described in Sections 3.1 and 3.2, several approaches to differential elimination have been proposed and resulted in practically useful algorithmic tools such as the Rosenfeld-Gröbner algorithms and its variations. These algorithms are quite general: they can be applied to an arbitrary system of polynomially nonlinear PDEs. There is a price to pay for such versatility: many interesting examples coming from applications cannot be tackled in a reasonable time. On the other hand, since differential equations in sciences and engineering are typically used to describe how the system of interest will evolve from a given state, many dynamical models in the literature are described by systems in *the state-space form*:

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}) \tag{3.4}$$

where $\mathbf{f}$ and $\mathbf{g}$ are tuples of rational functions, $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{u}$ are tuples of differential unknowns (the state, output, and input variables, respectively). For such a system, one typically wants to eliminate the x-variables, that is, compute the *input-output relations*, the relations between the y-variables and u-variables (see [27, 118]). For instance, the predator-prey model from Example 3.1 is problem of this sort with $\mathbf{x} = (x, y)$, no inputs, and the output being equal to y, i.e. $g(x, y) = y$.

In a joint work with R. Dong (a master student), C. Goodbrake, and H. Harrington [G12], we proposed a new approach to performing elimination tailored to the systems of the form (3.4). The idea was to use a different way of representing a prime differential ideal generalizing (3.4). Modulo the differential ideal defined by (3.4), $\mathbf{x}$ together with $\mathbf{u}^{(\infty)}$ form a transcendence basis and then each of the equations in (3.4) gives a relation between one of the lowest-order non-basis elements (such as $\mathbf{x}'$ or $\mathbf{y}$) in terms of the basis ones. If one allows to consider different transcendence bases, then the elimination can be rephrased as *a change of ordering* task. Building upon this idea, we developed an algorithm based on specially organized iterated resultant computation.

For benchmarking problems from the literature, our implementation significantly outperformed general purpose packages such as DifferentialAlgebra [16] and DifferentialThomas [6] performing eliminations for problems which were out of reach before, see Table 3.1 (for details of the comparison, see [G12, Section 6.3]). This algorithm was a basis of our new software for assessing structural parameter identifiability, see Section 4.5.

3.5 Anti-elimination

In the previous section, we discussed a problem to compute, for a system in the state-space form standard in control theory, the relations between inputs $\mathbf{u}$ and outputs $\mathbf{y}$ or, equivalently, to eliminate the state variables $\mathbf{x}$. Interestingly, an inverse problem which can be thought as *anti-elimination* also appears in control theory and is referred to as *realization problem* [110, 117], here is one way of stating it.

Model	DiffAlgebra	DiffThomas	New algorithm form [G12]
SIWR model	> 5 h.	> 5 h.	3 s.
SIWR model - 2	> 5 h.	> 5 h.	0.2 s.
MAPK pathway - 1	13 s.	7 s.	2.5 s.
MAPK pathway - 2	> 5 h.	> 5 h.	27.6 s.
MAPK pathway - 3	> 5 h.	> 5 h.	397 s.
SEAIJRC model	> 5 h.	> 5 h.	28.6 s.
Akt pathway	0.2 s.	> 5 h.	0.2 s.
NFκB	> 5 h.	> 5 h.	> 5 h.
Mass-action	4.7 s.	> 5 h.	< 0.1 s.
SIRS w. forcing	> 5 h.	> 5 h.	1 s.

Table 3.1: Comparison of the algorithm from [G12] with general purpose libraries for differential elimination (for details on examples and benchmarking, see [G12, Table 5])

Problem 3.5.1 — Realization problem. Given a differential polynomial $p(y, \mathbf{u}) \in \mathbb{K}[y^{(\infty)}, \mathbf{u}^{(\infty)}]$, find a dynamical system of the form (3.4) such that p is the minimal (compared first by order and then by degree) relation between y and $\mathbf{u}$ in the ideal generated by the system. If such system exists, then p is said to be *realizable*. Often, an additional requirement that the system (3.4) should be affine in $\mathbf{u}$ is imposed (*input-affine realization*).

For example, the differential polynomial (3.2) is realizable and one of its realizations is the predator-prey model (3.1).

The case when there are no inputs was studied by Forsman [43] who proved an elegant algebro-geometric criterion: $p(y) \in \mathbb{K}[y^{(\infty)}]$ is realizable if and only if the algebraic hypersurface defined by p (i.e., regarding the derivatives of y as independent variables) is unirational. Furthermore, if p is realizable, then it can be realized by a model of order $h := \operatorname{ord}_y p$, that is, a model with $|\mathbf{x}| = h$. Since the inequality $|\mathbf{x}| \leqslant h$ holds for any realization, this means that there is always a realization of minimal possible order, in particular it is *locally observable*, which is a desired property in control theory (cf. [119]).

Together with Dmitrii Pavlov (a master student), we have extended the latter result to the case with inputs[3].

Theorem 3.5.2 — D. Pavlov, G. Pogudin [G36, Theorem 3.2]. If a differential polynomial $p(y, \mathbf{u}) \in \mathbb{K}[y^{(\infty)}, \mathbf{u}^{(\infty)}]$ is realizable (resp., input-affine realizable), then it is realizable (resp., input-affine realizable) by a model of order $\operatorname{ord}_y p$. In particular, it admits a locally observable realization.

Using this general theorem, we have proposed the first explicit realization algorithms for systems with inputs for the cases when $\operatorname{ord}_u p = 0$ (modulo an algorithm for checking rationality of a hypersurface) [G36, Algorithm 1] and when $\operatorname{ord} p \leqslant 1$ [G36, Algorithm 1].

[3] In the paper the theorem is stated and proved for $|\mathbf{u}| = 1$ but the same argument works for multiple inputs

3.6 Lifting solutions back: differential Noether normalization

In the classical algebraic geometry, if a prime ideal $I \subset \mathbb{K}[\mathbf{x},\mathbf{y}]$ over an algebraically closed field $\mathbb{K}$ has zero elimination ideal $I \cap \mathbb{K}[\mathbf{y}] = \{0\}$, then the projection of the zero set of I on the $\mathbf{y}$ plane is dominant, that is, surjective outside of a small (Zariski closed) set. An important question is, for a d-dimensional ideal I, find a d-variate subring $\mathbb{K}[\mathbf{z}] \subset \mathbb{K}[\mathbf{x},\mathbf{y}]$ such that the projection of the zero set of I on the $\mathbf{z}$-plane will actually be surjective. This would provide a "parametrization" of the zero set of I by the $\mathbf{z}$-coordinates. The existence of such parametrization is asserted by the celebrated Noether Normalization Lemma.

A differential analogue of this problem is also quite natural. Consider a prime differential ideal $I \subset \mathbb{K}[\mathbf{x}^{(\infty)}]$. Its *differential dimension* d is defined, roughly speaking, as the number of arbitrary functions defining a general solution of the ideal. Than a natural task is to find d coordinates in the ring $\mathbb{K}[\mathbf{x}^{(\infty)}]$ such that *any* assignment of functions to these coordinates can be completed to a solution of I. I have shown [G41] that this is always possible, here is a version of the main result for power series solutions (the paper also gives versions for differentially closed fields and analytic functions):

Theorem 3.6.1 — G. Pogudin [G41, Corollary 5.3]. Consider a prime differential ideal $I \subset \mathbb{K}[\mathbf{x}^{(\infty)}]$ of differential dimension d. Then there exist $a_1, \dots, a_d \in \mathbb{K}[\mathbf{x}^{(\infty)}]$ such that any assignment of elements of $\overline{\mathbb{K}}[[t]]$ to $a_1, \dots, a_d$ can be extended to a solution of I in $\overline{\mathbb{K}}[[t]]$.

The proof is constructive and can be turned into an algorithm.

3.7 Primitive element theorem

Another construction with strong elimination flavor in classical algebra is the primitive element theorem which says that an algebraic finitely generated field extension $F \subset E$ can be generated by a single element. For example, $\mathbb{Q}(\sqrt{2},\sqrt{3})$ can be generated over $\mathbb{Q}$ by $\sqrt{2}+\sqrt{3}$ only. This theorem is a very useful tool for performing computations in algebraic field extensions. One can observe that in the presence of derivation one can generate even transcendental extensions with a single element.

▪ **Example 3.3** Consider an extension $\mathbb{C} \subset \mathbb{C}(x, e^x)$. The field $\mathbb{C}(x, e^x)$ cannot be generated over $\mathbb{C}$ by a single element because the functions x and e^x are algebraically independent. However, if we take $\alpha = x + e^x$, then we will see that

$$x = \alpha - \alpha' + 1 \quad \text{and} \quad e^x = \alpha' - 1,$$

So one can generate the whole extension using only α and its derivatives. ▪

And indeed, Kolchin [68] has established a differential analogue of the primitive element theorem for Δ-fields.

Theorem 3.7.1 — Kolchin [68, §4]. Consider an extension of Δ-fields $F \subset E := F(a_1^{(\infty)}, \dots, a_\ell^{(\infty)})$ (*finitely Δ-generated*) such that every a_i is a solution of a nonzero Δ-polynomial from $F[x^{(\infty)}]$ (Δ-*algebraic*). Assume that there exist elements $b_1, \dots, b_m \in F$ such that the Jacobian matrix $(\partial_i b_j)_{i,j}$ is nonsingular.

Then there exists $a \in E$ such that $E = F(a^{(\infty)})$.

The condition on the Jacobian is imposed for the *ground field* F, not for E. In particular, Theorem 3.7.1 does not tell us anything about the existence of a single generator in Example 3.3. A similar theorem has been proven by Cohn [26, p. 203, Theorem III] for extensions of difference fields under an assumption that σ has infinite order but, again, on the ground field.

For the case of differential fields, Theorem 3.7.1 has been strengthened to require the existence of a nonconstant element only in E in my Ph.D. thesis [95] thus covering Example 3.3. Interestingly, while a single generator in the setup of Theorem 3.7.1 could be always constructed as a linear combination of $a_1, \ldots, a_\ell$, it was not the case if F is a constant field, so a polynomial in $a_1, \ldots, a_\ell$ has to be considered.

All the primitive element theorems mentioned above have been generalized and strengthened in my paper [G39].

Theorem 3.7.2 — G. Pogudin [G39, Theorem 2.1]. Consider an extension of fields $F \subset E$ equipped with derivations $\partial_1, \ldots, \partial_m$ and automorphisms $\sigma_1, \ldots, \sigma_s$ such that the derivations and automorphisms pairwise commute. Assume that

- E is generated over F using the derivations and automorphisms by finitely many elements;
- every $a \in E$ satisfies a difference-PDE (a polynomial relation involving the derivations and automorphisms);
- there exist m elements of E with nonsingular Jacobian, and no nontrivial power product of $\sigma_1, \ldots, \sigma_s$ acts as the identity on E.

Then there exists $a \in E$ such that E is generated over F by a using the derivations and automorphisms.

In the following example, Theorem 3.7.2 implies the existence of a primitive element while none of the prior results would be applicable.

▪ **Example 3.4** Consider $\Delta = \{\partial_z, \partial_\tau\}$. Let $\mathcal{M}(\mathbb{C}, \mathcal{H})$ denote the field of bivariate meromorphic functions on $\mathbb{C} \times \mathcal{H}$ in variables z and τ, where $\mathcal{H} = \{\tau \in \mathbb{C} \mid \operatorname{Im}(\tau) > 0\}$. We consider $\mathcal{M}(\mathbb{C}, \mathcal{H})$ as a Δ-field by letting ∂_z and ∂_τ act as the partial derivatives in z and τ, respectively. Let

$$\theta_1(z, \tau) := -i \sum_{j=-\infty}^{\infty} (-1)^j e^{(j+1/2)^2 \pi i \tau} e^{(2j+1)\pi i z}$$

be one of the Jacobi theta functions. Function $\theta_1(z, \tau)$ satisfies the heat equation [88, p. 433]:

$$\partial_z^2 \theta_1(z, \tau) = 4\pi i \partial_\tau \theta_1(z, \tau).$$

Thus, $\theta_1(z, \tau)$, $\theta_1(2z, \tau)$, and $\theta_1(3z, \tau)$ are Δ-algebraic over $\mathbb{C}$. Therefore, Theorem 3.7.2 applied to the extension

$$F := \mathbb{C} \subset E := \mathbb{C}(\theta_1(z, \tau)^{(\infty)}, \theta_1(2z, \tau)^{(\infty)}, \theta_1(3z, \tau)^{(\infty)})$$

implies that there exists a function $f(z, \tau) \in E$ such that $\theta_1(z, \tau)$, $\theta_1(2z, \tau)$, and $\theta_1(3z, \tau)$ can be written as rational functions in f and its partial derivatives. ▪

4. Structural Parameter Identifiability

Parametric dynamical models (such as parametric ODE systems which we will consider in this chapter) is one of the standard tools used for modeling in the sciences. Once such a model has been designed, the next step is usually to fit the parameter values from the experimental data. At this step, it may happen that, due to the structure of a model, several parameter values may yield *exactly* the same output time series. For example, consider a scalar ODE with two unknown parameters a and b:

$$x'(t) = x(t) + a + b. \tag{4.1}$$

Since the parameters appear only as a sum $a+b$, the values $\widetilde{a} = a+c$ and $\widetilde{b} = b-c$ will produce the same solution and, thus, time series for $x(t)$ regardless of the value of c. In such case, the parameters a and b are called *structurally nonidentifiable*. On the other hand, if the parameter value can be inferred uniquely under the assumption of complete noise-free data, the parameter is called *structurally identifiable*. Thus, structural parameter identifiability is a prerequisite for meaningful parameter estimation. An important advantage of this property is that it can be assessed based on the model only, before data is collected, and the results of the assessment can be used to revise the model or change the experiment design.

One could say that the identifiability issue in (4.1) was artificially created. Let us consider a less obvious yet still simple example.

▪ **Example 4.1** Consider a two-dimensional linear model which describes flows of material from compartment 2 to compartment 1 (at rate b) and from compartment 1 to the environment (at rate a), see Figure 4.1

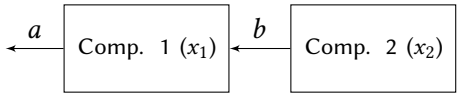
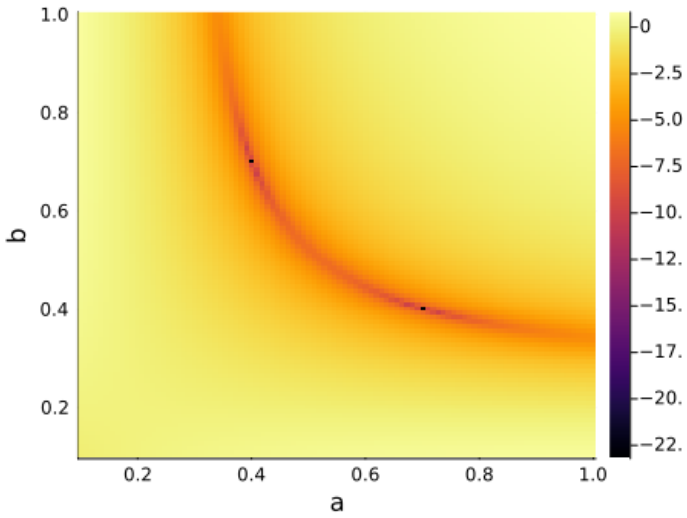


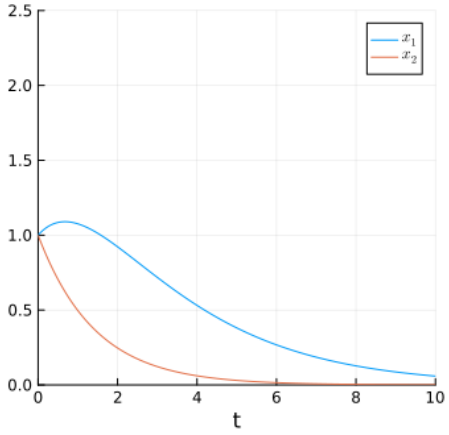
Figure 4.1: Scheme of the compartment model

$$\begin{cases} x_1'(t) &= -ax_1(t) + bx_2(t), \\ x_2'(t) &= -bx_2(t). \end{cases} \tag{4.2}$$

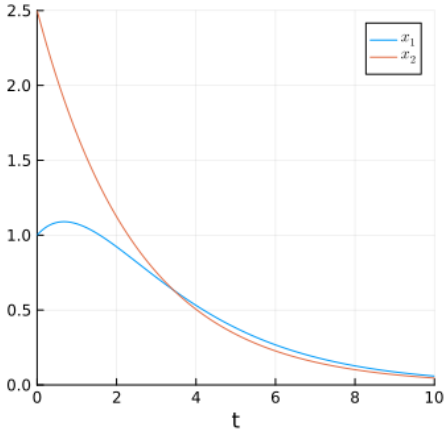
We generate a time series for $x_1(t)$ using the initial conditions $x_1(0) = x_2(0) = 1.0$ and parameter values $a = 0.4$, $b = 0.7$. For various pairs $(a,b) \in [0,1] \times [0,1]$, we compute $x_2(0)$ giving the best fit and plot the logarithm of the mean squared error on a heatmap in Figure 4.2a. While we see that the original pair $(0.4, 0.7)$ gives a perfect fit, there is another pair $(0.7, 0.4)$ which gives a perfect fit as well. Indeed, if we plot the solution for the system for $x_1(0) = 1.0$, $x_2(0) = 2.5$, $a = 0.7$, $b = 0.4$ (Figure 4.2c), we will see that the x_1-coordinate is exactly the same as in the original simulation



(a) Logarithm of the loss function



(b) Solution for $a = 0.4,\ b = 0.7$



(c) Solution for $a = 0.7,\ b = 0.4$

Figure 4.2: Compartment model (4.2): error landscape (a) and alternative parameter values (b, c)

(Figure 4.2b). In general, one can verify that the time series for x_1 produced by the initial conditions $x_1(0) = x_1^*$ and $x_2(0) = x_2^*$ and parameters $a = a^*$ and $b = b^*$ is the same as for

$$x_1(0) = x_1^*, \quad x_2(0) = \frac{b^* - a^*}{a^*} x_1^* + \frac{b^*}{a^*} x_2^*, \quad a = b^*, \quad b = a^*.$$

Note that there is no obvious symmetry between a and b in the original model (4.2). Therefore, if one did not look at a large enough region in the parameter space when searching for the parameter values, one could easily find only one of the two possible values and be convinced that these are the only "true" values. ∎

4.1 Problem statement and the state of the art

Throughout this chapter, we will study a parametric ODE model in the *state-space form* (commonly used in the modeling and control literature):

$$\Sigma = \begin{cases} \mathbf{x}(t)' = \mathbf{f}(\boldsymbol{\mu}, \mathbf{x}(t), \mathbf{u}(t)), \\ \mathbf{y}(t) = \mathbf{g}(\boldsymbol{\mu}, \mathbf{x}(t), \mathbf{u}(t)), \end{cases} \tag{4.3}$$

where

- $\mathbf{x}(t)$ is a vector of state variables defining the internal state of the system;
- $\mathbf{y}(t)$ is a vector of output variables (representing the data collected in an experiment);
- $\mathbf{u}(t)$ is a vector of input variables (external forces/inflows imposed on the system by an experimenter);
- $\boldsymbol{\mu}$ is a vector of scalar parameters to be inferred;
- $\mathbf{f}$ and $\mathbf{g}$ are tuples of functions; we will restrict ourselves to rational functions, that is, elements of $\mathbb{C}(\boldsymbol{\mu}, \mathbf{x}(t), \mathbf{u}(t))$.

In what follows, we will, for the sake of brevity, omit the explicit time dependence for $\mathbf{x}, \mathbf{y}, \mathbf{u}$. One can state the (structural) identifiability problem as follows.

Problem 4.1.1 — Identifiability. Given a function $h(\boldsymbol{\mu}, \mathbf{x})$ in parameters and states, determine if, for a generic trajectory of (4.3), the value of h can be reconstructed uniquely from the time series for inputs $\mathbf{u}$ and outputs $\mathbf{y}$ under the assumption of complete noise-free data and sufficiently exciting inputs.

For the moment, we will use this partially informal problem statement and postpone the discussion of rigorous definitions of identifiability to the next section. Below, we will outline several versions/refinements of this general problem and finish the section with a very brief overview of the history of the problem.

Local and global identifiability

If one requires, as in Problem 4.1.1 that the value of h has to be reconstructable uniquely, then this property is also often referred to as *global identifiability*. There also exists a weaker property of *local identifiability* meaning that the value can be reconstructed uniquely in a small enough neighborhood. For the dynamical models described by rational functions considered in this chapter this is equivalent to saying that the value can be reconstructed up to finitely many options. For example, parameters a and b in Example 4.1 are not globally identifiable but, since the possible values are always only (a, b) and (b, a), they are locally identifiable. On the other hand, in the

model (4.1), the parameters are not identifiable even locally because there are infinitely many possible values for any of them.

Identifiability vs. observability

In the literature, one frequently uses two different notions: identifiability for parameters and *observability* for the state variables. These notions are not completely unrelated since one can always consider a parameter μ as a state variable satisfying the equation $\mu' = 0$, therefore identifiability can always be reduced to observability. In this chapter, we will use word "identifiability" for everything: for parameters, states, and functions of them.

Identifiable functions

Problem 4.1.1 was formulated not only for individual parameters/states but also for functions of them. For example, while a and b in (4.1) are not identifiable, their sum $a+b$ can be expressed as $x'-x$, so can be uniquely estimated from perfect data. Similarly, parameters a and b in Example 4.1 are only locally identifiable since they are defined up to a swap, but their sum $a+b$ and ab are invariant under the swapping and, thus, identifiable. Such *identifiable functions* form a field: the sum, product, and quotient of reconstructable quantities is clearly reconstructable as well. Therefore, instead of asking about identifiability of a particular function as in Problem 4.1.1, one can also ask to *find the field of identifiable functions*, that is, everything what can be reconstructed (see Section 4.4 for further discussion).

Brief history of the problem

The identifiability problem goes back at least to the 1970s [9] and has attracted attention of researchers from different domains. In this section we do not attempt to survey all important results but highlight the ones relevant to the discussion in this chapter. The fundamental observability rank condition due to Hermann and Krener [57] reduced checking *local* identifiability to computing the rank of matrix composed of high-order Lie derivatives giving a good algorithm for medium-size models. Sedoglavic [111] found a way to compute these ranks very efficiently, and his algorithm allows assessing local identifiability of large models and is implemented in many modern software packages.

The problem of assessing *global* identifiability turned out to be more challenging. One of the first approaches was the Taylor series method due to Pohjanpalo [100] which reduced the problem to solving an infinite system of equations. For practical computations, the system had to be heuristically truncated (see Section 4.3). A complete algorithm based on reformulating identifiability as a differential elimination problem and applying the characteristic set method (see Section 3.1) was proposed by Ljung and Glad [81]. Another approach utilizing differential elimination for computing so-called input-output equations goes back the the thesis of Ollivier [89], and was later used in software tools such as DAISY [10] and COMBOS [84]. While these tools allowed practical computation for many important models, their applicability was still limited by the scalability of the underlying elimination algorithms. A more detailed overview of the field and available tools up until the early 2010-s can be found in [25, 85].

4.2 Definitions: via Algebra, Analysis, and Input-output equations

While the identifiability question as stated in Problem 4.1.1 is a natural one to ask, it is not an easy one to formalize. A number of different notions of identifiability can be found in a survey [3]. In this section, we will outline the three main different approaches to defining identifiability and

characterize how they relate to each other. To keep the presentation and notation simple, we will restrict ourselves to the case of polynomial models without inputs, and define identifiability for a single parameter. All the results cited below hold in the general situation as well (for a sufficiently rich class of inputs, see [125] for a discussion of different choices).

It is natural to define identifiability is in terms of the trajectories of the model (4.3). Let us give one such definition adapted from [G18, Definition 2.5].

Definition 4.2.1 — Identifiability: analytic definition. Consider a system in the state-space form

$$\begin{cases} \mathbf{x}' = \mathbf{f}(\boldsymbol{\mu}, \mathbf{x}), \\ \mathbf{y} = \mathbf{g}(\boldsymbol{\mu}, \mathbf{x}), \end{cases} \tag{4.4}$$

and let n, m, and s be the dimensions of $\mathbf{x}$, $\mathbf{y}$, and $\boldsymbol{\mu}$, respectively. Assume that $\mathbf{f}$ and $\mathbf{g}$ are vectors of polynomials. For a point $(\tilde{\mathbf{x}}^*, \tilde{\boldsymbol{\mu}}) \in \mathbb{C}^{n+s}$, by $\mathbf{Y}(\tilde{\mathbf{x}}^*, \tilde{\boldsymbol{\mu}}, t)$ we will denote the germ of the unique analytic solution of (4.4) with the initial condition $\mathbf{x}(0) = \tilde{\mathbf{x}}^*$ and $\boldsymbol{\mu} = \tilde{\boldsymbol{\mu}}$.

A parameter $\mu_1 \in \boldsymbol{\mu}$ is called *globally (resp., locally) identifiable* if there exists a proper Zariski open $\Omega \subset \mathbb{C}^{n+s}$ such that, for every $(\tilde{\mathbf{x}}^*, \tilde{\boldsymbol{\mu}}) \in \Omega$, the number of different μ_1-coordinates in the set

$$\{(\hat{\mathbf{x}}^*, \hat{\boldsymbol{\mu}}) \in \mathbb{C}^{n+s} \mid \mathbf{Y}(\hat{\mathbf{x}}^*, \hat{\boldsymbol{\mu}}, t) = \mathbf{Y}(\tilde{\mathbf{x}}^*, \tilde{\boldsymbol{\mu}}, t)\}$$

is equal to one (resp., is finite).

While Definition 4.2.1 captures the intuitive notion of identifiability, "no other parameter value can produce the same solution", it is not very convenient for algorithmic computation. To this end, one often uses an algebraic definition of identifiability (see, e.g. [31, Section II]) as follows.

Definition 4.2.2 — Identifiability: algebraic definition. In the notation of Definition 4.2.1, μ_1 is said *algebraically identifiable* if there exist polynomials $P_1, P_2 \in \mathbb{C}[\mathbf{y}^{(\infty)}]$ such that P_2 does not vanish identically on all the analytic solutions of (4.4) and, for every analytic solution $\boldsymbol{\mu}^*, \mathbf{x}^*(t), \mathbf{y}^*(t)$ of (4.4) on which P_2 does not vanish, we have

$$\mu_1^* = \frac{P_1(\mathbf{y}^*(t))}{P_2(\mathbf{y}^*(t))}. \tag{4.5}$$

Local algebraic identifiability is defined analogously but μ_1 is required to satisfy an algebraic equation over $\mathbb{C}[\mathbf{y}^{(\infty)}(t)]$, not a linear one as in (4.5).

It is quite plausible that identifiability in the sense of Definition 4.2.2 implies identifiability according to Definition 4.2.1: indeed, (4.5) gives a formula to uniquely reconstruct the value of a parameter from a trajectory. Surprisingly, these definitions are in fact equivalent, meaning that, if a reconstruction of the parameter value is possible in principle, it is possible via the formula (4.5).

Theorem 4.2.1 — H. Hong, A. Ovchinnikov, G. Pogudin, C. Yap [G18, Proposition 3.4]. Global (resp., local) identifiability in the sense of Definition 4.2.1 is equivalent to the global (resp., local) algebraic identifiability in the sense of Definition 4.2.2.

In the light of Theorem 4.2.1, we will refer to identifiability in the sense of Definitions 4.2.1

and 4.2.2 as simply *identifiability*.

Yet another way to define identifiability comes from the idea that all one can really observe is the relations between inputs and outputs of the system, so the coefficients of these relations are exactly the quantities which can be inferred from the input-output data. The following example illustrates the approach.

▪ **Example 4.2** Consider a model

$$x_1' = ax_2, \quad x_2' = -bx_1, \quad y = x_1 + c \tag{4.6}$$

with states x_1, x_2, parameters a, b, c and single output y. By doing differential elimination (see Chapter 3), we find the minimal relation on the output to be

$$y'' + aby - abc = 0.$$

On one hand, one can argue that, given a time series for y, we cannot fit anything which is not a function of the coefficients of the relation, ab and abc. On the other hand, evaluating the equation at two time points t_1 and t_2, we obtain a linear system

$$\begin{pmatrix} y(t_1) & 1 \\ y(t_2) & 1 \end{pmatrix} \begin{pmatrix} ab \\ -abc \end{pmatrix} = -\begin{pmatrix} y''(t_1) \\ y''(t_2) \end{pmatrix}$$

with respect to ab and abc. If *the matrix of this system is nonsingular*, then one can compute the values of ab and abc and, therefore, of c. ▪

One can turn these considerations into a definition. Unlike Definitions 4.2.1 and 4.2.2, this definition makes sense only for functions in parameters, not states (may seem an annoying limitation, but becomes completely natural once viewed from a right angle, see Theorem 4.2.3). To keep the presentation simple, we will consider the single-output case, the results below hold in full generality.

Definition 4.2.3 — Identifiability: input-output approach. In the notation of Definition 4.2.1, assume that $m = 1$. Consider the minimal (comparing first w.r.t. the order and then w.r.t. the degree) polynomial differential equation $P(y, y', \dots, y^{(h)}) = 0$ with coefficients in $\mathbb{C}(\boldsymbol{\mu})$ satisfied by y for every solution of (4.4).

Assume that P is normalized so that at least one of its coefficients is equal to one, and denote the remaining coefficients by $c_1(\boldsymbol{\mu}), \dots, c_\ell(\boldsymbol{\mu})$. Then a parameter μ_1 is *globally (resp., locally) input-output identifiable (IO-identifiable)* if $\mu_1 \in \mathbb{C}(c_1, \dots, c_\ell)$ (resp., μ_1 is algebraic over $\mathbb{C}(c_1, \dots, c_\ell)$).

Definition 4.2.3 has two attractive features: it is well-suited for computation [89] (used in software DAISY [10] and COMBOS [84]), and not only it allows to assess identifiability, but it also gives a direct way to find the generators of the *whole field* of identifiable functions.

However, in contrast to Theorem 4.2.1, we have found an example [G18, Example 2.14], in which a parameter is IO-identifiable but is not identifiable. We further proved, together with A. Ovchinnikov and P. Thompson, that identifiability implies IO-identifiability [G34, Theorem 1]. We can thus summarize the established relations between the definitions as follows:

$$\begin{array}{ccccc} \text{Definition 4.2.1} & & \text{Definition 4.2.2} & & \text{Definition 4.2.3} \\ \textit{analytic} & \Longleftrightarrow & \textit{algebraic} & \Longrightarrow & \textit{input-output} \end{array}$$

The attractive features of IO-identifiability outlined above make it natural to establish cases in which it actually agrees with identifiability. One result in this direction is the following theorem.

Theorem 4.2.2 — A. Ovchinnikov, G. Pogudin, P. Thompson [G33, Theorem 1]. For single-output linear models, identifiability is equivalent to IO-identifiability.

We have established the equivalence also to a class of multi-output linear compartment models (for details, see [G33, Theorems 2 and 3]). Identifiability of such models was further explored by Natali Gogishvili (undergraduate student) in a project under my supervision [46].

We have found a new characterization of IO-identifiability with A. Ovchinnikov, A. Pillay, and T. Scanlon in [G30] which nicely explains the aforementioned discrepancies between the definitions. We will say that a parameter μ_1 in (4.4) is *multi-experiment identifiable* if its value can be inferred from a finite number of generic trajectories of (4.4) (as opposed to a single trajectory in Definition 4.2.1). Such multi-experimental setup is quite natural for some application domains [122].

Theorem 4.2.3 — A. Ovchinnikov, A. Pillay, G. Pogudin, T. Scanlon [G30, Theorem 19]. Multi-experiment identifiability is equivalent to the IO-identifiability.

In a subsequent paper [G31], we developed an efficient algorithm for computing an upper bound for the sufficient number of experiments which overestimates the minimal number by at most one. Interestingly, this result, in addition to differential algebra, also used model theory in the sense of mathematical logic [G31, Section 7].

4.3 Taylor series revisited: `SIAN` algorithm and software

When it comes to assessing (global) identifiability in practice, the algorithms behind the existing software could be roughly subdivided into two classes:

1. *Based on input-output equations.* As described in Example 4.2, this approach consists of first eliminating the state variables and then considering if the parameters of interest can be expressed in terms of the coefficients of the obtained input-output relations.
2. *Based on the Taylor series.* The idea is to take the coefficients of the truncated power series for the output variables up to certain order, write then in terms of states and parameters, and then check if the parameters of interest can be expressed using these quantities (see Example 4.3 below).

The former approach has already been illustrated on a toy Example 4.2, let us show the latter using the same system.

▪ **Example 4.3 — Taylor series method.** We will again consider the model (4.6). Start with expressing the derivatives of y at zero (that is, the coefficients of the Taylor series for y) in terms of the parameters and initial conditions:

$$y(0) = x_1(0) + c, \qquad y'(0) = x_1'(0) = ax_2(0),$$
$$y''(0) = ax_2'(0) = -abx_1(0), \qquad y^{(3)}(0) = -abx_1'(0) = -a^2bx_2(0).$$

Since y is assumed to be observed, the above quantities are known. Therefore, we can conclude

that ab and c are globally identifiable because they can be written as:

$$ab = \frac{-y^{(3)}(0)}{y'(0)} \quad \text{and} \quad c = y(0) + \frac{y''(0)}{ab}.$$

In other words, we solve the identifiability problem for ab and c by solving the rational field membership problem for ab and c and field generated by $y(0), \dots, y^{(3)}(0)$. ∎

The method goes back to the work by Pohjanpalo [100] in the 1970s. However, the result of the computation clearly depends on the number of derivatives considered. In order to turn the approach into an algorithm, one has to find a way to choose this order so that the correctness is guaranteed.

The survey [25] from 2011 mentions three software tool capable of assessing global identifiability. Two of them, DAISY [10] and COMBOS [84], use the approach via the input-output equations but do not verify if IO-identifiability coincides with identifiability for the model in question and, thus, can return an incorrect (according to Definitions 4.2.1 and 4.2.2) result [G18, Example 2.14]. The remaining one, GenSSI 2.0 [80], followed the Taylor series approach with a heuristic used for choosing the truncation order can lead to incorrect result as well [G18, Example 2.16].

Together with H. Hong, A. Ovchinnikov, and C. Yap in [G18] we have designed a complete and rigorous global identifiability algorithm based on the Taylor series approach. Our implementation compared favorably to the state-of-the-art in terms of performance: we were able to analyze models which were out of reach before [G18, Table 6.1]. The key ingredients of the algorithm are:

- an effective characterization of the truncation order sufficient to retain all the identifiability information [G18, Theorem 3.16];
- a Monte Carlo randomized algorithms to check the corresponding field membership problem together with an explicit error bound [G18, Theorem 4.2].

The implementation based on this paper has been turned into a Maple-based software tool SIAN [G17].

The main bottleneck for SIAN is a Gröbner basis computation (used to check the field membership). Together with M. Bessonov, I. Ilmer, T. Konstantinova, A. Ovchinnikov, and P. Soto, we have recently proposed [G6] a monomial ordering defined in terms of the structure of the input ODE model which allowed to speed up the computation even further.

4.4 Computing identifiable functions

Once some parameters of interest turn out to be nonidentifiable, the next natural question is *what is identifiable then*? For instance, in Example 4.3 we saw that, while a and b are not identifiable, the function ab is identifiable. Furthermore, one can check that identifiable functions form a subfield in $\mathbb{C}(\boldsymbol{\mu})$ which will be called *the field of identifiable functions*.

Problem 4.4.1 — Identifiable functions computation. Given a model (4.3), find a generating set $h_1, \dots, h_\ell \in \mathbb{C}(\boldsymbol{\mu})$ of the field of identifiable functions.

It has been known [G34, Corollary 1] that generators of the field of IO-identifiable functions can be read off from the input-output equations. However, IO-identifiable functions do not always coincide with the identifiable ones. This challenge was addressed in our work with A. Ovchinnikov, A. Pillay, and T. Scanlon [G30]: we designed and implemented an algorithm solving Problem 4.4.1 even if identifiability does not agree with IO-identifiability for a model in question. The approach

was to start with input-output equations of the model, and find all additional relations between the same differential monomials which appear if one allows taking the coefficients from the field containing both $\mathbb{C}(\boldsymbol{\mu})$ and the first integrals of the model. We show [G30, Theorem 11] that, roughly speaking, the elements of $\mathbb{C}(\boldsymbol{\mu})$ appearing in these extra relations will generate the whole field of identifiable functions. One of the byproducts of this work was strengthening existing algorithms for subfield intersection [G30, Appendix B].

In order to make these developments usable by other researchers, together with I. Ilmer and A. Ovchinnikov, we have combined (in a synergistic way) the implementation of the algorithms from [G30] with SIAN [G17] into a single web-based application [G20]. Both SIAN and code from [G30] were written in Maple which is a proprietary software, and the format of web-application turned out to be ideal to make the software accessible to the researchers without a Maple license and was appreciated by the users. The software received the Best Software Presentation award at the ISSAC conference in 2021 [G19].

4.5 Faster IO-equations: `StructuralIdentifiability.jl` library

While the algorithms described in Section 4.4 provide a full solution of the Problem 4.4.1 in theory, the practical tools derived from them such as the web-based Structural Identifiability Toolbox were applicable only to models of moderate size. The main computational bottleneck for these implementations was computing the input-output equations for the model, that is, equations involving only inputs, outputs, and parameters (see Example 4.2). Since this computation is a special case of the *differential elimination problem* (see Chapter 3), classical Rosenfeld-Gröbner algorithm was used. However, being applicable to general nonlinear PDE systems, this algorithm was not taking advantage of the particular shape of the equations in the state-space from (4.3), and many important models remained out of reach.

In project started together with R. Dong (master student), C. Goodbrake, and H. Harrington, we aimed at reworking the approach based on input-output equations with efficiency and rigor in mind. Our joint paper [G12] describing the result of this work contained the key ingredients for a such a new algorithm:

- *Elimination.* We have developed a more efficient differential elimination algorithm tailored to the models in the state-space form (3.4), it is described in more detail in Section 3.4.
- *"Sanity check" for input-output equations.* Before the coefficients of an input-output equation can be used to assess identifiability, one has to check whether the IO-identifiability coincides with identifiability for the model at hand (see Section 4.2). This can be done by checking that a certain Wronskian determinant is nonzero [G34, Lemma 1]. However, the new elimination algorithm was capable to compute really large input-output equations, for which the way this Wronskian computation was done before (e.g., in [G20]) became completely impractical. We have proposed a new approach [G12, Section 5.3] which proved to be efficient for all the models we have encountered since then. Thanks to this new algorithm we could actually check *how often identifiability and IO-identifiability coincide* on a wide range of models from the literature, and have found out that they coincide in a vast majority of cases [G12, Table 3].
- *Efficient field membership testing*. Once the input-output equations have passed the "sanity check", the identifiability of a function of parameters $h(\boldsymbol{\mu})$ is verified by checking whether $h(\boldsymbol{\mu})$ belongs to the field generated by the coefficients of the input-output equations. In the earlier works, the classical algorithm [87] was employed which required computing Gröbner

Model	DAISY [10]	SIAN [G17]	New algorithm
SIWR model [G12, Example 6]	OOM	> 5 h.	18 s.
SIWR model - 2 [G12, Example 6]	OOM	213 s.	0.7 s.
Pharmacokinetics [G12, Example 7]	> 5 h.	> 5 h.	406 s.
MAPK pathway - 1 [G12, Example 8]	OOM	31 s.	39.5 s.
MAPK pathway - 2 [G12, Example 8]	> 5 h.	> 5 h.	58 s.
SEAIJRC model [G12, Example 9]	OOM	> 5 h.	131.3 s.
Akt pathway [G12, Example 11]	182 s.	28 s.	5 s.

Table 4.1: Comparison of the new algorithm [G12] with other software (OOM: "out of memory")

basis with coefficients in a rational function field. Again, for larger models we targeted, this approach was impractical. In order to address this challenge, we have established theory to allow randomized reduction to much more efficient Gröbner basis computation over $\mathbb{Q}$ [G12, Theorem 3.2].

Combining these ingredients, we have designed and developed an algorithm which could assess identifiability for models which were out of reach before on a laptop — Table 4.1 reproduces a fragment of [G12, Table 4].

Our implementation of the resulting algorithm was a basis for the first release in 2021 of the new identifiability software tool, StructuralIdentifiability[1] package written in Julia language. In the subsequent years the package has been actively developed: 38 releases made, 100+ stars on github received. The package was also included into the SciML (Scientific Machine Learning[2]) ecosystem. The software greatly benefited from new contributors including students. Now it can assess local identifiability for discrete-time system, compute identifiable functions of states and parameters, and propose reparametrizations of models (see Section 4.6).

4.6 Identifiable reparametrizations

This section describes current work in progress joint with Alexander Demin (student) and Christopher Rackauckas. The implementations of the algorithms presented below are already available in `StructuralIdentifiability.jl` together with tutorials[3]. The papers are in preparation.

Fast IO-equations computation implemented in `StructuralIdentifiability.jl` allows not only to assess identifiability of some given functions of parameters but also to find a generating set for *all* identifiable functions (i.e. solve Problem 4.4.1) since, in most cases, the coefficients of the IO-equations provide such a generating set. However, this solution of Problem 4.4.1 is not completely satisfactory: the coefficients may be too large to be analyzed by a human researcher.

We developed a new algorithm for computing a convenient generating set for the field of identifiable functions. Compared to our earlier work [G30], the algorithm is more efficient and performs more nuanced simplification of the resulting generating set. But the key novelty is the fact that it can also compute the generating set of all the identifiable functions in parameters and *states* (also called observable functions). This is achieved by combining the coefficients of the

[1] `https://github.com/SciML/StructuralIdentifiability.jl`

[2] `https://sciml.ai`

[3] `https://docs.sciml.ai/StructuralIdentifiability/stable/tutorials/identifiable_functions/` and `https://docs.sciml.ai/StructuralIdentifiability/stable/tutorials/reparametrization/`

IO-equations with few terms of the Taylor expansions of the outputs similarly to how it is done in SIAN (see Section 4.3) but the number of terms is much smaller, and this makes the computation efficient.

▪ **Example 4.4 — Computing identifiable functions.** The running example in this section will be the following SEUIR epidemiological model [109, Eq. (5.2)]:

$$\begin{cases} S' = -\frac{\beta}{N}(I+U)S, \\ E' = \frac{\beta}{N}(I+U)S - \gamma E, \\ U' = (1-\alpha)\gamma E - \delta U, \\ I' = \alpha\gamma E - \delta I, \\ y = I, \end{cases}$$

where $\alpha, \beta, \gamma, \delta, N$, are parameters, S, E, I, U are states, and the number of infected individuals I is the output of the model. The input-output equation has more than a hundred of coefficients including not very simple ones such as $\alpha^2\delta^7N^2 - 3\alpha^2\gamma\delta^6N^2 - 9\alpha^2\gamma^2\delta^5N^2 - 4\alpha^2\gamma^3\delta^4N^2$. Applying our simplification algorithm, we find that the whole field of identifiable functions in parameters is generated by $\gamma, \delta, \frac{\beta}{\alpha N}$.

Furthermore, we can now bring the states into the picture: the field of all identifiable function in parameters and states is generated by

$$\gamma,\ \delta,\ \frac{\beta}{\alpha N},\ I,\ \alpha E,\ \alpha S,\ \alpha(I+U). \tag{4.7}$$

▪

The ability to compute identifiable functions in the parameters and states is valuable not only as a source of additional information about the model. Simplified generators of the field of identifiable functions such as (4.7) are natural *new coordinates for an identifiable reparametrization* of the model. Indeed, a derivative of any of the generators is identifiable again and, thus, must be expressible in terms of them yielding a reparametrized model.

▪ **Example 4.5 — Reparametrizing the model.** We give names to the new coordinates from (4.7):

$$\widetilde{S} := \alpha S,\ \widetilde{E} := \alpha E,\ \widetilde{U} := \alpha(I+U),\ \widetilde{\beta} := \frac{\beta}{\alpha N}.$$

Then these quantities together with already identifiable I, γ, δ yield the following model

$$\begin{cases} \widetilde{S}' = -\widetilde{\beta}\widetilde{S}\widetilde{U}, \\ \widetilde{E}' = \widetilde{\beta}\widetilde{S}\widetilde{U} - \gamma\widetilde{E}, \\ \widetilde{U}' = -\delta\widetilde{U} + \gamma\widetilde{E}, \\ I' = \gamma\widetilde{E} - \delta I, \\ y = I. \end{cases}$$

In this model, all the states and parameters are globally identifiable. Both new coordinates and the new model were computed using the new functionality of `StructuralIdentifiability.jl`. Interestingly, the first three equations form themselves a so-called SEIR model. This has also been pointed out in the original paper [109, Eq. (5.8)] but the last equation connecting this SEIR-submodel to the original output found by our software has not been discovered. ▪

5. Exact Model Reduction

Realistic dynamical models used in life sciences can involve tens, hundreds, or even thousands of equations. Researchers working with such large-scale models face two challenges:

- *Complexity:* how to compute with such models efficiently?
- *Interpretability:* how to derive mechanistic insights from models of this size?

One way to address these challenges is to use *model reduction* algorithms that replace a model with a simpler one while preserving, at least approximately, some of the features of the original model. Most popular methods solve this task by performing an *approximate model reduction* (e.g., [4, 12]), that is, finding a simpler model which approximately agrees with the original one. These approaches are applicable in a wide range of contexts and can considerably reduce the model size. On the other hand, approximation errors are introduced and the structure of a model can be destroyed thus making the reduced model harder to interpret.

A complementary approach is to perform *exact model reduction*. Let us illustrate it with the following toy three-dimensional model with unknown functions x_1, x_2, x_3:

$$\begin{cases} x_1' = x_1 + (x_2 + x_3)^2, \\ x_2' = x_3, \\ x_3' = x_2 - x_1. \end{cases} \tag{5.1}$$

Consider quantities $y_1 = x_1$ and $y_2 = x_2 + x_3$ and study their dynamics:

$$y_1' = x_1' = x_1 + (x_2 + x_3)^2 = y_1 + y_2^2, \quad \text{and} \quad y_2' = x_2' + x_3' = y_2 - y_1.$$

We see that y_1 and y_2 satisfy themselves a system of equations of lower dimension:

$$\begin{cases} y_1' = y_1 + y_2^2, \\ y_2' = y_2 - y_1. \end{cases} \tag{5.2}$$

Then model (5.2) is a *reduction* of (5.1), and it is *exact* in the sense that, for every solution of (5.1), the corresponding quantities y_1 and y_2 satisfy (5.2) exactly.

This chapter presents algorithms and software for the problem of finding exact model reductions. Most of the chapter is devoted to *linear reductions* (also called *lumpings*) in which the new variables are linear combinations of the original ones. The last section (Section 5.5) will discuss ongoing work on nonlinear reductions and connection of this problem with the structural identifiability problem from Chapter 4.

5.1 Linear reductions: problem statement and prior results

Definition 5.1.1 — Lumping. Consider an ODE system

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}) \tag{5.3}$$

in the variables $\mathbf{x} = (x_1, \dots, x_n)$ with rational right-hand side, that is, $\mathbf{f} = (f_1, \dots, f_n)$ and $f_1, \dots, f_n \in \mathbb{C}(\mathbf{x})$. We say that a linear transformation $\mathbf{y} = \mathbf{x}L$ with $\mathbf{y} = (y_1, \dots, y_m)$, $L \in \mathbb{C}^{n\times m}$, and $\operatorname{rank} L = m$ is *a lumping of* (5.3) if there exists $\mathbf{g} = (g_1, \dots, g_m)$ with $g_1, \dots, g_m \in \mathbb{C}(\mathbf{y})$ such that

$$\mathbf{y}' = \mathbf{g}(\mathbf{y})$$

for every solution $\mathbf{x}$ of (5.3). In other words, the linear forms $\mathbf{y}$ satisfy themselves a self-contained ODE system. The number m is called *the dimension* of the lumping, and the entries of $\mathbf{y}$ are referred to as *macro-variables*.

▪ **Example 5.1** The reduction (5.2) of the system (5.1) can be presented in the framework of Definition 5.1.1 as follows:

$$y_1 = x_1,\ \ y_2 = x_2 + x_3 \implies \begin{pmatrix} y_1 & y_2 \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \implies L = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

▪ **Example 5.2 — Conservation laws as lumpings.** Assume that the system (5.3) has a linear conservation law, that is, there exist $\mathbf{c} \in \mathbb{C}^n$ such that $c_1x_1 + \ldots + c_nx_n = const$ along any trajectory of (5.3). Then such a conservation law yields a lumping of order one defined by $y = c_1x_1 + \ldots + c_nx_n$ and equation $y' = 0$. ▪

Definition 5.1.2 — Constrained lumping. In the notation of Definition 5.1.1, assume that one is additionally given a vector $\mathbf{x}_{\text{obs}}$ of linear forms in $\mathbf{x}$. Then a lumping $\mathbf{y} = \mathbf{x}L$ is called a *constrained lumping with respect to* $\mathbf{x}_{\text{obs}}$ if the entries of $\mathbf{x}_{\text{obs}}$ can be expressed as linear combinations of the entries of $\mathbf{y}$.

In other words, constrained linear lumping is a reduction required to preserve a set of observable variables $\mathbf{x}_{\text{obs}}$ of interest. For example, the lumping from Example 5.1 is a constrained linear lumping with respect to one-dimensional observable vectors $\mathbf{x}_{\text{obs}} = (x_1)$ and $\mathbf{x}_{\text{obs}} = (x_1 + x_2 + x_3)$.

The previous research on the topic has been mostly focused on the case when the macro-variables $\mathbf{y}$ are sums of subsets of $\mathbf{x}$ or, in other words, when the matrix L in Definition 5.1.1 is a 0/1-matrix. For this setting, powerful methods have been developed, see e.g. [20, 21, 39, 40]. These include, in particular, a software ERODE [19] which can very efficiently find the smallest lumping among the ones coming from a subdivision of the set $\mathbf{x}$ (that is, each macro-variable is the sum of one of the sets in the subdivision).

5.2 Constrained linear reductions: CLUE package

As described in the previous section, the prior algorithms for finding lumpings were restricted to the case of 0/1 coefficients which could be reformulated as a combinatorial optimization problem in finite (but huge) search space. Allowing the coefficients to be arbitrary numbers makes the

search space infinite and requires tools from algebra. We start with presenting a reduction of the problem of computing lumpings to a linear algebra problem.

We consider a polynomial dynamical system

$$\mathbf{x}' = \mathbf{p}(\mathbf{x}), \quad \text{where } \mathbf{p} \in \mathbb{C}[\mathbf{x}]. \tag{5.4}$$

We denote by $J(\mathbf{x})$ the Jacobian matrix of $\mathbf{p}$ with respect to $\mathbf{x}$, that is, the matrix with the columns being $\nabla(p_1), \ldots, \nabla(p_n)$. Since the system (5.4) is defined by polynomials, $J(\mathbf{x})$ can be viewed as a polynomial in $\mathbf{x}$ with matrix coefficients. Let $m_1(\mathbf{x}), \ldots, m_N(\mathbf{x})$ be the monomials appearing in $J(\mathbf{x})$, then we can write

$$J(\mathbf{x}) = \sum_{i=1}^{N} J_i m_i(\mathbf{x}), \quad \text{where } J_1, \ldots, J_N \in \mathbb{C}^{n \times n}.$$

The following lemma is the key tool for computing lumpings with arbitrary coefficients.

Proposition 5.2.1 — A. Ovchinnikov, I.C. Pérez Verona, G. Pogudin, M. Tribastone [G29, Section S.II]. A matrix $L \in \mathbb{C}^{n \times m}$ is a lumping of the system (5.4) if and only if the column space of L is invariant under $J_1, \ldots, J_N$.

While the importance of the Jacobian in the context of exact linear reduction has been recognized in the pioneering works by Li and Rabitz [77] and has been used to derive some necessary conditions, the "if and only if" criterion from Proposition 5.2.1 has not been formulated until our paper [G29].

Proposition 4.4.1 reduces the search of a constrained lumping of smallest order to the following problem in linear algebra.

Input A list of matrices $M_1, \ldots, M_s \in \mathbb{C}^{n \times n}$ and a list of vectors $v_1, \ldots, v_r \in \mathbb{C}^n$.

Output A basis of the smallest (with respect to inclusion) subspace $V \subset \mathbb{C}^n$ containing $v_1, \ldots, v_r$ and invariant under $M_1, \ldots, M_s$.

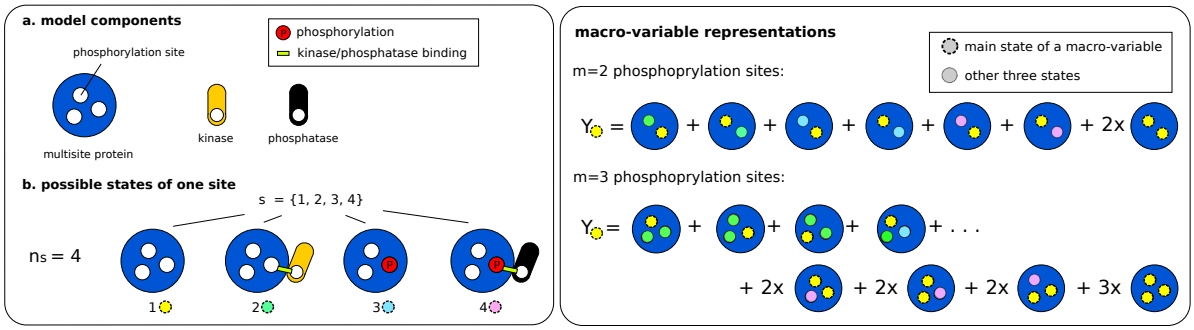
In theory, this problem is easy: one starts with $v_1, \ldots, v_r$ and adds products of already chosen vectors with M_i's until the generated subspace stabilizes (summarized in [G29, Algorithm 1]). Making such an algorithm to scale to dimensions of hundreds and thousands required carefully reorganizing this computation, utilizing modular computation, and taking advantage of the sparsity of the input matrices. We have implemented a resulting algorithm in a python package CLUE [G29]. Since it allows a larger class of reductions, we were able to produce reductions of lower dimension than the state-of-the-art methods while keeping the observed variables (see [G29, Table 1]). We describe a specific example in detail in the next section.

Proposition 5.2.1 and, thus, the CLUE algorithm are restricted to ODE systems with polynomial right-hand side. However, many models coming from applications (for example, describing reaction networks with enzymatic reactions) are defined by rational functions. Together with A. Jimenez-Pastor (postdoc) and J. Jacob (undergraduate student) in [G21], we have generalized Proposition 5.2.1 to the case of rational dynamics [G21, Lemma 6] and used the more general version to propose an efficient randomized algorithm for computing optimal constrained lumping for rational ODE systems. This algorithm was integrated into the CLUE package as well.

5.3 Showcasing CLUE and quest for interpretability

As an example of lumping for a system from the literature, we consider a model of multisite phosphorylation [115]. It describes a protein with m identical and independent binding sites that

simultaneously undergo phosphorylation and dephosphorylation. Each binding site can be either phosphorylated or unphosphorylated and bound or unbound to a kinase. This yields four different states for a site as on Figure 5.1a. The chemical species in the model are kinase, phosphatase, and all possible protein configurations giving $4^m + 2$ species in total. Possible reactions between these species include phosphorylation/dephosphorylation and binding/unbinding of kinase. The evolution of this system is described by an ODE system with the variables being the concentrations of species and the dynamics being defined by the standard mass-action kinetics [38].



(a) Setup of the model (for $m = 3$). (b) New interpretable macro-variables for $m = 2, 3$.

Figure 5.1: Multisite phosphorylation model

Reductions discovered by ERODE [19] (for $m = 2, \ldots, 8$) consisted in replacing the concentrations of protein configurations by the sums of the concentrations of configurations differing by a permutation of the sites. Therefore, the number of macro-variables is equal to $\binom{m+3}{3} + 2$. In contrast, the analysis performed by CLUE [G29] with the observable being the amount of kinase always results in just six macro-variables (independently on m!). We provide the corresponding numbers in Table 5.1 below. Note that each of these reductions (even when the original dimension was 16 thousands) was computed in less than a minute on a laptop.

m	Original dim	ERODE dim	CLUE dim	m	Original dim	ERODE dim	CLUE dim
2	18	12	6	5	1026	58	6
3	66	22	6	6	4098	86	6
4	258	37	6	7	16386	122	6

Table 5.1: The dimensions of the original model of multisite phosphorylation (for different m's) and the dimensions of the reductions discovered by ERODE and CLUE

While the reductions obtained by CLUE are very attractive in terms of dimension, not all computed macro-variables had meaningful interpretations: two of them were the concentrations of kinase and phosphatase, and the other four were linear combinations of protein configurations. Among the latter, one remained not understood during our work on the original CLUE paper. Similar interpretability challenges occurred in other examples we have considered in [G29].

This issue has been addressed in our work [G44] with X. Zhang (undergraduate student). Our idea was that, since any linear lumping is defined up to an invertible linear change of coordinates in the reduced model, one should be able to improve the interpretability of a reduction by a postprocessing which would choose, for a given reduction, a "better basis". We used convex geometry to design an algorithm finding a new set of macro-variables with all the coefficients

being nonnegative and subject to certain minimality condition. This notion of "better basis" proved to be efficient: it produced interpretable reductions for all three case studies from [G29] in which such reductions could not be obtained before.

For the multisite phosphorylation model, the new set of macro-variables produced by our algorithm again included the concentrations of kinase and phosphatase, and each of the four other macro-variables involving the protein configurations corresponded to a state of a site (e.g., unbounded and unphosphorylated), and each protein configuration appeared with a coefficient equal to the number of sites in it with this state. Examples of these new macro-variables are given on Figure 5.1b for $m = 2$ and $m = 3$. This admits a natural interpretation: the constructed reduction replaces the concentration of the protein configurations with the "concentrations" of each of the four states of the sites.

5.4 Computing hierarchies of reductions

The algorithms used in the `CLUE` described in the two previous sections were designed to compute *constrained* linear reductions (see Definition 5.1.2) meaning that the input consisted of an ODE model and a list of observables to be present in the reduction. While in some cases such observables are naturally available, in many situations they are not, so an *educated guess* is required.

A natural way to state the problem in the unconstrained case is to find an exact linear reduction of the smallest possible dimension. However, such a reduction may be too coarse: if a model has a linear first integral $\ell(\mathbf{x}) = const$, then a macro-variable $y := \ell(\mathbf{x})$ defines a one-dimensional reduced system $y' = 0$ (see Example 5.2) which carries too little information about the original system's dynamics.

In our work [G11] with A. Demin and E. Demitraki (both undergraduate students), we came up with a more flexible way of stating the exact linear reduction problem in the unconstrained case. To this end, we define a *chain of lumpings* [G11, Definition 2] as a sequence of lumpings where the reduced system of the i-th lumping is the starting system for the $i+1$-st one, and the first lumping is a lumping of the input model. We will call such a chain *maximal* if it has the largest possible length. Given a chain of lumpings, user can move along it and choose an optimal trade-off between the size of the reduced model and the amount of detail preserved.

We give an algorithm to compute a maximal chain of reductions and implement it in the Julia package `ExactODEReduction`[1]. Compared to the constrained lumpings in `CLUE`, the algorithm has much more powerful algebraic tools behind as it builds upon the structure theory of finite-dimensional algebras. These algebraic concept enter the scene as follows: in terms of Proposition 5.2.1, if $\mathcal{A}$ is an algebra generated by $J_1, \dots, J_N$, then the lumpings are in a bijective correspondence of the invariant subspaces (i.e., submodules) with respect to this algebra. Furthermore, the maximal chains of lumpings correspond to the Jordan-Hölder filtrations of $\mathbb{C}^n$. This interpretation allowed us to use numerous tools from the representation theory of such algebras, and our implementation efficiently works with the models of dimension around one hundred.

In addition to detailed case studies reported in the paper [G11, Section 5], we have evaluated the algorithm on models of dimension not exceeding 133 from the BioModels database [82], the average length of the produced chain of reductions and average runtime on a laptop are reported in Table 5.2. One can observe that there is typically a nontrivial number of reductions providing a user with a "menu" within a couple of minutes.

[1] `https://github.com/x3042/ExactODEReduction.jl`

Dimensions	Chain length	Runtime	Dimensions	Chain length	Runtime
2 - 9	1.39	0.6 s	40 - 59	6.08	4.58 s
10 - 19	2.61	0.21 s	60 - 79	6.95	34.57 s
20 - 29	2.13	0.44 s	80 - 99	7.09	96.38 s
30 - 39	2.71	1.74 s	100 - 133	21.5	202.52 s

Table 5.2: Lengths of chains of reductions and runtimes for models from BioModels database [G11, Table 1]

5.5 Nonlinear reductions: the other side of identifiable reparametrizations

Compared to the linear version discussed before and well-understood reductions by scaling transformations [62, 75], the general problem of finding exact reductions is much more challenging and, thus, much less studied. Interestingly, this problem turn out to be closely connected to the structural identifiability problem from Chapter 4 as we will explain below. As in the linear case, one can consider constrained (observables to be preserved provided) and unconstrained settings.

In the *constrained* case we start with an ODE system $\mathbf{x}' = \mathbf{f}(\mathbf{x})$ and a list of observed quantities $\mathbf{y} = \mathbf{g}(\mathbf{x})$, and want to produce new variables $\mathbf{z} = \mathbf{h}(\mathbf{x})$ which satisfy themselves an ODE system of lower dimension, and $\mathbf{y}$ can be expressed in terms of $\mathbf{z}$. This process is summarized on Figure 5.2.

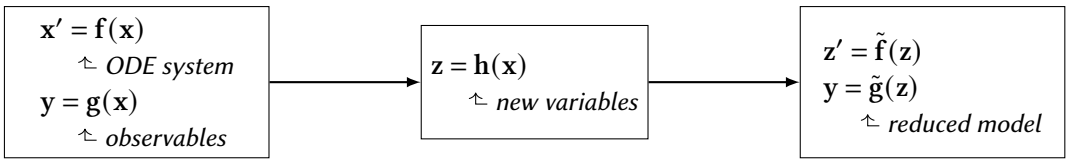


Figure 5.2: Constrained exact nonlinear reduction

The workflow on the Figure 5.2 is essentially the same as has been employed for identifiable reparametrizations in Section 4.6. Furthermore, by definition, any identifiable functions of states are expressible via the outputs and their derivatives and, thus, must be expressible in terms of the new state variables $\mathbf{z}$. This means that the minimal constrained nonlinear reduction we can hope for is a reduction with the states being generators of the field of identifiable functions, and this is exactly what we constructed in Section 4.6.

The idea of the connection between the exact reduction and identifiability was further developed in the *unconstrained* context in our work with A. Jimenez-Pastor [G22]. In this setting, we are not given any observables, so we can choose some observables and then use the methodology for the constrained case. The choice of observables is crucial here: the above discussion implies that, in order to make the reduction nontrivial, one should choose observables for which the model is nonidentifiable. In other words, unconstrained exact reduction can be viewed as an *adversarial version of the identifiability problem*. We used an ansatz to search for such "nonidentifying" observables. The observability rank condition allows expressing the desired lack of identifiability as a polynomial relation on the ansatz coefficients. Let us illustrate the approach using an example from [G22].

▪ **Example 5.3 — Exact nonlinear reduction [G22, Example 4].** The following four-dimensional model with two scalar parameters k and K can be obtained using `CLUE` from a 227-dimensional

model described in [14]:

$$\begin{cases} x_1' &= -2Kx_1x_2 + kx_3, \\ x_2' &= -2Kx_1x_2 - kx_2x_3 + kx_3 + 2kx_4, \\ x_3' &= 2Kx_1x_2 - Kx_2x_3 - kx_3 + 2kx_4, \\ x_4' &= Kx_2x_3 - 2kx_4. \end{cases} \tag{5.5}$$

If one takes a linear ansatz for an output $y = a_1x_1 + \ldots + a_4x_4$, then computation shows that the model will be nonidentifiable with respect to this output if and only if $a_1 + a_4 = 2a_3$. We will set $y = x_2$.

One way to compute the actual reduction is described in [G22], but we will apply directly the reparametrization algorithm from Section 4.6. We obtain

$$\begin{cases} z_1' = -2z_1z_2 + c_2, \\ z_2' = -c_1z_1z_2 + \frac{1}{2}c_1c_2, \\ y = z_1, \end{cases} \qquad \text{where} \begin{cases} z_1 = x_2, \\ z_2 = Kx_1 + \frac{K}{2}x_3 + \frac{k}{2}, \\ c_1 = K, \\ c_2 = kx_2 + kx_3 + 2kx_4. \end{cases}$$

Note that c_2 is in fact constant since it is a first integral of the model. Since the reduced model has a first integral $c_1z_1 - 2z_2 = c_3$, the number of states can be brought further down to one:

$$z_1' = z_1(c_1z_1 - c_3) + c_2, \quad y = z_1.$$

This is now a Ricatti equation, and it can be solved in a closed form. Recall that the whole reduction process has started with a *227-dimensional* ODE model! ∎

6. Quadratization

In Section 4.6 and Chapter 5, we saw that, given a dynamical model defined by an ODE system, it may be beneficial to pass from the original coordinates to new ones. The new coordinates were preferable either because they were identifiable as in Section 4.6 or they defined a state space of smaller dimension as in Chapter 5.

The present chapter is also devoted to a coordinate transformation, called *quadratization*. It aims at "simplifying" a model as well but in a different sense: reducing the *degrees* of the defining equations rather than the dimension. Let us illustrate this idea on a toy example.

▪ **Example 6.1 — Quadratization.** Consider a scalar ODE $x' = x^3$ in a single variable $x = x(t)$ with cubic right-hand side. We augment the state space with a coordinate $w = x^2$. Then we can write the original equation as $x' = xw$ with quadratic right-hand side, and we can do the same for w':

$$w' = 2xx' = 2x^4 = 2w^2.$$

Therefore, the coordinate transformation $x \longrightarrow (x, x^2)$ maps any solution of the original equation to a solution of the following ODE system with at most quadratic right-hand side:

$$x' = xw, \quad w' = 2w^2.$$ ▪

The fact that any ODE system can be embedded into a system with at most quadratic nonlinearities as in the example above has been established at least 100 years ago [5, 74] and has been rediscovered several times since then. In the recent years, such transformations have found applications in a number of areas including model order reduction [11, G8, 50, 71, 72, 94], synthetic biology [37, 55, 56], numerical integration [8, 49, 51, 52], and reachability analysis [42].

6.1 Quadratization algorithms for ODEs

In order to state the results for the ODE case explicitly, we formalize Example 6.1 in a definition.

Definition 6.1.1 — Quadratization. Consider a polynomial system of ODEs

$$\mathbf{x}' = \mathbf{p}(\mathbf{x}), \tag{6.1}$$

where $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), \ldots, p_n(\mathbf{x}))$ with $p_1, \ldots, p_n \in \mathbb{C}[\mathbf{x}]$. Then an ℓ-dimensional vector of new variables

$$\mathbf{w} = \mathbf{w}(\mathbf{x}) \quad \in \mathbb{C}[\mathbf{x}]^\ell \tag{6.2}$$

is said to be a *quadratization* of (6.1) if there exist vectors $\mathbf{q}_1(\mathbf{x},\mathbf{w})$ and $\mathbf{q}_2(\mathbf{x},\mathbf{w})$ of dimensions n and ℓ, respectively, such that $\deg \mathbf{q}_1, \deg \mathbf{q}_2 \leqslant 2$ and

$$\mathbf{x}' = \mathbf{q}_1(\mathbf{x},\mathbf{w}) \quad \text{and} \quad \mathbf{w}' = \mathbf{q}_2(\mathbf{x},\mathbf{w}). \tag{6.3}$$

The dimension ℓ of vector $\mathbf{w}$ is called the *order of quadratization*. A quadratization of the smallest possible order is called an *optimal quadratization*.

If all the polynomials $w_1(\mathbf{x}),\dots,w_\ell(\mathbf{x})$ are monomials, the quadratization is called *a monomial quadratization*. If a monomial quadratization has the smallest possible order among all the monomial quadratizations of the system, it is called *an optimal monomial quadratization*.

The quadratization constructed in Example 6.1 is monomial and of order one, and, therefore, it is an optimal (monomial) quadratization. In general, the orders of optimal quadratization and optimal monomial quadratization can significantly differ. Several examples of this phenomenon were constructed by F. Alauddin (high-school student) in a project [1] under my supervision. The fact that any polynomial ODE admits a monomial quadratization has been known since the early 20th century [5, 74] and has been reproved several times since then (e.g., in [18, 22, 50, 65, 66]). Furthermore, all these proofs are constructive and yield algorithms (essentially, the same algorithm) to construct a monomial quadratization for a given model. However, in most cases, the order of such a quadratization would be too large for meaningful applications.

The first practical, both in terms of the runtime and in terms of the order of the output, algorithm producing quadratizations was designed by Fages, Hemery, and Soliman [56] and implemented in the `BioCham` software [13]. The idea of this algorithm was to take the large quadratization extracted from the constructive proof of the existence and use SAT-solving techniques to find a smallest subset which would still be a quadratization. Although the algorithm does not guarantee optimality, the output often has low order and in many cases turns out to be optimal (see [G9, Table 3]).

Together with A. Bychkov (master student) we have designed an algorithm for finding an *optimal* monomial quadratizations for an ODE system [G9]. It performed a search in the space of all monomial quadratizations following the Branch-and-Bound paradigm and achieved efficiency thanks to domain-specific pruning rules (for example, relating quadratizations and C_4-free graphs). We have implemented this algorithm in `QBee` package in Python. Table 6.1 (an excerpt from [G9, Table 3]) compares `QBee` with `BioCham` in terms of the order of the resulting quadratization and runtime. We see that `QBee` can find quadratizations of lower order and compares favorably to the state-of-the-art in terms of the runtime.

	`BioCham`		`QBee`			`BioCham`		`QBee`	
ODE system	time	order	time	order	ODE system	time	order	time	order
Circular(6)	37.6	5	4.2	5	Hill(15)	64.1	5	0.34	5
Circular(8)	—	—	453.3	6	Hill(20)	—	—	2.4	6
Hard(3)	1.09	11	8.6	9	Monom(3)	0.44	13	84.2	10
Hard(4)	20.2	13	96.9	10	Cubic Cycle(7)	—	—	160.9	14

Table 6.1: Benchmarking packages `BioCham` and `QBee` for the quadratization problem *(for the benchmark details, see [G9])*

The next natural class of models to consider are ODE models *with input*, that is, systems of

the form

$$\mathbf{x}' = \mathbf{p}(\mathbf{x}, \mathbf{u}),$$

where $\mathbf{u} = (u_1, \dots, u_s)$ is a vector of external inputs. In this case, the new variables are allowed to involve inputs and, consequently, the quadratized system may contain the derivatives of the inputs (see [G8, Definition 3.1]).

▪ **Example 6.2 — Quadratization with inputs.** Consider, for example, a scalar ODE $x' = x^2 u$. Then, with a new variable $w = xu$, we can write the system in a quadratic form

$$x' = xw, \quad w' = x'u + xu' = x^2u^2 + xu' = w^2 + xu'.$$ ▪

In our work with A. Bychkov, O. Issan, and B. Kramer [G8], we prove that such quadratization always exists.

Theorem 6.1.1 — A. Bychkov, O. Issan, G. Pogudin, B. Kramer [G8, Theorem 3.3]. Every polynomial system with external inputs $\mathbf{x}' = \mathbf{p}(\mathbf{x}, \mathbf{u})$ admits a monomial quadratization.

We have implemented an efficient algorithm for searching quadratizations in the input case in QBee [G8, Sections 5.2]. In some applications it is desirable, however, to avoid the derivatives of the inputs in the right-hand side of the quadratized system (for example, if input is modeled by a step function). We show [G8, Proposition 3.9] that the problem of finding a quadratization with this restriction is equivalent to a long-open problem in computational algebra of determining if a polynomial vector field is locally finite [35].

Another natural way to extend the standard algorithms for quadratization is to perform quadratizations which would preserve some important properties of the model. Since the quadratized model is typically used in the context of *numerical* computation, it is desirable to preserve the *stability* properties of the input model (this may not happen by default, see [G10, Example 2]!). In our work [G10] with Y. Cai (undergraduate student), we study this problem focusing on the stability of the equilibrium points. Recall that an equilibrium is called *dissipative* if the real parts of the eigenvalues of the linearization of the vector field are negative; dissipativity implies asymptotic stability of the point (and any positive eigenvalue implies instability).

Theorem 6.1.2 — Y. Cai, G. Pogudin [G10, Theorem 1]. For every polynomial system $\mathbf{x}' = \mathbf{p}(\mathbf{x})$, there exists a monomial quadratization which maps the dissipative equilibria of the system to dissipative equilibria of the quadratization.

We also designed and implemented an extension of the QBee algorithm to compute such dissipativity-preserving quadratizations.

6.2 Towards quadratization of PDEs

One of the motivations for studying the quadratization problem is its applications to model order reduction. For any dynamical model, one can learn a quadratic reduced order model for time-series data by projecting on the dominant components and the least squares fitting. However, if the original model was not quadratic, the adequacy of the reduced model obtained by this procedure is questionable. *Lift & Learn* method [102] addresses this challenge by using the coordinate transformation given by a quadratization to lift the time-series data for the original model to the

one for the quadratized model. For the latter, approximating a projection by a quadratic reduced model turns out to be more accurate and better respect the underlying model structure.

High-dimensional models to be reduced often arise as semi-discretizations of PDEs. Consider, for example, a PDE in a single dependent variable $v(t,\xi)$ of the form

$$\frac{\partial v(t,\xi)}{\partial t} = p_0(v(t,\xi)) + p_1(v(t,\xi))\frac{\partial v(t,\xi)}{\partial \xi}, \tag{6.4}$$

where p_0 and p_1 are univariate polynomials. If we fix N and semi-discretize (6.4) in the ξ variable by introducing $x_i(t) := v(t, i/N)$ for $i = 0,\dots,N$, then we obtain a $(N+1)$-dimensional ODE system of the form

$$x_i' = p_0(x_i) + p_1(x_i)\ell_i(x_0,\dots,x_N), \quad i = 0,\dots,N, \tag{6.5}$$

where $\ell(x_0,\dots,x_N)$ is a linear function corresponding to the discretization of the operator $\frac{\partial}{\partial \xi}$ (for example, $\ell_i = \frac{x_{i+1}-x_i}{1/N}$). In order to apply the Lift & Learn method sketched above, one has to quadratize the ODE system (6.5). On one hand, the dimension of this system can be equal to hundreds or thousands which is beyond the capacities of the existing quadratization software. On the order hand, the equations in the system have the same structure, so one may expect to find structured quadratizations admitting a concise representation.

In our work with A. Bychkov, O. Issan, and B. Kramer [G8], we have formalized this observation by defining *dimension-agnostic quadratizations* [G8, Section 4] which is a way to simultaneously describe quadratizing transformation for the systems (6.5) derived from (6.4) with different N's and ℓ_i's. An example of such description would be "*all variables of the form* x_i^2 *and all variables of the form* $x_i x_j$*, where* x_j *appears in* ℓ_i". We prove that dimension-agnostic quadratization exists for a large class of cases including evolutionary PDEs which are affine with respect to the spatial derivatives (as, e.g., (6.4)) in [G8, Theorem 4.6] and show that it can be found by computing a quadratization for $N = 4$ and a specific set of ℓ_i's [G8, Proposition 4.7], which can be done efficiently using `QBee`.

We have applied the resulting quadratization algorithm for reducing an ODE system appearing as a semi-discretization of the *solar wind* model using real data [G8, Section 7]. The ODE system to be reduced had dimension more than a hundred but we could quadratize it by running `QBee` on a system of dimension eight. The resulting six-dimensional reduced model has achieved much better accuracy than the one obtained without using quadratization as shown on Figure 6.1.

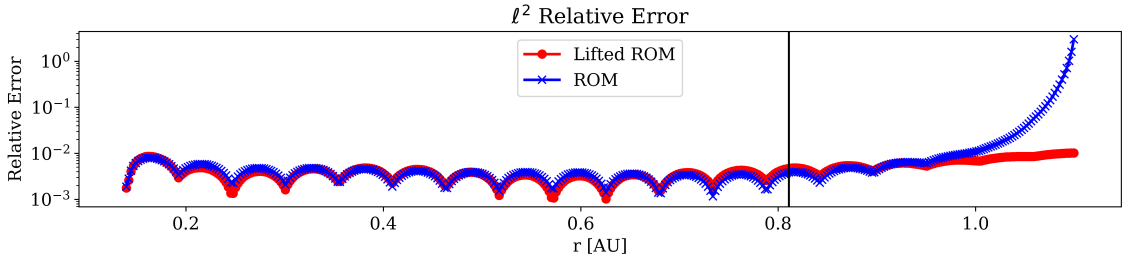


Figure 6.1: Approximation error for reduced order models obtained with (red) and without (blue) quadratization (taken from [G8, Figure 7.2])

7. Research project

In this final chapter I describe several topics for future research continuing the lines of work presented in this manuscript. The topics selected for this chapter represent three main axes of prospective studies: deeper understanding of already studied problems and, thus, more efficient algorithms for them (Section 7.1), overhauling existing methods to address challenges arising from applications (Section 7.2), and applying tools from differential/difference algebra to new domains (Section 7.3).

7.1 Fighting expression swell: concise representations and numerical computation

While the algorithms and software described in this thesis compare favorably to the state-of-the-art and can considerably push the boundaries of what can be practically computed, the majority of them (except for the linear model reduction algorithms from Chapter 5) are mostly limited to models of dimension not exceeding 20. One of the main reasons for them not to scale further is a well-known problem in symbolic computation: *intermediate expression swell*. This means that the polynomials or equations appearing during the computation may become huge and hinder the efficiency. For example, input-output equations of a model of dimension as small as four or five can easily contain thousands of terms. We propose two related ways to address this challenge.

Alternative representations

While the result of differential elimination indeed may contain thousands of terms, this may be not because of the *inherent complexity* of this differential polynomial but rather due to the fact that the standard *monomial representation* is not the most appropriate one. For example, consider a system of differential equations in $x_1, \dots, x_n$ and $y_1, \dots, y_n$:

$$x_1 y_1 + \dots + x_n y_n = 0, \;\; x_1' = 0, \; \dots \; x_n' = 0,$$

which expresses the fact that functions $y_1, \dots, y_n$ are linearly dependent over constants. Then, if one eliminates $\mathbf{x}$'s, the resulting ideal will contain the Wronskian of $\mathbf{y}$ which is a polynomial with $n!$ terms but admits a concise representation. Our initial studies indicate that Wronskian-like expressions are abundant in the results of differential elimination. One of the reasons for this is the fact that *differential resultants* are known to be *differentially homogeneous* [79, Theorem 5.13], and differentially homogeneous polynomials are spanned by Wronskians of certain form [36]. A short-term goal would be to study differential resultants in this special basis aiming at, in a longer term, developing a general theory of such representations and using it in the algorithms.

Another direction is to allow *integro-differential polynomials*: a recent work by Lemaire and Roussel [76] shows that the result of integral elimination may be smaller than the purely differential one by several orders of magnitude (and beneficial from the numerical standpoint [121])! Developing systematic theory and algorithms to take advantage of this is also a direction for future research. The first step in this direction would be to polish and refine a method for speeding up identifiability analysis via partial integration of an ODE model developed with Stefan Vayl (undergraduate student) which has already been successfully used in a case study [7, Section 3.2.2]. Questions for the longer term include theory for showing (im)possibility of integral/integro-differential elimination and general algorithms for performing it.

Numerical computation

In the case of polynomial system solving, one way to avoid computing with large expressions is to use numerical computation, this approach is referred to as *numerical algebraic geometry* [116]. In this approach, polynomial ideals are represented by several of their approximate numerical solutions (witness set). For differential equations, one could use numerical or power series solutions as they both can be viewed as approximations but in different topologies. In both cases, efficient algorithms for computing such solutions are available. Furthermore, power series solutions have already been used in our elimination algorithm for ODE models in Section 3.4 in order to distinguish between different components of an ideal. Further replacing ideals with their solutions for heavy computations and then employing sparse interpolation to reconstruct the result at the end can have dramatic impact on the performance of the algorithm. A short-term task in this direction is to develop an elimination algorithm for polynomial differential models following this evaluation-interpolation approach. Our work in progress with Yulia Mukhina (PhD student) yields such an algorithm and shows that it is indeed capable to push the limits of computation. Further steps in longer term would involve avoiding performing full reconstruction and using representations via solutions (power series and/or numerical) to answer the questions of interest directly.

7.2 Transformations for learning

One of the recent trends in computing with dynamical models is to apply *learning* techniques (also referred to as scientific machine learning). Examples of this include learning a smaller surrogate model or learning a numerical solution for the corresponding equations instead of applying classical numerical simulation methods. One of the key steps in applying machine learning techniques is transforming the data from the original representation to a potentially more convenient one (feature engineering). In the context of scientific machine learning, data often comes from a model, so one can search for such transformations on the level of the model. This is precisely what happens in the applications of quadratization algorithms to *model order reduction* in Section 6.2 where a reduced model learned using the quadratized equations turns out to be more accurate than the one learned using the original model. Continuing this research, the next short-term goal is to develop quadratization algorithms to be applied to PDEs directly in order to obtain more versatile and concise quadratizations in the context of model order reduction, this is a work in progress with Boris Kramer and Albani Olivieri (master student).

Another example involving the results presented in this thesis is *learning parameters* of ODE models from time series using neural networks: acquiring enough time series to ensure structural identifiability is one of the steps of the workflow presented in [29] (and our software `StructuralIdentifiability.jl` described in Section 4.5 is used in the workflow). Unlike the

model reduction example above, only the result of identifiability analysis is used in learning, and the model transformation remains behind the scenes. A short-term task is to make a full use of the developed techniques by performing model *reparametrization* described in Section 4.6 and using the new coordinates for learning. This requires making the experimental approach presented here efficient and robust. In longer term, an important question would also to make the new coordinates not only identifiable but *interpretable* so that learning them would be especially insightful. Furthermore, even when parameters and states are identifiable, i.e. could be in principle learned, one has still significant freedom in choosing the specific coordinates, and this affects the accuracy and robustness of computation. For a recent example involving differential elimination see [70]. Systematically developing tools for such coordinate choice in already identifiable model is an intriguing problem for future long-term research.

7.3 Symbolic computation and differential algebra for calculus of variations

Many dynamical models, especially defined by (partial) differential equations, are derived from fundamental *optimality principles* (minimization of energy, Fermat's principle, etc), and many important structural features they possess are in fact inherited from the functionals they optimize. It is therefore natural to develop symbolic and algebraic tools to work with these functionals directly. Several interesting and original works exploring and exploiting the connections between calculus of variations and differential algebra including [107] and papers by P. Olver [90, 92, 93] appeared in the 1980-s. An interesting direction for future research is to revisit these results from the point of view of recent developments in differential algebra and advances in symbolic computation including the ones described in the present thesis. In particular, our works with R. Ait El Manssour [G2, G3] on differential analogues of dual numbers could be a starting point for developing algebraic formalism for infinitesimal variations. Short- and medium-term projects suitable for entering the new domain and exploring the algebraic aspects of it may include dimension counting for high-order null Lagrangians [90] and homogeneous Lagrangians, and explicit symbolic computation algorithms for the inverse problem of calculus of variations [2, 33].

Papers by Gleb Pogudin

This section collects our papers written after my PhD in 2016. Most of them are discussed in the present thesis except for [G1, G4, G5, G7, G13, G16, G23, G24, G27, G35, G40, G43]. Student co-authors under my supervision are marked with an asterisk *.

[G1] S. Abramov and G. Pogudin. *On the dimension of the solution space of linear difference equations over the ring of infinite sequences*. 2023. URL: https://arxiv.org/abs/2311.02219 (cit. on p. 51).

[G2] R. Ait El Manssour* and G. Pogudin. "*Multiplicity structure of the arc space of a fat point*". In: Algebra & Number Theory (2024). URL: https://msp.org/ant/2024/18-5/p03.xhtml (cit. on pp. 11, 16, 50).

[G3] R. Ait El Manssour* and G. Pogudin. *Wronskians form the inverse system of the arcs of a double point*. 2024. URL: https://arxiv.org/abs/2405.08964 (cit. on pp. 11, 16, 50).

[G4] E. Amzallag, A. Minchenko, and G. Pogudin. "*Degree bound for toric envelope of a linear algebraic group*". In: Mathematics of Computation (Oct. 2021). URL: http://dx.doi.org/10.1090/mcom/3695 (cit. on p. 51).

[G5] E. Amzallag, M. Sun, G. Pogudin, and T. N. Vo. "*Complexity of triangular representations of algebraic sets*". In: Journal of Algebra 523 (Apr. 2019), 342–364. URL: http://dx.doi.org/10.1016/j.jalgebra.2019.01.007 (cit. on p. 51).

[G6] M. Bessonov, I. Ilmer, T. Konstantinova, A. Ovchinnikov, G. Pogudin, and P. Soto. *Faster Gröbner bases for Lie derivatives of ODE systems via monomial orderings*. accepted for publication to the International Symposium for Symbolic and Algebraic Computation. 2024. URL: https://arxiv.org/abs/2202.06297 (cit. on p. 33).

[G7] M. Buchacher, M. Kauers, and G. Pogudin. "*Separating variables in bivariate polynomial ideals*". In: Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation. ISSAC '20. ACM, July 2020. URL: http://dx.doi.org/10.1145/3373207.3404028 (cit. on p. 51).

[G8] A. Bychkov*, O. Issan, G. Pogudin, and B. Kramer. "*Exact and optimal quadratization of nonlinear finite-dimensional non-autonomous dynamical systems*". In: SIAM Journal on Applied Dynamical Systems 23.1 (2024). URL: https://doi.org/10.1137/23M1561129 (cit. on pp. 44, 46, 47).

[G9] A. Bychkov* and G. Pogudin. "*Optimal monomial quadratization for ODE systems*". In: International Workshop on Combinatorial Algorithms (IWOCA). Springer International Publishing, 2021, 122–136. URL: http://dx.doi.org/10.1007/978-3-030-79987-8_9 (cit. on p. 45).

[G10] Y. Cai* and G. Pogudin. "*Dissipative quadratizations of polynomial ODE systems*". In: Tools and Algorithms for Construction and Analysis of Systems (TACAS). Springer Nature Switzerland, 2024, 323–342. URL: http://dx.doi.org/10.1007/978-3-031-57249-4_16 (cit. on p. 46).

[G11] A. Demin*, E. Demitraki*, and G. Pogudin. "*Exact hierarchical reductions of dynamical models via linear transformations*". In: Communications in Nonlinear Science and Numerical Simulation 131 (Apr. 2024), p. 107816. URL: http://dx.doi.org/10.1016/j.cnsns.2024.107816 (cit. on pp. 41, 42).

[G12] R. Dong*, C. Goodbrake, H. A. Harrington, and G. Pogudin. "*Differential elimination for dynamical models via projections with applications to structural identifiability*". In: SIAM Journal on Applied Algebra and Geometry 7.1 (Mar. 2023), 194–235. URL: http://dx.doi.org/10.1137/22M1469067 (cit. on pp. 22, 23, 34, 35).

[G13] E. S. Golod and G. Pogudin. "*Modules of the Gorenstein dimension zero over algebras of graphs*". In: Sbornik: Mathematics 207.7 (2016), 81–100. URL: http://dx.doi.org/10.4213/sm8563 (cit. on p. 51).

[G14] R. Gustavson, A. Ovchinnikov, and G. Pogudin. "*New order bounds in differential elimination algorithms*". In: Journal of Symbolic Computation 85 (2018), pp. 128–147. URL: http://dx.doi.org/10.1016/j.jsc.2017.07.006 (cit. on p. 19).

[G15] R. Gustavson, A. Ovchinnikov, and G. Pogudin. "*Bounds for orders of derivatives in differential elimination algorithms*". In: Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation. 2016, pp. 239–246. URL: http://doi.acm.org/10.1145/2930889.2930922 (cit. on p. 19).

[G16] J. van der Hoeven and G. Pogudin. "*A zero test for σ-algebraic power series*". In: Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation. ISSAC '21. New York, NY, USA, 2021, 187–192. URL: https://doi.org/10.1145/3452143.3465549 (cit. on p. 51).

[G17] H. Hong, A. Ovchinnikov, G. Pogudin, and C. Yap. "*SIAN: software for structural identifiability analysis of ODE models*". In: Bioinformatics 35.16 (2019), pp. 2873–2874. URL: https://doi.org/10.1093/bioinformatics/bty1069 (cit. on pp. 33–35).

[G18] H. Hong, A. Ovchinnikov, G. Pogudin, and C. Yap. "*Global identifiability of differential models*". In: Communications on Pure and Applied Mathematics 73.9 (2020), pp. 1831–1879. URL: https://doi.org/10.1002/cpa.21921 (cit. on pp. 30, 31, 33).

[G19] I. Ilmer, A. Ovchinnikov, and G. Pogudin. "*Maple application for structural identifiability analysis of ODE models*". In: ACM Communications in Computer Algebra 55.2 (June 2021), 49–53. URL: http://dx.doi.org/10.1145/3493492.3493497 (cit. on p. 34).

[G20] I. Ilmer, A. Ovchinnikov, and G. Pogudin. "*Web-based structural identifiability analyzer*". In: Computational Methods in Systems Biology. 2021, 254–265. URL: http://dx.doi.org/10.1007/978-3-030-85633-5_17 (cit. on p. 34).

[G21] A. Jiménez-Pastor, J. P. Jacob*, and G. Pogudin. "*Exact linear reduction for rational dynamical systems*". In: Computational Methods in Systems Biology. 2022, 198–216. URL: http://dx.doi.org/10.1007/978-3-031-15034-0_10 (cit. on p. 39).

[G22] A. Jiménez-Pastor and G. Pogudin. "*Computing exact nonlinear reductions of dynamical models*". In: ACM Communications in Computer Algebra 56.2 (2022), 25–31. URL: http://dx.doi.org/10.1145/3572867.3572869 (cit. on pp. 42, 43).

[G23] M. Kauers and G. Pogudin. "*Bounds for substituting algebraic functions into D-finite functions*". In: Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation. ISSAC '17. ACM, July 2017. URL: http://dx.doi.org/10.1145/3087604.3087616 (cit. on p. 51).

[G24] I. Koswara, G. Pogudin, S. Selivanova, and M. Ziegler. "*Bit-complexity of classical solutions of linear evolutionary systems of partial differential equations*". In: Journal of Complexity 76 (June 2023), p. 101727. URL: http://dx.doi.org/10.1016/j.jco.2022.101727 (cit. on p. 51).

[G25] W. Li, A. Ovchinnikov, G. Pogudin, and T. Scanlon. "*Algorithms yield upper bounds in differential algebra*". In: Canadian Journal of Mathematics 75.1 (Sept. 2021), 29–51. URL: http://dx.doi.org/10.4153/S0008414X21000560 (cit. on pp. 19, 22).

[G26] W. Li, A. Ovchinnikov, G. Pogudin, and T. Scanlon. "*Elimination of unknowns for systems of algebraic differential-difference equations*". In: Transactions of the American Mathematical Society 374 (2021), pp. 303–326. URL: https://doi.org/10.1090/tran/8219 (cit. on p. 22).

[G27] T. Novikova, A. Ovchinnikov, G. Pogudin, and J. C. Ramella-Roman. "*Time-efficient filtering of imaging polarimetric data by checking physical realizability of experimental Mueller matrices*". In: Bioinformatics (June 2024). Ed. by H. Peng. URL: http://dx.doi.org/10.1093/bioinformatics/btae348 (cit. on p. 51).

[G28] A. Ovchinnikov, G. Pogudin, and T. N. Vo. "*Bounds for elimination of unknowns in systems of differential-algebraic equations*". In: International Mathematical Research Notices (2022). URL: https://doi.org/10.1093/imrn/rnaa302 (cit. on p. 20).

[G29] A. Ovchinnikov, I. Pérez Verona, G. Pogudin, and M. Tribastone. "*CLUE: exact maximal reduction of kinetic models by constrained lumping of differential equations*". In: Bioinformatics (2021). URL: https://doi.org/10.1093/bioinformatics/btab010 (cit. on pp. 39–41).

[G30] A. Ovchinnikov, A. Pillay, G. Pogudin, and T. Scanlon. "*Computing all identifiable functions of parameters for ODE models*". In: Systems and Control Letters 157 (Nov. 2021), p. 105030. URL: http://dx.doi.org/10.1016/j.sysconle.2021.105030 (cit. on pp. 32–35).

[G31] A. Ovchinnikov, A. Pillay, G. Pogudin, and T. Scanlon. "*Multi-experiment parameter identifiability of ODEs and model theory*". In: SIAM Journal on Applied Algebra and Geometry 6.3 (July 2022), 339–367. URL: http://dx.doi.org/10.1137/21M1389845 (cit. on p. 32).

[G32] A. Ovchinnikov, G. Pogudin, and T. Scanlon. "*Effective difference elimination and Nullstellensatz*". In: Journal of the European Mathematical Society 22.8 (Apr. 2020), 2419–2452. URL: http://dx.doi.org/10.4171/JEMS/968 (cit. on pp. 16, 17, 21).

[G33] A. Ovchinnikov, G. Pogudin, and P. Thompson. "*Input-output equations and identifiability of linear ODE models*". In: IEEE Transactions on Automatic Control 68.2 (Feb. 2023), 812–824. URL: http://dx.doi.org/10.1109/TAC.2022.3145571 (cit. on p. 32).

[G34] A. Ovchinnikov, G. Pogudin, and P. Thompson. "*Parameter identifiability and input–output equations*". In: Applicable Algebra in Engineering, Communication and Computing 34.2 (Feb. 2021), 165–182. URL: http://dx.doi.org/10.1007/s00200-021-00486-8 (cit. on pp. 31, 33, 34).

[G35] E. Paul*, G. Pogudin, W. Qin*, and R. Laubenbacher. "*The dynamics of canalizing boolean networks*". In: Complexity 2020 (Jan. 2020), 1–14. URL: http://dx.doi.org/10.1155/2020/3687961 (cit. on p. 51).

[G36] D. Pavlov* and G. Pogudin. "*On realizing differential-algebraic equations by rational dynamical systems*". In: Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation. ISSAC '22. ACM, July 2022. URL: http://dx.doi.org/10.1145/3476446.3535492 (cit. on p. 23).

[G37] D. Pavlov*, G. Pogudin, and Y. P. Razmyslov. "*From algebra to analysis: new proofs of theorems by Ritt and Seidenberg*". In: Proceedings of the American Mathematical Society (July 2022). URL: http://dx.doi.org/10.1090/proc/16065 (cit. on pp. 11, 16).

[G38] G. Pogudin. "*A differential analog of the Noether Normalization Lemma*". In: International Mathematics Research Notices (Dec. 2016), rnw275. URL: http://dx.doi.org/10.1093/imrn/rnw275 (cit. on p. 11).

[G39] G. Pogudin. "*A primitive element theorem for fields with commuting derivations and automorphisms*". In: Selecta Mathematica 25.4 (2019). URL: http://dx.doi.org/10.1007/s00029-019-0504-9 (cit. on pp. 11, 25).

[G40] G. Pogudin. "*Power series expansions for the planar monomer-dimer problem*". In: Physical Review E 96.3 (Sept. 2017). URL: http://dx.doi.org/10.1103/PhysRevE.96.033303 (cit. on p. 51).

[G41] G. Pogudin. "*Products of ideals and jet schemes*". In: Journal of Algebra 502 (May 2018), 61–78. URL: http://dx.doi.org/10.1016/j.jalgebra.2018.01.027 (cit. on pp. 20, 24).

[G42] G. Pogudin, T. Scanlon, and M. Wibmer. "*Solving difference equations in sequences: Universality and Undecidability*". In: Forum of Mathematics, Sigma 8 (2020). URL: http://dx.doi.org/10.1017/fms.2020.14 (cit. on pp. 17, 21).

[G43] G. Pogudin and A. Szanto. "*Irredundant triangular decomposition*". In: Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation. ISSAC '18. ACM, July 2018. URL: http://dx.doi.org/10.1145/3208976.3208996 (cit. on p. 51).

[G44] G. Pogudin and X. Zhang*. "*Interpretable exact linear reductions via positivity*". In: Computational Methods in Systems Biology. 2021, 91–107. URL: http://dx.doi.org/10.1007/978-3-030-85633-5_6 (cit. on p. 40).

References

[1] F. Alauddin*. "*Quadratization of ODEs: monomial vs. non-monomial*". In: SIAM Undergraduate Research Online 14 (2021). URL: http://dx.doi.org/10.1137/20S1360578 (cit. on p. 45).

[2] I. Anderson and G. Thompson. *The inverse problem of the calculus of variations for ordinary differential equations*. Vol. 473. American Mathematical Soc., 1992 (cit. on p. 50).

[3] F. Anstett-Collin, L. Denis-Vidal, and G. Millérioux. "*A priori identifiability: an overview on definitions and approaches*". In: Annual Reviews in Control 50 (2020), pp. 139–149 (cit. on p. 29).

[4] A. Antoulas. *Approximation of large-scale dynamical systems*. Adv. in Design and Control. SIAM, 2005 (cit. on p. 37).

[5] G. G. Appelroth. "*Basic form of a system of algebraic differential equations*". In: Sbornik: Mathematics 23.1 (1902), pp. 12–23. URL: http://mi.mathnet.ru/sm6683 (cit. on pp. 44, 45).

[6] T. Bächler, V. Gerdt, M. Lange-Hegermann, and D. Robertz. "*Thomas decomposition of algebraic and differential systems*". In: Computer Algebra in Scientific Computing. 2010, pp. 31–54. URL: https://doi.org/10.1007/978-3-642-15274-0_4 (cit. on pp. 9, 18, 22).

[7] G. Ballif, F. Clément, and R. Yvinec. "*Nonlinear compartmental modeling to monitor ovarian follicle population dynamics on the whole lifespan*". In: Journal of Mathematical Biology 89.1 (June 2024). URL: http://dx.doi.org/10.1007/s00285-024-02108-6 (cit. on p. 49).

[8] F. Bayer, R. I. Leine, O. Thomas, and A. Grolet. "*Koopman-Hill stability computation of periodic orbits in polynomial dynamical systems using a real-valued quadratic harmonic balance formulation*". In: (2024). URL: http://dx.doi.org/10.2139/ssrn.4811433 (cit. on p. 44).

[9] R. Bellman and K. Åström. "*On structural identifiability*". In: Mathematical Biosciences 7 (1970), pp. 329–339. URL: http://dx.doi.org/10.1016/0025-5564(70)90132-X (cit. on p. 29).

[10] G. Bellu, M. P. Saccomani, S. Audoly, and L. D'Angiò. "*DAISY: A new software tool to test global identifiability of biological and physiological systems*". In: Computer Methods and Programs in Biomedicine 88.1 (2007), pp. 52–61. URL: http://dx.doi.org/10.1016/j.cmpb.2007.07.002 (cit. on pp. 9, 29, 31, 33, 35).

[11] P. Benner and T. Breiten. "*Two-sided projection methods for nonlinear model order reduction*". In: SIAM Journal on Scientific Computing 37.2 (2015), B239–B260 (cit. on p. 44).

[12] P. Benner, S. Gugercin, and K. Willcox. "*A survey of projection-based model reduction methods for parametric dynamical systems*". In: SIAM Review 57.4 (2015), pp. 483–531. URL: https://doi.org/10.1137/130932715 (cit. on p. 37).

[13] *Biocham 4.0*. URL: http://contraintes.inria.fr/biocham/ (cit. on p. 45).

[14] N. Borisov, B. Kholodenko, J. Faeder, and A. Chistopolsky. "*Domain-oriented reduction of rule-based network models*". In: IET Systems Biology 2.5 (2008), pp. 342–351. URL: https://doi.org/10.1049/iet-syb:20070081 (cit. on p. 43).

[15] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. "*Representation for the radical of a finitely generated differential ideal*". In: Proceedings of the 1995 international symposium on Symbolic and algebraic computation - ISSAC'95. 1995. URL: https://doi.org/10.1145/220346.220367 (cit. on p. 9).

[16] F. Boulier. *BLAD: Bibliothèques Lilloises d'Algèbre Différentielle*. URL: https://pro.univ-lille.fr/francois-boulier/logiciels/blad/ (cit. on pp. 18, 22).

[17] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. "*Computing representations for radicals of finitely generated differential ideals*". In: Applicable Algebra in Engineering, Communication and Computing 20.1 (2009), pp. 73–121. URL: https://doi.org/10.1007/s00200-009-0091-7 (cit. on pp. 9, 18).

[18] L. Brenig. "*Reducing nonlinear dynamical systems to canonical forms*". In: Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 376.2124 (June 2018), p. 20170384. URL: http://dx.doi.org/10.1098/rsta.2017.0384 (cit. on p. 45).

[19] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin. "*ERODE: a tool for the evaluation and reduction of ordinary differential equations*". In: TACAS 2017. Vol. 10206. LNCS. 2017. URL: https://doi.org/10.1007/978-3-662-54580-5_19 (cit. on pp. 38, 40).

[20] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin. "*Maximal aggregation of polynomial dynamical systems*". In: Proceedings of the National Academy of Sciences 114.38 (2017), pp. 10029–10034. ISSN: 0027-8424. URL: https://www.pnas.org/content/114/38/10029 (cit. on p. 38).

[21] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin. "*Symbolic computation of differential equivalences*". In: Theoretical Computer Science 777 (2019), pp. 132–154. URL: https://doi.org/10.1016/j.tcs.2019.03.018 (cit. on p. 38).

[22] D. C. Carothers, G. Parker, J. S. Sochacki, and P. G. Warne. "*Some properties of solutions to polynomial systems of differential equations.*" In: Electronic Journal of Differential Equations (EJDE) (2005), Paper No. 40. URL: http://eudml.org/doc/125330 (cit. on p. 45).

[23] Z. Chatzidakis and E. Hrushovski. "*Model theory of difference fields*". In: Transactions of the American Mathematical Society 351.8 (1999), pp. 2997–3071. URL: http://dx.doi.org/10.1090/S0002-9947-99-02498-8 (cit. on p. 21).

[24] A. Chernikova, N. Gozzi, N. Perra, S. Boboila, T. Eliassi-Rad, and A. Oprea. "*Modeling self-propagating malware with epidemiological models*". In: Applied Network Science 8.1 (Aug. 2023). URL: http://dx.doi.org/10.1007/s41109-023-00578-z (cit. on p. 10).

[25] O.-T. Chis, J. R. Banga, and E. Balsa-Canto. "*Structural identifiability of systems biology models: a critical comparison of methods*". In: PLOS ONE 6.11 (Nov. 2011), pp. 1–16. URL: http://dx.doi.org/10.1371/journal.pone.0027755 (cit. on pp. 29, 33).

[26] R. M. Cohn. *Difference algebra*. Interscience Publishers, 1965 (cit. on pp. 9, 25).

[27] G. Conte, C. H. Moog, and A. M. Perdon. *Algebraic methods for nonlinear control systems*. Springer London, 2007. URL: https://doi.org/10.1007/978-1-84628-595-0 (cit. on p. 22).

[28] L. D'Alfonso, G. Jeronimo, and P. Solernó. "*Effective differential Nullstellensatz for ordinary DAE systems with constant coefficients*". In: Journal of Complexity 30.5 (2014), pp. 588–603. URL: http://dx.doi.org/10.1016/j.jco.2014.01.001 (cit. on pp. 19, 20).

[29] M. Daneker, Z. Zhang, G. E. Karniadakis, and L. Lu. "*Systems biology: identifiability analysis and parameter identification via systems-biology-informed neural networks*". In: Computational Modeling of Signaling Networks. Springer US, 2023, 87–105. URL: http://dx.doi.org/10.1007/978-1-0716-3008-2_4 (cit. on pp. 9, 49).

[30] E. A. Dankwa, C. A. Donnelly, A. F. Brouwer, R. Zhao, M. P. Montgomery, M. K. Weng, and N. K. Martin. "*Estimating vaccination threshold and impact in the 2017-2019 hepatitis A virus outbreak among persons experiencing homelessness or who use drugs in Louisville, Kentucky, United States*". In: Vaccine 39.49 (2021), pp. 7182–7190. URL: https://doi.org/10.1016/j.vaccine.2021.10.001 (cit. on p. 10).

[31] S. Diop and M. Fliess. "*Nonlinear observability, identifiability, and persistent trajectories*". In: Proceedings of the 30th IEEE Conference on Decision and Control. IEEE, 1991. URL: http://dx.doi.org/10.1109/CDC.1991.261405 (cit. on p. 30).

[32] S. Diop. "*Elimination in control theory*". In: Mathematics of Control, Signals, and Systems 4.1 (Mar. 1991), 17–32. URL: http://dx.doi.org/10.1007/BF02551378 (cit. on p. 9).

[33] J. Douglas. "*Solution of the inverse problem of the calculus of variations*". In: Transactions of the American Mathematical Society 50.1 (1941), 71–128. URL: http://dx.doi.org/10.1090/S0002-9947-1941-0004740-5 (cit. on p. 50).

[34] S. B. Ekhad and D. Zeilberger. "*How to generate as many Somos-like miracles as you wish*". In: Journal of Difference Equations and Applications 20 (5-6 2014), pp. 852–858. URL: http://dx.doi.org/10.1080/10236198.2013.823956 (cit. on p. 21).

[35] A. van den Essen. "*Locally nilpotent derivations and their applications, III*". In: Journal of Pure and Applied Algebra 98.1 (Jan. 1995), 15–23. URL: http://dx.doi.org/10.1016/0022-4049(95)90013-6 (cit. on p. 46).

[36] A. Etesse. *On the Schmidt-Kolchin conjecture on differentially homogeneous polynomials. applications to (twisted) jet differentials on projective spaces*. 2023. URL: https://arxiv.org/abs/2302.02634 (cit. on p. 48).

[37] F. Fages, G. Le Guludec, O. Bournez, and A. Pouly. "*Strong Turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs*". In: Computational Methods in Systems Biology: 15th International Conference, CMSB 2017, Darmstadt, Germany, September 27–29, 2017, Proceedings 15. Springer. 2017, pp. 108–127. URL: https://doi.org/10.1007/978-3-319-67471-1_7 (cit. on p. 44).

[38] M. Feinberg. *Foundations of chemical reaction network theory*. Springer Cham, 2019. URL: https://doi.org/10.1007/978-3-030-03858-8 (cit. on p. 40).

[39] J. Feret, V. Danos, J. Krivine, R. Harmer, and W. Fontana. "*Internal coarse-graining of molecular systems*". In: Proceedings of the National Academy of Sciences 106.16 (2009), pp. 6453–6458. URL: http://dx.doi.org/10.1073/pnas.0809908106 (cit. on p. 38).

[40] J. Feret, T. Henzinger, H. Koeppl, and T. Petrov. "*Lumpability abstractions of rule-based systems*". In: Theoretical Computer Science 431 (2012), pp. 137–164. URL: https://doi.org/10.1016/j.tcs.2011.12.059 (cit. on p. 38).

[41] M. Fliess. "*Nonlinear control theory and differential algebra: some illustrative examples*". In: IFAC Proceedings Volumes 20.5 (July 1987), 103–107. URL: http://dx.doi.org/10.1016/S1474-6670(17)55072-7 (cit. on p. 9).

[42] M. Forets and C. Schilling. "*Reachability of weakly nonlinear systems using Carleman linearization*". In: Lecture Notes in Computer Science. Springer International Publishing, 2021, pp. 85–99. URL: https://doi.org/10.1007%2F978-3-030-89716-1_6 (cit. on p. 44).

[43] K. Forsman. "*On rational state space realizations*". In: Nonlinear Control Systems Design. 1993, pp. 341–346. URL: https://doi.org/10.1016/b978-0-08-041901-5.50061-0 (cit. on p. 23).

[44] X.-S. Gao, Y. Luo, and C. Yuan. "*A characteristic set method for ordinary difference polynomial systems*". In: Journal of Symbolic Computation 44.3 (2009), pp. 242–260. URL: doi.org/10.1016/j.jsc.2007.05.005 (cit. on p. 21).

[45] K. Ghorbal and A. Platzer. "*Characterizing algebraic invariants by differential radical invariants*". In: Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, 279–294. ISBN: 9783642548628. URL: http://dx.doi.org/10.1007/978-3-642-54862-8_19 (cit. on p. 9).

[46] N. Gogishvili*. *Database for identifiability properties of linear compartmental models*. 2024. URL: https://arxiv.org/abs/2406.16132 (cit. on p. 32).

[47] O. Golubitsky, M. Kondratieva, A. Ovchinnikov, and A. Szanto. "*A bound for orders in differential Nullstellensatz*". In: Journal of Algebra 322.11 (2009), pp. 3852–3877. URL: http://dx.doi.org/10.1016/j.jalgebra.2009.05.032 (cit. on p. 19).

[48] O. Golubitsky, M. Kondratieva, M. Moreno Maza, and A. Ovchinnikov. "*A bound for the Rosenfeld–Gröbner algorithm*". In: Journal of Symbolic Computation 43.8 (Aug. 2008), 582–610. URL: http://dx.doi.org/10.1016/j.jsc.2007.12.002 (cit. on p. 19).

[49] C. Gözükirmizi and M. Demiralp. "*Solving ODEs by obtaining purely second degree multinomials via branch and bound with admissible heuristic*". In: Mathematics 7.4 (2019). URL: https://www.mdpi.com/2227-7390/7/4/367 (cit. on p. 44).

[50] C. Gu. "*QLMOR: a projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems*". In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 30.9 (2011), pp. 1307–1320. URL: https://doi.org/10.1109/TCAD.2011.2142184 (cit. on pp. 44, 45).

[51] L. Guillot, B. Cochelin, and C. Vergez. "*A generic and efficient Taylor series–based continuation method using a quadratic recast of smooth nonlinear systems*". In: International Journal for numerical methods in Engineering 119.4 (2019), pp. 261–280. URL: https://doi.org/10.1002/nme.6049 (cit. on p. 44).

[52] L. Guillot, B. Cochelin, and C. Vergez. "*A Taylor series-based continuation method for solutions of dynamical systems*". In: Nonlinear Dynamics 98.4 (2019), pp. 2827–2845. URL: https://doi.org/10.1007/s11071-019-04989-5 (cit. on p. 44).

[53] R. Gustavson, M. Kondratieva, and A. Ovchinnikov. "*New effective differential Nullstellensatz*". In: Advances in Mathematics 290 (2016), pp. 1138–1158. URL: http://dx.doi.org/10.1016/j.aim.2015.12.021 (cit. on pp. 19, 20).

[54] A. Haddak. "*Differential algebra and controllability*". In: Nonlinear Control Systems Design 1989. 1990, 13–16. URL: http://dx.doi.org/10.1016/B978-0-08-037022-4.50008-9 (cit. on p. 9).

[55] M. Hemery and F. Fages. "*Algebraic biochemistry: a framework for analog online computation in cells*". In: Computational Methods in Systems Biology. 2022, pp. 3–20. URL: https://doi.org/10.1007/978-3-031-15034-0_1 (cit. on p. 44).

[56] M. Hemery, F. Fages, and S. Soliman. "*On the complexity of quadratization for polynomial differential equations*". In: Computational Methods in Systems Biology. Ed. by A. Abate, T. Petrov, and V. Wolf. Cham: Springer International Publishing, 2020, pp. 120–140. URL: https://doi.org/10.1007/978-3-030-60327-4_7 (cit. on pp. 44, 45).

[57] R. Hermann and A. J. Krener. "*Nonlinear controllability and observability*". In: IEEE Transactions on automatic control 22.5 (1977), pp. 728–740. URL: http://doi.org/10.1109/TAC.1977.1101601 (cit. on p. 29).

[58] E. Hrushovski and F. Point. "*On von Neumann regular rings with an automorphism*". In: Journal of Algebra 315.1 (2007), pp. 76–120. URL: http://dx.doi.org/10.1016/j.jalgebra.2007.05.006 (cit. on p. 21).

[59] E. Hubert. "*Factorization-free decomposition algorithms in differential algebra*". In: Journal of Symbolic Computation 29.4–5 (May 2000), 641–662. URL: http://dx.doi.org/10.1006/jsco.1999.0344 (cit. on p. 9).

[60] E. Hubert. "*Notes on triangular sets and triangulation-decomposition algorithms I: polynomial systems*". In: Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 1–39. URL: https://doi.org/10.1007/3-540-45084-x_1 (cit. on p. 18).

[61] E. Hubert. "*Notes on triangular sets and triangulation-decomposition algorithms II: differential systems*". In: Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 40–87. URL: https://doi.org/10.1007/3-540-45084-x_2 (cit. on p. 18).

[62] E. Hubert and G. Labahn. "*Scaling invariants and symmetry reduction of dynamical systems*". In: Foundations of Computational Mathematics 13.4 (Aug. 2013), 479–516. URL: http://dx.doi.org/10.1007/s10208-013-9165-9 (cit. on p. 42).

[63] Z. Jelonek. "*On the effective Nullstellensatz*". In: Inventiones mathematicae 162.1 (Apr. 2005), 1–17. URL: http://dx.doi.org/10.1007/s00222-004-0434-8 (cit. on p. 11).

[64] I. Kaplansky. *An introduction to differential algebra*. Hermann, 1957 (cit. on p. 15).

[65] E. H. Kerner. "*Universal formats for nonlinear ordinary differential systems*". In: Journal of Mathematical Physics 22.7 (1981), pp. 1366–1371 (cit. on p. 45).

[66] M. Kinyon and A. Sagle. "*Quadratic dynamical systems and algebras*". In: Journal of Differential Equations 117.1 (Mar. 1995), 67–126. URL: http://dx.doi.org/10.1006/jdeq.1995.1049 (cit. on p. 45).

[67] E. R. Kolchin. *Differential algebra and algebraic groups*. Academic Press, 1971. ISBN: 978-0124157484 (cit. on p. 9).

[68] E. R. Kolchin. "*Extensions of differential fileds, I*". In: Annals of Mathematics 43.4 (), pp. 724–729. URL: https://www.jstor.org/stable/1968962 (cit. on p. 24).

[69] J. Kollár. "*Sharp effective Nullstellensatz*". In: Journal of the American Mathematical Society 1.4 (Oct. 1988), p. 963. URL: http://dx.doi.org/10.2307/1990996 (cit. on p. 11).

[70] M. Komatsu. *Estimate epidemiological parameters given partial observations based on algebraically observable PINNs*. 2024. URL: https://arxiv.org/abs/2407.12598 (cit. on pp. 9, 50).

[71] B. Kramer and K. Willcox. "*Balanced truncation model reduction for lifted nonlinear systems*". In: Realization and Model Reduction of Dynamical Systems: A Festschrift in Honor of the 70th Birthday of Thanos Antoulas. Ed. by C. Beattie, P. Benner, M. Embree, S. Gugercin, and S. Lefteriu. Cham: Springer International Publishing, 2022, pp. 157–174. ISBN: 978-3-030-95157-3. URL: https://doi.org/10.1007/978-3-030-95157-3_9 (cit. on p. 44).

[72] B. Kramer and K. Willcox. "*Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition*". In: AIAA Journal 57.6 (2019), pp. 2297–2307. URL: https://doi.org/10.2514/1.J057791 (cit. on p. 44).

[73] T. Krick, L. M. Pardo, and M. Sombra. "*Sharp estimates for the arithmetic Nullstellensatz*". In: Duke Mathematical Journal 109.3 (Sept. 2001). URL: http://dx.doi.org/10.1215/S0012-7094-01-10934-4 (cit. on p. 11).

[74] M. N. Lagutinskii. "*On the question about the simplest for of a system of ordinary differential equations*". In: Sbornik: Mathematics 27.4 (1911), pp. 420–423. URL: http://mi.mathnet.ru/sm6583 (cit. on pp. 44, 45).

[75] F. Lemaire and A. Ürgüplü. "*A method for semi-rectifying algebraic and differential systems using scaling type Lie point symmetries with linear algebra*". In: Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation. Vol. 2. ACM, 2010, 85–92. URL: http://dx.doi.org/10.1145/1837934.1837956 (cit. on p. 42).

[76] F. Lemaire and L. Roussel. "*Contribution to integral elimination*". Apr. 2024. URL: https://hal.science/hal-04570612 (cit. on p. 49).

[77] G. Li and H. Rabitz. "*A general analysis of exact lumping in chemical kinetics*". In: Chemical Engineering Science 44.6 (1989), pp. 1413–1430. URL: https://doi.org/10.1016/0009-2509(89)85014-6 (cit. on p. 39).

[78] W. Li, C.-M. Yuan, and X.-S. Gao. "*Sparse difference resultant*". In: Journal of Symbolic Computation 68 (2015), pp. 169–203. URL: https://doi.org/10.1016/j.jsc.2014.09.016 (cit. on p. 21).

[79] W. Li, C.-M. Yuan, and X.-S. Gao. "*Sparse differential resultant for Laurent differential polynomials*". In: Foundations of Computational Mathematics 15.2 (Feb. 2015), 451–517. URL: http://dx.doi.org/10.1007/s10208-015-9249-9 (cit. on pp. 19, 48).

[80] T. S. Ligon, F. Fröhlich, O. T. Chiş, J. R. Banga, E. Balsa-Canto, and J. Hasenauer. "*GenSSI 2.0: multi-experiment structural identifiability analysis of SBML models*". In: Bioinformatics 34.8 (Nov. 2017), 1421–1423. URL: http://dx.doi.org/10.1093/bioinformatics/btx735 (cit. on p. 33).

[81] L. Ljung and T. Glad. "*On global identifiability for arbitrary model parametrizations*". In: Automatica 30.2 (1994), pp. 265–276. URL: https://doi.org/10.1016/0005-1098(94)90029-9 (cit. on pp. 9, 29).

[82] R. S. Malik-Sheriff, M. Glont, T. V. N. Nguyen, K. Tiwari, M. G. Roberts, A. Xavier, M. T. Vu, J. Men, M. Maire, S. Kananathan, E. L. Fairbanks, J. P. Meyer, C. Arankalle, T. M. Varusai, V. Knight-Schrijver, L. Li, C. Dueñas-Roca, G. Dass, S. M. Keating, Y. M. Park, N. Buso, N. Rodriguez, M. Hucka, and H. Hermjakob. "*BioModels — 15 years of sharing computational models in life science*". In: Nucleic Acids Research 48.D1 (Jan. 2020), pp. D407–D415. ISSN: 0305-1048. URL: https://doi.org/10.1093/nar/gkz1055 (cit. on p. 41).

[83] G. Margaria, E. Riccomagno, M. J. Chappell, and H. P. Wynn. "*Differential algebra methods for the study of the structural identifiability of rational function state-space models in the biosciences*". In: Mathematical Biosciences 174.1 (Nov. 2001), 1–26. URL: http://dx.doi.org/10.1016/S0025-5564(01)00079-7 (cit. on p. 9).

[84] N. Meshkat, C. Kuo, and J. DiStefano. "*On finding and using identifiable parameter combinations in nonlinear dynamic systems biology models and COMBOS: a novel web implementation*". In: PLoS ONE 9.10 (2014), e110261. URL: http://dx.doi.org/10.1371/journal.pone.0110261 (cit. on pp. 29, 31, 33).

[85] H. Miao, X. Xia, A. S. Perelson, and H. Wu. "*On identifiability of nonlinear ODE models and applications in viral dynamics*". In: SIAM Review 53.1 (2011), pp. 3–39. URL: https://doi.org/10.1137/090757009 (cit. on p. 29).

[86] C. Moog, J. Perraud, P. Bentz, and Q. Vo. "*Prime differential ideals in nonlinear rational control systems*". In: Nonlinear Control Systems Design 1989. Elsevier, 1990, 17–21. ISBN: 9780080370224. URL: http://dx.doi.org/10.1016/B978-0-08-037022-4.50009-0 (cit. on p. 9).

[87] J. Müller-Quade and R. Steinwandt. "*Basic algorithms for rational function fields*". In: Journal of Symbolic Computation 27.2 (Feb. 1999), 143–170. URL: http://dx.doi.org/10.1006/jsco.1998.0246 (cit. on p. 34).

[88] Y. Ohyama. "*Differential relations of theta functions*". In: Osaka Journal of Mathematics 32.2 (1995), pp. 431–450. URL: https://projecteuclid.org:443/euclid.ojm/1200786061 (cit. on p. 25).

[89] F. Ollivier. "*Le problème de l'identifiabilité structurelle globale: approche th éorique, méthodes effectives et bornes de complexité*". PhD thesis. École polytechnique, 1990. URL: https://www.theses.fr/1990EPXX0009 (cit. on pp. 9, 29, 31).

[90] P. J. Olver and J Sivaloganathan. "*The structure of null Lagrangians*". In: Nonlinearity 1.2 (May 1988), 389–398. URL: http://dx.doi.org/10.1088/0951-7715/1/2/005 (cit. on p. 50).

[91] P. J. Olver. *Applications of Lie groups to differential equations*. Springer New York, 1986. ISBN: 9781468402742. URL: http://dx.doi.org/10.1007/978-1-4684-0274-2 (cit. on p. 9).

[92] P. J. Olver. "*Hyperjacobians, determinantal ideals and weak solutions to variational problems*". In: Proceedings of the Royal Society of Edinburgh: Section A Mathematics 95.3–4 (1983), 317–340. URL: http://dx.doi.org/10.1017/S0308210500013020 (cit. on p. 50).

[93] P. J. Olver and C. Shakiban. "*Dissipative decomposition of ordinary differential equations*". In: Proceedings of the Royal Society of Edinburgh: Section A Mathematics 109.3–4 (1988), 297–317. URL: http://dx.doi.org/10.1017/S0308210500027785 (cit. on p. 50).

[94] S. E. Otto, G. R. Macchio, and C. W. Rowley. "*Learning nonlinear projections for reduced-order modeling of dynamical systems using constrained autoencoders*". In: Chaos: An Interdisciplinary Journal of Nonlinear Science 33.11 (Nov. 2023). URL: http://dx.doi.org/10.1063/5.0169688 (cit. on p. 44).

[95] G. Pogudin. "*The primitive element theorem for differential fields with zero derivation on the base field*". In: Journal of Pure and Applied Algebra 219.9 (2015), pp. 4035–4041. URL: https://dx.doi.org/10.1016/j.jpaa.2015.02.004 (cit. on pp. 11, 25).

[96] G. Pogudin. *Differential algebra*. lecture notes. 2023. URL: https://www.lix.polytechnique.fr/Labo/Gleb.POGUDIN/files/da_notes.pdf (cit. on p. 15).

[97] G. Pogudin. "*Primary differential nil-algebras do exist*". In: Moscow University Mathematics Bulletin 69.1 (Jan. 2014), 33–36. URL: http://dx.doi.org/10.3103/S0027132214010069 (cit. on p. 11).

[98] G. Pogudin. "*Prime differential algebras and lie algebras associated to them*". in Russian. PhD thesis. Moscow State University, 2016. URL: http://mech.math.msu.su/~snark/files/diss/0091diss.pdf (cit. on p. 11).

[99] G. Pogudin. "*Wronskian of derivations*". In: Moscow University Mathematics Bulletin 66.1 (Feb. 2011), 47–49. URL: http://dx.doi.org/10.3103/S0027132211010104 (cit. on p. 11).

[100] P. Pohjanpalo. "*System identifiability based on the power series expansion of the solution*". In: Mathematical Biosciences 41.1–2 (1978), pp. 21–33. URL: http://dx.doi.org/10.1016/0025-5564(78)90063-9 (cit. on pp. 29, 33).

[101] M. van der Put and M. F. Singer. *Galois theory of linear differential equations*. Springer Berlin Heidelberg, 2003. ISBN: 9783642557507. URL: http://dx.doi.org/10.1007/978-3-642-55750-7 (cit. on p. 9).

[102] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox. "*Lift & learn: physics-informed machine learning for large-scale nonlinear dynamical systems.*" In: Physica D: Nonlinear Phenomena 406 (2020), p. 132401. URL: https://doi.org/10.1016/j.physd.2020.132401 (cit. on p. 46).

[103] X. Rey Barreiro and A. F. Villaverde. "*Benchmarking tools for a priori identifiability analysis*". In: Bioinformatics 39.2 (Jan. 2023). Ed. by J. Wren. URL: http://dx.doi.org/10.1093/bioinformatics/btad065 (cit. on p. 10).

[104] J. F. Ritt. *Differential algebra*. American Mathematical Society, 1950. ISBN: 978-0-8218-3205-9 (cit. on pp. 9, 16).

[105] J. F. Ritt. *Differential equations from the algebraic standpoint*. American Mathematical Society, 1932. ISBN: 978-0821846056. URL: https://archive.org/details/differentialequa033050mbp (cit. on pp. 9, 18, 19).

[106] D. Robertz. *Formal algorithmic elimination for PDEs*. Springer International Publishing, 2014. ISBN: 9783319114453. URL: http://dx.doi.org/10.1007/978-3-319-11445-3 (cit. on p. 18).

[107] L. A. Rubel and M. F. Singer. "*A differentially algebraic elimination theorem with application to analog computability in the calculus of variations*". In: Proceedings of the American Mathematical Society 94.4 (1985), 653–658. URL: http://dx.doi.org/10.1090/S0002-9939-1985-0792278-1 (cit. on p. 50).

[108] S. L. Rueda. "*Differential elimination by differential specialization of Sylvester style matrices*". In: Advances in Applied Mathematics 72 (Jan. 2016), 4–37. URL: http://dx.doi.org/10.1016/j.aam.2015.07.002 (cit. on p. 19).

[109] T. Sauer, T. Berry, D. Ebeigbe, M. M. Norton, A. J. Whalen, and S. J. Schiff. "*Identifiability of infection model parameters early in an epidemic*". In: SIAM Journal on Control and Optimization 60.2 (Nov. 2021), S27–S48. URL: http://dx.doi.org/10.1137/20M1353289 (cit. on p. 36).

[110] A. J. Schaft. "*On realization of nonlinear systems described by higher-order differential equations*". In: Mathematical Systems Theory 19.1 (Dec. 1986), 239–275. URL: http://dx.doi.org/10.1007/BF01704916 (cit. on p. 22).

[111] A. Sedoglavic. "*A probabilistic algorithm to test local algebraic observability in polynomial time*". In: Journal of Symbolic Computation 33.5 (2002), pp. 735–755. URL: http://dx.doi.org/10.1006/jsco.2002.0532 (cit. on p. 29).

[112] A. Seidenberg. "*Abstract differential algebra and the analytic case*". In: Proceedings of the American Mathematical Society 9.1 (1958), 159–164. URL: http://dx.doi.org/10.1090/S0002-9939-1958-0093655-0 (cit. on p. 16).

[113] A. Seidenberg. "*An elimination theory for differential algebra*". In: University of California publications in Mathematics. New Series III.2 (1956), pp. 31–66 (cit. on p. 18).

[114] W. Simmons and A. Platzer. *Differential elimination and algebraic invariants of polynomial dynamical systems*. 2023. URL: https://arxiv.org/abs/2301.10935 (cit. on p. 9).

[115] M. W. Sneddon, J. R. Faeder, and T. Emonet. "*Efficient modeling, simulation and coarse-graining of biological complexity with NFsim*". In: Nature Methods 8.2 (2010), pp. 177–183. URL: https://doi.org/10.1038/nmeth.1546 (cit. on p. 39).

[116] A. Sommese and C. Wampler. *The numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005. ISBN: 9789812567727 (cit. on p. 49).

[117] E. D. Sontag and Y. Wang. "*Input/output equations and realizability*". In: Realization and Modelling in System Theory. Birkhäuser Boston, 1990, 125–132. URL: http://dx.doi.org/10.1007/978-1-4612-3462-3_12 (cit. on p. 22).

[118] E. D. Sontag. *Mathematical control theory*. Springer New York, 1998. URL: https://doi.org/10.1007/978-1-4612-0577-7 (cit. on p. 22).

[119] H. J. Sussmann. "*Existence and uniqueness of minimal realizations of nonlinear systems*". In: Mathematical Systems Theory 10.1 (1976), pp. 263–284. URL: https://doi.org/10.1007/bf01683278 (cit. on p. 23).

[120] A. P. Tran, M. A. Al-Radhawi, I. Kareva, J. Wu, D. J. Waxman, and E. D. Sontag. "*Delicate balances in cancer chemotherapy: modeling immune recruitment and emergence of systemic drug resistance*". In: Frontiers in Immunology 11 (2020). URL: https://doi.org/10.3389/fimmu.2020.01376 (cit. on p. 10).

[121] N. Verdiere and C. Jauberthie. "*Parameter estimation procedure based on input-output integro-differential polynomials. application to the Hindmarsh-Rose model*". In: 2020 European Control Conference (ECC). IEEE, May 2020, 220–225. URL: http://dx.doi.org/10.23919/ECC51009.2020.9143670 (cit. on p. 49).

[122] A. F. Villaverde, N. D. Evans, M. J. Chappell, and J. R. Banga. "*Input-dependent structural identifiability of nonlinear systems*". In: IEEE Control Systems Letters 3.2 (Apr. 2019), 272–277. URL: http://dx.doi.org/10.1109/LCSYS.2018.2868608 (cit. on p. 32).

[123] D. Wang. "*EPSILON: a library of software tools for polynomial elimination*". In: Mathematical Software. 2002. URL: https://doi.org/10.1142/9789812777171_0040 (cit. on pp. 9, 18).

[124] Y. Wang and E. D. Sontag. "*Algebraic differential equations and rational control systems*". In: SIAM Journal on Control and Optimization 30.5 (Sept. 1992), 1126–1149. URL: http://dx.doi.org/10.1137/0330060 (cit. on p. 9).

[125] Y. Wang and E. D. Sontag. "*On two definitions of observation spaces*". In: Systems and Control Letters 13.4 (Nov. 1989), 279–289. URL: http://dx.doi.org/10.1016/0167-6911(89)90116-3 (cit. on p. 30).