Accelerating Gröbner Basis Computation via Weighted Ordering in Parameter Identifiability of ODE Models*

Mariya Bessonov[1], Ilia Ilmer[2], Tatiana Konstantinova[4], Alexey Ovchinnikov[2,3,4], Gleb Pogudin[5], and Pedro Soto[2]

[1] Department of Mathematics, CUNY NYC College of Technology, New York, NY, USA,
[2] Ph.D. Program in Computer Science, CUNY Graduate Center New York, NY, USA,
[3] Ph.D. Program in Mathematics, CUNY Graduate Center, New York, NY, USA,
[4] Department of Mathematics, CUNY Queens College, New York, NY, USA
[5] LIX, CNRS, École Polytechnique, Institute Polytechnique de Paris, Paris, France

Abstract. We consider a specific class of polynomial systems that arise in parameter identifiability problems of models of ordinary differential equations (ODE) and discover a method for speeding up the Gröbner basis computation by using a weighted ordering. Our method explores the structure of the ODE model to generate weight assignments for each variable. We provide empirical results that show improvement across different symbolic computing frameworks.

Keywords: F4 Algorithm, Weighted Monomial Ordering, Parameter Identifiability, ODE Systems, Dynamical Systems

1 Introduction

Structural identifiability is a significant factor in designing high-quality mathematical models of real-world phenomena with ordinary differential equations (ODEs). *Local* structural identifiability describes a scenario where a parameter has finitely many values. At the same time, *global* structural identifiability corresponds to a unique parameter value. In this work, we present a solution to computational challenges of computing Gröbner bases [3] for polynomial ideals that arise from parameter identifiability problems. Gröbner bases, for instance, form the foundation of global identifiability tool SIAN [13]. The computation itself is well-known to be heavy, despite such algorithms as F4 [7] and F5 [8]. We present an approach that uses additional domain-specific information about the

* The authors are grateful to CCiS at CUNY Queens College for the computational resources and to Andrew Brouwer for pointing out the HPV example. This work was partially supported by the National Science Foundation under grants CCF-1563942, CCF-1564132, DMS-1760448, DMS-1853650, and DMS-1853482.

polynomial ideals that arise in parameter identifiability problems to significantly speed up the process.

Gröbner basis for a given polynomial system varies based on the order of monomials. The most common ordering of monomials that is also empirically reliable in terms of computing time is the so-called *total-degree-reverse-lexicographic* order, or `tdeg` in MAPLE notation. We can impose an additional layer on top of the monomial ordering by adding *weights* to individual variables. In that case, we first compare variables by the weight value multiplied by variable's degree exponent and then ties are broken according to, for example, `tdeg`. In MAPLE, the most efficient version of F4 algorithm is the compiled C implementation. This implementation does not support weighted ordering defined via MAPLE's `wdeg` function. To remedy this, we use the degree substitution method, wherein assigning weight of w to x is equivalent to substituting a variable x with x^w in the polynomial system.

Consider the following motivating example showing benefits of weights in general:

$$P := \begin{cases} x_1^2x_3^4 + x_1x_2x_3^2x_5^2 + x_1x_2x_3x_4x_5x_7 + x_1x_2x_3x_4x_6x_8 + \\ +x_1x_2x_4^2x_6^2 + x_2^2x_4^4, \quad x_2^6, \quad x_1^6 \end{cases} \tag{1}$$

Computing the Gröbner basis of this system with `tdeg`-order of x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 takes approximately *670 seconds* of total CPU time and 26 seconds of total elapsed time as computed in Maple 2021.1 on MacBook Pro with 16 GB of RAM and 16-core M1 processor. Modifying the system by substituting $x_8 \to x_8^2$ results in approximately *2 seconds* of CPU time and only about *1 second* of total elapsed time.

Carefully chosen weights benefit the computation time. The main result in this talk is the rule for assigning weights in parameter identifiability problems based on the data from the input ODE models. Given a polynomial system obtained by SIAN from an input ODE model, our rule produces a weight assignment to the variables based on available data from the original ODE that improves the Gröbner basis computation.We will demonstrate experimental results showing decrease in runtime and memory use for MAPLE and Magma. This accelerates solving parameter identifiability problem. After Gröbner basis computation, the weights are removed by backward substitution to perform the final identifiability assessment.

2 Related Work

Gröbner basis is the key part of SIAN [13]. Buchberger [3] presented the original solution to finding the basis of polynomial systems. This method can be resource heavy [17] and depends on many factors, for example, selection strategy for polynomials [11]. The improvement of the algorithm itself came from the works of Faugère with F4 [7] and later F5 [8] algorithms. The key idea of F4 was to take advantage of mathematical tools from the realm of linear algebra. The work [1]

addresses the termination and complexity properties of the F5 algorithm. For an excellent taxonomy of F5-based approaches, we refer to [6].

The connection between weights and homogenization of ideals was studied in [9, 10] and references therein. In the mentioned works, weights were used as a homogenization tool since homogeneous ideals are a special case of inputs for Gröbner basis algorithms. However, in the example 1, a weighted ordering breaks homogenization, while still offering extraordinary benefits. Polynomial systems produced by SIAN contain non-homogeneous polynomials in most cases by the nature of the input data and we are able to generate weighted orders that lead to runtime and memory reduction.

The problem of parameter identifiability itself has been of great interest to the modeling and scientific computing community. Scientific Machine Learning (SciML)[6] recently included an implementation of the global identifiability tool from [5]. The identifiability package SIAN [13], considered in this work, has been recently applied in [4, 20, 21, 19]. The work [15] applies a web-based analyzer package [14] which combines SIAN and results of [18].

3 Main results

Let us begin by defining the identifiability software input representation.

Definition 1 (Model in the state-space form). A model in the *state-space* form is a system

$$\Sigma := \begin{cases} \dot{\mathbf{x}} & = \mathbf{f}(\mathbf{x},\ \boldsymbol{\mu},\ \mathbf{u}), \\ \mathbf{y} & = \mathbf{g}(\mathbf{x},\ \boldsymbol{\mu},\ \mathbf{u}), \\ \mathbf{x}(0) & = \mathbf{x}^*, \end{cases} \tag{2}$$

where $\mathbf{f} = (f_1, \ldots, f_n)$ and $\mathbf{g} = (g_1, \ldots, g_m)$ with $f_i = f_i(\mathbf{x},\ \boldsymbol{\mu},\ \mathbf{u})$, $g_i = g_i(\mathbf{x},\ \boldsymbol{\mu},\ \mathbf{u})$ are rational functions over the complex numbers $\mathbb{C}$. The vector $\mathbf{x} = (x_1, \ldots, x_n)$ represents the time-dependent state variables and $\dot{\mathbf{x}}$ represents the derivative.

The vectors $\mathbf{u} = (u_1, \ldots, u_s)$, $\mathbf{y} = (y_1, \ldots, y_m)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_\lambda)$, and $\mathbf{x}^* = (x_1^*, \ldots, x_n^*)$ represent the input variables, output variables, parameters, and initial conditions, respectively.

Briefly, SIAN [13] checks identifiability of parameters in (2) as follows:

1. The original differential system (2) is transformed into a polynomial system in the functions' derivatives by successive differentiation of the original equations.
2. Random values are sampled for $\mathbf{x}^*, \boldsymbol{\mu}$, and the corresponding values of the derivatives of y_i's are being computed and plugged into the polynomial system.

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

3. It is checked whether the sampled values $\mathbf{x}^*, \boldsymbol{\mu}$ are the only possible solutions of the specialized polynomial system. If yes, the corresponding parameter is globally identifiable.

A Gröbner basis is computed at the last step, typically the bottleneck of the algorithm. Our goal is to find a weight assignment to each variable of the specialized polynomial system to speed up the Gröbner basis computation.

Given a system (2), one can define *the Lie derivative* $\mathcal{L}(h)$ of a function $h \in \mathbb{C}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{u}, \mathbf{u}', \ldots)$ with respect to the system by

$$\mathcal{L}(h) = \sum_{i=1}^{n} f_i \frac{\partial h}{\partial x_i} + \sum_{j=1}^{s} \dot{u}_j \frac{\partial h}{\partial u_j}. \tag{3}$$

By applying this formula to each output function y_i, we can define, for each state variable or a parameter $a \in \{\mathbf{x}, \boldsymbol{\mu}\}$ a *level* as

$$\boxed{\text{Level}(a) := \min_i \left[\exists\, y_j \in \mathbf{y} \colon a \text{ appears in } \mathcal{L}^i(y_j)\right]}. \tag{4}$$

Using that value, we assign weight as follows:

– for a state variable $x_i \in x$ (and all its derivatives)

$$\text{Weight}(x_i) := \text{Level}(x_i) + 1;$$

– for a parameter $\mu_i \in \boldsymbol{\mu}$:

$$\text{Weight}(\mu_i) := \begin{cases} \text{Level}(\mu) + 1, \text{ if Level}(\mu_i) = \max\limits_{e \in \boldsymbol{\mu} \cup \mathbf{x}} \text{Level}(e), \\ 1, \text{ otherwise.} \end{cases}$$

We apply the weights on top of the default variable ordering that has proven itself to be empirically faster. We will consider several ODE models and provide Gröbner basis results over a field of integers with positive prime characteristic $p = 11863279$ as well as $p = 0$. Each example will be summarized by the following metrics:

1. Number of polynomials and variables in the polynomial system.
2. Default (without weights) CPU time (min) and memory (GB).
3. CPU time (min) and memory (GB) with weights.
4. Speedup calculated as $\frac{\text{default order time}}{\text{weighted order time}}$.
5. Memory improvement calculated as $\frac{\text{default order memory}}{\text{default order memory}}$.

References

[1] M. Bardet, J.-C. Faugère, and B. Salvy. “On the complexity of the F5 Gröbner basis algorithm”. In: *Journal of Symbolic Computation* 70 (2015), pp. 49–70.

[2] A. F. Brouwer, R. Meza, and M. C. Eisenberg. "Transmission heterogeneity and autoinoculation in a multisite infection model of HPV". In: *Mathematical Biosciences* 270 (2015), pp. 115–125.

[3] B. Buchberger. "A theoretical basis for the reduction of polynomials to canonical forms". In: *SIGSAM Bull.* 10.3 (1976), pp. 19–29. ISSN: 0163-5824. (Visited on 01/11/2021).

[4] E. Dankwa et al. "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.

[5] R. Dong et al. *Differential elimination for dynamical models via projections with applications to structural identifiability*. 2022.

[6] C. Eder and J.-C. Faugère. "A survey on signature-based algorithms for computing Gröbner bases". In: *Journal of Symbolic Computation* 80 (2017), pp. 719–784.

[7] J.-C. Faugère. "A new efficient algorithm for computing Gröbner bases (F4)". In: *Journal of Pure and Applied Algebra* 139.1 (1999), pp. 61–88.

[8] J.-C. Faugère. "A new efficient algorithm for computing Gröbner bases without reduction to zero (F5)". In: *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*. 2002, pp. 75–83.

[9] J.-C. Faugère, M. Safey El Din, and T. Verron. "On the complexity of computing Gröbner bases for quasi-homogeneous systems". In: *Proceedings of the 38th International Symposium on Symbolic and Algebraic Computation*. 2013, pp. 189–196.

[10] J.-C. Faugère, M. Safey El Din, and T. Verron. "On the complexity of computing Gröbner bases for weighted homogeneous systems". In: *Journal of Symbolic Computation* 76 (2016), pp. 107–141.

[11] A. Giovini et al. ""One sugar cube, please" or selection strategies in the Buchberger algorithm". In: *Proceedings of the 1991 international symposium on Symbolic and algebraic computation*. 1991, pp. 49–54.

[12] B. C. Goodwin. "Oscillatory behavior in enzymatic control processes". In: *Advances in enzyme regulation* 3 (1965), pp. 425–437.

[13] H. Hong et al. "SIAN: software for structural identifiability analysis of ODE models". In: *Bioinformatics* 35.16 (2019), pp. 2873–2874.

[14] I. Ilmer, A. Ovchinnikov, and G. Pogudin. "Web-based Structural Identifiability Analyzer". In: *Computational Methods in Systems Biology*. Vol. 12881. Lecture Notes in Computer Science. 2021, pp. 254–265.

[15] M. Locke et al. "Quantification of Type I Interferon Inhibition by Viral Proteins: Ebola Virus as a Case Study". In: *Viruses* 13.12 (2021), p. 2441.

[16] G. Massonis, J. R. Banga, and A. F. Villaverde. "Structural Identifiability and Observability of Compartmental Models of the COVID-19 Pandemic". In: *Annual Reviews in Control* 51 (2021), pp. 441–459.

[17] E. W. Mayr and A. R. Meyer. "The complexity of the word problems for commutative semigroups and polynomial ideals". In: *Advances in Mathematics* 46.3 (1982), pp. 305–329. ISSN: 0001-8708. (Visited on 01/05/2021).

[18] A. Ovchinnikov et al. "Computing all identifiable functions of parameters for ODE models". In: *Systems & Control Letters* 157 (2021), p. 105030.
[19] M. A. Al-Radhawi, M. Sadeghi, and E. Sontag. "Long-term regulation of prolonged epidemic outbreaks in large populations via adaptive control: a singular perturbation approach". In: *IEEE Control System Letters* (2021).
[20] A. Tran et al. "Delicate Balances in Cancer Chemotherapy: Modeling Immune Recruitment and Emergence of Systemic Drug Resistance". In: *Frontiers in Immunology* (2020).
[21] S. Zhang et al. "An integrated framework for building trustworthy data-driven epidemiological models: Application to the COVID-19 outbreak in New York City". In: *PLoS Computational Biology* 17.9 (2021).

A Experimental results

Model information			Time (min)				Memory (GB)			
Model name	num. polys.	num. vars.	alphabetical order	default order	weighted order	speedup	alphabetical order	default order	default weighted	reduction
COVID Model 2, [16, Ex. ID 37]	49	48	N/A	N/A	602.0	∞	N/A	N/A	23.2	∞
HPV, [2]	97	92	N/A	N/A	13.9	∞	N/A	N/A	3.7	∞
COVID Model 1, [16, Ex. ID 14]	51	50	377.0	321.9	1.0	327.6	15.3	15.2	0.3	52.6
Goodwin Oscillator [12]	42	43	44.1	29.8	1.5	18.9	10.8	10.6	0.7	14.6
SEIR-1, [16]	44	45	3.5	2.2	0.1	17.4	3.3	3.3	0.1	44.8

Table 1. Results of applying the weighted ordering to only Gröbner basis computation step of SIAN with positive characteristic $p = 11863279$ using Maple 2021.2. "N/A" stands for the following error message returned by Maple: "`Error, (in Groebner:-F4:-GroebnerBasis) numeric exception: division by zero`"

Model information			Time (min)				Memory (GB)			
Model name	num. polys.	num. vars.	alphabetical order	default order	default weighted	speedup	alphabetical order	default order	default weighted	reduction
COVID Model 2, [16, Ex. ID 37]	49	48	4000.6	3471.2	517.4	6.7	38.6	36.4	21.6	1.7
HPV, [2]	97	92	321.7	126.6	51.5	2.4	21.4	9.8	18.6	0.5
COVID Model 1, [16, Ex. ID 14]	51	50	1331.1	1272.1	0.6	2207.9	9.2	8.7	1.8	4.8
Goodwin Oscillator [12]	42	43	26.9	22.4	0.8	28.5	3.5	3.1	0.5	6.0
SEIR-1, [16]	44	45	8.6	3.9	0.1	76.0	2.0	2.0	0.2	9.8

Table 2. Results of applying the weighted ordering to only Gröbner basis computation step of SIAN with positive characteristic $p = 11863279$ in Magma 2.26-8.