Structured linear algebra for recurrences and Gröbner bases

Ph.D. subject in Computer Algebra

Équipe PolSys, LIP6, Sorbonne Université, 4 place Jussieu, 75005 Paris, France

Ph.D. Advisor. Jérémy Berthomieu[1]

Context, scientific positioning

The main topic of this Ph.D. subject is to **exploit the link between polynomials and linear recurrence relations of a sequence** in order to accelerate either the computations of Gröbner bases for polynomial system solving or the guessing of all the recurrences satisfied by a multi-indexed sequence. To do so, we aim to **thoroughly analyze and take advantage of the structure of the linear systems** that appear in these computations.

On the one hand, polynomial systems arise in a wide range of areas of scientific domains such as biology [18], chemistry [26], quantum mechanics [37], robotics [38], and computing sciences, including coding theory [39], computer vision [28] and cryptography [22] to cite a few. On the other hand, polynomial system solving is NP-hard, even when the ground field is finite [25, Appendix A7.2]. Moreover, the non-linearity of such systems make reliability issues topical, in particular when complete and exhaustive outputs are required, in the context of numerical algorithms.

Likewise, sequences are a classical mathematical object and computing linear recurrence relations of a multi-indexed sequence or determining the nature of this sequence based on these relations is a fundamental problem in coding theory [27, 39], computer algebra [23, 43, 44] and enumerative combinatorics [13, 14, 15].

Whether it be for solving polynomial systems or for computing or *guessing* linear recurrence relations, one aims to obtain nice generators of an ideal that are able to answer the following questions. Is the number of solutions finite in an algebraic closure of the field of coefficients? How many initial terms and linear recurrence relations do one needs to compute any term of the sequence?

These questions are easily answered when we have a *Gröbner basis* of the ideal at hand.

State of the art. Buchberger developed the theory of Gröbner bases and designed a first algorithm [16] to compute them.

While lexicographic Gröbner bases are the tool of choice to represent the solution set of a polynomial system, often, they are the hardest Gröbner bases to compute. For n generic polynomials of degree d in n polynomials, computing the $\prec_{\text{LEX}}$-Gröbner basis of the ideal they spanned is bounded by $C_1 d^{C_2 n^3}$, see [17]. This is to compare with computing the $\prec_{\text{DRL}}$-Gröbner basis of the same ideal, where $\prec_{\text{DRL}}$ is a monomial order that filters monomials first by degree, which is bounded by $C_1 d^{C_2 n^2}$, see [29].

Since then, many efficient algorithms have been developed to calculate $\prec_{\text{DRL}}$-Gröbner bases, such as Faugère's F_4 [19] and F_5 [20] algorithms.

Since $\prec_{\text{LEX}}$-Gröbner bases are needed for polynomial system solving, as a caveat to their computation cost, Gröbner bases change of orders algorithms have been introduced. They take as an input a Gröbner basis $\mathcal{G}_1$ for a monomial order $\prec_1$ and another monomial order $\prec_2$ and they return $\mathcal{G}_2$, a Gröbner basis of $\langle \mathcal{G}_1 \rangle$ for $\prec_2$. This yields the following framework, in the zero-dimensional case, where $f_1, \ldots, f_s$ are the original polynomials that are given as an input to Buchberger's algorithm [16] or to Faugère's F_4 [19] or F_5 [20] algorithms to compute the $\prec_{\text{DRL}}$-Gröbner basis $\mathcal{G}_{\text{DRL}}$. Then, $\mathcal{G}_{\text{DRL}}$ is converted into the $\prec_{\text{LEX}}$-Gröbner basis $\mathcal{G}_{\text{LEX}}$ using the so-called FGLM algorithm [21] or faster variants like the Sparse-FGLM algorithm [23, 24] or more recently [36] and then [9]. This framework allows one to compute the $\prec_{\text{LEX}}$-Gröbner basis in $C_1 d^{C_2 n^2}$ operations as well.

[1] Sorbonne Université, jeremy.berthomieu@lip6.fr,
https://www-polsys.lip6.fr/~berthomieu
Habilitation defense planned on 21st September 2023

The main idea of Faugère and Mou's [23, 24] algorithm is to build a linear recurrent n-indexed sequence $\mathbf{u}$ using the Gröbner basis at hand, namely $\mathcal{G}_{\text{DRL}}$, and then to compute the smallest linear recurrence relations for $\prec_{\text{LEX}}$ in order to recover $\mathcal{G}_{\text{LEX}}$.

On the other hand, many algorithms have been developed to guess linear recurrence relations for a multi-indexed sequence. Regarding relations with constant coefficients, such as $u_{i+1,j+1} - u_{i,j+1} - u_{i,j} = 0$, we can mention Sakata's algorithm [40, 41, 42], based on adding polynomials or multiplying a polynomial with a monomial, Scalar-FGLM, based on linear algebra techniques, see [4, 5, 10], or Artinian Gorenstein border bases, based on a Gram-Schmidt process, see [34]. Finally, in [6, 8], the authors proposed an algorithm that extends both Sakata's and Scalar-FGLM, based on polynomial divisions. While its complexity is still not satisfactory, this polynomial division viewpoint is a key ingredient to the quasi-linear complexity of the guessing for uni-indexed sequences.

Likewise, algorithms were developed for relations with polynomial (in the indices) coefficients, such as $(i+1-j)u_{i+1,j} - (i+1)u_{i,j} = 0$. For instance, Beckermann-Labahn [2] for uni-indexed sequences or Naldi-Neiger [35] for multi-indexed sequences, which both fall in the more general category of divide-and-conquer algorithms for Padé-Hermite approximants. An extension of Scalar-FGLM is designed in [7] while a hybrid approach based on Gröbner bases computation, for quasi-commutative polynomials in an Ore algebra, is proposed in [7, 10].

Ph.D. Objectives

The main objective of this Ph.D. thesis is the design of fast algorithms for guessing linear recurrence relations, whether with constant or polynomial coefficients, in order to accelerate polynomial system solving or the description of the nature of the sequences coming from applications, such as combinatorics. This global goal will be decomposed into three ambitious objectives, each of which we envision taking about a year of the Ph.D.

Year 1: Polynomial divisions for recurrences with polynomial coefficients. Following the paradigm shift from a linear algebra viewpoint in Sakata's algorithm [40, 41, 42] and in Scalar-FGLM [4, 5] to a multivariate polynomial one in [6, 8] for relations with constant coefficients, the goal is to guess linear recurrence relations with polynomial coefficients using polynomial arithmetic instead of linear algebra [7]. From the generating series of a sequence and its derivatives, or more precisely the mirror polynomials of a truncation of these series, the goal is to find algebraic combinations thereof which are *small* modulo the monomial ideal $\langle x_1^{D_1}, \dots, x_n^{D_n} \rangle$, where $D_1, \dots, D_n$ depend on the sequence terms we allow ourselves to use.

Efficiency of guessing algorithms is based on two aspects: the number of performed operations and the number of sequence terms that are needed. Indeed, in many applications, computing the sequence is the bottleneck. Thus, to make this approach the most efficient, we shall closely look at the number of different sequences terms that are needed to correctly guess the relations. Furthermore, to optimize the number of operations, we shall rely on efficient algorithms for univariate polynomials and uni-indexed sequences. The main goal is to reach a **complexity at most quadratic in the size of the output** instead of only cubic.

These guessing algorithms may find fake relations, this happens in general when too few terms are used or when most of the terms are 0. This can be circumvent by *structured* guessing relations using mainly the nonzero sequence terms. We will pay attention to these bad sequences so that our new algorithm avoids these fake relations as much as possible.

Year 2: Guessing recurrences with constant coefficients for solving polynomial systems with multiplicities. Generic polynomial systems, i.e. those given by n generic polynomials of degree d in n variables, satisfy *two* properties allowing us to speed the change of order up. The first one is a *shape position* property satisfied by their $\prec_{\text{LEX}}$-Gröbner basis. It means that the sought $\prec_{\text{LEX}}$-Gröbner basis is of type

$$g_n(x_n), x_{n-1} - g_{n-1}(x_n), \dots, x_1 - g_1(x_n), \deg g_n = d^n = D$$

and this ensures that FGLM algorithms, such as [9, 23, 24] only need one $D \times D$-matrix, which is highly structured. For instance, it has $O(tD)$ nonzero coefficients and asymptotics of t are given in [24] for generic systems and in [3] for generic determinantal systems. The second property is a *stable* one and is satisfied by their $\prec_{\text{DRL}}$-Gröbner basis. Its main consequence is that the aforementioned matrix is computed for free [33].

In some situations where the system has roots with multiplicities, the sought $\prec_{\text{LEX}}$-Gröbner basis *cannot* be in shape position, even after a generic linear change of variables. Such a system is called 2-*thick* in [1] and it requires a second structured $D \times D$-matrix, of a similar kind, to be computed: it has $O(\tau D)$ non zero coefficients. Furthermore, in most situations the stability property is still satisfied which means that the computation of this second matrix is cheap. However, it does not ensure (actually it *almost never can*) that this second matrix is computed for free.

A first goal is to derive a sharp complexity estimate on the computation of this second matrix based on the $\prec_{\text{DRL}}$-Gröbner basis, exploiting the stability property and the work of Moreno-Socías [33], for the first matrix. As a by-product, we will obtain complexity bounds on the computation of the sequence terms that appear in the multi-Hankel matrix built by Scalar-FGLM [4, 5] to recover the $\prec_{\text{LEX}}$-Gröbner basis. Then, as a second goal, we will rely on the *quasi-Hankel* structure of this multi-Hankel matrix, and fast algorithms for quasi-Hankel matrices [11, 12], to analyze the complexity on the computation of the sought $\prec_{\text{LEX}}$-Gröbner basis. All in all, we will have a **complete description and complexity estimate of Faugère and Mou's [23, 24] Sparse-FGLM algorithm for generic 2-thick systems**.

Year 3: Quasi-commutative Gröbner bases computations. As stated before, more often than not, the bottleneck of guessing algorithms is the computation of the sequence terms that are needed for the guessing. Furthermore, the larger the sought relation, the larger the number of required sequence terms. Yet, the polynomials in a Gröbner basis are far from independent: generically the larger ones are algebraic combinations of the smallest ones. This is the key ingredient of the hybrid approach based on Gröbner bases computation to discover large relations without extra queries to the sequence. These Gröbner bases computations differ from the above topics as they are in $2n$ variables $x_1, \ldots, x_n, \delta_1, \ldots, \delta_n$ satisfying quasi-commutative properties such as $\delta_k x_k = x_k(\delta_k + 1)$. These commutation rules come from the fact that shifting the kth index of a sequence by one (encoded by x_k) and multiplying the sequence term by the kth index (encoded by δ_k) do not commute. Similar commutation rules may represent recurrence relations with *giant steps* such as $u_{2i,j} - 2u_{i,j} = 0$.

Following [30] and the generalization of Buchberger's criteria, the goal will be to dive into the understanding of how Faugère's F_4 algorithm [19] behaves or can be extended from the commutative setting to this one. To do so, we shall study the module of trivial syzygies, and in particular what kind of information the commutation rules provide on the syzygies, in order to get information on the sizes of the matrices that are built in F_4.

Hilbert series and Hilbert polynomials are powerful tools that allow one to understand the complexity of computing Gröbner bases. In the commutative case, one can derive a bound on the degree of the polynomials in a reduced Gröbner basis for a total degree order, thanks to them, see [32, Section 4.5, Corollary]. We will investigate how knowing in advance the Hilbert series can speed the Gröbner bases computations up, or how together with the Hilbert polynomials, they can give us a bound on the degrees of the polynomials in the reduced $\prec_{\text{DRL}}$-Gröbner basis.

The main **ambitious goal is to bring the complexity of $\prec_{\text{DRL}}$-Gröbner bases for quasi-commutative polynomials to that of classical polynomials**. This will be a first step for the computation of the *contiguity matrices* in particle physics and algebraic statistics, see [31], which are analogues of the matrices used in the previous section.

References

[1] E. Becker, T. Mora, M. G. Marinari, and C. Traverso, "The Shape of the Shape Lemma", in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, ISSAC '94, Oxford, United Kingdom: ACM, 1994, pp. 129–133, DOI: 10.1145/190347.190382.

[2] B. Beckermann and G. Labahn, "A Uniform Approach for the Fast Computation of Matrix-Type Pade Approximants", in *SIAM J. Matrix Anal. Appl.* 15.3 (1994), pp. 804–823, DOI: 10/dxv54n.

[3] J. Berthomieu, A. Bostan, A. Ferguson, and M. Safey El Din, "Gröbner bases and critical values: The asymptotic combinatorics of determinantal systems", in *J. Algebra* 602 (2022), pp. 154–180, DOI: 10.1016/j.jalgebra.2022.03.002.

[4] J. Berthomieu, B. Boyer, and J.-C. Faugère, "Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences", in *Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '15, Bath, United Kingdom: ACM, 2015, pp. 61–68, DOI: 10/gdnj.

[5] J. Berthomieu, B. Boyer, and J.-C. Faugère, "Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences", in *J. Symbolic Comput.* 83.Supplement C (2017), Special issue on the conference ISSAC 2015: Symbolic computation and computer algebra, pp. 36–67, DOI: 10/gkx4.

[6] J. Berthomieu and J.-C. Faugère, "A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations", in *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, ISSAC '18, New York, USA: ACM, 2018, pp. 79–86, DOI: 10/gkx5.

[7] J. Berthomieu and J.-C. Faugère, "Guessing Linear Recurrence Relations of Sequence Tuples and P-Recursive Sequences with Linear Algebra", in *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '16, Waterloo, ON, Canada: ACM, 2016, pp. 95–102, DOI: 10/ggck6q.

[8] J. Berthomieu and J.-C. Faugère, "Polynomial-division-based algorithms for computing linear recurrence relations", in *J. Symbolic Comput.* 109 (2022), pp. 1–30, DOI: 10/gs6r.

[9] J. Berthomieu, V. Neiger, and M. Safey El Din, "Faster Change of Order Algorithm for Gröbner Bases under Shape and Stability Assumptions", in *Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, ISSAC '22, Villeneuve-d'Ascq, France: ACM, 2022, pp. 409–418, DOI: 10.1145/3476446.3535484.

[10] J. Berthomieu and M. Safey El Din, "Guessing Gröbner bases of structured ideals of relations of sequences", in *J. Symbolic Comput.* 111 (2022), pp. 1–26, DOI: 10.1016/j.jsc.2021.11.001.

[11] A. Bostan, C.-P. Jeannerod, C. Mouilleron, and É. Schost, “On Matrices With Displacement Structure: Generalized Operators and Faster Algorithms”, in *SIAM Journal on Matrix Analysis and Applications* 38.3 (2017), pp. 733–775, DOI: 10.1137/16M1062855.

[12] A. Bostan, C.-P. Jeannerod, and É. Schost, “Solving Toeplitz- and Vandermonde-like Linear Systems with Large Displacement Rank”, in *ISSAC ’07*, ed. by C. W. Brown, ACM, 2007, pp. 33–40, DOI: 10.1145/1277548.1277554.

[13] A. Bostan, K. Raschel, and B. Salvy, “Non-D-finite excursions in the quarter plane”, in *J. Combin. Theory Ser. A* 121 (2014), pp. 45–63, DOI: 10/gs6s.

[14] M. Bousquet-Mélou and M. Mishna, “Walks with small steps in the quarter plane”, in *Algorithmic probability and combinatorics*, vol. 520, Contemp. Math. Providence, RI: Amer. Math. Soc., 2010, pp. 1–39, DOI: 10.1090/conm/520/10252.

[15] M. Bousquet-Mélou and M. Petkovšek, “Walks confined in a quadrant are not always D-finite”, in *Theoret. Comput. Sci.* 307.2 (2003), Random Generation of Combinatorial Objects and Bijective Combinatorics, pp. 257–276, DOI: 10.1016/S0304-3975(03)00219-6.

[16] B. Buchberger, “Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal”, PhD thesis, University of Innsbruck, 1965.

[17] L. Caniglia, A. Galligo, and J. Heintz, “Some New Effectivity Bounds in Computational Geometry”, in *Proceedings of the 6th International Conference, on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, AAECC-6, Berlin, Heidelberg: Springer-Verlag, 1988, pp. 131–151.

[18] C. Conradi, E. Feliu, M. Mincheva, and C. Wiuf, “Identifying parameter regions for multistationarity”, in *PLOS Computational Biology* 13.10 (2017), e1005751, DOI: 10.1371/journal.pcbi.1005751.

[19] J.-C. Faugère, “A New Efficient Algorithm for Computing Gröbner bases (F4)”, in *J. Pure Appl. Algebra* 139.1 (1999), pp. 61–88, DOI: 10/bpq5dx.

[20] 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*, ISSAC ’02, Lille, France: ACM, 2002, pp. 75–83, DOI: 10/bd4nnq.

[21] J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora, “Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering”, in *J. Symbolic Comput.* 16.4 (1993), pp. 329–344, DOI: 10/dv8t4j.

[22] J.-C. Faugère and A. Joux, “Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases”, en, in *Advances in Cryptology - CRYPTO 2003*, ed. by D. Boneh, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2003, pp. 44–60, DOI: 10/fpfzgc.

[23] J.-C. Faugère and C. Mou, “Fast Algorithm for Change of Ordering of Zero-dimensional Gröbner Bases with Sparse Multiplication Matrices”, in *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, ISSAC ’11, San Jose, USA: ACM, 2011, pp. 115–122, DOI: 10/fhhs56.

[24] J.-C. Faugère and C. Mou, “Sparse FGLM algorithms”, in *J. Symbolic Comput.* 80.3 (2017), pp. 538–569, DOI: 10/gfz47c.

[25] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, USA: W. H. Freeman & Co., 1990.

[26] K. Gatermann and B. Huber, “A Family of Sparse Polynomial Systems Arising in Chemical Reaction Systems”, in *J. Symbolic Comput.* 33.3 (2002), pp. 275–305, DOI: 10.1006/jsco.2001.0512.

[27] A. Hocquenghem, “Codes correcteurs d’erreurs”, in *Chiffres* 2 (1959), pp. 147–156.

[28] Y. Kuang, Y. Zheng, and K. øAström, “Partial Symmetry in Polynomial Systems and Its Applications in Computer Vision”, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 438–445, DOI: 10.1109/CVPR.2014.63.

[29] D. Lazard, “Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations”, in *Computer algebra (London, 1983)*, vol. 162, Lecture Notes in Comput. Sci. Berlin: Springer, 1983, pp. 146–156.

[30] V. Levandovskyy, “Non-commutative Computer Algebra for polynomial algebras: Gröbner bases, applications and implementation”, PhD thesis, Technische Universität Kaiserslautern, 2005.

[31] S.-J. Matsubara-Heo and S. Telen, *Twisted cohomology and likelihood ideals*, 2023, arXiv: 2301.13579 [math.AG].

[32] H. M. Möller and F. Mora, “Upper and lower bounds for the degree of Groebner bases”, in *EUROSAM 84*, ed. by J. Fitch, Berlin, Heidelberg: Springer, 1984, pp. 172–183.

[33] G. Moreno-Socías, “Degrevlex Gröbner bases of generic complete intersections”, in *J. Pure Appl. Algebra* 180.3 (2003), pp. 263–283.

[34] B. Mourrain, “Fast Algorithm for Border Bases of Artinian Gorenstein Algebras”, in *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC ’17, Kaiserslautern, Germany: ACM, 2017, pp. 333–340, DOI: 10.1145/3087604.3087632.

[35] S. Naldi and V. Neiger, “A Divide-and-Conquer Algorithm for Computing Gröbner Bases of Syzygies in Finite Dimension”, in *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*, ISSAC ’20, Kalamata, Greece: ACM, 2020, pp. 380–387, DOI: 10/gs6t.

[36] V. Neiger and É. Schost, “Computing syzygies in finite dimension using fast linear algebra”, in *J. Complexity* 60 (2020), p. 101502, DOI: 10/ht25.

[37] S. Odake and R. Sasaki, “Exactly solvable quantum mechanics and infinite families of multi-indexed orthogonal polynomials”, in *Physics Letters B* 702.2 (2011), pp. 164–170, DOI: 10.1016/j.physletb.2011.06.075.

[38] M. Raghavan and B. Roth, “Solving Polynomial Systems for the Kinematic Analysis and Synthesis of Mechanisms and Robot Manipulators”, in *Journal of Mechanical Design* 117.B (1995), pp. 71–79, DOI: 10.1115/1.2836473.

[39] S. Sakata, “N-dimensional Berlekamp-Massey algorithm for multiple arrays and construction of multivariate polynomials with preassigned zeros”, in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, ed. by T. Mora, vol. 357, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 1989, pp. 356–376, DOI: cmbzng.

[40] S. Sakata, “Extension of the Berlekamp-Massey Algorithm to N Dimensions”, in *Inform. and Comput.* 84.2 (1990), pp. 207–239, DOI: b3m55k.

[41] S. Sakata, “Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array”, in *J. Symbolic Comput.* 5.3 (1988), pp. 321–337, DOI: fvbmvf.

[42] S. Sakata, “The BMS Algorithm”, in *Gröbner Bases, Coding, and Cryptography*, ed. by M. Sala, S. Sakata, T. Mora, C. Traverso, and L. Perret, Berlin, Heidelberg: Springer, 2009, pp. 143–163, DOI: 10/ft76mf.

[43] B. Salvy, “Linear Differential Equations as a Data Structure”, in *Found. Comput. Math.* 19 (5 2019), pp. 1071–1112, DOI: 10/gfs3ww.

[44] T. Sauer, “Prony’s method in several variables: Symbolic solutions by universal interpolation”, in *J. Symbolic Comput.* 84 (2018), pp. 95–112, DOI: 10/j5m8.