Towards a new generation of Gröbner bases algorithms for polynomial system solving

Abstract

Polynomial systems arise in a wide range of topical applications related to computer communication and security, quantum physics, mechanism design and optimization.

This project aims at the significant improvement – by several orders of magnitude – of algorithms for solving polynomial systems of equations. These algorithms rewrite the input set of equations into an equivalent one, named Gröbner basis, from which it is easier to extract the solutions.

The key change of paradigm that will enable such improvements is to perform reductions of Gröbner bases computations to linear algebra operations over matrices with univariate polynomials as entries, and then benefit from advanced computer algebra algorithms for these operations.

1 Supervision

This doctoral project will be supervised by Mohab Safey El Din.

The PhD candidate will be hosted by the PolSys team, at LIP6 (UMR CNRS 7606) whose research activities focus on mathematical algorithms and software through computer algebra, a.k.a. symbolic computation, methods. It focuses on solving polynomial systems of equations and inequalities in a broad sense and its applications. It develops the reference library `msolve` for solving polynomial systems and computing Gröbner bases. The PhD applicant will enjoy a nice atmosphere and international working environment.

2 General context

Polynomial systems of equations and inequalities encode non-linear algebraic and arithmetic relations such as the distance between points in the Euclidean space, linearity relations, tensor computations, etc. The whole Euclidean geometry, many arithmetic, physical and biological properties can be modeled with polynomial, non-linear, constraints. Solving systems of polynomial constraints is an upstream scientific research problem which arises in topical issues and key areas related to computer and communication security, quantum physics, robotics modeling and analysis, to cite a few.

Solving systems of polynomial constraints is therefore an important, but a difficult, computational problem, actually, amongst the most difficult ones since it is $\mathcal{NP}$-hard. Also, polynomial systems with as many equations as variables have in the worst case a number of solutions which is exponential in the dimension of the ambient space, this bound being reached on generic instances. This hardness indicates

that in order to solve efficiently polynomial systems, algorithms must be designed with a lot of care, by controlling the number of operations they perform, i.e. the complexity of these algorithms.

The issue of controlling and understanding the complexity of algorithms for solving polynomial systems is, on its own, a crucial issue for topical applications. Indeed, the hardness of polynomial system solving is used to design cryptociphers and signature schemes in post-quantum cryptography through multivariate cryptography. Actually, a general methodology, named algebraic cryptanalysis, has been designed to assess the security of cryptociphers through polynomial system solving. In this context, polynomial systems of equations which arise have coefficients in finite fields and are usually overdetermined, i.e. they do have more equations than variables.

Polynomial systems with coefficients in the field of rational numbers also arise in a number of applications in engineering sciences such as robotics, where the design and analysis of a new class of robots, called soft robots, which enjoy deformation properties, brings new challenges to polynomial system solving. These systems need then to be solved over the real numbers and yield parameters, hence leading to difficult problems in the area of effective real algebraic geometry.

These important illustrative applications emphasize the need of "algebraic" a.k.a. symbolic algorithms for solving polynomial systems: finite fields as well as parameters cannot be handled with numerical algorithms naturally. More generally, the non-linearity of polynomial systems, which may lead to numerical issues, combined with the need of some applications to compute solutions exhaustively to polynomial systems makes computer algebra methods particularly well-suited and relevant to polynomial system solving.

Computer algebra algorithms rewrite the input system of polynomial equations, from which it is usually hard to extract the solutions, to polynomial systems which are equivalent (they do have exactly the same solutions) but from which it is easy to extract the solutions. This is typically the fundamental framework of Gaussian elimination for solving linear systems of equations, the easiest polynomial systems to solve. Hence, as does Gaussian elimination, computer algebra algorithms, taking as input a polynomial system in n variables, say $x_1, \ldots, x_n$, will rewrite this system into an equivalent one which is triangular as follows:

$$G_c(x_c, \ldots, x_n), G_{c-1}(x_{c-1}, \ldots, x_n), \ldots, G_1(x_1, \ldots, x_n)$$

where the G_i's are families of non-zero polynomials depending on the variables $x_c, \ldots, x_n$ and the subsystem $G_c = \cdots = G_k = 0$ describe the smallest closed set containing the projection of the solutions to the input system on the $(x_k, \ldots, x_n)$-coordinate subspace. This latter property, relating the projection of solution sets with a triangular rewriting through the elimination of variables is referred to as an "elimination property" further.

Note that, as done implicitly by Gaussian elimination, to achieve this triangular representation, one uses a lexicographical ordering $x_1 > x_2 > \cdots\cdots > x_n$ on the monomials. The ability to sort monomials with well-defined orderings is actually a fundamental property that will turn to be a cornerstone of the algorithmic framework of computer algebra algorithms for polynomial system solving.

Note also that when $c = n$, G_n becomes a family of univariate polynomials whose set of common solutions is exactly the one of the gcd of these polynomials. The triangular rewriting combined with the elimination property then shows the solution set is finite (when all the G_i's are non-empty). In this setting, one says that the solution set has dimension $d = n - c = 0$ (by convention, empty sets have dimension -1). The integer c is the codimension. More generally, again when all the G_i's are non-empty, the integer $d = n - c$ is the dimension of the solution set (number of degrees of freedom to move on this set) ; c is called the codimension.

Applications in post-quantum cryptography usually yield polynomial systems of dimension at most 0, whereas applications we aforementioned in robotics involve polynomial systems of positive dimension.

In this project, we will consider both kinds of polynomial systems of equations.

Hence, all in all, there is a need of efficient algorithms which on input polynomial systems of equations can identify the dimension of their solution set and rewrite input systems into equivalent ones which are triangular and enjoy the elimination property.

Gröbner bases computation and theory provide a versatile tool for polynomial system solving through the above machinery. Gröbner bases have the advantage to provide convenient various rewriting of input polynomial equations, depending on some monomial ordering which is specified as input. This includes triangular forms with the elimination property as described above when the aforementioned lexicographical monomial ordering is specified. Initially, Gröbner bases have been invented to solve the longstanding *ideal membership problem* in polynomial rings: this problems asks whether a given polynomial f can be rewritten as an algebraic combination of given polynomials $f_1, \ldots, f_s$ (the ideal generated by $f_1, \ldots, f_s$ the set of algebraic combinations of the f_i's). Buchberger's algorithm, named after its author [4], solves this problem. This project aims at providing significant improvements to Gröbner bases algorithms for solving systems of polynomial equations and leverage these improvements to applications in post-quantum cryptography on the one hand as well as robotics problems and effective real algebraic geometry on the other hand.

Before describing in detail our objectives, we start by recalling the current state of the art. We then introduce the main ingredients of the change of paradigms we will perform to achieve our goals.

3 State of the art and problem statements

Buchberger's algorithm computes iteratively polynomials (by reducing so-called S-polynomials of critical pairs w.r.t. the current basis) in the ideal generated by the input equations whose leading monomial (w.r.t. the some admissible monomial ordering) does not belong to the ideal generated by the leading monomials of the current basis. This algorithmic scheme has been extensively studied and developed until late in the 90's but was still suffering from several drawbacks, namely, *(a)* the unability to provide an advantageous strategy to choose the critical pairs in a way that optimizes the computations and *(b)* an important number of such critical pairs which lead to compute useless information (basically 0 is in the input ideal).

In 1998, Faugère proposed the so-called F4 algorithm [5] which makes the connection between Buchberger's algorithmic framework and a framework reducing to linear algebra operations (row echelonisation of so-called Macaulay matrices) algebraic elimination problems due to F. Macaulay, tackling problem *(a)*. This has led to impressive practical speed-up with software implementations of this algorithm tackling polynomial systems which were out of reach previously. Another major step has been taken with the F5 algorithm, still due to Faugère [6], which tackles problem *(b)* by showing how to obtain in F4 full rank matrices (hence removing all reductions to 0) in generic cases, but at the cost of removing some freedom in the algorithm design. The core idea to avoid these reductions to 0 is to add to F4 an additional layer which is a compact representation of the generic module of syzygies of polynomial ideals (under some genericity assumption). This algorithm has been used to tackle challenges in e.g. cryptography (see e.g. [7]).

Matrices which appear there, which are called Macaulay matrices, are sparse and linear algebra techniques which are used take advantage of this feature. Given $(f_1, \ldots, f_s) \subset R = \mathbb{K}[x_1, \ldots, x_n]$ (where $\mathbb{K}$ is the base field of coefficients) and $D \in \mathbb{N}$, the rows of these matrices encode the vector subspaces E_D of the set $R_{\leqslant D}$ of polynomials of degree $\leqslant D$ containing the polynomials mf_i where $1 \leqslant i \leqslant s$ and m ranges over all monomial such that $\deg(m) + \deg(f_i) \leqslant D$. Generically, these vector subspaces E_D coincide with those generated by the polynomials mf_i as we just defined.

In the worst case, computing a Gröbner basis is an extremely difficult task. Indeed, with input polynomials of degree bounded D and involving n variables, this computation can be doubly exponential in n (but still polynomial in D). In general, one cannot expect much better since it is proved that, again in the worst case, a Gröbner basis can contain $D^{2^{O(n)}}$ elements.

In the generic setting, but also in the case of homogeneous polynomial sequences, Lazard has proved that a Gröbner basis of the ideal generated by $f_1, \ldots, f_s$ can be obtained by computing the row echelon form of these matrices up to some degree $\mathbb{D}_{\text{reg}}$ (called degree of regularity) on the one hand, when the monomial ordering which is used is *graded*, i.e. it starts by sorting monomials w.r.t. their degrees. On the other hand, he proved that, generically, $\mathbb{D}_{\text{reg}} \leqslant 1 + \sum_{i=1}^{s}(d_i - 1)$ where $d_i = \deg(f_i)$. All in all, this shows that, under these genericity assumptions, computing a Gröbner basis for a graded monomial ordering can be done within

$$O\left(\binom{n + \mathbb{D}_{\text{reg}}}{n}^{\omega}\right)$$

arithmetic operations (here ω is the constant of matrix multiplication).

This prototype complexity result holds both for the algorithms F4 and F5 if the aforementioned genericity assumptions hold.

Our ambitious goal in this project is to obtain complexity statements for Gröbner bases computations which bring exponential speedup w.r.t. the state of the art.

We target such improvements for elimination monomial orderings, which are important for robotics applications, as well as graded monomial orderings on which we will specifically focus (in particular the important graded reverse lexicographical monomial ordering). Indeed, these latter orderings are important for applications in computer security and communication since, as already sketched above, these new complexity bounds might impact on the analysis of security of post-quantum cryptociphers and signature schemes.

We emphasize that we not only aim at complexity improvements but also at proof-of-concept computer programs which illustrate the practical gain we obtain from these expected upstream research results. These computer programs will be made open source so that research communities can reproduce our results and build upon our progress.

The path to achieve this goal is articulated around three research objectives:

- the first one is to revisit Lazard's algorithm which computes truncated Gröbner bases through the prism of a novel ingredient, the use of polynomial matrices;
- the second one will put into practice this novel ingredient inside the F4 algorithm, hence leading to the F4pm algorithm (which stands for F4-polynomial matrix algorithm) and the first complexity improvements using polynomial matrices;
- the third step will finally be to revisit the F5 algorithm and obtain the F5pm algorithm making F4pm "optimal" in the sense that it will remove all reductions to 0 performed by F4pm.

4 Work programme

Paradigm shift

To reach our ambitious goal, one obviously needs a paradigm shift that will make the difference.

This project is based on the following one. As already said, the matrices which appear during the Gröbner bases computations have coefficients in the base field are seen as sparse matrices. It turns out that those matrices are not only sparse: they are structured and this structure can be understood and leveraged from a module-theoretic point of view.

The starting point of this project comes from a recent work by Berthomieu, Neiger and Safey El Din on change of order algorithms for Gröbner bases [2] in the case of ideals of dimension zero. These are families of algorithms which allow us to compute a Gröbner basis for a lexicographical ordering (hence enjoying a triangular structure which is the Graal when it comes to solving) from a Gröbner basis for the same ideal but computed for a graded reverse lexicographical ordering – grevlex for short – (which is easier to compute in general). The key observation which was made, in some earlier work by Faugère and Mou, is that the lexicographical Gröbner can be obtained from a matrix which is encoded in the grevlex Gröbner basis. This matrix, which also encodes a multiplication operator in some quotient ring, is a $D \times D$ one, where D is the number of solutions counted with multiplicities, its rows are actually given by some unit vectors and t other vectors are not unit vectors.

This matrix is actually a generalized companion matrix whose representation can be compactified by considering the intersection of the ideal under study with a univariate module (whose basis is known from the grevlex Gröbner basis). This leads to a matrix of size $t \times t$ with entries of average degree $\frac{D}{t}$.

Combined with advanced computer algebra techniques dealing with linear algebra problems with matrices whose entries are univariate polynomials [8, 10, 9], this led to an algorithm for the change of ordering which brings exponential speed up. Indeed, one obtains a complexity which is, up to logarithmic factors, $O\left(t^{\omega-1}D\right)$; to be compared with the $O(tD^2)$ complexity of classical approaches. It turns out that generically, when all entries of the system to be solved have degree d, one has $D = d^n$ and it has been proved that when d is fixed and $n \longrightarrow \infty$ it holds that $t \simeq \frac{1}{\sqrt{n\pi}}\sqrt{\frac{6}{(d-1)^2-1}}d^n$ [3]. Hence, the speed-up being $t^{2-\omega}D$, it is up to some constant depending in d, $\frac{1}{n^{1-\omega/2}}d^{n(3-\omega)}$.

The prototype complexity statements we target, at least in the zero-dimensional case, are of that kind.

Year 1: Lazard's algorithm and polynomial matrices

The goal is to extend this module-theoretic point of view to the computation of Gröbner basis by considering Macaulay matrices, this time with univariate polynomial entries, by investigating at this stage of the computation, the intersection of the ideal under study with some univariate module (and compute directly a basis for it).

The prototype algorithm we will study under this prism is the one due to Lazard. It actually does compute, in the homogeneous case, a truncated Gröbner basis up to some degree which is given as input by the end-user. This algorithmic framework is ideal to make the connection between Gröbner basis computations and the elimination theory based on the notion of multivariate resultant. We will then study how to tune Lazard's algorithm, using polynomial matrices, to compute truncated Gröbner bases computations w.r.t. elimination orderings. From the point of view of algorithms operating over polynomial matrices, the key object to be used here is the one of Hermite normal forms for polynomial matrices. This is precisely the one which is already used in [2].

The next step will then be to investigate how to use polynomial matrices for computing Gröbner bases w.r.t. the grevlex ordering. Here, the situation is slightly different since the way the matrices are constructed needs to be revisited. This will be done thanks to the abstraction provided by the aforementioned module-theoretic point of view. Also, the algorithmic framework operating over polynomial matrices will be slightly different since we expect shifted Popov forms to be a key ingredient in replacement of Hermite normal forms (which are better suited to elimination orderings).

Year 2: Towards an F4pm algorithm

As already explained, the F4 algorithm provides a framework for computing Gröbner bases which has proved to be more efficient than Buchberger's algorithm.

On generic instances, the F4 algorithm can be seen as an extension of Lazard's algorithm, getting rid on the degree requirement, by incorporating therein Buchberger's criterion to ensure termination.

We will build upon the progress made on Lazard's algorithm to tune the F4 algorithm in a way that it will manipulate polynomial matrices and perform Hermite / Popov normal form computations on the fly in order to compute Gröbner bases. This explains the name of this new algorithm, its suffix 'pm' indicating that this variant works with polynomial matrices.

We then target complexity studies using the usual machinery, based on commutative algebra and Hilbert series, for predicting the sizes of the matrices as well as the average degrees of the polynomials therein, under some classical genericity assumptions. This should lead to the first estimates on the complexity of computing Gröbner bases, in the framework of F4, which are better than the prototype estimate $\binom{n+\mathbb{D}_{\mathrm{reg}}}{n}$ and which should be similar to the one obtained for the change of ordering step in [2].

This will open new perspectives: all previous studies for overdetermined systems, performed with the context of multivariate and postquantum cryptography in mind. We will also study how to tune the F4pm algorithm for some special class of structured polynomial systems, in particular the class of multi-homogeneous polynomial systems, since it arises frequently both in postquantum cryptography and robotics.

Last but not least, we will investigate the behaviour of the F4pm algorithm for elimination orderings.

This F4pm algorithm is a necessary step towards the F5pm algorithm which we explain below.

Year 3: Towards an F5pm algorithm

The strength of the F5 algorithm is that it allows one to produce Macaulay-like matrices which are full rank on generic situations. This is by contrast with the F4 algorithm which produces Macaulay like matrices which are rank defective generically, hence leading to useless computations (this corresponds to rows which are reduced to 0 during the row echelonization process). This drawback of the F4 algorithm will remain in the F4pm algorithm since the additional ingredient added by the F5 algorithm to F4 is not present in F4pm.

This layer is the notion of signature which allows us to relate the reductions to 0 with linear dependencies which are induced by the so-called Koszul syzygies, coming from the commutativity of polynomial multiplication.

The next step will then be to introduce this additional layer of signatures into the F4pm algorithm. This task is far from trivial since the use of signatures depends on some monomial ordering over an abstract algebraic object (module) which is intricate to consider in the context of polynomial matrices. Still, the research track we will follow will consist in transposing the idea of signatures into the algorithmic framework for computing (shifted) Popov forms associated to polynomial matrices.

Obtaining this F5pm algorithm will open many perspectives. The first one is to adapt the refined complexity analysis performed in [1] to F5pm. Again, we do expect speedups whose dependency to the number of variables is exponential.

Finally, we will tune the F5pm algorithm to the case where the base field is the boolean one, which is an important case for cryptographic applications.

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 (Sept. 2015), pp. 49–70.

[2] J. Berthomieu, V. Neiger, and M. Safey El Din. "Faster change of order algorithm for Gröbner bases under shape and stability assumptions". In: *2022 International Symposium on Symbolic and Algebraic Computation*. Lille, France, July 2022.

[3] J. Berthomieu, A. Bostan, A. Ferguson, and M. Safey El Din. "Gröbner bases and critical values: The asymptotic combinatorics of determinantal systems". In: *Journal of Algebra* 602 (July 2022), pp. 154–180.

[4] B. Buchberger. "Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero dimensional polynomial ideal". In: *Journal of symbolic computation* 41.3-4 (2006), pp. 475–511.

[5] 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.

[6] J.-C. Faugère. "A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F5)". In: *Proceedings ISSAC '02*. 2002.

[7] J.-C. Faugère and A. Joux. "Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases". In: *Annual International Cryptology Conference*. Springer. 2003, pp. 44–60.

[8] P. Giorgi, C.-P. Jeannerod, and G. Villard. "On the complexity of polynomial matrix computations". In: *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*. 2003, pp. 135–142.

[9] S. Gupta and A. Storjohann. "Computing Hermite forms of polynomial matrices". In: *Proceedings of the 36th international symposium on Symbolic and algebraic computation*. 2011, pp. 155–162.

[10] G. Labahn, V. Neiger, and W. Zhou. "Fast, deterministic computation of the Hermite normal form and determinant of a polynomial matrix". In: *Journal of Complexity* 42 (2017), pp. 44–71.