Structured linear algebra for polynomial system solving

Master internship in Computer Algebra

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

Environment

Advisors: Jérémy Berthomieu[1], Vincent Neiger[2].

This internship will take place in LIP6, a joint lab between Sorbonne Université and CNRS in Paris. The intern will join a dynamical and scientifically ambitious team which advises and co-advises Ph.D. students and postdoctoral researchers from France and many other countries (currently and recently: China, Germany, The Netherlands, Spain, UK, USA, Vietnam, etc.). The intern will have access to office space, to all the necessary software, and to computing servers owned by the team.

This internship is particularly appropriate for students willing to pursue a Ph.D. after obtaining their Master degree.

Context, scientific positioning

Modeling problems from biology, coding theory, combinatorics, robotics or aerospace engineering relies on fundamental problems such as sparse multivariate polynomial interpolation, multidimensional cyclic code decoding or solving polynomial systems exactly over a finite field or rational numbers. Solving exactly a system of polynomial equations $f_1 = \cdots = f_m = 0$ in variables $x_1, \ldots, x_n$ over a field $\mathbb{K}$ comes down to computing a representation of its solution set, for instance through a *lexicographic Gröbner basis* of the ideal spanned by $f_1, \ldots, f_m$. This is a powerful tool and, generically, if the solution set consists in finitely many D points over an algebraic closure of $\mathbb{K}$, then this Gröbner basis is in the so-called *shape position*: $g_n(x_n) = 0, x_{n-1} = g_{n-1}(x_n), \ldots, x_1 = g_1(x_n)$, $\deg g_n = \mathrm{D}$. While computing Gröbner bases is in general exponential in n, some Gröbner bases are easier to compute than others. Indeed, the traditional strategy to obtain a lexicographic Gröbner basis is in two steps. First, compute a Gröbner basis of the ideal, for a *total degree* ordering using Buchberger's (1976) or Faugère's F_4 (1999) and F_5 (2002) algorithms. Then, apply a change of ordering algorithm on it using the FGLM algorithm (Faugère et al. 1993) or its faster variant, the Sparse-FGLM algorithm (Faugère and Mou 2011, 2017). In the generic case, this step yields $g_n, \ldots, g_1$ in essentially $\mathrm{O}(\mathrm{D}^2)$ operations. This latter algorithm builds a recurrent sequence from the input Gröbner basis and finds new recurrence relations satisfied by this sequence as kernel vectors of a highly structured matrix through dedicated algorithms (Brent et al. 1980) when in shape position and (Berthomieu et al. 2015, 2017; Berthomieu and Faugère 2018, 2022; Hyun et al. 2020; Sakata 2009) otherwise. Let us notice that all these algorithms rely heavily on linear algebra routines on structured matrices. Still, their complexities are not satisfactory, mostly because the structure of the matrices does not seem to be sufficiently exploited.

Internship Objectives

In this master internship, the candidate will investigate the algebra and geometry of the variety defined by the systems. In particular, we will investigate the *block-shape position* situation, where the lexicographic Gröbner basis is of the form

$$g_m(y_{n_2}), \ldots, g_1(y_1, \ldots, y_{n_2}),$$
$$x_{n_1} = h_{n_1}(y_1, \ldots, y_{n_2}), \ldots, x_1 = h_1(y_1, \ldots, y_{n_2}),$$

for two blocks of variables $x_1, \ldots, x_{n_1}$ and $y_1, \ldots, y_{n_2}$. In examples where n_1 is small compared to n_2, the structure of the matrix we compute the kernel of is better-understood than in the general case: it is *quasi-Hankel*. Thus, a random vector of the kernel can be computed fast (Bostan et al. 2007) using univariate polynomial arithmetic. Yet, a full special basis of this kernel is needed to recover $g_m, \ldots, g_1, h_{n_1}, \ldots, h_1$.

The student will study the exact structure of this matrix depending on polynomials $g_m, \ldots, g_1$ in some special cases but also in more general one. An implementation of these algorithms in `C` is expected with a potential integration into msolve (Berthomieu et al. 2021a,b), an efficient library for solving polynomial system solving.

Requirements

In order to carry out this project, the student is expected to have good knowledge in polynomial arithmetic, linear algebra and `C` implementation.

[1] Sorbonne Université, jeremy.berthomieu@lip6.fr, `https://www-polsys.lip6.fr/~berthomieu`

[2] Sorbonne Université, vincent.neiger@lip6.fr, `https://vincent.neiger.science`

References

J. Berthomieu, B. Boyer, and J.-C. Faugère (2015), "Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences", in *Proceedings ISSAC '15*, Bath, United Kingdom: ACM, pp. 61–68, DOI: 10.1145/2755996.2756673.

— (2017), "Linear Algebra for Computing Gröbner Bases of Linear Recursive Multidimensional Sequences", in *Journal of Symbolic Computation* 83, pp. 36–67, DOI: 10.1016/j.jsc.2016.11.005.

J. Berthomieu, C. Eder, and M. Safey El Din (2021a), *msolve: A Library for Solving Polynomial Systems*, https://msolve.lip6.fr/.

J. Berthomieu, C. Eder, and M. Safey El Din (2021b), "Msolve: A Library for Solving Polynomial Systems", in *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*, ISSAC '21, Virtual Event, Russian Federation: Association for Computing Machinery, pp. 51–58, DOI: 10/gk8549.

J. Berthomieu and J.-C. Faugère (2018), "A Polynomial-Division-Based Algorithm for Computing Linear Recurrence Relations", in *Proceedings ISSAC '18*, New York, NY, USA: ACM, pp. 79–86, DOI: 10.1145/3208976.3209017.

— (2022), "Polynomial-division-based algorithms for computing linear recurrence relations", in *Journal of Symbolic Computation* 109, pp. 1–30, DOI: https://doi.org/10.1016/j.jsc.2021.07.002.

A. Bostan, C.-P. Jeannerod, and É. Schost (2007), "Solving Toeplitz- and Vandermonde-like Linear Systems with Large Displacement Rank", in *ISSAC'07*, ed. by C. W. Brown, ACM Press, pp. 33–40, DOI: 10.1145/1277548.1277554.

R. P. Brent, F. G. Gustavson, and D. Y. Yun (1980), "Fast solution of Toeplitz systems of equations and computation of Padé approximants", in *Journal of Algorithms* 1.3, pp. 259–295, DOI: https://doi.org/10.1016/0196-6774(80)90013-9.

B. Buchberger (1976), "A Theoretical Basis for the Reduction of Polynomials to Canonical Forms", in *SIGSAM Bull.* 10.3, pp. 19–29, DOI: 10.1145/1088216.1088219.

J.-C. Faugère (1999), "A New Efficient Algorithm for Computing Gröbner bases (F4)", in *Journal of Pure and Applied Algebra* 139.1, pp. 61–88, DOI: https://doi.org/10.1016/S0022-4049(99)00005-5.

— (2002), "A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F5)", in *Proceedings ISSAC '02*.

J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora (1993), "Efficient Computation of Zero-dimensional Gröbner Bases by Change of Ordering", in *J. Symbolic Comput.* 16.4, pp. 329–344, DOI: http://dx.doi.org/10.1006/jsco.1993.1051.

J.-C. Faugère and C. Mou (2011), "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, California, USA: ACM, pp. 115–122, DOI: 10.1145/1993886.1993908.

— (2017), "Sparse FGLM algorithms", in *Journal of Symbolic Computation* 80.3, pp. 538–569.

S. G. Hyun, V. Neiger, H. Rahkooy, and É. Schost (2020), "Block-Krylov techniques in the context of sparse-FGLM algorithms", in *Journal of Symbolic Computation* 98, Special Issue on Symb. and Alg. Comp.: ISSAC 2017, pp. 163–191, DOI: 10/gkx9.

S. Sakata (2009), "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 Berlin Heidelberg, pp. 143–163, DOI: 10.1007/978-3-540-93806-4_9.