Fast Gröbner basis computation and polynomial reduction for generic bivariate ideals

Joris van der Hoeven and Robin Larrieu

Laboratoire d'informatique de l'École polytechnique
LIX, UMR 7161 CNRS
Campus de l'École polytechnique
1, rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing, CS35003
91120 Palaiseau, France
vdhoeven@lix.polytechnique.fr
larrieu@lix.polytechnique.fr

Abstract

Let $A, B \in \mathbb{K}[X, Y]$ be two bivariate polynomials over an effective field $\mathbb{K}$, and let G be the reduced Gröbner basis of the ideal $I := \langle A, B\rangle$ generated by A and B with respect to the usual degree lexicographic order. Assuming A and B sufficiently generic, we design a quasi-optimal algorithm for the reduction of $P \in \mathbb{K}[X, Y]$ modulo G, where "quasi-optimal" is meant in terms of the size of the input A, B, P. Immediate applications are an ideal membership test and a multiplication algorithm for the quotient algebra $\mathbb{A} := \mathbb{K}[X, Y]/\langle A, B\rangle$, both in quasi-linear time. Moreover, we show that G itself can be computed in quasi-linear time with respect to the output size.

1 Introduction

Gröbner bases, sometimes known as standard bases, are a powerful tool for solving systems of polynomial equations, or to compute modulo polynomial ideals. The research area dedicated to their computation is very active, and there is an abundant literature on efficient algorithms for this task. See for example [2, 3, 5] and references therein. Although this problem requires exponential space in the worst case [12], it is in fact tractable for many practical instances. For example, computer algebra systems often implement Faugère's F5 algorithm [3] that is very efficient if the system has sufficient regularity. In this case, a polynomial complexity bound (counting the number of field operations in terms of the expected output size) was established in [1].

The F5 algorithm and all other currently known fast algorithms for Gröbner basis computations rely on linear algebra, and it may seem surprising that fast FFT-based polynomial arithmetic is not used in this area. This can be seen as an illustration of how difficult it is to compute standard bases, but there is another explanation: traditionally, Gröbner basis algorithms consider a large number of variables and the degree of the generating polynomials is kept small; but fast polynomial arithmetic works best in the opposite regime (a fixed number of variables and large degrees). Even in this setting, it is not clear how to use FFT techniques for Gröbner basis computation. As a first step, one may consider related problems, such as the reduction of multivariate polynomials. It was shown in [8] that reduction can be done in quasi-linear time with respect to the size of the equation $P = Q_0G_0 + \cdots + Q_nG_n + R$. However, this equation is in general much larger than the intrinsic complexity of the problem, given by the size of P and

the degree D of the ideal (which is linked to the size of the generating polynomials). Recent work in the bivariate setting [9] gave an asymptotically optimal reduction algorithm for a particular class of Gröbner bases. This algorithm relies on a terse representation of $G := (G_0, \ldots, G_n)$ in $\tilde{O}(D)$ space, where $\tilde{O}$ stands for the "soft Oh" notation (that hides poly-logarithmic factors) [7]. Assuming that this representation has been precomputed, the extended reduction can be performed in time $\tilde{O}(|P| + D)$, instead of the previous $\tilde{O}(|P| + |G|)$ where $|G| = \Theta(nD)$.

Instead of making regularity assumptions on the Gröbner basis itself, one may focus on the generating polynomials. If the ideal is defined by generic polynomials given in total degree, then the Gröbner basis presents a particular structure, as studied for example in [6, 13]. This situation is often used as a benchmark for polynomial system solving: see the PoSSo problem [4]. In this paper, we restrict ourselves to the bivariate case, as studied for example in [11]. In what follows, $A, B \in \mathbb{K}[X, Y]$ are generic polynomials of degree n, m respectively. We denote by $\langle A, B\rangle$ the ideal they generate, and we consider its Gröbner basis with respect to the graded lexicographic order. The computation of such a basis is classical, but the hypotheses from [9] are not satisfied. We aim to show how the ideas from [9] can be applied in this setting and lead to quasi-linear complexity bounds for various problems. The full paper is available at [10].

2 Idea of the algorithm

While the new setting is not directly compatible with [9], the key ingredients remain valid. Recall that the major obstruction to quasi-linear reduction algorithms is that the equation

$$P = Q_0G_0 + Q_1G_1 + \cdots + Q_nG_n + R \tag{1}$$

is much larger than the input P, A, B: if A, B have degree n, the input takes $\Theta(n^2)$ space, but writing down $G_0, \ldots, G_n$ explicitly requires already $\Theta(n^3)$.

2.1 Key ingredients

The first idea is to use a *dichotomic selection strategy* to control the degrees of the quotients. A selection strategy describes how one chooses against which basis element a given term is reduced. In the dichotomic selection strategy, each monomial is reduced preferably against one end of the Gröbner basis (G_0 or G_n), or the G_i where i has the highest 2-adic valuation. This way, most quotients have a very small degree: roughly speaking, there are $n/2$ quotients of degree d, plus $n/4$ quotients of degree $2d$, plus $n/8$ quotients of degree $4d$, and so on, where d is a constant independent of n.

This allows for a second ingredient that is to keep only sufficiently many head terms of each G_i. The definition of "sufficiently many" depends on the degree of the quotient Q_i. For example, G_0 and G_n have to be known entirely as Q_0 and Q_n can have very large degree; but on the other hand, for the $n/2$ indices i such that $\deg(Q_i) = d$, it suffices to know the (approximately) nd head terms of G_i.

Knowing G_i with this precision is sufficient to compute the quotient Q_i, but not the remainder R. The third idea is to keep track of the relations that exist between $G_0, \ldots, G_n$. Using these relations, equation (1) can be symbolically rewritten to use fewer terms, so that the remainder can be evaluated with the expected complexity.

The later two ingredients lead to a so-called *terse representation* for the Gröbner basis. This representation consists of the basis elements truncated to the appropriate precision (ingredient 2), and the collection of some well-chosen relations (ingredient 3). The whole representation requires only $\tilde{O}(n^2)$ space.

2.2 Application

In [9], we assumed that the Gröbner basis has specific regularity properties; we called such a basis a *vanilla Gröbner basis*. Then, obtaining its terse representation is seen as a precomputation. Unfortunately, if the ideal is generated by two polynomials given in total degree, then its Gröbner basis with respect to the

degree lexicographic order is not vanilla. On the other hand, the Gröbner basis is particularly easy to compute in this case, so we find a new type of relations between the elements. Then, we design a *concise representation* for the basis, that presents similar features as the terse representation.

Since the basis is especially easy to compute, the design on the concise representation is constructive, while in [9], the existence of a terse representation was more of an empirical observation. We are therefore able to compute the terse representation of G in quasi-linear time from the input A, B. Using the properties of the concise representation, we also design a quasi-linear reduction algorithm as a variant of the algorithm from [9].

The advantage of vanilla Gröbner bases is that the quotients can all be computed as a first step, using the truncated basis elements and a classical reduction algorithm. Then the substitutions in equation (1) can happen in a second step to compute the remainder. Such a nice property is not satisfied in the new setting, so the algorithm must be adapted. The solution is to merge the two steps, and the replacements happen "on the fly" during the reduction algorithm. We actually modify the classical relaxed reduction algorithm [8] using the following principle: as soon as a quotient Q_i is known, the term $Q_i G_i$ in equation (1) is replaced by $S_k G_k + S_{k+1} G_{k+1}$ for a well-chosen k. By doing so, we ensure that each step uses only basis elements that are known with sufficient precision to get the actual result.

3 Conclusion

For a bivariate ideal $I := \langle A, B\rangle$ defined by two generic polynomials given in total degree, let G be its Gröbner basis with respect to the degree lexicographic order. We can compute a concise representation for G in quasi-linear time with respect to the input A, B. From this concise representation, we also design an algorithm for the reduction of P modulo G, that is quasi linear with respect to A, B, P. Then the following problems can be solved in quasi-linear time:

- Ideal membership test $P \in^? \langle A, B\rangle$ (with respect to the size of A, B, P).
- Multiplication in the quotient algebra $P, Q \to PQ \in \mathbb{K}[X, Y]/\langle A, B\rangle$ (with respect to the size of A, B, P, Q).
- Computation of the reduced Gröbner basis $G^{\mathrm{red}} = (G_0^{\mathrm{red}}, \ldots, G_n^{\mathrm{red}})$ of I for the degree lexicographic order (with respect to the size of G^{red}).

References

[1] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of the F5 Gröbner basis algorithm. *Journal of Symbolic Computation*, pages 1–24, sep 2014.

[2] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, 1999.

[3] Jean-Charles 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, pages 75–83. New York, NY, USA, 2002. ACM.

[4] Jean-Charles Faugère, Pierrick Gaudry, Louise Huot, and Guénaël Renault. Polynomial systems solving by fast linear algebra. *ArXiv preprint arXiv:1304.6039*, 2013.

[5] Jean-Charles Faugère, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[6] Ralf Fröberg and Joachim Hollman. Hilbert series for ideals generated by generic forms. *Journal of Symbolic Computation*, 17(2):149 – 157, 1994.

[7] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3rd edition, 2013.

[8] Joris van der Hoeven. On the complexity of polynomial reduction. In I. Kotsireas and E. Martínez-Moro, editors, *Proc. Applications of Computer Algebra 2015*, volume 198 of *Springer Proceedings in Mathematics and Statistics*, pages 447–458. Cham, 2015. Springer.

[9] Joris van der Hoeven and Robin Larrieu. Fast reduction of bivariate polynomials with respect to sufficiently regular Gröbner bases. Technical Report, HAL, 2018. `http://hal.archives-ouvertes.fr/hal-01702547`.

[10] Joris van der Hoeven and Robin Larrieu. Fast Gröbner basis computation and polynomial reduction for generic bivariate ideals. Technical Report, HAL, 2018. `http://hal.archives-ouvertes.fr/hal-01770408`.

[11] Romain Lebreton, Eric Schost, and Esmaeil Mehrabi. On the complexity of solving bivariate systems: the case of non-singular solutions. In *ISSAC: International Symposium on Symbolic and Algebraic Computation*, pages 251–258. Boston, United States, Jun 2013.

[12] Ernst Mayr. Membership in polynomial ideals over Q is exponential space complete. *STACS 89*, pages 400–406, 1989.

[13] Guillermo Moreno-Socías. Degrevlex gröbner bases of generic complete intersections. *Journal of Pure and Applied Algebra*, 180(3):263 – 283, 2003.