On the complexity of polynomial reduction*

Joris van der Hoeven

LIX, CNRS
École polytechnique
91128 Palaiseau Cedex
France

Email: vdhoeven@lix.polytechnique.fr
Web: http://lix.polytechnique.fr/~vdhoeven

May 30, 2015

Sparse interpolation [1, 3, 2, 10] provides an interesting paradigm for efficient computations with multivariate polynomials. In particular, under suitable hypothesis, multiplication of sparse polynomials can be carried out in quasi-linear time, in terms of the expected output size. More recently, other multiplication algorithms have also been investigated, which outperform naive and sparse interpolation under special circumstances [11, 9]. An interesting question is how to exploit such algorithms for accelerating other operations. In this paper, we will focus on the reduction of a multivariate polynomial with respect to an autoreduced set of other polynomials and show that fast multiplication algorithms can indeed be exploited in this context in an asymptotically quasi-optimal way.

Consider the polynomial ring $\mathbb{K}[x] = \mathbb{K}[x_1, \ldots, x_n]$ over an effective field $\mathbb{K}$ with an effective zero test. Given a polynomial $P = \sum_{i\in\mathbb{N}^n} P_i\, x^i = \sum_{i_1,\ldots,i_n\in\mathbb{N}} P_{i_1,\ldots,i_n}\, x_1^{i_1} \cdots x_n^{i_n}$, we call $\operatorname{supp} P = \{i \in \mathbb{N}^n : P_i \neq 0\}$ the *support* of P. The naive multiplication of two sparse polynomials $P, Q \in \mathbb{K}[x]$ requires *a priori* $\mathcal{O}(|\operatorname{supp} P|\,|\operatorname{supp} Q|)$ operations in $\mathbb{K}$. This upper bound is sharp if P and Q are very sparse, but pessimistic if P and Q are dense.

Assuming that $\mathbb{K}$ has characteristic zero, a better algorithm was proposed in [2] (see also [1, 3] for some background). The complexity of this algorithm can be expressed in the expected size $s = |\operatorname{supp} P + \operatorname{supp} Q|$ of the *output* (when no cancellations occur). It is shown that P and Q can be multiplied using only $\mathcal{O}(\mathsf{M}(s)\log s)$ operations in $\mathbb{K}$, where $\mathsf{M}(s) = \mathcal{O}(s \log s \log\log s)$ stands for the complexity of multiplying two univariate polynomials in $\mathbb{K}[z]$ of degrees $<s$. Unfortunately, the algorithm in [2] has two drawbacks:

1. The algorithm leads to a big growth for the sizes of the coefficients, thereby compromising its bit complexity (which is often worse than the bit complexity of naive multiplication).
2. It requires $\operatorname{supp} P\, Q \subseteq \operatorname{supp} P + \operatorname{supp} Q$ to be known beforehand. More precisely, whenever a bound $\operatorname{supp} P\, Q \subseteq \operatorname{supp} P + \operatorname{supp} Q \subseteq \mathcal{S}$ is known, then we really obtain a multiplication algorithm of complexity $\mathcal{O}(\mathsf{M}(|\mathcal{S}|)\log|\mathcal{S}|)$.

*. This work has been supported by the ANR-09-JCJC-0098-01 MaGiX project, the Digiteo 2009-36HD grant and Région Ile-de-France.

In practice, the second drawback is of less importance. Indeed, especially when the coefficients in $\mathbb{K}$ can become large, then the computation of $\operatorname{supp} P + \operatorname{supp} Q$ is often cheap with respect to the multiplication $P\,Q$ itself, even if we compute $\operatorname{supp} P + \operatorname{supp} Q$ in a naive way.

Recently, several algorithms were proposed for removing the drawbacks of [2]. First of all, in [10] we proposed a practical algorithm with essentially the same advantages as the original algorithm from [2], but with a good bit complexity and a variant which also works in positive characterisic. However, it still requires a bound for $\operatorname{supp} P\,Q$ and it only works for special kinds of fields $\mathbb{K}$ (which nevertheless cover the most important cases such as $\mathbb{K} = \mathbb{Q}$ and finite fields). Even faster algorithms were proposed in [7, 11], but these algorithms only work for special supports. Yet another algorithm was proposed in [5, 9]. This algorithm has none of the drawbacks of [2], but its complexity is suboptimal (although better than the complexity of naive multiplication).

At any rate, these recent developments make it possible to rely on fast sparse polynomial multiplication as a building block, both in theory and in practice. This makes it natural to study other operations on multivariate polynomials with this building block at our disposal. One of the most important such operations is division.

The multivariate analogue of polynomial division is the reduction of a polynomial $A \in \mathbb{K}[x]$ with respect to an autoreduced tuple $B = (B_1, \ldots, B_b) \in \mathbb{K}[x]^b$ of other polynomials. This leads to a relation

$$A \;=\; Q_1 B_1 + \cdots + Q_b B_b + R, \tag{1}$$

such that none of the terms occurring in R can be further reduced with respect to B. In this paper, we are interested in the computation of R as well as $Q_1, \ldots, Q_b$. We will call this the problem of *extended reduction*, in analogy with the notion of an "extended g.c.d.".

Now in the univariate context, "relaxed power series" provide a convenient technique for the resolution of implicit equations [4, 5, 6, 8]. One major advantage of this technique is that it tends to respect most sparsity patterns which are present in the input data and in the equations. The main technical tool in this paper (see section 2) is to generalize this technique to the setting of multivariate polynomials, whose terms are ordered according to a specific admissible ordering on the monomials. This will make it possible to rewrite (1) as a so called recursive equation (see section 3.2), which can be solved in a relaxed manner. Roughly speaking, the cost of the extended reduction then reduces to the cost of the relaxed multiplications $Q_1 B_1, \ldots, Q_b B_b$. Up to a logarithmic overhead, we show that this cost is the same as the cost of checking the relation (1).

References

[1] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *STOC '88: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 301–309, New York, NY, USA, 1988. ACM Press.

[2] J. Canny, E. Kaltofen, and Y. Lakshman. Solving systems of non-linear polynomial equations faster. In *Proc. ISSAC '89*, pages 121–128, Portland, Oregon, A.C.M., New York, 1989. ACM Press.

[3] D. Y. Grigoriev and M. Karpinski. The matching problem for bipartite graphs with polynomially bounded permanents is in NC. In *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science*, pages 166–172, 1987.

[4] J. van der Hoeven. Lazy multiplication of formal power series. In W. W. Küchlin, editor, *Proc. ISSAC '97*, pages 17–20, Maui, Hawaii, July 1997.

[5] J. van der Hoeven. Relax, but don't be too lazy. *JSC*, 34:479–542, 2002.

[6] J. van der Hoeven. Relaxed multiplication using the middle product. In Manuel Bronstein, editor, *Proc. ISSAC '03*, pages 143–147, Philadelphia, USA, August 2003.

[7] J. van der Hoeven. The truncated Fourier transform and applications. In J. Gutierrez, editor, *Proc. ISSAC 2004*, pages 290–296, Univ. of Cantabria, Santander, Spain, July 4–7 2004.

[8] J. van der Hoeven. New algorithms for relaxed multiplication. *JSC*, 42(8):792–802, 2007.

[9] J. van der Hoeven and G. Lecerf. On the complexity of blockwise polynomial multiplication. In *Proc. ISSAC '12*, pages 211–218, Grenoble, France, July 2012.

[10] J. van der Hoeven and G. Lecerf. On the bit-complexity of sparse polynomial multiplication. *JSC*, 50:227–254, 2013.

[11] J. van der Hoeven and É. Schost. Multi-point evaluation in higher dimensions. *AAECC*, 24(1):37–52, 2013.