SADIO Electronic Journal of Informatics and Operations Research

http://www.dc.uba.ar/sadio/ejs/

vol. 5, no. 1, pp. 1–10 (2003)

Fast Multivariate Power Series Multiplication in Characteristic Zero

G. Lecerf[1] *É. Schost*[2]

[1] UMR 8100 CNRS, Laboratoire de Mathématiques, Université de Versailles St-Quentin-en-Yvelines, 45, avenue des États-Unis, 78035 Versailles, France. lecerf@math.uvsq.fr

[2] FRE 2341 CNRS, Laboratoire GAGE, École polytechnique, 91128 Palaiseau, France. schost@gage.polytechnique.fr

Abstract

Let k be a field of characteristic zero. We present a fast algorithm for multiplying multivariate power series over k truncated in total degree. Up to logarithmic factors, its complexity is optimal, i.e. *linear* in the number of coefficients of the series.

Keywords. Multivariate power series, fast multiplication, complexity.

1 Introduction

Let k be a field of characteristic zero. We denote by S the multivariate power series ring in n variables $k[[x_1, \ldots, x_n]]$ and by $\mathfrak{m}$ its maximal ideal $(x_1, \ldots, x_n)$. For any positive integer d we write $\deg(\mathfrak{m}^{d+1})$ for the *degree* of the ideal $\mathfrak{m}^{d+1}$, that is, the number of monomials in S which are not in $\mathfrak{m}^{d+1}$. It is well-known that

$$\deg(\mathfrak{m}^{d+1}) = \binom{d+n}{n}.$$

We view a power series f in S at precision $\mathfrak{m}^{d+1}$ as a vector in the k-algebra $S/\mathfrak{m}^{d+1}$. This algebra has dimension $\deg(\mathfrak{m}^{d+1})$.

In this article we give an asymptotically fast algorithm for multiplying two power series at precision $\mathfrak{m}^{d+1}$: the cost of one multiplication is linear in $\deg(\mathfrak{m}^{d+1})$ up to logarithmic factors. As for many other algorithms dedicated to fast multiplication, our method relies on multi-point evaluation and interpolation: this brings back the problem to multiplication in $k[t]$ modulo t^{d+1}, for which fast algorithms are known.

Previous work. To the best of our knowledge, this is the first time that the question of a fast multiplication algorithm is addressed in this context. Apart from the naive algorithm, with complexity quadratic in $\deg(\mathfrak{m}^{d+1})$, the best algorithm known up to now is hinted at by [3]: it relies on Kronecker's substitution [14]. This method requires to compute modulo $(x_1^{d+1}, \ldots, x_n^{d+1})$ instead of $\mathfrak{m}^{d+1}$ and amounts to multiplying two univariate polynomials in degree $(2d)^n$.

Using fast univariate multiplication algorithms (see below), the complexity of this approach is linear in the degree $(2d)^n$, up to logarithmic factors. On the other hand, the number of coefficients of a series at precision $\mathfrak{m}^{d+1}$ is the combinatorial number $\binom{d+n}{n}$, and the quantity $(2d)^n$ is not polynomially bounded in terms of $\binom{d+n}{n}$. For fixed n, $\binom{d+n}{n}$ grows like $\frac{d^n}{n!}$ as a function of d; thus the overhead of the approach through Kronecker's substitution has order $c2^n n!$, for some positive constant c.

Model of Computation. All along this paper, our model of computation is the *arithmetic circuit* over k, with operations $(+, \times)$. The *size* of an arithmetic circuit is the number of its internal nodes, see [4] (Chapter 4) for a detailed definition. All polynomials and all elements of $S/\mathfrak{m}^{d+1}$ are represented by the vector of their coefficients in the canonical monomial basis.

Main result. With these conventions, our main result is the following theorem:

Theorem 1 *Let n, d be integers, and let D denote $\deg(\mathfrak{m}^{d+1})$. There exists an arithmetic circuit which, taking as input two elements f and g of $S/\mathfrak{m}^{d+1}$, outputs their product fg, and has size*

$$\mathcal{O}\big(D \log(D)^3 \log(\log(D))\big).$$

Our result is closely related to the algorithms for sparse [1, 23] or dense [5] multivariate polynomial multiplication, which achieve a linear complexity (up to logarithmic factors) in the number of monomials in the output. All these results rely on a fast multi-point evaluation and interpolation scheme for multivariate polynomials, for a specific choice of sample points, namely powers of prime numbers. This idea was introduced by [11] and [22]. We recall this fundamental result in Lemma 1, following the presentation of [5].

Applications. Operations modulo a power of the maximal ideal in a power series ring appear frequently in relation to Newton-Hensel lifting techniques. A classical example is the factorization of multivariate polynomials. Our own initial interest comes from the field of polynomial system solving, where such lifting techniques are used as well. We refer to Section 3 for more details.

2 Proof of the main result

We first introduce some notation. In the following, C (resp. D) denotes the number of monomials in $n-1$ (resp. n) variables of degree at most d:

$$C := \binom{d+n-1}{n-1}, \qquad D := \binom{d+n}{n}.$$

The log function is the Neperian logarithm ($\log(e) = 1$).

Our proof is divided into three lemmas, the first of which is taken from [5]. Its result is stated in terms of the function $\mathcal{M}_u(\delta)$, which denotes the complexity of the multiplication of two univariate polynomials of degree δ in $k[t]$. Schönhage and Strassen [20, 19] proved that $\mathcal{M}_u(\delta)$ belongs to $\mathcal{O}(\delta \log(\delta) \log(\log(\delta)))$.

Lemma 1 **[5]** *Let d, n be integers, and $p_2, \ldots, p_n$ distinct prime numbers. For $i \in \{0, \ldots, C-1\}$, denote by P_i the point $(p_2^i, \ldots, p_n^i) \in k^{n-1}$. There exist arithmetic circuits* Ev *and* Int *of sizes $\mathcal{O}(\mathcal{M}_u(C)\log(C))$ such that,*

- **Multi-point evaluation:** *on input a polynomial f in $k[x_2, \ldots, x_n]$ of degree at most d,* Ev *computes the values $f(P_0), \ldots, f(P_{C-1})$;*
- **Interpolation:** *on input $a_0, \ldots, a_{C-1}$,* Int *computes the polynomial f in $k[x_2, \ldots, x_n]$ of degree at most d such that $f(P_i) = a_i$ for $i \in \{0, \ldots, C-1\}$.*

Proof. Let $M_1, \ldots, M_C$ be all monomials in $k[x_2, \ldots, x_n]$ of degree at most d, and $v_1, \ldots, v_C$ their values at P_1, that is, with x_j evaluated at p_j for $j \in \{2, \ldots, n\}$. Let next M be the $C \times C$ matrix over k,

$$M = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ v_1 & v_2 & \cdots & v_C \\ v_1^2 & v_2^2 & \cdots & v_C^2 \\ \vdots & \vdots & & \vdots \\ v_1^{C-1} & v_2^{C-1} & \cdots & v_C^{C-1} \end{bmatrix}.$$

Then evaluating a polynomial $f \in k[x_2, \ldots, x_n]$ of degree at most d on the points $P_0, \ldots, P_{C-1}$ is done by multiplying M by the column vector of the coefficients of f. Similarly, interpolating the coefficients of a polynomial $f \in k[x_2, \ldots, x_n]$ of degree at most d from its values at $P_0, \ldots, P_{C-1}$ is done by multiplying the column vector made of these values by the inverse of M.

Let us consider the matrix $N = M^t$, the transpose of M. The matrix N is the Vandermonde matrix associated to the points $v_1, \ldots, v_C$. Thus, there exist arithmetic circuits of sizes $\mathcal{O}(\mathcal{M}_u(C)\log(C))$ that perform the multiplication of a column vector by N (resp. by its inverse): these come from fast univariate evaluation and interpolation algorithms, see [4].

The conclusion follows from Tellegen's transposition principle [18]: since there exist arithmetic circuits of sizes $\mathcal{O}(\mathcal{M}_u(C)\log(C))$ for performing the product by M^t (resp. by its inverse), there exist arithmetic circuits of the same size that perform the product by M (resp. by its inverse). □

On the basis of this result, the following lemma gives a first upper bound on the complexity of multivariate power series multiplication, stated in terms of the function $\mathcal{M}_u$. In a second stage, we will show that Schönhage-Strassen's multiplication scheme yields the claim of Theorem 1.

Lemma 2 *Let n, d be integers. There exists an arithmetic circuit which, taking as input two elements f and g of $S/\mathfrak{m}^{d+1}$, outputs their product fg, and has size*

$$\mathcal{O}\Big(d\mathcal{M}_u(C)\log(C) + \mathcal{M}_u(d)C\Big),$$

where C is the number of monomials of degree at most d in $n-1$ variables as defined above.

Proof. Let f, g be two elements of $S/\mathfrak{m}^{d+1}$, h their product and $\mathfrak{f}, \mathfrak{g}, \mathfrak{h}$ the canonical preimages of f, g, h in $k[x_1, \ldots, x_n]$. It is enough to compute $\mathfrak{h}$ to conclude.

Let t be a new variable. We define polynomials $F, G, H \in k[x_1, \ldots, x_n][t]$ by

$$F := \mathfrak{f}(x_1 t, x_2 t, \ldots, x_n t),\ G := \mathfrak{g}(x_1 t, x_2 t, \ldots, x_n t),\ H := \mathfrak{h}(x_1 t, x_2 t, \ldots, x_n t).$$

The polynomials F, G and H satisfy the equality $H = FG \bmod t^{d+1}$. The polynomial $\mathfrak{h}$ can be recovered from H by evaluation at $t = 1$, so we now focus on a fast way to compute H.

The polynomials F, G and H can be written

$$\begin{aligned} F &= f_0 + f_1 t + \cdots + f_d t^d, \\ G &= g_0 + g_1 t + \cdots + g_d t^d, \\ H &= h_0 + h_1 t + \cdots + h_d t^d, \end{aligned}$$

where for all i, f_i, g_i and h_i belong to $k[x_1, \ldots, x_n]$ and are homogeneous of degree i. We will work with their de-homogenized counterparts, by introducing

the polynomials $\overline{F}, \overline{G}, \overline{H} \in k[x_2, \ldots, x_n][t]$ obtained by letting $x_1 = 1$ in F, G, H. Of course, it is enough to compute $\overline{H}$, since we can obtain H from it without additional cost by suitably homogenizing all its coefficients.

Just as above, the polynomials $\overline{F}$, $\overline{G}$ and $\overline{H}$ can be written

$$\begin{aligned} \overline{F} &= \overline{f}_0 + \overline{f}_1 t + \cdots + \overline{f}_d t^d, \\ \overline{G} &= \overline{g}_0 + \overline{g}_1 t + \cdots + \overline{g}_d t^d, \\ \overline{H} &= \overline{h}_0 + \overline{h}_1 t + \cdots + \overline{h}_d t^d, \end{aligned}$$

where for all i, $\overline{f}_i$, $\overline{g}_i$ and $\overline{h}_i \in k[x_2, \ldots, x_n]$ are obtained by evaluating f_i, g_i and h_i at $x_1 = 1$, and thus have degree at most i.

Following Lemma 1, we now take $P_i = (p_2^i, \ldots, p_n^i)$, for distinct prime numbers $p_2, \ldots, p_n$. For any P_i, we write $\overline{F}_{P_i}$ for the polynomial $\overline{f}_0(P_i) + \overline{f}_1(P_i)t + \cdots + \overline{f}_d(P_i)t^d$ in $k[t]$. We similarly define $\overline{G}_{P_i}$ and $\overline{H}_{P_i}$, so that the equality $\overline{H}_{P_i} = \overline{F}_{P_i}\overline{G}_{P_i}$ mod t^{d+1} holds. This leads to the following evaluation-interpolation scheme.

Algorithm. Given f and g in $S/\mathfrak{m}^{d+1}$, to compute $h = fg$ in $S/\mathfrak{m}^{d+1}$. Let $\overline{F}, \overline{G}, \overline{H}$ be as above.

1. Compute $\overline{F}_{P_i}, \overline{G}_{P_i}$ for C points $P_0, \ldots, P_{C-1}$, with $P_i \in k^{n-1}$ for all i.
2. Compute the C products $\overline{H}_{P_i} = \overline{F}_{P_i}\overline{G}_{P_i}$ mod t^{d+1} in $k[t]$.
3. Interpolate the polynomials $\overline{h}_0, \ldots, \overline{h}_d \in k[x_2, \ldots, x_n]$
4. For $i = 0, \ldots, d$, homogeneize $\overline{h}_i$ in degree i using the variable x_1, and call h_i the result. Let $H = h_0 + h_1 t + \cdots + h_d t^d$ and $\mathfrak{h} = H(1)$. Then $h = \mathfrak{h}$ mod $\mathfrak{m}^{d+1}$.

Complexity analysis. Step 4 has no arithmetic complexity, we examine the cost of steps 1, 2 and 3.

- By Lemma 1, there exists an arithmetic circuit of size $\mathcal{O}(d\mathcal{M}_u(C)\log(C))$ which, on input f and g, computes the values

  $$\overline{f}_j(P_0), \ldots, \overline{f}_j(P_{C-1}) \quad \text{and} \quad \overline{g}_j(P_0), \ldots, \overline{g}_j(P_{C-1}).$$

  for j in $\{0, \ldots, d\}$. This gives the complexity of step 1.

- For $i \in \{0, \ldots, C-1\}$, $\overline{H}_{P_i}$ is obtained by an univariate series product in $k[t]$, which can be done within $\mathcal{O}(\mathcal{M}_u(d))$ arithmetic operations. Step 2 then requires $\mathcal{O}(\mathcal{M}_u(d)C)$ arithmetic operations.

- Using the second result given in Lemma 1, the interpolation of all $\overline{h}_j$ can be done by an arithmetic circuit of size $\mathcal{O}(d\mathcal{M}_u(C)\log(C))$. This accounts for step 3. □

The result of Lemma 2 is given in terms of the number of monomials C, whereas our objective is a bound in terms of the quantity D. The last lemma answers this question.

Lemma 3 *For all $d \geq 0, n \geq 1$, the following inequality holds:*

$$\frac{dC}{D} = \frac{nd}{n+d} \leq \log(D).$$

Proof. The first equality is obvious. To prove the inequality, note first that the inequality $\frac{d}{1+d} \leq \log(1+d)$ holds for all $d \geq 0$. Indeed, let $f(d) = \frac{d}{1+d}$ and $g(d) = \log(1+d)$, then $f'(d) = \frac{1}{(1+d)^2}$, $g'(d) = \frac{1}{1+d}$, so $f'(d) \leq g'(d)$ holds for $d \geq 0$. Since $f(0) = g(0) = 0$, the assertion follows.

Now we rewrite D as $\frac{(n+1)\cdots(n+d)}{d!}$. Then we fix d and introduce the functions $u : n \mapsto \frac{nd}{n+d}$ and $v : n \mapsto \log \frac{(n+1)\cdots(n+d)}{d!}$, so we have to prove that $u(n) \leq v(n)$ for $n \geq 1$. For $n = 1$, the inequality to prove reads as $\frac{d}{1+d} \leq \log(1+d)$, which was proved above. We conclude the proof by proving the inequality $u'(n) \leq v'(n)$ for $n \geq 1$. The derivatives of u and v are

$$u'(n) = \frac{d^2}{(n+d)^2}, \quad v'(n) = \sum_{i=1}^{d} \frac{1}{n+i}.$$

The conclusion immediately follows from the inequalities

$$\sum_{i=1}^{d} \frac{1}{n+i} \geq \sum_{i=1}^{d} \frac{1}{n+d} = \frac{d}{n+d} \geq \frac{d^2}{(n+d)^2}.$$

□

We are now ready to prove Theorem 1. We first replace $\mathcal{M}_u(C)$ by the estimate $\mathcal{O}(C\log(C)\log(\log(C)))$ in the complexity of Lemma 2. Then, according to the above lemma we bound C by $D\log(D)/d$. This yields a complexity in:

$$\mathcal{O}\Bigg(\begin{array}{l} D\log(D)\Big(\log(D)+\log(\log(D))\Big)^2 \log\Big(\log(D)+\log(\log(D))\Big) \\ \quad +D\log(D)\log(d)\log(\log(d)) \end{array} \Bigg).$$

As for the second term we use $d \leq D$, therefore:

$$d\mathcal{M}_u(C)\log(C) + \mathcal{M}_u(d)C \in \mathcal{O}(D\log(D)^3\log(\log(D))).$$

This concludes the proof of Theorem 1.

3 Applications

As mentioned in the introduction, lifting techniques often require to compute modulo a power of the maximal ideal in a power series ring. A first and classical example is multivariate polynomial factorization through Newton-Hensel techniques. We refer to Chapter 15 in [6] for a presentation.

Our interest for power series multiplication originates from the field of polynomial system solving. In the second author's PhD thesis [21], the situation is as follows. We consider some polynomials $f_1, \ldots, f_m$ in $k(x_1, \ldots, x_n)[y_1, \ldots, y_m]$, and want to solve the system $f_1 = \cdots = f_m = 0$. The variables x_i play the role of parameters and we are looking for formulas expressing the solutions y_i in terms of these parameters.

Let us assume that the system is zero dimensional in the algebraic closure of $k(x_1, \ldots, x_n)$. Then, we will represent its solutions by a family of polynomials $Q, V_1, \ldots, V_m$ in $k(x_1, \ldots, x_n)[T]$, such that $f_i(V_1, \ldots, V_m) = 0$ modulo Q, for $i \in \{1, \ldots, m\}$. The techniques we now describe are inspired by previous work of Giusti, Heintz, Pardo and collaborators in [9, 7, 8, 12].

To compute $Q, V_1, \ldots, V_m$, we proceed the following way:

1. We pick up a point $(p_1, \ldots, p_n)$ at random in k^n, we substitute the variables x_i by p_i in the system and solve this specialized system. If $(p_1, \ldots, p_n)$ is generic enough, the solutions of this system can be represented by the polynomials $Q, V_1, \ldots, V_m$ with all coefficients specialized at $(p_1, \ldots, p_n)$. Up to a change of variables, we assume that $(p_1, \ldots, p_n) = (0, \ldots, 0)$, and let $\mathfrak{m}$ be the maximal ideal of $k[[x_1, \ldots, x_n]]$.

2. We lift the dependency of the solutions in the parameters in the formal power series ring $k[[x_1, \ldots, x_n]]$, using a formal version of Newton's iterator, as proposed by [10] and [15, 16]. The k-th lifting step takes as input the polynomials $Q, V_1, \ldots, V_m$ with coefficients reduced modulo $\mathfrak{m}^{2^\kappa}$ and outputs these polynomials with coefficients reduced modulo $\mathfrak{m}^{2^{\kappa+1}}$.

3. Once we have reached a sufficient precision we recover the coefficients of $Q, V_1, \ldots, V_m$ thanks to a multivariate version of Padé's approximants.

The lifting is the bottleneck of this method. Indeed, denoting the input of the k-th lifting step by $Q^{(\kappa)}, V_1^{(\kappa)}, \ldots, V_m^{(\kappa)}$, the lifting requires the evaluation of $f_i\left(V_1^{(\kappa)}, \ldots, V_m^{(\kappa)}\right)$ modulo $Q^{(\kappa)}$, for $i \in \{1, \ldots, m\}$. This is where we really need fast multivariate power series multiplication.

In a similar spirit, multiplication routines modulo $\deg(\mathfrak{m}^{d+1})$ are useful to treat systems of partial differential equations: roughly speaking, once a characteristic set of the system is known, the Taylor series expansions of non-singular solutions can be computed by successive approximations, which require arithmetic operations on power series. This idea is presented for instance by [2] and [17].

4 Conclusion

Van der Hoeven [13] generalized our algorithm to the case when

$$\mathfrak{I} = (x_1^{d_1} \cdots x_n^{d_n} \mid \alpha_1 d_1 + \cdots + \alpha_n d_n > d),$$

where the α_i and d are positive integers. However, the problem of fast computation with multivariate power series modulo any $\mathfrak{m}$-primary monomial ideal $\mathfrak{I}$ (with dense representation and using the canonical monomial basis) is not answered yet. For instance, such computations are motivated by the deflation algorithm presented in [16], which generalizes Newton's operator for isolated multiple roots. A first question in this direction is the cost of the multiplication modulo the ideal $(x_1^{d+1}, \ldots, x_n^{d+1})$. In this situation, our algorithm requires precision $\mathfrak{m}^{nd+1}$, this yields a complexity in $\mathcal{O}((ed)^n)$, up to logarithmic factors. We do not improve the best complexity result, which is in $\mathcal{O}(\mathcal{M}_u(2d)^n)$ using Kronecker's substitution.

Also, our result is stated in terms of arithmetic complexity. It is not immediate to design an efficient implementation of this algorithm. For instance, if the base field is the rational field $\mathbb{Q}$, we do not know the bit complexity and we would certainly want to use multi-modular and Chinese remainder techniques in order to avoid the growth of the integers in the intermediate computations; this would require to extend our result to finite fields.

This naturally opens the more general question of fast computation with multivariate power series over any ring. The point is to extend Lemma 1: the main difficulty is to choose points in the base ring such that distinct monomials take distinct values on these points. A first result in this direction, based on combinatorial arguments, is given by [23].

Acknowledgments

We greatly thank B. Salvy for his useful comments, O. Ruatta for sending typing mistakes to us and J. van der Hoeven for his pointing out to us an error in the complexity estimate in an earlier version of this paper. We also thank an anonymous referee for pointing out many inaccuracies in that previous version.

References

[1] M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *20th Annual ACM Symposium Theory of Computing*, 1988.

[2] F. Boulier, D. Lazard, F. Ollivier, and M. Petitot. Representation for the radical of a finitely generated differential ideal. In *Proceedings ISSAC'95*, pages 158–166. ACM Press, 1995.

[3] R. P. Brent and H. T. Kung. Fast algorithms for composition and reversion of multivariate power series. In *Proceedings of the Conference on Theoretical Computer Science, University of Waterloo, Ontario, Canada*, pages 149–158, 1977.

[4] P. Bürgisser, M. Clausen, and M. A. Shokrolahi. *Algebraic Complexity Theory*. Springer, 1997.

[5] J. Canny, E. Kaltofen, and Y. Lakshman. Solving systems of non-linear polynomial equations faster. In *Proceedings ISSAC'89*, pages 121–128. ACM Press, 1989.

[6] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999.

[7] M. Giusti, K. Hägele, J. Heintz, J.-E Morais, J.-L. Montaña, and L.-M. Pardo. Lower bounds for Diophantine approximation. In *Proceedings of MEGA'96*, number 117, 118 in Journal of Pure and Applied Algebra, pages 277–317, 1997.

[8] M. Giusti, J. Heintz, J.-E. Morais, J. Morgenstern, and L.-M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124:101–146, 1998.

[9] M. Giusti, J. Heintz, J.-E. Morais, and L.-M. Pardo. When polynomial equation systems can be solved fast? In *Proceedings of AAECC-11*, volume 948 of *Lecture Notes in Computer Science*, pages 205–231. Springer, 1995.

[10] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *Journal of Complexity*, 17(1):154–211, 2001.

[11] 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.

[12] J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *Journal of Complexity*, 16(1), 2000.

[13] J. van der Hoeven. Relax, but don't be too lazy. *Journal of Symbolic Computation*, 34(6):479–542, December 2002.

[14] L. Kronecker. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *Journal für die Reine und Angewandte Mathematik*, 92:1–122, 1882.

[15] G. Lecerf. *Une alternative aux méthodes de réécriture pour la résolution des systèmes algébriques*. PhD thesis, École polytechnique, 2001.

[16] G. Lecerf. Quadratic Newton iteration for systems with multiplicity. *Journal of FoCM*, 2(3):247–293, 2002.

[17] A. Péladan. *Tests effectifs de nullité dans des extensions d'anneaux différentiels*. PhD thesis, École polytechnique, 1997.

[18] P. Penfield, R. Spencer, and S. Duinker. *Tellegen's theorem and electrical Networks*. M.I.T. Press, Cambridge, Massachussetts, 1970.

[19] A. Schönhage. Schnelle Multiplikation von Polynomen über Körpern der Charakteristik 2. *Acta Informatica*, 7:395–398, 1977.

[20] A. Schönhage and V. Strassen. Schnelle Multiplikation großer Zahlen. *Computing*, 7:281–292, 1971.

[21] É. Schost. *Sur la résolution des systèmes polynomiaux à paramètres*. PhD thesis, École polytechnique, http://www.gage.polytechnique.fr/schost.html, 2000.

[22] P. Tiwari. Parallel algorithms for instance of the linear matroid parity with a small number of solutions. Technical Report IBM Reseach Report RC 12766, IBM Watson Research Center, Yorktown Heights, NY, 1987.

[23] R. Zippel. Interpolating polynomials from their values. *Journal of Symbolic Computation*, 9(3):375–403, 1990.