Complexity issues on Newton sums of polynomials

Alin Bostan, STIX, École polytechnique
91128 Palaiseau, France
bostan@stix.polytechnique.fr

Laureano Gonzalez-Vega, Universidad de Cantabria
39005 Santander, Spain
laureano.gonzalez@unican.es

Hervé Perdry, Universidad de Cantabria
39005 Santander, Spain
herve.perdry@unican.es

Éric Schost, STIX, École polytechnique
91128 Palaiseau, France
schost@stix.polytechnique.fr

Abstract

We consider the following problem: given the first d Newton sums of a degree d polynomial P, recover the coefficients of P. We propose fast algorithms to perform this conversion for polynomials over the p-adic ring $\mathbb{Z}_p$. As an application, we deduce a new algorithm to compute the *composed product* of polynomials over the finite field $\mathbb{F}_p$, whose bit-complexity improves the known results by a factor of $\log d$ in many cases.

1 Introduction

Univariate polynomials are usually represented by their coefficients in the monomial basis. An useful alternative representation is provided by the power sums (or Newton sums) of their roots. For instance, this representation was used for algorithmic purposes by [9] to speed-up the computation of a special kind of bivariate resultants, the composed products. For us, this is a typical example of an operation on univariate polynomials for which the Newton sums provide the appropriate data structure of the input polynomials. Other applications of Newton sums concern the computation of characteristic or minimal polynomials [19, 28, 31], the manipulations of symmetric functions in the context of algebraic elimination [18, 32, 11, 14, 12, 13, 25, 6], polynomial factorization [33], etc.

Let k be a field and let P be a polynomial of degree D in $k[X]$. The coefficients of P are uniquely determined by the first D power sums of its roots, at least if k has characteristic zero or larger than D. If k has positive characteristic p, the one-to-one correspondence between Newton sums and coefficients is no longer valid. However, under the stronger hypothesis that the multiplicities of all the roots of P are less then p, the first $2D$ power sums of P suffice to uniquely determine its coefficients.

From the computational point of view, one is interested to perform efficiently the conversions between the two representations. Newton formulas, which relate coefficients to power sums of roots, provide a straightforward algorithm to make these conversions, which has quadratic complexity in the degree D.

Fortunately, faster conversion methods exist. As far as we know, Schönhage [27] was the first to propose such a method, in the context of devising numerical root-finders for univariate polynomials. Over fields of characteristic zero, his algorithms extend to an exact setting as well, see [21, Appendix A] and [1, Problem 4.8]. Moreover, Schnhage's algorithm for translating a polynomial to its Newton representation remains valid over fields of any characteristic, while his algorithm for the converse direction works over fields of characteristic large enough.

The question of converting power sums of roots to coefficients for polynomials over fields of small characteristic is more delicate and many efforts have been done to bypass its difficulty. Historically, two kinds of approaches were proposed: on the one hand, the techniques of *recursive triangulation* originated by [16], on the other hand, those using *fundamental sets of power sums* of [28, 1, 22, 24]. The best currently known solution is that of [24].

In this paper, we restrict to a class of fields of positive characteristic, including the finite field $\mathbb{F}_p$. We consider the following kind of situation: one is given some polynomials in $\mathbb{F}_p[X]$ and wants to perform an operation on them whose result also lies in $\mathbb{F}_p[X]$. We suppose moreover that for this operation the representation by Newton sums is appropriate, that is, it has a simple interpretation on the coordinates of the input polynomials in the new representation.

2 Notation

Arithmetic complexity. To give complexity estimates over a ring R, we introduce a function $\mathsf{M} : \mathbb{N} \to \mathbb{N}$ such that $\mathsf{M}(d)$ denotes the complexity of degree d polynomial multiplication in $R[X]$, in terms of number of operations $(+, -, \times)$ in R. We suppose that the *multiplication time* function M verifies the inequality $\mathsf{M}(d_1) + \mathsf{M}(d_2) \leq \mathsf{M}(d_1 + d_2)$ for all positive integers d_1 and d_2; in particular, the inequality $\mathsf{M}(d/2) \leq 1/2\,\mathsf{M}(d)$ holds for all $d \geq 1$. We also make the hypothesis that $\mathsf{M}(cd)$ is in $O(\mathsf{M}(d))$, for all $c > 0$. The basic examples we have in mind are *classical* multiplication, for which $\mathsf{M}(n) \in O(n^2)$, but also *fast* multiplication algorithms like Karatsuba's [17] with $\mathsf{M}(n) \in O(n^{1.59})$. Better, using Fast Fourier Transform (FFT) algorithms, we have $\mathsf{M}(n) \in O(n \log n)$ if the base ring has enough primitive roots of unity, and $\mathsf{M}(n) \in O(n \log n \log\log n)$ unconditionally [29, 26, 8]. Our references for matters related to polynomial arithmetic are the books [1, 7, 10].

Bit complexity. Let next $\mathsf{M}_{\text{int}} : \mathbb{N} \to \mathbb{N}$ denote the complexity of integer multiplication, such that two numbers of bit-size b can be multiplied in $\mathsf{M}_{\text{int}}(b)$ bit operations. Using FFT algorithms, $\mathsf{M}_{\text{int}}(b) \in O(b \log b \log\log b))$. The following lemma gives the main properties used in the sequel.

Lemma 1. *Let M in $\mathbb{N}$ and $R = \mathbb{Z}/M\mathbb{Z}$. Then:*

- *All operations $(+, -, \times)$ in R can be performed for $\mathsf{M}_{\text{int}}(\log M)$ bit operations.*
- *If a, b are in R and b divides a in R, then one can compute c in R such that $a = bc$ in $\mathsf{M}_{\text{int}}(\log M \log\log M)$ bit operations.*

Proof. See [10, Corollary 11.10]. □

Others. Let P be the power series $\sum_{i\geq 0} p_i X^i$. The notation $[P]_a^b$ designates the polynomial $\sum_{i=a}^{b} p_i X^i$.

3 Basics on Newton sums

We begin by recalling some useful results concerning Newton sums of polynomials. The next lemma gathers together the most basic facts that will be used in the sequel. These results are classical; for instance, the equalities in *(ii)* are commonly known as *Newton's formulas* and those of *(iv)* as *Waring's formulas* [20], see also [4, Ex. 6, §6, Ch. A IV].

Lemma 2. *Let F be a field, let $P = X^d + A_1X^{d-1} + \cdots + A_d$ be a monic polynomial in $F[X]$ and let $S_i = \sum_{P(x)=0} x^i$ be its Newton sums, for $i \geq 1$. Let P^* be the reciprocal polynomial of P. Then:*

(i) the power series expansion of $(P^*)'/P^*$ in $F[[X]]$ equals $-\sum_{i\geq 0} S_{i+1}X^i$.

(ii) for any $1 \leq i \leq d$, the equality $iA_i + S_1A_{i-1} + \cdots + S_i = 0$ holds.

(iii) for any $1 \leq i \leq d$, the Newton sums of the polynomial $X^i + A_1X^{i-1} + \cdots + A_i$ are $S_1, \ldots, S_i$ respectively.

(iv) for all $i \geq 1$, the following formula holds

$$\frac{S_i}{i} = \sum_{\substack{\mu_1\theta_1 + \cdots + \mu_r\theta_r = i \\ \mu_1 > 0,\ \ldots,\ \mu_r > 0 \\ \theta_r > \cdots > \theta_1 > 0}} (-1)^{\mu_1+\cdots+\mu_r} \frac{(\mu_1 + \cdots + \mu_r - 1)!}{\mu_1! \cdots \mu_r!} A_{\theta_1}^{\mu_1} \cdots A_{\theta_r}^{\mu_r}.$$

Proof. *(i)* Let $x_1, \ldots, x_d$ be the roots of P in an algebraic closure of F. Then $P^* = \prod_{i=1}^d (1 - x_iX)$, so its logarithmic derivative equals $\sum_{i=1}^d -\frac{x_i}{1-x_iX} = -\sum_{i=1}^d (x_i + x_i^2X + x_i^3X^2 + \ldots)$. The latter sum is equals $-\sum_{i\geq 0} S_{i+1}X^i$, and this proves the first assertion.

(ii) simply follows by equating the coefficients of X^{i-1} on both hands of the equality $(P^*)' = P^* \times (-\sum_{i\geq 0} S_{i+1}X^i)$ in $F[[X]]$.

(iii) is a direct consequence of *(ii)*.

(iv) We will show that the generating series of the sequences on both sides of the formula are equal. First, by formally integrating the equality at point *(i)*, we deduce that the generating series $\sum_{i\geq 1} \frac{S_i}{i} X^i$ equals $-\ln(P^*)$. On the other hand, using the Taylor expansion $\sum_{i\geq 1} (-1)^i X^i/i$ of $-\ln(1 + X)$, we see that

$$-\ln(P^*) = \sum_{\mu} \frac{(-1)^\mu (A_1X + \cdots + A_dX^d)^\mu}{\mu}.$$

Now, the multinomial formula shows that the monomials of degree i that appear in the right-hand of the previous sum are

$$(-1)^{\mu_1+\cdots+\mu_d} \binom{\mu_1 + \cdots + \mu_d}{\mu_1, \ldots, \mu_d} \frac{(A_1X)^{\mu_1} \cdots (A_dX^d)^{\mu_d}}{\mu_1 + \cdots + \mu_d}$$

for all d-tuples $(\mu_1, \ldots, \mu_d)$ of non-negative integers such that $i = \mu_1 + 2\mu_2 + \cdots + d\mu_d$. Up to a convenient renumbering of the indices, this finishes the proof of the last assertion. □

4 The fundamental inequality

Let p be a prime and $\mathbb{Z}_p$ the ring of p-adic integers and $\mathbb{Q}_p$ the field of p-adic numbers, the p-adic valuation being denoted by $v(\,\cdot\,)$. Note that $v(xy) = v(x) + v(y)$ for all $x, y \in \mathbb{Q}_p$ and also that v is non-archimedian, that is, $v(x+y) \geq \min\{v(x), v(y)\}$, for all $x, y \in \mathbb{Q}_p$.

Let us also define $w : \mathbb{N}_{>0} \to \mathbb{N}$ by $w(i) = \lfloor \log_p(i) \rfloor$; in other words, $w(i)$ is characterized by the inequalities $p^{w(i)} \leq i < p^{w(i)+1}$. Note that for $i \geq 1$ we have that $v(i) \leq w(i)$; note also that w is increasing and verifies the inequality $w(x) + w(y) \leq w(xy)$, for all $x, y \geq 1$.

We state now the crucial inequality on which are based all the results of this paper. Intuitively, it asserts, in its simpler form (see Corollary 1 below), that two polynomials over $\mathbb{Z}_p$ whose Newton sums are *close* with respect to the p-adic topology, also have their coefficients *almost* as close. In the sequel, we will actually need the stronger version that we state now.

Theorem 1. *Let P and Q in $\mathbb{Q}_p[X]$ be two monic polynomials of degree d, with*

$$P = X^d + A_1 X^{d-1} + \cdots + A_d, \quad Q = X^d + B_1 X^{d-1} + \cdots + B_d.$$

Let $(S_i)_{i\geq 1}$ and $(T_i)_{i \geq 1}$ be the Newton sums of P and Q:

$$S_i = \sum_{P(x)=0} x^i, \quad T_i = \sum_{Q(x)=0} x^i.$$

Let $1 \leq i \leq d$ and $\alpha \in \mathbb{N}$ be such that

- *$v(S_i - T_i) \geq \alpha$;*
- *for all $1 \leq j < i$, A_j and B_j are in $\mathbb{Z}_p$ and $v(A_j - B_j) \geq \alpha - w(j)$.*

Then the inequality $v(A_i - B_i) \geq \alpha - w(i)$ holds.

Proof. We start by a lemma; by convention, the empty product equals 1.

Lemma 3. *Let $r \geq 0$, let $X_1, \ldots, X_r$ and $Y_1, \ldots, Y_r$ be indeterminates over $\mathbb{Z}$. For any r-uple $(\mu_1, \ldots, \mu_r)$ in $\mathbb{N}^r$, the polynomial $X_1^{\mu_1} \cdots X_r^{\mu_r} - Y_1^{\mu_1} \cdots Y_r^{\mu_r}$ can be written $\sum_{i=1}^r \Delta_i (X_i - Y_i)$, with $\Delta_i \in \mathbb{Z}[X_1, \ldots, X_r, Y_1, \ldots, Y_r]$ for $i = 1, \ldots, r$.*

Proof. We proceed by induction on r, the case $r = 0$ being obvious. Suppose now that the result holds for some $r \geq 0$; we prove it for $r+1$. Let us then consider a product of the form $X_1^{\mu_1} \cdots X_{r+1}^{\mu_{r+1}} - Y_1^{\mu_1} \cdots Y_{r+1}^{\mu_{r+1}}$. We rewrite it as

$$X_1^{\mu_1} \cdots X_r^{\mu_r} (X_{r+1}^{\mu_{r+1}} - Y_{r+1}^{\mu_{r+1}}) + Y_{r+1}^{\mu_{r+1}} (X_1^{\mu_1} \cdots X_r^{\mu_r} - Y_1^{\mu_1} \cdots Y_r^{\mu_r}).$$

To deal with the first term, we rewrite $(X_{r+1}^{\mu_{r+1}} - Y_{r+1}^{\mu_{r+1}})$ as $(X_{r+1} - Y_{r+1}) \sum_{i=0}^{\mu_{r+1}-1} X_{r+1}^i Y_{r+1}^{\mu_{r+1}-i-1}$. The second term is dealt with using the induction hypothesis. □

Applying the last part of Lemma 2 for both polynomials P and Q, we obtain by subtraction

$$\frac{S_i - T_i}{i} = \sum_{\substack{\mu_1\theta_1 + \cdots + \mu_r\theta_r = i \\ \mu_1 > 0, \ \ldots, \ \mu_r > 0 \\ \theta_r > \cdots > \theta_1 > 0}} (-1)^{\mu_1 + \cdots + \mu_r} \frac{(\mu_1 + \cdots + \mu_r - 1)!}{\mu_1! \cdots \mu_r!} (A_{\theta_1}^{\mu_1} \cdots A_{\theta_r}^{\mu_r} - B_{\theta_1}^{\mu_1} \cdots B_{\theta_r}^{\mu_r}).$$

In this sum, the term corresponding to $r = 1$, $\theta_r = i$, $\mu_r = 1$ equals $-(A_i - B_i)$. Rewriting the formula to isolate this term shows that $A_i - B_i$ is given by

$$\frac{T_i - S_i}{i} + \sum_{\substack{\mu_1\theta_1 + \cdots + \mu_r\theta_r = i \\ \mu_1 > 0,\ \ldots,\ \mu_r > 0 \\ i > \theta_r > \cdots > \theta_1 > 0}} (-1)^{\mu_1+\cdots+\mu_r} \frac{(\mu_1 + \cdots + \mu_r - 1)!}{\mu_1! \cdots \mu_r!} (A_{\theta_1}^{\mu_1} \cdots A_{\theta_r}^{\mu_r} - B_{\theta_1}^{\mu_1} \cdots B_{\theta_r}^{\mu_r}). \quad (1)$$

Now, using Lemma 3, and the fact that the coefficients $A_1, \ldots, A_{i-1}$ and $B_1, \ldots, B_{i-1}$ are in $\mathbb{Z}_p$, we remark that any term of the form $A_{\theta_1}^{\mu_1} \cdots A_{\theta_r}^{\mu_r} - B_{\theta_1}^{\mu_1} \cdots B_{\theta_r}^{\mu_r}$ belongs to $\mathbb{Z}_p[A_{\theta_1} - B_{\theta_1}, \ldots, A_{\theta_r} - B_{\theta_r}]$. Thus, the sum in Formula (1) can be written

$$\sum_{\theta,\mu} (-1)^{\mu_1+\cdots+\mu_r} \frac{(\mu_1 + \cdots + \mu_r - 1)!}{\mu_1! \cdots \mu_r!} \left(c_1^{(\theta,A,B)}(A_{\theta_1} - B_{\theta_1}) + \cdots + c_r^{(\theta,A,B)}(A_{\theta_r} - B_{\theta_r})\right), \quad (2)$$

for some coefficients $c_i^{(\theta,A,B)} \in \mathbb{Z}_p$, and where $\theta = (\theta_1, \ldots, \theta_r)$ and $\mu = (\mu_1, \ldots, \mu_r)$ satisfy the same conditions as in Formula (1). Regrouping terms, we deduce an equality of the form

$$A_i - B_i = \frac{T_i - S_i}{i} + \sum_{1 \le j < i} \gamma_j (A_j - B_j), \quad (3)$$

for some coefficients $\gamma_j \in \mathbb{Q}_p$. Let then j be in $1, \ldots, i - 1$; we now proceed to estimate the valuation of $\gamma_j(A_j - B_j)$. To this effect, we note that the term $A_j - B_j$ possibly appears several times in the sum of Formula (2): we inspect each of its occurrences.

Let thus $\theta = (\theta_1, \ldots, \theta_r)$ and $\mu = (\mu_1, \ldots, \mu_r)$, with $0 < \theta_1 < \cdots < \theta_r < i$, $\mu_k > 0$ for $1 \le k \le r$ and $\mu_1\theta_1 + \cdots + \mu_r\theta_r = i$, be such that $\theta_k = j$ for some $1 \le k \le r$. The corresponding coefficient μ_k then satisfies the inequality $j\mu_k = \theta_k\mu_k \le i$. The inequality $w(x) + w(y) \le w(xy)$, valid for all $x, y \ge 1$, shows that $w(j) + w(\mu_k) \le w(j\mu_k)$. Since w is increasing, we deduce that $w(\mu_k)$ is bounded from above by $w(i) - w(j)$, which implies that $v(\mu_k)$ admits the same upper bound.

Now, the corresponding term in Formula (2) is (up to sign)

$$\frac{(\mu_1 + \cdots + \mu_r - 1)!}{\mu_1! \cdots \mu_r!} c_k^{(\theta,A,B)}(A_j - B_j),$$

which we rewrite as

$$\binom{\mu_1 + \cdots + \mu_r - 1}{\mu_1, \ldots, \mu_k - 1, \ldots, \mu_d} c_k^{(\theta,A,B)} \frac{A_j - B_j}{\mu_k} = \delta_k^{(\mu,\theta,A,B)} \frac{A_j - B_j}{\mu_k},$$

where $\delta_k^{(\mu,\theta,A,B)}$ belongs to $\mathbb{Z}_p$. Due to our previous upper bound on $v(\mu_k)$, the valuation of the above term admits the lower bound $v(A_j - B_j) - w(i) + w(j)$; by our assumptions, this is bounded from below by $\alpha - w(i)$. Summing all such contributions, and using that v is non-archimedian, we deduce that the valuation of $\gamma_j(A_j - B_j)$ admits the same lower bound.

Next, recall that by assumption, $v(\frac{T_i - S_i}{i}) = v(T_i - S_i) - v(i) \ge \alpha - v(i)$; since $v(i) \le w(i)$, we obtain the inequality $v(\frac{T_i - S_i}{i}) \ge \alpha - w(i)$. Finally, we can deduce from Equation (3) and the above discussion the inequality $v(A_i - B_i) \ge \alpha - w(i)$. This concludes the proof. □

A straightforward consequence of Theorem 1 is the following corollary which will not be used in the sequel, but which merits however to be stated. Intuitively, it asserts that two polynomials over $\mathbb{Z}_p$ whose Newton sums are close with respect to the p-adic topology, also have their coefficients almost as close.

Corollary 1. *Let P and Q in $\mathbb{Z}_p[X]$ two monic polynomials of degree d, with*

$$P = X^d + A_1X^{d-1} + \cdots + A_d, \quad Q = X^d + B_1X^{d-1} + \cdots + B_d.$$

Let $(S_i)_{i\geq 1}$ and $(T_i)_{1\leq i\leq d}$ be the Newton sums of P and Q and let $\alpha \in \mathbb{N}$ be such that

$$v(S_i - T_i) \geq \alpha, \quad \textit{for all}\ 1 \leq i \leq d.$$

Then the inequality $v(A_i - B_i) \geq \alpha - w(i)$ holds for all $i \geq 1$.

5 Algorithms

Let P be in $\mathbb{Z}_p[X]$, with $P = X^d + \mathfrak{A}_1X^{d-1} + \cdots + \mathfrak{A}_d$, let $(S_i)_{i\geq 1}$ be its Newton sums. The aim of this section is to propose efficient algorithms which, given the Newton sums of P truncated at a certain precision $\alpha > 0$, recover the coefficients of P at some smaller, but still *positive* precision.

An immediate algorithm allows to do this via Newton's formulas, provided that we work at initial precision $\alpha = O(d)$. Indeed, the ith Newton formula (see Lemma 2, *ii*) involves a division by i, so that after using it to compute $\mathfrak{A}_i$, a precision $v(i)$ is potentially lost. Since $\sum_{i=1}^{d} v(i) = v(d!) \approx d/(p-1)$, it may thus appear that an initial precision of computations *linear in d* is needed to ensure that, at the end, all the $\mathfrak{A}_i$'s are recovered at some positive precision. In terms of computational amount of work, this algorithm would require $O(d^2)$ operations $(+, -, \times, \div)$ in $\mathbb{Z}/p^\alpha\mathbb{Z}$, where $\alpha = O(d/p)$, hence a bit-complexity at least of order $O(d^2\, \mathsf{M}_{\text{int}}(\frac{d}{p}\log p))$ after Lemma 1.

In this section we propose two faster algorithms. Our first contribution is to show that the algorithm sketched above (and based on Newton's relations) already works if we compute only at precision $O(\log_p(d))$. This is a non-trivial and quite surprising result, based on Theorem 1. As a direct consequence, the resulting algorithm has a smaller bit-complexity, that is, within $O(d^2\, \mathsf{M}_{\text{int}}(\log d))$ bit operations. Our second algorithm relies on the use of a formal Newton operator with quadratic convergence and is designed to be employed in conjunction with the fast multiplication routines. This algorithm also requires to work only in precision $O(\log_p(d))$. Its bit-complexity is in $O(\mathsf{M}(d)\, \mathsf{M}_{\text{int}}(\log d))$. Thus, if the FFT-based polynomial multiplication is used, it is faster than the first algorithm by an order of magnitude.

In what follows we use the following limpid notations: we let $\alpha \geq w(d)$ be the precision of computations and let $R = \mathbb{Z}_p/p^\alpha\mathbb{Z}_p$ be the ring of p-adic integers truncated at precision α. Since it is isomorphic to $\mathbb{Z}/p^\alpha\mathbb{Z}$, we will make no distinction between these two rings. For $i \geq 1$, we define s_i and $\mathfrak{a}_i$ in R by $s_i = S_i \mod p^\alpha$ and $\mathfrak{a}_i = \mathfrak{A}_i \mod p^\alpha$.

5.1 The quadratic algorithm

The first result proves that the loss of precision during the successive divisions by $1, 2, \ldots, d$ in the algorithm based on Newton formulas does not exceed $w(d)$. This is quite surprising, since it replaces the *a priori* pessimistic bound $v(d!)$, which is linear in d/p, by a bound which is linear in $\log_p(d)$. From the computational view-point, this enables to save precision and thus to improve the bit-complexity of the algorithm based on Newton formulas.

Theorem 2. *Let $1 \leq k < d$ and let $a_1, \ldots, a_k$ be in R such that the equality $ia_i + s_1a_{i-1} + \cdots + s_i = 0$ holds for $i = 1, \ldots, k$. Then:*

(i) There exists a_{k+1} in R such that $(k+1)a_{k+1} + s_1a_k + \cdots + s_{k+1} = 0$.

(ii) Any such a_{k+1} satisfies the equality $a_{k+1} = \mathfrak{a}_{k+1} \mod p^{\alpha - w(k+1)}$.

Proof. Let $A_1, \dots, A_k$ be arbitrary lifts of $a_1, \dots, a_k$ in $\mathbb{Z}_p$, and let $A_{k+1} \in \mathbb{Q}_p$ be defined by the relation $(k+1)A_{k+1} + S_1 A_k + \cdots + S_{k+1} = 0$. To prove point *(i)*, we show that A_{k+1} is in $\mathbb{Z}_p$; this will be done by proving the stronger assertion that A_{k+1} satisfies the inequality $v(A_{k+1} - \mathfrak{A}_{k+1}) \geq \alpha - w(k+1)$. To this effect, we introduce two polynomials P_k and Q, and consider their Newton sums:

- Let first P_k be the polynomial $X^{k+1} + \mathfrak{A}_1 X^k + \cdots + \mathfrak{A}_{k+1}$; by Lemma 2, the first $k+1$ Newton sums of P_k are $S_1, \dots, S_{k+1}$.
- Let next $Q \in \mathbb{Q}_p[X]$ be the polynomial $X^{k+1} + A_1 X^k + \cdots + A_{k+1}$ and let $T_1, \dots, T_{k+1}$ be its first $k+1$ Newton sums. Then, for $i = 1, \dots, k+1$ we have simultaneously

  $$v(iA_i + S_1 A_{i-1} + \cdots + S_i) \geq \alpha \quad \text{and} \quad iA_i + T_1 A_{i-1} + \cdots + T_i = 0.$$

  Since A_i is in $\mathbb{Z}_p$ for $i = 1, \dots, k$, we deduce by successive subtractions that $v(S_i - T_i) \geq \alpha$ for $i = 1, \dots, k+1$.

Applying inductively Theorem 1 to P_k and Q for all indices $i = 1, \dots, k+1$, we obtain the inequalities $v(A_i - \mathfrak{A}_i) \geq \alpha - w(i)$. In particular, since $\alpha \geq w(k+1)$, we deduce that A_{k+1} is in $\mathbb{Z}_p$.

Let then $a_{k+1} = A_{k+1} \mod p^\alpha$: we see that $(k+1)a_{k+1} + s_1 a_k + \cdots + s_{k+1} = 0$, proving point *(i)*, and that furthermore $a_{k+1} = \mathfrak{a}_{k+1} \mod p^{\alpha - w(k+1)}$.

We now prove point *(ii)*. Let then $\overline{a}_{k+1}$ be any element of R such that $(k+1)\overline{a}_{k+1} + s_1 a_k + \cdots + s_{k+1} = 0$. By subtraction, we deduce that $(k+1)(\overline{a}_{k+1} - a_{k+1}) = 0$. This implies that $p^{v(k+1)}(\overline{a}_{k+1} - a_{k+1}) = 0$; since $v(k+1) \leq w(k+1)$, we deduce that $p^{w(k+1)}(\overline{a}_{k+1} - a_{k+1}) = 0$. In other words, $\overline{a}_{k+1} = a_{k+1} \mod p^{\alpha - w(k+1)}$; since we also have $a_{k+1} = \mathfrak{a}_{k+1} \mod p^{\alpha - w(k+1)}$, we obtain $\overline{a}_{k+1} = \mathfrak{a}_{k+1} \mod p^{\alpha - w(k+1)}$, concluding the proof. □

Corollary 2. *Let $f \in \mathbb{Z}_p[X]$ be a monic polynomial of degree d. Knowing the first d Newton sums of f at precision $\alpha = \lfloor \log_p(d) \rfloor + 1$, its coefficients can be computed at precision 1 using*

$$O\left(d^2\, \mathsf{M}_{\text{int}}(\log d)\right)$$

bit operations.

Proof. By Theorem 2, the algorithm described in Figure 1 is correct and requires $O(d^2)$ operations $(+, -, \times)$ in $\mathbb{Z}/p^\alpha\mathbb{Z}$ and d exact divisions. Lemma 1 finishes the proof. □

5.2 The fast algorithm

We present now a fast algorithm to perform the conversion between Newton sums and coefficients. It is based on the classical fact that this conversion mainly amounts to exponentiate the primitive of the generating series of the Newton sums (as can be seen from point *(ii)* of Lemma 2). Now, nearly optimal algorithms to compute exponentials of power series have been given by [5] (and already used in [27] for this purpose). They use a formal version of Newton's operator, whose quadratic convergence ensures the good complexity behaviour. Thus, in principle, a fast version of our algorithm is possible, the only problem concerns the divisions occurring during the algorithm. More specifically, each iteration of Newton's operator involves a primitive computation. Thus, in order to control the loss of precision, we must deal with these divisions.

The following theorem shows that, as in the case of the quadratic algorithm, a precision of at most $w(d)$, that is, logarithmic in d, is lost during the Newton's iterations. This result allows to obtain a faster (in terms of bit-operations) algorithm than the algorithm described in Section 5.1.

Recovering a monic polynomial
from its Newton series

Input: the Newton sums $s_1, \ldots, s_d$ of f in $R = \mathbb{Z}_p/p^\alpha\mathbb{Z}_p$
Output: the coefficients of f in $\mathbb{F}_p = \mathbb{Z}_p/p\mathbb{Z}_p$.

```
$a_1 \leftarrow -s_1$
for $k$ from $1$ to $d-1$ do
    $a_{k+1} \leftarrow -\frac{1}{k+1}\big(s_1 a_k + \cdots + s_k a_1 + s_{k+1}\big)$
return the image of $F = X^d + \sum_{i=1}^{d} a_i X^{d-i}$ in $\mathbb{F}_p[X]$
```

Figure 1: Recovering a monic polynomial from its Newton sums in small characteristic using Newton's formulas

Theorem 3. *Let $k \geq 1$ and let $a_1, \ldots, a_k$ be in R such that $a_i = \mathfrak{a}_i \mod p^{\alpha - w(i)}$ for $1 \leq i \leq k$. Let $K = \min(2k+1, d)$ and let g and δ in $R[X]$ be the polynomials*

$$g = 1 + a_1 X + \cdots + a_k X^k, \quad \delta = -\sum_{i=k}^{K-1} s_{i+1} X^i - \left[\frac{g'}{g}\right]_k^{K-1}.$$

Then:

(i) There exists $h \in R[X]$ such that $h' = \delta$.

(ii) Let h be such that $h' = \delta$ and $h(0) = 0$, and let $f = g(1+h) \mod X^{K+1}$ in $R[X]$. Then f has the form $1 + a_1 X + \cdots + a_K X^K$, and a_i satisfies the equality $a_i = \mathfrak{a}_i \mod p^{\alpha - w(i)}$ for $1 \leq i \leq K$.

Proof. First, we prove the existence of h in $R[X]$ which simultaneously satisfies the conditions of points *(i)* and *(ii)*. To this effect, let G be a lift of g in $\mathbb{Z}_p[X]$ of the form $1 + A_1 X + \cdots + A_k X^k$, and let

$$\Delta = -\sum_{i=k}^{K-1} S_{i+1} X^i - \left[\frac{G'}{G}\right]_k^{K-1} \in \mathbb{Z}_p[X];$$

note that $\delta = \Delta \mod p^\alpha$. Let now $H \in \mathbb{Q}_p[X]$ be such that $H' = \Delta$ and $H(0) = 0$; let also $F = G(1+H) \mod X^{K+1} \in \mathbb{Q}_p[X]$, and note that F has the form $1 + A_1 X + \cdots + A_K X^K$. To establish our claim, it suffices to prove that:

- H and F are in $\mathbb{Z}_p[X]$;
- $v(A_i - \mathfrak{A}_i) \geq \alpha - w(i)$ for $i = 1, \ldots, K$.

Indeed, the polynomials $h = H \mod p^\alpha$ and $f = F \mod p^\alpha$ are then seen to satisfy the conditions of point *(ii)*.

Let us consider the logarithmic derivative of F in $\mathbb{Q}_p[[X]]$: first, the definition $F = G(1+H) \mod X^{K+1}$ yields the equality

$$\frac{F'}{F} = \frac{G'}{G} + \frac{H'}{1+H} \mod X^K.$$

Since $H' = \Delta$ and H has valuation at least $k+1$, this implies

$$\frac{F'}{F} = \frac{G'}{G} + \Delta \mod X^K;$$

in particular, considering the coefficients of degrees $k, \dots, K-1$ shows that

$$\left[\frac{F'}{F}\right]_k^{K-1} = -\sum_{i=k}^{K-1} S_{i+1} X^i. \tag{4}$$

We now define two polynomials F^* and P_k, and study their Newton sums:

- Let first P_K be the polynomial $X^K + \mathfrak{A}_1 X^{K-1} + \dots + \mathfrak{A}_K$; by Lemma 2, the first K Newton sums of P_K are $S_1, \dots, S_K$.
- Let next $F^* = X^K + A_1 X^{K-1} + \dots + A_K$ be the reciprocal polynomial of F and let $T_1, \dots, T_K$ be its first K Newton sums. In view of Equation (4), we deduce from Lemma 2 that $T_i = S_i$ for $i = k+1, \dots, K$.

Thus, the Newton sums of P_K of indices $k+1, \dots, K$ coincide with those of F^*. On the other hand, our assumption on $a_1, \dots, a_k$ shows that $v(A_i - \mathfrak{A}_i) \geq \alpha - w(i)$ for $i = 1, \dots, k$. We then apply inductively Theorem 1 to F^* and P_K, for $i = k+1, \dots, K$: for any such i, this yields the inequality $v(A_i - \mathfrak{A}_i) \geq \alpha - w(i)$, and thus that A_i is in $\mathbb{Z}_p$, since $\alpha \geq w(i)$.

Thus, F is in $\mathbb{Z}_p[X]$. Since G is invertible in $\mathbb{Z}_p[[X]]$, we deduce that H is in $\mathbb{Z}_p[X]$ as well. Letting $h = H \mod p^\alpha$, we see that h satisfies the conditions of points *(i)* and *(ii)*.

Let us now complete the proof, by establishing point *(ii)*. Let thus $\overline{h}$ be any polynomial in $R[X]$ such that $\overline{h}(0) = 0$ and $\overline{h}' = \delta$, and let $\overline{f} = g(1 + \overline{h}) \mod X^{K+1}$, so that the difference $\overline{f} - f$ equals $g(\overline{h} - h) \mod X^{K+1}$.

The polynomial $\lambda = \overline{h} - h$ satisfies $\lambda' = 0$ in $R[X]$; writing $\lambda = \sum_{i=k+1}^K \lambda_i X^i$, this means that λ_i satisfies $p^{v(i)} \lambda_i = 0$ for $i = k+1, \dots, K$, and thus $p^{w(i)} \lambda_i = 0$ as well since $v(i) \leq w(i)$. Let us write $\overline{f} - f = \sum_{i=k+1}^K \Lambda_i X^i$; since $\overline{f} - f = g(\overline{h} - h) \mod X^{K+1}$ and w is increasing, we see by expanding the product that $p^{w(i)} \Lambda_i = 0$ for $i = k+1, \dots, K$. This finishes the proof of the theorem. □

Corollary 3. *Let $f \in \mathbb{Z}_p[X]$ be a monic polynomial of degree d. Knowing the first d Newton sums of f at precision $\alpha = \lfloor \log_p(d) \rfloor + 1$, its coefficients can be computed at precision 1 using*

$$O\big(\mathsf{M}(d)\, \mathsf{M}_{\mathrm{int}}(\log d)\big)$$

bit operations.

Proof. Theorem 3 shows that the algorithm described in Figure 2 is correct and that it requires $\sum_{i=1}^{\log_2(d)} O(\mathsf{M}(2^i)) = O(\mathsf{M}(d))$ operations $(+, -, \times)$ and $O(d)$ exact divisions in $\mathbb{Z}/p^\alpha\mathbb{Z}$. The result follows from Lemma 1. □

Remark. Without using Theorem 3, we are able to prove only a weaker version of Corollary 3. That version is not sufficient to our purposes, since it would require as input the first d Newton sums of f at precision $\alpha' = O(\log^2 d)$ instead of $\alpha = O(\log d)$ and would have a bit-complexity upper bounded by $O\big(\mathsf{M}(d)\, \mathsf{M}_{\mathrm{int}}(\log^2 d)\big)$. Indeed, an immediate count of the divisions performed by the fast algorithm in Figure 2 shows that at the ith iteration, only divisions by $1, 2, \dots, 2^i$ occur, so a precision of at most $w(2^i)$ could be lost. Thus, an *a priori* bound of the total needed precision would be $\sum_{i \leq \log d} w(2^i)$, which is in $O\big(\log^2 d / \log_p(d)\big)$. As in the case of the quadratic algorithm, these bounds are pessimistic compared to those provided by the more refined results in Corollary 3.

Recovering a monic polynomial from its Newton series

Input: the Newton sums $s_1, \ldots, s_d$ of f in $R = \mathbb{Z}_p/p^\alpha\mathbb{Z}_p$
Output: the coefficients of f in $\mathbb{F}_p = \mathbb{Z}_p/p\mathbb{Z}_p$

```
$S \leftarrow s_1 + s_2T + \cdots + s_dT^{d-1}$
$g \leftarrow 1$
prec $\leftarrow 1$
while prec $\leq d-1$ do
    $\delta \leftarrow -\left[\frac{g'}{g} + S\right]_{\text{prec}/2}^{\text{Min}(\text{prec},d-1)}$
    $h \leftarrow \sum_{i\geq 1}$ Coeff$(\delta, i-1)\frac{T^i}{i}$
    $g \leftarrow \lceil g(1+h)\rceil^{2\times\text{prec}}$
    prec $\leftarrow 2 \times$ prec
$f \leftarrow \text{rev}(g)$
return the image of $F$ in $\mathbb{F}_p[X]$
```

Figure 2: Recovering a monic polynomial from its Newton sums in small characteristic using Newton's iteration

5.3 Previous work

Let us compare the complexity of our second algorithm to the other results existing in the literature. The previous fastest known algorithm for recovering a polynomial from its Newton sums in small characteristic is due to Schönhage and Pan [28, 22, 23, 24].

As input, this algorithm requires the first $2d$ Newton sums of f; it works under the assumption that all roots of f have multiplicities less than p. This algorithm decomposes in two steps: the first one is quite similar to the fast exponentiation algorithm of [5, 27], which is also the base of our algorithm; the second step consists in $p-1$ Padé approximant computations in degrees $(d/p, d/p)$. Thus, its arithmetic complexity is $O(\mathsf{M}(d) + p\,\mathsf{M}(d/p)\log(d/p))$ operations in $\mathbb{F}_p$, and its bit-complexity is within

$$\Big(O\big(\mathsf{M}(d) + p\,\mathsf{M}(d/p)\log(d/p)\big)\Big)\,\mathsf{M}_{\text{int}}(\log p).$$

To simplify the comparison, let us suppose that the FFT is used and that $\mathsf{M}(d) = O(d\log d)$. Thus, the cost of our algorithm is within $O(d\log^2 d\log\log d)$ bit operations, while that of Schönhage-Pan's algorithm is of $O(d\log^2(d/p)\log p\log\log p)$ bit-operations. Let $p = d^\beta$, with $0 < \beta < 1$. Then the ratio between the two costs equals

$$\frac{d(1-\beta)^2\log^2(d)\beta\log d\log(\beta\log d)}{d\log^2 d\log\log d} \approx \beta(1-\beta)^2\log d.$$

This estimate shows that is β varies in a fixed range $0 < \beta_0 \leq \beta \leq \beta_1 < 1$, our algorithm is faster roughly by a factor of $\log d$. On the other hand, for $\beta \simeq 0$, that is, for constant p, the bit complexity of Schönhage-Pan's algorithm is within $O(\mathsf{M}(d)\log d)$, which is better than ours by a $\log\log d$ factor. Similarly, for $\beta \simeq 1$, that is, when the ratio d/p is constant, the bit complexity of Schönhage-Pan's algorithm is within $O(\mathsf{M}(d)\mathsf{M}_{\mathrm{int}}(\log d))$, which is in the same complexity class as our estimates.

One should keep in mind that the input of the two algorithms differ. The next section will compare the behavious of the two algorithms for a classical application, composed products.

6 Application: composed products

Let k be a field and let f and g be monic polynomials in $k[T]$, of degrees m and n respectively. We are interested in computing efficiently their *composed product* $f \otimes g$. This is the polynomial of degree $D = mn$ defined by

$$f \otimes g = \prod_{\alpha,\beta} (T - \alpha\beta),$$

the product running over all the roots α of f and β of g, counted with multiplicities, in an algebraic closure $\overline{k}$ of k. The algorithm of Figure 3 was initially proposed by Dvornicich and Traverso in characteristic zero; its arithmetic complexity is $O(\mathsf{M}(D))$ operations in k.

```
Composed product

Input: $f$ and $g$ in $k[T]$
Output: $f \otimes g$

$D \leftarrow \deg(f)\deg(g)$
$S \leftarrow [\mathrm{Newton}_i(f) : i \leq D]$
$T \leftarrow [\mathrm{Newton}_i(g) : i \leq D]$
$U \leftarrow [S_i T_i : i \leq D]$
Recover $f \otimes g$ from its first $D$ Newton sums $U$
```

Figure 3: Computing a composed product

If k has positive characteristic, then the first D Newton sums of $f \otimes g$ are not enough to recover it. In [2], it is proposed to use Schönhage-Pan's algorithm, which requires to compute the first $2D$ Newton sums of $f \otimes g$, and the assumption that the multiplicities of the roots of $f \otimes g$ are less than the characteristic of k. The arithmetic complexity of this algorithm is dominated by this last part, that is,within $O(\mathsf{M}(D) + p\mathsf{M}(D/p)\log(D/p))$ operations in k. If $k = \mathbb{F}_p$, then the bit complexity is in $\big(O(\mathsf{M}(D) + p\,\mathsf{M}(D/p)\log(D/p))\big)\,\mathsf{M}_{\mathrm{int}}(\log p)$.

Now, our algorithm can be used as well in the case $k = \mathbb{F}_p$. The algorithm is a straightforward consequence of the previous considerations; an important difference is that we now have to compute the Newton sums of (arbitrary lifts of) f and g in $\mathbb{Z}/p^\alpha\mathbb{Z}[T]$ instead of $\mathbb{F}_p[T]$. As for the Schönhage-Pan's approach, the predominant part in the complexity estimate is the last step; thus, the whole bit-complexity of this approach is $\big(O(\mathsf{M}(D)\mathsf{M}_{\mathrm{int}}(\log D))\big)$ bit operations. In particular, the expected ratio of $O(\log D) = O(\log d)$ obtained in the previous section when $p = d^\beta$ and β varies in $[\beta_0, \beta_1]$ carries over to this situation.

Experimental results We have implemented our fast algorithm for conversion between Newton sums and coefficients and applied it to the computation of the composed product of polynomials and compared it to the strategy using Schönhage-Pan's algorithm.

```
Composed product

Input: $f$ and $g$ in $\mathbb{F}_p[T] = \mathbb{Z}/p\mathbb{Z}[T]$
Output: $f \otimes g$ in $\mathbb{F}_p[T] = \mathbb{Z}/p\mathbb{Z}[T]$

$\alpha \leftarrow \lfloor \log_p(d) \rfloor + 1$
$R \leftarrow \mathbb{Z}_p/p^\alpha\mathbb{Z}_p = \mathbb{Z}/p^\alpha\mathbb{Z}$
Lift $f$ and $g$ to $R[T]$
$D \leftarrow \deg(f)\deg(g)$
$S \leftarrow [\mathrm{Newton}_i(f) : i \leq D]$
$T \leftarrow [\mathrm{Newton}_i(g) : i \leq D]$
$U \leftarrow [S_iT_i : i \leq D]$
Recover $f \otimes g$ from its first $D$ Newton sums $U$ using Algorithm of Figure 2, and reduce it mod $p$.
```

Figure 4: Computing a composed product in small characteristic

We used the NTL C++ package [30]. The computations are done over base rings of type $\mathbb{Z}/m\mathbb{Z}$, where m is a prime power; over such rings, NTL implements polynomial arithmetic using classical, Karatsuba and FFT multiplications.

Both algorithms share (up to minor modifications) a common exponentiation step based on a Newton iteration; this step uses the middle product techniques introduced in [15], for which we relied on the implementation presented in [3]. Schönhage-Pan's algorithm also requires Padé approximant computations; we used NTL's built-in fast extended GCD routines for this purposes.

For the tests presented in Figure 5, the input is formed by two polynomials of equal degree d, whose coefficients are randomly picked in $\mathbb{F}_p$, where p is chosen to be the smaller prime greater than d. Note that the output $f \otimes g$ has degree $D = d^2$. We let d vary from 10 to 200, so that D varies from 100 to 40000. All tests were performed on a 500 MB, 2GHz 32 bit Intel processor.

References

[1] D. Bini and V. Y. Pan. *Polynomial and matrix computations. Vol. 1.* Birkhäuser Boston Inc., Boston, MA, 1994.

[2] A. Bostan, Ph. Flajolet, B. Salvy, and É. Schost. Fast computation with two algebraic numbers. Research Report 4579, Institut National de Recherche en Informatique et en Automatique, October 2002. 20 pages.

[3] A. Bostan, G. Lecerf, and É. Schost. Tellegen's principle into practice. In *ISSAC'03*, pages 37–44. ACM Press, 2003.

[4] N. Bourbaki. *Éléments de mathématique.* Masson, Paris, 1981. Algèbre. Chapitres 4 à 7.

[5] R. P. Brent. Multiple-precision zero-finding methods and the complexity of elementary function evaluation. In *Analytic computational complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1975)*, pages 151–176. Academic Press, New York, 1976.

[6] E. Briand and L. González-Vega. Multivariate Newton sums: identities and generating functions. *Communications in Algebra*, 30(9):4527–4547, 2002.

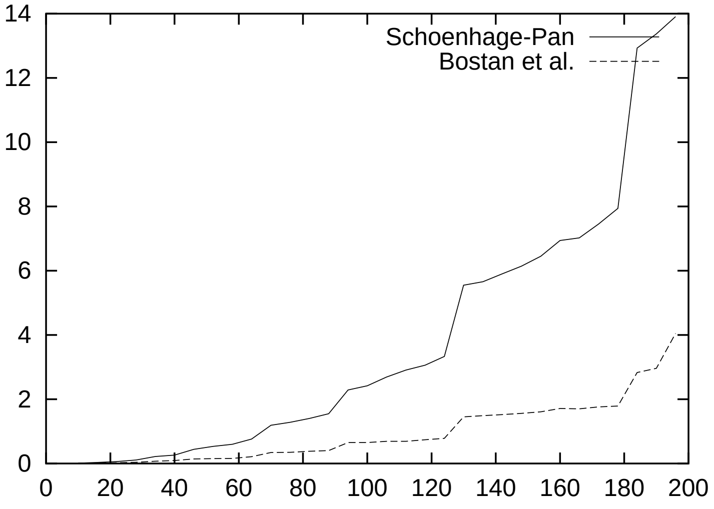

Figure 5: Composed product of polynomials over fields of small characteristic. (Input degree on the horizontal axis and time in seconds on the vertical axis.)

[7] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren Math. Wiss.* Springer–Verlag, 1997.

[8] D. G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Inform.*, 28(7):693–701, 1991.

[9] R. Dvornicich and C. Traverso. Newton symmetric functions and the arithmetic of algebraically closed fields. In *AAECC-5 (Menorca, 1987)*, volume 356 of *LNCS*, pages 216–224. Springer, Berlin, 1989.

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

[11] M. Giusti, D. Lazard, and A. Valibouze. Algebraic transformations of polynomial equations, symmetric polynomials and elimination. In P. Gianni, editor, *Proceedings of ISSAC'88*, volume 358 of *LNCS*, pages 309–314. Springer Verlag, 1989.

[12] M.-J. González-López and L. González-Vega. Newton identities in the multivariate case: Pham systems. In *Gröbner bases and applications (Linz, 1998)*, volume 251 of *London Math. Soc. Lecture Note Ser.*, pages 351–366. Cambridge Univ. Press, Cambridge, 1998.

[13] L. González-Vega and G. Trujillo. Implicitization of parametric curves and surfaces by using symmetric functions. In *Proceedings of ISSAC'95*, pages 180–186. ACM Press, 1995.

[14] L. González-Vega and G. Trujillo. Using symmetric functions to describe the solution set of a zero-dimensional ideal. In *AAECC-11 (Paris, 1995)*, volume 948 of *LNCS*, pages 232–247. Springer, Berlin, 1995.

[15] G. Hanrot, M. Quercia, and P. Zimmermann. The Middle Product Algorithm, I. *Appl. Algebra Engrg. Comm. Comput.*, 14(6):415–438, 2004.

[16] E. Kaltofen and V. Y. Pan. Parallel solution of Toeplitz and Toeplitz-like linear systems over fields of small positive characteristic. In *First International Symposium on Parallel Symbolic Computation—PASCO '94 (Hagenberg/Linz, 1994)*, volume 5 of *Lecture Notes Ser. Comput.*, pages 225–233. World Sci. Publishing, River Edge, NJ, 1994.

[17] A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Soviet Math. Dokl.*, 7:595–596, 1963.

[18] A. Lascoux. La résultante de deux polynômes. In *Séminaire d'algèbre Paul Dubreil et Marie-Paule Malliavin, 37ème année (Paris, 1985)*, volume 1220 of *Lecture Notes in Math.*, pages 56–72. Springer, Berlin, 1986.

[19] U. J. J. Le Verrier. Sur les variations séculaires des éléments elliptiques des sept planètes principales : Mercure, Venus, La Terre, Mars, Jupiter, Saturne et Uranus. *J. Math. Pures Appli.*, 4:220–254, 1840.

[20] M. P. Macmahon. *Combinatory analysis.* Reprinted by Chelsea Publ. Company, 1960.

[21] V. Y. Pan. Parallel least-squares solution of general and Toeplitz-like linear systems. In *Proc. 2nd Ann. ACM Symp. on Parallel Algorithms and Architecture*, pages 244–253. ACM Press, 1990.

[22] V. Y. Pan. Parallel computation of polynomial GCD and some related parallel computations over abstract fields. *Theoretical Computer Science*, 162(2):173–223, 1996.

[23] V. Y. Pan. Faster solution of the key equation for decoding BCH error-corecting codes. In *Proceedings STOC'97*, pages 168–175. ACM Press, 1997.

[24] V. Y. Pan. New techniques for the computation of linear recurrence coefficients. *Finite Fields and their Applications*, 6(1):93–118, 2000.

[25] F. Rouillier. Solving zero-dimensional systems through the Rational Univariate Representation. *Applicable Algebra in Engineering, Communication and Computing*, 9(5):433–461, 1999.

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

[27] A. Schönhage. The fundamental theorem of algebra in terms of computational complexity. Technical report, Univ. Tübingen, 1982. 73 pages.

[28] A. Schönhage. Fast parallel computation of characteristic polynomials by Leverrier's power sum method adapted to fields of finite characteristic. In *Automata, languages and programming (Lund, 1993)*, volume 700 of *LNCS*, pages 410–417. Springer, Berlin, 1993.

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

[30] V. Shoup. NTL: A library for doing number theory. `http://www.shoup.net`, 1996–2003.

[31] J.-A. Thiong Ly. Note for computing the minimum polynomial of elements in large finite fields. In *Coding theory and applications (Toulon, 1988)*, volume 388 of *LNCS*, pages 185–192. Springer, New York, 1989.

[32] A. Valibouze. Fonctions symétriques et changements de bases. In *Proc. EUROCAL–87*, volume 378 of *LNCS*, pages 323–332, 1989.

[33] M. van Hoeij. Factoring polynomials and the knapsack problem. *Journal of Number Theory*, 95(2):167–189, 2002.