J. Symbolic Computation (1990) **9**, 175–183

Inversion in finite fields using logarithmic depth

JOACHIM VON ZUR GATHEN

Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A4, Canada

(*Received* 22 *June* 1988)

Litow & Davida (1988) show that inverses in large finite fields of small characteristic p, say $p = 2$, can be computed by Boolean circuits of (order-optimal) logarithmic depth. We note that their numerical approach can also be implemented purely algebraically, and that the resulting much simpler algorithm yields, also for large p, both arithmetic and Boolean reductions of inversion in F_{p^n} to inversion in F_p.

The theory of parallel computation tries to classify problems according to their parallel complexity. A fundamental tool are the complexity classes $\mathcal{NC}^1 \subseteq \mathcal{NC}^2 \subseteq \cdots$, where $\mathcal{NC}^k$ consists of those Boolean problems that can be solved by (uniform) Boolean circuits of depth $O((\log N)^k)$ and size $N^{O(1)}$, for input size N. Computing a result depending on N inputs requires depth at least $\lceil \log N \rceil$, and so in this setting one cannot go below $\mathcal{NC}^1$.

There is an analogous "arithmetic" theory for algebraic problems (say, over a field F) that can be solved by the arithmetic operations $+, -, \times$, and $\div$, with corresponding complexity classes $\mathcal{NC}^k_F$.

In this paper, we deal with problems (viz, exponentiation and inversion) over finite fields which are meaningful both in the Boolean and in the arithmetic theory. The ultimate goal is to put our problems into the lowest possible of the above complexity classes, namely $\mathcal{NC}^1$ and $\mathcal{NC}^1_F$. We do not achieve this goal in its most natural environment, but only do so after relaxing some technical constraints from their standard setting to a more favourable setting (log-space uniform to $\mathcal{P}$-uniform, general characteristic to small characteristic, algorithms to reductions). The open questions at the end of the paper focus on the necessity of these relaxations.

An excellent overview of the Boolean theory—including complexity classes, uniformity, and reductions—is given in Cook (1985); for the arithmetic theory, we refer to von zur Gathen (1986). For the necessary algebra, we will give reference to various textbooks.

We have to deal with a potentially confusing array of problems, models, time bounds, uniformity conditions, and constraints on the field size. We classify these into four groups, with two choices in each group. The expert reader may safely ignore our notation like (A.1.I.ii) in Theorem 1.

This work was partly supported by National Science and Engineering Council of Canada, grant 3-650-126-40, and Fundación Andes, beca C-10246. Parts of it were done during a visit to Pontificia Universidad Católica, Santiago, Chile, and as a Visiting Fellow at the Computer Science Laboratory, Australian National University, Canberra, Australia.

0747–7171/90/020175+09 \$03.00/0

Let p be a prime, $m \in \mathbf{N}$, $q = p^m$, $F = \mathbf{F}_q$ a finite field, $f \in F[x]$ monic and irreducible of degree n, and $K = F[x]/(f) = \mathbf{F}_{q^n}$. Then $\alpha = x + (f) \in K$ is the usual generator of K over F, and $(1, \alpha, \alpha^2, \ldots, \alpha^{n-1})$ the standard basis for K over F. We consider two computational problems in K, namely *exponentiation* and *inversion*:

(A) EXP(K): Input: $u_0, \ldots, u_{n-1} \in F$, and $e \in \mathbf{N}$.
Output: $v_0, \ldots, v_{n-1} \in F$ such that
$(\sum_{0 \le i < n} u_i \alpha^i)^e = \sum_{0 \le i < n} v_i \alpha^i$ in K.

(B) INV(K): Input: $u_0, \ldots, u_{n-1} \in F$, not all zero.
Output: $v_0, \ldots, v_{n-1} \in F$ such that
$(\sum_{0 \le i < n} u_i \alpha^i)(\sum_{0 \le i < n} v_i \alpha^i) = 1$ in K.

We consider two models of computation:

(a) arithmetic circuits over F, using $+$, $-$, $\times$, $\div$, inputs $u_0, \ldots, u_{n-1}$ as above, and constants from F,

(b) Boolean circuits.

For Boolean computations, an element $a \in \mathbf{F}_p$ is given by the binary representation $(a_0, \ldots, a_{l-1}) \in \{0,1\}^l$ of $\bar{a} = \sum_{0 \le i < l} a_i 2^i \in \mathbf{N}$, with $l = \lceil \log p \rceil$, $0 \le \bar{a} < p$, and $a = (\bar{a} \bmod p)$. An element u_i of $F = \mathbf{F}_{p^m}$ is presented by a vector of m such elements, and an element of K by a vector from F^n. Thus the usual input size is $N = nm\lceil \log p \rceil$, or roughly $n \log q$. Strictly speaking, only n inputs are given for the arithmetic problem (a), and Theorem 2 below will indeed give depth $O(\log n)$, independent of q. We obtain circuits with the following bounds:

(I) depth $O(\log N)$ and size $N^{O(1)}$, under $\mathcal{P}$-uniformity,

(II) depth $O(\log^2 N)$ and size $N^{O(1)}$, under log-space uniformity.

Uniformity refers to a preprocessing Turing machine M which, on input N in unary, constructs a circuit that solves the problem at hand for all input sizes $n\lceil \log q \rceil \le N$. Thus the input also has to provide a description of F and K in the Boolean case; in the arithmetic case, F is fixed and we need a description of K (or f). If M uses $O(\log N)$ worktape, we have *log-space uniformity*; if M uses time $N^{O(1)}$, we have *$\mathcal{P}$-uniformity*. Log-space uniformity implies $\mathcal{P}$-uniformity.

We also need to distinguish between possibly exponential and only polynomial ratios of q to n:

(i) q is arbitrary,

(ii) $q \le n$ ("q is *small*").

In (ii), it is actually sufficient to say $q = n^{O(1)}$, or else consider the input size to be nq.

Any arithmetic circuit over K—with input $u \in K$ rather than the coordinates $u_0, \ldots, u_{n-1} \in F^n$ of u—computing EXP(K) has linear depth $\Omega(n \log q)$ for appropriate e (von zur Gathen 1987), and INV(K) can trivially be solved on an arithmetic circuit over K of size 1. Thus arithmetic circuits over K are not interesting for our topic.

It was a pleasant surprise when Fich & Tompa (1988) showed that EXP(K) can be solved by arithmetic circuits over F of depth $O(\log n \cdot \log(nq))$ and size $(n \log q)^{O(1)}$, and

that the problem is in log-space uniform Boolean NC^2 if q is small. (In our notation, all problems (A.*.II.ii) are solved.) They obtain the same bounds for INV(K), since $u^{-1} = u^{q^n-2}$ for $u \in K \setminus \{0\}$. It is not clear how to strictly improve their results. However, by relaxing the log-space uniformity—in which the NC^2 results hold—to $\mathcal{P}$-uniformity, we now obtain circuits for INV(K) using (order-optimal) logarithmic depth.

THEOREM 1. *1. (A.a.I.ii) There exist $\mathcal{P}$-uniform arithmetic circuits over F of depth $O(\log(nq))$ and size $(n \log q)^{O(1)}$ for* EXP(K).

2. (A.b.I.ii) There exist $\mathcal{P}$-uniform Boolean circuits of depth $O(\log(nq))$ and size $(n \log q)^{O(1)}$ for EXP(K).

PROOF. Following Fich & Tompa, we write the exponent as $e = \sum_{0 \le j < n} e_j q^j$ with $0 \le e_j < q$ for $0 \le j < n$, and

$$u^e = \left(\sum_{0 \le i < n} u_i \alpha^i \right)^{\sum_{0 \le j < n} e_j q^j} = \prod_{0 \le j < n} \left(\sum_{0 \le i < n} u_i \alpha^{i q^j} \right)^{e_j}.$$

Fich & Tompa then proceed by calculating the matrix of the F-linear map $u \mapsto u^q$ on K, and its powers. However, we now use Eberly's (1989) results that the "iterated product" of k polynomials in $F[x]$ of degree at most k, or the quotient and remainder of two such polynomials, can be computed with $\mathcal{P}$-uniform Boolean circuits of depth $O(\log(k \log q))$. We precompute all α^{iq^j} for $0 \le i, j < n$ in the standard basis, and then form the above iterated product, by first calculating the corresponding iterated product of less than nq polynomials in $F[x]$, each of degree less than n, and then taking it modulo f. This solves EXP(K) in Boolean depth $O(\log(nq))$, and proves 2.; 1. follows from Eberly's corresponding results on arithmetic circuits. □

Since inversion is a special case of powering, the statements of Theorem 1 also hold for INV; this is a perfectly satisfactory solution for the case (ii) of small q. (In our notation, all problems (*.*.*.ii) are solved.) When q is large, however, no Boolean circuits of poly-logarithmic depth for inversion even for the special case $p = q$ and $F = \mathbf{F}_p = \mathbf{Z}/(p)$ are known. We circumvent this problem by allowing the "redundant notation" of $u \in F$ by (a, b), where $a, b \in F$, $b \neq 0$, and $u = a/b$. Thus if $F = \mathbf{F}_p = \mathbf{Z}/(p)$, each input $u_i = u_i/1$ is given by the binary representations of u_i and 1, with $0 \le u_i < p$, and each output is represented in binary as $v_i = a_i/b_i$, with $0 \le a_i, b_i < p$. The conversion from redundant notation to standard notation (say, the binary representation of $c_i \in \mathbf{N}$ with $0 \le c_i < p$ and $c_i \equiv a_i/b_i \bmod p$) is essentially the problem INV(F). This is trivial for arithmetic circuits over F. However, strictly speaking, the Boolean algorithm presented below which uses redundant notation does not solve INV(K), but rather provides a Boolean NC^1-reduction from INV(K) to INV(F). (In deviation from the log-space uniformity required for reductions in Cook (1985), this reduction is only $\mathcal{P}$-uniform.) We obtain a $\mathcal{P}$-uniform $\mathcal{NC}^1$-result only when q is small (Theorem 1.2).

Even before the result of Fich & Tompa (1988), it was known that a subresultant approach can reduce INV(K) to linear algebra over F and put it into arithmetic NC^2_F (Borodin *et al.* 1982) and Boolean NC^2 (Borodin *et al.* 1983, using redundant notation). Thus we have results (B.*.II.i). We now improve this to logarithmic depth, again trading log-space uniformity for $\mathcal{P}$-uniformity.

THEOREM 2. *1. (B.a.I.i) There are $\mathcal{P}$-uniform arithmetic circuits over F of depth $O(\log n)$ for $\mathrm{INV}(K)$; i.e., $\mathrm{INV}(K) \in \mathcal{NC}^1_F(\mathcal{P}\text{-uniform})$.*

2. (B.b.I.i) There are $\mathcal{P}$-uniform Boolean circuits of depth $O(\log(n \log q))$ for $\mathrm{INV}(K)$, in the redundant notation.

Litow & Davida (1988) prove 2. with a numerical approach. (They do not mention the problem with inversion in F, possibly because they are mainly interested in small q; however, Theorem 1 shows that that case is covered by the methods of Fich & Tompa and Eberly.) In their words, it still requires a "rather tedious proof" to fill in the details of the error analysis necessary to actually obtain a completely specified algorithm. We translate their method into its natural algebraic setting, and describe a simple algorithm for the inversion problem.

A high level description of this method is to view K as an n-dimensional algebra $\mathcal{A}$ of $n \times n$-matrices over F, via the regular representation (see e.g., Herstein 1968, ch. 1). The Cayley-Hamilton theorem says that $\chi(A) = 0$, if χ is the characteristic polynomial of the matrix A (see e.g., Gantmacher 1960, IV.4). This will allow us to invert matrices using small powers. $\mathcal{A}$ turns out to be diagonalizable, so that these powers are easy to compute. This idea of simultaneous diagonalization of matrices to compute inverses has been used in different contexts by Bini (1984) and Bini & Pan (1986), and is implicit in the work of Reif (1986) and Eberly (1989).

We write $f = f_0 + f_1 x + \cdots + f_{n-1}x^{n-1} + x^n$, and let

$$C = \begin{pmatrix} 0 & \cdots & 0 & -f_0 \\ 1 & \ddots & \vdots & -f_1 \\ & \ddots & 0 & \vdots \\ 0 & & 1 & -f_{n-1} \end{pmatrix} \in F^{n \times n}$$

be the *companion matrix* of f (see Gantmacher 1960, VI.6). Then f is the characteristic polynomial of C, and the Cayley-Hamilton theorem says that $f(C) = 0$. Thus the F-algebra $\mathcal{A} \subseteq F^{n \times n}$ generated by C equals the F-linear span of $1, C, C^2, \ldots, C^{n-1}$. One checks that the first column of C^i is the transpose of the ith unit vector. Hence $1, C, C^2, \ldots, C^{n-1}$ are linearly independent, and the map

$$\begin{aligned} M : K &\to \mathcal{A} \\ u = \textstyle\sum u_i \alpha^i &\mapsto M_u = \textstyle\sum u_i C^i \end{aligned}$$

is an F-algebra isomorphism; i.e., it is bijective, and $M_{u+v} = M_u + M_v$, $M_{uv} = M_u M_v$ for $u, v \in K$, and M_1 is the identity matrix. (The reader may recognize it as the *regular representation* of K, where M_u is the matrix of the F-linear map "multiplication by u" in the standard basis.) The first column of M_u is $(u_0, \ldots, u_{n-1})^t$. Let $\beta_j = \alpha^{q^j} \in K$, so that $\beta_0, \ldots, \beta_{n-1}$ are the roots of f, and $V = VDM(\beta_0, \ldots, \beta_{n-1}) \in K^{n \times n}$ be their Vandermonde matrix, with $V_{ij} = \beta_i^{j-1}$. f is irreducible, and thus has n distinct roots in K (Lidl & Niederreiter 1983, Theorem 2.14). For $0 \le j < n$, we consider the automorphism $\sigma_j : K \longrightarrow K$ over F with $\sigma_j(\beta_0) = \beta_j$. Thus $\sigma_j(\sum_{0 \le i < n} u_i \alpha^i) = \sum_{0 \le i < n} u_i \sigma_j(\alpha^i) = \sum_{0 \le i < n} u_i \beta_j^i$ for an arbitrary element $\sum_{0 \le i < n} u_i \alpha^i$ of K (see Lidl & Niederreiter 1983, Theorem 2.21).

LEMMA 3. *Let* $u = \sum_{0\le i<n} u_i\alpha^i \in K$ *be nonzero,* $c(u) = c_0 + c_1 t + \cdots + t^n \in F[t]$ *the characteristic polynomial of* M_u, *and* $D_u = \mathrm{diag}(\sigma_0(u), \ldots, \sigma_{n-1}(u)) \in K^{n\times n}$ *a diagonal matrix. Then*

1. $M_u = V^{-1} D_u V$,
2. $c(u) = \prod_{0\le j<n}(t - \sigma_j(u))$,
3. $c_0 = (-1)^n \prod_{0\le j<n} \sigma_j(u) \neq 0$,
4. $M_u^{-1} = \frac{-1}{c_0}(c_1 + c_2 M_u + \cdots + M_u^{n-1})$.

PROOF. 1. We start with $u = \alpha$, so that $M_u = C$. For $0 \le j, k < n$ we have

$$(VC)_{jk} = \sum_{0\le i<n} \beta_j^i C_{ik} = \begin{cases} \beta_j^{k+1} & \text{if } k \le n-2, \\ -\sum_{0\le i<n} \beta_j^i f_i & \text{if } k = n-1, \end{cases}$$
$$(D_\alpha V)_{jk} = \beta_j \cdot \beta_j^k = \beta_j^{k+1},$$

which implies 1. in the case $M_u = C$. Since V diagonalizes C, for any $u = \sum u_i\alpha^i \in K$ it also diagonalizes M_u, which is a polynomial in C. The diagonal entries of VM_uV^{-1} are:

$$(VM_uV^{-1})_{jj} = (V(\sum_{0\le i<n} u_i C^i)V^{-1})_{jj} = \sum_{0\le i<n} u_i (VC^iV^{-1})_{jj}$$
$$= \sum_{0\le i<n} u_i (D_\alpha^i)_{jj} = \sum_{0\le i<n} u_i \sigma_j(\alpha^i) = \sigma_j(u).$$

2. Using 1., we know that $c(u)$ is the characteristic polynomial of M_u and D_u, which proves the claim.

3. The expression for c_0 follows from 2. Since $\sigma_j(0) = 0$ and σ_j is bijective, we have $\sigma_j(u) \neq 0$, for all j.

4. follows immediately from the Cayley-Hamilton theorem. □

The following algorithm consists of two stages: a precomputation step 0, which takes a description of F and f as inputs, and produces the arithmetic circuit over F described in steps 1 through 5, with input $(u_0, \ldots, u_{n-1}) \in F^n$.

ALGORITHM.

0. On input F, f as above, compute each β_j^i for $0 \le i, j < n$, then V and V^{-1}, and produce the following arithmetic circuit over F.
1. On input $u = \sum_{0\le i<n} u_i\alpha^i \in K$, compute $\sigma_j(u) = \sum u_i\beta_j^i$ for $0 \le j < n$.
2. Compute $c(u) = \prod_{0\le j<n}(t - \sigma_j(u)) = \sum_{0\le i\le n} c_i t^i \in F[t]$, with $c_i \in F$ for $0 \le i \le n$.
3. Compute $D_u, D_u^2, D_u^3, \ldots, D_u^{n-1}$.
4. Compute $M_u^{-1} = \frac{-1}{c_0} V^{-1}(\sum_{1\le i\le n} c_i D_u^{i-1})V$.
5. Return the first column of M_u^{-1}.

The precomputation step 0, not depending on u, can be done in polynomial time. By Eberly (1989), steps 2 and 3 can be performed in depth $O(\log n)$ on arithmetic circuits over F, and in depth $O(\log(n \log q))$ on Boolean circuits. The same bounds hold for step 4. This proves Theorem 2.

When q is small, then INV(F) is in $\mathcal{NC}^1(\mathcal{P}\text{-uniform})$ for small q (Beame *et al.* 1986, Eberly 1989), and this method yields an alternative proof of Theorem 1.2 for INV instead of EXP.

Remarks

1. The most interesting case for our problems is when $F = \mathsf{F}_q$ is a prime field, with $p = q$. However, even in the general case $q = p^m$, one might ask for efficient arithmetic circuits over F_p. The immediate simulation only gives depth $O(\log(n) \cdot \log(m))$ over F_p in Theorem 1, and one has to take a fresh look at the problems, with a view to arithmetic circuits over F_p. Thus for Theorem 1 one replaces the q-ary by the p-ary representation, and for Theorem 2 one considers K as an extension of F_p of degree mn. The resulting algorithms over F_p have depth $O(\log(mnp))$ and size $(mn \log p)^{O(1)}$ in Theorem 1, and depth $O(\log(mn))$ and size $(mn)^{O(1)}$ in Theorem 2.

2. Eberly's results are in fact $\mathcal{NC}^1$-reductions to the "iterated product of integers", which can be solved by log-space uniform Boolean circuits of depth $O(\log N \log\log N)$ and size $N^{O(1)}$ (Beame *et al.* 1986). Thus if the roots $\beta_0, \ldots, \beta_{n-1}$ are given, we also obtain log-space uniform arithmetic and Boolean circuits with these bounds. It is, however, not clear how to find the roots of f log-space uniformly, or how to precompute the α^{iq^j} required for Theorem 1.

3. The precomputation of α^{iq^j} in the algorithm of Theorem 1 is not necessary if K is given by a *normal basis* $(\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{n-1}})$ over F. Such a basis always exists (Lidl & Niederreiter 1983, Theorem 2.35). Setting $\alpha_i = \alpha^{q^i}$, we have $\alpha_i \alpha_j = \alpha_{i+j}$, with index arithmetic modulo n. Thus we obtain log-space uniform depth $O(\log N \log\log N)$ with size $N^{O(1)}$ if we only allow inputs (F, f, u) for which $\alpha = x + (f)$ generates a normal basis.

4. The Boolean complexity class $\mathcal{AC}^k$ is the set of problems solvable by Boolean circuits of depth $O(\log^k N)$ and size $N^{O(1)}$ (for input size N) with unbounded fan-in gates $\vee$ and $\wedge$ (see Cook 1985). We can define a corresponding arithmetic class $\mathcal{AC}^k_F$, by allowing unbounded fan-in for the $+$-gates (it is not necessary to use unbounded fan-in $*$-gates). Then clearly $\mathcal{NC}^k_F \subseteq \mathcal{AC}^k_F \subseteq \mathcal{NC}^{k+1}_F$. Eberly's results actually show that iterated product and division with remainder of polynomials is in $\mathcal{AC}^0_F$ (at least for large p), and our algorithm provides a $\mathcal{P}$-uniform $\mathcal{AC}^0_F$-reduction from INV(K) to INV(F).

5. We briefly discuss the case where f is reducible. First consider the case where q is small, so that depth $O(\log q)$ is acceptable. The powering algorithm à la Fich & Tompa (Theorem 1) in $K = F[x]/(f)$ goes through unchanged. We can compute the complete factorization $f = f_1^{e_1} \cdots f_r^{e_r}$ of f by Berlekamp's algorithm in deterministic polynomial time, where $f_1, \ldots, f_r \in F[x]$ are pairwise distinct monic irreducible

polynomials and $e_1, \ldots, e_r$ positive integers (see Knuth 1981). In fact, this computation can be done by arithmetic circuits over F of depth $O(\log^2 n \log(np))$ (von zur Gathen & Seroussi 1986). Set $n_i = \deg f_i$. The order of the group of units in $K = F[x]/(f)$ is $k = \prod_{1 \le i \le r} (q^{n_i} - 1) q^{n_i(e_i - 1)}$. Thus $u^{-1} = u^{k-1}$ for a unit $u \in K$, and again $\mathrm{INV}(K)$ is in arithmetic $\mathcal{P}$-uniform $\mathcal{NC}^1_F$, and also in $\mathcal{P}$-uniform Boolean $\mathcal{NC}^1$, if q is small. (In fact, $k = q^d \cdot \mathrm{lcm}(q^{n_1} - 1, \ldots, q^{n_r} - 1)$ is sufficient, where $d = \max_{1 \le i \le r} n_i(e_i - 1)$.)

For the case of large q, we use a probabilistic version of Berlekamp's algorithm (see e.g., Knuth (1981), and von zur Gathen (1984) for a parallel version), working in the complexity class $\mathcal{ZPP}$, defined by probabilistic polynomial-time computations, where the random algorithm either returns the correct answer or "failure"; the latter with controllably small probability. For $i \le r$, set $R_i = F[x]/(f_i^{e_i})$, and consider the isomorphism $\phi : K \longrightarrow R_1 \times \cdots \times R_r$ of the Chinese Remainder Theorem. Since the entries of the matrix of the F-linear map ϕ^{-1} can be precomputed, it is sufficient to consider the case $r = 1$. Given a unit $u = \sum u_i \alpha^i \in K$, we let $\bar{u} = \sum u_i x^i \in F[x]$, and using the algorithm for Theorem 2, we may assume that we have computed $v, w \in F[x]$ with $\bar{u}v \equiv 1 \bmod f_1$ and $\deg v < n_1$. Then

$$\bar{u} \cdot \left(\sum_{1 \le j \le e_1} \binom{e_1}{j} (-\bar{u})^{j-1} v^j \right) - 1 = -(1 - \bar{u}v)^{e_1} \equiv 0 \bmod f_1^{e_1},$$

and the inverse of u, given by the parenthesized expression, can be computed in depth $O(\log n)$. The upshot is that the problem of inverting units in K is in $\mathcal{ZPP}$-uniform $\mathcal{NC}^1_F$, and in $\mathcal{ZPP}$-uniform Boolean $\mathcal{NC}^1$, using redundant notation. (Since f is factored, it is also easy to test whether u is a unit in K, by computing $\gcd(f, \bar{u})$.)

6. The above arithmetic reduction goes through for arbitrary F (say $F = \mathbb{Q}$), provided K is separable and normal over F, i.e., generated over F by the roots of f. (The roots are not of a form α^{q^j}, of course.) An example is a cyclotomic field $K = \mathbb{Q}(\alpha)$, where $\alpha = \exp(2\pi i/k) \in \mathbb{C}$ is a primitive kth root of unity, and $n = \varphi(k)$. Step 0 would first factor the kth cyclotomic polynomial into linear factors over K to find the roots $\beta_0 = \alpha, \ldots, \beta_{n-1} \in K$; this can be done in polynomial time (Chistov & Grigoryev 1982, Landau 1985, Lenstra 1983). Without further changes, the algorithm gives $\mathcal{P}$-uniform arithmetic circuits over $\mathbb{Q}$ of depth $O(\log n)$ computing inversion in $\mathbb{Q}(\alpha)$, and also $\mathcal{P}$-uniform Boolean circuits of logarithmic depth, in redundant notation (here, the binary length of the input coefficients $u_i \in \mathbb{Q}$ has to be taken into account). For general irreducible polynomials f, however, the splitting field of f may have degree $n!$ over F, so that exact computations in K are infeasible.

7. For $F = \mathbb{Q}$ and general irreducible f, the numerical algorithm of Litow & Davida (1988) could be used to yield real approximations to the rational entries of v (as in $\mathrm{EXP}(K)$ or $\mathrm{INV}(K)$). It is, unfortunately, not clear how to recover the integral numerators and denominators fast in parallel; the sequential algorithm is via an Extended Euclidean algorithm.

8. Open question: (A.*.I.i) Given $F = \mathbb{F}_p, f, K = \mathbb{F}_{p^n}$ as above, with a large prime $p = q$, is the problem of computing large powers in K $\mathcal{NC}$-reducible to the same problem in F? Note that any arithmetic circuit over F computing general large powers in F has linear depth $\Omega(\log p)$ (von zur Gathen 1987).

9. The number-theoretical analogues of our problems are exponentiation and inversion of integers modulo m^n. Both problems are in $\mathcal{P}$-uniform $\mathcal{NC}^1$ if m has only small prime factors (Beame *et al.* (1986) for small m, von zur Gathen (1987) in general), but no fast parallel solution is known if m is a large prime.

10. OPEN QUESTION: (B.b.II.i) Given a (large) prime p, is the problem of computing $a^{-1} \bmod p$ for $1 \leq a < p$ in $\mathcal{NC}$ (allowing precomputation depending on p)? Or are there interesting classes of primes for which this is the case?

Acknowledgment

It is a pleasure to acknowledge the many discussions with Allan Borodin about the subject. I also thank an anonymous referee for several valuable suggestions.

References

Beame, P. W., Cook, S. A., Hoover, H. J. (1986). Log depth circuits for division and related problems. SIAM J. Comput. **15**, 994-1003.

Bini, D. (1984). Parallel solution of certain Toeplitz linear systems. SIAM J. Comput. **13**, 268-276.

Bini, D., Pan, V. (1986). Polynomial division and its computational complexity. J. Complexity **2**, 179-203.

Borodin, A., Cook, S., Pippenger, N. (1983). Parallel computation for well-endowed rings and space-bounded probabilistic machines. Information and Control **58**, 113-136.

Borodin, A., von zur Gathen, J., Hopcroft, J. (1982). Fast parallel matrix and GCD computations. Information and Control **52**, 241-256.

Chistov, A. L., Grigoryev, D. Yu. (1982). Polynomial-time factoring of the multivariable polynomials over a global field. LOMI preprint E-5-82, Leningrad, 1982.

Cook, S. A. (1985). A taxonomy of problems with fast parallel algorithms. Information and Control **64**, 2-22.

Eberly, W. (1989). Very fast parallel polynomial arithmetic. SIAM J. Comput., in press.

Fich, F. E., Tompa, M. (1988). The Parallel Complexity of Exponentiating Polynomials over Finite Fields. J. Assoc. Comput. Mach. **35**, 651-667.

Gantmacher, F. R. (1960). *The Theory of Matrices*, vol. 1, Chelsea: New York, x+374 pp.

von zur Gathen, J. (1984). Parallel algorithms for algebraic problems. SIAM J. Comput. **13**, 802-824.

von zur Gathen, J. (1986). Parallel arithmetic computations: a survey. Proc. 12th Int. Symp. Math. Foundations of Computer Science, Bratislava, Lecture Notes in Computer Science **233**, 93-112, Springer: Berlin.

von zur Gathen, J. (1987). Computing powers in parallel. SIAM J. Comput. **16**, 930-945.

von zur Gathen, J., Seroussi, G. (1986). Boolean circuits versus arithmetic circuits. Proc. 6th Int. Conf. Computer Science, Santiago, Chile, 171-184.

Herstein, I. N. (1968). *Noncommutative rings.* Carus Math. Monographs, vol. **15**, Wiley.

Knuth, D. E. (1981). *The Art of Computer Programming*, Vol. 2, Seminumerical algorithms, 2nd Ed. Addison-Wesley: Reading MA.

Landau, S. (1985). Factoring polynomials over algebraic number fields. SIAM J. Comput. **14**, 184-195.

Lenstra, A. K. (1983). Factoring polynomials over algebraic number fields. Proc. Conf. Math. Foundations of Computer Science, Lecture Notes in Computer Science **176**, 389-396, Springer: Berlin.

Lidl, R. Niederreiter, H. (1983). *Finite Fields.* Encyclopedia Math. Appl., Vol. 20, Addison-Wesley, Reading MA, (now distributed by Cambridge Univ. Press).

Litow, B. E., Davida, G. I. (1988). $O(\log(n))$ Parallel Time Finite Field Inversion. Proc. Aegean Workshop on Computing, Lecture Notes in Computer Science **319**, 74-80, Springer: Berlin.

Reif, J. (1986). Logarithmic depth circuits for algebraic functions. SIAM J. Comput. **15**, 231-242.