# Deterministic distinct-degree factorization of polynomials over finite fields ⋆

Shuhong Gao [a,1] Erich Kaltofen [b,2] Alan G.B. Lauder [c,3]

[a] *Department of Mathematical Sciences*
*Clemson University*
*Clemson, SC 29634-0975 USA*

[b] *Department of Mathematics*
*North Carolina State University*
*Raleigh, NC 27695-8205, USA*

[c] *Mathematical Institute*
*Oxford University*
*Oxford OX1 3LB, UK*

**Abstract**

A deterministic polynomial time algorithm is presented for finding the distinct-degree factorization of multivariate polynomials over finite fields. As a consequence, one can count the number of irreducible factors of polynomials over finite fields in deterministic polynomial time, thus resolving a theoretical open problem of Kaltofen from 1987.

*Key words:* multivariate polynomial, deterministic algorithm, distinct-degree factorization

⋆ This research was done while the authors were members at the Mathematical Sciences Research Institute in Berkeley, CA, USA.

*Email addresses:* sgao@math.clemson.edu (Shuhong Gao), kaltofen@math.ncsu.edu (Erich Kaltofen), lauder@maths.ox.ac.uk (Alan G.B. Lauder).

[1] S. Gao was supported in part by NSF under Grants DMS-9970637 and DMS-0302549, NSA under Grant MDA904-00-1-0048 and ONR under Grant N00014-00-1-0565.

[2] E. Kaltofen was supported in part by NSF under Grants DMS-9977392, CCR-9988177, CCR-0113121, and CCR-0305314.

[3] A. Lauder is a Royal Society University Research Fellow.

*Appears in J. Symbolic Comput., vol. 38, nr. 6, pp. 1461–70 (2004).*

## 1 Introduction

It is a classical result in algorithmic number theory that one may compute the distinct-degree factorization of a univariate polynomial over a finite field in deterministic polynomial time. For a polynomial $f$ of degree $n$ this is a factorization of the form $f = \prod_{d=1}^{n} f^{[d]}$ where $f^{[d]}$ is the product of all irreducible factors of $f$ of degree $d$. That this may be done follows easily from the theory of finite field extensions; von zur Gathen and Gerhard (1999, notes to section 14) trace the standard algorithm back to C. F. Gauss. For polynomials in more than one variable no such analogous theory is available. Moreover, a straightforward approach based upon computing the distinct-degree factorization of some univariate image of a multivariate polynomial fails on Swinnerton-Dyer-like polynomials. We take a different approach and present a deterministic polynomial time algorithm for distinct-degree factorization of multivariate polynomials over finite fields, which does not use any univariate factorization subroutine. The algorithm is based upon earlier work of Kaltofen, who was able to show that one may test irreducibility of multivariate polynomials in deterministic polynomial time (Kaltofen 1987). In the same paper, Kaltofen asks whether it is also possible to count the number of irreducible factors; that this may be done is a consequence of our more general result. Our main goal is to show that the distinct-degree factorization of multivariate polynomials can be computed in deterministic polynomial time, so no attempt is made to optimize the steps involved and the detailed time complexity analysis is omitted as well. Our algorithm constitutes a theoretical de-randomization result. Its running time both in terms of asymptotical complexity and in terms of practicability is at this time not at all competitive with the existing randomized approaches (Bernardin 1999; Gao 2003; Gao and Lauder 2002; Noro and Yokoyama 2002; Bostan et al. 2004).

Our paper is organized in the following way. Section 2 contains preliminary results on linear systems over polynomial algebras. In Section 3 we describe an algorithm for distinct-degree factorization, based upon the method of Kaltofen, which uses a randomized univariate factoring algorithm as a subroutine. Next in Section 4 we modify this algorithm using the methods of Section 2 so as to remove the need for any univariate factorization, just computing gcds instead. As in (Kaltofen 1987) for simplicity we shall focus on the bivariate case; the method extends to all multivariate polynomials, and we shall briefly discuss this at the end of the paper.

## 2 Linear systems over polynomial algebras

Let $w \in \mathbb{F}_q[z]$ be a squarefree polynomial of degree $n$ with irreducible factors $w_j \in \mathbb{F}_q[z]$, $1 \le j \le r$. In this section we consider homogeneous linear systems over the ring

$$R_w := \mathbb{F}_q[z]/(w(z)) \cong \bigoplus_{j=1}^{r} \mathbb{F}_q[z]/(w_j(z)).$$

The Berlekamp subalgebra of $R_w$ is

$$B_w := \{u \in R_w \,|\, u^q = u\}.$$

We have

$$B_w \cong \bigoplus_{j=1}^{r} B_{w_j}$$

and each $B_{w_j} \cong \mathbb{F}_q$. We shall denote by $\pi_{w_j}$ the projection of $R_w$ onto the $j$th summand $\mathbb{F}_q[z]/(w_j(z))$, and also the restriction of this map to $B_w$ (which has codomain $B_{w_j} \cong \mathbb{F}_q$). Thus for $u \in R_w$,

$$\pi_{w_j}(u) := u \bmod w_j.$$

Let $\mathcal{L}$ be a linear system given by

$$Lv = 0 \tag{1}$$

where $L$ is a matrix with entries in $R_w$ and $v$ is a vector of unknowns. This is a linear system over $R_w$, however we wish to find solutions $v$ to this system which have entries in $B_w$. Since $B_w$ and $R_w$ are vector spaces over $\mathbb{F}_q$, the set of vectors $v$ with entries in $B_w$ which are solutions to $\mathcal{L}$ forms an $\mathbb{F}_q$-vector space.

**Lemma 1** *One may compute a vector space basis over $\mathbb{F}_q$ of solutions to (1) with entries in $B_w$ in deterministic polynomial time.*

*Proof.* We have an explicit basis for $R_w$ over $\mathbb{F}_q$, namely $1, z, \ldots, z^{n-1} \bmod w$ where $n$ is the degree of $w$. All the entries of $L$ are represented in this basis. Since $q$th power is a linear map of $R_w$ over $\mathbb{F}_q$, a basis for $B_w$ over $\mathbb{F}_q$ can be computed in deterministic polynomial time ((Butler 1954; Lidl and Niederreiter 1983); the idea goes back to Karel Petr in 1937 as cited in (Schwarz 1956)). Suppose a basis for $B_w$ is found to be $b_1, \ldots, b_r \in R_w$ where $r$ is the number of irreducible factors of $w$. Let $\ell$ denote the number of columns in the matrix $L$. Then any solution $v = (v_1, \ldots, v_\ell)^{tr} \in B_w^\ell$ to (1) must be of the form

$$v_i = \sum_{j=1}^{r} v_{ij} b_j, \quad 1 \le i \le \ell,$$

where $v_{ij} \in \mathbb{F}_q$ are unknowns. Now we plug $v_i$ into the equation (1) and expand all the expressions in the basis $1, z, \ldots, z^{n-1}$ modulo $w$. This yields a system of linear equations in $v_{ij}$ with coefficients in $\mathbb{F}_q$ (each row of $L$ gives $n$ linear equations). Solving this system, say by Gauss elimination, gives all the solutions of (1) in $B_w$. Hence a basis for the solution space of (1) can be found in deterministic polynomial time. $\boxtimes$

We now suppose that $t(z)$ is an irreducible factor of $w$. Given $\mathcal{L}$ we denote by $\mathcal{L}_t$ the system of linear equations over $R_t$ obtained by projecting each matrix entry in $L$ under $\pi_t$. We shall call this the *projected system* of $\mathcal{L}$ under $\pi_t$. Once again we wish to find solutions of $\mathcal{L}_t$ which are vectors over $B_t \cong \mathbb{F}_q$. Certainly any solution $v$ of $\mathcal{L}$ with entries in $B_w$ will be sent under the map on vectors induced by $\pi_t$ to a solution of $\mathcal{L}_t$ with entries in $B_t$. Moreover, this solution will be non-zero if and only if $t(z)$ does not divide all of the entries in $v$ thought of as polynomials in $\mathbb{F}_q[z]$. Conversely, any solution to $\mathcal{L}_t$ with entries in $B_t$ can be lifted using the Chinese remainder theorem to a solution for $\mathcal{L}$ with entries in $B_w$. Precisely, we take the unique lifting of each entry in the solution from $B_t$ to $B_w$ which reduces to zero for all other projections $\pi_{w_j}\,(w_j \neq t)$.

Our deterministic factoring method is based upon the following proposition.

**Proposition 2** *Let $\mathcal{L}$ be any linear system over $R_w$. Let $S \subseteq \{1, 2, \ldots, r\}$ with the following properties: The dimension over $\mathbb{F}_q$ of the solution space in $B_{w_j}$ of the projected system $\mathcal{L}_{w_j}$ is non-zero if and only if $j \in S$. Then we can compute in deterministic polynomial time the factorization*

$$w = \left(\prod_{j \in S} w_j\right)\left(\prod_{k \notin S} w_k\right).$$

*Proof.* Compute a basis over $\mathbb{F}_q$ for the space of solutions in $B_w$ of the linear system $\mathcal{L}$. We claim the greatest common divisor $h$, say, of $w$ and the polynomials which occur as entries in the basis vectors is exactly $\prod_{k \notin S} w_k$. To see this, suppose $j \in S$. Then there exists some non-zero solution $\bar{v}$ of the linear system $\mathcal{L}_{w_j}$ which can be lifted to a non-zero solution $v$ of the linear system $\mathcal{L}$, as previously described. This solution of $\mathcal{L}$ must lie in the span of the basis vectors, and thus if $w_j$ divided all the entries in the basis vectors we would have that $w_j$ divides all the entries in $v$, but then $\bar{v} = 0$ — a contradiction. Hence $w_j$ does not divide the greatest common divisor $h$. Now suppose that $k \notin S$, and also that $w_k$ does not divide $h$. Then $w_k$ does not divide all the entries in the basis vectors of the solution space of $\mathcal{L}$. Thus there exists at least one basis element which projects down to a non-zero solution of $\mathcal{L}_{w_k}$ under $\pi_{w_k}$ — a contradiction. Thus $h$ is as claimed.

Now one may compute the factor $h$ in deterministic polynomial time using only

the deterministic algorithm for computing bases for $\mathcal{L}$ from Lemma 1, and the Euclidean algorithm for greatest common divisors of univariate polynomials. This completes the proof. $\boxtimes$

## 3 Randomized factorization

We present a variation of Kaltofen's algorithm (Kaltofen 1982; von zur Gathen and Kaltofen 1985; Kaltofen 1985b). For simplicity, we only give the version for bivariate polynomials, but his algorithm works for polynomials with any number of variables.

Following tradition, we shall give the definition of a *nice* polynomial. We shall say that $f \in \mathbb{F}_q[x, y]$ of total degree $n$ is *nice* if $f(x, 0)$ is squarefree and of degree $n$. We observe that the coefficient of $x^i$ of a nice polynomial $f$ as a polynomial in $y$ has degree no more than $n - i$, in particular that the leading coefficient of $f$ with respect to $x$ is in $\mathbb{F}_q$, and that factors of nice polynomials must be also nice.

ALGORITHM 3.1 [Randomized Distinct-Degree Factorization]
Input: A nice polynomial $f(x, y) \in \mathbb{F}_q[x, y]$ of total degree $n$. A positive integer $m$ such that $f$ has no factors of degree strictly less than $m$.
Output: Nice polynomials $g, h \in \mathbb{F}_q[x, y]$ and integer $s$ which satisfy the following conditions: $g$ is a product of $s$ irreducible polynomials of degree $m$, and $f = gh$ where $h$ has no factors of degree strictly less than $m + 1$. (Here $s$ may equal 0.)

Step 0: Set $s \leftarrow 0, g \leftarrow 1, h \leftarrow f$. Factor $f(z, 0)$ using a randomized algorithm and for each irreducible factor $t(z)$ of $f(z, 0)$ do Steps 1 and 2.

Step 1: [Approximate a root of $f(x, y)$ in $R_t[[y]]$ where $R_t = \mathbb{F}_q[z]/(t(z))$.] Letting $k = (2n - 1)n$ and $a_0 := z \in R_t$, by Newton iteration compute $a_1, \ldots, a_k \in R_t$ such that

$$f(a_0 + a_1 y + \ldots + a_k y^k, y) \equiv 0 \bmod y^{k+1}.$$

For $0 \le i \le m$ compute

$$\alpha^{(i)} := (a_0 + \ldots + a_k y^k)^i \bmod y^{k+1} \in R_t[y].$$

Step 2: [Try to find a polynomial of degree $\leq m$ in $\mathbb{F}_q[x, y]$ for which $\alpha^{(1)}$ is the approximation of one of its roots.]
Compute a basis over $\mathbb{F}_q$ for solutions over $B_t(\cong \mathbb{F}_q)$ of the linear system $\mathcal{L}_t$ over $R_t$ given by

$$\sum_{i=0}^{m} u_i(y)\alpha^{(i)} \equiv 0 \bmod y^{k+1}, \tag{2}$$

where $u_i(y) \in \mathbb{F}_q[y]$, $\deg_y(u_i) \leq (m-i)$, and the coefficients of $u_i(y)$ are the unknowns. If there exists a non-zero solution $\{u_i\}$ then it is unique up to scaling by $\mathbb{F}_q$. In which case define

$$u = \sum_{i=0}^{m} u_i(y)x^i,$$

which is necessarily an irreducible factor of $f$ of degree $m$ whose reduction modulo $y$ is divisible by $t(x)$. Now check whether $u$ divides $h$, for we may have already found this factor. If so set $g \leftarrow gu, h \leftarrow h/u$ and $s \leftarrow s+1$.

Step 3: Output $g, h, s$.

The justification of the correctness of this algorithm follows from Theorem 1 in (Kaltofen 1985b), where it is shown that a nice polynomial of degree $m$ that has $\sum a_i y^i$ as a root of order $O(y^{k+1})$ is in fact an irreducible factor of $f$. Comment must be made however on certain minor aspects which distinguish this algorithm from the version in (Kaltofen 1985b). First, we take as an input assumption that $f$ does not have any factors of degree less than $m$, whereas in (Kaltofen 1985b) $m$ is increased from 1 until the first factor is found. Second, we allow solutions to the linear system (2) in which $u_m(y)$ may be zero. In (Kaltofen 1985b) the author fixes $u_m(y)$ to be the unity in $\mathbb{F}_q$. We remove this restriction simply to make (2) a homogeneous system so the theory developed in Section 2 directly applies. Indeed any solution will necessarily have $u_m(y) \neq 0$, since factors of nice polynomials are nice. Note that only those irreducible factors $t(z)$ of degree $\leq m$ can possibly yield non-zero solutions to $\mathcal{L}_t$. Also each such factor can yield at most one non-zero solution, up to scaling, as $t(z)$ can occur as a factor of the reduction modulo $y$ of at most one irreducible factor of $f$. This justifies the claim on the uniqueness of $u$.

It is perhaps helpful to also explain precisely how Step 2 of the algorithm relates to Section 2. The linear system $\mathcal{L}_t$ may be made into a more explicit linear system of the form "$Lv = 0$" by equating coefficients of each power of $y$. In this case the matrix $L$ would be of size $(k+1) \times ((m+1)(m+2)/2)$ with entries from $R_t$. Note also that the rather unwieldy sentence "Compute a basis ..." could be replaced by "Solve the following linear system over $\mathbb{F}_q$ ...". However, we choose the more cumbersome version to preserve the analogy with

Section 2, and in preparation for Section 4.

## 4 Deterministic distinct-degree factorization

As in (Kaltofen 1987), the problem is that the univariate factor $t(z)$ is not known to be computable in deterministic polynomial time. Following Kaltofen, our approach is to work in

$$R_w = \mathbb{F}_q[z]/(w(z)), \qquad \text{where } w(z) = f(z,0),$$

and construct an analogous linear system to (2), only with solutions as vectors over the Berlekamp algebra $B_w$ of $R_w$. We begin with nice polynomials.

Algorithm 4.1 [Deterministic Distinct-Degree Factorization]
Input: A nice polynomial $f(x,y) \in \mathbb{F}_q[x,y]$ of total degree $n$. A positive integer $m$ such that $f$ has no factors of degree strictly less than $m$.
Output: Nice polynomials $g, h \in \mathbb{F}_q[x,y]$ and an integer $s$ which satisfy the following conditions: $g$ is a product of $s$ irreducible polynomials of degree $m$, and $f = gh$ where $h$ has no factors of degree strictly less than $m+1$. (Note that $s$ may equal 0.)

Step 1: [Approximate a root of $f(x,y)$]
Define $k := (2n-1)n$, $w(z) := f(z,0)$, $R_w := \mathbb{F}_q[z]/(w(z))$, and $a_0 := z \in R_w$. By Newton iteration compute $a_1, a_2, \ldots, a_k \in R_w$ such that

$$f(a_0 + a_1 y + \ldots + a_k y^k, y) \equiv 0 \bmod y^{k+1}.$$

For $0 \le i \le m$ compute

$$\alpha^{(i)} := (a_0 + \ldots + a_k y^k)^i \bmod y^{k+1}.$$

Since the coefficient ring $R_w$ is not a field, care must be taken so that the iteration does not divide by a zero-divisor. In (Kaltofen 1985a, Algorithm 2, Steps I and N) it is shown that standard Newton iteration works as a consequence of the squarefreeness of $w(z) = f(z,0)$.

Step 2: [Try to find a polynomial of degree $\le m$ in $B_w[x,y]$ for which $\alpha^{(1)}$ is the approximation of one of its roots, where $B_w$ is the Berlekamp subalgebra of $R_w$.]
Compute a basis over $\mathbb{F}_q$ of solutions over $B_w$ to the homogeneous linear system

$\mathcal{L}$ over $R_w$ given by,

$$\sum_{i=0}^{m} u_i(y)\alpha^{(i)} \equiv 0 \bmod y^{k+1}, \tag{3}$$

where the coefficients of $u_i(y) \in B_w[y]$, $\deg_y(u_i) \leq (m-i)$, are the unknowns. If the dimension is zero then output "$s = 0$, $g = 1$, and $h = f$" and halt.

Step 3: Compute the gcd of $w(z)$ and the entries of all basis elements of the solution space of $\mathcal{L}$, thought of as polynomials in $\mathbb{F}_q[z]$. This gives a factor of $w(z) = f(z, 0)$, denoted by $h_0(z)$. Let $g_0(z)$ be the cofactor of $h_0(z)$ in $w(z)$.

Step 4: Switching $z$ to $x$, we have the factorization $f(x, 0) = w(x) = g_0(x)h_0(x)$. Using Hensel lifting compute a factorization $f = gh$ with $g \equiv g_0 \bmod y$ and $h \equiv h_0 \bmod y$. Output $g$ and $h$, and also $s := \deg_x(g_0(x))/m$.

**Proposition 3** *Algorithm 4.1 outputs correctly and runs in deterministic polynomial time.*

*Proof.* For each irreducible factor $t(z)$ of $w(z) = f(z, 0)$, the linear system (2) is the projection $\mathcal{L}_t$ of the linear system $\mathcal{L}$ defined by (3) under $\pi_t$. (It was briefly explained how to present these linear systems in the form "$Lv = 0$" in the second paragraph following Algorithm 3.1 and we will not labor this point.) Now $\mathcal{L}_t$ has a non-zero solution if and only if $f$ has a factor of degree $\leq m$ whose reduction modulo $y$ is divisible by $t(z)$. Such a factor must have degree exactly $m$ and be irreducible by the input assumption on $f$.

Thus we are in the situation of Proposition 2 with $w(z) = f(z, 0)$. Let $f_1, \ldots, f_r$ be all the irreducible factors of $f(z, 0)$, and let $f^{[m]}(x, y)$ be the product of all irreducible factors of $f$ of degree $m$. By Proposition 2, we can compute in deterministic polynomial time the factorization

$$f(z, 0) = \Bigg(\underbrace{\prod_{j \in S} f_j}_{g_0}\Bigg)\Bigg(\underbrace{\prod_{k \notin S} f_k}_{h_0}\Bigg)$$

where $S$ is the set of all indices $j$ such that the polynomial $f_j(z)$ divides $f^{[m]}(z, 0)$. This means that $g_0(x) = f^{[m]}(x, 0)$. Hence using Hensel lifting we may recover in deterministic polynomial time this factor $g = f^{[m]}$ and its cofactor $h$, say, in $f$. Finally, we can compute the number of irreducible factors of $f$ of degree exactly $m$ as $\deg_x(\prod_{j \in S} f_j(x))/m$. This completes the proof. ⊠

Note that the above algorithm may be used to remove equal-degree irreducible factors of a squarefree univariate polynomial in a somewhat different manner from the usual method (von zur Gathen and Gerhard 1999, Section 14.2). It

is illuminating to describe the main features of the algorithm in this special case: Given such a univariate polynomial $f \in \mathbb{F}_q[x] \subseteq \mathbb{F}_q[x,y]$ we have that $w := f \bmod y = f$. In Step 1 of the algorithm the approximate root $\alpha$ of $f$ in $R_f[[y]]$ is just the exact root $z \bmod f$. In Step 2, in the linear system $\mathcal{L}$ we may ignore higher powers of $y$, and the problem reduces to finding a sequence of elements $u_0, u_1, \ldots, u_m \in R_f$ such that $\sum_{i=0}^{m} u_i z^i = 0$ in $R_f$. Now suppose that $f_j$ is an irreducible factor of $f$ of (total) degree $m$, and write $f_j = \sum_{i=0}^{m} v_i x^i$ where $v_i \in \mathbb{F}_q$. For $0 \le i \le m$, define $u_i$ as $(0, 0, \ldots, v_i, \ldots, 0) \in B_f \cong \oplus_{i=1}^{r} \mathbb{F}_q$, where the non-zero entry is in the $j$th position. Then the sequence $u_i$ gives a solution to our linear system. For this reason essentially, in Step 3 one recovers the product of all irreducible factors of degree $m$ of the univariate polynomial $f$. Note that the above approach does not seem to lead to an asymptotically faster algorithm than the current best (von zur Gathen and Shoup 1992; Kaltofen and Shoup 1998).

Algorithm 4.1 may be iterated in a straightforward manner to compute the complete distinct-degree factorization of a nice bivariate polynomial. That is, one starts with an arbitrary nice $f$ taking $m = 1$ and by repeated application of the algorithm with $m$ incremented by one each time successively remove factors of increasing degree. So we have

**Proposition 4** *There is an algorithm for computing the distinct-degree factorization of any nice polynomial of total degree $n$ in $\mathbb{F}_q[x,y]$ which runs in deterministic polynomial time in $n$ and $\log(q)$.*

Now we show how to reduce general polynomials to nice ones. Let $f \in \mathbb{F}_q[x,y]$ of total degree $n$. By the algorithm of Yun (1976), with well-known modifications when both partial derivatives vanish, one can compute its squarefree decomposition in deterministic polynomial time. Hence we may assume that $f$ is already squarefree in $\mathbb{F}_q[x,y]$.

For small $q$, say $q < 2n^2$, since Berlekamp's algorithm (Berlekamp 1967; Lidl and Niederreiter 1983) for univariate polynomials runs in deterministic polynomial time, Kaltofen's original version of Algorithm 3.1 in the previous section can factor $f$ in deterministic polynomial time. So in this case one can certainly find the distinct-degree factorization of $f$.

Assume that $q \ge 2n^2$. Consider the following substitution:

$$\tilde{f} = f(x, y + ax + b)$$

for some $a, b \in \mathbb{F}_q$. Certainly, any factorization of $f$ gives a factorization of $\tilde{f}$ and vice versa. Note that the coefficient of $x^n$ in $\tilde{f}$ is a polynomial $h$ in $a$ of degree at most $n$, and

$$\tilde{f}(x, 0) = f(x, ax + b).$$

To make $\tilde{f}$ nice, we just need to pick $a \in \mathbb{F}_q$ such that $h(a) \neq 0$ and then find $b \in \mathbb{F}_q$ such that

$$\text{Resultant}_x(\tilde{f}(x,0), \frac{\partial}{\partial x}\tilde{f}(x,0)) \neq 0,$$

which is a nonzero polynomial in $b$ of degree at most $n(2n-1) < 2n^2$, since $f$ is squarefree in $\mathbb{F}_q[x,y]$. Hence both $a$ and $b$ can be found after trying at most $2n^2$ elements in $\mathbb{F}_q$. So $a$ and $b$ can be found in deterministic polynomial time. Combining with Proposition 4, we have the following result.

**Theorem 5** *There is an algorithm for computing the distinct-degree factorization of any polynomial of total degree $n$ in $\mathbb{F}_q[x,y]$ in deterministic polynomial time in $n$ and $\log(q)$.*

When the distinct-degree factorization of $f$ is computed, it is simple to find the number of irreducible factors of $f$.

**Corollary 6** *There is an algorithm for counting the number of irreducible factors of any polynomial of total degree $n$ in $\mathbb{F}_q[x,y]$ in deterministic polynomial time in $n$ and $\log(q)$.*

This resolves an open problem posed in (Kaltofen 1987).

Note that having obtained a distinct total degree factorization, one may attempt to find finer factorizations by considering different degree orderings. In the algorithm we restrict to the standard degree ordering obtained by giving both variables equal "weight" and defining the (total) degree of a polynomial to be the greatest weight of any monomial. This is an inessential restriction, and our algorithm works with different degree orderings, such as degree in $x$ or degree in $y$, or other degree orderings in which the two variables are assigned different weights. The easiest way to obtain a nice input while accounting for those degree orders is to work with a new main variable $z$ and factor the tri-variate polynomial $f(z+x, az+b+y)$. We give a brief explanation for a weighted degree $d = w_x \deg_x(f) + w_y \deg_y(f)$ with respect to integral weights $w_x > 0$ and $w_y > 0$. We can assume that all irreducible factors of $f$ have total degree $m$. We may also suppose that $f$, or equivalently $f(z+x, az+b+y)$, has no irreducible factors of weighted degree $< d$. We then find the factor of $f(z, az+b)$ in $\mathbb{F}_q[z]$ that lifts to the product of the irreducible factors of $f(z+x, az+b+y)$ of weighted degree $d$ by solving the linear system corresponding to (3), now for the coefficients of $u_i(x,y)$, with the restriction that the weighted degree of the constant term $u_0(x,y)$ be $d$.

We should remark that the method applies equally well to polynomials with more than two variables. For a polynomial with $v$ variables and (total) degree $n$, we assume the input size is $\mathcal{O}(\binom{n+v}{v} \log q)$ that is, we use dense multivariate representation. The Newton approximation in Step 1 of Algorithm 4.1 and the shift step (to make the polynomial nice) can be carried out similarly

as in (Kaltofen 1985b) and gcds of multivariate polynomials can be computed in deterministic polynomial time (Brown and Traub 1971). Hence for any polynomial with $v$ variables and of total degree $n$ over $\mathbb{F}_q$ one can count the number of its irreducible factors in deterministic polynomial time in the input size.

**Acknowledgements:** We thank the three anonymous referees and Barry Trager for their comments, which have substantially improved the presentation of our ideas.

## References

Berlekamp, E. R., 1967. Factoring polynomials over finite fields. Bell Systems Tech. J. 46, 1853–1859, republished in revised form in: E. R. Berlekamp, *Algebraic Coding Theory*, Chapter 6, McGraw-Hill Publ., New York, 1968.

Bernardin, L., 1999. Factoring multivariate polynomials over a finite field. Dissertation, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, available at `http://bernardin.com/publications.html`.

Bostan, A., Lecerf, G., Salvy, B., Schost, É., Wiebelt, B., 2004. Complexity issues in bivariate polynomial factorization. In: Gutierrez, J. (Ed.), ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput. ACM Press, New York, N. Y., pp. 42–49.

Brown, W. S., Traub, J. F., 1971. On Euclid's algorithm and the theory of subresultants. J. ACM 18, 505–514.

Butler, M. C. R., 1954. On the reducibility of polynomials over a finite field. Quart. J. Math., Oxford Ser. (2) 5, 102–107.

Gao, S., 2003. Factoring multivariate polynomials via partial differential equations. Math. Comput. 72 (242), 801–822.

Gao, S., Lauder, A. G. B., 2002. Hensel lifting and polynomial factorisation. Math. Comput. 71, 1663–1676.

von zur Gathen, J., Gerhard, J., 1999. Modern Computer Algebra. Cambridge University Press, Cambridge, New York, Melbourne.

von zur Gathen, J., Kaltofen, E., 1985. Factoring multivariate polynomials over finite fields. Math. Comput. 45, 251–261.

von zur Gathen, J., Shoup, V., 1992. Computing Frobenius maps and factoring polynomials. Comput. Complexity 2, 187–224.

Kaltofen, E., 1982. A polynomial-time reduction from bivariate to univariate integral polynomial factorization. In: Proc. 23rd Annual Symp. Foundations of Comp. Sci. IEEE, pp. 57–64, journal version in Kaltofen (1985b).

Kaltofen, E., 1985a. Fast parallel absolute irreducibility testing. J. Symbolic Comput. 1 (1), 57–67, misprint corrections: *J. Symbolic Comput.* vol. 9, p. 320 (1989).

Kaltofen, E., 1985b. Polynomial-time reductions from multivariate to bi- and

univariate integral polynomial factorization. SIAM J. Comput. 14 (2), 469–489.

Kaltofen, E., 1987. Deterministic irreducibility testing of polynomials over large finite fields. J. Symbolic Comput. 4, 77–82.

Kaltofen, E., Shoup, V., Jul. 1998. Subquadratic-time factoring of polynomials over finite fields. Math. Comput. 67 (223), 1179–1197.

Lidl, R., Niederreiter, H., 1983. Finite Fields. Addison-Wesley, Reading, Massachusetts, USA, now distributed by Cambridge University Press.

Noro, M., Yokoyama, K., 2002. Yet another practical implementation of polynomial factorization over finite fields. In: Mora, T. (Ed.), Proc. 2002 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'02). ACM Press, New York, N. Y., pp. 200–206.

Schwarz, Št., 1956. On the reducibility of polynomials over a finite field. Quart. J. Math. Oxford Ser. (2) 7, 110–124.

Yun, D. Y. Y., 1976. On square-free decomposition algorithms. In: Jenks, R. D. (Ed.), Proc. 1976 ACM Symp. Symbolic Algebraic Comput. ACM, pp. 26–35, SYMSAC was held at IBM Research in Yorktown Heights, New York.