Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Symbolic Computation 38 (2004) 1227–1246
Journal of Symbolic Computation
www.elsevier.com/locate/jsc

Implementation of prime decomposition of polynomial ideals over small finite fields

Masayuki Noro[a], Kazuhiro Yokoyama[b,*]

[a] *Department of Mathematics, Kobe University, 1-1 Rokkodai, Nada-ku, Kobe 657-8501, Japan*
[b] *Faculty of Mathematics, Kyushu University, 6-10-1 Higashi-ku, Fukuoka 812-8581, Japan*

Received 26 January 2003; accepted 14 August 2003

Available online 6 May 2004

Abstract

An algorithm for the prime decomposition of polynomial ideals over small finite fields is proposed and implemented on the basis of previous work of the second author. To achieve better performance, several improvements are added to the existing algorithm, with strategies for computational flow proposed, based on experimental results. The practicality of the algorithm is examined by testing the implementation experimentally, which also reveals information about the quality of the implementation.

1. Introduction

The theory of primary decomposition of ideals in noetherian rings is very classical, with many works having studied the computation over fields of characteristic 0 (see Decker et al. (1999) for a more detailed history and a summary of more recent work). For fields of positive characteristic, existing works (Kalkberner, 1994; Gianni and Trager, 1996; Matsumoto, 2001; Kemper, 2002; Fortuna et al., 2002) on the subject and related topics are largely general and theoretical. However, to develop a practical algorithm for the primary decomposition of polynomial ideals over finite fields is not only very interesting as a computational problem, but also very useful for studies on pure mathematics and engineering problems. Thus, our goal is to develop a practical algorithm for the primary decomposition of a polynomial ideal over a finite field. To do this, we apply

* Corresponding author.
E-mail address: yokoyama@math.kyushu-u.ac.jp (K. Yokoyama).

0747-7171/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsc.2003.08.004

the "localization technique" of Shimoyama and Yokoyama (1996), where primary components are extracted from prime divisors. This technique is based on Gröbner basis computation and does not depend on the characteristic of the coefficient field. Therefore primary decomposition computations can be efficiently reduced to prime decomposition computations.

We propose a precise algorithm for the prime decomposition of polynomial ideals over small finite fields based on Yokoyama (2002), and report on the results of our implementation on a real computer. To achieve better performance, several improvements are added to the work of Yokoyama (2002), with strategies regarding computational flow proposed, based on experimental results. The practicality of the algorithm is examined through experiments on examples, which also demonstrate the quality of the implementation.

As noted by Yokoyama (2002), there are differences between prime decompositions of cases of characteristic 0 and those of positive characteristic, and we cannot apply methods used for polynomial ideals over the rational number field directly to those over finite fields. A procedure that handles the differences is therefore developed. To achieve the most efficient computation, the algorithm and its implementation are given the following features:

(1) We employ the well-established strategy of Gianni et al. (1988), but modify the "decomposition using generic position" that is very successful in cases of characteristic 0. (We note that the original method of Gianni et al. (1988) may work in larger characteristic cases, where the problem in (2) hardly occurs.)
(2) To solve problems arising from positive characteristic, we introduce the notion of "separable ideals" and "separable closure of ideals". Using separable closure, factorization of polynomials over finite fields results in true prime decomposition (see Section 2.2 for details).
(3) Using factorization of the minimal polynomial of each variable, a partial decomposition of the given ideal called "intermediate decomposition" is computed (we call each computed ideal an *intermediate ideal*). In many cases, intermediate ideals tend to be prime, and so this decomposition improves computational efficiency.
(4) Radicals only need to be computed for some primary ideals, and in these cases, we apply the efficient method of Matsumoto (2001) based on "inverse Frobenius map computation". Note that the entire computation can be done without determining the radical of the given ideal I. To improve overall efficiency, it may be necessary to compute the intersection of computed prime ideals (see Section 3.6). In these cases, the radical ideal of a given ideal is computed as a by-product. In our experiment, there are cases where this computed radicals much faster than existing methods (Matsumoto, 2001; Kemper, 2002).
(5) As we want to compute the prime divisors of a given ideal I, we can make use of "partial decompositions" (Caboara et al., 1997; Shimoyama and Yokoyama, 1996). This is done by first applying the "pre-decomposition" suggested by Shimoyama and Yokoyama (1996), and involves factoring all elements in a Gröbner basis of I. For each computed ideal, we then apply an algorithm based on Yokoyama (2002).

Of course, as the whole procedure consists of basic arithmetical operations, Gröbner basis computation and polynomial factorization over finite fields, overall efficiency is dependent on the efficiencies of the individual computations. We thus apply the following methods:

(6) Zech representation is employed to allow efficient arithmetic over extension fields $GF(q)$, as it is well suited to polynomial factorization over $GF(q)$ (see Noro and Yokoyama, 2002).
(7) For polynomial factorization over $GF(q)$, the most recent algorithms of Bernardin and Monagan (1997) and Noro and Yokoyama (2002) are used.
(8) An "FGLM-type" method Faugère et al. (1993) is used in Gröbner computations to allow efficient computation of the minimal polynomial of each variable.

All procedures were implemented in Risa/Asir, with computational tests of a number of examples conducted. Experimentally, the algorithm was found to work very well with a set of examples. Although testing using only a limited number of examples cannot fully verify the efficiency of the implementation, it does give an indication of the quality of the algorithm and its implementation. Note that Pfister has also implemented the algorithm of Yokoyama (2002) in SINGULAR, and this is used for comparison in our testing.

For fields of larger positive characteristic, the method of Wu (1984) and that of Eisenbud et al. (1992) with integral closure computation algorithm of de Jong (1998) may also compute the prime/primary decomposition of polynomial ideals. But, these methods suffer the same problem in (2) arising from *inseparability* in small characteristic cases (see Section 2.2 for details), because the method of Wu (1984) requires factorization of polynomials over algebraic extension fields of rational function fields, which implies that it requires *elements in generic position*; also the method of Eisenbud et al. (1992) requires such elements as it needs zero-dimensional prime decomposition. As this problem is resolved by using "separable closures" proposed by Yokoyama (2002), these methods can be improved to handle such cases by utilizing our implementation.

The current problem is strongly related to *radical ideal computation in positive characteristic* (Matsumoto, 2001; Fortuna et al., 2002; Kemper, 2002), with the difference being analogous to that between the irreducible factorization of polynomials and square-free factorization using only derivations. Since the algorithm uses minimal polynomials of variables, it is similar to the radical ideal computation of Kemper (2002).

2. Review of approach and key points

A summary of the key points of the computation of prime decomposition in positive characteristic of Yokoyama (2002) is given. Throughout this paper, we consider a polynomial ring $K[x_1, \ldots, x_n]$, where K is a finite field $GF(q)$ of order q and characteristic p, and we denote the set of variables $\{x_1, \ldots, x_n\}$ by X. For a noetherian commutative ring R, we write $Id_R(f_1, \ldots, f_t)$ for the ideal generated by elements $f_1, \ldots, f_t$ of R, and $(I : f)$ for the quotient ideal of an ideal I of an element f of R. For an ideal I of R, we denote the radical by $\sqrt{I}$, the set of all prime divisors of I by $\mathrm{Ass}(I)$ and the set of all isolated prime divisors of I by $\mathrm{Ass}_{\mathrm{iso}}(I)$. Then $\sqrt{I} = \cap_{P \in \mathrm{Ass}_{\mathrm{iso}}(I)} P$

and $\mathrm{Ass}(\sqrt{I}) = \mathrm{Ass}_{\mathrm{iso}}(I)$. From the prime decomposition of I, we thus mean to compute $\mathrm{Ass}_{\mathrm{iso}}(I)$. For a polynomial ideal J of $L[Z]$, where L is an extension field of K and Z is a set of variables, we denote the algebraic variety of J, i.e. the set of all zeros of J, by $V_{\tilde{L}}(J)$, where we consider zeros in the algebraic closure $\tilde{L}$ of L. Conversely, for an algebraic variety W, we denote its corresponding ideal $\{f \in L[Z] \mid f(\alpha) = 0 \text{ for any } \alpha \in W\}$ by $I_{L[Z]}(W)$.

2.1. Successive and simultaneous approaches

There are currently two approaches to prime decomposition, *successive* and *simultaneous*. Kalkberner (1994) discussed the prime decomposition of ideals of $R[x]$ by inductive arguments for a noetherian commutative ring R with identity under the assumption that one can compute the prime decomposition of ideals of R and one can compute factorization of univariate polynomials over the quotient field $Q(R/P)$ for every prime ideal P. In our case, by computing pure-dimensional components (Gianni et al., 1988; Shimoyama and Yokoyama, 1996; Caboara et al., 1997), we can reduce the problem to that of zero-dimensional ideals over rational function fields. Letting $L = K(Y)$ for some $Y \subset X$, one can compute the prime decomposition of the elimination ideal $I \cap L[Z \cup \{z\}]$ for $Z \subset X \backslash Y$ and $z \in X \backslash (Z \cup Y)$ from that of $I \cap L[Z]$ if the factorization of univariate polynomials over any algebraic extension field of L can be calculated. This is exactly equivalent to the "construction of successive extension fields over rational function fields", and we thus call it the *successive approach*. (The method of Wu (1984) can be considered as belonging to this approach in view of its procedure.)

If we employ the *successive approach*, decomposition efficiency is related to the efficient factorization of univariate polynomials over algebraic extension fields of rational function fields. However, since this factorization reduces to the factorization of polynomials over the ground field, a more practical approach to prime decomposition is the *simultaneous approach* through which prime decomposition can be performed using non-iterative methods. (The method of Eisenbud et al. (1992) can be considered as belonging to this approach type.) The *strategy* of Gianni et al. (1988) and *decomposition using generic position* are therefore used, as the most practical among the simultaneous approaches, to develop an algorithm using the following principles:

- The notion of *separable closure* is introduced to overcome certain difficulties arising in cases of positive characteristic.
- We only consider factorization of polynomials over the ground (perfect) field $K = GF(q)$ to increase the efficiency of the implementation.

2.2. Recasting the problem as decomposition of separable ideals

A general method for prime decomposition of zero-dimensional ideals using generic position is first presented, with problems that arise in the method with regard to positive characteristic noted where appropriate. The notion of separable closure and an algorithm for computing it are then presented and, using these, we overcome the problems surrounding positive characteristic.

Hereafter, let Y be a proper subset of X, $Z = X \backslash Y$ and $L = K(Y)$. For simplicity, we write $Z = \{x_1, \ldots, x_s\}$ and $Y = \{x_{s+1}, \ldots, x_n\}$. Moreover, t is always used to represent new variables.

2.2.1. Decomposition using generic position

We begin with giving the definition of minimal polynomials and polynomials in generic position (slightly different from the standard one), and then show decomposition by using (polynomials in) generic position.

Definition 2.1. Let J be a zero-dimensional ideal of $L[Z]$. For a polynomial $f(Z)$ in $L[Z]$, the *minimal polynomial* $m_f(t)$ *with respect to* J is defined as the monic, univariate polynomial over L having the smallest degree among all univariate polynomials h such that $h(f) \in J$.

For each variable x in Z, the minimal polynomial $m_x(x)$ with respect to J is the generator of the elimination ideal $J \cap L[x]$.

Definition 2.2. Let J be an ideal of $L[Z]$. A polynomial $g(Z) \in L[Z]$ is said to be *in generic position with respect to* J if $\deg(m_g(t)) = \dim_L(L[Z]/J)$ for the minimal polynomial m_g with respect to J.

Proposition 2.3 can be considered a special case of Proposition 8.69 of Becker and Weispfenning (1993).

Proposition 2.3. *Let J be an ideal of $L[Z]$, and suppose that a polynomial $g(Z)$ is in generic position with respect to J and that m_g is the minimal polynomial of $g(Z)$ with respect to J. Moreover, suppose that*

$$m_g(t) = m_1(t)^{e_1} \cdots m_r(t)^{e_r}$$

is the irreducible factorization of m_g over L. Then $P_i = Id_{L[Z]}(J, m_i(g))$ is a prime divisor for each m_i, and $\sqrt{J} = \cap_{i=1}^{r} P_i$ is the prime decomposition of J.

Proof. As $g(Z)$ is in generic position, there is a polynomial $x_i - h_i(g(Z))$ in J for each variable $x \in Z$, where h_i is a univariate polynomial over L, considering the ideal $J' = Id_{L[Z \cup \{t\}]}(J \cup \{t - g(Z)\})$. Then, J' is "in normal position" with respect to t in the sense of Becker and Weispfenning (1993). Hence, each $P_i = Id_{L[Z]}(J, m_i(g))$ is a primary ideal by Proposition 8.69 of Becker and Weispfenning (1993). If $\sqrt{P_i} \neq P_i$, then $\dim_L(L[Z]/\sqrt{P_i}) < \dim_L(L[Z]/P_i)$. This implies that the minimal polynomial m_i' of $g(Z)$ with respect to $\sqrt{P_i}$ must be a non-trivial divisor of m_i. But, as m_i is irreducible, this is a contradiction. Hence, P_i is a prime ideal. □

By Gauss's lemma, the factorization described in Proposition 2.3 can be performed in $K[X]$, where the minimal polynomial m_g with respect to J is taken to be a polynomial in a new variable t of Y over K by removing the denominator.

Thus, once we find a polynomial in generic position, one can compute its prime decomposition by factorization of its minimal polynomial. We refer to this procedure as *decomposition using generic position*. When K is of characteristic 0, each radical ideal

is *separable* (see Definition 2.4) and almost all of the linear polynomials are in generic position. However, there are certain computational problems in applying this method:

- Even if J is a radical ideal, the existence of a polynomial in generic position is not guaranteed. Moreover, even if such a polynomial does exist, there may not be polynomials of lower degrees, such as linear polynomials over K, in generic position. As the choice of a polynomial in generic position has a great impact on the total efficiency, it is desirable to find linear polynomials in generic position.
- We cannot apply Seidenberg's theorem (Seidenberg, 1974) to compute radical ideals, but instead have to rely on other existing algorithms (Matsumoto, 2001; Kemper, 2002). However, as radical ideal computation tends to be computationally difficult, unnecessary radical ideal computation is to be avoided.

To overcome this difficulty, we introduce the notion of "separable closure". From the separable closure $\mathrm{sc}(J)$, one can compute the prime decomposition of $\mathrm{sc}(J)$ by *decomposition using generic position*, from which one can extract the prime divisors of J.

2.2.2. Decomposition via separable closure

Definition 2.4. For an ideal J of $L[Z]$, J is said to be *separable* if

(1) J is a zero-dimensional radical ideal and
(2) for every prime divisor P of J, the residue class ring $L[Z]/P$ is a separable extension field of L.

Separability was also discussed by Kemper (2002) and generalized by Fortuna et al. (2002).

Definition 2.5. For a univariate polynomial $f(x)$ over L, f is said to be *separable* if f has no multiple root in the algebraic closure $\tilde{L}$ of L. Moreover, if there is a separable polynomial h such that $f(x) = h(x^{p^e})$ for some non-negative integer e, h is called the *separable closure of* f and denoted by $\mathrm{sc}(f)$.

Proposition 2.6. *Let J be a zero-dimensional ideal of $L[Z]$. If the minimal polynomial m_x with respect to J of each x in Z is separable, then J is separable.*

Proof. By the definition of separability, $\gcd(m_x, \mathrm{d}m_x/\mathrm{d}x) = 1$ for every x in Z. By Lemma 92 of Seidenberg (1974) or Lemma 8.13 of Becker and Weispfenning (1993), J is a radical ideal (see also Kemper, 2002, Proposition 4).

Next consider $\mathrm{Ass}(J)$. For each $P \in \mathrm{Ass}(J)$, $L' = L[Z]/P$ is an extension field of L and $L' \cong L(\alpha_1, \ldots, \alpha_s)$ for any $\alpha = (\alpha_1, \ldots, \alpha_s)$ in $V_{\tilde{L}}(P)$. Since each α_i is a root of the separable polynomial m_{x_i}, each α_i is a separable element over L and, thus, $L(\alpha_1, \ldots, \alpha_s)$ is a separable extension field of L. □

For a separable ideal, a polynomial in generic position corresponds to a common primitive element of separable extensions (this is used in the standard definition of "generic position"). Conversely, if there is a primitive element for each $L[Z]/P$, we can apply the Chinese remainder theorem to show that there also exists a common primitive element by the co-maximality of prime divisors.

Lemma 2.7. *Let J be a separable ideal of $L[Z]$. A polynomial $g(Z)$ in $L[Z]$ is in generic position with respect to J if and only if for each prime divisor P of J, $g(Z)$ is a primitive element of the separable extension $L[Z]/P$.*

As there exists a primitive element for each separable extension field, we can show the existence of polynomials in generic position.

Corollary 2.8. *Let J be a separable ideal of $L[Z]$. There exists a polynomial in $L[Z]$ in generic position with respect to J.*

If K has enough number of elements, a polynomial $g(Z)$ in generic position can be found among the linear polynomials $\sum_{x_i \in Z} a_i x_i$, $a_i \in K$ (see Lemma 3.2).

Example 2.9. In $GF(p)(u, v)[x, y]$, $Id(x^p - u, y^p - v)$ is prime but inseparable and thus there are no polynomials in generic position. In $GF(p)(z)[x, y]$, $Q = Id(x^p - z, y^p - z)$ is a primary ideal associated with the prime ideal $Id(x^p - z, x - y)$. The minimal polynomial of each variable with respect to Q is irreducible but inseparable.

Definition 2.10. Let J be a zero-dimensional ideal of $L[Z]$. If an ideal J' of $L[Z]$ satisfies the following conditions then we call it the *separable closure of* J and denote it by $\mathrm{sc}(J)$.

(1) J' is a separable ideal of $L[Z]$.
(2) There is a correspondence between the zeros of J (in $V_{\bar{L}}(J)$) and those of J' (in $V_{\bar{L}}(J')$) as follows: for each zero $\alpha = (\alpha_1, \ldots, \alpha_s)$ of J there exists a unique zero $\beta = (\beta_1, \ldots, \beta_s)$ of J' such that $\beta_i = \alpha_i^{p^{e_i}}$ for each i, where e_i is a non-negative integer determined by α.

The following theorem asserts the existence of the separable closure for a zero-dimensional ideal. But, as we compute separable closures only for "ideals of special types", we omit the proof (see Yokoyama, 2002).

Theorem 2.11. *For each zero-dimensional ideal J of $L[Z]$, there exists a unique* $\mathrm{sc}(J)$. *Moreover, there is a correspondence between the prime divisors of J and those of* $\mathrm{sc}(J)$. *Suppose a prime divisor P of J corresponds to a prime divisor Q of* $\mathrm{sc}(J)$. *Then there exist non-negative integers $e_1, \ldots, e_s$ such that each zero $(\alpha_1, \ldots, \alpha_s)$ of P corresponds uniquely to a zero $(\alpha_1^{p^{e_1}}, \ldots, \alpha_s^{p^{e_s}})$ of Q. We call $E = (e_1, \ldots, e_s)$ the exponent vector of P.*

The correspondence, however, is not necessarily one to one, i.e. distinct prime divisors of J may correspond to the same prime divisor of $\mathrm{sc}(J)$, but with different exponent vectors. If J is of *special type*, as defined below, every prime divisor of J has the same exponent vector, and the correspondence is one to one.

Definition 2.12. Let J be a zero-dimensional ideal of $L[Z]$. J is said to be *of special type*, if the minimal polynomial m_{x_i} of x_i with respect to J is irreducible for every x_i in Z.

Example 2.13. Consider the second example of Example 2.9.

$$J = Id(x^p - z, y^p - z) \leftrightarrow \mathrm{sc}(J) = Id(x - z, y - z)$$
$$V(J) \ni (\sqrt[p]{z}, \sqrt[p]{z}) \leftrightarrow (z, z) \in V(\mathrm{sc}(J))$$

$$P = Id(x^p - z, x - y) \leftrightarrow \mathrm{sc}(J) = Q = Id(x - z, y - z).$$

In our algorithm, we do not compute $\mathrm{sc}(J)$ directly from J, but compute ideals J_j such that $\sqrt{J} = \cap_{j=1}^{r}\sqrt{J_j}$ and every J_j is of special type. We call each of these J_j' *an intermediate ideal* of J. For each J_j, we compute $\mathrm{sc}(J_j)$ as follows.

Take an intermediate ideal J_j and write it simply as H. Then H is a zero-dimensional ideal of special type in $L[Z]$. By definition, the minimal polynomial m_{x_i} of x_i with respect to H is irreducible over L for every x_i in Z. (And by removing the denominator, m_{x_i} is also irreducible over K.) By considering the square-free decomposition, it follows that a separable closure $\mathrm{sc}(m_{x_i})$ exists for each m_{x_i} and that $m_{x_i}(t) = \mathrm{sc}(m_{x_i})(t^{q_i})$ and $q_i = p^{e_i}$ for each $x_i \in Z$. We now define the *Frobenius map*:

$$\phi_E : L[Z] \ni f(x_1, \ldots, x_s) \to f(x_1^{q_1}, \ldots, x_s^{q_s}) \in L[Z],$$

where $E = (e_1, \ldots, e_s)$. We can then compute $\mathrm{sc}(H)$ as follows.

Theorem 2.14. *For the separable closure* $\mathrm{sc}(H)$ *of* H*, we have*

$$\mathrm{sc}(H) = \phi_E^{-1}(H) = \{f \in L[Z] \mid \phi_E(f) \in H\}.$$

Moreover, there is a one to one correspondence between the prime divisors P *of* H *and the prime divisors* Q *of* $\mathrm{sc}(H)$ *such that*

$$Q = \mathrm{sc}(P) = \phi_E^{-1}(P).$$

Proof. Let $H' = \phi_E^{-1}(H)$. For each x_i in Z, $\mathrm{sc}(m_{x_i})(x_i)$ belongs to H' because $\mathrm{sc}(m_{x_i})(x_i^{q_i}) = m_{x_i}$ belongs to H. Then H' is separable because its minimal polynomial $\mathrm{sc}(m_{x_i})$ is a separable polynomial for every variable x_i in Z. In addition, there is a one to one correspondence between $V_{\tilde{L}}(H)$ and $V_{\tilde{L}}(\phi_E^{-1}(H)) = V_{\tilde{L}}(H')$ because for each zero $\alpha = (\alpha_1, \ldots, \alpha_s)$ of H, $\beta = (\alpha_1^{q_1}, \ldots, \alpha_s^{q_s})$ is a zero of H' and, conversely, for each zero $\beta = (\beta_1, \ldots, \beta_s)$ of H', $\alpha = (\sqrt[q_1]{\beta_1}, \ldots, \sqrt[q_s]{\beta_s})$ is a zero of H. Thus, by Definition 2.10, we have $H' = \mathrm{sc}(H)$ and a correspondence between $\mathrm{Ass}(H)$ and $\mathrm{Ass}(\mathrm{sc}(H))$. □

Once one has obtained all prime divisors of $\mathrm{sc}(H)$, one can recover the corresponding prime divisors of H as follows:

Proposition 2.15. *Let* Q *be a prime divisor of* $\mathrm{sc}(H)$*,* P *the corresponding prime divisor of* H *and* $P_0 = Id(\phi_E(Q))$*. Then,* $\sqrt{P_0} = P$*, that is,* P_0 *is either the corresponding prime divisor or its associated primary ideal.*

Proof. Consider each zero $\alpha = (\alpha_1, \ldots, \alpha_s)$ of P_0. As $P_0 = Id(\phi_E(Q))$, $(\alpha_1^{q_1}, \ldots, \alpha_s^{q_s})$ must be a zero of Q, and hence α is a zero of the corresponding prime divisor P of Q. Thus, $V_{\tilde{L}}(P_0) \subset V_{\tilde{L}}(P)$. But, as P is a maximal ideal, we have $V_{\tilde{L}}(P_0) = V_{\tilde{L}}(P)$, and $\sqrt{P_0} = P$ by Nullstellensatz. □

Frobenius map computation

Both *inverse Frobenius map computation* $\phi_E^{-1}(H)$ and *Frobenius map computation* $Id(\phi_E(Q))$ can be performed by elimination ideal computation (see Adams and Loustaunau, 1994, Chapter 2).

For the *inverse Frobenius map*, we introduce an elimination ordering $x_i \gg y_j$ and compute a Gröbner basis G_0 of $Id(H \cup \{x_i^{p^{e_i}} - y_i \mid 1 \leq i \leq s\})$ in $L[x_1, \ldots, x_s, y_1, \ldots, y_s]$. Then, $G_0 \cap L[y_1, \ldots, y_s]$ with y_i replaced by x_i for each i is a Gröbner basis of $\phi_E^{-1}(H)$ (see Matsumoto, 2001, Propositions 2.5 and 2.6).

For the *Frobenius map*, we introduce an elimination ordering $x_i \gg y_j$ and compute a Gröbner basis G_1 of $Id(Q \cup \{y_i^{p^{e_i}} - x_i \mid 1 \leq i \leq s\})$ in $L[x_1, \ldots, x_s, y_1, \ldots, y_s]$. Then, $G_1 \cap L[y_1, \ldots, y_s]$ with y_i replaced by x_i for each i is a Gröbner basis of $Id(\phi_E(Q))$. This can be shown by using the property that ϕ_E is a ring endomorphism.

Radical ideal computation

When $J \neq \sqrt{J}$, the ideal P obtained by Frobenius map computation may not be a prime ideal but a primary ideal (see Proposition 2.15). We thus need to compute $\sqrt{P}$. However, as our goal is to compute the prime divisors of the original ideal I in $K[X]$ and $(\sqrt{P})^c = \sqrt{P} \cap K[X]$ is the required prime divisor of I (see Lemma 3.1), we compute $(\sqrt{P})^c$ directly using

$$(\sqrt{P})^c = \sqrt{P^c} = \sqrt{P \cap K[X]},$$

where P^c denotes the *contraction* of P, that is, $P^c = P \cap K[X]$. In this case, as the ground field of K is $GF(q)$, we can compute the radical $\sqrt{P^c}$ efficiently using the method of Matsumoto (2001) which consists of inverse Frobenius map computation and p-th root computation of field elements. We can modify the method of Kemper (2002) to suit our situation, which may correspond to radical computation using the exponent vector E, while the method of Matsumoto (2001) may correspond to that without using E.

3. Computation of prime decomposition

This section describes the proposed algorithm in its entirety. Let I be an ideal of $K[X]$. As we want to compute prime divisors of I, we make use of a number of existing decomposition formulas (Shimoyama and Yokoyama, 1996; Caboara et al., 1997). For example, the following are used frequently:

(A) $\sqrt{Id(I, fg)} = \sqrt{Id(I, f)} \cap \sqrt{Id(I, g)}$,

(B) $\sqrt{I} = \sqrt{(IR_f \cap R)} \cap \sqrt{Id(I, f)}$.

3.1. Pre-decomposition

If I has no inseparable prime divisors, we can apply the same procedure as in the characteristic 0 case without using the special procedure described in Section 2. As it seems very unlikely that randomly generated ideals will have an inseparable divisor, we must consider ideals with inseparable prime divisors as *special cases*. In implementation, it is thus not efficient to apply the procedure designed for special cases to all cases of I directly; it is better to compute "partial decompositions" obtained by simply applying useful decomposition formulas to each generator of I. Our implementation employs

the following pre-procedure as proposed in Section 5.1 of Shimoyama and Yokoyama (1996).

Pre-procedure

By applying decomposition (A) to the given ideal I, we can compute ideals $I_i, i = 1, \ldots, r$, such that $\sqrt{I} = \sqrt{I_1} \cap \cdots \cap \sqrt{I_s}$ and for each i, every element of the computed Gröbner basis of I_i is irreducible in $K[X]$.

We call each I_i a *pre-component* of $\sqrt{I}$. The prime decomposition of $\sqrt{I}$ is then obtained by gathering isolated prime divisors of all pre-component I_i.

3.2. Reduction to zero-dimensional ideals

We first compute pure-dimensional components from I by techniques using *independent sets* modulo I (see Becker and Weispfenning, 1993, Chapter 8 for details). Using a Gröbner basis, we compute a maximal strongly independent set Y modulo I, and lift I to its *extension ideal* J of $K(Y)[Z]$, where $Z = X \backslash Y$. Then, for each prime divisor P of J, we extract the corresponding prime divisor $P^c = P \cap K[X]$ by *contraction* computation, giving the following prime decompositions:

$$\sqrt{J} = \cap_{i=1}^{r} P_i, \qquad \sqrt{I} = (\cap_{i=1}^{r} P_i^c) \cap \sqrt{I'},$$

where $I' = Id_{K[X]}(I, f)$ for some polynomial f computed from J such that $\sqrt{(IR_f \cap R)} = \cap_{i=1}^{r} P_i^c$. Useful properties of contractions follow (see Becker and Weispfenning, 1993).

Lemma 3.1. *Let J be an ideal of $L[Z]$ and $J^c = J \cap K[X]$ a contraction. Then:*

(i) *If J is a radical ideal, then J^c is also a radical ideal.*
(ii) *If J is a prime ideal, then J^c is also a prime ideal.*
(iii) *If J is a primary ideal, then J^c is also a primary ideal.*

As I is a proper subset of I', we can compute all prime divisors of I in finitely many steps by applying the above computations to I' recursively. Decomposition (A) is also applied to improve total efficiency. Using the factorization $f = \prod_{i=1}^{s} f_i^{e_i}$ in $K[X]$, we get

$$\sqrt{I'} = \sqrt{Id_{K[X]}(I, f)} = \cap_{i=1}^{s} \sqrt{Id_{K[X]}(I, f_i)}.$$

We then compute the prime decomposition of each $Id_{K[X]}(I, f_i)$, instead of I'.

3.3. Intermediate decomposition

We consider a zero-dimensional ideal J of $L[Z]$, where $Y \subset X, L = K(Y)$ and $Z = X \backslash Y$. For each variable $x_i \in Z$, we compute the minimal polynomial $m_{x_i}(t)$ with respect to J. This can be considered a polynomial in t and Y over K by removing its denominator. We then factorize over K to give

$$m_{x_i}(t) = \prod_j m_{i,j}(t)^{e_{i,j}},$$

where each $m_{i,j}$ is irreducible over K and thus over $K[Y]$. By Gauss's lemma, $m_{i,j}$ is also irreducible over L. Adjoining each $m_{i,j}$ to J gives the following intermediate decomposition, where each J_k is of special type:

$$\sqrt{J} = \cap_{k=1}^{r} \sqrt{J_k}.$$

Let $\mathcal{F}_i$ be the set of all distinct irreducible factors of m_{x_i} over K, and $n_i = \#\mathcal{F}_i$ for each i. Since intermediate ideals are of the form $Id_{L[Z]}(J, g_1, \ldots, g_s)$, where each g_i is chosen from $\mathcal{F}_i$, we have to deal with $n_1 \cdots n_s$ combinations of $(g_1, \ldots, g_s)$, which will require a large number of computations. Moreover, many of these computations are unnecessary, that is, they tend to coincide with $L[Z]$. And worse, computation of all the minimal polynomials at once tends to be very difficult. It is thus better to apply *incremental decomposition* where we adjoin g_i to each ideal and then compute the next minimal polynomial g_{i+1}, *successively*.

3.4. Prime decomposition of intermediate ideals

Let $\mathcal{J}$ be the intermediate decomposition of J. We present a concrete method for prime decomposition of H in $\mathcal{J}$. Let g_i denote the minimal polynomial of x_i for each x_i in Z. The ideal H can be classified as one of the following cases:

Generic Case in which some $x_i \in Z$ are in generic position with respect to H:

By Proposition 2.3, the factorization of the minimal polynomial $g_i(x_i)$ gives the prime decomposition. But as $g_i(x_i)$ is already irreducible, H is a prime ideal.

Non-Generic and Separable Case in which no x_i is in generic position, but H is a separable ideal:

As H is a separable ideal, there exists a polynomial in generic position. Thus, we search for such a polynomial h among all linear polynomials, and then compute the prime decomposition by factorization of the minimal polynomial of h. (Note: if H is zero-dimensional over K, then H is always separable.)

Non-Generic and Inseparable Case in which no x_i is in generic position and H is an inseparable ideal:

We first compute the separable closure $\mathrm{sc}(H)$. Then $\mathrm{sc}(g_i)$ is the minimal polynomial of x_i with respect to $\mathrm{sc}(H)$. This case can be further divided into the following sub-cases:

Generic Sub-Case in which some $x_i \in Z$ are in generic position with respect to $\mathrm{sc}(H)$:

In this case, $\mathrm{sc}(H)$ is a prime ideal, and so the corresponding ideal H is a prime or a primary ideal. Thus, the prime ideal is calculated by computing $\sqrt{H}$.

Non-Generic Sub-Case: This case is similar to the *Non-Generic and Separable Case*. We search for a linear polynomial h in generic position with respect to $\mathrm{sc}(H)$. We then compute the prime decomposition of $\mathrm{sc}(H)$ using the factorization of the minimal polynomial of h, and compute the corresponding prime ideals by Frobenius map computation and radical ideal computation.

3.5. Remarks on finding polynomials in generic position

By Proposition 2.3, we can find a polynomial g in generic position by checking whether the degree of the minimal polynomial m_g equals $\dim_L(L[Z]/\mathrm{sc}(H))$. To increase efficiency, we want to find a linear polynomial $g(Z)$ in generic position for the following reasons:

(1) The efficiency of computation of minimal polynomials increases for polynomials of smaller degrees.
(2) An efficient strategy for choosing candidate polynomials in generic position from all linear polynomials, by which we place a bound on the number of trials as shown in Lemma 3.2, is applied (see Yokoyama et al., 1992, for details). The bound in Lemma 3.2 is theoretical and it is likely that we will find a linear polynomial in generic position even if K does not satisfy this bound.
(3) Even if the ground field is extended, the effect on the efficiency of basic arithmetical operations is small because we are using the Zech representation.

Lemma 3.2 (Theorem 4.2 in Yokoyama et al. (1992)). *Let* $T = s \times \ell \times \dim_L(L[Z]/\mathrm{sc}(H))$, *where* $s = \#Z$ *and* $\ell = \#\mathrm{Ass}(\mathrm{sc}(H))$. *Then, if* $\#K > T$, *there exists a polynomial* g *in generic position among all linear polynomials in* Z *over* K.

If the order of the finite ground field K is too small, a problem may occur in finding a linear polynomial in generic position. To avoid this, we extend K to K' to a large enough order. After computing the prime decomposition over an extension field, we recover the prime decomposition over K as follows.

Consider the case where we must replace K with the extension field K_1. In this case, we deal with the ideal $J_1 = K_1 \otimes J$ of $K_1(Y)[Z]$ instead of the ideal J. We then apply *prime decomposition* to obtain the set $\mathcal{P}_{K_1}$ of all prime divisors of $J_1^c = J_1 \cap K_1[X]$. By the action of the Galois group $\mathcal{G} = \mathrm{Galois}(K_1/K) \subset \mathrm{Galois}(\tilde{K}/K)$, $\mathcal{P}_{K_1}$ is divided into $\mathcal{G}$-orbits, where $\mathcal{G}$ acts on $K_1[X]$ as $\mathcal{G}$ acts on the coefficients of polynomials, and thus acts on the set of ideals P_{K_1} by $\sigma(P_{K_1}) = \{\sigma(h) \mid h \in P_{K_1}\}$. Then, from Nullstellensatz and the fact that the conjugate of each zero α of a prime ideal in $\mathcal{P}_{K_1}$ is also a zero of some prime ideal in $\mathcal{P}_{K_1}$, we have the following:

Lemma 3.3. *Suppose that* $P_{K_1,1}, P_{K_1,2}, \ldots, P_{K_1,r}$ *form one* $\mathcal{G}$*-orbit. Then,* $W = V_{\tilde{K}}(P_{K_1,1}) \cup \cdots \cup V_{\tilde{K}}(P_{K_1,r})$ *forms a minimal invariant set among unions of sets in* $\mathcal{P}_{K_1}$ *for* $\mathcal{G}$. *(We note* $\tilde{K} = \tilde{K}_1$.*) Thus, there exists a unique prime divisor* P *of* $J^c = J \cap K[X]$ *such that* $V_{\tilde{K}}(P) = W$.

There are two ways to compute the prime ideal P in Lemma 3.3: one is to compute the intersection of the $P_{K_1,i}$s and the other is to use elimination techniques. For the intersection of $P_{K_1,i}$s, using the same notation as in 3.3:

Lemma 3.4. *Let* $P' = \cap_{i=1}^{r} P_{K_1,i}$ *and* G' *be the reduced Gröbner basis of* P' *such that the leading coefficient of* g *is 1 for every element* g *in* G'. *Then* $G' \subset K[X]$ *and* G' *is the Gröbner basis of* P.

Lemma 3.4 can be seen by the fact that for any $g \in G'$ and $\sigma \in \mathcal{G}$, $\sigma(g)$ also belongs to P' and must be reduced to 0 by G'.

The method using elimination is now explained. Consider P_{K_1} in $\mathcal{P}_{K_1}$. As K_1 is a finite extension field of K, it can be considered as $K[T]/P_0$, where T is a set of new variables and P_0 is a maximal ideal of $K[T]$. Consider a Gröbner basis G_{K_1} of P_{K_1} as a set of $K[T, X]$ and let $P' = Id_{K[T,X]}(P_0, G_{K_1})$. Then, P_{K_1} contains $P' \cap K[X]$ as subsets of $K_1[X]$, and the $\mathcal{G}$-conjugates of P_{K_1} also contain $P' \cap K[X]$. It can then be shown that $P' \cap K[X]$ is the prime divisor of J^c corresponding to P_{K_1}. This divisor can be computed by Gröbner basis computation using the elimination ordering $T \gg X$.

Remark 3.5. As the ideal over a larger field tends to have prime divisors with smaller linear dimension as zero-dimensional ideals over rational function fields, using an extension field K_1 may improve the total efficiency of the prime decomposition. However, there are also cases where this effect might reduce the total efficiency. For example, consider the case where the ideal J^c is a prime ideal over K but not over an extension field K_1. In this case, unnecessary Gröbner computations are performed. An efficient *primality check* is thus needed to handle such cases.

3.6. Removing redundant divisors and early termination

Throughout the procedure, a number of redundant prime ideals appear because we calculate prime ideals from a number of different ideals appearing in the computation. To access the true prime divisors, we need to eliminate all redundant prime ideals. We therefore give a procedure for such elimination, from which we derive an effective check for "early termination" of the procedure and another for avoiding unnecessary prime decomposition. If these checks are computed efficiently, they greatly improve the total efficiency (see examples in Section 5).

Let $\mathcal{P}$ be the set of all computed prime ideals, and P be a newly computed prime ideal.

Redundant ideal elimination

If P contains a prime ideal P' in $\mathcal{P}$, then P is a redundant prime ideal and we discard it. Otherwise, we add P to $\mathcal{P}$. Moreover, if P is contained in a prime ideal P' in $\mathcal{P}$, we have to remove P' from $\mathcal{P}$. The "ideal inclusion" $A \subset B$ for ideals A and B can be checked by computing the normal forms of generators A with respect to a Gröbner basis of B.

Early Termination

If P passes the above check, we compute the intersection $J = P \cap (\cap_{Q\in\mathcal{P}} Q)$, where $\cap_{Q\in\mathcal{P}} Q$ has already been computed. (Note that if there is a prime ideal P' containing P and P' is removed from $\mathcal{P}$, the intersection J will be unchanged. Thus, in this case $P \cap (\cap_{Q\in\mathcal{P}} Q) = P \cap (\cap_{Q\in\mathcal{P}\setminus\{P'\}} Q)$.) The entire procedure can then be terminated if $\sqrt{I} = J$.

Of course, if we have already computed $\sqrt{I}$, the *Early Termination* check is merely a test of the coincidence of Gröbner bases. However, as the radical computation tends to be very time-consuming, we can perform the check without computing $\sqrt{I}$ as follows:

If $\sqrt{I} \supset J$, we get the equality $\sqrt{I} = J$ because $\sqrt{I} \subset J$. To check whether $\sqrt{I} \supset J$ it suffices to check whether each generator f of J belongs to $\sqrt{I}$ by radical membership computation. Actually, the radical membership of f can be determined by

checking whether $(J : f^{\infty}) = K[X]$. Since we already computed a Gröbner basis of J with respect to some term order $>$, we can compute $(J : f^{\infty})$ efficiently as the elimination ideal $Id_{K[X\cup\{t\}]}(J, f\cdot t-1)\cap K[X]$ with respect to a block order $>'$ such that the restriction $>'_X$ of $>'$ on X coincides with $>$.

As radical ideal computation can be conducted in a similar manner to prime decomposition, it seems inefficient to do the computations independently. For an input ideal with a smaller number of prime divisors, a smaller number of checks in *Early Termination* are required, improving overall efficiency.

Moreover, Lemma 3.6 also suggests the use of the *Early Termination* technique.

Lemma 3.6. *Let $\mathcal{P}$ be the set of prime ideals that have already been computed at some point in a computation, $J = \cap_{P\in\mathcal{P}} P$ and I' be a newly computed ideal to which we apply prime decomposition. If $J \subset \sqrt{I'}$, then there is no prime divisor of I among all the prime divisors of I', and we avoid unnecessary computation for I'.*

4. Implementation details

4.1. Multivariate factorization and GCD over finite fields

To decompose an ideal, it is necessary to factorize the minimal polynomials. If the minimal polynomials are computed over fields of rational functions, then a multivariate factorizer over finite fields is required. A multivariate polynomial can be factorized by a modular algorithm composed of evaluation at a point, multivariate Hensel lifting and trial division. The current implementation is based on the algorithm described by Bernardin and Monagan (1997), who noted that cases where we cannot find feasible evaluation points are often encountered. In these cases we have to extend the ground field. In our implementation, such field extension is represented in Zech representation, that is, $GF(q)\backslash\{0\}$ is represented by $\{1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$, where α is a primitive $(q-1)$-th root of unity. Addition in $GF(q)$ is performed via a table of length $q-1$ and if q is sufficiently small, e.g. if $q < 2^{16}$, performance loss is negligible. Practically speaking, if the order of the ground field is more than 10^4, it is large enough for finding feasible evaluation points. We therefore currently use the ordinary representation for $GF(p)$ with $p > 2^{14}$, and consequently our implementation can handle any "reasonable" input ideal over $GF(p)$ when p is a prime of the machine integer size.

In cases of characteristic 0, we can apply the modular method to square-free factorization and GCD computation. But in cases of positive characteristic, evaluation points for execution of the modular method do not often exist. Therefore, in the current implementation we apply the Chinese remainder theorem to a set of GCDs computed at sufficiently many evaluation points. Field extension is used where necessary.

4.2. Incremental intermediate decomposition

Intermediate decomposition of the radical of a zero-dimensional ideal J can be performed by extracting non-trivial ideals from the set of ideals:

$\{Id_{K[X]}(J, g_1, \ldots, g_n) \mid g_i$ is an irreducible factor of the minimal polynomial of $x_i\}$.

In general, many $Id_{K[X]}(J, g_1, \ldots, g_n)$ turn out to be the trivial ideal. To avoid such unnecessary computations, we adopt an incremental method for intermediate decomposition. When we decompose an ideal I, we proceed with recursive decomposition by adjoining one irreducible factor of the minimal polynomial of a variable at each step. With each factor adjoined, the degrees of the minimal polynomials of other variables may be decreased, making subsequent computations faster. Furthermore, if x is in generic position with respect to I, then $\sqrt{I} = \cap Id_{K[X]}(I, m_i(x))$, where $\{m_i(x)\}$ is the set of all irreducible factors of the minimal polynomial of x. In this case, we do not have to consider the minimal polynomials of variables other than x.

We note that this kind of incremental decomposition is also applicable to the complete decomposition of each intermediate ideal. To this end, we attempt to find a polynomial in generic position. Usually this is done by generating a linear polynomial g and computing the minimal polynomial m_g. Even if g is not in generic position, if m_g is reducible, then we obtain a non-trivial decomposition using the procedure described above (see also Anai et al., 1996).

4.3. Computation of minimal polynomials

It is often hard to compute minimal polynomials using the Buchberger algorithm because Gröbner bases have to be computed with respect to an elimination order. To overcome this difficulty, we implemented a direct computation of minimal polynomials via an FGLM method Faugère et al. (1993) when the ground field is a finite field or a field of univariate rational functions. The former is obvious and we briefly explain the latter case.

The argument of Noro and Yokoyama (1999) can be generalized as the following lemma:

Lemma 4.1. *Let $\phi : h(u_1, \ldots, u_l) \mapsto h(a_1, \ldots, a_l)$ be an evaluation map from $K[U]_M$ to K, where $U = \{u_1, \ldots, u_l\}$, $(a_1, \ldots, a_l) \in K^l$ and $K[U]_M$ is the localization of $K[U]$ at $M = Id(x_1 - a_1, \ldots, x_l - a_l)$. Let $G \subset K[U]_M[X]$ be a reduced Gröbner basis over $K(U)$. Then $\phi(G) \subset K[X]$ is well defined and is a Gröbner basis over K. Let $\overline{m}(t)$ be the minimal polynomial of $\phi(f)$ for a given $f \in K[U][X]$. If there exists a monic $m(t) \in K[U]_M[t]$ such that $\deg_t(m(t)) = \deg_t(\overline{m}(t))$ and $\phi(m(t)) = \overline{m}(t)$, then $m(t)$ is the minimal polynomial of f over $K(U)$.*

The coefficients of $m(t)$ satisfy a system of linear equations over $K(U)$, which is derived from the membership condition with respect to the Gröbner basis G. The coefficients $f(U)$ can be computed by a modular method similar to that used for rational number coefficients. Starting from the minimal polynomial over K, we compute the solution mod M^k by Hensel lifting. We then apply polynomial–rational function transformation. That is, we try to find polynomials $g(U)$ and $h(U)$ such that $\deg(g), \deg(h) < k/2$ and $h(U)f(U) \equiv g(u) \bmod M^k$ for each component of the mod M^k solution. Only the univariate case is implemented, with efficient implementation of the general case left as a future work.

4.4. Competitive computation

To fully implement the procedure, there are a number of parameters in various parts of the procedure that need to be determined. For example, it is necessary to choose a

term ordering for Gröbner basis computation, and this is often crucial for efficiency. We have two methods of minimal polynomial computation: elimination by the Buchberger algorithm and the direct method described in the previous section. Experimentally, we have found that it is difficult to predict which is better for any given case. For this reason, competitive computation is applied as described by Maekawa et al. (2001). When the ground field is a finite field or a field of univariate rational functions, the two methods are executed simultaneously on two different servers, with the result returned first used. The remaining server is reset immediately and the subsequent minimal polynomial computation can start at once.

5. Experiments

The entire algorithm was implemented on Risa/Asir,[1] using the built-in multivariate factorizer and Buchberger algorithm driver over small finite fields. In addition to the examples from Caboara et al. (1997) and Matsumoto (2001), we prepared several examples from famous benchmark problems and those derived from generic polynomials of small Galois groups in Kemper and Mattig (2000). The ideals are all positive dimensional because we are primarily interested in cases in which inseparable ideals may appear.

Logar: $2ahi + bh^2 + 2cdj - cei - cgh - deh, ai^2 + 2bhi + 2cfj - cgi + d^2j - dei - dgh - efh, bi^2 + 2dfj - dgi - efi - fgh, f(fj - gi)$.

8_3: $C + cE - eC - E, F - C, E - G, eF + fH + hE - fE - hF - eH, fG - gF, gH + G - hG - H, cH - hC$.

H katsura (n) (homogenized katsura-n) system: $u_l u - \sum_{i=-n}^{n} u_i u_{l-i} (l = 0, \ldots, n-1), \sum_{l=-n}^{n} u_l - u$ where $u_\ell = u_{-\ell}$.

H cyclic (n) (homogenized cyclic-n) system: $\sum_{i=1}^{n} \prod_{j=i}^{k+j-1} c_{j \bmod n} (k = 1, \ldots, n-1), \prod_{j=1}^{n} c_j - c^n$.

P_{4444}: $x^8 + x^2 + t, y^8 + y^2 + t, z^8 + z^2 + t, u^8 + u^2 + t$.

P_{666}: $x^{12} + x^2 + t, y^{12} + y^2 + t, z^{12} + z^2 + t$.

P_{765}: $z^{14} + z^2 + t, y^{12} + z^2y^{10} + z^4y^8 + z^6y^6 + z^8y^4 + z^{10}y^2 + z^{12} + 1, x^{10} + (y^2 + z^2)x^8 + (y^4 + z^2y^2 + z^4)x^6 + (y^6 + z^2y^4 + z^4y^2 + z^6)x^4 + (y^8 + z^2y^6 + z^4y^4 + z^6y^2 + z^8)x^2 + y^{10} + z^2y^8 + z^4y^6 + z^6y^4 + z^8y^2 + z^{10}$.

$P_{12,12,12}$: $x^{12} + x^{10} + x^8 + x^2 + t, y^{12} + y^{10} + y^8 + y^2 + t, z^{12} + z^{10} + z^8 + z^2 + t$.

Q_{765}: $z^{21} + z^3 + t^2, y^{18} + z^3y^{15} + z^6y^{12} + z^9y^9 + z^{12}y^6 + z^{15}y^3 + z^{18} + 1, x^{15} + (y^3 + z^3)x^{12} + (y^6 + z^3y^3 + z^6)x^9 + (y^9 + z^3y^6 + z^6y^3 + z^9)x^6 + (y^{12} + z^3y^9 + z^6y^6 + z^9y^3 + z^{12})x^3 + y^{15} + z^3y^{12} + z^6y^9 + z^9y^6 + z^{12}y^3 + z^{15}$.

Q_{4321}: $z^9 + z^3 + t^2, y^9 + z^3y^6 + z^6y^3 + z^9 + 1, x^6 + (y^3 + z^3)x^3 + y^6 + z^3y^3 + z^6, y^6 + (z^3 + u^3)y^3 + z^6 + u^3z^3 + u^6$.

R_{543}: $z^{25} + z^5 + t^2, y^{20} + z^5y^{15} + z^{10}y^{10} + z^{15}y^5 + z^{20} + 1, x^{15} + (y^5 + z^5)x^{10} + (y^{10} + z^5y^5 + z^{10})x^5 + y^{15} + z^5y^{10} + z^{10}y^5 + z^{15}$.

[1] http://www.math.kobe-u.ac.jp/Asir/asir.html.

Table 1
Prime decomposition of the radical over $GF(2)$

	Dim	*Comp*	*ET*	T_C	T_D	T_B	Singular
Logar	7	8	on	12	12	12	20
Logar	–	–	off	–	>5 min	–	–
8_3	5	3	on	8.4	8.6	8.4	0.8
$Hcyclic(6)$	3	6	on	2	2	2	>5 min
$Hcyclic(6)$	3	6	off	0.8	0.8	0.8	–
P_{4444}	1	36	on	1.9	12	1.4	2
P_{666}	1	5	on	2.4	4	>5 min	20
P_{765}	1	2	on	6	6	>5 min	24
$P_{12,12,12}$	1	5	on	11	10	>5 min	8

For each problem, the prime decomposition of its radical was computed using a number of different methods. In the following tables, *Field*, *Dim* and *Comp* represent the ground field, the dimension of the ideal and the number of prime components respectively. ET indicates whether Early Termination was enabled or not. T_C, T_D and T_B indicate the elapsed computing times when the competitive computation (Strategy **C**), direct computation (Strategy **D**) and elimination using the Buchberger algorithm (Strategy **B**) were used for minimal polynomial computation. As a reference, we also show the timing data of `minAssGTZ` in Singular 2-0-4.[2] In Asir, computations are done over sufficiently large extension fields and final results are computed using the method described in Section 3.5. All measurements were performed on an SMP PC containing two Athlon MP1900+. Times are shown in seconds, with "–" indicating "not measured" or "unnecessary".

Table 1 shows the results over $GF(2)$. In Asir, computation over $GF(2^{13})$ is used internally. During the computation of *Logar*, a large number of unnecessary components are calculated and so Early Termination works very well with this kind of input. P_{666} and $P_{12,12,12}$ could not be computed using Strategy **B**, but the minimal polynomials were easily computed by Strategy **D**. However, the result for P_{4444} shows that there is a case where minimal polynomials are computed efficiently by Strategy **B**. Furthermore, we observe that $T_C < \min(T_D, T_B)$ for P_{666} and so neither Strategy **D** nor Strategy **B** is superior throughout the computation in this case. Competitive computation is thus very effective in such cases.

Table 2 shows results over various finite fields $GF(3), GF(5)$ and $GF(53)$, where $GF(3^7), GF(5^5)$ and $GF(53^2)$ were used internally. The results of $Hkatsura(6)$ and $Hkatsura(7)$ over $GF(53)$ show the performance of the minimal polynomial computation of Strategy **D**. For $Hkatsura$ and $Hcyclic$, Early Termination does not improve efficiency because they are low dimensional ideals and the number of redundant components produced during the whole procedure is thus very small. If ET is on, $T_C = 326$ for $Hcyclic(6)$ and $T_C = 47$ for $Hkatsura(6)$. The additional computational cost is due to the computation of ideal intersections and radical equality checks.

For prime decomposition of each example over extension fields, the efficiency of the basic arithmetic does not change. Therefore, the elapsed computing times are almost the

[2] http://www.singular.uni-kl.de/.

Table 2
Prime decomposition of the radical over various fields

	Field	*Dim*	*Comp*	*ET*	T_C	T_D	T_B	Singular
Logar	$GF(3)$	7	8	on	32	32	32	>5 min
Logar	$GF(53)$	7	8	on	92	95	93	32
8_3	$GF(3)$	5	3	on	1	1	1	0.5
8_3	$GF(53)$	5	3	on	14	14	14	2
Q_{765}	$GF(3)$	1	1	on	22	22	>5 min	>5 min
Q_{4321}	$GF(3)$	2	2	on	1	1	3	>5 min
R_{543}	$GF(5)$	1	6	on	1.2	1.6	1.2	>5 min
$Hcyclic(6)$	$GF(3)$	2	6	off	2.5	2.5	2.5	>5 min
$Hcyclic(6)$	$GF(53)$	2	65	off	39	39	38	>5 min
$Hkatsura(6)$	$GF(3)$	1	6	off	5	5	5	26
$Hkatsura(6)$	$GF(53)$	1	10	off	22	22	80	>5 min
$Hkatsura(7)$	$GF(53)$	1	11	off	238	229	1190	>1 h

same as their counterparts if each decomposition has the same form as that over the prime field. For those over $GF(2^{13})$, $GF(3^7)$, $GF(5^5)$ and $GF(53^2)$, the elapsed computation times do not exceed those over the corresponding prime fields $GF(2)$, $GF(3)$, $GF(5)$ and $GF(53)$.

6. Concluding remarks

We have implemented an algorithm for prime decomposition of polynomial ideals over small finite fields on a computer, and have evaluated the practicality and quality of our implementation by computational experiments on several examples.

As prime decomposition consists of many sub-procedures, the efficiencies of the sub-procedures and basic arithmetical operations affect the overall efficiency. Moreover, the choice of strategy (i.e. combination of sub-procedures) also affects the overall efficiency. We summarize our results and give recommendations for future work:

- Partial decomposition

  To compute the prime divisors of I, we can incorporate existing decomposition algorithms (Caboara et al., 1997; Shimoyama and Yokoyama, 1996). We have employed two types of partial decomposition: pre-decomposition and intermediate decomposition. In twofold partial decomposition, many of the intermediate ideals tend to be prime. Thus, a special procedure for inseparable intermediate ideals is applied only to a limited number of such ideals.
- Computation of minimal polynomials

  The experiments showed that direct computation of minimal polynomials is efficient in many cases. In some cases, however, elimination by the Buchberger algorithm is more efficient than the direct method. By applying competitive computation, we can benefit from both methods. Even if a single CPU machine is used, total elapsed time does not exceed twice the time taken by the faster algorithm, and we eliminate large delays that can result from choosing a single inefficient algorithm.

- Basic arithmetical operations, Gröbner basis computation and polynomial factorization

  Computation over an extension field can be reduced to computation over the ground field by considering the defining polynomial of the extension as a member of the ideal to be decomposed, but this adds a substantial overhead. As we have implemented such extension fields as "primitive data types" and implemented all necessary functionality over them, the performance losses due to field extension are negligible. However, the performance of Gröbner basis computation over finite fields is not fully optimized and affects overall performance in some cases. Although the multivariate factorizer is efficient in most cases, we often encounter minimal polynomials that are difficult to factor using the standard modular method. Fortunately a practical method for factorizing such polynomials has been developed (Noro and Yokoyama, 2002), by which we are able to factorize them efficiently.
- Special procedure for non-generic and inseparable cases

  Experimentally, the special procedure aimed at inseparable cases works primarily in cases of small characteristic such as $p = 2, 3, 5$, because, in practice, it is difficult to handle ideals J with large $\dim_L(L[Z]/J)$. (If the characteristic p is not small, the ideals that we can handle are separable and we simply apply ordinary *decomposition using generic position*.) Thus, in order to utilize such small characteristics, it is important to improve the efficiency of the basic arithmetical operations. The Zech representation is employed in our implementation for this reason. For zero-dimensional ideals over $GF(q)$, there always exists a polynomial in generic position. It is thus better to design another method optimized for zero-dimensional ideals over $GF(q)$.
- Early Termination

  Unnecessary computations were avoided by employing an *Early Termination* scheme. As the checking procedure can be performed without computing the radical ideal of the input I, *Early Termination* was found to improve overall efficiency very much in a number of test cases. However, the check requires additional computation, ideal intersection computation and radical membership computation, and therefore offers a "trade-off". It seems very difficult to estimate the complexity of the algorithm theoretically. And thus we will search for a good strategy based on further experimental work.

References

Adams, W.W., Loustaunau, P., 1994. An Introduction to Gröbner Bases. Graduate Studies in Mathematics, vol. 3. American Mathematical Society.

Anai, H., Noro, M., Yokoyama, K., 1996. Computation of the splitting fields and the Galois groups of polynomials. In: Algorithms in Algebraic Geometry and Applications. Birkhäuser, Basel, pp. 29–50.

Becker, T., Weispfenning, V., 1993. Gröbner Bases. Springer-Verlag, New York.

Bernardin, L., Monagan, M.B., 1997. Efficient multivariate factorization over finite fields. In: Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-12). Lecture Notes in Computer Science, vol. 1255. pp. 15–28.

Caboara, M., Conti, P., Traverso, C., 1997. Yet another ideal decomposition algorithm. In: Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (AAECC-12). Lecture Notes in Computer Science, vol. 1255. pp. 39–54.

Decker, D., Greuel, G.-M., Pfister, P., 1999. Primary decomposition: algorithms and comparisons. In: Algorithmic Algebra and Number Theory. Springer, pp. 187–220.

de Jong, T., 1998. An algorithm for computing the integral closure. J. Symbolic Comput. 26, 273–277.

Eisenbud, D., Huneke, C., Vasconcelos, W.V., 1992. Direct methods for primary decomposition. Invent. Math. 110, 207–235.

Faugère, J.-C., Gianni, P., Lazard, D., Mora, T., 1993. Efficient computation of zero-dimensional Gröbner bases by change of ordering. J. Symbolic Comput. 16, 329–344.

Fortuna, E., Gianni, P., Trager, B., 2002. Derivations and radicals of polynomial ideals over fields of arbitrary characteristic. J. Symbolic Comput. 33, 609–625.

Gianni, P., Trager, B., 1996. Square-free algorithms in positive characteristic. Appl. Algebra Engrg. Comm. Comput. 7, 1–14.

Gianni, P., Trager, B., Zacharias, G., 1988. Gröbner bases and primary decomposition of polynomial ideals. J. Symbolic Comput. 6, 149–167.

Kalkbrener, M., 1994. Prime decomposition of radicals in polynomial rings. J. Symbolic Comput. 18, 365–372.

Kemper, G., 2002. The calculation of radical ideals in positive characteristic. J. Symbolic Comput. 34, 229–238.

Kemper, G., Mattig, E., 2000. Generic polynomials with few parameters. J. Symbolic Comput. 30, 843–858.

Maekawa, M., Noro, M., Ohara, K., Takayama, N., Tamura, Y., 2001. The design and implementation of OpenXM-RFC 100 and 101. In: Proceedings of ASCM2001. World Scientific, pp. 102–111.

Matsumoto, R., 2001. Computing the radical of an ideal in positive characteristic. J. Symbolic Comput. 32, 263–271.

Noro, M., Yokoyama, K., 1999. A modular method to compute rational univariate representation of zero-dimensional ideals. J. Symbolic Comput. 28, 243–263.

Noro, M., Yokoyama, K., 2002. Yet another practical implementation of polynomial factorization over finite fields. In: Proceedings of ISSAC' 02. ACM Press, pp. 200–206.

Seidenberg, A., 1974. Constructions in algebra. Trans. Amer. Math. Soc. 197, 272–313.

Shimoyama, T., Yokoyama, K., 1996. Localization and primary decomposition of polynomial ideals. J. Symbolic Comput. 22, 247–277.

Wu, W.-T., 1984. Basic principles of mechanical theorem proving in elementary geometry. J. Systems Sci. Math. Sci. 4, 207–235.

Yokoyama, K., 2002. Prime decomposition of polynomial ideals over finite fields. In: Mathematical Software (Proceedings of ICMS2002). World Scientific, pp. 217–227.

Yokoyama, K., Noro, M., Takeshima, T., 1992. Solution of systems of algebraic equations and linear maps on residue class rings. J. Symbolic Comput. 14, 399–417.