J. Symbolic Computation (2001) **32**, 263–271
doi:10.1006/jsco.2001.0446
Available online at http://www.idealibrary.com on IDEAL®

Computing the Radical of an Ideal in Positive Characteristic

RYUTAROH MATSUMOTO[†]

Uyematsu Laboratory, Department of Communications and Integrated Systems, Tokyo Institute of Technology, 2-12-1, Ookayama, Meguro-ku, Tokyo, 152-8552 Japan

We propose a method for computing the radical of an arbitrary ideal in the polynomial ring in n variables over a perfect field of characteristic $p > 0$. In our method Buchberger's algorithm is performed once in n variables and a Gröbner basis conversion algorithm is performed at most $\lceil n \log_p d \rceil$ times in $2n$ variables, where d is the maximum of total degrees of generators of the ideal and 3. Next we explain how to compute radicals over a finitely generated coefficient field over a field K, when we have a radical computation method over the field K. Thus we can compute radicals over any finitely generated field over a perfect field.

1. Introduction

Let K be a field, $K[X_1, \dots, X_n]$ be the polynomial ring in n variables over K, and $I \subset K[X_1, \dots, X_n]$ be an ideal. The radical of I is

$$\sqrt{I} = \{x \in K[X_1, \dots, X_n] \mid \exists i,\ x^i \in I\}.$$

We define the index of nilpotency of $x \in \sqrt{I}$ with respect to I as

$$\mathrm{nil}(x, I) := \min\{i \mid x^i \in I\}.$$

The dimension of an ideal I is the Krull dimension of its residue ring $K[X_1, \dots, X_n]/I$. If I is zero dimensional, there are well-known methods for computing $\sqrt{I}$. For an ideal of positive dimension, various algorithms computing $\sqrt{I}$ have been proposed (Wu, 1984; Gianni *et al.*, 1988; Krick and Logar, 1991; Eisenbud *et al.*, 1992; Wang, 1993; Armendáriz and Pablo, 1995; Caboara *et al.*, 1997; Wang, 1998). All of them successfully compute the radical of an ideal when the characteristic is sufficiently large (or 0). Among them the methods of Wu (1984), Eisenbud *et al.* (1992), Wang (1993) and Caboara *et al.* (1997) are applicable for small positive characteristic.

The method of Eisenbud *et al.* (1992) is only applicable for an ideal I whose radical $\sqrt{I}$ is generated by elements whose indexes of nilpotency with respect to I are less than the characteristic. Thus they posed the problem of computing the radical of an arbitrary ideal in a polynomial ring of positive characteristic. This problem was solved by Wu (1984), Wang (1993) and Caboara *et al.* (1997), though Wu (1984) and Wang (1993) did not claim that their methods were applicable for positive characteristic.

[†] E-mail: `ryutaroh@rmatsumoto.org`; `WWW:http://www.rmatsumoto.org/`

0747–7171/01/090263 + 09 $35.00/0

The irreducible characteristic set method (Wu, 1984) and its improvement (Wang, 1993) compute the minimal associated primes of an ideal. Its radical can be computed from their intersection. The further improved method proposed by Wang (1998) can fail in positive characteristic. The method of Caboara *et al.* (1997) computes pure-dimensional radical ideals $I_1, \dots, I_l$ for a given ideal I, where each I_i is the intersection of associated primes of the same dimension. $\sqrt{I}$ is easily computed from those I_i's.

Yanagawa and Hashimoto (1998) briefly mentioned that the radical can be computed by successively computing the kernel of the q-th power endomorphism when the coefficient field is the finite field with q elements. In this paper, we make their method applicable to a wider class of coefficient field of positive characteristic p, efficient for a huge finite coefficient field, and minimize the number of times we perform Buchberger's algorithm. In our method Buchberger's algorithm is performed once in n variables and a Gröbner basis conversion algorithm is performed at most $\lceil n \log_p d \rceil$ times in $2n$ variables, where d is the maximum of total degrees of generators of the ideal and 3. This method depends on the coefficient field and it can be performed only over a perfect field in which we can compute the inverse of the Frobenius automorphism. Next we explain how to compute radicals over a finitely generated coefficient field over a field K, when we have a radical computation method over the field K. Finally we evaluate the performance of the proposed algorithm by applying it to examples from Caboara *et al.* (1997). It should be stressed that the proposed algorithm is computationally intractable when the characteristic is not sufficiently small.

2. An Algorithm Computing the Radical of an Ideal in Positive Characteristic

Let K be an arbitrary field of characteristic $p > 0$. We will find the radical of a proper ideal $I \subset R := K[X_1, \dots, X_n]$. Let q be a power of p. Consider the endomorphism φ:

$$\begin{aligned} R &\longrightarrow R, \\ x &\longmapsto x^q. \end{aligned}$$

Since $(a+b)^q = a^q + b^q$ for $a, b \in R$, φ is actually an endomorphism of R.

We consider the ideal

$$\varphi^{-1}(I) := \{x \in R \mid \varphi(x) \in I\}.$$

We have

$$I \subseteq \varphi^{-1}(I) \subseteq \sqrt{I}.$$

Moreover the following holds.

Proposition 2.1. *(1) $\varphi^{-1}(I)$ is strictly larger than I if $I \neq \sqrt{I}$.*
(2) For $x \in \sqrt{I} \setminus \{0\}$ we have

$$\mathrm{nil}(x, \varphi^{-1}(I)) = \lceil \mathrm{nil}(x, I)/q \rceil.$$

Proof. (1) Suppose there is an element $x \in \sqrt{I} \setminus I$. Then $x^{\lceil \mathrm{nil}(x,I)/q \rceil} \in \varphi^{-1}(I) \setminus I$.
(2)

$$\begin{aligned} q(\lceil \mathrm{nil}(x, I)/q \rceil - 1) &< \mathrm{nil}(x, I), \\ q\lceil \mathrm{nil}(x, I)/q \rceil &\geq \mathrm{nil}(x, I). \end{aligned}$$

By the inequalities above, we have

$$\begin{aligned} x^{\lceil \mathrm{nil}(x,I)/q \rceil - 1} &\notin \varphi^{-1}(I), \\ x^{\lceil \mathrm{nil}(x,I)/q \rceil} &\in \varphi^{-1}(I). \ \square \end{aligned}$$

If $I \neq \sqrt{I}$ then by the previous proposition we have the chain of strict inclusion of ideals:

$$I \subsetneq \varphi^{-1}(I) \subsetneq \varphi^{-2}(I) \subsetneq \cdots.$$

By the ascending chain condition there exists j such that

$$\varphi^{-j+1}(I) \subsetneq \varphi^{-j}(I) = \sqrt{I}.$$

We will bound the value of j from above. If $n = 1$, we can use the square-free algorithm for univariate polynomials proposed by Gianni and Trager (1996). We may assume $n \geq 2$.

PROPOSITION 2.2. (VASCONCELOS, 1998, PROPOSITION 9.2.1) *We assume $n \geq 2$. Let d be the maximum of total degrees of generators of I and 3. For $x \in \sqrt{I}$,*

$$\mathrm{nil}(x, I) \leq d^n.$$

Note that d depends on a generating set of I and is not uniquely determined by I.

PROPOSITION 2.3. *(1) $j = \lceil \log_q \max\{\mathrm{nil}(x, I) \mid x \in \sqrt{I}\} \rceil$.*
(2) $j \leq \lceil n \log_q d \rceil$.

PROOF. The second assertion follows from the first one and the previous proposition. We will prove the first. For a positive integer i, we define

$$\begin{aligned} \sigma(i) &:= \lceil i/q \rceil, \\ \tau(i) &:= \min\{m \mid \sigma^m(i) = 1\}. \end{aligned}$$

Then $j = \tau(\max\{\mathrm{nil}(x, I) \mid x \in \sqrt{I}\})$. Let $l := \lceil \log_q i \rceil$. Then

$$\begin{array}{rcl} q^{l-1} < & i & \leq q^l, \\ \sigma(q^{l-1}) = q^{l-2} < & \sigma(i) & \leq q^{l-1} = \sigma(q^l), \\ & \vdots & \\ \sigma^{l-1}(q^{l-1}) = 1 < & \sigma^{l-1}(i) & \leq q = \sigma^{l-1}(q^l). \end{array}$$

Thus $\tau(i) = \lceil \log_q i \rceil$. $\square$

We next consider how to compute $\varphi^{-1}(I)$.

DEFINITION 2.4. We define the endomorphism φ_{c} of R as

$$\sum a_{m_1 \cdots m_n} X_1^{m_1} \cdots X_n^{m_n} \longmapsto \sum a_{m_1 \cdots m_n}^q X_1^{m_1} \cdots X_n^{m_n},$$

and the endomorphism φ_{v} of R as

$$f(X_1, \ldots, X_n) \longmapsto f(X_1^q, \ldots, X_n^q).$$

Since $\varphi = \varphi_{\mathrm{c}} \circ \varphi_{\mathrm{v}} = \varphi_{\mathrm{v}} \circ \varphi_{\mathrm{c}}$,

$$\varphi^{-1}(I) = \varphi_{\mathrm{c}}^{-1}(\varphi_{\mathrm{v}}^{-1}(I)) = \varphi_{\mathrm{v}}^{-1}(\varphi_{\mathrm{c}}^{-1}(I)).$$

$\varphi_{\mathrm{v}}^{-1}$ can be computed with Buchberger's algorithm and a Gröbner basis conversion as follows.

Proposition 2.5. (Adams and Loustaunau, 1994, Theorem 2.4.10) *Let*

$$J := IK[X_1, \ldots, X_n, Y_1, \ldots, Y_n] + (Y_1 - X_1^q, \ldots, Y_n - X_n^q).$$

Then $\varphi_{\mathrm{v}}^{-1}(I)$ is $J \cap K[Y_1, \ldots, Y_n]$ with Y_i replaced by X_i for $i = 1, \ldots, n$.

A generating set of $J \cap K[Y_1, \ldots, Y_n]$ can be computed from a Gröbner basis for J with respect to an elimination order with the X variables larger than the Y variables (Adams and Loustaunau, 1994, Definition 2.3.1 and Theorem 2.3.4). Although the Gröbner basis can be computed with Buchberger's algorithm, we can use Gröbner basis conversion algorithms (Faugère *et al.*, 1993; Traverso, 1996; Collart *et al.*, 1997; Amrhein *et al.*, 1997; Tran, 2000), because we can immediately derive a Gröbner basis for J with respect to an elimination order with the Y variables larger than the X variables from that for I as follows. Basis conversion algorithms are considered faster than Buchberger's algorithm in many cases.

Proposition 2.6. *Let $\prec_X$be a monomial order on $X_1, \ldots, X_n$, and $\prec_Y$that on $Y_1, \ldots, Y_n$. We define the product monomial order $\prec_{XY}$ of $\prec_X$ and $\prec_Y$ to be*

$$X_1^{a_1} \cdots X_n^{a_n} Y_1^{b_1} \cdots Y_n^{b_n} \prec_{XY} X_1^{c_1} \cdots X_n^{c_n} Y_1^{d_1} \cdots Y_n^{d_n}$$
$$\iff \begin{cases} Y_1^{b_1} \cdots Y_n^{b_n} \prec_Y Y_1^{d_1} \cdots Y_n^{d_n} \\ \textit{or} \\ Y_1^{b_1} \cdots Y_n^{b_n} = Y_1^{d_1} \cdots Y_n^{d_n} \textit{ and } X_1^{a_1} \cdots X_n^{a_n} \prec_X X_1^{c_1} \cdots X_n^{c_n}. \end{cases}$$

Note that $\prec_{XY}$ is an elimination order with Y larger than X. Suppose that we have a Gröbner basis G for an ideal $I \subset K[X_1, \ldots, X_n]$ with respect to the monomial order $\prec_X$. Then $G \cup \{Y_1 - X_1^q, \ldots, Y_n - X_n^q\}$ is a Gröbner basis for J with respect to $\prec_{XY}$, where J is defined as the previous proposition.

Proof. LM denotes the leading monomial of a polynomial. For distinct $f, g \in G$, the remainder on division of the S-polynomial $S(f, g)$ by $G \cup \{Y_1 - X_1^q, \ldots, Y_n - X_n^q\}$ is zero, because

$$\mathrm{LM}(Y_i - X_i^q) = Y_i \succ_{XY} \text{a monomial in } X = \mathrm{LM}(S(f, g)),$$

and G is a Gröbner basis for I. For $Y_i - X_i^q$ and $Y_i - X_i^q \neq f \in G \cup \{Y_1 - X_1^q, \ldots, Y_n - X_n^q\}$, $S(f, Y_i - X_i^q)$ reduces to zero modulo $G \cup \{Y_1 - X_1^q, \ldots, Y_n - X_n^q\}$, because the leading monomials of f and $Y_i - X_i^q$ are relatively prime (Cox *et al.*, 1996, Proposition 4 in Section 2.9). Since all S-polynomials reduce to zero, $G \cup \{Y_1 - X_1^q, \ldots, Y_n - X_n^q\}$ is a Gröbner basis (Cox *et al.*, 1996, Theorem 3 in Section 2.9). □

Remark 2.7. We can compute the kernel of any homomorphism from a polynomial ring to an affine ring by Gröbner basis conversion.

The remaining problem is how to compute $\varphi_{\mathrm{c}}^{-1}(I)$. When K is a perfect field, the restriction $\varphi_{\mathrm{c}}|K$ is an automorphism of K, thus $\varphi_{\mathrm{c}}^{-1}$ is an automorphism of $K[X_1,\dots,X_n]$ and

$$\begin{aligned}&\varphi_{\mathrm{c}}^{-1}(I)\\ &=\{x\in R \mid \varphi_{\mathrm{c}}(x)\in I\}\\ &=\{\varphi_{\mathrm{c}}^{-1}(y)\mid y\in I\}.\end{aligned}$$

If I is generated by $F_1,\dots,F_s$, $\varphi_{\mathrm{c}}^{-1}(I)$ is generated by $\varphi_{\mathrm{c}}^{-1}(F_1),\dots,\varphi_{\mathrm{c}}^{-1}(F_s)$. Thus if the inverse $\varphi_{\mathrm{c}}^{-1}|K$ of the qth power automorphism is computable in K, we can compute $\varphi_{\mathrm{c}}^{-1}(I)$.

If K is the finite field with p^m elements, then the restriction of φ_{c} to $\mathbf{F}_{p^m}$ is

$$\begin{aligned}\mathbf{F}_{p^m}&\longrightarrow \mathbf{F}_{p^m},\\ \alpha&\longmapsto \alpha^{p^{(\log_p q)\bmod m}}.\end{aligned}$$

Thus the inverse automorphism $\varphi_{\mathrm{c}}^{-1}|\mathbf{F}_{p^m}$ is

$$\begin{aligned}\mathbf{F}_{p^m}&\longrightarrow \mathbf{F}_{p^m},\\ \alpha&\longmapsto \alpha^{p^{m-(\log_p q \bmod m)}}.\end{aligned}$$

If K is not perfect, a general method computing $\varphi_{\mathrm{c}}^{-1}(I)$ is not known. But we can get around computation of $\varphi_{\mathrm{c}}^{-1}(I)$ if K is a finitely generated field over a perfect field, as described in the next section.

Summing up the results in this section, we get the following algorithm for computing the radical of an ideal.

Algorithm 2.8.
Input: A Gröbner basis B (with respect to any order) for a proper ideal I of the polynomial ring $K[X_1,\dots,X_n]$, and a power q of the characteristic p.
Output: A Gröbner basis for the radical $\sqrt{I}$.

Description of the Algorithm.

(1) Let $B=\{F_1,\dots,F_s\}$. Compute $\varphi_{\mathrm{c}}^{-1}(F_i)$ for $i=1,\dots,s$ (recall the definition of φ_{c} in Definition 2.4).

(2) Compute a Gröbner basis B' for $\{\varphi_{\mathrm{c}}^{-1}(F_1),\dots,\varphi_{\mathrm{c}}^{-1}(F_s), Y_1-X_1^q,\dots,Y_n-X_n^q\}$ with respect to an elimination order with the X variables larger than the Y variables, by either Buchberger's algorithm or a Gröbner basis conversion algorithm. Let B'' be $B'\cap K[Y_1,\dots,Y_n]$ with Y_i replaced by X_i for each i.

(3) If the ideal generated by B'' is equal to that by B, then output B'' and terminate the algorithm. Otherwise let $B=B''$ and return to the first step.

By Proposition 2.3 the number of iterations in the algorithm is less than or equal to $\lceil n\log_q d\rceil$, where d is the maximum of total degrees of generators of I and 3.

Let us give an example illustrating the algorithm. Consider the following ideal in characteristic 7 borrowed from Eisenbud *et al.* (1992, p.214):

$$I=(z^7-xyu^5, y^4-x^3u).$$

Step 1 in Algorithm 2.8 does not change the generating set of the ideal I, because φ_{c} in Definition 2.4 is the identity map in this case.

In Step 2 in Algorithm 2.8, we compute $\varphi_{\mathrm{v}}^{-1}(I)$. By Proposition 2.5 we consider the following ideal and eliminate the variables u, x, y, z:

$$(z^7 - xyu^5, y^4 - x^3u, u^7 - U, x^7 - X, y^7 - Y, z^7 - Z)$$

We can eliminate the variables by computing a Gröbner basis with respect to an elimination order with u, x, y, z larger than U, X, Y, Z, which is

$$\begin{array}{lll}
-UX + YZ, & -Y^3 + X^2Z, & -UY^2 + XZ^2, \\
U^2Y - Z^3, & -ux^3 + y^4, & z^7 - Z, \\
y^7 - Y, & x^7 - X, & u^7 - U, \\
-u^5xy + z^7, & -x^4y^4 + uX, & u^2X - xyY, \\
-u^2y^6Y + xXZ, & -u^4y^5 + x^2Z, & -xyU + u^2Z, \\
-y^5U + u^3x^2Z, & u^3x^2y^2Z - UY, & -u^2y^6Z + xUY, \\
u^2UY - xyZ^2, & -u^2Y^2 + xyXZ, & -u^6y^4 + x^3U, \\
uy^3X - x^4Y, & -u^4Y + x^2y^2Z, & x^4U - uy^3Z, \\
uy^3UY - x^4Z^2, & -uy^3Y^2 + x^4XZ, & -u^3y^2Y + x^5Z.
\end{array}$$

The set of polynomials containing none of u, x, y, z is $\{-UX + YZ,\ -Y^3 + X^2Z,\ -UY^2 + XZ^2,\ U^2Y - Z^3\}$. By Proposition 2.5, $\ker(\varphi_{\mathrm{v}})$ is generated by $\{-ux + yz,\ -y^3 + x^2z,\ -uy^2 + xz^2,\ u^2y - z^3\}$. If we apply Steps 1 and 2 in Algorithm 2.8, the result is again $\{-ux + yz,\ -y^3 + x^2z,\ -uy^2 + xz^2,\ u^2y - z^3\}$. Thus we conclude that $\sqrt{I}$ is generated by $\{-ux + yz,\ -y^3 + x^2z,\ -uy^2 + xz^2,\ u^2y - z^3\}$.

3. Lifting Arbitrary Radical Computation Methods Over Some Field to a Finitely Generated Field

Suppose that we have a method computing the radical of an ideal in a polynomial ring over some field K. In this section we explain how we can compute the radical of an ideal in a polynomial ring over $K(t_1, \ldots, t_l)$, a field finitely generated over K. The method presented here is applicable to lifting any radical computation methods. With this method we can compute radicals in a polynomial ring over a field finitely generated over a perfect field.

Let $K[T_1, \ldots, T_l]$ be the polynomial ring in l variables over K and $S := K[t_1, \ldots, t_l, X_1, \ldots, X_n]$. Consider the following evaluation map ρ:

$$\begin{aligned}
K[T_1, \ldots, T_l, X_1, \ldots, X_n] &\longrightarrow S, \\
f(T_1, \ldots, T_l, X_1, \ldots, X_n) &\longmapsto f(t_1, \ldots, t_l, X_1, \ldots, X_n).
\end{aligned}$$

Let $M := K[t_1, \ldots, t_l] \setminus \{0\}$, then we have

$$K(t_1, \ldots, t_l)[X_1, \ldots, X_n] = (K[T_1, \ldots, T_l, X_1, \ldots, X_n]/\ker\rho)_M.$$

For a proper ideal $I \subset K(t_1, \ldots, t_l)[X_1, \ldots, X_n]$, we have

$$\sqrt{I} = S_M \rho \sqrt{\rho^{-1}(I \cap S)}$$

and we assume that we know how to compute $\sqrt{\rho^{-1}(I \cap S)}$ from $\rho^{-1}(I \cap S) \subset K[T_1, \ldots, T_l, X_1, \ldots, X_n]$.

In order to compute $\sqrt{I}$ from I, we have to compute $\rho^{-1}(I \cap S)$ from I and $S_M \rho \sqrt{\rho^{-1}(I \cap S)}$ from $\sqrt{\rho^{-1}(I \cap S)}$. We know how to compute $\rho^{-1}(I \cap S)$ from $I \cap S$ and $S_M \rho \sqrt{\rho^{-1}(I \cap S)}$ from $\sqrt{\rho^{-1}(I \cap S)}$. The remaining task is computing $I \cap S$ from I.

Proposition 3.1. (Gianni *et al.*, 1988, Corollary 3.8) *Let R be an integral domain with the quotient field Q. Let $G \subset R[X_1, \ldots, X_n]$ be a Gröbner basis for an ideal $I \subset Q[X_1, \ldots, X_n]$ and J be the ideal in $R[X_1, \ldots, X_n]$ generated by G. We define*

$$f := \prod_{g \in G} \mathrm{LC}(g),$$

where $\mathrm{LC}(g)$ is the leading coefficient of $g \in G$ as a polynomial over the coefficient field Q. Then we have

$$I \cap R[X_1, \ldots, X_n] = J : f^{\infty}.$$

By applying the previous proposition with $R = K[t_1, \ldots, t_l]$ and $Q = K(t_1, \ldots, t_l)$, we can compute $I \cap S$ from I.

Remark 3.2. When we can compute radicals over some field K, we can compute those over a finitely generated field over K. This fact can be viewed as a generalization of the fact that if we can compute the square-free decomposition of any univariate polynomial over K, then we can do it over any finitely generated field over K (Gianni and Trager, 1996).

4. Performance Evaluation

In this section, we apply the proposed algorithm to several examples in Caboara *et al.* (1997), and evaluate its performance. For original sources and backgrounds of the following examples, refer to Caboara *et al.* (1997).

E2: $x + 3xy^3 + y^4 + yz^2$, $-x^2z + 2y^3z + z^2 + 2yz^2 + 3xyz^2$, $3x^3 + xy^2 + yz^2 - 2xz^3$.

E3: $x^2 + y^4 + x^3z + yz - 2xz^3$, $-x^2y^2 - y^3z - z^3 - 3yz^3$, $y^4 - x^2z + 2y^2z - 2xyz^2$.

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

M: $-a^3d + b^4$, $-a^3c + ab^3$, $-ac^3d + ad^4 + bc^4 - bcd^3$, $-bc^3d^2 + bd^5 + c^6 - c^3d^3$, $ac^5 - ac^2d^3 - b^2c^3d + b^2d^4$, $-a^3d^3 + a^2c^4 - a2cd^3 + b^3d^3$, $-a^3d^3 + b^3c^3$, $-a^3cd^2 + ab^2c^3 - ab^2d^3 + b^3cd^2$, $-a^3c^2d + a^2bc^3 - a^2bd^3 + b^3c^2d$, $-a^3bd^2 + a^3c^3$, $a^4c^2 - a^3b^2d$.

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

C: $a_1b_2 + b_1x_2 + x_1a_2 - a_2b_1 - b_2x_1 - x_2a_1$, $b_1c_2 + c_1y_2 + y_1b_2 - b_2c_1 - c_2y_1 - y_2b_1$, $a_1c_2 + c_1z_2 + z_1a_2 - a_2c_1 - c_2z_1 - z_2a_1$, $c_1o_2 + o_1x_2 + x_1c_2 - c_2o_1 - o_2x_1 - x_2c_1$, $a_1o_2 + o_1y_2 + y_1a_2 - a_2o_1 - o_2y_1 - y_2a_1$, $b_1o_2 + o_1z_2 + z_1b_2 - b_2o_1 - o_2z_1 - z_2b_1$, a_1, a_2, $b_1 - 1$, b_2.

We applied the procedure given in Figure 1 on the Singular computer algebra system version 1.2.3 developed by Greuel *et al.* (1998). The computer used was an Intel Celeron 300 MHz with 128 MB memory. The running time is tabulated in Table 1, where ">1800 s" indicates that the computation did not end in 30 minutes, and "out of memory" indicates that it was not completed within 48 MB of memory. From Table 1, we can

```
proc rad2(ideal i)
{
  def R=basering;
  int p=char(R);
  ideal k;
  int l;
  ideal j;
  for(l=nvars(R);l>=1;l--)
  {
     j=maxideal(1);
     j[l]=var(l)^p;
     map phi(l)=R,j;
  }
  k=i;
  while(1)
  {
     for(l=nvars(R);l>=1;l--)
     {
       k=preimage(R,phi(l),k);
     }
     if(size(reduce(k,std(i),1))==0)
     {
        return(i);
     }
     i=k;
   }
}
```

Figure 1. An implementation of the proposed algorithm in singular.

Table 1. Timing Results.

Characteristic	**E2**	**E3**	**L**	**M**	$\mathbf{8}_3$	**C**
2	1 s	1 s	58 s	1 s	5 s	3 s
3	1 s	3 s	>1800 s	1 s	5 s	105 s
5	7 s	5 s	>1800 s	1 s	>1800 s	out of memory
7	7 s	10 s	>1800 s	1 s	>1800 s	out of memory
11	5 s	9 s	>1800 s	1 s	>1800 s	out of memory
53	161 s	358 s	>1800 s	1 s	>1800 s	out of memory
251	>1800 s	>1800 s	>1800 s	25 s	>1800 s	out of memory

see that the proposed algorithm is rather efficient when the characteristic is small. However, it is computationally intractable when the characteristic is not sufficiently small.

Let φ_i be the endomorphism of $K[X_1, \ldots, X_n]$ sending $f(X_1, \ldots, X_i, \ldots, X_n)$ to $f(X_1, \ldots, X_i^q, \ldots, X_n)$. Note that in Figure 1 $(\varphi_1 \circ \cdots \circ \varphi_n)^{-1}$ is computed instead of $\varphi_{\mathrm{v}}^{-1}$. Computing $(\varphi_1 \circ \cdots \circ \varphi_n)^{-1}$ is usually faster than computing $(\varphi_{\mathrm{v}})^{-1}$ directly.

Acknowledgements

We would like to thank Dr. Kazuhiro Yokoyama for stimulating discussions, the anonymous referees for helpful comments, in particular, for providing the procedure given in Figure 1, and Mr Konstantinos Slavakis for pointing out shortcomings in the manuscript. R. Matsumoto is supported by the JSPS Research Fellowship for Young Scientists.

References

Adams, W. W., Loustaunau, P. (1994). *An Introduction to Gröbner Bases, Graduate Studies in Mathematics* **3**. Providence, RI, American Mathematical Society.

Amrhein, B., Gloor, O., Kuchlin, W. (1997). On the walk. *Theor. Comput. Sci.*, **187**, 179–202.

Armendáriz, I., Pablo, S. (1995). On the computation of the radical of polynomial complete intersection ideals. In Cohen, G. *et al.* eds, *Proc. AAECC-11*, *Lecture Notes in Computer Science* **948**, pp. 106–119. Berlin, Springer-Verlag.

Caboara, M., Conti, P., Traverso, C. (1997). Yet another algorithm for ideal decomposition. In Mora, T., Mattson, H. eds, *Proc. AAECC-12, Lecture Notes in Computer Science* **1255**, pp. 39–54. Berlin, Springer Verlag.

Collart, S., Kalkbrener, M., Mall, D. (1997). Converting bases with the Gröbner walk. *J. Symb. Comput.*, **24**, 465–469.

Cox, D., Little, J., O'Shea, D. (1996). *Ideals, Varieties, and Algorithms*, 2$^{\text{nd}}$ edn. Berlin, Springer-Verlag.

Eisenbud, D., Huneke, C., Vasconcelos, W. (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. Symb. Comput.*, **16**, 329–344.

Gianni, P., Trager, B. (1996). Square-free algorithms in positive characteristic. *Appl. Algebra Eng. Commun. Comput.*, **7**, 1–14.

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

Greuel, G.-M., Pfister, G., Schönemann, H. (1998). Singular version 1.2 user manual. In *Reports On Computer Algebra*, volume 21. Centre for Computer Algebra, University of Kaiserslautern, Germany, `http://www.singular.uni-kl.de/`

Krick, T., Logar, A. (1991). An algorithm for the computation of the radical of an ideal in the ring of polynomials. In Mattson, H. F. *et al.* eds, *Proc. AAECC-9, Lecture Notes in Computer Sciences* **539**, pp. 195–205. Berlin, Springer-Verlag.

Tran, Q. -N. (2000). A fast algorithm for Gröbner basis conversion and its applications. *J. Symb. Comput.*, **30**, 451–467.

Traverso, C. (1996). Hilbert functions and the Buchberger algorithm. *J. Symb. Comput.*, **22**, 355–376.

Vasconcelos, W. V. (1998). *Computational Methods in Commutative Algebra and Algebraic Geometry, Algorithms and Computation in Mathematics* **2**. Berlin, Springer-Verlag.

Wang, D. (1993). An elimination method for polynomial systems. *J. Symb. Comput.*, **16**, 83–114.

Wang, D. (1998). Unmixed and prime decomposition of radicals of polynomial ideals. *SIGSAM Bull.*, **32**(4), 2–9.

Wu, W. -T. (1984). Basic principles of mechanical theorem proving in elementary geometry. *J. Syst. Sci. Math. Sci.*, **4**, 207–235, republished as Wu (1986).

Wu, W. -T. (1986). Basic principles of mechanical theorem proving in elementary geometry. *J. Autom. Reasoning*, **2**, 221–252.

Yanagawa, K., Hashimoto, M. (1998). Computations on determinantal ideals and normality check by Macaulay 2. *Chûbu Forum for Mathematical Sciences*, **3**, 33–55 (in Japanese).

Received 5 June 1998
Accepted 20 March 2001
Published electronically 21 May 2001