Journal of Symbolic Computation 46 (2011) 672–684

Contents lists available at ScienceDirect

Journal of Symbolic Computation

journal homepage: www.elsevier.com/locate/jsc

Parallelization of Modular Algorithms☆

Nazeran Idrees [a], Gerhard Pfister [b], Stefan Steidel [b,1]

[a] Abdus Salam School of Mathematical Sciences, GC University, Lahore, 68-B, New Muslim Town, Lahore 54600, Pakistan
[b] Department of Mathematics, University of Kaiserslautern, Erwin-Schrödinger-Str., 67663 Kaiserslautern, Germany

ARTICLE INFO

Article history:
Received 28 May 2010
Accepted 25 January 2011
Available online 3 February 2011

Keywords:
Gröbner bases
Primary decomposition
Modular computation
Parallel computation

ABSTRACT

In this paper we investigate the parallelization of two modular algorithms. In fact, we consider the modular computation of Gröbner bases (resp. standard bases) and the modular computation of the associated primes of a zero-dimensional ideal and describe their parallel implementation in SINGULAR. Our modular algorithms for solving problems over $\mathbb{Q}$ mainly consist of three parts: solving the problem modulo p for several primes p, lifting the result to $\mathbb{Q}$ by applying the Chinese remainder algorithm (resp. rational reconstruction), and verification. Arnold proved using the Hilbert function that the verification part in the modular algorithm for computing Gröbner bases can be simplified for homogeneous ideals (cf. Arnold, 2003). The idea of the proof could easily be adapted to the local case, i.e. for local orderings and not necessarily homogeneous ideals, using the Hilbert–Samuel function (cf. Pfister, 2007). In this paper we prove the corresponding theorem for non-homogeneous ideals in the case of a global ordering.

1. Introduction

We consider an ideal in a polynomial ring over the rationals. In Section 2 we describe a parallel modular implementation of the Gröbner basis (resp. standard basis) algorithm. Afterwards we restrict ourselves to the case of a zero-dimensional ideal and introduce a parallel modular implementation of the algorithm for computing the associated primes in Section 3. Finally we give a couple of examples with corresponding timings and some conclusions in Section 4. Both algorithms are implemented in

☆ Part of the work was done at ASSMS, GCU Lahore – Pakistan.

E-mail addresses: nazeranjawwad@gmail.com (N. Idrees), pfister@mathematik.uni-kl.de (G. Pfister), steidel@mathematik.uni-kl.de (S. Steidel).
URLs: http://www.mathematik.uni-kl.de/~pfister (G. Pfister), http://www.mathematik.uni-kl.de/~steidel (S. Steidel).
1 Tel.: +49 631 205 3241; fax: +49 631 205 4795.

0747-7171/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsc.2011.01.003

SINGULAR. The Gröbner basis (resp. standard basis) algorithm can be found in the library `modstd.lib` and the algorithm for computing the associated primes in `assprimeszerodim.lib`. They are included in the release SINGULAR 3-1-2.

The task of computing a Gröbner basis G of an ideal I using modular methods consists of three steps. In the first step, we compute the Gröbner basis modulo p for sufficiently many primes p and, in the second step, use the Chinese remainder algorithm and rational reconstruction to obtain a result over $\mathbb{Q}$. In the third step, we have to verify that the result obtained in this way is correct, i.e. to verify that $I = \langle G\rangle$ and G is a Gröbner basis of $\langle G\rangle$. If this fails we go back to the first step. The third step is usually at least as time-consuming as the first step. Omitting the third step would produce a Gröbner basis only with high probability and the result could be wrong in extreme situations. It is known that some of the commercial computer algebra systems have problems in this direction.[2]

Arnold proved using the Hilbert function that the verification part in the modular algorithm for computing Gröbner bases can be simplified for homogeneous ideals (cf. Arnold, 2003): Let $I \subseteq \mathbb{Q}[x_1, \ldots, x_n]$ be a homogeneous ideal, $>$ a global monomial ordering and $G \subseteq \mathbb{Q}[x_1, \ldots, x_n]$ be a set of polynomials such that $I \subseteq \langle G\rangle$, G is a Gröbner basis of $\langle G\rangle$ and $\mathrm{LM}(G) = \mathrm{LM}(I\mathbb{F}_p[x_1, \ldots, x_n])$ for some prime number p where $\mathrm{LM}(G)$ denotes the set of leading monomials of G w.r.t. $>$; then G is a Gröbner basis of I. The idea of the proof could easily be adapted to the local case, i.e. for local orderings and I not necessarily homogeneous, using the Hilbert–Samuel function (cf. Pfister, 2007). In this paper we prove the corresponding theorem for non-homogeneous ideals in the case of a global ordering. Two important assumptions of the theorem are the facts that $I \subseteq \langle G\rangle$ and G is a Gröbner basis of $\langle G\rangle$. This verification can be very time-consuming in a negative case. Hence, we use a so-called PTESTSB which is one of the new ideas for our algorithm. Therefore we randomly choose a prime number p which has not been used in the previous computations and perform the verification modulo p. Only if PTESTSB is positive do we perform the verification over $\mathbb{Q}$, and the last required condition that $\mathrm{LM}(G) = \mathrm{LM}(I\mathbb{F}_p[x_1, \ldots, x_n])$ is then automatically fulfilled.

The implementation of our algorithm as in the SINGULAR library implies that we did not change the kernel routines of SINGULAR. We plan to implement the algorithm in the kernel of SINGULAR in the future. For this purpose we can apply the ideas of Gräbe (cf. Gräbe, 1993) – using multimodular coefficients – and Traverso (cf. Traverso, 1989) — using the trace algorithm. The trace algorithm would speed up the computations in positive characteristic a lot. We compute a Gröbner basis of an ideal $I \subseteq \mathbb{Q}[x_1, \ldots, x_n]$ over $\mathbb{F}_p[x_1, \ldots, x_n]$ for a random prime p and keep in mind the zero-reductions of the s-polynomials such that we do not perform these reductions in any other Gröbner basis computation over $\mathbb{F}_q[x_1, \ldots, x_n]$ for primes $q \neq p$. We do not need this information, i.e. the guarantee that we really obtain a Gröbner basis over $\mathbb{F}_q[x_1, \ldots, x_n]$, since we have the verification step – that the lifted result over $\mathbb{Q}[x_1, \ldots, x_n]$ is a Gröbner basis of I – at the end anyway.

Our idea regarding the primary decomposition of a zero-dimensional ideal $I \subseteq \mathbb{Q}[X]$ is to compute the associated primes $M_1, \ldots, M_s$ of I and use separators $\sigma_1, \ldots, \sigma_s$[3] such that the saturation $I : \sigma_i^\infty$ of I w.r.t. σ_i is the primary ideal corresponding to M_i (cf. Shimoyama and Yokoyama, 1996). The computation of the associated primes is based on the so-called shape lemma (Proposition 3.1(2)). Here, one new idea is to choose a generic linear form $r = a_1x_1 + \cdots + a_{n-1}x_{n-1} + x_n$ with $a_1, \ldots, a_{n-1} \in \mathbb{Z}$ and a random prime p to test whether $\dim_{\mathbb{F}_p}(\mathbb{F}_p[X]/I\mathbb{F}_p[X]) = \dim_{\mathbb{F}_p}\left(\mathbb{F}_p[x_n]/(\psi(I)\mathbb{F}_p[X] \cap \mathbb{F}_p[x_n])\right)$, i.e. $\psi(I)\mathbb{F}_p[X] = \langle x_1 - g_1(x_n), \ldots, x_{n-1} - g_{n-1}(x_n), F(x_n)\rangle$ where ψ denotes the linear map defined by $\psi(x_i) = x_i$ for $i = 1, \ldots, n-1$ and $\psi(x_n) = 2x_n - r$. If this test called PTESTRAD is positive, then the ideal I in $\mathbb{Q}[X]$ has the same property with high probability. If the test is negative then we compute the radical of I using the idea of Krick and Logar (Proposition 3.3(1)) combined with modular methods, and replace I by $\sqrt{I}$. Afterwards we compute $\langle F\rangle = \langle I, T - r\rangle_{\mathbb{Q}[X,T]} \cap \mathbb{Q}[T]$,

[2] Let N be the product of all primes smaller than 2^{32} and $I = \langle v + w + x + y + z, vw + wx + xy + yz + vz, vwx + wxy + xyz + vyz + vwz, vwxy + wxyz + vxyz + vwyz + vwxz, vwxyz + N\rangle \subseteq \mathbb{Q}[v, w, x, y, z]$. Then MAGMA V2.16–11 (64-bit version) computes an incorrect Gröbner basis; in particular it computes the Gröbner basis of the ideal $J = \langle v + w + x + y + z, vw + wx + xy + yz + vz, vwx + wxy + xyz + vyz + vwz, vwxy + wxyz + vxyz + vwyz + vwxz, vwxyz\rangle \subseteq \mathbb{Q}[v, w, x, y, z]$ which obviously differs from I.

[3] We call σ_i a *separator* w.r.t. M_i if $\sigma_i \notin M_i$ and $\sigma_i \in M_j$ for $j \neq i$.

again using modular methods, i.e. we compute $F^{(p)}$ such that $\langle F^{(p)}\rangle = \langle I, T - r\rangle_{\mathbb{F}_p[X,T]} \cap \mathbb{F}_p[T]$ and $\deg(F^{(p)}) = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$ for sufficiently many primes p, and we use the Chinese remainder algorithm and rational reconstruction to obtain $F \in \mathbb{Q}[T]$. The verification is the test of whether $F(r) \in I$ and no proper factor of $F(r)$ is in I. If $F = F_1^{\nu_1} \cdots F_s^{\nu_s}$ is the factorization of F in $\mathbb{Q}[T]$ into irreducible factors then $M_1 = \langle I, F_1(r)\rangle, \ldots, M_s = \langle I, F_s(r)\rangle$ are the associated primes of I. The new ideas in this approach are pTestRad described above and the fact that we do not compute the associated primes in positive characteristic but instead one special generator of the radical, $F(r)$, which is much easier to control.[4]

We use the following notation. Let $X = \{x_1, \ldots, x_n\}$ be a set of variables. We denote by $\mathrm{Mon}(X)$ the set of monomials, and by $\mathbb{Q}[X]$ the polynomial ring over $\mathbb{Q}$ in these n indeterminates. Let $S \subseteq \mathbb{Q}[X]$ be a set of polynomials; then $\mathrm{LM}(S) := \{\mathrm{LM}(f) \mid f \in S\}$ is the set of leading monomials of S. Given an ideal $I \subseteq \mathbb{Q}[X]$ we can always choose a finite set of polynomials F_I such that $I = \langle F_I\rangle$. If $I = \langle f_1, \ldots, f_r\rangle \subseteq \mathbb{Q}[X]$ and p is a prime number which does not divide any denominator of the coefficients of $f_1, \ldots, f_r$ we will write $I_p := \langle f_1 \bmod p, \ldots, f_r \bmod p\rangle \subseteq \mathbb{F}_p[X]$.

2. Computing Gröbner bases using modular methods

In the following we consider an ideal $I = \langle f_1, \ldots, f_r\rangle \subseteq \mathbb{Q}[X]$ together with a monomial ordering $>$ and set $F_I = \{f_1, \ldots, f_r\}$. We assume that $>$ is either global or local. Within this section we describe an algorithm for computing a Gröbner basis resp. a standard basis[5] $G \subseteq \mathbb{Q}[X]$ of I by using modular methods.

The basic idea of the algorithm is as follows. Choose a set P of prime numbers, compute standard bases G_p of $I_p \subseteq \mathbb{F}_p[X]$, for every $p \in P$, and finally lift these modular standard bases to a standard basis $G \subseteq \mathbb{Q}[X]$ of I. The lifting process consists of two steps. Firstly, the set $GP := \{G_p \mid p \in P\}$ is lifted to $G_N \subseteq \mathbb{Z}/N\mathbb{Z}[X]$ with $N := \prod_{p\in P} p$ by applying the Chinese remainder algorithm to the coefficients of the polynomials occurring in GP. Since G_N is uniquely determined modulo N, theory requires N to be larger than the moduli of all coefficients occurring in a standard basis of I over $\mathbb{Q}$. This issue is not resolvable a priori and will be discussed later in this section. Secondly, we obtain $G \subseteq \mathbb{Q}[X]$ by pulling back the modular coefficients occurring in G_N to rational coefficients via the Farey rational map.[6] This map is guaranteed to be bijective provided that $\sqrt{N/2}$ is larger than the moduli of all coefficients in G.[7] The latter condition on N concerning the Farey rational map obviously implies the former condition concerning the Chinese remainder algorithm. We consequently define two corresponding notions that are essential regarding the algorithm.

Definition 2.1. Let G be a standard basis of I.

(1) If G_p is a standard basis of I_p, then the prime number p is called *lucky for I* if and only if $\mathrm{LM}(G) = \mathrm{LM}(G_p)$. Otherwise p is called *unlucky for I*.
(2) A set P of lucky primes for I is called *sufficiently large for I* if and only if $\prod_{p\in P} p \geq \max\{2 \cdot |c|^2 \mid c \text{ coefficient occurring in } G\}$.

Now we can make the theoretical idea of the algorithm concrete. Consider a sufficiently large set P of lucky primes for I such that none of these primes divides any coefficient occurring in F_I, compute the set GP, and lift this result to a rational standard basis G of I as mentioned above. More details can be found in Arnold (2003).

---

4 The computation of the associated primes in positive characteristic would create similar problems to the factorization of polynomials: different behaviours of the splitting in different characteristics. Therefore it is easier and faster to compute $F \in \mathbb{Q}[T]$ and factorize this polynomial.

5 For definitions and properties, cf. Greuel and Pfister (2007).

6 *Farey fractions* refer to rational reconstruction. A definition of *Farey fractions* and the *Farey rational map* can be found in Arnold (2003), Kornerup and Gregory (1983) and Pfister (2007); for remarks concerning the computation, cf. Kornerup and Gregory (1983).

7 Remarks on the required bound on the coefficients are given in Kornerup and Gregory (1983).

In practice, we have to handle two difficulties, since naturally the standard basis G of I is a priori unknown. In fact, it is necessary to ensure that every prime number used is lucky for I, and to decide whether the chosen set of primes is sufficiently large for I.

Therefore, we fix a natural number s and an arbitrary set of primes P of cardinality s. After having computed the set of standard bases $GP := \{G_p \mid p \in P\}$ we delete the unlucky primes in the following way.

DELETEUNLUCKYPRIMESSB: *We define an equivalence relation on* (GP, P) *by* $(G_p, p) \sim (G_q, q) :\Longleftrightarrow \mathrm{LM}(G_p) = \mathrm{LM}(G_q)$. *Then the equivalence class of largest cardinality is stored in* (GP, P)*; the others are deleted.*

With the aid of this method we are able to choose a set of lucky primes with high probability. A faulty decision will be compensated by subsequent tests.

Since we cannot predict whether a given set of primes P is sufficiently large for I, we have to proceed by trial and error. Hence, we lift the set GP to $G \subseteq \mathbb{Q}[X]$, as per the description at the beginning of this section, and test whether G is already a standard basis of I. Otherwise we enlarge the set P by adding s new prime numbers and continue analogously until the test is positive. The test in particular verifies whether G is a standard basis of $\langle G\rangle$, but this computation in $\mathbb{Q}[X]$ can be very expensive if P is far away from being sufficiently large for I. Hence, we pre-fix a test in positive characteristic that is a sufficient criterion if P is not sufficiently large for I.

PTESTSB: *We randomly choose a prime number* $p \notin P$ *such that* p *does not divide the numerator and denominator of any coefficient occurring in* F_I. *The test is positive if and only if* $(G \bmod p)$ *is a standard basis of* I_p. *We explicitly test whether* $(f_i \bmod p) \in \langle G \bmod p\rangle$ *for* $i = 1, \ldots, r$ *and* $(G \bmod p) \subseteq \mathtt{std}(I_p)$.[8]

This test in positive characteristic accelerates the algorithm enormously. It is much faster than that in characteristic zero since the standard basis computation in PTESTSB is as expensive as in any other positive characteristic, i.e., as any other standard basis computation within the algorithm.

If the PTESTSB is negative, then P is not sufficiently large for I, that is, G cannot be a standard basis of I over $\mathbb{Q}$. Contrariwise, if the PTESTSB is positive, then G is most probably a standard basis of I.

Algorithm 1 shows the modular standard basis algorithm.[9]

Algorithm 1 MODSTD

Assume that $>$ is either a global or a local monomial ordering.

Input: $I \subseteq \mathbb{Q}[X]$.
Output: $G \subseteq \mathbb{Q}[X]$, the standard basis of I.

```
choose P, a list of random primes;
GP = ∅;
loop
  for p ∈ P do
    compute a standard basis G_p of I_p;
    GP = GP ∪ {G_p};
  (GP, P) = DELETEUNLUCKYPRIMESSB(GP, P);
  lift (GP, P) to G ⊆ Q[X] by applying the Chinese remainder algorithm and Farey rational map;
  if PTESTSB(I, G, P) then
    if I ⊆ ⟨G⟩ then
      if G is a standard basis of ⟨G⟩ then
        return G;
  enlarge P;
```

Remark 2.2. The version of the algorithm presented is just pseudo-code, whereas its implementation in SINGULAR is optimized. For example, the standard bases G_p of $I_p \subseteq \mathbb{F}_p[X]$ for $p \in P$ are not computed repeatedly, but stored and reused in further iteration steps.

[8] The procedure `std` is implemented in SINGULAR and computes a Gröbner basis resp. standard basis of the input.

[9] The corresponding procedures are implemented in SINGULAR in the library `modstd.lib`.

Remark 2.3. Algorithm 1 can easily be parallelized in the following way:

(1) Compute the standard bases G_p in parallel.
(2) Parallelize the final tests:
- Check whether $I \subseteq \langle G\rangle$ by checking whether $f \in \langle G\rangle$ for all $f \in F_I$.
- Check whether G is a standard basis of $\langle G\rangle$ by checking whether every s-polynomial not excluded by well-known criteria vanishes by reduction w.r.t. G.

Algorithm 1 terminates by construction, and its correctness is guaranteed by the following theorem which is proven in Arnold (2003) in the case where I is homogeneous, or in Pfister (2007) in the case where the ordering is local. The case where the ordering is global follows by using weighted homogenization as in Theorem 7.5.1 of Greuel and Pfister (2007).

Theorem 2.4. *Let $G \subseteq \mathbb{Q}[X]$ be a set of polynomials such that $\mathrm{LM}(G) = \mathrm{LM}(G_p)$ where G_p is a standard basis of I_p for some prime number p, G is a standard basis of $\langle G\rangle$ and $I \subseteq \langle G\rangle$. Then $I = \langle G\rangle$.*

Note that the first condition follows from a positive result of pTestSB whereas the second and third condition are verified explicitly at the end of the algorithm.

Proof of Theorem 2.4. We assume that $>$ is a global monomial ordering. The proof for a local ordering is similar. Let $F_I = \{f_1, \ldots, f_r\} \subseteq \mathbb{Q}[X]$ such that $I = \langle F_I\rangle$ and $G = \{g_1, \ldots, g_s\} \subseteq \mathbb{Q}[X]$. Since G is a standard basis of $\langle G\rangle$ w.r.t. $>$ and $I \subseteq \langle G\rangle$, there exist for each $i = 1, \ldots, r$ polynomials $\xi_{ij} \in \mathbb{Q}[X]$ such that

$$f_i = \sum_{j=1}^{s} \xi_{ij} g_j \quad \text{satisfying } \mathrm{LM}_>(f_i) \geq \mathrm{LM}_>(\xi_{ij} g_j) \text{ for all } j = 1, \ldots, s.$$

Due to Corollary 1.7.9 of Greuel and Pfister (2007) there exists a finite set $M \subseteq \mathrm{Mon}(X)$ with the following property: Let $>'$ be any monomial ordering on $\mathrm{Mon}(X)$ coinciding with $>$ on M; then $\mathrm{LM}_>(G) = \mathrm{LM}_{>'}(G)$ and G is also a standard basis of $\langle G\rangle$ w.r.t. $>'$.

Moreover, due to Lemma 1.2.11 resp. Exercise 1.7.17 of Greuel and Pfister (2007) we possibly enlarge the set M and choose some $w = (w_1, \ldots, w_n) \in \mathbb{Z}^n_{>0}$ such that $>=>_w$ on M, i.e. $\mathrm{LM}_>(G) = \mathrm{LM}_{>_w}(G)$ (resp. G is a standard basis of $\langle G\rangle$ w.r.t. $>_w$), and[10]

$$\begin{aligned}
&\text{w-deg}\left(\mathrm{LM}_{>_w}(f_i)\right) &&> \text{w-deg}\left(\mathrm{LM}_{>_w}(\mathrm{tail}(f_i))\right),\\
&\text{w-deg}\left(\mathrm{LM}_{>_w}(g_j)\right) &&> \text{w-deg}\left(\mathrm{LM}_{>_w}(\mathrm{tail}(g_j))\right),\\
&\text{w-deg}\left(\mathrm{LM}_{>_w}(\xi_{ij} g_j)\right) &&> \text{w-deg}\left(\mathrm{LM}_{>_w}(\mathrm{tail}(\xi_{ij} g_j))\right),
\end{aligned}$$

for all $i = 1, \ldots, r$ and $j = 1, \ldots, s$.

We consider on $\mathbb{Q}[X, t]$ the weighted degree ordering with weight vector $(w_1, \ldots, w_n, 1)$ refined by $>_w$ on $\mathbb{Q}[X]$ and denote it also by $>_w$. For $f \in \mathbb{Q}[X]$ let $f^h = t^{\text{w-deg}(f)} \cdot f(x_1/t^{w_1}, \ldots, x_n/t^{w_n})$ be the weighted homogenization of f w.r.t. t. We set $\overline{F}_I := \left\{f_1^h, \ldots, f_r^h\right\}$, $\overline{I} := \left\langle \overline{F}_I\right\rangle$ and $\overline{G} := \left\{g_1^h, \ldots, g_s^h\right\}$. Then Proposition 7.5.3 of Greuel and Pfister (2007) guarantees that $\overline{G}$ is a standard basis of $\left\langle \overline{G}\right\rangle$ and since $\mathrm{LM}_{>_w}(G) = \mathrm{LM}_{>_w}(G_p)$ it also holds by construction that $\mathrm{LM}_{>_w}(\overline{G}) = \mathrm{LM}_{>_w}(\overline{G}_p)$. Now let $i \in \{1, \ldots, r\}$; then $f_i = \sum_{j=1}^{s} \xi_{ij} g_j$, satisfying $\mathrm{LM}_{>_w}(f_i) \geq_w \mathrm{LM}_{>_w}(\xi_{ij} g_j)$ for all $j = 1, \ldots, s$. This implies $\text{w-deg}(f_i) \geq \text{w-deg}(\xi_{ij} g_j)$ for all $j = 1, \ldots, s$ by the choice of $w \in \mathbb{Z}^n_{>0}$. Consequently we have

$$t^{\text{w-deg}(f_i)} f\left(\frac{x_1}{t^{w_1}}, \ldots, \frac{x_n}{t^{w_n}}\right) = \sum_{j=1}^{s} t^{\text{w-deg}(f_i)} \xi_{ij}\left(\frac{x_1}{t^{w_1}}, \ldots, \frac{x_n}{t^{w_n}}\right) g_j\left(\frac{x_1}{t^{w_1}}, \ldots, \frac{x_n}{t^{w_n}}\right) \in \left\langle \overline{G}\right\rangle,$$

and thus $f_i^h \in \left\langle \overline{G}\right\rangle$ or $\overline{I} \subseteq \left\langle \overline{G}\right\rangle$, since $i \in \{1, \ldots, r\}$ was arbitrarily chosen.

[10] For a polynomial $f \in \mathbb{Q}[X]$, we define by $\mathrm{tail}(f) := f - \mathrm{LM}(f)$ the *tail* of f; cf. Greuel and Pfister (2007).

It remains to prove that $\bar{I} = \langle \bar{G} \rangle$. Let $n \in \mathbb{N}$. We know that $\bar{I}_p = \langle \bar{G}_p \rangle$ due to the fact that $\mathrm{LM}_{>_w}(\bar{G}) = \mathrm{LM}_{>_w}(\bar{G}_p)$, so in particular it holds that $\mathrm{HF}_{\bar{I}_p}(n) = \mathrm{HF}_{\langle \bar{G}_p \rangle}(n) = \mathrm{HF}_{\langle \bar{G} \rangle}(n)$ for the corresponding Hilbert functions. On the other hand we have

$$\mathrm{HF}_{\bar{I}}(n) \leq \mathrm{HF}_{\bar{I}_p} = \mathrm{HF}_{\langle \bar{G} \rangle}(n) \leq \mathrm{HF}_{\bar{I}}(n) < \infty,$$

where the second inequality is true since $\bar{I} \subseteq \langle \bar{G} \rangle$. The first inequality follows from the fact that $\dim_{\mathbb{Q}}(\bar{I}[n]) \geq \dim_{\mathbb{F}_p}(\bar{I}_p[n])$, where $\bar{I}[n]$ resp. $\bar{I}_p[n]$ denotes the vector space generated by all (weighted) homogeneous polynomials of degree n. Namely we can find a $\mathbb{Q}$-basis of $\bar{I}[n]$ of polynomials in $\mathbb{Z}[X, t] \cap \bar{I}$ which induces generators of $\bar{I}_p[n]$. □

Remark 2.5. Algorithm 1 is also applicable without applying the final tests, i.e. skipping the verification that $I \subseteq \langle G \rangle$ and G is a standard basis of $\langle G \rangle$. In this case the algorithm is probabilistic, i.e. the output G is a standard basis of the input I, only with high probability. This usually accelerates the algorithm enormously. Note that the probabilistic algorithm works for any ordering, i.e. also for the so-called mixed ordering. In the case of a mixed ordering one could homogenize the ideal I, compute a standard basis using modStd and dehomogenize afterwards. Experiments showed that this is usually not efficient since the standard basis of the homogenized input often has many more elements than the standard basis of the ideal that we started with.

3. A modular approach to primary decomposition

In the following let $I \subseteq \mathbb{Q}[X]$ be a zero-dimensional ideal and $d := \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$. Within this section we describe an algorithm for computing the associated primes of I using modular methods. In conclusion, we make remarks on how to achieve the corresponding primary ideals starting from the associated primes of I.

The following well-known proposition (cf. Gianni et al., 1988 or Greuel and Pfister, 2007) describes how to compute the associated prime ideals of a radical ideal over $\mathbb{Q}$. Note that these results are also valid for perfect infinite fields.

Proposition 3.1. *Let $I \subseteq \mathbb{Q}[X]$ be a radical ideal.*

(1) *Let $\langle F \rangle = I \cap \mathbb{Q}[x_n]$ and assume $\deg(F) = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$. Let $F = F_1 \cdots F_s$ be the factorization of F into irreducible factors over $\mathbb{Q}$. Then $I = \bigcap_{i=1}^{s} \langle I, F_i \rangle$ and $\langle I, F_i \rangle$ is prime for $i = 1, \ldots, s$.*

(2) *There exists a non-empty Zariski open subset $U \subseteq \mathbb{Q}^{n-1}$ such that for all $a = (a_1, \ldots, a_{n-1}) \in U$ the linear coordinate change φ_a defined by $\varphi_a(x_i) = x_i$ for $i = 1, \ldots, n-1$ and $\varphi_a(x_n) = x_n + \sum_{i=1}^{n-1} a_i x_i$ satisfies*

$$\dim_{\mathbb{Q}}\left(\mathbb{Q}[X]/\varphi_a(I)\right) = \dim_{\mathbb{Q}}\left(\mathbb{Q}[x_n]/(\varphi_a(I) \cap \mathbb{Q}[x_n])\right).$$

Corollary 3.2. *Let $F \in \mathbb{Q}[T]$, T a variable, be square-free and $r = x_n + \sum_{i=1}^{n-1} a_i x_i$ with $a_1, \ldots, a_{n-1} \in \mathbb{Z}$ such that $\deg(F) = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$, and $F(r) \in I$ but no proper factor of $F(r)$ is in I; then I is a radical ideal. Let $F = F_1 \cdots F_s$ be the factorization of F into irreducible factors over $\mathbb{Q}$. Then $I = \bigcap_{i=1}^{s} \langle I, F_i(r) \rangle$ and $\langle I, F_i(r) \rangle$ is prime for $i = 1, \ldots, s$.*

Proof. Using a linear change of variables we may assume that $r = x_n$. Since no proper factor of $F(r)$ is in I we obtain $\langle F(x_n) \rangle = I \cap \mathbb{Q}[x_n]$. Since $\deg(F) = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$ we have $I = \langle x_1 - h_1(x_n), \ldots, x_{n-1} - h_{n-1}(x_n), F(x_n) \rangle$ for suitable $h_1, \ldots, h_{n-1} \in \mathbb{Q}[x_n]$. Thus, I is radical because F is square-free. The rest is an immediate consequence of Proposition 3.1(1). □

Consequently, for the computation of the primary decomposition, we first have to establish whether I is already radical. Therefore we choose a generic linear form $r = a_1 x_1 + \cdots + a_{n-1} x_{n-1} + x_n$ with $a_1, \ldots, a_{n-1} \in \mathbb{Z}$, and use a test in positive characteristic, similarly to Section 2.

pTestRad: *We randomly choose a prime number p such that $\dim_{\mathbb{F}_p}(\mathbb{F}_p[X]/I_p) = d$. Let $\varphi : \mathbb{F}_p[T] \longrightarrow \mathbb{F}_p[X]$ be defined by $\varphi(T) = r \bmod p$ (cf. Lemma 3.6(1)) and $\langle F_p \rangle := \varphi^{-1}(I_p)$. We test whether $\deg(F_p) = d$.*

In the case of a negative result of the test there is a high probability that the ideal is not radical (cf. Proposition 3.1(2)) and we compute the radical using modular methods. The computation of the radical is usually much more time-consuming than that of pTestRad even if the ideal is already radical. The following proposition (cf. Krick and Logar, 1991; Greuel and Pfister, 2007) is the basis for computing the radical of a zero-dimensional ideal.

Proposition 3.3. *Let $I \subseteq \mathbb{Q}[X]$ be a zero-dimensional ideal and $\langle f_i\rangle = I \cap \mathbb{Q}[x_i]$ for $i = 1, \ldots, n$. Moreover, let g_i be the square-free part of f_i. Then the following hold.*

(1) $\sqrt{I} = I + \langle g_1, \ldots, g_n\rangle$.
(2) *If* $\deg(f_n) = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$ *then* $\sqrt{I} = \langle I, g_n\rangle$.

Proof. Part (1) of the proposition is proved in Krick and Logar (1991). For part (2) we notice that if $\deg(f_n) = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$ then there exist $h_1, \ldots, h_{n-1} \in \mathbb{Q}[x_n]$ such that $\{x_1 - h_1, \ldots, x_{n-1} - h_{n-1}, f_n\}$ is a Gröbner basis of I w.r.t. the lexicographical ordering $x_1 > \cdots > x_n$. Thus, we have $\sqrt{I} = \langle x_1 - h_1, \ldots, x_{n-1} - h_{n-1}, g_n\rangle$. □

With considerations analogous to those in Section 2, the essential idea of the algorithm for computing the radical of I is as follows. Choose a set P of prime numbers, compute, for every $p \in P$, monic polynomials $f_1^{(p)}, \ldots, f_n^{(p)}$ satisfying $\langle f_i^{(p)}\rangle = I_p \cap \mathbb{F}_p[x_i]$ for $i = 1, \ldots, n$ and finally lift these polynomials via the Chinese remainder algorithm and Farey rational map to $(f_1, \ldots, f_n) \in \mathbb{Q}[x_1] \times \cdots \times \mathbb{Q}[x_n]$.

Definition 3.4. Let $(f_1, \ldots, f_n) \in \mathbb{Q}[x_1] \times \ldots \times \mathbb{Q}[x_n]$ satisfy $\langle f_i\rangle = I \cap \mathbb{Q}[x_i]$ for $i = 1, \ldots, n$.[11]

(1) If $(f_1^{(p)}, \ldots, f_n^{(p)}) \in \mathbb{F}_p[x_1] \times \ldots \times \mathbb{F}_p[x_n]$ satisfies $\langle f_i^{(p)}\rangle = I_p \cap \mathbb{F}_p[x_i]$ for $i = 1, \ldots, n$, then the prime number p is called *lucky for I* if and only if $\deg(f_i) = \deg(f_i^{(p)})$ for $i = 1, \ldots, n$. Otherwise p is called *unlucky for I*.
(2) A set P of lucky primes for I is called *sufficiently large for I* if and only if $\prod_{p\in P} p \geq \max\{2 \cdot |c|^2 \mid c$ coefficient occurring in $f_1, \ldots, f_n\}$.

After having computed the set $FP := \{(f_1^{(p)}, \ldots, f_n^{(p)}) \mid p \in P\}$ we delete the unlucky primes in the following way.

deleteUnluckyPrimesRad: *We define an equivalence relation on (FP, P) by $(F^{(p)}, p) \sim (F^{(q)}, q) :\Longleftrightarrow \deg(f_i^{(p)}) = \deg(f_i^{(q)})$ for $i = 1, \ldots, n$. Then the equivalence class of largest cardinality is stored in (FP, P); the others are deleted.*

With the aid of this method we are able to choose a set of lucky primes with high probability. A faulty decision will be compensated by the subsequent test of whether $f_i \in I$ for $i = 1, \ldots, n$.

Since we cannot predict whether a given set of primes P is sufficiently large for I, we have to proceed by trial and error, as already described in Section 2.

Algorithm 2 computes the radical of I.[12]

If the pTestRad is positive then, with high probability, after a generic coordinate change, it holds that $\dim_{\mathbb{Q}}(\mathbb{Q}[x_n]/(I \cap \mathbb{Q}[x_n])) = d$. In this case it is not necessary to compute the radical of I and we rely on the following corollary.

Corollary 3.5. *Let $I \subseteq \mathbb{Q}[X]$ be a zero-dimensional ideal and $r = x_n + \sum_{i=1}^{n-1} a_i x_i$ with $a_1, \ldots, a_{n-1} \in \mathbb{Z}$. Let $F \in \mathbb{Q}[T]$, T a variable, such that $\deg(F) = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$ and $F(r) \in I$ but no proper factor of $F(r)$ is in I. Moreover, let H be the square-free part of F. Then $\sqrt{I} = \langle I, H(r)\rangle$.*

Proof. The proof is a consequence of Proposition 3.3(2) and Corollary 3.2. □

Consequently we need to obtain a polynomial $F \in \mathbb{Q}[T]$ satisfying the required properties of Corollary 3.2 or Corollary 3.5. The following lemma is helpful in this direction.

[11] By abuse of the notation we use the same terminology as in Definition 2.1 since it is always clear from the context which definition we are referring to.
[12] The corresponding procedure is implemented in Singular in the library `assprimeszerodim.lib`.

Algorithm 2 zeroRadical

Input: $I = \langle G_I \rangle \subseteq \mathbb{Q}[X]$, a zero-dimensional ideal generated by a Gröbner basis G_I w.r.t. some global ordering.

Output: $G \subseteq \mathbb{Q}[X]$, a Gröbner basis of the radical of I w.r.t. a degree ordering.

```
choose P, a list of random primes;
FP = ∅;
loop
  for p ∈ P do
    compute monic polynomials f_i^(p) such that ⟨f_i^(p)⟩ = I_p ∩ F_p[x_i] for i = 1, ..., n;
    FP = FP ∪ {(f_1^(p), ..., f_n^(p))};
  (FP, P) = deleteUnluckyPrimesRad(FP, P);
  lift (FP, P) to (f_1, ..., f_n) ∈ Q[x_1] × ... × Q[x_n] by applying the Chinese remainder algorithm and
  Farey rational map;
  use G_I to test if f_i ∈ I for i = 1, ..., n;
  if f_i ∈ I for all i = 1, ..., n then
    exit loop;
  enlarge P;
for i = 1, ..., n do
  compute g_i, the square-free part of f_i;
I = I + ⟨g_1, ..., g_n⟩;
compute G ⊆ Z[X], a Q[X]-Gröbner basis of I w.r.t. a degree ordering;[13]
return G;
```

Lemma 3.6. *Let K be a field,*[14] *and $F \in K[T]$, T a variable, be monic and square-free, and let $r = x_n + \sum_{i=1}^{n-1} a_i x_i$, $a_1, \ldots, a_{n-1} \in K$, such that $\deg(F) = \dim_K(K[X]/I)$ and $F(r) \in I$ but no proper factor of $F(r)$ is in I.*

(1) *Let $\varphi : K[T] \to K[X]$ be defined by $\varphi(T) = r$. Then $\varphi^{-1}(I) = \langle F \rangle$.*
(2) *Let $\psi : K[X] \to K[X]$ be defined by $\psi(x_i) = x_i$ for $i = 1, \ldots, n-1$ and $\psi(x_n) = 2x_n - r$. Then $\psi(I) \cap K[x_n] = \langle F(x_n) \rangle$.*
(3) *Let $\lambda : K[X]/I \to K[X]/I$ be the map defined by the multiplication with r, $\lambda(g + I) = r \cdot g + I$. Then F is the characteristic polynomial of λ.*

Proof. (1) Since $\varphi(F) = F(r) \in I$ we obtain $F \in \varphi^{-1}(I)$. Thus we have $\langle F \rangle = \varphi^{-1}(I)$ because no proper factor of $F(r)$ is in I.
(2) It holds that $F(x_n) = \psi(F(r)) \in \psi(I)$ by definition of ψ. The assumption implies that no proper factor of $F(x_n)$ is in $\psi(I)$, i.e. $\langle F(x_n) \rangle = \psi(I) \cap K[x_n]$.
(3) Using the map ψ of (2) we may assume that $r = x_n$. As in the proof of Corollary 3.2 we obtain $I = \langle x_1 - h_1, \ldots, x_{n-1} - h_{n-1}, F(x_n) \rangle$ for suitable $h_1, \ldots, h_{n-1} \in K[x_n]$ since $\deg(F) = \dim_K(K[X]/I) = d$. Hence, we may choose $\{1, x_n, \ldots, x_n^{d-1}\}$ as a basis of $K[X]/I \cong K[x_n]/\langle F(x_n) \rangle$, and obtain the polynomial F to be the characteristic polynomial of the multiplication with x_n. □

Lemma 3.6 shows that the approach of Eisenbud, Hunecke, and Vasconcelos (cf. Eisenbud et al., 1992) using (1) of the lemma, the approach of Gianni, Trager, and Zacharias (cf. Gianni et al., 1988) using (2) of the lemma and the approach of Monico (cf. Monico, 2002) using (3) of the remark are in principle the same. The computations for (1) or (2) require Gröbner bases with respect to suitable block orderings, whereas in (3) we do not need a special ordering for the Gröbner basis but we have to compute a determinant. All three algorithms are implemented in Singular.

Remark 3.7. We can also compute the polynomial $F \in \mathbb{Q}[T]$ using modular methods. For this purpose we compute $F^{(p)} \in \mathbb{F}_p[T]$, monic such that $\langle F^{(p)} \rangle = \ker(\varphi_p)$, where $\varphi_p : \mathbb{F}_p[T] \longrightarrow \mathbb{F}_p[X]/I_p$,

[13] Here we use the procedure modStd as described in Section 2.
[14] We substitute $\mathbb{Q}$ by an arbitrary field K since we also need the results of Lemma 3.6 for finite fields.

$\varphi_p(T) = r \bmod I_p$, for several prime numbers p, and preserve just those $F^{(p)}$ with $\deg(F^{(p)}) = d$. Afterwards we lift the results to $F \in \mathbb{Q}[T]$ by applying the Chinese remainder algorithm and Farey rational map.

Remark 3.8. If $K = \mathbb{C}$ is the field of complex numbers we can use the polynomial F of Corollary 3.2 to compute the zeros of the ideal I. The zeros of F are the eigenvalues of the multiplication map λ defined in Lemma 3.6. Let $\lambda_1, \ldots, \lambda_d$ be the (different) eigenvalues of λ; then $I = \bigcap_{i=1}^{d} \langle I, r - \lambda_i \rangle$. Moreover, $\langle I, r - \lambda_i \rangle$ is a maximal ideal in $\mathbb{C}[X]$ representing a zero of I for $i = 1, \ldots, d$.

Referring to Proposition 3.1, Corollary 3.2 and the above considerations, Algorithm 3 computes the associated primes of I.[15]

Algorithm 3 ASSPRIMES

Input: $I \subseteq \mathbb{Q}[X]$, a zero-dimensional ideal.
Output: $L = \{M_1, \ldots, M_s\}$, M_i prime and $\sqrt{I} = \bigcap_{i=1}^{s} M_i$.

compute $G \subseteq \mathbb{Z}[X]$, a $\mathbb{Q}[X]$-Gröbner basis of I w.r.t. a degree ordering;[16]
compute $d = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/I)$ using G;
choose $a_1, \ldots, a_{n-1} \in \mathbb{Z}$ randomly, and $r = a_1 x_1 + \ldots + a_{n-1} x_{n-1} + x_n$;
if not PTESTRAD(d, r, G) **then**
  $G =$ ZERORADICAL(G);
  $d = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/\langle G \rangle)$;
choose P, a list of random primes;
$FP = \emptyset$;
$l = 0$;
loop
  for $p \in P$ **do**
    compute $F^{(p)} \in \mathbb{F}_p[T]$, monic such that $\langle F^{(p)} \rangle = \ker(\varphi_p)$, whereas $\varphi_p : \mathbb{F}_p[T] \longrightarrow \mathbb{F}_p[X]/I_p$, $\varphi_p(T) = r \bmod I_p$;[17]
    if $\deg(F^{(p)}) = d$ **then**
      $FP = FP \cup \{F^{(p)}\}$;
  if $\#(FP) = l$ **then**
    $G =$ ZERORADICAL(G);
    $d = \dim_{\mathbb{Q}}(\mathbb{Q}[X]/\langle G \rangle)$;
    choose $a_1, \ldots, a_{n-1} \in \mathbb{Z}$ randomly, and $r = a_1 x_1 + \ldots + a_{n-1} x_{n-1} + x_n$;
  else
    lift (FP, P) to $F \in \mathbb{Q}[T]$ by applying the Chinese remainder algorithm and Farey rational map;
    factorize $F = F_1^{\nu_1} \cdots F_s^{\nu_s}$ with $F_1, \ldots, F_s$ irreducible;
    compute $F(r)$ and $F_1(r), \ldots, F_s(r)$;
    if $F(r) \in I$ **then**
      if no proper factor of $F(r)$ is in I **then**
        return $\{\langle I, F_1(r) \rangle, \ldots, \langle I, F_s(r) \rangle\}$;
      else
        choose a non-trivial factor H of F of minimal degree such that $H(r) \in I$;
        let $F_{i_1}, \ldots, F_{i_t}$ correspond to H;
        return ASSPRIMES$(\langle I, F_{i_1}(r) \rangle) \cup \cdots \cup$ ASSPRIMES$(\langle I, F_{i_t}(r) \rangle)$;
    enlarge P;
    $l = \#(FP)$;

[15] The corresponding procedures are implemented in SINGULAR in the library `assprimeszerodim.lib`.
[16] Here we use the procedure MODSTD as described in Section 2.
[17] All approaches mentioned in Lemma 3.6 are applicable to verify this step.

Remark 3.9. The versions of Algorithms 2 and 3 presented are just pseudo-code whereas their implementation in SINGULAR is optimized. For example, the polynomials $f_i^{(p)} \in \mathbb{F}_p[x_i]$ and $F^{(p)} \in \mathbb{F}_p[T]$ for $p \in P$ are not computed repeatedly, but stored and reused in further iteration steps.

Remark 3.10. Algorithm 2 resp. Algorithm 3 can easily be parallelized by computing the polynomials $f_i^{(p)} \in \mathbb{F}_p[x_i]$ resp. $F^{(p)} \in \mathbb{F}_p[T]$ in parallel. Experiments indicate that the difficult and time-consuming part of Algorithm 3 is the test of whether $F(r) \in I$ and the computation of $F_1(r), \ldots, F_s(r)$. These $s+1$ computations are independent from each other, such that they can also be verified separately in parallel.

Following the idea of one of the referees, we tried to avoid the computation of $F(r)$ by computing a $\mathbb{Q}[X,T]$-Gröbner basis of $\langle I, T-r\rangle$ w.r.t. an elimination ordering (eliminating X) by using modular methods (cf. Section 2) and the FGLM algorithm (cf. Faugère et al., 1993). In this case we directly compute $\langle I, T-r\rangle_{\mathbb{Q}[X,T]} \cap \mathbb{Q}[T] = \langle F\rangle$ and may consequently omit the verification. Experiments showed that this is as time-consuming as the method presented in Algorithm 3.

Remark 3.11. Knowing the associated primes, it is easy to compute the primary ideals using the method of Shimoyama and Yokoyama (cf. Shimoyama and Yokoyama, 1996): Let $M_1, \ldots, M_s$ be the associated primes of the zero-dimensional ideal I and $\sigma_1, \ldots, \sigma_s$ a system of separators, i.e. $\sigma_i \notin M_i$ and $\sigma_i \in M_j$ for $j \neq i$; then the saturation of I w.r.t. σ_i is the primary ideal corresponding to M_i. Each σ_i can be chosen as $\prod_{j\neq i} m_j$ where m_j is an element of a Gröbner basis of M_j which is not in M_i. The saturation can be computed modularly, like with MODSTD, and in parallel.

4. Examples, timings and conclusion

In this section we provide examples on which we time the algorithms `modStd` (cf. Section 2) resp. `assPrimes` (cf. Section 3) and their parallelizations as opposed to the usual algorithms `std` resp. `minAssGTZ`[18] implemented in SINGULAR. Timings are conducted by using the 32-bit version of SINGULAR 3-1-2 on an AMD Opteron 6174 with 48 CPUs, 800 MHz each, 128 GB RAM under the Gentoo Linux operating system. All examples are chosen from The SymbolicData Project (cf. Gräbe, 2010).

Remark 4.1. The parallelization of our modular algorithms is attained via multiple processes organized by SINGULAR library code. Consequently a future aim is to enable parallelization in the kernel via multiple threads.

We choose the following examples to emphasize the superiority of modular standard basis computation and especially its parallelization:

Example 4.2. Characteristic: 0, ordering: `dp`,[19] `Cyclic_8.xml` (cf. Björck and Fröberg, 1991).

Example 4.3. Characteristic: 0, ordering: `dp`, `Paris.ilias13.xml` (cf. Kotsireas and Lazard, 1999).

Example 4.4. Characteristic: 0, ordering: `dp`, homog. `Cyclic_7.xml` (cf. Björck and Fröberg, 1991).

Example 4.5. Characteristic: 0, ordering: `ds`,[20] `Steidel_1.xml` (cf. Pfister, 2007).

Table 1 summarizes the results where `modStd`$^*(n)$ denotes the parallelized version of the algorithm applied on n cores. In all tables, the symbol "-" indicates out of memory failures. All timings are given in seconds.

The basic algorithm `std` runs out of memory for Examples 4.2 and 4.5. As mentioned in Section 2, it is possible to parallelize the computation in several parts of the algorithm `modStd`. In many cases

---

[18] The procedure `minAssGTZ` is implemented in SINGULAR in the library `primdec.lib` and computes the minimal associated prime ideals of the input.

[19] *Degree reverse lexicographical ordering:* Let $X^\alpha, X^\beta \in \mathrm{Mon}(X)$. $X^\alpha >_{dp} X^\beta :\Longleftrightarrow \deg(X^\alpha) > \deg(X^\beta)$ or $(\deg(X^\alpha) = \deg(X^\beta)$ and $\exists\, 1 \leq i \leq n : \alpha_n = \beta_n, \ldots, \alpha_{i-1} = \beta_{i-1}, \alpha_i < \beta_i)$, where $\deg(X^\alpha) = \alpha_1 + \cdots + \alpha_n$; cf. Greuel and Pfister (2007).

[20] *Negative degree reverse lexicographical ordering:* Let $X^\alpha, X^\beta \in \mathrm{Mon}(X)$. $X^\alpha >_{ds} X^\beta :\Longleftrightarrow \deg(X^\alpha) < \deg(X^\beta)$ or $(\deg(X^\alpha) = \deg(X^\beta)$ and $\exists\, 1 \leq i \leq n : \alpha_n = \beta_n, \ldots, \alpha_{i-1} = \beta_{i-1}, \alpha_i < \beta_i)$, where $\deg(X^\alpha) = \alpha_1 + \ldots + \alpha_n$; cf. Greuel and Pfister (2007).

Table 1
Total running times for computing a standard basis of the examples considered via `std`, and `modStd` and its parallelized variant `modStd`$^*(n)$ for $n = 4, 9, 30$.

Example	`std`	`modStd`	`modStd`$^*(4)$	`modStd`$^*(9)$	`modStd`$^*(30)$
4.2	–	8271	4120	2927	1138
4.3	37734	1159	676	580	380
4.4	3343	3436	886	408	113
4.5	–	6	3	3	3

Table 2
Running times for `modStd` and `modStd`$^*(n)$ with $n = 4, 9, 30$ without a verification test.

Example	$\texttt{modStd}_{\text{w/o ver.}}$	$\texttt{modStd}^*_{\text{w/o ver.}}(4)$	$\texttt{modStd}^*_{\text{w/o ver.}}(9)$	$\texttt{modStd}^*_{\text{w/o ver.}}(30)$
4.2	7929	3751	2698	920
4.3	941	614	552	370
4.4	52	38	31	36
4.5	6	3	3	3

Table 3
Total running times for computing the associated prime ideals of the examples considered via `minAssGTZ`, and `assPrimes` and its parallelized variant `assPrimes`$^*(n)$ for $n = 4, 9$.

Example	`minAssGTZ`	`assPrimes`			`assPrimes`$^*(4)$			`assPrimes`$^*(9)$		
		(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
4.6	–	1	1	0	1	1	1	1	1	1
4.7	–	169	169	188	104	98	104	95	100	105
4.8	–	129	131	230	90	87	114	76	77	103
4.9	189	4	5	5	10	8	8	8	8	8
4.10	589	35	35	35	24	23	19	25	24	25

it turns out that the final test – the verification of whether the lifted set of polynomials includes the input and is itself a standard basis; see also Remark 2.5 – is a time-consuming part. Therefore we extract the timings for the computation without the verification test in Table 2, again in seconds.

We consider the following examples for the computation of the associated prime ideals of a given zero-dimensional ideal:

Example 4.6. Characteristic: 0, ordering: `dp`, `Becker-Niermann.xml` (cf. Decker et al., 1998).

Example 4.7. Characteristic: 0, ordering: `dp`, `FourBodyProblem.xml` (cf. Bini and Mourrain, 2010).

Example 4.8. Characteristic: 0, ordering: `dp`, `Reimer_5.xml` (cf. Bini and Mourrain, 2010).

Example 4.9. Characteristic: 0, ordering: `lp`,[21] `ZeroDim.example_12.xml` (cf. Gräbe, 2010).

Example 4.10. Characteristic: 0, ordering: `dp`, `Cassou_1.xml` (cf. Bini and Mourrain, 2010).

Using modular methods via the algorithm `assPrimes` we apply all three variants mentioned in Section 3:

(1) the approach of Eisenbud, Hunecke, and Vasconcelos (cf. Eisenbud et al., 1992),
(2) the approach of Gianni, Trager, and Zacharias (cf. Gianni et al., 1988),
(3) the approach of Monico (cf. Monico, 2002).

We summarize the results of the timings in Tables 3 and 4 where `assPrimes`$^*(n)$ denotes the parallelized version of the algorithm applied on n cores.

[21] *Lexicographical ordering:* Let $X^\alpha, X^\beta \in \mathrm{Mon}(X)$. $X^\alpha >_{lp} X^\beta :\Longleftrightarrow \exists\, 1 \le i \le n :\ \alpha_1 = \beta_1, \ldots, \alpha_{i-1} = \beta_{i-1}, \alpha_i > \beta_i$; cf. Greuel and Pfister (2007).

Table 4
Running times for `assPrimes` and `assPrimes*`(n) with $n = 4, 9$ without a final verification step.

Example	`assPrimes`$_{\text{w/o ver.}}$			`assPrimes`$^*_{\text{w/o ver.}}(4)$			`assPrimes`$^*_{\text{w/o ver.}}(9)$		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
4.6	1	1	0	1	0	0	1	1	1
4.7	15	14	34	7	7	13	5	5	15
4.8	41	37	139	39	38	64	30	26	55
4.9	4	5	5	9	8	8	8	8	8
4.10	7	6	7	5	5	5	5	4	6

The usual algorithm `minAssGTZ` runs out of memory for Examples 4.6–4.8. Like for the modular standard basis algorithm, we also list the timings needed for `assPrimes` and `assPrimes`*(n) without the final verification step – the check of whether $F(r) \in I$ and the computation of $F_1(r), \ldots, F_s(r)$; see also Remark 3.10 – in Table 4.

Final Remarks and Conclusion 4.11. (1) For the computation of Gröbner bases (resp. standard bases) of ideals $I \subseteq \mathbb{Q}[X]$ w.r.t. global (resp. local) orderings, MODSTD should be used. This is usually faster even without parallel computing.

(2) The probabilistic algorithm for computing standard bases works without any restriction on the ordering. It is much faster than the deterministic one. It can be used to obtain ideas in algebraic geometry and other fields by computing several examples, like in computations in positive characteristic 20 years ago when computations of standard bases in characteristic zero were impossible or too slow.

(3) A kernel implementation of MODSTD could speed up the modular part using the trace algorithm of Traverso (cf. Traverso, 1989).

(4) We also implemented a variant for modular computing of Gröbner bases using p-adic lifting of the coefficients of a Gröbner basis modulo p for a random prime p. This requires us to compute the so-called conversion matrix modulo p, i.e. a matrix which represents the Gröbner basis in terms of the generators of the input ideal. It turned out that this is more expensive than using several primes and the Chinese remainder algorithm combined with rational reconstruction. In addition, we tested the idea of Gräbe (cf. Gräbe, 1993) for computing the conversion matrix and a syzygy matrix for the input, replacing the Chinese remainder algorithm and rational reconstruction by a system of linear equations with coefficients in $\mathbb{Q}$ whose solution gives the unique lifting of the Gröbner basis. This turned out to be slower than our approach because the computation of the conversion matrix takes more time than using the Chinese remainder algorithm and rational reconstruction.

(5) An increasing number of cores used during the parallel computation of standard bases or associated primes speeds up the computation if the corresponding problem in positive characteristic takes some time to compute. If the computations in positive characteristic are fast then an increasing number of cores may slow down the computations because of too much overhead.

(6) In the current implementation the Chinese remainder algorithm and Farey fractions are not parallelized. Experiments (e.g. the computation of the Gröbner basis of `Cyclic_9`) show that the computations in positive characteristic need different amounts of time on different cores. Therefore one should apply the Chinese remainder algorithm and Farey fractions already to partial results. This could save about 3% computing time.

(7) For zero-dimensional primary decomposition the modular approach is very efficient. This should be extended to higher-dimensional ideals.

Acknowledgements

The authors would like to thank Wolfram Decker and Gert-Martin Greuel for helpful discussions as regards proving Theorem 2.4. We also thank Christian Eder for important hints and discussions concerning the implementation of the algorithms described. In addition, we thank Frank Seelisch for

revealing an observation about MAGMA V2.16–11 specified in Section 1. Finally, we would like to thank Daniel Lichtblau and the anonymous referees whose comments led to great improvement of the paper.

References

Arnold, E.A., 2003. Modular algorithms for computing Gröbner bases. Journal of Symbolic Computation 35, 403–419.
Björck, G., Fröberg, G., 1991. A faster way to count the solution of inhomogeneous systems of algebraic equations, with applications to cyclic n-roots. Journal of Symbolic Computation 12, 329–336.
Bini, D., Mourrain, B., Polynomial test suite. Frisco project (LTR 21.024). http://www-sop.inria.fr/saga/POL/ (2010).
Becker, E., Wörmann, T., 1996. Radical computations of zero-dimensional ideals and real root counting. Mathematics and Computers in Simulation 42, 561–569.
Dickenstein, A., Emiris, I.Z., 2005. Solving Polynomial Equations. In: Algorithms and Computation in Mathematics, vol. 14. Springer.
Decker, W., Greuel, G.-M., Pfister, G., 1998. Primary Decomposition: Algorithms and Comparisons. In: Algorithmic Algebra and Number Theory, Springer, pp. 187–220.
Decker, W., Greuel, G.-M., Pfister, G., Schönemann, H., Singular 3-1-1 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de (2010).
Ebert, G.L., 1983. Some comments on the modular approach to Gröbner bases. ACM SIGSAM Bulletin 17, 28–32.
Eisenbud, D., Huneke, C., Vasconcelos, W., 1992. Direct methods for primary decomposition. Inventiones Mathematicae 110, 207–235.
Faugère, J.C., Gianni, P., Lazard, D., Mora, T., 1993. Efficient computation of zero-dimensional Gröbner bases by change of ordering. Journal of Symbolic Computation 16, 329–344.
Gräbe, H.-G., 1993. On lucky primes. Journal of Symbolic Computation 15, 199–209.
Gräbe, H.-G., The SymbolicData Project — Tools and Data for Testing Computer Algebra Software. http://www.symbolicdata.org (2010).
Greuel, G.-M., Pfister, G., 2007. A SINGULAR Introduction to Commutative Algebra, Second edition. Springer.
Gianni, P., Trager, B., Zacharias, G., 1988. Gröbner bases and primary decomposition of polynomial ideals. Journal of Symbolic Computation 6, 149–167.
Kornerup, P., Gregory, R.T., 1983. Mapping integers and Hensel codes onto Farey fractions. BIT Numerical Mathematics 23 (1), 9–20.
Kotsireas, I., Lazard, D., 1999. Central configurations of the 5-body problem with equal masses in three-dimensional space Representation theory, dynamical systems, combinatorial and algorithmic methods. Part IV. In: Zap. Nauchn. Sem. POMI, vol. 258. POMI, St. Petersburg, pp. 292–317.
Krick, T., Logar, A., 1991. An algorithm for the computation of the radical of an ideal in the ring of polynomials. In: Applied Algebra, Algebraic Algorithms and Error Correcting Codes, 9th International Symposium AAECC-9, Springer. Lecture Notes in Computer Science, vol. 539, pp. 195–205.
Monico, C., 2002. Computing the Primary Decomposition of zero-dimensional Ideals. Journal of Symbolic Computation 34, 451–459.
Pauer, F., 1992. On lucky ideals for Gröbner bases computations. Journal of Symbolic Computation 14, 471–482.
Pfister, G., 2007. On modular computation of standard basis. Analele Stiintifice ale Universitatii Ovidius, Mathematical Series XV (1) 129–137.
Sasaki, T., Takeshima, T., 1989. A modular method for Gröbner-basis construction over $\mathbb{Q}$ and solving system of algebraic equations. Journal of Information Processing 12, 371–379.
Shimoyama, T., Yokoyama, K., 1996. Localization and primary decomposition of polynomial ideals. Journal of Symbolic Computation 22, 247–277.
Traverso, C., 1989. Gröbner trace algorithms. In: Symbolic and Algebraic Computation, International Symposium ISSAC'88, Springer Lecture Notes in Computer Science, vol. 358, pp. 125–138.
Wang, P.S., Guy, M.J.T., Davenport, J.H., 1982. P-adic Reconstruction of Rational Numbers. ACM SIGSAM Bulletin 16, 2–3.
Winkler, F., 1987. A p-adic approach to the computation of Gröbner bases. Journal of Symbolic Computation 6, 287–304.