Primary Decomposition of Polynomial Ideals

Eli Vigneron

July 2020

Abstract. In this brief note, we survey some of the algorithms for computing the primary decomposition of polynomial ideals. Our discussion takes us through a number of the well known algorithms, focusing in particular on the algorithm due to Gianni, Trager, and Zacharias [GTZ88]. We then turn to a relatively recent addition to the suite of known algorithms, looking at an algorithm due to Masayuki Noro [Nor10], and discussing a class of examples to which it is well suited.

1 Introduction

Primary decomposition of polynomial ideals is an important aspect of algebraic geometry and commutative algebra. Thought of geometrically, this can be viewed as decomposing an affine variety into irreducible components.

There are a number of known algorithms for computing primary decomposition's of polynomial ideals — we will begin by providing a brief overview of some of the most well known algorithms. The first is an algorithms due to Gianni, Trager, and Zacharias [GTZ88], henceforth refered to it as the GTZ algorithm.

The GTZ algorithm works by extracting maximum dimensional primary components say, $Q_1, \ldots, Q_k$ of an ideal I by means of a reduction to the zero-dimensional case. Along the way an element $f^s \notin I$ is obtained such that

$$I = (I : f^s) \cap (I + f^s), \quad I : f^s = I : f^\infty = Q_1, \ldots, Q_k.$$

The procedure is repeated on $I + f^s$ thus obtaining what remains of the primary components of I.

Another well known algorithm is due to Shimoyama and Yokoyama [SY96] (SY algorithm). This algorithm works by first computing the set $\{P_1, \ldots, P_n\}$ of minimally associated primes of I. Then, using these primes, computations are performed producing ideals $Q'_1, \ldots, Q'_n$ as well as elements $f_1, \ldots, f_n$ such that

$$\sqrt{Q'_j} = P_j, \quad I = (Q'_1 \cap \cdots \cap Q'_n) \cap (I + (f_1, \ldots, f_n)), \quad \dim(I + (f_1, \ldots, f_n)) < \dim I.$$

Each of the Q'_j will contain just one isolated primary component Q_j of I. Further an ideal, call it I', can be computed such that $Q'_j = Q_j \cap I'$ where the dimension of I' is strictly smaller than that of I. The procedure is then applied to I' and $I+(f_1, \dots, f_n)$ resulting the remaining primary components of I.

The last algorithm that we will mention in passing is an algorithm due to Eisenbud, Huneke, and Vasconcelos [EHV92] (EHV algorithm). This algorithm has a slightly different flavour than the other two mentioned in that it does not work by reduction to the zero-dimensional case. Instead it employs methods of homological algebra to obtain the set of associated primes. The primary components are then found via localization.

All three of these algorithms have been implemented in various computer algebra systems, for example the GTZ algoithm has an implementation in MAPLE, the SY algorithm has been implemented in Macaulay 2, and the EHV algorithm in Singular.

Despite there being a number of different algorithms and approaches addressing the problem of decomposing a given ideal, it is difficult in practice to predict which algorithm will be efficient for any given ideal. We refer the reader to [DGP99] for an in-depth comparison of the above mentioned algorithms. Important additions to the known algorithms are still being made. Part of the goal of this paper will be to look at one of these recent additions which is based on the SY algorithm and due to Masayuki Noro [Nor10]. Before turning to this new algorithm, we present an overview of the GTZ algorithm and the general process of reduction to the zero-dimensional case. We end the paper by considering some examples of ideals that have been found difficult to decompose using the standard methods [NN10] but for which Noro's algorithm is well suited.

2 Zero-dimensional Decomposition's

We will start by looking at zero-dimensional primary decomposition's.

First we recall that an ideal is called zero-dimensional if the associated variety $V(I)$ is finite. So for example if we have a system of polynomial equations and we are looking to determine if there are a finite number of solutions, this is the same as asking if the corresponding ideal is zero-dimensional.

This section examines the following question. Given some zero-dimensional ideal I in the polynomial ring $k[x_1, \dots, x_n]$ where k is some field, how can we compute the primary decomposition of the ideal I, written

$$I = Q_1 \cap \cdots \cap Q_m.$$

As an initial step, we will start by discussing an algorithm for computing the primary decomposition of zero-dimensional ideals in $R[x_1, \dots, x_n]$. This algorithm can be briefly summarized as follows. Start by computing the primary decomposition of $I \cap R[x_n]$. Next extend it to a decomposition of all of I where this extended decomposition is not necessarily primary. We then proceed inductively to find a complete primary decomposition of each of the components. This inductive step can be neatly summarized as the following proposition. We make the simplifying assumption that for a given maximal ideal M of R polynomials can be factored over finitely generated algebraic extensions of R/M.

Proposition 1. *Let M be a maximal ideal of R and suppose that $I \subseteq R[x_1, \ldots, x_n]$ is a zero-dimensional ideal so that $I \cap R$ is M-primary. Then we can construct zero-dimensional ideals $I_1, \ldots, I_m$ of $R[x_1, \ldots, x_n]$ as well as distinct maximal ideals $M_1, \ldots, M_m$ of $R[x_n]$ so that $I = \bigcap_j I_j$ and each of the $I_j \cap R[x_n]$ are M_j primary.*

If we apply this proposition to M_j and I_j over $R[x_n]$ in a recursive manner, we can obtain the full primary decomposition of I together with the associated primes.

Let us state this algorithm in pseudocode, throughout we assume that $I \cap R$ is M-primary.

Procedure 1 Zero Dimensional Primary Decomposition

procedure ZPD($R[x_1, \ldots x_n], M$)

    input:

      R a ring

      $M \subset R$ a maximal ideal

      $I \subseteq R[x_1, \ldots, x_n]$ a zero-dimensional ideal

    output:

      $\{(Q_1, M_1), \ldots (Q_m, M_m)\}$ where Q_j, M_j are ideals in $R[x_1, \ldots x_n]$ the M_j are maximal, each Q_j is M_j-primary and $I = \bigcap_j Q_j$

    if $n = 0$ **then return** $\{(I, M)\}$

    $G \leftarrow$ minimal Gröbner basis for $I \cap R[x_n]$

    $g \leftarrow g \in G$ such that g has largest degree

    factorize $g \mod M$ as $g = \prod p_j^{s_j}$ in $(R/M)[x_n]$, $p_j \in R[x_n]$

    Find s so that $(\prod p_j^{s_j})^s \in I \cap R[x_n]$

    $I_j \leftarrow (p_j^{s_j s}, I)$

    $M_j \leftarrow (p_j, M)R[x_n]$

    return $\coprod_j$ ZPD($R[x_n]; x_1, \ldots, x_{n-1}; I_j; M_j$)

Before moving on we will outline an algorithm for computing primary decomposition's when R is a field of characteristic 0 and I is a zero-dimensional ideal. This algorithm makes use of the following notion.

Definition 1. *An ideal I is said to be in **general position** if it is a prime zero-dimensional ideal in $K[x_1, \ldots, x_n]$ with minimal Gröbner basis $G = \{g_1(x_1, \ldots, x_n), \ldots, g_n(x_n)\}$ under the lexicographical order such that almost all of the linear transformations of the g_j look like $g_j = x_j - p_j(x_{j+1}, \ldots, x_n)$.*

Procedure 2 Primary Decomposition over a Field

procedure ZPDF($K[x_1, \dots x_n], I$)

  input:

   K a field with characteristic 0

   $I \subseteq K[x_1, \dots, x_n]$ a zero-dimensional ideal

  output:

   $\{Q_1, \dots Q_m\}$ where each Q_j is primary and $I = \bigcap_j Q_j$

  Pick $c_1, \dots, c_{n-1} \in K$ randomly

  $x_n \leftarrow x_n + \sum c_j x_j$

  $(g) \leftarrow I \cap K[x_n]$

  $g \leftarrow \prod p_j^{s_j}$

  if $(p_j^{s_j}, I)$ is not primary and in general position **then** goto Step 4.

  $x_n \leftarrow x_n = \sum c_j x_j$

  return $\{(p_j^{s_j}, I)\}$

3 Primary Decomposition's in PIDs

This section focuses on the main steps of the GTZ algorithm. We will see that general primary decomposition's can be reduced to the zero-dimensional case, hence our treatment of them first.

We will start by looking at the following claim.

Proposition 2. *Let R be a principal ideal domain, and I an ideal of $R[x_1, \dots, x_n]$. Then there is a primary decomposition for I.*

Before proving this claim, we will need an auxiliary lemma.

Lemma 1. *Suppose that R is Noetherian. If R is an integral domain and $(p) \subset R$ a principal prime ideal. Then for any $I \subseteq R[x_1, \dots, x_n]$ there is an $r \in R \setminus (p)$ with the property that*

$$I = (I, r) \cap (IR_{(p)}[x_1, \dots, x_n] \cap R[x_1, \dots, x_n]).$$

Proof. By [GTZ88, Prop 3.7] there is an $s \in R \setminus (p)$ satisfying $IR_{(p)}[x_1, \dots, x_n] \cap R[x_1, \dots, x_n] = IR_s[x_1, \dots, x_n] \cap R[x_1, \dots, x_n]$. Thus $IR_{(p)}[x_1, \dots, x_n] \cap R[x_1, \dots, x_n]$ can be computed using the methods given in [GTZ88, Prop 3.2 (v)]. R is Noetherian so there is an m such that $s^m IR_{(p)}[x_1, \dots, x_n] \cap R[x_1, \dots, x_n] \subseteq I$. So if G is some basis for $IR_{(p)}[x_1, \dots, x_n] \cap R[x_1, \dots, x_n]$, then checking whether $s^m G$ is in I for iterated values of m will allow us to find the m we need. Setting $r = s^m$ yields the result. □

We return now to the proof of Proposition 2.

Proof. If $I \cap R$ is not zero-dimensional, then apply Lemma 1 to the zero ideal in R to obtain some nonzero r such that $I = (I, r) \cap (IR_{(0)}[x_1, \ldots, x_n] \cap R[x_1, \ldots, x_n])$. $R_{(0)}$ is the quotient field of R so we can decompose $IR_{(0)}[x_1, \ldots, x_n]$ using Procedure 2. Making use of [GTZ88, Prop 3.7] we can contract the output obtained from Procedure 2 to $R[x_1, \ldots, x_n]$. What remains is then (I, r) and this contracts to a zero-dimensional ideal in R. All said, we may assume that $I \cap R$ is zero-dimensional. Without loss of generality suppose that $I \cap R = (\prod p_j^{s_j})$ and $(p_j)R$ is maximal. Then we find that $(p_j^{m_j}) \cap R$ is (p_j)-primary. Thus we may employ the algorithm outlined in [GTZ88, Prop 8.2] to decompose $(p_j^{m_j})$. Finally since $I = \bigcap_j (p_j^{m_j}, I)$ we have found the sought after decomposition. $\square$

In summary we present the pseudocode for the GTZ algorithm.

Procedure 3 Primary Decomposition over a PID

procedure PPD($R[x_1, \ldots, x_n], I$)

  input:

   R a PID

   $I \subseteq R[x_1, \ldots, x_n]$

  output:

   $\{Q_1, \ldots Q_m\}$ where each Q_j is primary and $I = \bigcap_j Q_j$

  r is such that $I = (I, r) \cap (IR_{(0)}[x] \cap R[x])$

  $Q \leftarrow$ PPD-0($R_{(0)}; x_1, \ldots, x_n; IR_{(0)}[x_1, \ldots, x_n]; 0$)

  $Q_j^c \leftarrow Q_j \cap R[x_1, \ldots, x_n]$

  $(r') \leftarrow (I, r) \cap R$

  if r' is a unit **then return** $\{Q_1^c, \ldots, Q_m^c\}$

  $r' \leftarrow \prod p_j^{m_j}$ (p_j irreducible)

  for j = 1, ..., n **do**

   $Q^j \leftarrow$ PPD-0($R; x_1, \ldots, x_n; (I, p_j^{m_j}); p_j$)

  return $\{Q_1^c, \ldots, Q_m^c\} \cup \bigcup_j \{Q_1^j, \ldots, Q_{k_j}^j\}$

Procedure 4 Primary decomposition over a PID (Primary Contraction Case)

procedure PPD-0($R[x_1, \dots, x_n], I, p$)
  input:
    R a PID
    $I \subseteq R[x_1, \dots, x_n]$
    $p \in R$ so that $(p)R$ is maximal and $I \cap R$ is (p)-primary
  output:
    $\{Q_1, \dots Q_m\}$ where each Q_j is primary and $I = \bigcap_j Q_j$

  if I is zero-dimensional **then** call ZPD, i.e. Procedure 1
  Obtain j such that $I \cap R[x_j]$ is not zero-dimensional
  $R' \leftarrow R[x_j]$
  $x' \leftarrow x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_n$
  $I^e \leftarrow IR'_{(p)}[x']$
  Obtain $r' \in R' \setminus (p)R'$ so that $I = (I, r') \cap (I^e \cap R'[x'])$
  $\{Q_1, \dots, Q_m\} \leftarrow$ PPD-0($R'_{(p)}; x', I^e; p$)
  $Q_j^c = Q_j \cap R'[x']$
  if $(I, r') = (1)$ **then return** $\{Q_1^c, \dots, Q_m^c\}$
  $\{Q'_1, \dots, Q'_m\} \leftarrow$ PPD-0($R; x, (I, r'); p$)
  return $\{Q_1^c, \dots, Q_m^c, Q'_1, \dots, Q'_k\}$

4 Noro's Algorithm

As mentioned in the introduction, it is generally difficult to know which algorithm to use for a given ideal and in practice several methods are tried to determine the primary decomposition. A recent class of examples was found by Nishiyama and Noro [NN10] for which each of the algorithms mentioned in the introduction have difficulty decomposing. Noro's algorithm seeks to deal with this class of examples. It is similar to the SY algorithm in that all of the isolated primary components of a given ideal are determined from its minimal associated primes.

The modifications to the SY algorithm can be summarized as follows. First call an ideal I' a **remaining ideal** of a decomposition of an ideal I whenever I can be written as $I = (Q \cap \cdots \cap Q_k) \cap I'$ where the Q_j's are primary. Noro's algorithm works by keeping the primary components of I just until their intersection coincides with that of the input ideal I. That is, at each step all of the isolated primaries $Q_1, \dots, Q_k$ are computed via the minimal associated primes. Next a J is found such that $I = (Q_1, \dots, Q_k) \cap (I + J)$. Repeating this procedure with $I = I + J$ until the intersection of the found primary components coincide with I. To ensure that the ideal $I + J$ is as large as it can be, J is computed as a subset of $I : Q$.

In the SY algorithm, at each step, remaining ideals I' are found for each psuedo-primary component Q'_j and I'' — which in turn need to be decomposed recursively.

In the case where there are embedded primary components of I, such components can be found in either I' or I''. When they are found in I', I' can be written as $I' = I + (f^s)$, $f \in R$. For the class of examples under consideration, it can be difficult to decompose $I+(f^s)$ due to the unnecessary introduced components. Noro's algorithm deals with this by enlarging I'. In other words elements are added to I' while the relation $Q' = Q \cap I'$ is preserved (where Q' is some pseudo-primary ideal).

Let us now turn to the algorithm itself. We start with a definition. We set $R = k[x_1, \dots, x_n]$ where k is a field, and let I, J and Q be ideals in R.

Definition 2. *An ideal J with the following properties is called a **separating ideal** for (I, Q). J is such that*

$$J \not\subseteq I, \quad I + J \neq R, \quad I = Q \cap (I + J).$$

We note that in order to find a non-trivial decomposition of the form $I = Q \cap (I+J)$ we need a J where $J \subseteq I : Q$ and $J \not\subseteq I$. Indeed this is because if $I = Q \cap (I + J)$ then $QJ \subseteq Q \cap J \subseteq Q \cap (I + J) = I$ and we see that $J \subseteq I : Q$.

It turns out that we can use $J = (I : Q)^m$ for large enough m. Let us examine this more closely.

Proposition 3. *There is an $m > 0$ such that $(I : Q)^m \cap Q \subseteq I$.*

Proof. Artin-Rees guarantees that there is some m such that $(I : Q)^j \cap Q = (I : Q)^{j-m}((I : Q)^m \cap Q)$ for $j > m$. So then if we have an $j > m$ we can write $(I : Q)^{j-m} \subseteq I : Q$ which in turn yields $(I : Q)^{j-m}((I : Q)^m \cap Q) \subseteq (I : Q)Q \subseteq I$. $\square$

If I is an ideal of R of dimension d with an irredundant primary decomposition say $I = Q_1 \cap \cdots Q_r \cap Q_{r+1} \cap \cdots \cap Q_s$ where each of the isolated primaries are contained in $Q_1, \dots, Q_r$ then we can say the following:

1. $\dim(I : Q_1 \cap \cdots \cap Q_r)^m = \dim(I : Q_1 \cap \cdots \cap Q_r) < d$ for any $m \in \mathbb{N}$
2. When m is large enough, $I = (Q_1 \cap \cdots \cap Q_r) \cap (I + (I : Q_1 \cap \cdots \cap Q_r)^m)$ and the dimension $\dim(I + (I : Q_1 \cap \cdots \cap Q_r)^m)$ is strictly smaller than d. This is therefore a non-trivial decomposition of I.

Proof. To suppress notation, set $Q = Q_1 \cap \cdots \cap Q_r$. We have that $I : Q = (Q : Q) \cap (Q_{r+1} : Q) \cap \cdots \cap (Q_s : Q) = (Q_{r+1} : Q) \cap \cdots \cap (Q_s : Q)$. Then since $\dim(Q_j : Q) \leq \dim Q_j < d$ for all $j > r$ the first claim obtains.

By the above proposition, there is an m such that $(I : Q)^m \cap Q \subseteq I$, thus $I = Q \cap (I + (I : Q)^m)$ and the $\dim(I + (I : Q)^m) \leq \dim(I : Q)^m = \dim(I : Q) < d$. So we find that I is properly contained in $I + (I : Q)^m$. Now I must be properly contained in Q since I has an embedded component. This means that $I + (I : Q)^m$ is in fact a proper ideal since it is contained in $I + (I : Q) \subseteq (I : Q) \neq R$. Hence $I = Q \cap (I + (I : Q)^m)$ is indeed a non-trivial decomposition of I. $\square$

Suppose that we have the following auxiliary procedures:
$MinimalAssociatedPrimes(I)$ which, as the name suggests, returns the minimal associated primes of I; $IsolatedPrimaryComponents(I, \{P_1, \dots, P_k\})$ that returns the

set of all isolated primary components $Q_1, \dots Q_k$ of I. Here $\{P_1, \dots, P_k\}$ consists of all the minimal associated primes of I and each of the P_j in $\{P_1, \dots, P_k\}$ is the associated prime of Q_j. *SeparatingIdeal*$(I, Q, I : Q)$ that finds a separating ideal J for (I, Q). Lastly suppose that *RemoveRedundancy*(QL) combines primary components with their respective associated primes and removes unneeded components.

With this groundwork in-place we can now state Noro's algorithm as follows:

Procedure 5 Noro's Algorithm

procedure NORO(k, I)
    input:
      k a field
      $I \subseteq k[x_1, \dots, x_n]$
    output:
      $\{Q_1, \dots Q_m\}$ where each Q_j is primary and $I = \bigcap_j Q_j$ and the decomposition is irredundant

    $QL \leftarrow \emptyset$
    $Q \leftarrow k[x_1, \dots, x_n]$
    $I_t \leftarrow I$
    while True **do**
        if $I_t = k[x_1, \dots, x_n]$ **then**
            $QL \leftarrow RemoveRedundancy(QL)$
            return QL
        $PL_t \leftarrow MinimalAssociatedPrimes(I_t)$
        $QL_t \leftarrow IsolatedPrimaryComponents(I_t, PL_t)$
        $Q_t \leftarrow \bigcap_{J \in QL_t} J$
        if $Q \nsubseteq Q_t$ **then**
            $Q \leftarrow Q \cap Q_t$
            $QL \leftarrow QL \cup QL_t$
        if $Q_t = I_t$ or $Q = I$ **then**
            $QL \leftarrow RemoveRedundancy(QL)$
            return QL
        $J_t \leftarrow SeparatingIdeal(I_t, Q_t, (I_t : Q_t))$
        $I_t \leftarrow I_t + J_t$

5 Noro's Algorithm Examples

The class of examples that Noro's algorithm is well suited to comes from the ring of differential operators with polynomial coefficients. In other words this ring consists of expressions of the form

$$f_m(x)\partial_x^m + f_{m-1}(x)\partial_x^{m-1} + \cdots + f_1(x)\partial_x + f_0(x)$$

where each of the $f_i \in F[x]$ for some base field F. This forms an algebra known as the Weyl algebra and it is generated by x and ∂_x.

More generally an n-th Weyl algebra is the ring of differential operators with polynomial coefficients in n variables and it is generated by x_i, ∂_{x_i}.

For the examples under consideration, set $x = x_1, \dots, x_n$ and let $\mathbb{Q}(x, \partial_x)$ be the n-th Weyl algebra over $\mathbb{Q}$. Let $f \in \mathbb{Q}[x]$ then ideals in our Weyl algebra can be written as $I_f = (\partial_{x_1} + \frac{\partial f}{\partial x_1}\partial_t, \dots, \partial_{x_n} + \frac{\partial f}{\partial x_n})$.

If we let $w = e_1$ a standard basis vector with $n+1$ components, and then consider the weight vector $(-w, w)$ for $(t, x_1, \dots, x_n, \partial_t, \partial_{x_1}, \dots, \partial_{x_n})$ we find that the ideal $J_f = \mathrm{in}_{(-w,w)}(I_f) \cap \mathbb{Q}[t\partial_t]$ can be viewed as an ideal in $\mathbb{Q}[x_1, \dots, x_n, s]$, where $s = t\partial_t$. When f has non-isolated singularities it is often very difficult to determine primary decomposition's of this J_f. Typically J_f will contain a polynomial $b(s)$ and will have the decomposition $J_f = (J_f + ((s - s_1)^{m_1})) \cap \cdots \cap (J_f + ((s - s_k)^{m_k}))$ corresponding to the factorization of $b(s)$ over $\mathbb{Q}$, namely $b(s) = (s - s_1)^{m_1} \cdots (s - s_k)^{m_k}$. The goal is then to compute the primary decomposition's of each of the $J_f + ((s - s_i)^{m_i})$. It turns out that this becomes difficult whenever m_i is larger than one.

Turning to a concrete example, we let

$$f_k(x, u_1, \dots, u_k) = x^{k+1} + u_1 + u_2 x + \cdots u_k x^{k-1}.$$

and consider the ideal

$$I_1 = J_{\mathrm{disc}(f_4)} + (s^2)$$

Letting Q be the unique isolated prime component and R_j the embedded primary component, Noro's algorithm finds that

$$\begin{aligned} I_1 &= Q \cap R_1 \\ I_2 &= J_{\mathrm{disc}(f_5)} + (s^3) = Q \cap R_1 \cap R_2. \end{aligned}$$

Most implementations of GTZ, SY and EHV fail to decompose I_2. Even I_1 is often difficult to decompose. All said, we see that Noro's algorithm serves to enrich the established suite of algorithms that are geared towards computing primary decomposition's.

References

[DGP99] Wolfram Decker, Gert-Martin Greuel, and Gerhard Pfister. Primary decomposition: Algorithms and comparisons. In *Algorithmic Algebra and Number Theory*, pages 187–220. Springer Berlin Heidelberg, 1999.

[EHV92] David Eisenbud, Craig Huneke, and Wolmer Vasconcelos. Direct methods for primary decomposition. *Inventiones Mathematicae*, 110(1):207–235, December 1992.

[GTZ88] Patrizia Gianni, Barry Trager, and Gail Zacharias. Gröbner bases and primary decomposition of polynomial ideals. *Journal of Symbolic Computation*, 6(2-3):149–167, October 1988.

[NN10] Kenta Nishiyama and Masayuki Noro. Stratification associated with local b-functions. *Journal of Symbolic Computation*, 45(4):462–480, April 2010.

[Nor10] Masayuki Noro. New algorithms for computing primary decomposition of polynomial ideals. In *Mathematical Software – ICMS 2010*, pages 233–244. Springer Berlin Heidelberg, 2010.

[SY96] Takeshi Shimoyama and Kazuhiro Yokoyama. Localization and primary decomposition of polynomial ideals. *Journal of Symbolic Computation*, 22(3):247–277, March 1996.

[Vas04] Wolmer Vasconcelos. *Computational methods in commutative algebra and algebraic geometry*. Springer, Berlin New York, 2004.