Zero Decomposition Algorithms for System of Polynomial Equations 1)

D.K. Wang 2)

Abstract. This paper will give the zero decomposition algorithms of the characteristic set method. The main algorithms include those for computing characteristic set, triangular set and the zero decomposition algorithm by factorizing multi-variable polynomials. We will also give a brief description of wsolve which is an implementation of the above algorithms and use it to solve a system of polynomial equations as an example.

Key words: zero decomposition, characteristic set, polynomial factorization.

1. Introduction

In this paper, we will discuss how to give the zero decomposition of system of polynomial equations of the following form

\[
\begin{align*}
P_1(x_1, x_2, \cdots, x_n) &= 0 \\
. &= . \\
P_m(x_1, x_2, \cdots, x_n) &= 0
\end{align*}
\]

where \( K \) is a field of characteristic 0 and \( x_1, x_2, \cdots, x_n \) are indeterminates. \( P_1, P_2, \cdots, P_m \)

are polynomials in the ring \( K[x_1, x_2, \cdots, x_n] \).

Nonlinear algebraic equations of the above form arise in many theoretical and applied problems. Solving such a system is one of the central problems of computer algebra.

To solve this kind of polynomial equations, one important classic method is the resultant of \( n \) homogeneous polynomials in \( n \) variables, a tool for deciding whether these polynomials have nontrivial common solutions. In addition to the resultant method, such systems of polynomial equations can be solved by the Grobner basis method, especially in the case of dimension 0 [?]. Another method is Wu’s method, also called characteristic set method, which can be used to solve such systems for dimension 0 or higher. Here we suppose the readers are familiar with the characteristic set method. In this paper, we will give algorithms to improve the efficiency of this method.

In the following, a polynomial \( F(x_1, x_2, \cdots, x_n) \in K[x_1, x_2, \cdots, x_n] \) will be denoted as \( F(x) \). the totality of zeros of \( F \) is denoted by \( \text{Zero}(F) \),

\[
\text{Zero}(F) = \{ x | F(x) = 0 \}.
\]

1) This work is supported by the 973 project

2) Institute of Systems Science, Academia Sinica. Beijing 100080, P.R. China
Similarly, for a polynomial set $PS : P_1, \cdots, P_m$ the total common zeros of $PS$ will be denoted by $\text{Zero}(PS)$.

$$\text{Zero}(PS) = \{x | P(x) = 0 \ \forall \ P \in PS\}$$

The zeros of $PS$ which are not zeros of polynomial $J$ will be denoted by $\text{Zero}(PS/J)$

$$\text{Zero}(PS/J) = \{x | P(x) = 0, \ \forall \ P \in PS \text{ and } J(x) \neq 0\}$$

Supposing $DS$ is another polynomial set, the common zeros of $PS$ which are not zeros of the polynomials in $DS$ will be denoted by $\text{Zero}(PS/DS)$,

$$\text{Zero}(PS/DS) = \text{Zero}(PS) - \bigcup_{P \in DS} \text{Zero}(P)$$

2. The Zero Decomposition Algorithm

For a polynomial system $PS$, Wu give an algorithm $\text{CharSet}$ to compute its characteristic set which is in a triangular form\([\text{?}]\).

The $\text{CharSet}$ algorithm would form a scheme

$$PS = PS_0 \rightarrow PS_1 \rightarrow \cdots \rightarrow PS_m$$

in which

- $BS_i = \text{the basic set of } PS_i$
- $RS_i = \text{the remainder set of } PS_i \text{ w.r.t. } BS_i$
- $PS_{i+1} = PS_i + RS_i$

Let $CS = BS_m$, we will call $CS$ the characteristic set of the polynomial set $PS$, $PS$ and $CS$ have the following zero relations.

\begin{align*}
(1) \text{Zero}(CS/J) &\subset \text{Zero}(PS) \subset \text{Zero}(CS) \\
(2) \text{Zero}(PS) &\subset \text{Zero}(CS/J) + \bigcup_{i=1}^{p} \text{Zero}(PS + I_i) \tag{3}
\end{align*}

in which \(CS = C_1, C_2, \cdots, C_n, I_i = \text{Initial}(C_i), J = \prod_{i=1}^{n} (I_i)\)

For each $PS, I_i$, we do the above process repeatedly, finally we have

**Zero Structure Theorem** (Coarse Form) (Wu Wentsün) For a polynomial set $PS$, the zero set of $PS$ has the following decomposition.

$$\text{Zero}(PS) = \bigcup_{k} \text{Zero}(AS_k/J_k) \tag{4}$$

in which $AS_i$ is an ascending set, $J_k$ is the product of the initials of the polynomials in $AS_k$.

In \([\text{?}]\), an irreducible decomposition of polynomial system is given.
Zero Structure Theorem (Refined Form) (Ritt-Wu)

For a polynomial set $PS$, the zero set of $PS$ has the following decomposition.

$$\text{Zero}(PS) = \bigcup_k \text{Zero}(AS_k/J_k) \quad (5)$$

where $AS_i$ is an irreducible ascending set, $J_k$ is the product of the initials of the polynomials in $AS_k$.

For polynomial sets $PS$ and $DS$, Wu’s method can give a zero decomposition $\text{Zero}(PS/DS) = \bigcup_i \text{Zero}(AS_i/DS_i)$, in which $AS_i$ is an ascending set, $DS_i$ is a polynomial set.

Theoretically speaking, this mechanical method is complete. D.M. Wang has given a complete implementation of Wu’s method for the general purpose in Maple system [?]. For some complicated problems, it is not so efficient because of the very big size of the polynomials which is appeared in the computation of the characteristic set. For the irreducible decomposition, factorization over extension field is needed. The factorization over algebraic extension field is also time consuming. In our algorithms, we will factorize every multivariate polynomial over $\mathbb{Q}$ and the algorithm of factorizing polynomials over $\mathbb{Q}$ is very efficient in the existing symbolic computation softwares, such as Maple and Mathematica. The size of the polynomials will be reduced after factorization so that it can improve the efficiency greatly sometimes.

3. Variants of the Characteristic Set Algorithm and Triangular Set

3.1. Variants of the Characteristic Set Algorithm

The algorithm for computing the characteristic set is given as following

Algorithm: CharSet

Input: a polynomial set $PS$

Output: the characteristic set of $PS$

1. Initialize: $PS_0 = PS$, $BS = \emptyset$, if $BS$ is contradictory then return $\emptyset$.
2. Compute BS: $BS = BS(PS)$
3. Compute RS: $RS = \text{RemainderSet}(PS, BS)$, if $RS = \emptyset$ then return(BS). if $RS$ has only one polynomial $F \neq 0$ and Class(F)=0 then return $\emptyset$
4. Iteration: $PS = PS \cup RS$, go to 2.

Remark: For the above algorithm, there are many variants, we will give several variants as following.

variant 1 In the algorithms described above, if the $BS$, $RS$ become WBS (basic set in weak form) and WRS (remainder set for weak form), then the final ascending set is a weak characteristic set. The weak characteristic set is the same as characteristic set in the zero relation between $PS$ and the final WCS.

variant 2 In step Iteration, we can use $PS = PS_0 \cup RS$ instead.

variant 3 In step Iteration, if we let $PS = BS \cup RS$, the algorithm also terminate, the returned $BS$ is not a characteristic set of $PS$, but we can use the original CharSet($PS \cup BS$) to compute the characteristic set.
3.2. Algorithm for Computing the Triangular Set

From the above algorithm, for a polynomial set $PS$, we can obtain the characteristic set in a finite number of steps. But it generally costs too much time, and sometime we only need the final polynomial set in a triangular form.

The system

$$AS : A_1, A_2, \cdots, A_p$$

will be called an quasi ascending set (or triangular set) if either

(a) $p = 1$ and $A_1 \neq 0$

or

(b) $p > 1$, $\text{Class}(A_1) < \text{Class}(A_2) < \cdots < \text{Class}(A_p)$

For a polynomial set $PS$, the minimal quasi ascending set of $PS$ will be call quasi basic set of PS.

For a polynomial $F$ and a quasi ascending set $QAS = \{A_1, A_2, \cdots, A_n\}$, we define the pseudo-remainder of a polynomial $F$ to the quasi-ascending set $QAS$ as following,

$$QPremSet(F, QAS) = \begin{cases} 
\text{PseudoRemainder}(F, A_i) & \text{if for some } i, \text{Class}(F) = \text{Class}(A_i) \\
F & \text{else}
\end{cases}$$

We define the quasi-remainder set of polynomial set $PS$ w.r.t to quasi ascending set $QAS$ as following:

$$QRemainderSet(PS, QAS) = \{ R | R = QPremSet(p, QAS), p \in PS \}$$

For a polynomial set $PS$, we can use the following algorithm to get the quasi-characteristic set of $PS$.

Algorithm: Tri-Form

Input: a polynomial set $PS$

Output: the triangular form of $PS$

1. Let $QBS = QuasiBasicSet(PS)$, if $QBS$ is contradictory then return $\emptyset$.

2. Let $RS = QRemainderSet(PS, BS)$, if $RS = \emptyset$ then return(QBS).

3. Let $PS = BS \cup RS$, go to 1.

For such a quasi-characteristic set $QCS$, we only have

$$\text{Zero}(PS) \subset \text{Zero}(QCS).$$

4. The Modified Zero Decomposition Algorithms

Polynomial factorization over $\mathbb{Q}$ is one of most efficient method to improve the zero decomposition algorithm. We notice that: if there is a polynomial $F \in PS$ and $F = F_1 F_2$ then $F_1$ and $F_2$ have lower degrees and generally small sizes than that of $F$, and $\text{Zero}(PS) = \text{Zero}(PS + F_1/F) + \text{Zero}(PS + F_2/F)$. 
Consider the following problem.

\[
PS = \begin{cases} 
    p_1 = x_1 + x_2 + x_3 + x_4 + x_5 \\
    p_2 = x_1 x_2 + x_2 x_3 + x_3 x_4 + x_4 x_5 + x_5 x_1 \\
    p_3 = x_1 x_2 x_3 + x_2 x_3 x_4 + x_3 x_4 x_5 + x_4 x_5 x_1 + x_5 x_1 x_2 \\
    p_4 = x_1 x_2 x_3 x_4 + x_2 x_3 x_4 x_5 + x_3 x_4 x_5 x_1 + x_4 x_5 x_1 x_2 + x_5 x_1 x_2 x_3 \\
    p_5 = x_1 x_2 x_3 x_4 x_5 - 1 
\end{cases} \tag{6}
\]

Without factorization, for the above PS, the characteristic set cannot be obtained by the CharSet algorithm or the modified versions. Even the computation of weak characteristic set and the quasi-characteristic set also failed in a reasonable time limit. The reason is that the polynomial sizes become very large and the computation take too much time and memory.

Now the following definition is useful.

**Definition** For a polynomial set PS and an ascending set AS, the AS is called the sub-characteristic of PS, if \(\text{RemainderSet}(PS, AS) = \emptyset\), i.e. the remainder set of PS w.r.t. AS is empty.

If we factorize every polynomial in the remainder set, we have

**Algorithm (SCS)**

Input: a polynomial set PS

Output: a series of sub-characteristic sets

1. To find the basic set \(BS = \text{BasicSet}(PS)\), if BS is contradictory then return BS

2. Compute the Remainder set \(\text{RemainderSet}(PS, BS)\) of PS w.r.t BS, if RS is empty then return BS.

3. Factorize the polynomials in \(RS = \{R_1, R_2, \cdots, R_s\}\). Suppose

\[
R_1 = R_1^{i_1} \cdot \cdots \cdot R_1^{i_n}, \\
R_2 = R_2^{j_1} \cdot \cdots \cdot R_2^{j_n}, \\
R_s = R_s^{i_1} \cdot \cdots \cdot R_s^{i_n},
\]

4. Give all the possible combinations of the form \(RS_j = \{R_1^{j_1}, R_2^{j_2}, \cdots, R_s^{j_s}\}\). \(j = \{j_1, \cdots, j_s\}\);

\(j_i \leq n_i\).

5. Let \(PS_j = PS + RS_j\) of which \(RS_j\) is of the above form. Return \(\sum_j SCS(PS_j)\).

**Remark:**

1. Certainly, we can factorize the polynomials in the original given polynomial set \(PS\) first, but in general, the original \(PS\) can not be factorized over \(Q\).

2. In the step 4 of the above algorithm, if \(RS_i\) includes another \(RS_j\), we will delete the \(RS_i\). The reason is based on the following simple fact:

Suppose \(RS: \{R_1, R_2\}\) is factorized as following,

\[
R_1 = FG_1, \quad R_2 = FG_2
\]

According to the above algorithm in the step 3, we will obtain the polynomial sets \(RS_1 = \{F\}, RS_2 = \{F, G_2\}, RS_3 = \{2, FG_1\}, RS_4 = \{G_1, 2\}\), but we only need two of them: \(RS_1\) and \(RS_4\).
3. In the last step of the above algorithm, $SCS_1(PS_j)$ means for $PS_j$, let $PS$ be $PS_j$ and go to step 1.

From the above algorithm, one can easily see that $PS$ and sub-characteristic set $AS_i$s have the following zero relations.

**Zero Structure Theorem (Modification version)**

\[
\begin{align*}
\text{Zero}(PS) & \subset \bigcup_i \text{Zero}(AS_i) \\
\text{Zero}(AS_i / J_i) & \subset \text{Zero}(PS) \\
\text{Zero}(PS) & = \bigcup_i \text{Zero}(AS_i / J_i) + \bigcup_i \bigcup_j \text{Zero}(PS + I_{ij})
\end{align*}
\]

where $AS_i : A_{i1}, A_{i2}, \ldots, A_{in}$ is an ascending set, $I_{ij}$ is the initial of $A_{ij}$ and $J_i = \prod_j I_{ij}$. For each $i$, $AS_i$ is sub-characteristic set of $PS$ and $AS_i$ is an ascending set. For each $PS + I_{ij}$, we will decompose it to ascending set series, then we have the SCSS algorithm.

**Algorithm SCSS**

*Input:* a polynomial set $PS$

*Output:* a sequence of ascending sets.

1. find the sub-characteristic set $AS_i$ of the $PS$ and the enlarged polynomial set $PS'_i$ obtained in algorithm $SCS$, $AS_i : A_{i1}, A_{i2}, \ldots, A_{in}$.

2. for each $i$, let $I_{ij} = \text{Initial}(A_{ij})$, $I_i = \bigcup_j I_{ij}$, let $PS_{ij} = PS'_i + I_{ij}$, return $\bigcup_i (AS_i) \cup \bigcup_j \text{SCSS}(PS_{ij})$

**Theorem** For a polynomial set $PS$, the zero set of $PS$ has the following decomposition.

\[
\text{Zero}(PS) = \sum_k \text{Zero}(AS_k / J_k)
\]

in which $AS_k$ is an ascending set.

**Corollary** Given two polynomial sets $PS$ and $DS$, there is an algorithm to decompose the $\text{Zero}(PS/DS)$ as

\[
\text{Zero}(PS/DS) = \text{Zero}(AS_k / DS_k)
\]

in which $AS_k$ is an ascending set, $DS_k$ is a polynomial set.

**Algorithm SCSS2**

*Input:* a polynomial set $PS$

*Output:* a sequence of ascending sets.

1. if $\text{Zero}(PS)$ is the zero of some $D \in DS$ then return $\emptyset$.

2. Find the basic set of $PS$, $BS = \text{BasicSet}(PS)$ and $\text{InitialSet} = \bigcup_i \text{InitialSet}(B_i)$. $DS = DS \cup \text{InitialSet}$

3. Let $RS = \text{RemainderSet}(PS, BS)$, if $RS = \emptyset$ then return $BS$.

4. Factorize the polynomials in $RS = R_1, R_2, \ldots, R_s$.

   Suppose
   \[ R_1 = R_1^{n_1} \star \cdots \star R_1^{n_1} \]
   \[ R_2 = R_2^{n_2} \star \cdots \star R_2^{n_2} \]
   \[ R_s = R_s^{n_s} \star \cdots \star R_s^{n_s} \]
5. Remove the factors of \( R_i \) in \( DS \), and give all the possible combinations of the form 
\[
RS_i = \{R_i^1, R_i^2, \ldots, R_i^s\} \quad i = \{i_1, \ldots, i_s\}.
\]

6. Let \( PS_i = BS + RS_i \) in which \( RS_i \) is of the above form. Let \( PS_j = BS + I_j \) in which 
\( I_j \in \text{InitialSet} \) and return \( \sum_i SCSS2(PS_i) \).

To check if the \( \text{Zero}(PS) \) is the zero of \( D \) in step 1, we will use the trick first proposed 
by Chou and Gao in [?].

A set will be given for collecting the initials in \( BS_i \). From [?], one can see that the initials, multiplied during the pseudo-division, of the polynomials in \( BS_i \) will become the factors of the polynomials in the following \( RS_j \). If the factors occur when we factorize the 
polynomials in \( RS \) in the step 3, we will remove them. Certainly, this may remove some 
solutions of the system, so another set will be given to collect the removed factors. We will 
add the removed factors to the original polynomial set respectively and compute again. This 
technique has the largest possibility of success in the quasi-sense. In the standard sense and 
the weak sense, it has less possibility of success because the initials have been changed in 
form.

5. A Maple Package for Solving System of Polynomial Equations

According to the above algorithms, a package \textit{wsolve} based on the computer algebra 
systems \textit{MAPLE} has been implemented. In real procedure, there are four modes to be 
selected for the user, these modes are

1. Standard sense
2. Weak sense
3. Quasi-sense
4. Resultant-sense

The specification of \textit{wsolve} is as follows:

\[
Tlist \leftarrow \text{wsolve}(PS, X)
\]

Input:
\[
PS \text{ is a list of polynomials;}
X \text{ is a list of ordered indeterminates.}
\]

Output:
\[
Tlist \text{ is either } NIL, \text{ which implies } \text{Zero}(PS) = \emptyset, \text{ or}
\text{ a list, which consists of a finite number of ascending set } AS_i,
\text{ Zero}(PS) = \bigcup_i \text{Zero}(AS_i/J_i) \text{ where } J_i \text{ is the product of the initials of}
\text{ the polynomials in } AS_i. \text{ If } PS \text{ has only finitely many solutions, then}
\text{ Zero}(PS) = \bigcup_i \text{Zero}(AS_i)
\]
Example  Give the zero decomposition for the following polynomial set.

\[
PS = \begin{cases} 
P_1 = a^2bc + ab^2c + abc^2 + ab + ac + bc \\
P_2 = a^2b^2c + ab^2c^2 + bc^2 + c + a + a^2b \\
P_3 = a^2c^2 + b^2c^2 + c^2 + 1 + a^2 + a^2b^2
\end{cases}
\]

By \texttt{wsolve}, we can get the zero decomposition for any order in few seconds. For the order \( b < c < a \), we have the zero decomposition.

\[
\text{Zero}(PS) = \text{Zero}(AS_1) + \text{Zero}(AS_2) + \text{Zero}(AS_3)
\]

\[
AS_1 = \begin{cases} 
1 + b^2 \\
-1 - 2bc + 2bc^3 - c^4 \\
-(ab) + 2ac - bc + 2c^2 - abc^2 + bc^3
\end{cases}
\]

\[
AS_2 = \begin{cases} 
-1 - 4b^2 + 6b^4 - 4b^6 - b^8 \\
-5 + 7b^2 - 5b^4 - b^6 - 12bc + 14b^3c - 8b^5c - 2b^7c \\
a + b
\end{cases}
\]

\[
AS_3 = \begin{cases} 
1 + 4b^2 - 6b^4 + 4b^6 + b^8 \\
b + c \\
-a + 2b - 2b^3 - ab^4
\end{cases}
\]

References