

# Multiplicity Preserving Triangular Set Decomposition of Two Polynomials

Jin-San Cheng, Xiao-Shan Gao

Key Lab of Mathematics Mechanization

Institute of Systems Science, AMSS, Chinese Academy of Sciences

jcheng@amss.ac.cn, xgao@mmrc.iss.ac.cn

## Abstract

In this paper, a multiplicity preserving triangular set decomposition algorithm is proposed for a system of two polynomials. The algorithm decomposes the variety defined by the polynomial system into unmixed components represented by triangular sets, which may have negative multiplicities. In the bivariate case, we give a complete algorithm to decompose the system into multiplicity preserving triangular sets with positive multiplicities. We also analyze the complexity of the algorithm in the bivariate case. We implement our algorithm and show the effectiveness of the method with extensive experiments.

**Keywords.** Triangular set decomposition, multiplicity preserving decomposition, extended Euclidean algorithm.

## 1 Introduction

Decomposing a polynomial system into triangular sets is a classical method to solve polynomial systems. The method was first introduced by Ritt [20] and revised by Wu in his work of elementary geometry theorem proving [24, 25]. There exist many work about this topic [1, 3, 4, 6, 8, 10, 11, 13, 15, 16, 17, 18, 22, 27]. The main tool to decompose a polynomial system is pseudo-division. In most existing triangular decomposition methods based on the pseudo-division algorithm, one need to deal with the initial of certain polynomial(s), say  $h$ , which will bring extraneous zeros. Usually, one decomposes here the system into two systems corresponding to the cases  $h = 0$  and  $h \neq 0$ . Doing so, the number of the systems increases quickly. Moreover, this leads to some repeated computations which can be avoided.

Another reason why we consider the topic is that the multiplicity of a component or a zero of a polynomial system is an important information which helps us to obtain a further understanding of the structure of the variety defined by the polynomial system.

Most triangular set decomposition algorithms do not preserve the multiplicities of the zeros or the components. One approach to remedy this drawback is to decompose the polynomial system into triangular sets first and then recover the multiplicities. Li proposed a method to compute the multiplicities of zeros of a zero-dimensional polynomial system after obtaining a triangular decomposition of the system [12]. Recently, Li, Xia, and Zhang proved that the characteristic sets in Wu's sense for zero-dimensional polynomial system is actually multiplicity preserving with a minor modification [14]. They also gave a multiplicity preserving decomposition, but some of the components are not in triangular form.

In this paper, we use the concept of multiplicative variety, that is, the components and their multiplicities in the original polynomial system. We consider not only the components themselves but the multiplicities of these components. During the decomposition, the initials bring some extraneous multiplicative varieties in each pseudo-division step. We record them during the computation and remove them later, which helps us to recover the multiplicative varieties of the original system. We also avoid some repeated computation during the decomposition. Currently, the theory is complete for polynomial system with two polynomials. In particular, we provide a method to compute the multiplicative-zeros of a zero-dimensional bivariate system with two polynomials. We also analyze the complexity of the algorithm under some conditions.

Kalkbrenner's method for zero-dimensional bivariate polynomial system is similar to our method [17]. But his method is not multiplicity preserving. And our method is in a different sense: we remove the extraneous zeros from the system.

The paper is organized as below. In the next section, we provide some properties of primitive polynomial remainder sequences. In Section 3, we provide the theories to decompose a polynomial system with two polynomials into triangular sets which preserve the multiplicities of the components of the original system. We provide a multiplicity preserving algorithm to decompose a zero-dimensional bivariate polynomial system into triangular sets in Section 4. The complexity of the algorithm under some conditions are analyzed. Algorithms and examples are used to illustrate the effectiveness and efficiency of our method. We also compare our method with other related methods. We draw a conclusion in the last section.

## 2 Primitive Polynomial Remainder Sequence

In this section, we introduce some basic properties for primitive polynomial remainder sequences. In fact, there are many references for this topic, in particular [2, 16, 17]. We modify the procedure for our own purpose.

Let  $K$  be a computable field with characteristic zero, such as the field of rational numbers and  $K[y_1, \dots, y_n]$  the polynomial ring in the indeterminates  $y_1, \dots, y_n$ .

Let  $p \in K[x_1, \dots, x_n, x]$ . We define

$$\begin{aligned}\text{Cont}(p, x) &= \gcd(\text{coeff}(p, x^i), i = 0, 1, \dots, \deg(p, x)), \\ \text{Prim}(p, x) &= p/\text{Cont}(p, x),\end{aligned}$$

where  $\text{coeff}(p, x^i)$  means the coefficient of  $x^i$  in  $p$  and  $\deg(p, x)$  means the degree of  $p$  in  $x$ .  $p$  is called **primitive** w.r.t.  $x$  if  $\text{Cont}(p, x) = 1$ .

The pseudo-division can be extended to the following form.

**Lemma 2.1** *Let  $f, g \in K[x_1, \dots, x_n, x]$ ,  $\deg(f, x) = d_1$ ,  $\deg(g, x) = d_2$ ,  $d_1 \geq d_2$ , and  $\gcd(f, g) = 1$ . There exist  $q, r \in K[x_1, \dots, x_n, x]$  such that*

$$l^{\delta+1}f + qg = r, \quad (1)$$

where  $l$  is the leading coefficient of  $g$  in  $x$ ,  $\delta = d_1 - d_2$ ,  $\deg(g, x) > \deg(r, x)$ . Furthermore,  $q$  has the form:

$$q = ltx + s, \quad (2)$$

where  $t \in K[x_1, \dots, x_n, x]$ ,  $s \in K[x_1, \dots, x_n]$ . Moreover, if  $r_1 = \text{Cont}(f, x)$ ,  $r_2 = \text{Cont}(g, x)$ , then

$$r_1|q, r_2^{d_1-d_2}|q, r_1|r, r_2^{d_1-d_2+1}|r. \quad (3)$$

**Proof.** Write  $f, g$  as univariate polynomials in  $x$ ,

$$\begin{aligned}f &= a_1 x^{d_1} + a_2 x^{d_1-1} + \dots + a_{d_1+1}, \\ g &= b_1 x^{d_2} + b_2 x^{d_2-1} + \dots + b_{d_2+1}.\end{aligned}$$

To eliminate the terms of  $f$  with degree  $d_1$  in  $x$ , we have

$$T_0(x) = b_1 f + q_0 g = h_0 x^{d_1-1} + \text{lower powers in } x,$$

where  $q_0 = -a_1 x^{d_1-d_2}$ ,  $h_0 = b_1 a_2 - a_1 b_2$ . It is clear that  $r_1|q_0$ , since  $r_1|a_1$ ,  $r_2^0(=1)|q_0$  and  $r_1|T_0, r_2|T_0$ . So the lemma holds when  $\delta = 0$ . Note that  $r = T_0$  when  $\delta = 0$ . Now, we need to eliminate  $h_0 * x^{d_1-1}$  from  $T_0(x)$ . If  $h_0 \neq 0$ ,

$$\begin{aligned}T_1(x) &= b_1 T_0(x) - (b_1 a_2 - a_1 b_2) x^{d_1-d_2-1} g \\ &= b_1^2 f + (b_1 q_0 - (b_1 a_2 - a_1 b_2) x^{d_1-d_2-1}) g \\ &= b_1^2 f + q_1 g \\ &= h_1 x^{d_1-2} + \text{lower powers in } x,\end{aligned}$$

where  $h_1 \in K[x_1, \dots, x_n]$ . Each term of  $q_1$  contains a factor of the form  $a_i b_j$ . So  $r_1|q_1, r_2|q_1$  and  $r_1|T_1, r_2^2|T_1$ . And  $q_1 = -b_1 a_1 x^{d_1-d_2} - (b_1 a_2 - a_1 b_2) x^{d_1-d_2-1}$ . If  $h_0 = 0$ , the results is

still true. So the lemma holds when  $\delta = 1$ . Assuming that the lemma holds for the cases  $\delta \leq i$ , then we have

$$\begin{aligned} T_j(x) &= b_1^{j+1} f + q_j g = h_j x^{d_1-j-1} + \text{lower powers in } x, \\ q_j &= b_1 q_{j-1} - h_{j-1} x^{d_1-d_2-j}, \\ r_1 | q_j, r_2^j | q_j, r_1 | T_j, r_2^{j+1} | T_j, j \leq i. \end{aligned}$$

Note that  $\deg(q_{j-1}, x) > \deg(q_j, x)$  and the lowest power of  $q_{j-1}$  in  $x$  is larger than  $d_1 - d_2 - j$  and  $r_1 | h_j, r_2^{j+1} | h_j$ . Then

$$T_{i+1}(x) = b_1^{i+2} f + q_{i+1} g = h_{i+1} x^{d_1-i-2} + \text{lower powers in } x,$$

We can similarly derive  $q_{i+1} = b_1 q_i - h_i x^{d_1-d_2-i-1}$ . When  $\delta = i+1$ , we have  $q_{i+1} = b_1 q_i - h_i$ . So  $q_{i+1}$  has form (2) since the lowest power of  $q_i$  is larger than 0. And  $r_1 | q_{i+1}$  since  $r_1 | q_i, r_1 | h_i$ . So  $r_1 | T_{i+1}(x)$ .  $r_2^{i+1} | q_{i+1}$  since  $r_2 | b_1, r_2^i | q_i$  and  $r_2^{i+1} | h_i$ . So  $r_2^{i+2} | T_{i+1}(x)$ . So the lemma holds for  $\delta = i+1$ . The lemma is proved.  $\blacksquare$

**Corollary 2.2** *Let  $f, g \in K[x_1, \dots, x_n, x]$  be primitive,  $d_1 = \deg(f, x) \geq d_2 = \deg(g, x)$ , and  $\gcd(f, g) = 1$ . Regard  $f, g$  as univariate polynomials in  $x$ . Then there exist an  $m \in K[x_1, \dots, x_n]$  such that*

$$m f = q g + r,$$

and  $\gcd(m, q) = \gcd(m, r) = 1$ . Furthermore,

$$(m f, g) = (g, r),$$

where  $(P)$  represents the ideal generated by  $P$ .

**Proof.** The corollary is obvious.  $\blacksquare$

**Corollary 2.3** *Cont( $g, x$ ) = 1 if  $\gcd(l, s) = 1$  and  $d_1 > d_2$ , where  $l$  and  $s$  are from (1) and (2).*

**Proof.** Regard  $f, g$  as univariate polynomials in  $x$ , and  $q$  a polynomial in  $x$  and  $a_i, b_j$ , where  $i = 1, \dots, d_1 + 1, j = 1, \dots, d_2 + 1$ . Let  $r_2 | g$  and  $r_2 \in K[x_1, \dots, x_n]$ . From Lemma 2.1,  $r_2 | q$  if  $d_1 > d_2$ . So  $r_2 | s$ . Since  $r_2 | l, r_2 | \gcd(l, s)$ . We have  $r_2 = 1$  if  $\gcd(l, s) = 1$ . The lemma is proved.  $\blacksquare$

The above result is a necessary condition to check whether  $g$  has factors in  $K[x_1, \dots, x_n]$ .

**Lemma 2.4** *Let  $f_1, f_2 \in K[x_1, \dots, x_n, x]$ ,  $d_1 = \deg(f_1, x) \geq d_2 = \deg(f_2, x)$ . Assume that  $\text{Cont}(f_i, x) = 1, i = 1, 2$ . Applying the extended Euclidean algorithm for  $f_1, f_2$  w.r.t. the variable  $x$ , we obtain a polynomial sequence  $\{f_1, f_2, \dots, f_{k+2}\}$  such that*

$$m_i f_i + q_i f_{i+1} = m_{i-1} p_i f_{i+2}, i = 1, \dots, k, \quad (4)$$

where  $m_0 = 1, p_k = 1, m_i, p_i, f_{k+2} \in K[x_1, \dots, x_n], q_i \in K[x_1, \dots, x_n, x], i = 1, \dots, k$ , and  $\text{Cont}(f_i, x) = 1 (1 \leq i \leq k+1), \text{gcd}(m_i, p_i) = 1$ .

**Proof.** We prove the lemma by induction on  $i$ . When  $i = 1$ , from Lemma 2.1, there exist  $q \in K[x_1, \dots, x_n, x], r \in K[x_1, \dots, x_n, x]$  such that  $l_2^{\delta+1} f_1 + q f_2 = r$ , where  $l_i$  is the leading coefficient of  $f_i$  in  $x, \delta = d_1 - d_2$ . Let  $t = \text{gcd}(l_2^{\delta+1}, q), m_1 = \frac{l_2^{\delta+1}}{t}, q_1 = \frac{q}{t}$ . Let  $p_1 = \frac{\text{Cont}(r, x)}{t}$  and  $f_3 = \text{Prim}(r, x)$ . It is clear that  $\text{gcd}(m_1, p_1) = 1$ . Assume that for  $1 \leq j < i$ , (4) holds. Denote  $d_i = \text{deg}(f_i)$ . For  $j = i$ , we have  $l_{i+1}^{\theta+1} f_i + q_t f_{i+1} = r_{i+2}$ , where  $\theta = d_i - d_{i+1}$ . If  $m_{i-1}$  is a factor of  $r_{i+2}$ , set  $p'_i$  as the product of all the factors of  $\frac{r_{i+2}}{m_{i-1}}$  in  $K[x_1, \dots, x_n]$ . Let  $h = \text{gcd}(l_{i+1}^{\theta+1}, p'_i)$ . Then  $m_i = \frac{l_{i+1}^{\theta+1}}{h}, q_i = \frac{q_t}{h}, p_i = \frac{p'_i}{h}, \text{gcd}(m_i, p_i) = 1$ . If  $m_{i-1}$  is not a factor of  $r_{i+2}$ , we can multiply  $g = \frac{m_{i-1}}{\text{gcd}(m_{i-1}, r_{i+2})}$  to the two sides of the equation. Then doing the same operation as before, we can derive  $m_i f_i + q_i f_{i+1} = m_{i-1} p_i f_{i+2}$  which satisfies all the conditions. This proves the lemma.  $\blacksquare$

**Remark:** In most cases, we have  $\text{gcd}(m_i, q_i) = 1$  and  $p_i = 1$  which helps us to design efficient algorithms.

**Corollary 2.5** Let  $f_1, f_2 \in K[x_1, \dots, x_n, x], \text{gcd}(f_1, f_2) = 1$ , and  $\text{Cont}(f_i, x) = 1, i = 1, 2$ . From the extended Euclidean algorithm, we can obtain

$$m_i f_i + q_i f_{i+1} = g_i f_{i+2}, i = 1, \dots, k, \quad (5)$$

$$(m_i f_i, f_{i+1}) = (f_{i+1}, g_i f_{i+2}), \quad (6)$$

where  $m_i, g_i \in K[x_1, \dots, x_n], \text{gcd}(m_i, g_i) = 1, g_k = 1, \text{gcd}(m_k, g_k f_{k+2}) = 1$ , and  $f_{i+2} (1 \leq i \leq k-1)$  are primitive.

**Proof.** From Lemma 2.4, we have (4). Note that  $\text{gcd}(m_i, p_i) = 1$ . Let  $h = \text{gcd}(m_i, m_{i-1})$ , denote  $m_i = \frac{m_i}{h}, q_i = \frac{q_i}{h}, g_i = \frac{m_{i-1} p_i}{h}$ . Then we have  $\text{gcd}(m_i, g_i) = 1$ . Since  $f_{k+2} \in K[x_1, \dots, x_n]$ , we can set  $g_k = 1$ . We can delete  $\text{gcd}(m_k, g_k f_{k+2})$  if it exists. (6) is obvious. So the corollary holds.  $\blacksquare$

The following corollary is clear and useful.

**Corollary 2.6** We can rewrite (5) and (6) as below.

$$m_i f_i + q_i f_{i+1} = \frac{m_{i-1}}{w_i} p_i f_{i+2}, i = 1, \dots, k, \quad (7)$$

$$(m_i f_i, f_{i+1}) = (f_{i+1}, \frac{m_{i-1}}{w_i} p_i f_{i+2}), \quad (8)$$

where  $w_i$  is a factor of  $m_{i-1}, g_i = \frac{m_{i-1}}{w_i} p_i$ , and  $p_k = 1$ .

### 3 Triangular Decomposition of Two Polynomials

In this section, we will give the method to decompose a system of two polynomials into triangular sets. We need the concept of multiplicity variety.

**Definition 3.1** ([9] pp. 129-130) *Let  $V_d$  be an unmixed variety of dimension  $d$  in a projective space  $S_n$  of dimension  $n$  over  $K$ . And*

$$V_d = \sum_{i=1}^h V_d^{(i)},$$

where  $V_d^{(i)}$  is an irreducible variety of dimension  $d$  and order  $g_i$ . Let  $F_i(u_0, \dots, u_d)$  be the Chow form (see [9] pp.32) of  $V_d^{(i)}$ ; which is irreducible over  $K$  and of degree  $g_i$  in  $u_j = (u_{j0}, \dots, u_{jn})$ , for  $j = 0, \dots, d$ . The form

$$F(u_0, \dots, u_d) = \prod_{i=1}^h [F_i(u_0, \dots, u_d)]^{a_i}, \quad (9)$$

where  $a_1, \dots, a_h$  are positive integers, satisfies the conditions for a Chow form of an algebraic variety which, regarded as a set of points, coincides with  $V_d$ . We consider a new entity, consisting of the variety  $V_d$  associated with the form  $F(u_0, \dots, u_d)$ , for a given choice of the exponents  $a_1, \dots, a_h$ , denoted as  $\mathbb{M}_d$ . We write

$$\mathbb{M}_d = \sum_{i=1}^h a_i \mathbb{M}_d^{(i)}, \quad (10)$$

where  $\mathbb{M}_d^{(i)}$  corresponds to  $V_d^{(i)}$ . We call  $\mathbb{M}_d$  a **multiplicative variety** and  $a_i$  the **multiplicity** of  $\mathbb{M}_d^{(i)}$ . Especially, we call  $\mathbb{M}_d$  a **multiplicative-zero set** when  $d = 0$ .

**Remark:** Since an affine variety can be easily transformed into a projective variety, we will consider directly affine multiplicative varieties in  $K^n$  in this paper. And we assume that the system has no solutions at  $\infty$ .

**Theorem 3.2** ([9] pp160)  $\mathbb{M}_a, \mathbb{M}'_b, \mathbb{M}''_b$  are unmixed multiplicative varieties with dimension  $a, b$  respectively. The intersection of  $\mathbb{M}_a$  and  $\mathbb{M}'_b + \mathbb{M}''_b$  are with dimension  $a + b - n$ . So are the intersections of  $\mathbb{M}_a$  and  $\mathbb{M}'_b$ ,  $\mathbb{M}_a$  and  $\mathbb{M}''_b$ . Then we have

$$\mathbb{M}_a \cdot (\mathbb{M}'_b + \mathbb{M}''_b) = \mathbb{M}_a \cdot \mathbb{M}'_b + \mathbb{M}_a \cdot \mathbb{M}''_b,$$

where  $\cdot$  represents the intersection of two multiplicative varieties, which preserves the multiplicities of each intersection component (for more details see [9], pp158-160).

**Definition 3.3** ([5] pp.139) Let  $I$  be a zero dimensional ideal in  $K[x_1, \dots, x_n]$  such that the variety  $V(I)$  defined by  $I$  consists of finitely many points in  $\overline{K}^n$ , where  $\overline{K}$  is algebraic closure of  $K$ , and assume  $p = (a_1, \dots, a_n) \in V(I)$ . Then the **multiplicity** of  $p$  as a zero of  $I$ , denoted by  $m(p)$ , is the dimension of the ring obtained by localizing  $\overline{K}[x_1, \dots, x_n]$  at the maximal ideal  $M = I(p) = (x_1 - a_1, \dots, x_n - a_n)$  corresponding to  $p$ , that is:

$$m(p) = \dim_{\overline{K}} \overline{K}[x_1, \dots, x_n]_M / I \overline{K}[x_1, \dots, x_n]_M.$$

**Lemma 3.4** ([5] pp. 144 ) Let  $\{f_1, \dots, f_n\} \subset K[x_1, \dots, x_n]$  be zero-dimensional, and have total degrees at most  $d_1, \dots, d_n$  and no solutions at  $\infty$ . If  $f_0 = u_0 + u_1 x_1 + \dots + u_n x_n$ , where  $u_0, \dots, u_n$  are independent variables, then there is a nonzero constant  $C$  such that the Chow form of  $(f_1, \dots, f_n)$  is

$$\begin{aligned} & \text{Res}_{1, d_1, \dots, d_n}(f_0, \dots, f_n) \\ &= C \prod_{p \in V_{\overline{K}}(f_1, \dots, f_n)} (u_0 + u_1 \xi_1 + \dots + u_n \xi_n)^{m(p)}, \end{aligned}$$

where  $p = (\xi_1, \dots, \xi_n) \in V_{\overline{K}}(f_1, \dots, f_n)$  and  $m(p)$  is the multiplicity of  $p$  in  $(f_1, \dots, f_n)$ .

The lemma illustrates the relationship between Chow form and the multiplicity of a point of a zero-dimensional polynomial system. The lemma tells us that  $m(p)$  is the multiplicity of the corresponding irreducible zero-dimensional component of the zero-dimensional polynomial system  $\{f_1, \dots, f_n\}$  when  $d = 0$  in Definition 3.1.

From Theorem 3.2, we have the corollary below.

**Corollary 3.5** Using the notations as Corollary 2.2, we have

$$\mathbb{M}(m f, g) = \mathbb{M}(m, g) + \mathbb{M}(f, g).$$

The following lemma is important to our algorithm.

**Lemma 3.6** Let  $f, g, r, m$  be as Corollary 2.2. We have

$$\mathbb{M}(f, g) = \mathbb{M}(g, r) - \mathbb{M}(m, g). \tag{11}$$

**Proof.** By Corollary 2.2,  $(m f, g) = (g, r)$ , so we have

$$\mathbb{M}(m f, g) = \mathbb{M}(g, r). \tag{12}$$

We can find that  $m f, g$  define a multiplicative variety with dimension  $n - 2$  since  $\gcd(m f, g) = 1$ . Note that  $f, g$  both are primitive and  $m \in K[x_1, \dots, x_n]$ . And  $f, g$  and  $m, g$  both are  $n - 2$  dimensional.

We are going to prove that

$$\mathbb{M}(f, g) = \mathbb{M}(mf, g) - \mathbb{M}(m, g). \quad (13)$$

Assume that the Chow form  $T_0$  of  $\mathbb{M}(mf, g)$  is as (9) and the order of  $\mathbb{M}(mf, g)$  is  $G_0 = \sum_{i=1}^h a_i g_i$ . From (3.5), we can assume that the Chow form of  $\mathbb{M}(m, g), \mathbb{M}(f, g)$  are  $T_1 = \prod_{i=1}^k [F_i(u_0, \dots, u_d)]^{a_i}$  and  $T_2 = \prod_{i=k+1}^h [F_i(u_0, \dots, u_d)]^{a_i}$ , respectively, where  $1 < k < h$  and  $d = n - 2$ . And the orders are  $G_1 = \sum_{i=1}^k a_i g_i$  and  $G_2 = \sum_{i=k+1}^h a_i g_i$ , respectively. We define the Chow form  $T'_2$  of  $\mathbb{M}(mf, g) - \mathbb{M}(m, g)$  as  $T_0/T_1$  and the order as  $G'_2 = G_0 - G_1$ . Since the Chow form of the algebraic variety is not equal to zero, the definition is well defined. Here  $(m, g)$  defines some components of  $(mf, g)$ , including the multiplicities of the components. Thus  $\mathbb{M}(mf, g) - \mathbb{M}(m, g)$  still has positive exponent for each simplified component. And we have

$$\begin{aligned} T'_2 &= \prod_{i=1}^h [F_i(u_0, \dots, u_d)]^{a_i} / (\prod_{i=1}^k [F_i(u_0, \dots, u_d)]^{a_i}) = \prod_{i=k+1}^h [F_i(u_0, \dots, u_d)]^{a_i} = T_2, \\ G'_2 &= \sum_{i=1}^h a_i g_i - \sum_{i=1}^k a_i g_i = \sum_{i=k+1}^h a_i g_i = G_2. \end{aligned}$$

So (13) holds. Combining (12) and (13), we have (11). This ends the proof.  $\blacksquare$

**Definition 3.7** *A multiplicity preserving triangular decomposition of a polynomial system  $\Sigma$  is a group of triangular sets  $\{T_i^+, T_j^-, i = 1, \dots, m^+, j = 1, \dots, m^-\}$  in multiplicative variety sense such that*

$$\mathbb{M}(\Sigma) = \sum_{i=1}^{m^+} \mathbb{M}(T_i^+) - \sum_{j=1}^{m^-} \mathbb{M}(T_j^-). \quad (14)$$

**Remark:** We will show that a multiplicity preserving triangular sets decomposition exists for systems with two polynomials in the rest of the paper. Note that for a zero-dimensional polynomial system, the existence of (14) is obvious. The existence of (14) for general case (dimension mixed, more polynomials) is our future work.

The following is a key result of the paper.

**Theorem 3.8** *Let  $f_1, f_2 \in K[x_1, \dots, x_n, x]$  such that  $\gcd(f_1, f_2) = 1$  and  $\text{Cont}(f_i, x) = 1, i = 1, 2$ . Then*

$$\mathbb{M}(f_1, f_2) = \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k \mathbb{M}(g_i, f_{i+1}) - \sum_{i=1}^k \mathbb{M}(m_i, f_{i+1}), \quad (15)$$

where  $f_i, g_i, m_i$  are defined in Corollary 2.5.

**Proof.** From (6), we have

$$\mathbb{M}(m_i f_i, f_{i+1}) = \mathbb{M}(f_{i+1}, g_i f_{i+2}).$$

Then by Corollary 3.5, for  $1 \leq i \leq k$ , we have

$$\mathbb{M}(m_i, f_{i+1}) + \mathbb{M}(f_i, f_{i+1}) = \mathbb{M}(g_i, f_{i+1}) + \mathbb{M}(f_{i+1}, f_{i+2}), \quad (16)$$

$$\mathbb{M}(f_i, f_{i+1}) = \mathbb{M}(f_{i+1}, f_{i+2}) + \mathbb{M}(g_i, f_{i+1}) - \mathbb{M}(m_i, f_{i+1}). \quad (17)$$

So we have

$$\begin{aligned} \mathbb{M}(f_1, f_2) &= \mathbb{M}(f_2, f_3) + \mathbb{M}(g_1, f_2) - \mathbb{M}(m_1, f_2) \\ &= \mathbb{M}(f_3, f_4) + \mathbb{M}(g_1, f_2) + \mathbb{M}(g_2, f_3) - \mathbb{M}(m_1, f_2) - \mathbb{M}(m_2, f_3) \\ &\quad \dots \\ &= \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k \mathbb{M}(g_i, f_{i+1}) - \sum_{i=1}^k \mathbb{M}(m_i, f_{i+1}). \blacksquare \end{aligned}$$

**Remark:** The decomposition is about the  $(n-2)$ -dimensional component of  $\mathbb{M}(f_1, f_2)$ .

**Corollary 3.9** *Use the notations as Corollary 2.6, we have*

$$\begin{aligned} \mathbb{M}(f_1, f_2) &= \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k (\mathbb{M}(p_i, f_{i+1}) - \mathbb{M}(w_i, f_{i+1})) \\ &\quad - \sum_{i=1}^{k-1} (\mathbb{M}(m_i, \frac{m_{i-1}}{w_i}) + \mathbb{M}(m_i, p_i) - \mathbb{M}(m_i, q_i)) - \mathbb{M}(m_k, f_{k+1}). \quad (18) \end{aligned}$$

**Proof.** From Corollary 2.6, we have  $w_i g_i = m_{i-1} p_i$ . So by (12) and Corollary 3.5, we have

$$\mathbb{M}(g_i, f_{i+1}) = \mathbb{M}(p_i, f_{i+1}) + \mathbb{M}(m_{i-1}, f_{i+1}) - \mathbb{M}(w_i, f_{i+1}). \quad (19)$$

This helps us simplifying the computation. By (17), we have

$$\begin{aligned} \mathbb{M}(f_1, f_2) &= \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k (\mathbb{M}(p_i, f_{i+1}) + \mathbb{M}(m_{i-1}, f_{i+1}) - \mathbb{M}(w_i, f_{i+1})) - \sum_{i=1}^k \mathbb{M}(m_i, f_{i+1}) \\ &= \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k \mathbb{M}(p_i, f_{i+1}) - \sum_{i=1}^k \mathbb{M}(w_i, f_{i+1}) + \mathbb{M}(m_0, f_2) + \sum_{i=1}^{k-1} \mathbb{M}(m_i, f_{i+2}) \\ &\quad - \sum_{i=1}^{k-1} \mathbb{M}(m_i, f_{i+1}) - \mathbb{M}(m_k, f_{k+1}). \quad (20) \end{aligned}$$

From

$$m_i f_i + q_i f_{i+1} = \frac{m_{i-1}}{w_i} p_i f_{i+2},$$

we have

$$\mathbb{M}(m_i, q_i f_{i+1}) = \mathbb{M}(m_i, \frac{m_{i-1}}{w_i} p_i f_{i+2}).$$

By (12) and Corollary 3.5, we have

$$\mathbb{M}(m_i, f_{i+1}) = \mathbb{M}(m_i, f_{i+2}) + \mathbb{M}(m_i, \frac{m_{i-1}}{w_i}) + \mathbb{M}(m_i, p_i) - \mathbb{M}(m_i, q_i).$$

And  $m_0 = 1$ , so  $\mathbb{M}(m_0, f_2) = \emptyset$ . Then we have (18). ■

**Remark:** The components  $\mathbb{M}(m_i, \frac{m_{i-1}}{w_i})$ ,  $\mathbb{M}(m_i, p_i)$ , and  $\mathbb{M}(m_i, q_i)$  only involve polynomials in  $K[x_1, \dots, x_n]$ . Note that by Lemma 2.4, the coefficient of  $q_i$  in  $x^t$  for  $t > 0$  is zero when  $m_i = 0$ . These components can also be decomposed into triangular sets recursively. This corollary is very important since it provides a method to eliminate the main variable  $x$  in  $f_i$ 's, which simplifies the decomposition. We can obtain another interesting phenomenon from simple observation, that is, the degree of all the resulting polynomials is bounded by the square of the degree of  $f_1, f_2$ .

**Example 3.10** Consider the system  $[f_1, f_2] = [x^2 + y^2 + z^3 - 1, xz^2 - zy + 1]$  with Corollary 3.9 under the variable order  $x \prec y \prec z$ ,

$$f_3 = x^4 + x^2 y^2 - x^2 - xz - y + y^2 z,$$

$$f_4 = 1 - 3x^3 y - 3xy^3 + 3xy + x^2 y^3 + y^5 - y^3 + x^7 + 2x^5 y^2 - 2x^5 + x^3 y^4 - 2x^3 y^2 + x^3.$$

$m_1 = x^2, q_1 = xz + y, m_0 = w_1 = p_1 = 1, m_2 = (-x + y^2)^2, q_2 = -x^2 z + xy^2 z + 2xy - y^3 - x^5 - x^3 y^2 + x^3. w_2 = p_2 = 1$ . By Corollary 3.9, we have the following decomposition.

$$\mathbb{M}(f_1, f_2) = \mathbb{M}(f_4, f_3) + \mathbb{M}(m_1, q_1) - \mathbb{M}(m_2, f_3),$$

where  $\mathbb{M}(m_1, q_1) = 2\mathbb{M}(x, y)$  and  $\mathbb{M}(m_2, f_3) = 2\mathbb{M}(x, y) + 2\mathbb{M}(x - 1, y - 1) + 2\mathbb{M}(h_1, h_2)$ , where  $h_1 = x^6 + 3x^5 + 2x^4 + x^2 + x + 1, h_2 = y - x^4 - x^3 + x^2$ . Thus we have

$$\mathbb{M}(f_1, f_2) = \mathbb{M}(f_4, f_3) - 2\mathbb{M}(x - 1, y - 1) - 2\mathbb{M}(h_1, h_2).$$

Note that the component with negative multiplicity cannot be removed if using triangular form.

## 4 Multiplicity Preserving Decomposition for System of Two Bivariate Polynomials

In this section, we will consider the triangular decomposition of a zero-dimensional bivariate polynomial system with two polynomials, that is,  $\Sigma = \{f, g\} \subset K[x, y]$ . The method provided here is complete for a zero dimensional bivariate polynomial system with two polynomials. When  $\Sigma$  is zero dimensional,  $\mathbb{M}(\Sigma)$  defines a multiplicative-zero set.

## 4.1 Algorithm

**Lemma 4.1** *Using the similar notations as Corollary 2.6, if  $\{f_1, f_2\}$  is zero-dimensional, we have*

$$\mathbb{M}(f_1, f_2) = \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k \mathbb{M}(p_i, f_{i+1}) - \sum_{i=1}^k \mathbb{M}(w_i, f_{i+1}) - \mathbb{M}(m_k, f_{k+1}). \quad (21)$$

**Proof.** The lemma is a consequence of Corollary 3.9. Note that  $\mathbb{M}(m_i, \frac{m_i-1}{w_i}) = \mathbb{M}(m_i, p_i) = \mathbb{M}(m_i, q_i) = \emptyset$  since  $\gcd(m_i, p_i)$ ,  $\gcd(m_i, \frac{m_i-1}{w_i})$ ,  $\gcd(m_i, q_i)$  are constant.

The following corollary is useful.

**Corollary 4.2** *If  $w_i (1 \leq i \leq k)$  are constants and  $f_{k+1} = l_1(x)y^t + l_0(x)$  for  $t > 0$  and  $l_0, l_1 \in K[x]$  in (21), we have*

$$\mathbb{M}(f_1, f_2) = \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k \mathbb{M}(p_i, f_{i+1}). \quad (22)$$

Furthermore, if  $p_i (1 \leq i \leq k)$  are constant, we have

$$\mathbb{M}(f_1, f_2) = \mathbb{M}(f_{k+1}, f_{k+2}). \quad (23)$$

**Proof.**

Since  $f_{k+1} = l_1(x)y^t + l_0(x)$  and  $\text{Cont}(f_{k+1}, y) = 1$ ,  $\mathbb{M}(m_k, f_{k+1}) = \emptyset$ . Note that  $m_k$  is a factor of  $l_1^n$  for some positive integer  $n$ . Thus  $\mathbb{M}(m_0, f_2) = \emptyset$ ,  $\mathbb{M}(m_k, f_{k+1}) = \emptyset$ . So from (21), we have (22). If  $p_i = 1$ , (23) is a consequence of (22).  $\blacksquare$

Now we will consider the complexity of our method under the condition of Corollary 4.2. We consider this case because it is usually the case for almost all zero-dimensional bivariate polynomial system with two polynomials. So the result is interesting. At first, we need to introduce some notations, which can be found in [7]. Let  $\mathcal{L}(f)$  bound the bitsize of the coefficients of  $f \in K[x, y]$  (including a bit for the sign). We assume  $\lg(\deg(f)) = \mathcal{O}(\mathcal{L}(f))$ . For  $a \in \mathbb{Q}$ ,  $\mathcal{L}(a)$  is the maximum bitsize of  $a$ 's numerator and denominator. Let  $M(\tau)$  denote the bit complexity of multiplying two integers of size  $\tau$ , and  $M(d, \tau)$  the complexity of multiplying two univariate polynomials of degrees  $\leq d$  and coefficient bitsize  $\leq \tau$ . Using FFT,  $M(\tau) = \tilde{\mathcal{O}}_B(\tau)$  and  $M(d, \tau) = \tilde{\mathcal{O}}_B(d\tau)$ .

**Lemma 4.3** [7, 19] *Let  $f, g \in \mathbb{Z}[x]$ ,  $\deg(f), \deg(g) \leq n$ , and  $\mathcal{L}(f), \mathcal{L}(g) \leq \tau$ . We can compute  $\gcd(f, g)$  in  $\tilde{\mathcal{O}}_B(n^2\tau)$ .*

**Lemma 4.4** [7, 19] *Let  $f, g \in \mathbb{Z}[x, y]$ ,  $\deg(f), \deg(g) \leq n$ , and  $\mathcal{L}(f), \mathcal{L}(g) \leq \tau$ . We can compute the subresultant sequence of  $f$  and  $g$  in  $\tilde{\mathcal{O}}_B(n^6\tau)$ .*

**Theorem 4.5** *If  $w_i (1 \leq i \leq k)$  are constant and  $f_{k+1} = l_1(x)y^t + l_0(x)$  for some positive integer  $t$ ,  $l_0, l_1 \in K[x]$  in (21), and  $K = \mathbb{Z}$ , we can decompose a zero-dimensional bivariate system with two polynomials into multiplicity preserving triangular sets in  $\tilde{\mathcal{O}}_B(n^7\tau)$ .*

**Proof.** We can compute a subresultant sequence of  $f$  and  $g$  at first. It can be computed in  $\tilde{\mathcal{O}}_B(n^6\tau)$  by Lemma 4.4. Then we simplify each pseudo-division step to derive (7) from the highest degree of the sequence in  $y$  to the lowest degree. Let  $\{F_1, \dots, F_{k+2}\}$  be the subresultant sequence of  $f$  and  $g$ . We need only consider the case of regular subresultant sequence since the complexity of the regular case also bounds the degenerate case. Consider the formula

$$l_{i+1}^2 F_i + Q_i F_{i+1} = l_i^2 F_{i+2}. \quad (24)$$

Assume that we have computed the contents of  $F_i$  and  $F_{i+1}$ , say  $r_i, r_{i+1}$ .  $F_1, F_2$  are  $f_1, f_2$ . And the contents of  $f$  or  $g$  can be computed in  $\tilde{\mathcal{O}}_B(n^3\tau)$  by Lemma 4.3, which can be ignored comparing to  $\tilde{\mathcal{O}}_B(n^6\tau)$ . For each  $F_i$ , it is well known that  $\deg(F_i) \leq n^2, \mathcal{L}(F_i) = \mathcal{O}(n\tau)$  (for reference see [7]). And  $\deg(F_i, y) \leq n-1$  for  $i \geq 3$ . Thus for any coefficient of  $F_i$ , say  $h \in \mathbb{Z}[x]$ , we have  $\deg(h) \leq n^2, \mathcal{L}(h) = \mathcal{O}(n\tau)$ . So to compute the content of  $F_i$  with  $\deg(F_i, y) = h$ , we need to compute at most  $h$  gcd each in  $\tilde{\mathcal{O}}_B(n^5\tau)$ . Let  $r_{i+2} = \text{Cont}(F_{i+2}, y)$ . In order to derive (7), we need to delete  $\gcd(l_{i+1}^2 F_i, Q_i F_{i+1}, l_i^2 F_{i+2})$  from the two side of (24). So we need to bound  $\gcd(l_{i+1}^2 r_i, l_i^2 r_{i+2})$ . Note that  $\deg(s) \leq n^2, \mathcal{L}(s) = \mathcal{O}(n\tau)$  holds for  $s = l_i$  or  $s = l_{i+1}$  and we can not optimize the degree of  $l_{k+1}$ . But we can compute  $r = \gcd(l_i, l_{i+1})$ , which is bounded by  $\tilde{\mathcal{O}}_B(n^5\tau)$ . And  $\gcd((\frac{l_i}{r})^2, r_{i+2})$  can be bounded by  $2\tilde{\mathcal{O}}_B(n^5\tau)$  as below. We can compute  $w = \gcd((\frac{l_i}{r}), r_{i+2})$ , and then  $\gcd((\frac{l_i}{r}), \frac{r_{i+2}}{w})$ .  $\gcd((\frac{l_{i+1}}{r})^2, r_i)$  is also bounded by  $2\tilde{\mathcal{O}}_B(n^5\tau)$ . So to obtain (7) from (24) for  $\deg(F_i, y) = k$ , we need  $(k+5)\tilde{\mathcal{O}}_B(n^5\tau)$ . Then we can decide  $m_i, g_i$  in (5) by two divisions. Since  $w_i$ 's are constant,  $m_{i-1} | g_i$ . Thus, we can obtain  $p_i$  by one division. When  $h$  changes from  $n$  to 1, we can bound it by  $\frac{n^2+9n}{2}\tilde{\mathcal{O}}_B(n^5\tau)$ , that is,  $\tilde{\mathcal{O}}_B(n^7\tau)$ . Then the total complexity is  $\tilde{\mathcal{O}}_B(n^7\tau)$ . ■

**Remark:** For many  $f, g \in K[x, y]$ , the last two elements of the subresultant sequence  $F_{k+1}, F_{k+2}$  form a multiplicity preserving triangular decomposition of  $f, g$ . Thus, we can compute the decomposition in  $\tilde{\mathcal{O}}_B(n^6\tau)$ .

The lemma gives a multiplicity preserving triangular decomposition of a bivariate polynomial system. But there exist some triangular sets with negative multiplicities. The following results gives an algorithm to remove the triangular sets with negative multiplicities.

**Theorem 4.6** *There exists an algorithm to decompose a zero-dimensional bivariate polynomial system  $\{f_1, f_2\} \subset K[x, y]$  into a set of triangular sets, such that*

$$\mathbb{M}(f_1, f_2) = \sum_{i=1}^N \mathbb{M}(g_i, h_i), \quad (25)$$

where  $g_i \in K[x], h_i \in K[x, y]$ .

**Proof.** In the case  $f_1 = h_1 f'_1, f_2 = h_2 f'_2$  having factors in  $K[x]$  but  $\gcd(f_1, f_2) = 1$ , where  $h_i = \text{Cont}(f_i, y), i = 1, 2$ , we have

$$\mathbb{M}(f_1, f_2) = \mathbb{M}(h_1, f'_1) + \mathbb{M}(f'_1, h_2) + \mathbb{M}(f'_1, f'_2). \quad (26)$$

Since  $\gcd(h_1 f'_1, h_2 f'_2) = 1, \mathbb{M}(h_1, h_2) = \emptyset$ .

By (26), we can assume that  $\text{Cont}(f_i, y) = 1, i = 1, 2$ . From Lemma 4.1, we have (21). We put the triangular sets on the righthand side of (21) into two sets:  $W_1 = \mathbb{M}(f_{k+1}, f_{k+2}) + \sum_{i=1}^k \mathbb{M}(p_i, f_{i+1}), W_2 = \sum_{i=1}^k \mathbb{M}(w_i, f_{i+1}) + \mathbb{M}(m_k, f_{k+1})$ . It is clear that  $W_2 \subset W_1$ . Our aim is to delete the multiplicative-zeros of  $W_2$  from  $W_1$  to derive a group of triangular sets with positive coefficients.

Take any triangular set  $U = (u_1(x), u_2(x, y))$  out of  $W_2$ , we can compute gcd of  $u_1(x)$  and some  $v_1(x)$ , where  $V = (v_1(x), v_2(x, y))$  is one triangular set of  $W_1$ . Denote the gcd as  $p(x)$ . Then decomposing  $U$ , we have  $(p(x), u_2(x, y))$ , and put  $(\frac{u_1(x)}{p(x)}, u_2(x, y))$  into  $W_2$  again. Decomposing  $V$ , we have  $(p(x), v_2(x, y))$ . And put  $(\frac{v_1(x)}{p(x)}, v_2(x, y))$  into  $W_1$ . We can compute the gcd, say  $w(x, y)$ , of  $u_2(x, y)$  and  $v_2(x, y)$  modulo  $p(x)$ , then remove the factor  $w(x, y)$  from  $u_2(x, y)$  ( $v_2(x, y)$ ) and put the left part into  $U(V)$ . We can also use the method in [21] to decompose  $(p(x), u_2(x, y))$  and  $(p(x), v_2(x, y))$  into irreducible and regular triangular sets, and then we can easily decide that whether two irreducible and regular triangular sets have same zero set or not. Thus we can remove the triangular sets in  $W_2$ . In the end, we can remove all the zero sets in  $W_2$ . We prove the theorem.  $\blacksquare$

We will give a multiplicity preserving algorithm to decompose a bivariate polynomial system into triangular sets based on the theory above.

**Algorithm 4.7** *Input:* a zero-dimensional bivariate polynomial system  $\mathcal{P}_1 = \{f_1(x, y), f_2(x, y)\} \in K[x, y]$ , and  $d_1 = \deg(f_1, y) \geq d_2 = \deg(f_2, y)$ . *Output:* a group of triangular sets  $\mathcal{P} = \{[g_i(x), h_i(x, y)], i = 1, \dots, n\}$  such that  $\mathbb{M}(f_1, f_2) = \sum_{i=1}^n \mathbb{M}(f_i, h_i)$ .

1.  $\mathbb{M}_p = \emptyset, \mathbb{M}_n = \emptyset$ .
2. Compute  $h_i = \text{Cont}(f_i, y), i = 1, 2$ . Let  $f_i = f_i/h_i$ .  $\mathbb{M}_p = \mathbb{M}_p \cup \{[h_1, f_2], [h_2, f_1]\}$ .
3. Let  $m_0 = 1$ . While  $\deg(f_2, y) > 0$ , do
  - By pseudo-division, we have  $m_1 f_1 + q_1 f_2 = f_3$  and  $h = \text{Cont}(f_3, x)$ .  $f_3 = \frac{f_3}{h}, v = \gcd(m_1, h), m_1 = \frac{m_1}{v}, h = \frac{h}{v}$ . Let  $q = \gcd(m_0, h), w = \frac{m_0}{q}, p = \frac{h}{q}$ . If  $p$  is not a constant,  $\mathbb{M}_p = \mathbb{M}_p \cup \{[p, f_2]\}$ . If  $w$  is not a constant,  $\mathbb{M}_n = \mathbb{M}_n \cup \{[w, f_2]\}$ .
  - $f_1 = f_2, f_2 = f_3, m_0 = m_1$ .

If  $\deg(f_1, y) > 1$ ,  $\mathbb{M}_n = \mathbb{M}_n \cup \{[m_0, f_1]\}$ .

4. If  $\mathbb{M}_n$  is not empty, following the method in the proof of Theorem 4.6, we can remove the multiplicative varieties in  $\mathbb{M}_n$  from  $\mathbb{M}_p$ . Thus we obtain a group of triangular sets as (25).

**Proof.** The termination of the algorithm is clear since the degree of  $f_1$  and  $f_2$  is finite. The correctness of the algorithm is guaranteed by Theorem 4.6 and Lemma 4.1.

**Example 4.8** Let  $\mathbb{C}$  be the curve defined by

$$f = 2y^4 - 3y^2x + x^2 - 2x^3 + x^4.$$

We will compute the  $y$ -critical points ( $f = \frac{\partial f}{\partial y} = 0$ ) of  $\mathbb{C}$ .

$$f_y = \frac{\partial f}{\partial y} = 8y^3 - 6yx.$$

Delete the content 2,  $f_y = \frac{f_y}{2}$ . In the following, we will solve the system  $\Sigma = \{f, f_y\}$ . Following our algorithm, we have

$$m_1 f + q_1 f_y = p_1 f_3,$$

where  $m_1 = 4, p_1 = x, q_1 = -y, f_3 = -3y^2 + 2x - 4x^2 + 2x^3$ .

$$m_2 f_y + q_2 f_3 = p_2 f_4,$$

where  $m_2 = 3, q_2 = 4y, p_2 = x(-1 - 16x + 8x^2), f_4 = y$ . Note that here  $w_2 = 4$ . We ignore it since it is a constant.

$$m_3 f_3 + q_3 f_4 = f_5,$$

where  $m_3 = 1, q_3 = 3y, f_5 = 2x(x-1)^2$ . Similarly, we ignore  $w_3 = 3$ . With Corollary 4.2, we have

$$\mathbb{M}(f, f_y) = \mathbb{M}(f_5, f_4) + \mathbb{M}(p_1, f_y) + \mathbb{M}(p_2, f_3).$$

We can find that

$$\begin{aligned} \mathbb{M}(f_5, f_4) &= \mathbb{M}(x, y) + 2\mathbb{M}(x-1, y), \\ \mathbb{M}(p_1, f_y) &= \mathbb{M}(x, 4y^3 - 3yx) = 3\mathbb{M}(x, y), \\ \mathbb{M}(p_2, f_3) &= 2\mathbb{M}(x, y) + \mathbb{M}(-1 - 16x + 8x^2, -4y^2 + 3x). \end{aligned}$$

We find that  $\mathbb{M}(x, y)$  and  $\mathbb{M}(x-1, y)$  are zeros with multiplicities 6, 2, respectively. And the other zeros are with multiplicities 1.

degree	[5,4]	[7, 5]	[9, 7]	[13, 11]	[23, 21]	[33,31]
MPTD	0.006	0.019	0.105	1.363	62.894	884.577
BMT	0.014	0.024	0.036	0.077	0.280	0.848
RC	0.194	0.343	0.810	4.520	127.194	1075.653
CS	0.082	0.880	25.101	-	-	-
WS	0.134	3.881	410.399	-	-	-

Table 1: Timings for different methods

## 4.2 Implementation and Comparison

We implement our algorithm in Maple. We compare the computing time of our methods with several other related methods. One is the regular chains method [6, 15] (package RegularChains in Maple 13, including two functions), one is Characteristic set method in Epsilon[23], the other is a package wsolve (see [26]). All the results are collected on a PC with a 3.2GHz CPU, 2.00G memory, and running Microsoft Windows XP. We use Maple 13 in the experiments.

We run 100 examples in each case and compute their average computing time in Table 1. We take random dense polynomials with coefficients bounded by  $[-100, 100]$  for each example. Table 1 is the timings for the given methods in seconds. Here MPTD means the method provided in this paper, RC (BMT) means regular chains method of function “Triangularize” (“BivariateModularTriangularize”), CS means characteristic set method (“charsets”) in Epsilon, and WS means wsolve method. The first row is  $[\deg(f_1), \deg(f_2)]$ . The first column represents the methods. “-” means out of memory or we did not test it.

We need to mention that only MPTD can compute the multiplicities of the zeros of the bivariate polynomial system. BMT and RC are implemented by C code but other’s are in Maple. MPTD and BMT are only for bivariate polynomial system but other methods work for general system. Note that with a mirror modification, MPTD can multiplicity preserving triangular decompose system with two multivariate polynomials.

We can conclude from the table that MPTD and BMT are always faster than RC, CS and WS. For the system with low degrees, CS, WS are a little faster than RC, but they both are very slow when the degree of the system more than 10. MPTD is always a little faster than RC. MPTD is a little faster than BMT for low degree system, but much slower than BMT for systems with high degrees. There are several reasons: BMT is in C, using modular method and using FFT based arithmetic, but MPTD does not use these techniques.

## 5 Conclusion

We present an algorithm to decompose a polynomial system with two polynomials into triangular sets. Different from the existing methods for triangular decomposition, our method

preserves the multiplicity of the zeros or components of the systems. We implement the method for bivariate polynomial systems. We will extend the method to the systems with more polynomials in the future.

## 6 Acknowledgement

The work is partially supported by National Key Basic Research Project of China and China-France cooperation project EXACTA. The first author is partially supported by NSFC Grant 11001258.

## References

- [1] P. Aubry, D. Lazard, M. Moreno Maza. On the theories of triangular sets. *J. Symb. Comput.*, 28(1-2): 105-124, 1999.
- [2] W.S. Brown. The subresultant PRS algorithm. *ACM Trans. on Mathematical Software*, 4: 237-249, 1978.
- [3] C. Chen, O. Golubitsky, F. Lemaire, M. Moreno Maza, W. Pan. Comprehensive triangular decomposition. *CASC*: 73-101, 2007.
- [4] S.C. Chou and X.S. Gao, Ritt-Wu's decomposition algorithm and geometry theorem proving, *CADE'10*, M.E. Stickel (Ed.), 207-220, LNCS449, Springer-Verlag, 1990.
- [5] D. A. Cox, J. Little, D. O'Shea. Using algebraic geometry. *Springer, Second Edition*, 2004.
- [6] X. Dahan, M. Moreno Maza, É. Schost, W. Wu, and Y. Xie. Lifting techniques for triangular decompositions. *Proc. ISSAC 2005*. Beijing, 2005.
- [7] D. I. Diochnos, I. Z. Emiris, E. P. Tsigaridas. On the asymptotic and practical complexity of solving bivariate systems over the reals, *J. Symb. Comput.* 44: 818-835, 2009.
- [8] X. S. Gao, S.C. Chou. On the dimension of an arbitrary ascending chain. *Chinese Sci. Bull.*, 38: 799-804, 1993.
- [9] W. V. D. Hodge, D. Pedoe. *Methods of algebraic geometry, Volume II*, University Press Cambridge, ISBN 0 521 46901 5 paperback, 1994.
- [10] D. Lazard. A new method for solving algebraic systems of positive dimension. *Discrete Appl. Math.*, 33: 147-160, 1991.

- [11] D. Lazard, Solving zero-dimensional algebraic systems. *J. Symb. Comput.* 13: 117-131, 1992.
- [12] B.H. Li: A method to solve algebraic equations up to multiplicities via Ritt- Wu's characteristic sets. *Acta Analysis Functionalis Applicata*, 5(3): 98-109, 2003.
- [13] X. Li, M. Moreno Maza, É. Schost: Fast arithmetic for triangular sets: from theory to practice. *Proc. ISSAC*, 269-276, 2007.
- [14] Y. Li, B. Xia, Z. Zhang, Zero decomposition with multiplicity of zero-dimensional polynomial systems (in Chinese), The Third Computer Mathematics Conference of China, Shanghai, China, October, 19-22, 2010.
- [15] M. Moreno Maza. On triangular decompositions of algebraic varieties. *MEGA-2000 Conference*, Bath, England.
- [16] M. Kalkbrener. A generalized euclidean algorithm for computing triangular representations of algebraic varieties. *J. Symb. Comput.* 15(2): 143-167, 1993.
- [17] M. Kalkbrener. Primitive polynomial remainder sequence in elimination theory. *Applicable Algebra in Engineering, Communication and Computing*, 6: 65-79, 1995.
- [18] M. Kalkbrener. Algorithmic properties of polynomial rings. *J. Symb. Comput.* 26(5): 525-581, 1998.
- [19] D. Reischert. Asymptotically fast computation of subresultants. *Proc. ISSAC 1997*, 233-240, 1997.
- [20] J. Ritt. Differential algebra. New York, Dover Publications, 1966.
- [21] Y. Sun, D.K. Wang. An efficient algorithm for factoring polynomials over algebraic extension field. arXiv:0907.2300v2 [cs.SC]
- [22] D. Wang. Computing triangular systems and regular systems. *J. Symb. Comput.* 30(2): 221-236, 2000.
- [23] D. Wang. Elimination Practice, Software Tools and Applications. Imperial College Press, 2004.
- [24] Wu, W.T., Basic Principles of Mechanical Theorem-proving in Elementary Geometries, *Journal Automated Reasoning*, 2, 221-252, 1986.
- [25] W.T. Wu, *Basic Principle of Mechanical Theorem Proving in Geometries*, (in Chinese) Science Press, Beijing, 1984; Springer, Wien, 1994.

- [26] D.K. Wang. Zero decomposition for system of polynomial equations. *Proc. ASCM 2000*. World Scientific, 67-70, 2000.
- [27] L.Yang, J. Zhang. Searching dependency between algebraic equations: an algorithm applied to automated reasoning. *Artificial Intelligence in Mathematics*, 147-156, Oxford University Press, 1994.