Decomposition of Differential Polynomials with Constant Coefficients*

Xiao-Shan Gao and Mingbo Zhang
Key Laboratory of Mathematics Mechanization
Institute of Systems Science, AMSS,
Academia Sinica, Beijing 100080, China
(xgao,mzhang)@mmrc.iss.ac.cn

ABSTRACT

In this paper, we present an algorithm to decompose differential polynomials in one variable and with rational number as coefficients. Besides arithmetic operations, the algorithm needs only factorization of multi-variable polynomials and solution of linear equation systems. Experimental results show that our method is quite efficient.

Categories and Subject Descriptors

I.1.2 [**SYMBOLIC AND ALGEBRAIC MANIPULATION**]: Algorithms —*Algebraic algorithms*

General Terms

Algorithm

Keywords

differential polynomial, decomposition, pseudo linear, differential degree

1. INTRODUCTION AND PRELIMINARIES

The study on functional decomposition started with the decomposition of univariate polynomials (*pols*). The first decomposition algorithms were presented by Barton-Zippel [3] and Alagar-Tanh [1]. This was followed by the work of Kozen and Landau [16] who proposed a *pol* time decomposition algorithm. Similar algorithms were given by Gutierrez [12]. Gathen proposed an algorithm of better complexity and a parallel algorithm in [9]. In [29], Zippel presented a *pol* time algorithm to decompose a given univariate rational function over an arbitrary field.

Decomposing linear ordinary differential equations (LODEs) was first discussed by Singer in [19]. The algorithms to decompose LODEs proposed by Bronstein, Petkovšek, and Schwarz [4, 5, 18] were based on the classical work of Beke. Another approach based on the eigenring of the LODE was proposed by van der Put and Singer [28]. In [26, 27], van Hoeij proposed an algorithm to decompose an LODE with rational functions as well as power series coefficients, which was extended to decompose LODEs over an exponential extension of a base field by Fredet [7]. In [23], Tsarev presented an algorithm for complete enumeration of all factorizations of an LODE. In [2, 6], Barkatou-Pflügel and Cluzeau proposed decomposition algorithms for LODE systems. In [10], Giesbrecht and Zhang presented methods to find decomposition for the more general Ore *pols*. In [11], Grigor'ev gave an algorithm to decompose LODEs and showed that the worst case complexity of the algorithm is exponential. Applications of the decomposition of LODEs to compute closed form solutions and to determine the Galois group were discussed by Singer and Ulmer [19, 21]. In [13], Li et al gave the first algorithm to decompose systems of linear PDEs with finite-dimensional solution spaces.

The problem of decomposing non-linear differential polynomials (*d-pols*) was discussed in the classic work by Königsberger [15]. Methods for some special classes of *d-pols* were given in [22, 25]. In [24], Tsarev considered *d-pols* of the form $y_n - R(x, y, \ldots, y_{n-1})$ and gave a decision procedure for the existence of factorization. As far as we know, there exist no effective methods to decompose non-linear *d-pols*.

In this paper, we give an algorithm for decomposing *d-pols* in one variable and with constant coefficients. The algorithm to find a decomposition $f = g \circ h$ for a given f consists of three main steps. First, we try to find a decomposition where g is a univariate *pol*. Second, the problem is reduced to the case where g is pseudo linear. Third, the pseudo linear case is solved.

Besides arithmetic operations, the algorithm needs only factorization of multi-variable *pols* and solution of linear equation systems, both of which have *pol*-time algorithms. Our algorithm is exponential in the worst case due to the need to select all possible left decomposition factors of an LODE. We implement our algorithm in Maple and experimental results show that our method is quite efficient in handling large *d-pols*. We use examples to show that the algorithm can be used to simplify the solution of non-linear differential equations. We also discuss shortly how to extend the algorithm in the paper to the case when the coefficients of the *d-pols* are rational functions.

* Partially supported by a National Key Basic Research Project of China and by a USA NSF grant CCR-0201253.

ISSAC'04, July 4–7, 2004, Santander, Spain.

2. PRELIMINARY RESULTS

Let $\mathbb{Q}$ be the field of rational numbers, $\mathbb{Q}(t)$ the field of rational functions in t, x a differential indeterminate, $\mathbb{Q}(t)\{x\}$ the ordinary *d-pol* ring over $\mathbb{Q}(t)$ with differentiation $\frac{\partial}{\partial t}$ [14]. An element in $\mathbb{Q}(t)\{x\}$ is called a *d-pol*. We denote by x_i the i-th derivative of x. For a *d-pol* f , let x_o be the highest derivative appearing in f. Then o is called the *order* of f and is denoted by o_f. Let o_f be the order of f, d_f the degree of f in x_{o_f}, i_f the coefficient of $x_{o_f}^{d_f}$ in f, $s_f = \frac{\partial f}{\partial x_{o_f}}$. d_f, i_f, s_f are called the *degree, initial, and separant* of f respectively.

For a *d-pol* f and a non-negative integer k, let $f_{(k)}$ be the k-th derivative of f. Then for $k > 0$, we have

$$f_{(k)} = s_f x_{o_f+k} + R_f \tag{1}$$

where R_f is a *d-pol* of lower order in x than $o_f + k$.

A monomial in $\mathbb{Q}(t)\{x\}$ is always arranged in the form $a\prod_{i=1}^{r} x_{\beta_i}^{\alpha_i}$ where $a \in \mathbb{Q}(t)$, $\alpha_i \in \mathbb{N}^+$, $\beta_1 > \ldots > \beta_r \geq 0$. The number $\sum_{i=1}^{r} \alpha_i$ is called the *total degree* and $\sum_{i=1}^{r} \alpha_i \cdot \beta_i$ is called the *differential degree* of the monomial. The largest total degree and differential degree for all monomials in a *d-pol* f is called the *total degree* and *differential degree* of f, denoted by $\mathrm{tdeg}(f)$ and $\mathrm{ddeg}(f)$ respectively.

Let q be a *d-pol*. If the total (differential) degrees of the monomials in q are all equal, q is called *total* (*differential*) *degree homogeneous*. If $\mathrm{tdeg}(q) = k$ and q is total degree homogeneous, q is called k-total degree homogeneous. Furthermore, if $k = 1$ we say that q is *linear*.

We may define a rank between two monomials according to the pure lexicographical order induced by the variable order $x < x_1 < x_2 < \ldots$. In a *d-pol* p, the term with the highest rank is called the *leading term* of p.

In the rest of this paper, we will assume that *d-pols* are with coefficients in $\mathbb{Q}$ unless mentioned otherwise. Let $g, h \in \mathbb{Q}\{x\}$, we use $g \circ h$ to denote the *composition* of two *d-pols* g and h, which is obtained by substituting x_i in g by $h_{(i)}$. If $f = g \circ h$, g, h are called the *left* and *right decomposition factor* of f respectively. A *decomposition* $f = g \circ h$ is called *non-trivial*, if both g and h are not in the form $ax + b (a, b \in \mathbb{Q})$.

LEMMA 1. *The composition operation is associate:* $f \circ (g \circ h) = (f \circ g) \circ h$.

Proof. Omitted. See the preprint [8]. ∎

For $c \in \mathbb{Q}$, we have $f = g \circ h = [g \circ (x + c)] \circ [(x - c) \circ h]$. Then we may always assume that h has no term in $\mathbb{Q}$. In this case, the constant term of f is the same as that of g. So we may further assume that f and g have no constant terms.

In what follows, we assume that f is a d-pol *in $\mathbb{Q}\{x\}$ of positive order and has no term in $\mathbb{Q}$. We will find a non-trivial decomposition $f = g \circ h$.* We may write f, g, h as follows.

$$f = i_f x_{o_f}^{d_f} + f_1, g = i_g x_{o_g}^{d_g} + g_1, h = i_h x_{o_h}^{d_h} + h_1 \tag{2}$$

where f_1, g_1, h_1 are of lower degree in $x_{o_f}, x_{o_g}, x_{o_h}$ than d_f, d_g, d_h respectively.

LEMMA 2. *If $f = g \circ h$, we have*

$$o_f = o_g + o_h, f_{(1)} = g_{(1)} \circ h, s_f = (s_g \circ h) \cdot s_h. \tag{3}$$

Proof. Omitted. See the preprint [8]. ∎

LEMMA 3. *If $f = g \circ h$ and g is a univariate* pol*, we have*

$$o_f = o_h, d_f = d_g d_h, i_f = (i_g \circ h) i_h^{d_g} \tag{4}$$

$$f_{(1)} = (s_g \circ h) h_{(1)}. \tag{5}$$

Proof. Omitted. See the preprint [8]. ∎

LEMMA 4. *If $f = g \circ h$ and $o_g > 0$, we have*

$$d_f = d_g, i_f = (i_g \circ h) \cdot s_h^{d_f} \tag{6}$$

Proof. Omitted. See the preprint [8]. ∎

LEMMA 5. *q, u, v are* d-pols*, $q = u \circ v$. Let $d = tdeg(q), d_1 = tdeg(u), d_2 = tdeg(v)$, then $d = d_1 \cdot d_2$. Furthermore, if Q_d, U_{d_1}, V_{d_2} are the sums of monomials in q, u, v with total degree d, d_1, d_2 respectively, then $Q_d = U_{d_1} \circ V_{d_2}$.*

Proof. Omitted. See the preprint [8]. ∎

3. POLYNOMIAL DECOMPOSITION

We first consider a special case $o_g = 0$, called *polynomial decomposition*. This can be treated as a composition of a univariate *pol* and a multi-variable *pol* in $\mathbb{Q}[x, x_1, \ldots, x_{o_f}]$, which is considered in [9]. In what follows, we give a simple algorithm based on differentiation, which is quite efficient as shown by our experimental results.

Before presenting the algorithm, we first give two basic algorithms which will also be used in other sections.

ALGORITHM 1. *Input: d-pols $f, h \in \mathbb{Q}\{x\}$.*
Output: a d-pol $g \in \mathbb{Q}\{x\}$ such that $f = g \circ h$ if such a g exists.

S1 If $f \in \mathbb{Q}$, return $g = f$.

S2 Write f and h as (2). If $o_f < o_h$, g does not exist. The algorithm terminates.

S3 If $o_f = o_h$, from (3), $o_g = 0$. From (4), $d_g = d_f/d_h$. If d_g is not an integer, g does not exist and the algorithm terminates. Otherwise, let $t = \frac{i_f}{i_h^{d_g}}$. From (4), t should be $i_g \circ h$. Go to **S5**.

S4 If $o_f > o_h$, from (3) and (6) the order o_g and leading degree d_g for g can be computed as follows: $o_g = o_f - o_h, d_g = d_f$. Let $t = \frac{i_f}{s_h^{d_g}}$. From (6), t should be $i_g \circ h$.

S5 If t is not a *d-pol* in $\mathbb{Q}\{x\}$, g does not exist. The algorithm terminates. Otherwise, let $f_1 = f - t \cdot h_{(o_g)}^{d_g}$. Then $f_1 = g \circ h - (i_g \circ h)(x_{o_g}^{d_g} \circ h) = (g - i_g x_{o_g}^{d_g}) \circ h$, which also has a right decomposition factor h. Call Algorithm 1 with f_1, h and t, h as inputs. Let the outputs be g_1 and i_g. If i_g and g_1 exist, return $g = i_g x_{o_g}^{d_g} + g_1$. Otherwise, g does not exist.

EXAMPLE 1. *Let $f = 2x_2(x_2^2 + x_1)x_3 + x_2^4 + x_2^3 + 2x_1x_2^2 + x_1x_2 + x_1^2, h = x_2^2 + x_1$. Since $o_f = 3 > o_h = 2$, we will execute S4, where $d_g = d_f = 1, o_g = o_f - o_h = 1, t = \frac{i_f}{s_h^{d_g}} = \frac{2x_2(x_2^2+x_1)}{2x_2} = x_2^2 + x_1 = i_g \circ h$. In S5, $f_1 = f - t \cdot h_{(1)} = x_2^4 + 2x_1x_2^2 + x_1^2$. Execute Algorithm 1 with input t, h we*

have $i_g = x$. Executing Algorithm 1 with input f_1, h we have $g_1 = x^2$. So h is a right decomposition factor of f and the corresponding left decomposition factor is $g = i_g x_{o_g}^{d_g} + g_1 = xx_1 + x^2$.

ALGORITHM 2. *Input: a d-pol $p \in \mathbb{Q}\{x\}$.*
Output: a d-pol $q \in \mathbb{Q}\{x\}$ such that $q_{(1)} = p$ if such a q exists.

S1 Set $q = 0$.

S2 If $p \notin \mathbb{Q}$ then go to the next step. If $p = 0$, return q. Otherwise, q does not exist. The algorithm terminates.

S3 Let the leading term of p under the lexical order be $t = c_p \prod_{i=1}^{r} x_{\beta_i}^{\alpha_i}$ where $\beta_1 > \ldots > \beta_r \geq 0$, and $c_p \in \mathbb{Q}$.

S4 If $\alpha_1 \neq 1$ then q does not exist and terminate.

S5 If $r = 1$ or $\beta_1 > \beta_2 + 1$, let

$$s = c_p x_{\beta_1 - 1} \prod_{i=2}^{r} x_{\beta_i}^{\alpha_i}.$$

Otherwise, let

$$s = \frac{c_p}{\alpha_2 + 1} x_{\beta_1 - 1}^{\alpha_2 + 1} \prod_{i=3}^{r} x_{\beta_i}^{\alpha_i}.$$

If $t > r$ in a product $\prod_{i=t}^{r}$, the value of the product is defined to be 1.

S6 Set $q = q + s$, $p = p - s_{(1)}$. Go to S2.

To see that Algorithm 2 is correct, let the leading term of q be $t = c_q \prod_{i=1}^{r} x_{\tau_i}^{\gamma_i}$ where $\tau_1 > \ldots > \tau_r \geq 0$. The case $r = 1$ is obvious. For $r > 1$, the leading term of $q_{(1)}$ is

$$l = c_q \gamma_1 x_{\tau_1 + 1} x_{\tau_1}^{\gamma_1 - 1} \prod_{i=2}^{r} x_{\tau_i}^{\gamma_i}.$$

Since $q_{(1)} = p$, l should be the leading term of p. Then the degree of p should be one, which implies the correctness of S4. If $\gamma_1 > 1$, x_{τ_1} appears in l, which implies the first case in S5. If $\gamma_1 = 1$, x_{τ_1} does not appear in l, which implies the second case in S5.

ALGORITHM 3. *Input: a d-pol f.*
Output: a univariate pol g and a d-pol h such that $f = g \circ h$ if such g and h exist.

S1 If $f = g \circ h$ is a *pol* decomposition of f, from (5) $h_{(1)}$ is a proper factor of $f_{(1)}$ of order $o_f + 1$ and with no constant terms. Let S be the set of such primitive factors of $f_{(1)}$.

S2 For each $p \in S$, do S3 and S4.

S3 Use Algorithm 2 to find an $h \in \mathbb{Q}\{x\}$ such that $h_{(1)} = p$. If such an h does not exist, goto S2.

S4 Use Algorithm 1 to find a $g \in \mathbb{Q}[x]$ such that $f = g \circ h$. If such a g exists, return g and h; else, goto S2.

EXAMPLE 2. *Let $f = x^2 x_2^4 + (2xx_1 + x)x_2^2 + x_1^2 + x_1$. We have*

$$f_{(1)} = x_2(2xx_2^2 + 2x_1 + 1)(2xx_3 + x_1x_2 + 1)$$

The set of proper factors for $f_{(1)}$ of order $o_f + 1 = 3$ and containing no terms in $\mathbb{Q}$ is $S = \{p : p = x_2(2x_3x + x_2x_1 + 1)\}$. With Algorithm 2 we find $h = xx_2^2 + x_1$ such that $h_{(1)} = p$. With Algorithm 1 we find $g = x^2 + x$ such that $f = g \circ h$. So we get the pol *decomposition $f = (x^2 + x) \circ (xx_2^2 + x_1)$.*

4. REDUCTION TO PSEUDO LINEAR CASE

A *d-pol* p in $\mathbb{Q}\{x\}$ is called *pseudo linear* if p is of the form

$$p = cx_{o_p} + p_1 \tag{7}$$

where $c \in \mathbb{Q}$ and p_1 is of lower order than o_p.

In what follows, we will assume that $o_g > 0$ and compute the decomposition $f = g \circ h$. Since $o_g > 0$, by (6), we have $d_f = d_g$. Let $d = d_f = d_g$ and

$$\begin{aligned} f &= f_d x_{o_f}^d + f_{d-1} x_{o_f}^{d-1} + \ldots + f_1 x_{o_f} + f_0 \\ g &= g_d x_{o_g}^d + g_{d-1} x_{o_g}^{d-1} + \ldots + g_1 x_{o_g} + g_0 \end{aligned}$$

By (1), $x_{o_g} \circ h = s_h x_{o_g + o_h} + R_h = s_h x_{o_f} + R_h (o_{R_h} < o_g + o_h)$. We have $f = (g_d \circ h)(s_h x_{o_f} + R_h)^d + \ldots + g_0 \circ h$. Let $a_i = g_i \circ h$, comparing the coefficients of $x_{o_f}^i (0 \leq i \leq d)$, we have

$$\begin{aligned} f_d &= s_h^d (g_d \circ h) \\ &\ldots \\ f_k &= s_h^k \left(\binom{d}{k} a_d R_h^{d-k} + \binom{d-1}{k} a_{d-1} R_h^{d-k-1} + \ldots + a_k \right) \\ &\ldots \\ f_0 &= a_d R_h^d + a_{d-1} R_h^{d-1} + \ldots + a_1 R_h + a_0 \end{aligned} \tag{8}$$

Let $T = \{H : H = 1$ or H is a primitive *d-pol* with integer coefficients and H^i is the factor of $f_i (1 \leq i \leq d)\}$. From (8), we can assume that s_h is in T. The basic idea of our algorithm is as follows: *for each H in T, we will examine whether there exists a decomposition $f = g \circ h$ such that $s_h = H$.*

LEMMA 6. *Let $f = g \circ h (o_g > 0)$, $d = d_f (= d_g)$. $s^{(k)}(f) = \frac{\partial^k f}{\partial^k x_{o_f}}$, $s^{(k)}(g) = \frac{\partial^k g}{\partial^k x_{o_g}} (1 \leq k \leq d)$. Then we have $s^{(i)}(f)/s_h^i = s^{(i)}(g) \circ h$ $(1 \leq i \leq d)$.*

Proof. By the third equation in (3), we have $s^{(1)}(f)/s_h = s_f/s_h = s_g \circ h$. This proves the case for $i = 1$. Suppose that the lemma is valid for $i = k$, that is, we have $s^{(k)}(f)/s_h^k = s^{(k)}(g) \circ h$. Using (3) to the formula above again, we have $s^{(k+1)}(f)/s_h^{k+1} = \frac{\partial (s^{(k)}(f)/s_h^k)}{\partial x_{o_f}} s_h^{-1} = (\frac{\partial (s^{(k)}(g))}{\partial x_{o_g}} \circ h) s_h s_h^{-1} = s^{(k+1)}(g) \circ h$. We may move s_h out from the scope of the partial differentiation, because $o_{s_h} < o_f = o_{s_f}$ when $d > 1$. ∎

In Lemma 6, setting $i = d - 1$, we have $s^{d-1}(f)/s_h^{d-1} = s^{d-1}(g) \circ h$. By direct computation, $s^{d-1}(f) = \frac{\partial^{d-1} f}{\partial^{d-1} x_{o_f}} = d! f_d x_{o_f} + (d-1)! f_{d-1}, s^{d-1}(g) = d! g_d x_{o_g} + (d-1)! g_{d-1}$. Substituting them into $s^{d-1}(f)/s_h^{d-1} = s^{d-1}(g) \circ h$, we have

$$\left(f_d x_{o_f} + \frac{1}{d} f_{d-1}\right) \Big/ s_h^{d-1} = (g_d x_{o_g} + \frac{1}{d} g_{d-1}) \circ h \tag{9}$$

We consider two cases .

Case 1. If $\frac{f_d}{s_h^d} \notin \mathbb{Q}$, then from (8) h is a right decomposition factor of $\frac{f_d}{s_h^d}$. Then we can repeat the procedure for $\frac{f_d}{s_h^d}$. Notice that the left decomposition factor of $\frac{f_d}{s_h^d}$ could be a *pol*, but the analysis above assumes that $o_g > 0$. So we will use Algorithm 3 to test whether $\frac{f_d}{s_h^d}$ has a *pol* decomposition before the recursion.

Case 2. If $\frac{f_d}{s_h^d} = a \in \mathbb{Q}$, then by (9) and (1) we have

$$\begin{aligned} s_h x_{o_f} + \frac{1}{ad} \cdot \frac{f_{d-1}}{s_h^{d-1}} &= (\frac{g_d}{a} x_{o_g} + \frac{1}{ad} g_{d-1}) \circ h \\ &= \frac{g_d}{a}(s_h x_{o_f} + R_h) + \frac{1}{ad} g_{d-1} \circ h. \end{aligned}$$

Compare the coefficients of x_{o_f}, we have $s_h = (\frac{g_d}{a} \circ h) \cdot s_h$, which implies $\frac{g_d}{a} = 1$. Let $w = \frac{1}{ad} \cdot \frac{f_{d-1}}{s_h^{d-1}}$, $g_1 = \frac{1}{ad} g_{d-1}$. Then

$$p = s_h x_{o_f} + w = (x_{o_g} + g_1) \circ h \tag{10}$$

The left decomposition factor of p is pseudo linear.

The following algorithm is based on the analysis above.

ALGORITHM 4. *Input: d-pols f, H.*

Output: d-pols g and h such that $f = g \circ h$, or a d-pol p such that if h is a right decomposition factor of f with separant H, then h is a right decomposition factor of p and the corresponding left decomposition factor of p w.r.t.h is pseudo linear, or return the empty set which denotes that f has no right decomposition factor with separant H.

S1 $t = f$.

S2 Let $d = d_t$. Write t as the form $t = t_d x_{o_t}^d + \ldots + t_1 x_{o_t} + t_0$. If $o_t < o_H$, then output the empty set and terminate the algorithm; otherwise, we will try to find a decomposition $t = r \circ h$ such that $s_h = H$.

S3 If $a = \frac{s_t}{H} \in \mathbb{Q}$, by (3), $s_r \circ h = s_t / s_h = s_t / H = a$, so $s_r = a$ and r is pseudo linear. Notice that the order of r could be 0, which means t is a possible right decomposition factor of f. Call Algorithm 1 with f and t as input. If we find a g such that $f = g \circ t$, then output g and t; else, output $p = t$. Terminate the algorithm.

S4 Execute Algorithm 3 with input t. If we find r, h such that $t = r \circ h$ and $\frac{s_h}{H} \in \mathbb{Q}$, then use Algorithm 1 with input f and h to find g such that $f = g \circ h$. If such a g exists, output g and h and terminate the algorithm; otherwise, go to next step. This step solves the case the left decomposition factor of t being a non-trivial *pol*.

S5 Now t does not have a *pol* decomposition in which the separant of the right decomposition factor equals H and so the analysis in this section is correct. From (8), H^i should be a factor of t_i, for $i = 1, \ldots, d$. If there exists any i such that $H^i \nmid t_i$, then output the empty set and terminate the algorithm.

S6 Let $c = t_d / H^d$. If $c \in \mathbb{Q}$, let $w = \frac{1}{ad} \frac{t_{d-1}}{H^{d-1}}$. By (10), output $p = H x_{o_f} + w$ and terminate the algorithm.

S7 If $c \notin \mathbb{Q}$, then by (8), h is also a right decomposition factor of c. Let $t = c - c_1$, where c_1 denotes the constant term of c. Then h is also a right decomposition factor of t. Go to S2.

EXAMPLE 3. *$f = 2x_2(x_2^2+x_1)x_3 + x_2^4 + x_2^3 + 2x_1x_2^2 + x_1x_2 + x_1^2, H = x_2$. Let $t = f$, we have $d_t = 1, t_1 = 2x_2(x_2^2+x_1), H \mid t_1$. From* **S2** *to* **S6**, *the algorithm does nothing. In* **S7**, *we get $t = 2(x_2^2 + x_1)$. Return to* **S2**. *In* **S3**, $\frac{s_t}{H} = 4 \in \mathbb{Q}$. *By Algorithm 1 we find a $g = \frac{1}{4}xx_1 + \frac{1}{4}x^2$ such that $f = (\frac{1}{4}xx_1 + \frac{1}{4}x^2) \circ (2x_2^2 + 2x_1)$.*

5. PSEUDO LINEAR CASE

We will solve the following problem: *for two given* d-pols *H and p with $d_p = 1, s_p = H$, find a decomposition $p = r \circ h$ under the condition that r is pseudo linear and $s_h = H$.*

We write *d-pols p, r, h* as the sum of total degree homogeneous parts:

$$\begin{aligned} p &= P_d + P_{d-1} + \ldots + P_2 + P_1 \\ r &= R_{d_1} + R_{d_1-1} + \ldots + R_2 + R_1 \\ h &= H_{d_2} + H_{d_2-1} + \ldots + H_2 + H_1 \end{aligned}$$

where d, d_1, d_2 denote the total degree of p, r, h respectively. Notice that p, r, h have no terms in $\mathbb{Q}$. Since r is pseudo linear, $o_{R_i} < o_{R_1}$ for $2 \le i \le d_1$.

Denote $\boxed{f}_k$ the sum of the monomials included in f with total degree k. We assume that $d_1 < d_2$ (the case $d_1 \ge d_2$ is similar). Comparing the sum of the monomials with total degree l $(1 \le l \le d)$ in $p = r \circ h$, we have

$$\begin{aligned} P_1 &= R_1 \circ H_1 \\ &\ldots \\ P_k &= R_1 \circ H_k + R_k \circ H_1 + \sum_{1<i<k} \boxed{R_i \circ (\textstyle\sum_{1\le j\le k-1} H_j)}_k \quad (k < d_1) \\ &\ldots \\ P_{d_1} &= R_1 \circ H_{d_1} + R_{d_1} \circ H_1 + \sum_{1<i<d_1} \boxed{R_i \circ (\sum_{1\le j\le d_1-1} H_j)}_{d_1} \\ &\ldots \\ P_s &= R_1 \circ H_s + \sum_{1<i\le d_1} \boxed{R_i \circ (\sum_{1\le j\le s-1} H_j)}_{d_2} \quad (d_1 < s \le d_2) \\ &\ldots \\ P_v &= \boxed{\sum_{1<i\le d_1} R_i \circ (\sum_{1\le j\le v} H_j)}_v \quad (v > d_2) \\ &\ldots \\ P_d &= R_{d_1} \circ H_{d_2} \end{aligned} \tag{11}$$

The basic idea of our algorithm is as follows: find R_1, H_1 from the first equation in (11) and substitute R_1, H_1 into the second equation to obtain $P_2' = P_2 - (R_1 \circ H_2 + R_2 \circ H_1)$. From $P_2' = 0$, we obtain a system of linear equations in the coefficients of R_2 and H_2. To solve this linear equation system, we may obtain R_2 and H_2 and so on. To give a precise algorithm, we need to solve the following problems.

1. How to determine d_1 and d_2?
2. How can we obtain R_1, H_1 from the first equation of (11), that is, how to decompose linear *d-pols* with constant coefficients?
3. Do $R_k, H_k (1 \le k \le min\{d_1, d_2\})$ exist and are they unique? If they exist but are not unique, we will face the difficult problem of solving algebraic equations about the coefficients in the next step.

4. If $d_1 < d_2$, we will obtain $R_i, H_i (1 \le i \le d_1)$ firstly and then compute $H_j (d_1 < j \le d_2)$. We need to determine whether $H_j (d_1 < j \le d_2)$ is unique. Similarly, if $d_1 > d_2$, is $R_j (d_2 < j \le d_1)$ unique?

5. If $P_1 = 0$, what shall we do?

For the first problem, by Lemma 5 we have $d = d_1 d_2$. It is obvious that $d_2 = \mathrm{tdeg}(h) \ge \mathrm{tdeg}(H) + 1 (H$ is the separant of h which has been given). So we will search all possible pairs (d_1, d_2) satisfying these two conditions.

For the second problem, we have the following result.

LEMMA 7. *Let $q = \sum_{k=0}^{n} a_k x_k, u = \sum_{i=0}^{m} b_i x_i$, $v = \sum_{j=0}^{n-m} c_j x_j$, where $m < n$ and $a_k, b_i, c_j \in \mathbb{Q}$(or $\mathbb{C}$). Then $q = u \circ v$ if and only if $\hat{q} = \hat{u}\hat{v}$, where $\hat{q} = \sum_{k=0}^{n} a_k y^k$, $\hat{u} = \sum_{i=0}^{m} b_i y^i, \hat{v} = \sum_{j=0}^{n-m} c_j y^j$.*

Proof. From $q = u \circ v$, we have $q = (\sum_{i=0}^{m} b_i x_i) \circ (\sum_{j=0}^{n-m} c_j x_j)$ $= \sum_{i=0}^{m} \sum_{j=0}^{n-m} b_i c_j x_{i+j} = \sum_{k=0}^{n} (\sum_{i+j=k} b_i c_j) x_k$. Comparing the coefficients of x_i, we have $a_k = \sum_{i+j=k} b_i c_j$. Then $\hat{u}\hat{v} = (\sum_{i=0}^{m} b_i y^i)(\sum_{j=0}^{n-m} c_j y^j) = \sum_{k=0}^{n} (\sum_{i+j=k} b_i c_j) y^k = \sum_{k=0}^{n} a_k y^k = \hat{q}$. ∎

By Lemma 7, the problem of decomposing an LODE q with constant coefficients is equivalent to the problem of factoring $\hat{q}$, so R_1, H_1 could have many choices. Since $o_H \le o_h$ and r is pseudo linear, we have $o_r = o_{R_1} = o_p - o_h \le o_p - o_H$. So $o_r = o_{R_1} \le \min\{o_{P_1}, o_p - o_H\}$. This may help us to reduce the choices of R_1 and H_1. Our algorithm will start with a possible pair (R_1, H_1) and (d_1, d_2) obtained from above.

LEMMA 8. *Let q, u be total degree homogeneous d-pols. Let $ddeg(q) = \tilde{d}, ddeg(u) = \tilde{d_2}$, and $q_{\tilde{d}}, u_{\tilde{d_2}}$ denote the sum of the monomials in q, u with differential degree $\tilde{d}, \tilde{d_2}$ respectively. n is a positive integer, $a_i \in \mathbb{Q} (0 \le i \le n)$, we have:*

(1). If $q = (a_n x_n + a_{n-1} x_{n-1} + \ldots + a_1 x_1 + a_0 x) \circ u$, then $q_{\tilde{d}} = a_n x_n \circ u_{\tilde{d_2}}, \tilde{d} = n + \tilde{d_2}$.

(2). If $q = u \circ (a_n x_n + a_{n-1} x_{n-1} + \ldots + a_1 x_1 + a_0 x)$, then $q_{\tilde{d}} = a_n^{tdeg(q)} u_{\tilde{d_2}} \circ x_n, \tilde{d} = n \cdot tdeg(q) + \tilde{d_2}$.

Proof. Omitted. See the preprint [8]. ∎

LEMMA 9. *Let q be a d-pol, $k = min\{i : x_i$ appears in $q\}$, n a positive integer. If q is not of the form $cx_k + q'$, where $c \in \mathbb{Q}$ and x_k does not appear in q', then x_k appears in $x_n \circ q$.*

Proof. Omitted. See the preprint [8]. ∎

The theorem below answers problem 3 affirmatively.

THEOREM 1. *Let R_1, H_1 be linear d-pols and P_k a k-total degree homogeneous d-pol with $k > 1$. If there exist k-total degree homogeneous d-pols R_k, H_k such that $o_{R_k} < o_{R_1}$ and $P_k = R_1 \circ H_k + R_k \circ H_1$, then they are unique.*

Proof. Suppose that (R_{k1}, H_{k1}), (R_{k2}, H_{k2}) both satisfy the condition, that is

$$P_k = R_1 \circ H_{k1} + R_{k1} \circ H_1 = R_1 \circ H_{k2} + R_{k2} \circ H_1 \quad (12)$$

Since R_1 is linear, from (12) we have

$$R_1 \circ (H_{k1} - H_{k2}) + (R_{k1} - R_{k2}) \circ H_1 = 0$$

Since $R_{k1} - R_{k2}$ and $H_{k1} - H_{k2}$ are still k-total degree homogeneous, we need only to prove that there exist no nonzero R_k and H_k such that $R_1 \circ H_k = R_k \circ H_1$. Suppose such H_k, R_k exist. Let $m = o_{R_1}, n = o_{H_1}$, and the sum of monomials included in R_k, H_k with differential degree $\mathrm{ddeg}(R_k), \mathrm{ddeg}(H_k)$ be $\tilde{R}_k, \tilde{H}_k$ respectively. Compare the sum of monomials with maximal differential degree in both sides, since R_1, H_1 are linear, by Lemma 8 we have

$$a_1 x_m \circ \tilde{H}_k = b_1^k \tilde{R}_k \circ x_n \quad (13)$$

where a_1, b_1 are the initial of R_1, H_1 respectively $(a_1, b_1 \in \mathbb{Q})$. Let $k = \min\{i : x_i$ appears in $\tilde{R}_k \circ x_n\}$. Then $k \ge n$, namely, $x_i (0 \le i < n)$ does not appear in the right side of (13). By Lemma 9, since $\tilde{H}_k$ is k-total degree homogeneous with $k > 1$, we have $o_{\tilde{H}_k} \ge n$. Therefore,

$$o_{H_k} \ge o_{\tilde{H}_k} \ge n = o_{H_1}.$$

But by $R_1 \circ H_k = R_k \circ H_1$, we have $o_{R_1} + o_{H_k} = o_{R_k} + o_{H_1}$, which implies $o_{R_1} \le o_{R_k}$. This contradicts with the hypothesis. ∎

The fourth problem is easier to explain: when $d_1 \ge d_2$, we obtain all $H's$ firstly, then we have obtained the right decomposition factor of p and the corresponding left decomposition factor is unique certainly; when $d_1 < d_2$, we obtain all $R's$ firstly. $H_i (d_1 < i \le d_2)$ is unique from the fact: for two given *d-pols* f and g, if g is linear, then the h which satisfies $f = g \circ h$ is unique if it exists.

For the last problem, the method mentioned above is also valid. Let $k = min\{i : P_i \ne 0\}$. We will obtain R_1, H_k from the relation $R_1 \circ H_k = P_k$ and get other R_i, H_j by solving linear equations with their coefficients as unknowns. When R_1 is given , the uniqueness for $H_j (k \nmid j))$ is obvious and the uniqueness for $H_j (k \mid j)$ is guaranteed by Theorem 2 which follows below.

LEMMA 10. *If $x_1 \circ r = s \circ t$, where r, s, t are total degree homogeneous and $tdeg(s) > 1$, then there exists a d-pol s' such that $s = x_1 \circ s'$.*

Proof. Omitted. See the preprint [8]. ∎

THEOREM 2. *Let R_1, H_1 be linear d-pols and P_{ik} a $i \cdot k$-total degree homogeneous d-pol. If there exist an $i-$total degree homogeneous d-pol R_i and $i \cdot k$-total degree homogeneous d-pol H_{ik} such that $o_{R_i} < o_{R_1}$ and $P_{ik} = R_1 \circ H_{ik} + R_i \circ H_k$, then they are unique.*

Proof. Since R_1, H_1 are linear, we need only to prove that there exist no nonzero H_{ik} and R_i such that $R_1 \circ H_{ik} = R_i \circ H_k$, as in Theorem 1. Suppose that such H_{ik}, R_i exist. Let $m = o_{R_1}$, and the sum of the monomials included in H_{ik}, R_i, H_k with differential degree $\mathrm{ddeg}(H_{ik})$, $\mathrm{ddeg}(R_i)$, $\mathrm{ddeg}(H_k)$ be $\tilde{H}_{ik}, \tilde{R}_i, \tilde{H}_k$ respectively. Compare the sum of monomials with maximal differential degree in both sides we have

$$a_1 x_m \circ \tilde{H}_{ik} = \tilde{R}_i \circ \tilde{H}_k$$

where a_1 is the initial of $R_1 (a_1 \in \mathbb{Q})$. By Lemma 10, there exists some differential polynomial R_i' such that $\tilde{R}_i = x_m \circ R_i'$, so we have $o_{R_1} = m \le o_{\tilde{R}_i} \le o_{R_i}$, which contradicts with the hypothesis. ∎

To get R_1 and H_k, we need to solve the problem: *given a total degree homogeneous d-pol q, how to obtain d-pols s, t such that $q = s \circ t$ and s is linear.*

Assume that q can be decomposed as $q = s \circ t$, where s is linear and t is total degree homogeneous. By Lemma 7, we can assume that $s = x_1 + ax, a \in \mathbb{C}$. Let $\tilde{d} = \text{ddeg}(q)$. Since $\text{ddeg}(q) - \text{ddeg}(t) = 1$, we write q and t as differential degree homogeneous parts:

$$\begin{aligned} q &= Q_{\tilde{d}} + Q_{\tilde{d}-1} + \ldots + Q_1 + Q_0 \\ t &= T_{\tilde{d}-1} + T_{\tilde{d}-2} + \ldots + T_1 + T_0. \end{aligned}$$

Compare the sum of the monomials with differential degree $i (0 \leq i \leq \tilde{d})$ of both sides of $q = s \circ t$, we have

$$Q_{\tilde{d}} = x_1 \circ T_{\tilde{d}-1}, \ldots, Q_0 = ax \circ T_0.$$

So we have: $T_0 = \frac{Q_0}{a}$, $T_i = \frac{Q_i - x_1 \circ T_{i-1}}{a} (1 \leq i \leq \tilde{d} - 1)$. Substituting $T_0, \ldots, T_{\tilde{d}-1}$ into $Q_{\tilde{d}} - x_1 \circ T_{\tilde{d}-1} = 0$, we have

$$a^{\tilde{d}} Q_{\tilde{d}} - a^{\tilde{d}-1}(x_1 \circ Q_{\tilde{d}-1}) + \ldots + (-1)^{\tilde{d}} x_{\tilde{d}} \circ Q_0 = 0.$$

When q is given, the $Q's$ are decided. So we obtain a set of equations about a. Conversely, if a satisfies these equations and $a \neq 0$, then $s = x_1 + ax$ is a left decomposition factor of q(to deal with the case $a = 0$, we will use Algorithm 2). When consider s in $\mathbb{Q}\{x\}$, we may use Lemma 7 to obtain a linear left decomposition factor of maximal order of q.

ALGORITHM 5. *Input: a total degree homogeneous d-pol q.*

Output: the linear left decomposition factor s of q with maximal order, if it exists.

S1 Using Algorithm 2 to get the maximal n such that $q = x_n \circ q'$. Let $\tilde{d} = \text{ddeg}(q')$, write q' as $q' = Q_{\tilde{d}} + Q_{\tilde{d}-1} + \ldots + Q_1 + Q_0$, where Q_i is the sum of the monomials included in q' with differential degree $i (0 \leq i \leq \tilde{d})$. Calculate $V_i = x_{\tilde{d}-i} \circ Q_i (0 \leq i \leq \tilde{d})$.

S2 Let $w = a^{\tilde{d}} V_{\tilde{d}} - a^{\tilde{d}-1} V_{\tilde{d}-1} + \ldots + (-1)^{\tilde{d}-i} a^i V_i + \ldots + (-1)^{\tilde{d}-1} a V_1 + (-1)^{\tilde{d}} V_0$, and S the set of the coefficients of each monomial included in w. We obtain the equations $S = 0$ of a.

S3 To solve the equations $S = 0$ in $\mathbb{Q}$ with $a \neq 0$, we basically need to compute the GCD g of the *pols* in S. If $g \in \mathbb{Q}$, then there is no solution for a and output $s = x_n$(if $n = 0$, then the decomposition does not exist). Otherwise, let $g = a^l(a^m + a_{m-1}a^{m-1} + \ldots + a_1 a + a_0)$ where $a_i \in \mathbb{Q}$ and $a_0 \neq 0$. By Lemma 7, the output is $s = x_n \circ (x_m + a_{m-1}x_{m-1} + \ldots + a_1 x_1 + a_0 x)$.

EXAMPLE 4. *Let $q = 2x_1x_3 + 2x_2^2 + xx_3 + 3x_1x_2 + 2xx_2 + x_1^2 - xx_1 - x^2$. In S1, $n = 0$ and we have $\tilde{d} = 4, Q_4 = 2x_1x_3 + 2x_2^2, Q_3 = xx_3 + 3x_1x_2, Q_2 = 2xx_2 + x_1^2, Q_1 = -xx_1, Q_0 = -x^2$. So,*

$$\begin{cases} V_4 = Q_4 = 2x_1x_3 + 2x_2^2 \\ V_3 = x_1 \circ Q_3 = xx_4 + 4x_1x_3 + 3x_2^2 \\ V_2 = x_2 \circ Q_2 = 2xx_4 + 6x_1x_3 + 4x_2^2 \\ V_1 = x_3 \circ Q_1 = -(xx_4 + 4x_1x_3 + 3x_2^2) \\ V_0 = x_4 \circ Q_0 = -(8x_1x_3 + 6x_2^2 + 2xx_4) \end{cases}$$

In S2, we have $p = a^4V_4 - a^3V_3 + a^2V_2 - aV_1 + V_0 = (2a^4 - 4a^3 + 6a^2 + 4a - 8)x_1x_3 + (2a^4 - 3a^3 + 4a^2 + 3a - 6)x_2^2 + (-a^3 + 2a^2 + a - 2)xx_4 = 0$. Collecting the coefficients, we have

$$\begin{cases} h_1 = 2a^4 - 4a^3 + 6a^2 + 4a - 8 \\ h_2 = 2a^4 - 3a^3 + 4a^2 + 3a - 6 \\ h_3 = -a^3 + 2a^2 + a - 2 \end{cases}$$

In S3, compute the GCD of h_1, h_2, h_3, we have $g = a^2 - 1$. So the left decomposition factor of q with the maximal order is: $s = x_2 - x$.

We can now present the algorithm.

ALGORITHM 6. *Input: d-pols f, p, H with $d_p = 1, i_p = H$. Output:* d-pols g, h *such that $f = g \circ h$ where h is a right decomposition factor of p and $\frac{s_h}{H} \in \mathbb{Q}$. If such g and h do not exist, return nothing.*

S1 Let $d = \text{tdeg}(p)$. Write p as $p = \sum_{i=1}^{d} P_i$ where P_i denotes the sum of the monomials in p with total degree $i (1 \leq i \leq d)$.

S2 Let $k = min\{i : P_i \neq 0\}$. If $k = 1$, let $L = P_1$; else, let L be the linear left decomposition factor of P_k with maximal order obtained with Algorithm 5. If L does not exist, the algorithm terminates; otherwise, let $S = \{(A, d') : A$ is a linear left decomposition factor of L and $i_A = 1$, d' is an integer such that $d' \mid \text{tdeg}(p)$ and $d' \geq \text{tdeg}(H)+1\}$. By Lemma 7, a left decomposition factor of L can be found by factoring a univariate *pol*.

S3 If $S \neq \emptyset$, select a (A, d') from S and let $R_1 = A, d_2 = d', d_1 = \text{tdeg}(p)/d'$. Otherwise, the algorithm terminates.

S4 Let $H_i = 0 (1 \leq i < k)$. Solve the equations (11) to find $H_k, R_2, H_{k+1}, \ldots, R_{d_1}, H_{d_2}$ by treating the coefficients of $R's$ and $H's$ as unknowns with the conditions $o_{R_l} < o_{R_1}, o_{H_j} \leq o_p - o_{R_1}$ and $\text{tdeg}(R_l)=l$, $\text{tdeg}(H_j)=j$ for $2 \leq l \leq d_1, k \leq j \leq d_2$. By Theorems 1 and 2, such a solution must be unique if it exists. If the solution does not exist, $S := S - \{(A, d')\}$, go to S3. Otherwise, let $h = H_k + H_{k+1} + \ldots + H_{d_2}$.

S5 if $\frac{s_h}{H} \notin \mathbb{Q}$, $S := S - \{(A, d')\}$, go to S3.

S6 Execute Algorithm 1 with input f, h, if we obtain a g such that $f = g \circ h$, then output g and h; else, $S := S - \{(A, d')\}$, go to S3.

The example follows below shows how to obtain the decomposition of a *d-pol p* **when we have known that the left decomposition factor of p is pseudo linear.**

EXAMPLE 5. *Let $p = (xx_2 + x_1^2)^2 + 4x_1(xx_2 + x_1^2) + 4x_1^2 - xx_1 + 3x_1x_2 + xx_3 + 2x_2 - 2x$.*

S1. $d = tdeg(p) = 4$, write p as $p = P_4 + P_3 + P_2 + P_1$, where $P_4 = x^2x_2^2 + x_1^4 + 2xx_1^2x_2$, $P_3 = 4xx_1x_2 + 4x_1^3$, $P_2 = 4x_1^2 - xx_1 + 3x_1x_2 + xx_3$, $P_1 = 2x_2 - 2x$.

S2. From $P_1 = 2(x_2 - x) = R_1 \circ H_1$ and by Lemma 7, we choose R_1 to be one of $x_2 - x$, $x_1 + x$, $x_1 - x$. Since $tdeg(p) = 4, tdeg(H) = 2$, we have $S = \{(x_2 - x, 2), (x_2 - x, 4), (x_1 + x, 2), (x_1 + x, 4), (x_1 - x, 2), (x_1 - x, 4)\}$.

S3. For $R_1 = x_2 - x, d_2 = 2$, $H_1 = 2x, d_1 = d/d_2 = 2$. Substituting these into (11), we have:

$$P_2 = (x_2 - x) \circ H_2 + R_2 \circ 2x \quad (14)$$

$$P_3 = \boxed{R_2 \circ (H_1 + H_2)}_3 \quad (15)$$

$$P_4 = R_2 \circ H_2 \quad (16)$$

S4. Since $o_{R_2} < 2, o_{H_2} \leq o_f - o_{R_1} = 3-2 = 1$ *and* R_2, H_2 *are* $2-$*total degree homogeneous, so we can assume*

$$R_2 = a_1x^2 + a_2xx_1 + a_3x_1^2, H_2 = b_1x^2 + b_2xx_1 + b_3x_1^2$$

Substitute them in (14)*, we have:* $2b_1xx_2 + (2b_1 - b_3 + 4a_3 - 4)x_1^2 + (b_2 - 1)xx_3 + (3b_2 - 3)x_1x_2 + 2b_3x_1x_3 + 2b_3x_2^2 + (4a_2 - b_2 + 1)xx_1 + 4a_1x^2 = 0$*. We obtain the equations for the coefficients of* R_2, H_2*:*

$$\begin{cases} 2b_1 = 2b_3 = a_1 = 0, 2b_1 - b_3 + 4a_3 - 4 = 0 \\ b_2 - 1 = 0, 3b_2 - 3 = 0, 4a_2 - b_2 + 1 = 0 \end{cases}$$

The unique solution is: $a_1 = 0, a_2 = 0, a_3 = 1; b_1 = 0, b_2 = 1, b_3 = 0$*. So we obtain* $R_2 = x_1^2, H_2 = xx_1$*. Now we substitute* R_1, R_2, H_1, H_2 *in* (15) *and* (16)*, we find that they really hold. So we have the decomposition for* p*:*

$$p = (R_1 + R_2) \circ (H_1 + H_2) = (x_1^2 + x_2 - x) \circ (xx_1 + 2x).$$

When we choose the other elements of S*, we do not obtain solutions for* R_2, H_2*. Here we omit the calculation.*

6. THE GENERAL CASE AND EXPERIMENTAL RESULTS

ALGORITHM 7. *Input: a* f *in* $\mathbb{Q}\{x\}$.
Output: a non-trivial decomposition $f = g \circ h$*, if such* g, h *exist.*

S1 Find a *pol* $g \in \mathbb{Q}[x]$ and an $h \in \mathbb{Q}\{x\}$ such that $f = g \circ h$ with Algorithm 3. If such g and h exist, output g, h. Otherwise, go to next step.

S2 Let $d = d_f$, write f as the form $f_d x_{o_f}^d + f_{d-1}x_{o_f}^{d-1} + \ldots + f_1 x_{o_f} + f_0$, where f_i denotes the coefficient of $x_{o_f}^i$ $(0 \leq i \leq d)$. Let $T = \{H : H = 1$ or H^i is a factor of $f_i (1 \leq i \leq d)\}$. To make the selection of factors unique, we assume that H is with integer coefficients and primitive.

S3 If $T \neq \phi$, choose $H \in T$, go to next step; otherwise, terminate the algorithm and return "no non-trivial decomposition exists".

S4 By (8), H could be the separant of h. Execute Algorithm 4 with input f, H. There are three cases:

(1) We obtain a decomposition of f. Output the decomposition and terminate the algorithm.

(2) We obtain a *d-pol* p such that if h is a right decomposition factor of f with separant H, then h is a right decomposition factor of p and the corresponding left decomposition factor of p w.r.t.h is pseudo linear. Go to next step.

(3) The output is the empty set. Let $T := T - \{H\}$, go to **S3**;

S5 Execute Algorithm 6 with input f, p, H. If we get a decomposition $f = g \circ h$, then output g and h; otherwise $T := T - \{H\}$, go to **S3**.

EXAMPLE 6. *Let* $f = 2x_1x_3 + 4x_1^2x_2^2 + 4x_1^2x_2 + 4x_1^3x_2 + 4x_1x_2x + x_1^2 + 2x_1^3 + 2x_1x + x_1^4 + 2x_1^2x + x^2 + 2x_2^2 + x_2 + 2x_1x_2 + x_1$.

S1. Execute Algorithm 3*, we find that* f *does not have a* pol *decomposition.*

(o_f, t_f, l_f)	time(s)	(o_f, t_f, l_f)	time(s)
(2,10,32)	0.237	(2, 20, 116)	1.174
(2,30,254)	4.200	(2, 40, 414)	12.938
(2,50,624)	35.021	(3, 10, 102)	2.365
(3, 12,322)	14.078	(3, 15, 368)	11.167
(4, 8,415)	19.786	(4, 10, 555)	32.171
(5, 8,1084)	55.986	(6, 6,596)	68.843
(7, 6,1308)	164.44.	(8, 5, 325)	14.266
(9, 4,415)	19.786	(10, 4, 677)	72.534

Table 1: Decomposing Randomly Generated Differential Polynomials

(o_h, t_g)	(o_h, t_h)	(o_f, t_f, l_f)	time(s)
(0,8)	(1,8)	(1,64,639)	10.079
(1,6)	(1,8)	(2,48,1174)	85.640
(1,4)	(2,4)	(3,16,458)	38.672
(1,4)	(3,2)	(4,8,994)	95.063
(1,4)	(3,4)	(4,16,970)	144.467
(2,4)	(1,4)	(3,16,1229)	189.109
(2,3)	(2,4)	(4,12,1360)	120.093
(2,3)	(3,2)	(5,6,709)	90.405
(3,2)	(1,4)	(4,8,231)	15.562
(3,2)	(2,4)	(5,8,535)	32.891

Table 2: Decomposing Differential Polynomials Composed Randomly

S2. $d = d_f = 1, f_1 = 2x_1, T = \{1, x_1\}$.
S3. Select $H = 1$.
S4. Execute Algorithm 4, we obtain $p = x_1$.
S5. By Algorithm 6, we get no decomposition for f.
Now, return to S3. Select $H = x_1$.
S4. Execute Algorithm 4, we obtain $p = 2x_1x_3 + 4x_1^2x_2^2 + 4x_1^2x_2 + 4x_1^3x_2 + 4x_1x_2x + x_1^2 + 2x_1^3 + 2x_1x + x_1^4 + 2x_1^2x + x^2 + 2x_2^2 + x_2 + 2x_1x_2 + x_1$.
S5. By Algorithm 6, we have $g = x_1 + x^2, h = 2x_1x_2 + x_1 + x_1^2 + x$*. So we obtain a decomposition* $f = g \circ h$.

We may use Algorithm 7 recursively to find an irreducible decomposition $f = (x_1 + x^2) \circ (x_1 + x) \circ (x_1^2 + x)$. So the solution of $f = 0$ is reduced to the solving of three first order ODEs.

6.1 Experimental Results

We implement Algorithm 7 in Maple. In Table 1, we generate a *d-pol* randomly and decompose it. All the *d-pols* in Table 1 are indecomposable. In Table 2, we generate two *d-pols* g and h randomly and decompose $f = g \circ h$. These g and h could be found in [8]. The running times are collected on a PC with a 1.6G CPU and 128M memory and are given in seconds. In the table, t_f means the total degree of f and l_f means the number of terms in f. From these results, we may conclude that our algorithm is quite efficient in handling large *d-pols* with hundreds of terms.

6.2 Rational Function Coefficient Case

If the coefficient field is $\mathbb{Q}(t)$, Algorithm 3, Lemma 7, and Algorithm 5 are not correct anymore. For the *pol* decomposition of *d-pols*, we have given an algorithm [8]. Corresponding to Lemma 7, we need to decompose linear *d-pols*

in $\mathbb{Q}(t)\{x\}$, which is much more difficult than the constant coefficient case and has been solved in [4, 26, 18, 23]. Corresponding to Algorithm 5, to find linear left decomposition factors of a total degree homogeneous *d-pol* with coefficients in $\mathbb{Q}(t)$, we need to solve high degree and high order differential equations. How to give an efficient algorithm to find such a left decomposition factor needs further research. While the uniqueness in Theorem 1 and Theorem 2 is still correct, which implies that the method in this paper is also valid in many cases for the decomposition of f in $\mathbb{Q}(t)\{x\}$, such as when the given *d-pol* f is pseudo linear.

7. CONCLUSION

We give an efficient algorithm for decomposing *d-pols* in one variable and with constant coefficients. Besides arithmetic operations, the algorithm needs only factorization of multi-variable *pols* and solution of linear equation systems. Experimental results show that this algorithm can be used to decompose large *d-pols*.

Many problems on the decomposition of *d-pols* are still open. The decomposition of *pols* is unique in certain sense [17]. Similar results were proved for Ore polynomials and hence for LODEs [20]. It is interesting to see whether this property is correct for *d-pol* decomposition. Whether we can extend the methods in [16, 9] to give a *pol* time decomposition algorithm for *d-pols* is also interesting. Finally, to give an efficient algorithm to decompose *d-pols* with coefficients of rational functions is also desirable.

Acknowledgment. We would like to thank the anonymous referees for providing valuable suggestions on this paper.

8. REFERENCES

[1] V.S. Alagar and M. Thanh. Fast decomposition algorithms. In *Proc. EUROCAL 85*, vol 2, 150-153, Springer, 1985.

[2] M.A. Barkatou and E. Pflügel. On the equivalence problem of linear differential systems and its application for factoring completely reducible systems. In *Proc. of ISSAC'98*, 268-275, ACM Press, 1998.

[3] D.R. Barton and R.E. Zippel. Polynomial decomposition algorithms. *J. of Symbolic Computation*, 1(2), 159-168, 1985.

[4] M. Bronstein. An improved algorithm for factorizing linear ordinary differential operators. In *Proc. of ISSAC'94*, ACM Press, 336-340, 1994.

[5] M. Bronstein and M. Petkovšek. On Ore rings, linear operators and factorization. *Programming & Computer Software 20*, 1, 27-44, 1994.

[6] T. Cluzeau. Factorization of differential systems in characteristic p. In *Proc. ISSAC'03*, 58-65, ACM Press, 2003.

[7] A. Fredet. Factorization of linear differential operators in exponential extensions. In *Proc. of ISSAC03*, 103-110, ACM Press, 2003.

[8] X.S. Gao and M. Zhang. Decomposition of differential polynomials. In *MM Res. Preprints*, No. 22, 163-185, 2003. http://www.mmrc.iss.ac.cn/pub/mm-pre.html

[9] J.von zur Gathen. Functional decomposition of polynomials: the wild case. *J. of Symbolic Computation*, 9, 437-452, 1990; the tame case. *J. of Symbolic Computation*, 9, 281-299, 1990.

[10] M. Giesbrecht and Y. Zhang. Factoring and decomposing Ore polynomials over $Fq(t)$. In *Proc. of ISSAC03*, 127-135, ACM Press, 2003.

[11] D.Y. Grigor'ev. Complexity of factoring and calculating the GCD of linear ordinary differential operators. *J. of Symbolic Computation*, 10, 7-37, 1990.

[12] J. Gutierrez, T. Recio and C.R. de Velasco. Polynomial decomposition of almost quadratic complexity. In *Proc. AAECC 6*, Springer, 1989.

[13] Z. Li, F. Schwarz and S.P. Tsarev. Factoring systems of linear PDEs with finite-dimensional solution spaces. *J. Symbolic Computation* 36(3-4), 443-471, 2003.

[14] E.R. Kolchin. *Differential Algebra and Algebraic Groups*, Academic Press, London, 1973.

[15] L. Königsberger. *Allgemeine Untersuchungen aus der Theorie der Differentialgleichungen*, Teubner, Leipzig, 1882.

[16] D. Kozen and S. Landau. Polynomial decomposition algorithms. *J. of Symbolic computation 7*, 445-456, 1989.

[17] J.F. Ritt. Prime and composite pols. *Trans. AMS*, **23**, 51-66, 1922.

[18] F. Schwarz. A factoring algorithm for linear ordinary differential equations. In *Proc. of ISSAC 89*, ACM Press, 17-25, 1989.

[19] M.F. Singer. Liouillian solutions of nth order homogeneous linear differential equations. *Amer. J. of Math.*, 103(4), 661-682, 1981.

[20] O. Ore. Theory of non-commutative polynomials. *The Annals of Mathematics*, 34(3), 480-508, 1933.

[21] M.F. Singer and F. Ulmer. Galois groups of second and third order linear differential equations. *J. of Symbolic Computation 16*, 9-36, 1993.

[22] M. Sosnin. Decomposition of polynomial ordinary differential equations. Krasnoyarsk, to be published, 1999.

[23] S.P. Tsarev. An algorithm for complete enumeration of all factorizations of a linear ordinary differential operator. In *Proc. of ISSAC'96*, 226-231, ACM Press, 1996.

[24] S.P. Tsarev. On factorization of non-linear ordinary differential equations. In *Proc. ISSAC 99*, 159-164, ACM Press, 1999.

[25] H. Umemura. On the irreducibility of the first differential equation of Painleve. In *Algebraic geometry and commutative algebra in honor of Masayoshi Nagata*, Tokyo, 101-109, 1987.

[26] M. van Hoeij. Factorization of differential operators with rational functions coefficients. *J. Symbolic Computation* 24(5), 537-561, 1997.

[27] M. van Hoeij. Formal solutions and factorization of differential operators with power series coefficients. *J. Symbolic Computation*, 24, 1-30 1997.

[28] M. Van der Put and M.F. Singer. *Galois theory of linear differential equations*, Springer, Berlin, 2003.

[29] R.E. Zippel. Rational function decomposition. In *Proc. of ISSAC'91*, 1-6, ACM Press, 1991.