Computing Resultants of Partially Composed Polynomials

Manfred Minimair

E-mail: manfred@minimair.org
WWW: http://minimair.org
Department of Mathematics and Computer Science
Seton Hall University, 400 South Orange Avenue
South Orange, NJ 07079, USA

Abstract. This paper studies resultants of two homogeneous *partially composed* polynomials. By two homogeneous partially composed polynomials we mean a pair of polynomials of which one does not have any given composition structure and the other one is obtained by composing a bivariate homogeneous polynomial with two bivariate homogeneous polynomials. The main contribution of this paper is to show that the resultant of two partially composed polynomials is a certain iterated resultant of the component polynomials. Furthermore, experiments show that, in many cases, this iterated resultant can be computed with dramatically increased efficiency. This paper is part of the author's work on resultants of composed polynomials. This paper is also the completion of a work by McKay and Wang who considered inhomogeneous partially composed polynomials.

1 Introduction

Resultants are fundamental in solving systems of polynomial equations and therefore have been extensively studied ([20], [4], [6], [12], [17], [21], [7], [18], [9], [2]). Recent research is focused on utilizing structure of polynomials, naturally occurring in real life problems, for example, sparsity ([30], [11], [10], [8], [5], [31], [3], [28]) as well as composition ([22], [17], [23], [7], [19], [16], [24], [26], [25], [27]). This paper is part of the author's work on utilizing composition structures. The work [24] also contains a section explaining the importance of composition structures that are considered in this and in previous works.

Previous papers ([16], [26], [25], [27])) by the author considered "fully" composed polynomials. That is, composed polynomials such as $h_1 = f_1 \circ (g_1, g_2, g_3)$, $h_2 = f_2 \circ (g_1, g_2, g_3)$ and $h_3 = f_3 \circ (g_1, g_2, g_3)$, where each composed polynomial h_i is obtained from the polynomial f_i in the variables y_1, y_2, y_3 by replacing y_j with the bivariate polynomial g_j. Note that each composed polynomial has the same inner components g_1, g_2, g_3. The previous works have determined the irreducible factors of projective (Macaulay) or toric (sparse) resultants of such "fully" composed polynomials.

The focus of the current paper is entirely different from the one of the previous papers ([16], [26], [25], [27])). It considers "partially" composed polynomials. By two partially composed polynomials h_1 and h_2, we mean a bivariate homogeneous polynomial h_1 that *does not have any composition structure* and a bivariate homogeneous *composed polynomial* $h_2 = f_2 \circ (g_1, g_2)$ that is obtained from the homogeneous bivariate polynomial f_2 in the variables y_1 and y_2 by replacing y_j with the bivariate homogeneous polynomial g_j. (Of course, g_1 and g_2 are required to have the same total degrees to ensure that h_2 is homogeneous.) The finding of the current paper is also quite different from previous findings ([16], [26], [25], [27])). We find that the projective (dense, Sylvester/Macaulay) resultant of two partially composed polynomials h_1 and h_2 is a certain iterated resultant. More precisely, it is the resultant of the polynomials f_1 and f_2, where f_1 is the resultant of certain polynomials derived from the component polynomials h_1, g_1 and g_2. Interestingly, we find *two different* natural formulas for f_1, one involving a projective (dense, Sylvester/Macaulay) resultant and another one involving a toric (sparse) resultant. Moreover, we show in experiments that for many cases this iterated resultant can be computed, over the integers modulo a prime, with dramatically improved efficiency.

This work can also be considered as a completion of works ([22] and [23]) by McKay and Wang. In [22] they study resultants of two inhomogeneous composed polynomials as well as two inhomogeneous *partially composed* polynomials (in Theorem 7 of [22]). Additionally, in [23] they study the homogeneous generalization for the case of two composed polynomials. However, they *ignore* the case of two *homogeneous partially composed* polynomials. Furthermore, they do not address efficient computation of partially composed polynomials. In fact, their presentation of their result (Theorem 7 of [22]) does not allow an immediate computational application. Also note that Jouanolou's work [17] that considers resultants of composed polynomials in Section 5.12 ignores the partially composed case as well.

Note that the main theorem of the present paper (Theorem 1) can be considered a generalization (to the homogeneous case) of Theorem 7 of the work [22] by McKay and Wang. Therefore we briefly state Theorem 7 of [22]. For the sake of a more uniform presentation, with respect to the current work and to previous works ([16], [26], [25], [27])) of the current author, we use different symbols for the polynomials than in [22]. Let F_2 be a univariate polynomial in the variable y and G and H_1 be univariate polynomials in the variable x. Then, the projective (dense, Sylvester) resultant of H_1 and $H_2 = F_2 \circ G$ is the resultant of F_1 and F_2 where F_1 is given by a certain formula involving the roots of H_1. More precisely,

$$F_1 = H_1(0)^d \prod_{\alpha} (y - G(\alpha)), \tag{1}$$

where d is the degree of G and α ranges over the roots of H_1. (In Line (1) $G(\alpha)$ is obtained from G by replacing the variable x of G with the value α.) Note that the polynomials F_2, G and H_1 can indeed be considered as a sub-case of the homogeneous polynomials subject of the current paper. That is, for homogeneous bivariate polynomials f_2, g_1, g_2 and h_1, we have $F_2 = f_2(g_1(x,1), y)$, where $g_1(x,1) = 1$, $G = g_2(x,1)$ and $H_1 = h_1(x,1)$. (Again, as in Line (1), $g_1(x,1)$ is obtained from the polynomial g_1 by replacing the variable x_1 with x and the variable x_2 with 1. Furthermore, $f_2(g_1(0,1), y)$ and $h_1(x,1)$ are obtained accordingly.) Note that the formula for F_1 looks quite different from the formulas for f_1 in Theorem 1 of the current paper. Please, see Remark 4 for an explanation how they are related.

The reader might wonder whether one can utilize composition structures for other fundamental operations. In fact, this has already been done for some operations. For examples, projective (Macaulay) resultant, Gröbner bases, SAGBI bases, subresultants and Galois groups of certain differential operators have been studied respectively in [26], [14] and [13], [29], [15] and [1] using various mathematical techniques. However, it seems that those techniques cannot be applied to the study of resultants. Therefore in this paper we use mathematical methods that are essentially different from those.

2 Main results

We assume the reader is familiar with the notions of projective (dense, Sylvester/Macaulay) resultant, toric (sparse) resultant and supports of sparse polynomials (see [8], [11], [30]).

Before we state the main theorem we fix a few notations. Let's assume that all the polynomials h_1, f_2, g_1, g_2 in Theorem 1 are defined over the complex numbers. Let h_1 be a bivariate homogeneous polynomial in the variables x_1, x_2 of degree e_1. Let f_2 be a homogeneous bivariate polynomial in the variables y_1, y_2 of degree c_2. Let g_1 and g_2 be bivariate homogeneous polynomials in the variables x_1, x_2 of equal total degrees, denoted by d. Let the composed polynomial $h_2 = f_2 \circ (g_1, g_2)$ be obtained from the polynomial f_2 by replacing y_j with g_j. Note that we had to assume that g_1 and g_2 have equal total degrees in order to ensure that h_2 is homogeneous. Let Res_{c_1,c_2} and $\mathrm{Res}_{\mathcal{C}_1,\mathcal{C}_2,\mathcal{C}_3}$ respectively denote the projective (dense, Sylvester/Macaulay) resultant of two bivariate homogeneous polynomials of respective total degrees c_1 and c_2, and the toric (sparse) resultant of three not necessarily homogeneous polynomials with supports $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$.

Now we are ready to state the main theorem.

Theorem 1 (Main theorem)

$$\mathrm{Res}_{e_1,e_2}(h_1, f_2 \circ (g_1, g_2)) = \mathrm{Res}_{c_1,c_2}(f_1, f_2), \tag{2}$$

where f_1 *is given by both equalities:*

$$f_1 = \operatorname{Res}_{e_1,d}(h_1,\, y_2\, g_1 - y_1\, g_2)\,, \quad \textit{and} \tag{3}$$

$$f_1 = (-1)^{e_1} \operatorname{Res}_{\mathcal{C}_1,\mathcal{C}_2,\mathcal{C}_3}(h_1,\, y_1 - g_1,\, y_2 - g_2)\,. \tag{4}$$

In the above formulas, we have $e_2 = c_2\, d$ *and* $c_1 = e_1$*. Furthermore, the set* $\mathcal{C}_1$ *is the support of a dense homogeneous bivariate polynomial of degree* e_1*. That is,* $\mathcal{C}_1 = \{(e_1, 0), (e_1 - 1, 1), \ldots, (0, e_1)\}$*. Whereas the sets* $\mathcal{C}_2 = \mathcal{C}_3$ *consist of the origin and the support of a dense homogeneous bivariate polynomial of degree* d*. That is,* $\mathcal{C}_2 = \mathcal{C}_3 = \{(0, 0), (d, 0), (d - 1, 1), \ldots, (0, d)\}$*. Moreover, we normalize the sign of the resultant* $\operatorname{Res}_{\mathcal{C}_1,\mathcal{C}_2,\mathcal{C}_3}$ *such that we have* $\operatorname{Res}_{\mathcal{C}_1,\mathcal{C}_2,\mathcal{C}_3}(x_1^{e_1}, x_2^d, 1) = 1$*.*

Remark 2 Note that the resultants in Lines (3) and (4) eliminate that variables x_1, x_2 rather than y_1, y_2.

Notation 3 Let us fix the following notation for the rest of this paper. If p is a bivariate polynomial in the variables x_1 and x_2 then $p(c_1, c_2)$ is obtained from p by replacing x_i with c_i.

Remark 4 The formula in Line (3) can be viewed as a generalization of McKay's and Wang's formula of Line (1). That is, Line (1) implies that, using the notation of Section 1,

$$F_1 = \operatorname{Res}_{e_1,d}(H_1,\, y - G) = \operatorname{Res}_{e_1,d}(h_1,\, y_2\, g_1 - y_1\, g_2)\,,$$

where $y_2 = y$, $y_1 = 1$, $g_1(x, 1) = 1$, $g_2(x, 1) = G$ and $h_1(x, 1) = H_1$.

Also note that McKay's and Wang's formula in Line (1) cannot be easily used for computations because it involves the roots of the polynomial H_1. On the contrary to this, the formula in Line (3) does not involve roots and thus can be easily used for computations.

Furthermore note an interesting difference between the proofs of Line (1) and Line (3). That is, the proof of Line (1) of [22] proceeds with polynomials with arbitrary complex coefficients. Whereas the proof of Line (3) in Section 3 of the current paper relies on polynomials with symbolic (algebraically independent) coefficients. Only after showing Line (3) for polynomials with symbolic coefficients, we observe that Line (3) is stable under specialization and thus Line (3) is valid for polynomials with any complex coefficients. This approach allows avoiding case distinctions in the proof.

Remark 5 Since this paper considers projective (dense, Sylvester/Macaulay) resultants of partially composed polynomials, the reader might find it surprising that the polynomial f_1 is expressed in terms of a toric (sparse) resultant (see Line 4) and not in terms of a projective (dense, Macaulay) resultant. Indeed, one can show that f_1 is also related to a projective resultant. That is, Corollary 5 of [28] implies that the power f_1^d is the projective (dense, Macaulay) resultant of h_1, $y_1 - g_1$ and $g_2 - g_2$ with respect to the total degrees e_1, d and d.

Remark 6 Naturally, one asks how Theorem 1 is related to the well-known formula for resultants of composed polynomials derived by [23] in the homogeneous bivariate case. It turns out that one can rewrite resultants of composed polynomials in terms of resultants of linearly combined polynomials by applying Theorem 1 twice. However, it seems that one cannot derive the main result of [23] only by applying Theorem 1.

To illustrate the previous paragraph, in the following we apply Theorem 1 to resultants of homogeneous bivariate composed polynomials twice. Let f_1 and f_2 be homogeneous bivariate polynomial in the variables y_1, y_2 of respective degrees c_1 and c_2. Let g_1 and g_2 be bivariate homogeneous polynomials in the variables x_1, x_2 of equal total degrees, denoted by d. Then, by Theorem 1,

$$\operatorname{Res}_{c_1 d, c_2 d}(f_1 \circ (g_1, g_2)\,, f_2 \circ (g_1, g_2)) = \operatorname{Res}_{c_1 d, c_2 d}(p, f_2)\,, \tag{5}$$

where $p = \operatorname{Res}_{c_1 d, d}(f_1 \circ (g_1, g_2)\,, y_2 g_1 - y_1 g_2)$ which equals, by Corollary 5 of [23], the formula $(-1)^{c_1 d^2} \operatorname{Res}_{c_1 d, d}(y_2 g_1 - y_1 g_2, f_1 \circ (g_1, g_2))$. Furthermore, by Theorem 1, $p = \operatorname{Res}_{d, c_1}(q, f_1)$, where $q = \operatorname{Res}_{d,d}(y_2 g_1 - y_1 g_2, z_2 g_1 - z_1 g_2)$, where z_1 and z_2 are new distinct variables. Therefore, indeed, one can use Theorem 1 to rewrite the resultant of two composed polynomials in terms of the resultant of two linearly combined polynomials. If one factors q into $(-y_2z_1-y_1z_2)^d \operatorname{Res}_{d,d}(g_1,g_2)$, applying Lemma 7 of [23], and if one utilizes the bi-homogeneity of the resultant, one can simplify Line (5) to obtain McKay's and Wang's formula

$$\operatorname{Res}_{c_1d,c_2d}(f_1\circ(g_1,g_2), f_2\circ(g_1,g_2)) = \operatorname{Res}_{c_1,c_2}(f_1,f_2)^d \operatorname{Res}_{d,d}(g_1,g_2)^{c_1c_2}$$

for resultants of two homogeneous bivariate composed polynomials ([23]).

Remark 7 In the following subsection, "Computational application of the main theorem", we will use Theorem 1 for efficiently computing resultants of partially composed polynomials. The reader will notice that we will not utilize Line (4). It is important to point out that we have stated Line (4) because it is of independent theoretical interest. That is, it makes an explicit connection between projective (dense, Sylvester/Macaulay) resultants of two polynomials and bivariable toric (sparse) resultants of three polynomials.

Computational application of the main theorem

In this subsection we describe how one can apply Theorem 1 to efficiently compute resultants of partially composed polynomials.

Step 1: Computation of f_1 We ask the reader to examine the resultant in Line (3) in Theorem 1. Note that the bi-homogeneity of this resultant implies that the polynomial f_1 is homogeneous in the variables y_1 and y_2. Furthermore the total degree of f_1 is e_1. Thus, in order to compute f_1 it is sufficient to compute the polynomial $p(y_1) = \operatorname{Res}_{e_1,d}(h_1(y_1,1), g_1 - y_1g_2)$. This polynomial p can be computed via interpolation letting y_1 range over the values $0, 1, \ldots, e_1$.

Step 2: Computation of $\operatorname{Res}_{c_1,c_2}(f_1,f_2)$ Note that f_1 and f_2 are *bivariate homogeneous* polynomials. Therefore the resultant $\operatorname{Res}_{c_1,c_2}(f_1,f_2)$ can be computed as the univariable (Sylvester) resultant $\operatorname{Res}_{c_1,c_2}(f_1(y_1,1), f_2(y_1,1))$.

Running Time experiments Now, we discuss some practical running time experiments carried out under Maple 9 on a PC with a 2.2 GHz processor and 3 GB main memory. For this subsection, we assume that all the polynomials h_1, f_2, g_1, g_2 have **integer coefficients modulo a fixed 32 bit prime number**. The author has measured how the running times of the method described in Step 1 and Step 2 above compare to the running times of computing resultants of partially composed polynomials without utilizing the composition structure of $f_2\circ(g_1,g_2)$. For the rest of this subsection, in order to be able to easily compare both methods, we refer to the first method with "UseStruc" (use the structure via Step 1 and Step 2) and to the second one with "NoStruc" (do not use the structure, expand the composed polynomial and compute the resultant).

The measurements have been taken for random dense g_1's and g_2's of equal degrees ranging from 10 to 30 and for random dense h_1's and f_2's of degrees independently ranging from 10 to 30 as well. This choice of inputs results in a large amount of computations and running times measured. That is, the degrees (c_2, d, e_1) of the inputs range over the set $\{10,\ldots,30\}^3$ and for each triple in the latter set we get running time measurements. In order to make the presentation of the timings more compact, we compute averages of the running times in a systematic way described as follows. For fixed degree e_1 of h_1, we partition the set $\{10,\ldots,30\}^2 \times \{e_1\}$ into small sets of four triples. That is, these partitioning sets are $P_{l,e_1} = \{10+2l,\, 10+(2l+1)\}^2 \times \{e_1\} = \{(10+2l, 10+2l, e_1),\ (10+2l, 10+(2l+1), e_1),\ (10+(2l+1), 10+2l, e_1),\ (10+(2l+1), 10+(2l+1), e_1)\}$. For each triple in P_{l,e_1}, we generate random polynomials of corresponding degrees and measure the running times of methods UseStruc and NoStruc. Then we compute the averages $\text{time}_{l,e_1}^{\text{UseStruc}}$ and $\text{time}_{l,e_1}^{\text{NoStruc}}$, of these measured times for the four triples in P_{l,e_1}. One can observe that these averages vary not very much as e_1 ranges from 10 to 30. Thus we compute the averages $\text{time}_l^{\text{UseStruc}}$ and $\text{time}_l^{\text{NoStruc}}$, for e_1 ranging from 10 to 30, further simplifying the presentation of the running times but still remaining faithful to the experimental measurements. Finally, these values are listed in Table 1.

The author believes that intuitively it is not surprising that the averages $\mathrm{time}_{l,e_1}^{\mathrm{UseStruc}}$ and $\mathrm{time}_{l,e_1}^{\mathrm{NoStruc}}$ vary little for varying e_1. That is, e_1, the degree of the unstructured h_1, is relatively small in comparison to the degree of the composed polynomial $f_2 \circ (g_1, g_2)$. Therefore, changes of e_1 have little impact on the running time. Furthermore, note that in this case utilizing the composition structure is also very efficient computationally. If e_1 becomes larger then the efficiency of Step 1 and Step 2 decreases. This behavior is expected because, intuitively, for large e_1, in comparison to the degree of the composed polynomial $f_2 \circ (g_1, g_2)$, one expects to achieve only little or even no gain in efficiency through utilizing the composition structure of $f_2 \circ (g_1, g_2)$.

l	$\mathrm{time}_l^{\mathrm{NoStruc}}$ in sec.	$\mathrm{time}_l^{\mathrm{UseStruc}}$ in sec. Application of Theorem 1
0	0.763	.025
1	1.320	.027
2	3.059	.027
3	4.902	.028
4	7.675	.030
5	12.414	.031
6	18.843	.031
7	31.393	.033
8	58.322	.035
9	99.768	.036

Fig. 1. Running times for increasing degrees of f_2, g_1, g_2. Averages for (c_2, d, e_1) in $\{10 + 2l,\ 10 + 2l + 1\}^2 \times \{10, 11, \ldots, 30\}$.

In Table 1 one can see that the speedup of Method UseStruc (Theorem 1 applied in Step 1 and Step 2) is quite dramatic as l, i.e. the degrees of f_2, g_1 and g_2, increases.

3 Proof of the main theorem

The main theorem, Theorem 1, consists of two parts. In this paper we only prove the first part and leave out the proof of the second part. That is, we prove Line (2) and Line (3). The author intends to prove the second part, Line (4), in a subsequent publication.

Proof of Line (2) and Line (3) of Theorem 1 We start with an auxiliary lemma.

Lemma 8 *Suppose* $\mathrm{Res}_{e_1,d}(h_1, g_2) \neq 0$. *Then the leading coefficient, with respect to the variable* z, *of the polynomial* $\mathrm{Res}_{e_1,d}(h_1,\ g_1 - z\, g_2)$ *equals the resultant* $\mathrm{Res}_{e_1,d}(h_1, g_2)$ *and the degree in* z *of the polynomial is* e_1.

Proof: Let $p(z) = \mathrm{Res}_{e_1,d}(h_1,\ g_1 - z\, g_2)$. By the bi-homogeneity of the resultant, the degree of p is at most e_1. Therefore, if $p^{\mathrm{h}}(1, 0) \neq 0$, where $p^{\mathrm{h}}(y_1, y_2) = y_2^{e_1}\, p(\frac{y_1}{y_2})$, then the leading coefficient of p is $p^{\mathrm{h}}(1, 0)$ and its degree is e_1. Since $p^{\mathrm{h}}(1, 0) = \mathrm{Res}_{e_1,d}(h_1, g_2) \neq 0$, we have shown the lemma. □

Now we are ready for the next lemma, Lemma 9, which shows Line (2) and Line (3) of Theorem 1.

The proof of Lemma 9 extends and generalizes the proof of Theorem 7 of [22]. Note that there is an interesting difference between the two proofs. The proof of Lemma 9 in a first step shows the lemma for polynomials with symbolic (algebraically independent) coefficients and only in a second step it shows the lemma for polynomials with arbitrary coefficients. Whereas, the proof of Theorem 7 of [22] shows the theorem for polynomials with arbitrary coefficients without any first step dealing with symbolic coefficients (compare Remark 4). This approach allows avoiding case distinctions in the proof.

It is also important to point out that one can find a different extension of the proof of Theorem 7 of [22] in the literature. That is, in [23], McKay and Wang extend the techniques presented in [22] in order to derive a product formula for resultants of two homogeneous composed polynomials (see Remark 6). This extension is different from the one included in the proof of Lemma 9. Moreover, it seems not possible to utilize the extended proof techniques presented in [23] to prove Lemma 9 of the current paper.

Furthermore, note that the proof of Lemma 9 is different from the proofs of the results of other papers ([17], [7], [19], [16], [26], [25], [27]) deriving product formulas for various resultants of composed polynomials.

Lemma 9 *We have*

$$\mathrm{Res}_{e_1,e_2}(h_1, f_2 \circ (g_1, g_2)) \;=\; \mathrm{Res}_{c_1,c_2}(f_1, f_2),$$

where $f_1 \;=\; \mathrm{Res}_{e_1,d}(h_1,\; y_2\, g_1 \;-\; y_1\, g_2)$.

Proof: Let us first assume that all the polynomials h_1, f_2, g_1 and g_2 have distinct symbolic coefficients. Let x be a new variable. Then we have by well known properties of the resultant ([20]) that $\mathrm{Res}_{e_1,e_2}(h_1, f_2 \circ (g_1, g_2)) \;=\; \mathrm{Res}_{e_1,e_2}(h_1(x,1), f_2 \circ (g_1, g_2)(x,1))$. Note that the resultant on the left-hand side of this equality eliminates the variables x_1 and x_2 from two homogeneous polynomials. Whereas, on the right-hand side it eliminates the variable x from two univariate (not necessarily homogeneous) polynomials. Furthermore, let α range over the roots of $h_1(x,1)$. Then, since $g_2(\alpha,1) \neq 0$ and by well known properties of the resultant (see [22], [20]), we have

$$\begin{aligned}
\mathrm{Res}_{e_1,e_2}(h_1, f_2 \circ (g_1, g_2)) &= h_1(0,1)^{c_2 d} \prod_{\alpha} f_2 \circ (g_1, g_2)(\alpha, 1) \\
&= h_1(0,1)^{c_2 d} \prod_{\alpha} f_2(g_1(\alpha,1), g_2(\alpha,1)) \\
&= h_1(0,1)^{c_2 d} \prod_{\alpha} g_1(\alpha,1)^{c_2} \prod_{\alpha} f_2\Big(\frac{g_1(\alpha,1)}{g_2(\alpha,1)}, 1\Big) \\
&= (\mathrm{Res}_{e_1,d}(h_1, g_2))^{c_2} \prod_{\alpha} f_2\Big(\frac{g_1(\alpha,1)}{g_2(\alpha,1)}, 1\Big).
\end{aligned}$$

Now, observe that $\beta \;=\; \frac{g_1(\alpha,1)}{g_2(\alpha,1)}$ for some α iff

$$\prod_{\alpha} (g_1(\alpha,1) - \beta\, g_2(\alpha,1)) \;=\; 0.$$

Since $h_1(1,0)$, the leading coefficient of $h_1(x,1)$, does not vanish, the latter is equivalent to

$$\mathrm{Res}_{e_1,d}(h_1(x,1),\; g_1(x,1) - \beta\, g_2(x,1)) \;=\; 0,$$

which is equivalent to $\mathrm{Res}_{e_1,d}(h_1,\; g_1 - \beta\, g_2) \;=\; 0$. Therefore and by Lemma 8,

$$\begin{aligned}
&\mathrm{Res}_{e_1,e_2}(h_1, f_2 \circ (g_1, g_2)) \;=\; \\
&\quad (\mathrm{Res}_{e_1,d}(h_1, g_2))^{c_2} \;\times \prod_{\substack{\beta \\ \mathrm{Res}_{e_1,d}(h_1,\, g_1 - \beta\, g_2) \,=\, 0}} f_2(\beta, 1) \;=\; \\
&\quad (\mathrm{Res}_{e_1,d}(h_1, g_2))^{c_2} \;\times\; \frac{\mathrm{Res}_{e_1,c_2}(\mathrm{Res}_{e_1,d}(h_1,\; g_1 - y\, g_2),\; f_2(y,1))}{(\mathrm{Res}_{e_1,d}(h_1, g_2))^{c_2}} \;=\; \\
&\quad \mathrm{Res}_{c_1,c_2}(f_1, f_2).
\end{aligned}$$

Therefore we have shown Lemma 9 for polynomials with symbolic coefficients.

Next, observe that the formulas of Lemma 9 are stable under specialization. Therefore Lemma 9 also holds for polynomials with arbitrary coefficients. □

Thus we have shown Line (2) and Line (3), that is, the first part of Theorem 1.

4 Conclusion

This paper has studied resultants of partially composed polynomials. We have found that these resultants are certain iterated resultants of the component polynomials. Furthermore, we saw in experiments that, in many cases, these iterated resultants can be computed with dramatically increased efficiency.

Future research might address multi-variable generalizations of the results of this paper.

References

1. P. H. Berman and M. F. Singer. Calculating the Galois group of $L_1(L_2(y)) = 0$, L_1, L_2 completely reducible operators. *J. Pure Appl. Algebra*, 139(1-3):3–23, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).
2. L. Busé, M. Elkadi, and B. Mourrain. Generalized resultants over unirational algebraic varieties. *J. Symbolic Computation*, 29(4-5):515–526, 2000.
3. J. Canny and I. Emiris. A subdivision-based algorithm for the sparse resultant. *J. ACM*, 47(3):417–451, May 2000.
4. J. Canny, E. Kaltofen, and Lakshman Y. Solving systems of non-linear polynomial equations faster. In *Proc. ACM-SIGSAM 1989 Internat. Symp. Symbolic Algebraic Comput. ISSAC'89*, pages 121–128. ACM, 1989.
5. E. Cattani, A. Dickenstein, and B. Sturmfels. Residues and resultants. *J. Math. Sci. Univ. Tokyo*, 5(1):119–148, 1998.
6. M. Chardin. *Contributions à l'algèbre commutative effective et à la théorie de l'élimination.* PhD thesis, Université Paris VI, 1990.
7. C. C. Cheng, J. H. McKay, and S. S. Wang. A chain rule for multivariable resultants. *Proceedings of the American Mathematical Society*, 123(4):1037–1047, April 1995.
8. D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry.* Springer Verlag, New York, Berlin, Heidelberg, 1998.
9. C. D'Andrea and A. Dickenstein. Explicit formulas for the multivariate resultant. *Pure Appl. Algebra*, 164(1-2):59–86, 2001.
10. I. Z. Emiris and V. Pan. The structure of sparse resultant matrices. In *Proc. Int. Symp. on Symbolic and Algebraic Computation (ISSAC).* ACM Press, 1997.
11. I. M. Gelfand, M. M. Kapranov, and A. V. Zelevinsky. *Discriminants, Resultants and Multidimensional Determinants.* Birkhäuser, Boston, 1994.
12. L. González-Vega. Une théorie des sous-résultants pour les polynômes en plusieurs variables. *C. R. Acad. Sci. Paris Sér. I Math.*, 313/13:905–908, 1991.
13. J. Gutierrez and R. Rubio San Miguel. Reduced Gröbner bases under composition. *J. Symbolic Computation*, 26(4):433–444, 1998.
14. H. Hong. Subresultants under composition. *J. Symbolic Computation*, 23(4):355–365, 1997.
15. H. Hong. Groebner basis under composition I. *J. Symbolic Computation*, 25(5):643–663, 1998.
16. H. Hong and M. Minimair. Sparse resultant of composed polynomials I. *J. Symbolic Computation*, 33:447–465, 2002.
17. J. P. Jouanolou. Le formalisme du résultant. *Adv. Math.*, 90(2):117–263, 1991.
18. D. Kapur and T. Saxena. Sparsity considerations in Dixon resultants. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 184–191, New York, 1996. ACM.
19. D. Kapur and T. Saxena. Extraneous factors in the Dixon resultant formulation. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI)*, pages 141–148, New York, 1997. ACM.
20. F. S. Macaulay. *The algebraic theory of modular systems.* Cambridge Mathematical Library, 1916.
21. D. Manocha and J. Canny. Multipolynomial resultant algorithms. *J. Symbolic Computation*, 15(2):99–122, 1993.
22. J. H. McKay and S. S. Wang. A chain rule for the resultant of two polynomials. *Arch. Math.*, 53(4):347–351, 1989.
23. J. H. McKay and S. S. Wang. A chain rule for the resultant of two homogeneous polynomials. *Arch. Math.*, 56(4):352–361, 1991.
24. M. Minimair. *Sparse Resultants of Composed Polynomials.* PhD thesis, North Carolina State University, Raleigh, NC, USA, 2001.
25. M. Minimair. Sparse resultant of composed polynomials II. *J. Symbolic Computation*, 33:467–478, 2002.

26. M. Minimair. Dense resultant of composed polynomials. *J. Symbolic Computation*, 36(6):825–834, December 2003.
27. M. Minimair. Factoring resultants of linearly combined polynomials. In J. R. Sendra, editor, *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 207–214, New York, NY, 2003. ACM. ISSAC 2003, Philadelphia, PA, USA, August 3-6, 2003.
28. M. Minimair. Sparse resultant under vanishing coefficients. *J. Algebraic Combinatorics*, 18(1):53–73, July 2003.
29. P. Nordbeck. SAGBI bases under composition. *J. Symbolic Computation*, 33(1):67–76, 2002.
30. P. Pedersen and B. Sturmfels. Product formulas for resultants and Chow forms. *Mathematische Zeitschrift*, 214:377–396, 1993.
31. J. M. Rojas. Solving degenerate sparse polynomial systems faster. *J. Symbolic Computation*, 28(1 and 2):155–186, July/August 1999. Special Issue Polynomial Elimination – Algorithms and Applications.