




# Elimination Algorithms for Skew Polynomials with Applications

Raqeeb Rasheed , Ali Safaa Sadiq\*  and Omprakash Kaiwartya 

Department of Computer Science, Nottingham Trent University, Clifton Campus, Nottingham NG11 8NS, UK; raqeeb.rasheed2023@my.ntu.ac.uk, ali.sadiq@ntu.ac.uk, and omprakash.kaiwartya@ntu.ac.uk

\* Correspondence: ali.sadiq@ntu.ac.uk (A.S.S.)

**Abstract:** It is evident that skew polynomials offer promising direction for developing cryptographic schemes. This paper focuses on exploring skew polynomials and studying their properties, with the aim of exploring their potential applications in fields such as cryptography and combinatorics. We begin by deriving the concept of resultant for bivariate skew polynomials. Then, we employ the derived resultant to incrementally eliminate indeterminates in skew polynomial systems, utilising both direct and modular approaches. Finally, we discuss some applications of the derived resultant including cryptographic schemes (such as Diffie–Hellman) and combinatorial identities (such as Pascal’s identity). We start by considering a bivariate skew polynomial system with two indeterminates, our intention is to isolate and eliminate one of the indeterminates to reduce the system to a simpler form (that is relying only on one indeterminate in this case). The methodology is composed of two main techniques; in the first technique, we apply our definition of (bivariate) resultant via a Sylvester style matrix directly from the polynomials’ coefficients, while the second is based on modular methods where we compute the resultant by using evaluation and interpolation approaches. The idea of this second technique is that instead of computing the resultant directly from the coefficients, we propose to evaluate the polynomials at a set of valid points to compute its corresponding set of partial resultants first, then we can deduce the original resultant by combining all these partial resultants using an interpolation technique by utilising a theorem we have established.

**Keywords:** Ore algebra, skew polynomials, elimination, resultant, symbolic computation, modular method, noncommutative algebra.

## 1. Introduction

Naturally, computations with polynomials in multivariate (commutative or noncommutative) polynomial rings are essential in computer algebra and have broad applications in both computer science and mathematics, just to name some areas of interest such as cryptography, combinatorics, and coding theory.

Skew polynomials represent a generalisation of ordinary polynomials, characterised by a (not necessarily commutative) multiplication rule.

This study focuses on bivariate skew polynomials and explores the feasibility of a novel resultant of multivariate skew polynomials, with the aim of exploring their potential applications in fields such as cryptography. Given the ongoing research in this field, it is evident (e.g., [1]) that skew polynomials offer a promising direction for developing new cryptographic schemes due to the increased complexity of computations involving skew polynomials as well as the limited availability of tools and techniques in noncommutative algebras that could be used by attackers to recover or decrypt the information.

Skew polynomials (or Ore polynomials) were first introduced by Ore in 1933 [2]. Chyzak and Salvy studied elimination through Gröbner bases in noncommutative algebra in 1996 [3]. Collins, in 1967 [4], used the Sylvester matrix to compute the determinant in the commutative case. The resultant of univariate skew polynomials was studied by Li in 1998 [5] and by Vesnik and Eric in 2008 [6]. Modular methods from commutative

**Citation:** Rasheed, R.; Shakarchi, A.; Kaiwartya, O. Elimination Algorithms for Skew Polynomials with Applications. *Mathematics* **2024**, *1*, 0. <https://doi.org/>

Received:

Revised:

Accepted:

Published:

**Copyright:** © 2024 by the authors. Submitted to *Mathematics* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

algebra were used for solving nonlinear polynomial systems through resultant by Rasheed in 2007 [7]. Evaluation and interpolation techniques for skew polynomial multiplication were considered by Caruso and Le Borgne in 2017 [8] and by Giesbrecht, Huang, and Schost in 2020 [9]. Fairly recently, Rasheed described resultant-based methods for skew elimination in 2021 [10]. Expanding upon the foundation established in [10], this study offers a comprehensive understanding of the entire process, including the new part of *interpolation stage* which was not addressed in the earlier work [10], along with some potential applications enhancing the study's applicability to relevant use cases.

At its core, this paper presents novel elimination methods for multivariate (initially bivariate) skew polynomial systems, transforming them to another (simpler) than the originals.

Working with skew polynomials (in noncommutative algebra) poses various challenges. Not only does the noncommutative nature of the algebra introduce its own challenges, but also essential tools for elimination (such as resultant) is not available for multivariate skew polynomials (not even for the bivariate case) until fairly recently, as described in [10]. Additionally, the evaluation map is not common in the noncommutative algebra as it does not preserve multiplication. Determinant is another challenge that does not have a standard formulation for computing a unique determinant. We need to discuss noncommutative analogs of these components to address the gaps. Sections 4.2 and 4.3 will explore these challenges, as well as any additional ones encountered, along with the techniques we used to overcome each one.

**What is new:** In comparison to the previous study [10], the main improvements and additions in this new version are:

First, the study derives the resultant formula and establishes a theorem (along with its proof) stating that the resultant of two operator polynomials annihilates any common solution of those polynomials.

Second, the interpolation part, which was briefly mentioned in the previous study [10] and left for future work, is now thoroughly examined by addressing and overcoming the challenges associated with it, this step is illustrated by examples.

Third, the feasibility of this research approach is studied and applied to some applications. An example is the use of one of the established theorems to identify new identities with fewer indeterminates, as demonstrated in a provided example. Furthermore, the techniques developed in this study can be used to establish a Diffie-Hellman key exchange between two users, a widely recognized cryptographic protocol for secure key exchange over a public channel.

## 2. Background

This part provides background information for this paper.

### 2.1. Ore polynomial rings

The use of Ore polynomial rings is a convenient way of unifying some classes of (noncommutative) polynomial rings, for example it can analogously be applied to linear differential equations, linear difference equations, and other similar substances. A benefit of this model is that all the operations can be studied and implemented once, then they can suitably be applied to all the substances. The following is the definition of Ore polynomial ring [2].

**Definition 2.1.1** (Ore polynomial ring). *Let  $\sigma$  be a ring automorphism of a (skew) field  $\mathcal{F}$  and  $\delta$  be a  $\sigma$ -derivation of  $\mathcal{F}$ . The Ore polynomial ring  $\mathcal{F}[\theta; \sigma, \delta]$  is the set of polynomials in indeterminate  $\theta$  over  $\mathcal{F}$  with the usual polynomial addition and noncommutative multiplication defined as*

$$\theta a = \sigma(a)\theta + \delta(a), \quad \forall a \in \mathcal{F}. \quad (1)$$

Elements of  $\mathcal{F}[\theta; \sigma, \delta]$  are called (univariate) Ore polynomials or Ore operators. 87

Table 1 contains some examples of Ore operators with their commutation rules of  $\theta t$  over a (skew) field  $\mathcal{F}$ , for more of such examples please see [11]. 88  
89

$\mathcal{F}$	Operator	$\sigma(a(t))$	$\delta(a(t))$	Commutation rule of $\theta t$
$\mathcal{K}(t)$	Differential	$a(t)$	$a'(t) = \frac{d}{dt}(a(t))$	$t\theta + 1$
	Difference	$a(t+1)$	$a(t+1) - a(t)$	$(t+1)\theta + 1$
	Shift	$a(t+1)$	0	$(t+1)\theta$
	Eulerian	$a(t)$	$ta'(t)$	$t\theta + t$
$\mathcal{K}(q, t)$ $q \in \mathbb{Q} \setminus \{0, 1, -1\}$	q-differential	$a(qt)$	$\frac{a(qt) - a(t)}{(q-1)t}$	$qt\theta + 1$
	q-difference	$a(qt)$	$a(qt) - a(t)$	$qt\theta + (q-1)t$
	q-shift	$a(qt)$	0	$qt\theta$

**Table 1.** Some examples of Ore operators in  $\mathcal{K}(t)[\theta; \sigma, \delta]$ , for some field  $\mathcal{K}$ .

**Remark 2.1.2.** The case when  $\delta = 0$  in Definition 2.1.1 is called skew polynomial ring\* and denoted by  $\mathcal{F}[\theta, \sigma]$ . The ordinary commutative polynomial ring is a special case of Ore polynomial ring when  $\sigma$  is an identity map of a commutative ring  $\mathcal{F}$  in addition to  $\delta = 0$ , which means the results in this study can be applied to the commutative case as well. 90  
91  
92  
93

We can construct a bivariate Ore polynomial ring by iterating the univariate case of Definition 2.1.1 to have a ring of Ore polynomials in two indeterminates with coefficients in the univariate Ore polynomial ring. This process can be extended to have multivariate Ore polynomials in general as defined below for the case when  $\delta = 0$ , the specific case under consideration in this paper. The following definition is used in [10, Definition 2]; similar definitions can be found in [12, Definition 46.13.1, p. 85] and [13, Note 3.16, p. 28]. 94  
95  
96  
97  
98  
99

**Definition 2.1.3.** A multivariate skew polynomial ring over  $\mathcal{F}$  is the iterated skew polynomial ring  $\mathcal{S} = \mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2] \cdots [\theta_n; \sigma_n]$  with commuting indeterminates  $\theta_i$ , and automorphisms  $\sigma_i$  of  $\mathcal{F}$  that satisfy the following relations; 100  
101  
102

$$\sigma_j(\theta_i) = \theta_i \ (i \neq j), \ \sigma_j \sigma_i = \sigma_i \sigma_j \ \text{and} \ \theta_i a = \sigma_i(a) \theta_i, \ \text{for all } 1 \leq i, j \leq n \ \text{and} \ a \in \mathcal{F}. \quad (2)$$

Note that a skew polynomial ring  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2] \cdots [\theta_n; \sigma_n]$  is sometimes written in a recursive form such that, for all  $i = 1, \dots, n$ ,  $\mathcal{S}_i = (\mathcal{S}_{i-1})[\theta_i; \sigma_i, \delta_i]$  where  $\mathcal{S}_0 = \mathcal{F}$ . For example, we may write  $(\mathcal{F}[\theta_1; \sigma_1])[\theta_2; \sigma_2]$  instead of  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2]$ , which is our primary initial focus. We assume that  $\mathcal{F}[\theta_1; \sigma_1]$  is a domain, unless noted otherwise. 103  
104  
105  
106

Also, in this study all Ore polynomials are left Ore polynomials which are in the form  $f = \sum_{i=0}^n a_i \theta^i$  in  $\mathcal{F}[\theta; \sigma]$ . Let  $\mathcal{S} = \mathcal{F}[\theta; \sigma]$  and consider  $\mathcal{V}$  to be an  $\mathcal{S}$ -module. Any pseudo linear map  $\varphi: \mathcal{V} \rightarrow \mathcal{V}$  can create an action operation (denoted by  $\bullet$ ) on a member  $\mathbf{u}$  in  $\mathcal{V}$  as follows: 107  
108  
109  
110

$$\begin{aligned} & \mathcal{F}[\theta; \sigma] \times \mathcal{V} \rightarrow \mathcal{V} \\ \bullet : & \quad f(\theta) \bullet \mathbf{u} = \left( \sum_{i=0}^n a_i \theta^i \right) \bullet \mathbf{u} \mapsto f(\varphi)(\mathbf{u}) = \sum_{i=0}^n a_i \varphi^i(\mathbf{u}). \end{aligned} \quad (3)$$

Sometimes the dot  $\bullet$  is omitted for simplicity. If  $f(\varphi)(a) = 0$  for some  $a \in \mathcal{F}$  then we say  $\mathbf{u} = a$  is a solution of  $f(\varphi)(\mathbf{u}) = 0$ . Similarly, for the bivariate case, when we have the 111  
112

\*The term skew polynomial ring may refer to a different ring in some references.

map  $f(\theta_1, \theta_2) \cdot \mathbf{u} \mapsto f(\varphi_1, \varphi_2)(\mathbf{u})$ , for some pseudo linear maps  $\varphi_1$  and  $\varphi_2$ , then  $\mathbf{u} = a$  is a solution of  $f$  if  $f(\varphi_1, \varphi_2)(\mathbf{u}) = 0$ .

## 2.2. Euclidean domain

In this section, we review the definition of a Euclidean domain and illustrate that bivariate skew polynomials along with their multivariate extensions (involving two or more indeterminates) do not satisfy the properties of a Euclidean domain.

**Definition 2.2.1.** A (not necessarily commutative) domain  $\mathcal{R}$ , endowed with a map

$$N : \mathcal{R} \setminus \{0\} \rightarrow \mathbb{N}_0,$$

is a right Euclidean domain with respect to  $N$  if the following properties hold for all  $f, g \neq 0 \in \mathcal{R}$

(i) there exist  $q, r \in \mathcal{R}$  such that

$$f = gq + r, \text{ where } r = 0 \text{ or } N(r) < N(g), \quad (4)$$

(ii)  $N(f) \leq N(fg)$ .

The elements  $q$  and  $r$  in (4) are called *right quotient* (denoted by  $\text{rquo}$ ) and *right remainder* (denoted by  $\text{rrem}$ ), respectfully, of the right division of  $f$  by  $g$ . If  $r = 0$  then  $g$  is called a *right divisor* of  $f$  in  $\mathcal{R}$ . The left Euclidean domain uses analogous conventions and notations. A ring that is both left and right Euclidean domain is called a *Euclidean domain*.

In the context of a skew polynomial ring  $\mathcal{R} = \mathcal{F}[\theta; \sigma]$ , where  $N$  denotes the degree  $\deg$ , if  $f$  and  $g$  are elements of  $\mathcal{F}[\theta; \sigma]$ , we can obtain a quotient  $q$  and a remainder  $r$  using the Euclidean division algorithm, which depends on the presence of invertible elements in  $\mathcal{F}$ . However, if the underlying ring of coefficients is not a division ring, the Euclidean division algorithm cannot be applied as illustrate in the follow example.

**Example 2.2.2.** Let  $\mathcal{R} = (\mathcal{F}[\theta_1; \sigma_1])[\theta_2; \sigma_2]$  be a bivariate skew polynomial ring. Consider the attempt to divide  $f = \theta_2$  by  $g = \theta_1$ , where we seek  $q, r \in \mathcal{R}$  such that  $\theta_2 = \theta_1q + r$  and  $\deg_{\theta_2}(r) < \deg_{\theta_2}(\theta_1) = 0$ . Since  $\deg_{\theta_2}(\theta_2) = 1$  and  $\deg_{\theta_2}(\theta_1) = 0$ , the condition  $\deg_{\theta_2}(r) < 0$  is impossible even if  $\mathcal{F}$  is a field. Therefore, no such  $q$  and  $r$  exist, indicating that  $\mathcal{R}$  is not a Euclidean domain.

Section 3.2 further examines this issue and presents a method for a key property related to relation (8).

## 3. Elimination

Many models of mathematical physics and engineering can be described in terms of extra indeterminates (or extra parameters) which are present usually in mixed forms, then it becomes helpful to eliminate one or more of these extra indeterminates through an elimination method.

In this section we will study how our approaches can be used to eliminate such extra indeterminates in a system of bivariate skew polynomial equations while retaining the original model's behaviour, i.e. retaining the way the system acts. Furthermore, we show how to use a Euclidean relation to derive a formula that can compute the resultant directly from the polynomial's coefficients, generalising the commutative case of the Sylvester's determinant and the noncommutative univariate case of [6], then we describe elimination methods of bivariate skew polynomials based on our resultant computations. Consequently, a suitable noncommutative formulation of determinant is needed and for this we use the *Dieudonné determinant*.

### 3.1. Dieudonné determinant of matrices with skew polynomial entries

From the perspective of normal forms of matrices, one can think of diagonalising (or triangularising) an invertible  $n \times n$  matrix  $A$  over  $\mathcal{F}[\theta; \sigma]$  to obtain

$$D = UAV = \text{diag}(d_1, \dots, d_n), \quad (5)$$

where  $U$  and  $V$  are *unimodular matrices*,  $D$  is a diagonal matrix with diagonal entries  $d_i \in \mathcal{F}(\theta; \sigma)$ ,  $i = 1, \dots, n$ . Knowing that the Dieudonné determinant of a diagonal (or a triangular matrix) is the product of the diagonal entries (see for example, [14] p. 822 or [15] p. 3 Example 2), we can obtain the Dieudonné determinant of  $D$  as

$$\det(D) = \prod d_i. \quad (6)$$

Please note that in general, the entries  $d_i$  may become rational functions in  $\mathcal{F}(\theta; \sigma)$  rather than remaining polynomials in  $\mathcal{F}[\theta; \sigma]$ . However, for the purpose of this study, we need to perform the computations in such a way that the entries remain polynomials, especially the diagonal entries  $d_i$ . This ensures that their product  $\prod d_i$ , which is the determinant, also remains a polynomial in  $\mathcal{F}[\theta; \sigma]$ , and this can be achieved according to the following lemma (from [10, Lemma 11]).

**Lemma 3.1.1.** *Dieudonné determinant of a matrix  $A \in \mathcal{F}[\theta; \sigma]^{n \times n}$  can be represented by a unique skew polynomial (modulo commutators).*

**Remark 3.1.2.** *To compute the Dieudonné determinant in Lemma 3.1.1, one approach is to use elementary row operations by using Ore condition in order to transform the matrix into a diagonal (or a triangular) form. Thus, the determinant can be computed by simply multiplying the elements on the main diagonal. For more details please see [10].*

Another useful property of the Dieudonné determinant is that it does not depend on the choice of elementary row operations, neither on the order in the product of diagonal entries of the matrix, despite the noncommutative nature of the entries (see for example, [14] p. 822). Furthermore, an important feature of Dieudonné determinant is that it preserves multiplication; that is

$$\det(AB) = \det(A) \det(B), \quad (7)$$

for any two invertible matrices  $A, B$  over  $\mathcal{F}[\theta; \sigma]$ .

### 3.2. Deriving a resultant formula for bivariate skew polynomials

It is evident that we can employ the *extended Euclidean algorithm* in its skew version (see Algorithm 1) to find the greatest common right divisor, denoted by  $\text{gcdr}$ . Additionally, the same algorithm, can be used (by permitting  $r_{i-1} = 0$  in the algorithm) to derive the following relation for any two skew polynomials  $f$  and  $g$  in  $\mathcal{F}[\theta; \sigma]$ ;

$$u f = v g, \quad (8)$$

for some non zeros  $u, v \in \mathcal{F}[\theta; \sigma]$  such that degrees of  $u$  and  $v$  are less than the degrees of  $g$  and  $f$  respectively.

**Algorithm 1:** Skew extended euclidean algorithm

---

**Input:**  $f, g \in \mathcal{F}[\theta; \sigma]$ , where  $\mathcal{F}$  is a (skew) field  
**Output:**  $d \in \mathcal{F}[\theta; \sigma]$ , where  $d$  is a gcd of  $f$  and  $g$  together with  $s, t \in \mathcal{F}[\theta; \sigma]$  such that  $sf + tg = d$

```

1   $r_0 := f; s_0 := 1; t_0 := 0$ 
2   $r_1 := g; s_1 := 0; t_1 := 1$ 
3   $i := 2$ 
4  while  $r_{i-1} \neq 0$  repeat
5       $q_i := r_{i-2} \text{ rquo } r_{i-1}$  // rquo is the right quotient
6       $r_i := r_{i-2} \text{ rrem } r_{i-1}$  // rrem is the right remainder
7       $s_i := s_{i-2} - q_i s_{i-1}$ 
8       $t_i := t_{i-2} - q_i t_{i-1}$ 
9       $i := i + 1$ 
10 return  $(r_{i-2}, s_{i-2}, t_{i-2})$ 

```

---

While effective for univariate cases, the algorithm fails for the bivariate cases in  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2]$  due to the domain not being a Euclidean domain anymore. 186

Now, let's examine the relation (8) more closely in the bivariate case  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2]$  as  $f = \sum_{j=0}^n a_j \theta_2^j$ ,  $g = \sum_{j=0}^m b_j \theta_2^j$ ,  $u = \sum_{i=0}^{m-1} u_i \theta_2^i$ , and  $v = \sum_{i=0}^{n-1} v_i \theta_2^i$ , with polynomial coefficients  $a_j, b_j, u_i$ , and  $v_i$  in  $\mathcal{F}[\theta_1; \sigma_1]$ , which then the relation  $uf = vg$  becomes 187 188 189 190 191

$$\sum_{i=0}^{m-1} \sum_{j=0}^n u_i \sigma^i(a_j) \theta_2^{i+j} = \sum_{i=0}^{n-1} \sum_{j=0}^m v_i \sigma^i(b_j) \theta_2^{i+j}. \quad (9)$$

Thus, by comparing the coefficients on both sides of (9), we can obtain a system of  $n + m$  equations for the unknowns  $u_i$  ( $i = 0, \dots, m - 1$ ) and  $-v_i$  ( $i = 0, \dots, n - 1$ ) as follows: 192 193

$$\left\{ \begin{array}{l} u_{m-1} a_n^{[m-1]} = v_{n-1} b_m^{[n-1]} \\ u_{m-1} a_{n-1}^{[m-1]} + u_{m-2} a_n^{[m-2]} = v_{n-1} b_{m-1}^{[n-1]} + v_{n-2} b_m^{[n-2]} \\ u_{m-1} a_{n-2}^{[m-1]} + u_{m-2} a_{n-1}^{[m-2]} + u_{m-3} a_n^{[m-3]} = v_{n-1} b_{m-2}^{[n-1]} + v_{n-2} b_{m-1}^{[n-2]} + v_{n-3} b_m^{[n-3]} \\ \vdots \\ u_1 a_0^{[1]} + u_0 a_1^{[0]} = v_1 b_0^{[1]} + v_0 b_1^{[0]} \\ u_0 a_0^{[0]} = v_0 b_0^{[0]} \end{array} \right. \quad (10)$$

where, for each  $i = 1, \dots, m$ , the sequence  $a_j^{[m-i]}$  ( $j = n, \dots, 0$ ) represents the coefficients of the multiplication  $\theta_2^{m-i} f$ , while for each  $i = 1, \dots, n$ , the sequence  $b_j^{[n-i]}$  ( $j = m, \dots, 0$ ) represents the coefficients of the multiplication  $\theta_2^{n-i} g$ . It is worth noting that they are free of the indeterminate  $\theta_2$ . Accordingly, an associate determinant for the system (10) can be formulated as follows: 194 195 196 197 198





Recall that the indeterminates  $\theta_1$  and  $\theta_2$  do not commute with the coefficients but rather act according to the ring automorphisms  $\sigma_1$  and  $\sigma_2$  such that for each  $a \in \mathcal{F}$ ,

$$\theta_1 a = \sigma_1(a)\theta_1 \quad \text{and} \quad \theta_2 a = \sigma_2(a)\theta_2, \quad (12)$$

which means the noncommutative properties in the determinants' entries are properly performed as the rows are multiplied on the left by  $\theta_2$  to some power.

The difference between this definition (Definition 3.2.2) and the definition used in [6] is that we consider the case of *bivariate* skew polynomials with *two commuting indeterminates* which is particularly suitable for Ore algebra (the framework of this research). This is a new combination from [6] that allows elimination of indeterminates in the general case (with  $n > 1$  indeterminates) including an elimination method using an *operator evaluation map*, which is the aim of this study.

**Remark 3.2.3.** *The determinant described in Definition 3.2.2 requires entries to be in a (skew) field, this can be obtained by embedding  $\mathcal{F}[\theta_1, \sigma_1]$  in a (skew) field [16, Corollary 0.7.2], written as  $\mathcal{F}(\theta_1, \sigma_1)$ , and its elements can be in the form of  $g^{-1}f$  where  $f, g \in \mathcal{F}$ , in which case the degree is defined as*

$$\deg(g^{-1}f) = \deg(f) - \deg(g). \quad (13)$$

While the Dieudonné determinant is only unique up to a multiple of some commutators, its degree is well defined and is always the same value, this was pointed out by Taelman [17] as the degree is always zero on commutators.

Note that the involvement of commutator factors will be encountered when dealing with the elimination process in the noncommutative case only. Let's illustrate this phenomenon with the following simple system of two polynomial equations of the first degree with respect to the indeterminate  $\theta$  as:

$$\begin{cases} a_1\theta + a_0 = 0 \\ b_1\theta + b_0 = 0 \end{cases} \quad (14)$$

where  $a_0, a_1, b_0$ , and  $b_1$  are elements in a (skew) field that they do not commute with  $\theta$ . Assume  $a_1, b_1 \neq 0$  (otherwise the case is straightforward).

Now, multiply the first equation by  $-b_1a_1^{-1}$  and add it to the second equation to obtain

$$b_0 - b_1a_1^{-1}a_0 = 0,$$

multiply by  $a_1 \neq 0$ ;

$$a_1(b_0 - b_1a_1^{-1}a_0) = 0. \quad (15)$$

Similarly, multiply the second equation of system (14) by  $-a_1b_1^{-1}$  and add it to the first equation to obtain

$$a_0 - a_1b_1^{-1}b_0 = 0,$$

which can be written as

$$b_1(a_1b_1^{-1}b_0 - a_0) = 0, \quad b_1 \neq 0. \quad (16)$$

In the following we show that the left sides of the two obtained equations (15) and (16) are different by a factor of type commutators, i.e. they become the same mod commutators, denoted by mod  $\mathcal{C}$ .

$$\begin{aligned} \text{l.s. of equation (16)} &= b_1a_1b_1^{-1}b_0 - b_1a_0 \\ &= b_1a_1b_1^{-1}a_1^{-1}a_1b_0 - b_1a_0 \\ &= \mathbf{c}^{-1}(a_1b_0 - \mathbf{c}b_1a_0), \quad \text{where } \mathbf{c} = a_1b_1a_1^{-1}b_1^{-1} \\ &= a_1b_0 - \mathbf{c}b_1a_0 \quad (\text{mod } \mathcal{C}) \end{aligned}$$



$$\begin{aligned}
&= a_1 b_0 - a_1 b_1 a_1^{-1} b_1^{-1} b_1 a_0 \pmod{\mathcal{C}} \\
&= a_1 b_0 - a_1 b_1 a_1^{-1} a_0 \pmod{\mathcal{C}} \\
&= a_1 (b_0 - b_1 a_1^{-1} a_0) \pmod{\mathcal{C}} \\
&= \text{l.s. of equation (15)} \pmod{\mathcal{C}}.
\end{aligned}$$

This is also true for the general  $n \times n$  case. Note that in the commutative case, the commutator factor  $\mathbf{c} = a_1 b_1 a_1^{-1} b_1^{-1}$  is always 1 and hence in the commutative case there is no need to involve commutators. Essentially, a similar phenomenon happens for the determinant used in this study (Dieudonné determinant) which is unique up to commutators (as mentioned).

In the following section, we consider computations in *almost commutative* rings where the Dieudonné determinant becomes well-defined (i.e., it becomes unique).

### 3.3. Uniqueness of the resultant

As noted, our resultant relies on the Dieudonné determinant and as we have seen the Dieudonné determinant can be computed by multiplying its diagonal entries in any order, which is only unique up to an *undesirable factor of products of commutators*. However, in some rings, this factor does not change its effect on the determinant value.

In the context of graded rings (which will be described shortly), we show that a computation that satisfies a property within the graded ring can be sufficient to prove that the same computation holds that property in the original ring. In the following, we discuss a suitable graded ring to obtain a well defined determinant (i.e. to assign a unique value to it), and for this, we need the following two definitions.

**Definition 3.3.1.** A not necessarily commutative ring  $\mathcal{S}$  is called *filtered* if for an indexed family of additive subgroups  $\mathcal{S}_i$  we have

$$\bigcup_i \mathcal{S}_i = \mathcal{S}, \quad \mathcal{S}_i \subseteq \mathcal{S}_{i+1}, \quad \mathcal{S}_i \mathcal{S}_j \subseteq \mathcal{S}_{i+j}, \quad \text{and } 1 \in \mathcal{S}_0,$$

for any  $i, j \in \mathbb{Z}$ , or its special case  $i, j \in \mathbb{Z}_{\geq 0}$  when  $\mathcal{S}_{-1} = 0$ .

**Definition 3.3.2.** Let  $\mathcal{S} = \bigcup_i \mathcal{S}_i$  be a filtered ring. The associated graded ring is denoted by  $\text{gr}(\mathcal{S})$  and defined as

$$\bigoplus_i \mathcal{S}_i / \mathcal{S}_{i-1},$$

such that for all  $r \in \mathcal{S}_i$  and  $s \in \mathcal{S}_j$ ,

$$(r + \mathcal{S}_{i-1})(s + \mathcal{S}_{j-1}) = (rs + \mathcal{S}_{i+j-1}).$$

For each  $r \in \mathcal{S}_i$ , let us denote the image of  $r$  in  $\mathcal{S}_i / \mathcal{S}_{i-1}$  by  $\tilde{\sigma}_i(r)$ , or simply by  $\tilde{\sigma}(r)$  when  $r \in \mathcal{S}_i \setminus \mathcal{S}_{i-1}$  (i.e. if  $r$  is in  $\mathcal{S}_i$  but not in  $\mathcal{S}_{i-1}$ ) also when it is clear from the context. Note that if  $\tilde{\sigma}(r)\tilde{\sigma}(s) \neq 0$  then

$$\tilde{\sigma}(rs) = \tilde{\sigma}(r)\tilde{\sigma}(s), \quad \forall r, s \in \mathcal{S}. \quad (17)$$

For details about filtration and graded rings, see for example [18, Chapter 1, §6]. In the following two subsections, we discuss the uniqueness of our resultant.

#### 3.3.1. Almost commutative rings

Following a similar concept in terms of differential operators, we call a not necessarily commutative ring  $\mathcal{S}$  an *almost commutative ring* if the associated graded ring  $\text{gr}(\mathcal{S})$  is commutative (see for example [19, §3.3]). Under this assumption we will have a rather well defined representation for the resultant of a matrix  $M$  with entries in  $\mathcal{S}$  by considering

computations in  $\text{gr}(\mathcal{S})$  [10], assuming that  $\text{gr}(\mathcal{S})$  is a unique factorization domain and  $\mathcal{S}$  can be embedded in a (skew) field.

Essentially, we compute the resultant as before, by transforming its matrix, let's call it  $M$ , to a diagonal (or a triangular) form then multiplying the diagonal elements (in any order) and fix the obtained value by denoting it as  $\widetilde{\det}(M)$ . Note that, multiplying these elements in any other order will result in a value that is only differ by a multiplication of a factor of the type  $fgf^{-1}g^{-1}$  ( $f, g \in \mathcal{S} \setminus \{0\}$ ) but since  $\mathcal{S}$  is assumed to be almost commutative, all those other values will be the same when written as  $\tilde{\sigma}(\widetilde{\det}(M))$ . Hence, in this case, the determinant has a well defined value, which in turn means that  $\tilde{\sigma}(\widetilde{\det}(M))$  has equivalent properties to the corresponding one of the usual commutative case. By applying this concept to our resultant, we can observe that all the values of the resultant are the same in almost commutative rings.

Considering computations in associated graded rings is particularly useful for homogeneous polynomials or with polynomials when their highest-degree components are the only important parts (similar to the use of the principal symbol in the context of differential operators).

### 3.3.2. Hermite form

The Hermite form for invertible matrices is a canonical matrix representation whose entries reside within either commutative or noncommutative rings [20]. It satisfies the following properties;

- (i) it is an upper triangular matrix, serving as a normal form of the original matrix,
- (ii) the diagonal entries are monic, that is the leading coefficient is 1 for each of them,
- (iii) the degrees of the off-diagonal entries are strictly less than the corresponding degree of the diagonal entry in the same column.

This normal form offers two main advantages; it provides a *unique* representation for the original matrix, and it can be computed *efficiently* [20]. The uniqueness property of this form is particularly valuable for our purpose, as it ensures a unique representation of a Sylvester matrix upon transformation, thus ensuring a consistent value for our resultant.

Next, we turn our attention towards operator elimination techniques using our proposed resultant.

### 3.4. Operator elimination

The focus of this section is to study a process that will allow us to reduce an operator system to a more manageable and tractable alternative to the original system, that is to exclude selected indeterminates that perhaps deemed irrelevant, while retaining those that are of interest. Let us illustrate what we mean by looking at the following well known parametric families of functions called *orthogonal polynomials* [21], such as Chebyshev and Legendre polynomials.

**Example 3.4.1.** Consider the Chebyshev polynomial

$$T_n(t) = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2k} (t^2 - 1)^k t^{n-2k},$$

with variable  $t$  and parameters  $n$  and  $k$ , which satisfy the relations:

$$\begin{aligned} (1 - t^2)T'_{n+1}(t) - (n + 1)tT_{n+1}(t) - (n + 1)T_n(t) &= 0, \\ T_{n+2}(t) - 2tT_{n+1}(t) + T_n(t) &= 0, \\ (1 - t^2)T''_n(t) - tT'_n(t) + n^2T_n(t) &= 0, \end{aligned}$$

they can be translated to the operators language of differential ( $D_t$ ) and difference ( $S_n$ ) in  $\mathbb{Q}[n, t][[D_t; 1, D_t][S_n; S_n, 0]]$  as following

(i) with a mix of differential and difference operators

$$(1 - t^2)D_t S_n - (n + 1)tS_n - (n + 1), \quad (18)$$

(ii) with difference operator only

$$S_n^2 - 2tS_n + 1, \quad (19)$$

(iii) with differential operator only

$$(1 - t^2)D_t^2 - tD_t + n^2, \quad (20)$$

where each of the above relations (18), (19), and (20) annihilate  $T_n(t)$  assuming

$$\begin{aligned} D_t(T_n(t)) &= T_n'(t), \\ S_n(T_n(t)) &= T_{n+1}(t). \end{aligned}$$

Let us consider only (18) and (19) as the following operator system:

$$\begin{cases} F &= (1 - t^2)D_t S_n - (n + 1)tS_n - (n + 1) \\ G &= S_n^2 - 2tS_n + 1. \end{cases} \quad (21)$$

Now, it will be convenient to have a method to eliminate the indeterminate  $S_n$  from  $F$  and  $G$  in order to obtain the relation (20), that is to have a relation with a pure operator of  $D_t$ .

Additionally, if we have a system consisting of the operator relations (18) and (20) then we can think of eliminating  $D_t$  in order to have a relation with only  $S_n$ .

In a similar manner, other orthogonal polynomials can also be expressed in the language of (mixed) operators, for example, the *Legendre polynomial*

$$P_n(t) = \frac{1}{2^n} \sum_{k=0}^n \binom{n}{k}^2 (t - 1)^{n-k} (t + 1)^k,$$

satisfies the following different forms of operator relations:

$$\begin{aligned} (1 - t^2)D_t + (n + 1)S_n - (n + 1)t, \\ (n + 2)S_n^2 - (2n + 3)tS_n + (n + 1), \\ (t^2 - 1)D_t^2 + 2tD_t - n(n + 1). \end{aligned}$$

In other instances, they may appear as multiple parameters with the same operator type as we will see in Example 3.4.5. (for more details on the orthogonal polynomials and other types of special polynomials with their relations see for example [21]).

Therefore, a method that would allow us to simplify an operator system by excluding unwanted indeterminates would be beneficial. The next theorem will look at how the resultant (Definition 3.2.2) can help with this, where our resultant can annihilate the same solution that one can obtain from solving the operator system, in general.

Before we can state and prove the theorem, we need to establish a property that relates the resultant to the original polynomials. In the commutative case, such a property exists, where the resultant of two polynomials can be expressed as the sum of the products of the original polynomials each multiplied by a suitable polynomial. The conventional method (e.g. [22,23]) for proving this property involves rewriting the original polynomials in terms of matrix representations, this allows the utilisation of determinant computations using Cramer's rule where the proof proceeds by expanding the determinant along a selected row (or column) of the matrix. Unfortunately, the Dieudonné determinant can not be expanded by cofactors. For instance, consider a  $2 \times 2$  matrix  $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ , over a skew field, where the matrix elements do not commute. In this scenario, an attempt to obtain the determinant





Now we are ready to prove the following theorem regarding the belonging of resultant to the annihilating ideal  $\text{Ann } \mathbf{u}$ , when  $\mathbf{u}$  is annihilated by the resultant's input polynomials.

**Theorem 3.4.4.** *The resultant of a system of two bivariate skew polynomials annihilates the solution of the system.*

**Proof.** Let  $r = \text{res}_{\theta_2}(f, g)$  be the resultant of bivariate skew polynomials  $f$  and  $g$  in a skew polynomial ring  $\mathcal{S}$ . From Proposition 3.4.2 we now know that there are polynomials  $p$  and  $q$  in  $\mathcal{S}$  such that

$$p f + q g = r. \quad (26)$$

Let  $\mathbf{u}$  be the solution of the system, that is  $f \cdot \mathbf{u} = 0$  and  $g \cdot \mathbf{u} = 0$ . When applying this to the Equation (26) yields

$$\begin{aligned} r \cdot \mathbf{u} &= (p f + q g) \cdot \mathbf{u} \\ &= (p f) \cdot \mathbf{u} + (q g) \cdot \mathbf{u} \\ &= p \cdot (f \cdot \mathbf{u}) + q \cdot (g \cdot \mathbf{u}) \\ &= p \cdot 0 + q \cdot 0 \\ &= 0. \end{aligned}$$

Therefore,  $r = \text{res}_{\theta_2}(f, g)$  belongs to  $\text{Ann } \mathbf{u}$ .

□

The following example illustrates the theorem.

**Example 3.4.5.** *Consider the binomial coefficient*

$$C(n, k) = \binom{n}{k}$$

which satisfies the Pascal identity

$$\binom{n+1}{k+1} - \binom{n}{k+1} - \binom{n}{k} = 0, \quad (27)$$

in addition to

$$(k+1) \binom{n}{k+1} - (n-k) \binom{n}{k} = 0. \quad (28)$$

The above relations (27) and (28) can be rewritten in the shift operator notations  $S_n$  and  $S_k$  to form the following operator system:

$$\begin{cases} F &= S_n S_k - S_k - 1 \\ G &= (k+1) S_k - (n-k). \end{cases} \quad (29)$$

Now, to obtain another relation relying on just  $S_n$ , we can employ our resultant (Definition 3.2.2) to eliminate  $S_k$  from the system (29), and to achieve that we can view the system as a bivariate skew polynomial system in  $\mathbb{Q}(\alpha, \beta)[\theta_1; \sigma_1][\theta_2; \sigma_2]$  as follows

$$\begin{cases} f &= \theta_1 \theta_2 - \theta_2 - 1 \\ g &= (\beta + 1) \theta_2 - (\alpha - \beta), \end{cases} \quad (30)$$

where  $\sigma_1$  and  $\sigma_2$  are the shift operators  $S_n$  and  $S_k$ , respectively. Also,  $\theta_1, \theta_2, \alpha$  and  $\beta$  are  $S_n, S_k, n$  and  $k$ , respectively. Accordingly, we can use the resultant method through the Dieudonné determinant to compute  $\text{res}_{\theta_2}(f, g)$  as following

$$\begin{aligned}
\text{res}_{\theta_2}(f, g) &= \begin{vmatrix} \theta_1 - 1 & -1 \\ \beta + 1 & -(\alpha - \beta) \end{vmatrix} \\
&\xrightarrow{\text{row}_1 \leftrightarrow \text{row}_2} \begin{vmatrix} \beta + 1 & -(\alpha - \beta) \\ \theta_1 - 1 & -1 \end{vmatrix} \\
&\xrightarrow{-(\theta_1 - 1)(\beta + 1)^{-1} \text{row}_1 + \text{row}_2} \begin{vmatrix} \beta + 1 & -(\alpha - \beta) \\ 0 & (\theta_1 - 1)(\beta + 1)^{-1}(\alpha - \beta) - 1 \end{vmatrix} \\
&= \begin{vmatrix} \beta + 1 & -(\alpha - \beta) \\ 0 & (\theta_1(\beta + 1)^{-1} - (\beta + 1)^{-1})(\alpha - \beta) - 1 \end{vmatrix} \\
&= \begin{vmatrix} \beta + 1 & -(\alpha - \beta) \\ 0 & (\sigma_1((\beta + 1)^{-1})\theta_1 - (\beta + 1)^{-1})(\alpha - \beta) - 1 \end{vmatrix} \\
&= \begin{vmatrix} \beta + 1 & -(\alpha - \beta) \\ 0 & ((\beta + 1)^{-1}\theta_1 - (\beta + 1)^{-1})(\alpha - \beta) - 1 \end{vmatrix} \\
&= \begin{vmatrix} \beta + 1 & -(\alpha - \beta) \\ 0 & (\beta + 1)^{-1}(\theta_1 - 1)(\alpha - \beta) - 1 \end{vmatrix} \\
&= [(\beta + 1)((\beta + 1)^{-1}(\theta_1 - 1)(\alpha - \beta) - 1)] \\
&= [(\theta_1 - 1)(\alpha - \beta) - (\beta + 1)] \\
&= [\theta_1(\alpha - \beta) - (\alpha - \beta) - (\beta + 1)] \\
&= [\theta_1\alpha - \theta_1\beta - \alpha - 1] \\
&= [\sigma_1(\alpha)\theta_1 - \sigma_1(\beta)\theta_1 - \alpha - 1] \\
&= [(\alpha + 1)\theta_1 - \beta\theta_1 - \alpha - 1] \\
&= [(\alpha - \beta + 1)\theta_1 - (\alpha + 1)]. \tag{31}
\end{aligned}$$

Finally, we can substitute back for the chosen quantities in (31) to obtain

$$(n - k + 1)S_n - (n + 1), \tag{32}$$

and this is another desired operator relation, relying only on the operator  $S_n$ , annihilating  $C(n, k)$ , in which its validity can easily be confirmed by applying it to the binomial coefficient  $\binom{n}{k}$ .

Therefore the Theorem 3.4.4 enables us to identify a new relation, which is the resultant, annihilating the same function that is annihilated by the original input polynomials. Our primary focus, thus far, has been on the bivariate case of finding resultants of skew polynomials. However, moving forward to the trivariate case, and ultimately generalising to the multivariate case with  $n$  indeterminants, things become more complicated. This is because the coefficient matrix will contain polynomials with two indeterminants (or more for multivariate case) which means the matrix entries are no longer over a field, and thus we can't use the direct technique we used for the previous case. To overcome this difficulty, we introduce an alternative method by utilising a suitable noncommutative evaluation and interpolation technique. This proposed method not only enables elimination for the multivariate case but also improves the processing speed of the algorithms, as detailed in the following section.

#### 4. Efficient computing through evaluation and interpolation

In this section, we describe another method to compute the resultant of matrices with skew polynomial entries by using evaluation and interpolation techniques. First, we need to identify a suitable evaluation map from the available valid noncommutative



versions of evaluation maps that best suits our purpose. Second, we state a theorem that shows how bivariate resultants and evaluation maps are connected, then we will demonstrate the crucial role this theorem plays in the elimination process. Consequently, we will describe our evaluation and interpolation method, generalising the commutative case presented in [7]. However, both of the evaluation and interpolation processes present several challenges, arising from the significant differences in evaluation maps between skew and ordinary polynomials, and from the need to maintain the noncommutative product rule during the computations. Recall from the previous methods in the commutative case [7] that the resultant's input polynomials were evaluated at some scalar values for the evaluation stage where the computations proceeded consistently and smoothly. However, the direct scalar evaluation for skew polynomials leads to inconsistency here, for example, consider  $\theta t$  for  $\theta$  be a derivative operator  $D$  with respect to  $t$ , which in this case, it is equivalent to  $t\theta + 1^*$  and an attempt to evaluate  $t\theta + 1$  with  $\theta$  set to a scalar value say 2 results in  $t2 + 1 = 2t + 1$ , while an attempt to evaluate the original  $\theta t$  with the same value of  $\theta = 2$  yields  $2t$  and the two obtained results are not the same. This inconsistency illustrates the need for a valid evaluation map from several available formulations of noncommutative evaluation maps in the literature (e.g. remainder theorem [25], product formula [25, Lemma 8.6.4], operator evaluation [26], recursive relation formula [27, §2], etc.), the choice of evaluation method is crucial and depends on the specific study at hand where we will discuss this in more details in the following section.

#### 4.1. Skew polynomial evaluation

When it comes to evaluations in noncommutative rings, one of the main differences compared to the commutative case (where one can simply substitute values for the variables) is that the noncommutative evaluation maps generally do not preserve the products, as observed in the previous example.

In this section, we are searching for a suitable evaluation map that not only preserves the product rule, but also be a ring homomorphism.

Here, we are working with skew polynomials in  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2]$  and a key aspect of this algebra is the presence of operators, such as  $\sigma_1$  or  $\sigma_2$ . In fact, all the elements of Ore polynomial rings can be viewed as operators, which naturally suggests considering an evaluation map that deals with evaluating at operators. Thus, an interesting option would be evaluating at  $\sigma_1$  or  $\sigma_2$ , especially if we know that, in our study, these maps are ring homomorphisms, and this property will be a significant desired factor in the evaluation process. Therefore, we adapt the *operator evaluation* [26] with some adjustments in order to make it compatible with the bivariate case.

Applying the operator evaluation techniques can also improve the efficiency of the polynomial multiplications during the computation of the resultant as shown in [8,9], but we proceed slightly differently as we are working in  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2]$ , we consider one of the operators, namely  $\sigma_1$  itself as an element of the base field  $\mathcal{F}$  in a valid manner, that is ensuring its compatibility with the field  $\mathcal{F}$ . For convenience, we introduce the notion  $\tilde{\mathcal{F}}$  to denote the field of rational maps of operators  $\mathcal{F}(\sigma_1, \circ)$  [10, Remark 14], this approach of considering  $\sigma_1$  in  $\tilde{\mathcal{F}}$  offers a more efficient and convenient way to process our elimination technique which is the main focus of this study.

Next we discuss the definition of operator evaluation for bivariate skew polynomials (which will be a ring homomorphism) as in [10].

**Definition 4.1.1.** Let  $\tilde{\mathcal{F}}[\theta_1; \sigma_1][\theta_2; \sigma_2]$  be a bivariate skew polynomial ring. Since we have commuting indeterminates  $\theta_1$  and  $\theta_2$ , we can consider  $\mathcal{S} = E[\theta_1; \sigma_1]$  where  $E = \tilde{\mathcal{F}}[\theta_2; \sigma_2]$ , that is,

\*The derivative operator  $D$  satisfies  $Du = uD + \frac{d}{dt}u$ , for any function  $u$  with respect to  $t$ . Thus, we have  $Dt = tD + 1$ , when  $u = t$ .

polynomials are regarded with respect to  $\theta_1$ . Then for each polynomial  $f = \sum_{i=0}^n \alpha_i \theta_1^i$ ,  $\alpha_i \in E$ , we define the evaluation map  $\text{eval}_{(\theta_1-\sigma_1)}(f)$  as

$$\text{eval}_{(\theta_1-\sigma_1)} : E[\theta_1; \sigma_1] \rightarrow E$$

$$f = \sum_{i=0}^n \alpha_i \theta_1^i \mapsto f(\theta_1, \sigma_1) = \sum_{i=0}^n \alpha_i \sigma_1^i.$$

The following Lemma [10, Lemma 16] shows that the map  $\text{eval}_{(\theta_1-\sigma_1)}$  is a ring homomorphism for bivariate Ore polynomials.

**Lemma 4.1.2.** *The map  $\text{eval}_{(\theta_1-\sigma_1)}$  is a ring homomorphism.*

In the following, we define the evaluation map when the polynomials are regarded as modulo commutators [10].

**Definition 4.1.3.** *Let  $\tilde{\mathcal{F}}(\theta; \sigma)$  be a (skew) field and let  $\tilde{\mathcal{F}}^\times(\theta; \sigma)$  denote multiplicative group of  $\tilde{\mathcal{F}}(\theta; \sigma)$  containing nonzero elements of  $\tilde{\mathcal{F}}(\theta; \sigma)$ . Let  $f = g^{-1}f$  be an element in*

$$\tilde{\mathcal{F}}^\times(\theta; \sigma) / [\tilde{\mathcal{F}}^\times(\theta; \sigma), \tilde{\mathcal{F}}^\times(\theta; \sigma)].$$

The modular evaluation map  $\text{eval}_{(\theta-\sigma)}(f)$  is defined as:

$$\text{eval}_{(\theta-\sigma)} : \tilde{\mathcal{F}}^\times(\theta; \sigma) / [\tilde{\mathcal{F}}^\times(\theta; \sigma), \tilde{\mathcal{F}}^\times(\theta; \sigma)] \rightarrow \tilde{\mathcal{F}}^\times / [\tilde{\mathcal{F}}^\times, \tilde{\mathcal{F}}^\times]$$

$$f = g^{-1}f \text{ mod } \mathcal{C} \mapsto f(\sigma) = (g(\sigma))^{-1}f(\sigma) \text{ mod } \mathcal{C}', g(\sigma) \neq 0.$$

In particular, if  $f = \sum_{i=0}^n a_i \theta^i$  is an Ore polynomial in  $\tilde{\mathcal{F}}^\times(\theta; \sigma) / [\tilde{\mathcal{F}}^\times(\theta; \sigma), \tilde{\mathcal{F}}^\times(\theta; \sigma)]$  then

$$\text{eval}_{(\theta-\sigma)} : f = \sum_{i=0}^n a_i \theta^i \text{ mod } \mathcal{C} \mapsto f(\sigma) = \sum_{i=0}^n a_i \sigma^i \text{ mod } \mathcal{C}'.$$

**Remark 4.1.4.** *Note that in this study, we assume the evaluation map  $\text{eval}$  becomes modular (by default) as in Definition 4.1.3 when the input argument computed modulo commutators.*

We can now describe the behaviour of resultant under specializations from applying  $\text{eval}$  to polynomials with indeterminate coefficients [10].

**Theorem 4.1.5.** *Let  $\mathcal{S} = \tilde{\mathcal{F}}[\theta_1; \sigma_1][\theta_2; \sigma_2]$ . For all polynomials  $f, g \in \mathcal{S}$ , if  $\deg_{\theta_2}(f) = \deg_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(f))$  and  $\deg_{\theta_2}(g) = \deg_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(g))$  then the following formula holds:*

$$\text{eval}_{(\theta_1-\sigma_1)}(\text{res}_{\theta_2}(f, g)) = \text{res}_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(f), \text{eval}_{(\theta_1-\sigma_1)}(g)). \quad (33)$$

By Theorem 4.1.5, we can conclude the two methods  $\text{eval}_{(\theta_1-\sigma_1)}(\text{res}_{\theta_2}(f, g))$  and  $\text{res}_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(f), \text{eval}_{(\theta_1-\sigma_1)}(g))$  are the same (viewed as operators). Thus, for all  $a$  in  $\mathcal{F}$  we have:

$$\text{eval}_{(\theta_1-\sigma_1)}(\text{res}_{\theta_2}(f, g))(a) = \text{res}_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(f), \text{eval}_{(\theta_1-\sigma_1)}(g))(a). \quad (34)$$

The left side of equation (34) describes the operator evaluation of direct resultant of two bivariate skew polynomials  $f$  and  $g$  at a value  $a$  in  $\mathcal{F}$ , while the right side provides a way on how to obtain the resultant through operator evaluation of its entries which follows by applying evaluation at  $a$ . This ultimately means reducing the computation of resultant to the base ring which then can efficiently and more easily be computed.

**Remark 4.1.6.** *An advantage of using Dieudonné determinant in Theorem 4.1.5 is that the case can be reduced to a triangular determinant with diagonal entries of polynomials  $d'_i(\theta_1)$  ( $i =$*

$1, \dots, k; k = n + m$ ) for the direct method of the left side of equation (34), while the right side will be in the form  $d_i(\sigma_1)$  ( $i = 1, \dots, k; k = n + m$ ) which can be computed by the following product:

$$\begin{aligned} \text{res}_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(f), \text{eval}_{(\theta_1-\sigma_1)}(g))(a) &= \left(\prod_{i=1}^k d_i(\sigma_1)\right)(a) \\ &= (d_1(\sigma_1)d_2(\sigma_1) \cdots d_k(\sigma_1))(a) \\ &= d_1^*(d_2^*(\cdots d_k^*(a))), \end{aligned} \quad (35)$$

where  $d_i^* = d_i(\sigma_1)$  for all  $i = 1, \dots, k$ .

Our method of evaluation and interpolation offers the benefit of enabling elimination for multivariate skew polynomials, which would have been difficult to process otherwise. Additionally, our prior observations show that evaluation and interpolation techniques offer a significant asymptotic advantage over the direct method in the commutative case [7]. We have also observed substantial speed improvements in computing skew polynomial products using the evaluation and interpolation method, as applied over finite fields [8,9]. Consequently, we can expect an asymptotic improvement in the computational speed of computing the product in (35) using evaluation and interpolation techniques. Thus, the resultant can be computed more efficiently, and it obtains the same result as directly computing the resultant. In the bivariate case, for instance, it achieves the same result as directly computing the resultant of two bivariate skew polynomials,  $f(\theta_1, \theta_2)$  and  $g(\theta_1, \theta_2)$ , with respect to  $\theta_2$ .

While achieving more efficient computation is highly desirable, it often comes with challenges that requires investigation and resolutions. The following sections will detail the main steps used in our study and the challenges encountered during its development.

#### 4.2. Efficiency steps and challenges encountered

The efficiency of this method is achieved by breaking down the computation into three main stages, as proven by Theorem 4.1.5;

- (i) *Evaluation*: Choose distinct values  $a_i$  ( $i = 1, \dots, k$ ), then compute the evaluation of  $f$  and  $g$  at  $a_i$  with respect to  $\theta_1$ . These become univariate polynomials in  $\theta_2$ .
- (ii) *Partial resultants*: Obtain the *partial resultants* of these evaluated polynomials  $f$  and  $g$  (step i) with respect to  $\theta_2$ .
- (iii) *Interpolation*: Combine these partial resultants using a suitable interpolation technique to recover the complete resultant of the original  $f$  and  $g$ .

However, applying these steps to skew polynomials presents several challenges compared to the commutative case. These challenges include:

- (i) *Evaluation values*: Since there are several evaluation maps to choose from, evaluating a polynomial using a specific evaluation map (in this case operator evaluation at  $\sigma_1$  then applying it to a value  $a$ ) may not be the actual evaluation value that is expected by the chosen interpolation method (such as a Lagrange or Newton interpolation technique). A suitable interpolation method should be used to match the chosen evaluation map.
- (ii) *Validity of the evaluation map*: The evaluation map needs to be a ring homomorphism to preserve the product.
- (iii) *Distinct conjugacy classes*: All chosen evaluation values must belong to pairwise distinct conjugacy classes for the evaluation and interpolation techniques to function correctly. We can achieve this by choosing primitive elements in which they inherently belong to different conjugacy classes.
- (iv) *Unlucky evaluations*: To avoid *unlucky evaluations*, where a chosen value eliminates the leading coefficient and alters the original polynomial's degree, we need to identify and exclude such unconstructive values. This can be determined by

examining the leading term status at the time of evaluation, if it is unlucky then skip to the next value, and continue only if the evaluation is valid/lucky.

- (v) *Insufficient evaluation values*: In some cases, we may not have enough valid values for the evaluation stage. To address this, we can extend the domain by including additional valid values. Then, these new values can be utilised by the evaluation map as long as they belong to distinct conjugacy classes (as in (iii)).

The following sections address these challenges in more details and illustrate potential solutions with examples.

#### 4.3. Skew polynomial interpolation

This part describes the interpolation stage on a normal basis. The first subsection is definitions and notations then we provide some technical details on how to compute the resultant of bivariate skew polynomials through evaluation and interpolation techniques including how to overcome the challenges encountered, followed by an example to illustrate the idea.

##### 4.3.1. Galois theory (finite and infinite field extensions)

For clarity and convenience, we recall some definitions and notations used in the remainder of this study regarding Galois field extensions for both finite and infinite field extensions.

Recall, a field  $\mathcal{F}$  is a *field extension* of a field  $\mathcal{K}$  if  $\mathcal{K} \subset \mathcal{F}$ , denoted by  $\mathcal{F}/\mathcal{K}$ . In this study,  $\mathcal{F}$  is always a field extension of  $\mathcal{K}$ , unless otherwise specified. In Section 4.3.3 we study a particular case when  $\mathcal{K} = \mathbb{Q}$  the field of rational numbers and  $\mathcal{F} = \mathbb{C}$  the field of complex numbers, or  $\mathcal{F}$  maybe a subfield of  $\mathbb{C}$ .

Let  $X \subset \mathcal{F}$ , we type  $\mathcal{K}(X)$  for the smallest subfield generated by  $X$  in  $\mathcal{F}$  (that is the smallest subfield of  $\mathcal{F}$  that contains both  $\mathcal{K}$  and  $X$ ). A field extension  $\mathcal{F}/\mathcal{K}$  is called *finitely generated* if there is a finite set  $X \subset \mathcal{F}$  such that  $\mathcal{F} = \mathcal{K}(X)$ ; furthermore it is called *simple* if there exists a single element  $\alpha \in \mathcal{F}$  such that  $\mathcal{F} = \mathcal{K}(\alpha)$ , it is common to write  $\mathcal{K}(\alpha)$  instead of  $\mathcal{K}(\{\alpha\})$ .

An element  $\alpha \in \mathcal{F}$  is called an *algebraic* over  $\mathcal{K}$  if there exists a non-zero polynomial  $m_\alpha$  over  $\mathcal{K}$  such that  $m_\alpha(\alpha) = 0$ . The *minimal polynomial* of an algebraic element  $\alpha \in \mathcal{F}$  over  $\mathcal{K}$  is a monic irreducible polynomial  $m_\alpha$  over  $\mathcal{K}$  such that  $m_\alpha(\alpha) = 0$ ; furthermore, if  $\alpha$  is a root of any other polynomial  $m'_\alpha$  over  $\mathcal{K}$  then  $m_\alpha$  divides  $m'_\alpha$  (that is  $m_\alpha$  is of minimal degree).

Viewing  $\mathcal{F}$  as a vector space over  $\mathcal{K}$ , the dimension of  $\mathcal{F}$  over  $\mathcal{K}$  is the *degree* of the extension  $\mathcal{F}/\mathcal{K}$  and is denoted by  $[F : K]$ . We say  $\mathcal{F}$  is *finite extension* or *finite dimensional* if  $[F : K] < \infty$ , and in this case we denote the dimension by  $\dim_{\mathcal{K}}(\mathcal{F}) = [\mathcal{F} : \mathcal{K}]$ .

Consider the finite extension  $\mathcal{K}(\alpha)/\mathcal{K}$  for an algebraic element  $\alpha$  over  $\mathcal{K}$  defined as:

$$\mathcal{F} = \mathcal{K}(\alpha) = \left\{ \sum_{i=0}^{r-1} a_i \alpha^i : a_i \in \mathcal{K} \right\}, \quad (36)$$

where  $r = \deg(m_\alpha)$  for some minimal polynomial  $m_\alpha$  over  $\mathcal{K}$ . That is the set of all the finite linear combinations of basis elements  $\{1, \alpha, \alpha^2, \dots, \alpha^{r-1}\}$  over  $\mathcal{K}$ . Let  $\sigma$  be an automorphism of  $\mathcal{F}$  that fixes  $\mathcal{K}$  (called  $\mathcal{K}$ -automorphism) which satisfies  $\sigma(\sum_{i=0}^{r-1} a_i \alpha^i) = \sum_{i=0}^{r-1} a_i \sigma(\alpha)^i$ . Note that  $\sigma$  is determined by the image  $\sigma(\alpha)$ , where  $\sigma(\alpha)$  represents a root of  $m_\alpha$  which yields a one-to-one correspondence between the  $\mathcal{K}$ -automorphisms of  $\mathcal{K}(\alpha)$  and the roots of  $m_\alpha$  (since these automorphisms send a root to another root of the same minimal polynomial  $m_\alpha$ ). Moreover, this also yields an isomorphism between  $\mathcal{K}(\alpha)$  and  $\mathcal{K}(\sigma(\alpha))$ .

We call a monic irreducible polynomial over  $\mathcal{K}$  *separable* if it has no double roots in any field extension of  $\mathcal{K}$  (i.e. all its roots are distinct). An extension  $\mathcal{F}/\mathcal{K}$  is separable if for every element  $\alpha \in \mathcal{F}$ , its minimal polynomial  $m_\alpha$  over  $\mathcal{K}$  is separable. A well known theorem (primitive element theorem) states that any separable finite field extension is simple.

An extension  $\mathcal{F}/\mathcal{K}$  is called *normal* if every irreducible polynomial over  $\mathcal{K}$  that has at least one root in  $\mathcal{F}$  has all its roots in  $\mathcal{F}$  (i.e. the polynomial splits completely in  $\mathcal{F}$ ).

A finite extension  $\mathcal{F}/\mathcal{K}$  is called *Galois extension* if it is both separable and normal. Additionally, the *Galois group* of a Galois extension is defined as the group (under composition) of all the automorphisms  $\sigma$  of  $\mathcal{F}$  that fix  $\mathcal{K}$ , and this group is denoted by  $\text{Gal}(\mathcal{F}/\mathcal{K})$ .

The *Fundamental Theorem of Galois Theory* establishes a fundamental relation between the structure of a Galois field extension and its Galois group. It states that there is a one-to-one correspondence between the intermediate subfields of  $\mathcal{F}$  containing  $\mathcal{K}$  and the subgroups of  $\text{Gal}(\mathcal{F}/\mathcal{K})$ .

Extending Galois theory to the case of *infinite field extensions*, where the Galois group  $\text{Gal}(\mathcal{F}/\mathcal{K})$  can be infinite, the fundamental theorem requires a more careful examination. While the concepts of separability and normality remain relevant for infinite extensions, the one-to-one correspondence between subgroups and subfields, as established in the finite case, no longer holds [28]. To address this issue, we can define a topology, known as the *Krull topology* [28], on the Galois group  $\text{Gal}(\mathcal{F}/\mathcal{K})$ . While the precise definition of this topology and the corresponding closed subgroups is not essential for our current study, it is important to recognize that this topological structure enables us to restrict our consideration to the *closed subgroups* of  $\text{Gal}(\mathcal{F}/\mathcal{K})$ . This restriction approach effectively resolves the issue at hand, where the fundamental theorem can be restated as there is a one-to-one correspondence between intermediate subfields and closed subgroups of the Galois group  $\text{Gal}(\mathcal{F}/\mathcal{K})$ . Readers interested in more details on Krull topology can refer, for example, to [28].

A *number field* is a field extension of  $\mathbb{Q}$  which is finite dimensional when viewed as a vector space over  $\mathbb{Q}$ . An algebraic number that has its minimal polynomial with integer coefficients is called *algebraic integer*. An interesting fact is that any number field is generated by a single algebraic integer (for example see Theorem 2.2 and Corollary 2.12 in [29]).

Next, we briefly describe *normal basis* [30] which is a particular basis for finite Galois extensions when viewed as a vector space over the base field. It has also been described for the infinite case [31], as we will discuss in the following subsection.

#### 4.3.2. Normal basis (finite and infinite cases)

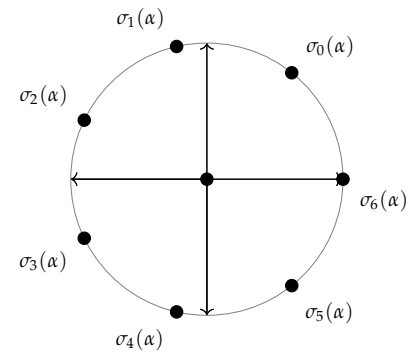
Let  $\mathcal{F}/\mathcal{K}$  be a finite Galois extension. The *Galois  $\mathcal{K}$ -conjugates* of an element  $\alpha \in \mathcal{F}$  is the set of all elements  $\sigma(\alpha) \in \mathcal{F}$  where  $\sigma \in \text{Gal}(\mathcal{F}/\mathcal{K})$ , this set represents the action of the Galois group on the element  $\alpha$ , which is the reason it is sometimes denoted by  $\text{Gal}(\mathcal{F}/\mathcal{K}) \cdot \alpha$  or simply by  $G \cdot \alpha$  where  $G = \text{Gal}(\mathcal{F}/\mathcal{K})$ .

An element  $\alpha \in \mathcal{F}$  is called *normal* if the Galois conjugates set  $G \cdot \alpha = \{\sigma(\alpha) \mid \sigma \in \text{Gal}(\mathcal{F}/\mathcal{K})\}$  forms a basis for  $\mathcal{F}$  when viewed as a vector space over  $\mathcal{K}$ , which is characterised by the group action that forms a single orbit, that is its elements lie on the same  $\text{Gal}(\mathcal{F}/\mathcal{K})$ -orbit (see Figure 2). Such a basis is called normal [30] as defined below.

**Definition 4.3.1.** Let  $\mathcal{F}/\mathcal{K}$  be a finite Galois extension of degree  $n$  with the Galois group  $\text{Gal}(\mathcal{F}/\mathcal{K}) = \{\sigma_0, \sigma_1, \dots, \sigma_{n-1}\}$ . A basis that consists of all the  $\mathcal{K}$ -conjugate elements

$$G \cdot \alpha = \{\sigma_0(\alpha), \sigma_1(\alpha), \dots, \sigma_{n-1}(\alpha)\}$$

is called *normal basis* and denoted by  $\mathcal{N}(\alpha)$ .



**Figure 2.** Single  $\text{Gal}(\mathcal{F}/\mathcal{K})$ -orbit of an element  $\alpha \in \mathcal{F}$  when  $[\mathcal{F} : \mathcal{K}] = 7$



One of the simplest examples of constructing normal basis for finite Galois extensions is when we have the field extension  $\mathbb{Q}(i)$  over  $\mathbb{Q}$  with its Galois group  $\text{Gal}(\mathbb{Q}(i)/\mathbb{Q}) = \{\sigma_0, \sigma_1\}$  defined as

$$\sigma_0(i) = i \text{ and } \sigma_1(i) = -i.$$

A normal basis can be defined by the Galois group action on the element  $\alpha = 1 + i \in \mathbb{Q}(i)$  as

$$\sigma_0(\alpha) = 1 + i \text{ and } \sigma_1(\alpha) = 1 - i.$$

Thus, the normal basis in this case is  $\mathcal{N}(\alpha) = \{1 + i, 1 - i\}$ .

Note that, although the field extension is in the form  $\mathbb{Q}(i)/\mathbb{Q}$ , the element  $\alpha = i$  does not constitute a normal basis since the set of its conjugates  $\{\sigma_0(i), \sigma_1(i)\} = \{i, -i\}$  is not linearly independent.

In the case of an infinite Galois extension  $\mathcal{F}/\mathcal{K}$ , the original definition of a normal basis (as in Definition 4.3.1) does not make sense anymore. This is because the set of  $\mathcal{K}$ -conjugates of an element in  $\mathcal{F}$  remains finite in which it can not be a  $\mathcal{K}$ -basis when  $\mathcal{F}/\mathcal{K}$  is infinite. To address this limitation, Lenstra [31] described a reformulation of the normal basis definition that allows the concept to be applicable in the infinite case as well, the idea is based on the correspondence between  $\mathcal{F}$  and  $C(G, \mathcal{K})$  the set of all continuous maps from the Galois group  $G$  to the field  $\mathcal{K}$ , assuming that  $G$  is equipped with the Krull topology and  $\mathcal{K}$  has the discrete topology. This reformulated definition of the normal basis reduces to the original one when the Galois extension  $\mathcal{F}/\mathcal{K}$  is finite.

Other, more sophisticated examples of normal basis can be constructed by adjoining roots of unity to a number field as we will see in the following section.

### 4.3.3. Cyclotomic extension

Cyclotomic fields are essential for various applications involving roots of unity, such as representation theory, Kummer theory, and the discrete Fourier transform. In the area of cryptography, certain elliptic curves defined over cyclotomic fields are utilised in modern cryptography.

In this section, we provide a quick overview of the cyclotomic field extension which will be needed in the next subsection.

It is easy to check that the polynomial  $\theta^n - 1$  is a separable polynomial over  $\mathcal{K}$  (since  $\theta^n - 1$  is relatively prime to its derivative which is the non-zero polynomial  $n\theta^{n-1}$ ), where  $\theta^n - 1$  has  $n$  distinct roots in its splitting field over  $\mathcal{K}$ . In  $\mathbb{C}$ , the set of these roots is in the form

$$\mu_n = \{e^{2\pi i k/n} \mid k = 0, \dots, n-1\} = \langle \alpha \rangle, \text{ where } \alpha = e^{2\pi i/n},$$

which is a multiplicative cyclic group of order  $n$  generated by  $\alpha$ . In general, this  $\alpha$  is not the only generator for  $\mu_n$ . Any element of the cyclic group  $\mu_n$  which generates  $\mu_n$  is called *primitive  $n$ -th roots of unity*. The term cyclotomic refers to circle-cutting where  $n$ -th roots of unity divide a circle into  $n$  equal parts on the complex plane (see Figure 3 when  $n = 7$ ). For any integer  $k$ , primitive  $n$ -th roots of unity can be identified through  $\text{gcd}(k, n)$  since we know that the order of  $\alpha^k$  in  $\mu_n$  is  $n/\text{gcd}(n, k)$  which means any  $\alpha^k$  is a primitive  $n$ -th root of unity iff  $\text{gcd}(k, n) = 1$ .

As  $\mu_n$  is a cyclic group, the mapping  $\alpha \mapsto \alpha^k$  sends a generator to another generator iff  $\text{gcd}(k, n) = 1$ , and as a consequence; this mapping is automorphism iff  $\text{gcd}(k, n) = 1$ .

Note that the primitive elements in  $\mu_n$  are powers of each others, therefore the extension  $\mathcal{K}(\alpha)$  is irrelevant to the choice of  $\alpha$  in  $\mu_n$ . In the case when  $\mathcal{K} = \mathbb{Q}$  is the field of rational numbers then the field extension  $\mathcal{K}(\alpha)$  is called

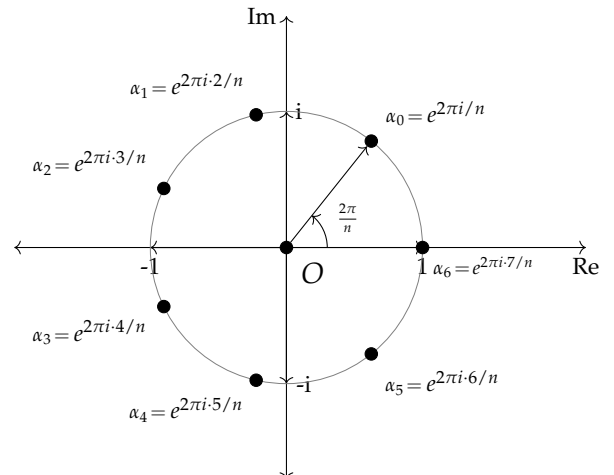


Figure 3. Roots of unity when  $n = 7$

cyclotomic field of  $n$ -th roots of unity, for a positive integer  $n$ . In particular if  $n = p$  is a prime then a basis can be constructed which acts as a normal basis (compare Figure 2 and Figure 3) by simply taking the basis that starts with  $\alpha_0 = e^{2\pi i/n}$ .

An advantage of utilising cyclotomic extensions is that it works for any arbitrary base field by simply adjoining the  $n$ -th roots of unity for a fixed integer  $n \neq 0$  in  $\mathcal{K}$  (i.e. if  $\mathcal{K}$  is not of characteristic 0 then  $n$  should not be divisible by the characteristic of the field  $\mathcal{K}$ ). For the modular algorithms we can choose a large prime number  $p$  for the value  $n$  in the examples. This is to make sure we can generate enough values to be available for the interpolation stage which is the topic of the next subsection.

#### 4.3.4. Interpolation

In this subsection, we apply our theorem to obtain an evaluation and interpolation method for computing the resultant of bivariate skew polynomials over number fields, in particular over  $\mathbb{Q}(\alpha)$  where  $\alpha$  is a complex root of unity, followed by examples to illustrate the idea.

Recall that in the theorem, we use operator evaluation to evaluate the polynomials, which then applied to selected evaluation values. However, the properties of this evaluation are different from the other evaluation maps for noncommutative polynomials. For instance when using operator evaluation  $\text{eval}_{(\theta, \sigma)(a)}$  for evaluating  $f = \theta^2$  in a skew polynomial ring  $\mathcal{F}[\theta; \sigma]$  for a value  $a \in \mathcal{F}$  we first obtain  $\sigma^2$  and then applying it to the value  $a$  to find  $\sigma^2(a)$ , this image is not an actual evaluation at the value  $a$  in the typical noncommutative evaluation manner which should be  $\sigma(a)a$ ; neither is it an evaluation of the form  $(\sigma(a))^2$  because  $(\sigma(a))^2 = \sigma(a^2)$  which is different than  $\sigma^2(a)$ , and it is certainly not a naive substitution in the form  $a^2$ . So the main question from an interpolation point of view would be at which specific value is the indeterminate of the original polynomial evaluated? For this, we need a suitable interpolation method in order to be able to properly recover the original polynomial, which is the topic of the following subsection.

#### 4.4. Evaluation and interpolation technique

In this section we describe a modular method that can compute the resultant through an evaluation and interpolation technique by applying our resultant formula in the Theorem 4.1.5.

The method uses coefficient compression, that is by equating the coefficients (of both sides of the formula) at enough evaluation values, and then solving its corresponding linear system. A similar method is also used in [9] and [8] for multiplication of univariate skew polynomials.

Let  $\mathcal{F}/\mathcal{K}$  be a finite Galois extension of degree  $r$ . We consider the computation of the resultant of two bivariate skew polynomials  $f$  and  $g$  in  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2]$ . The particular case we have in mind is that when  $\mathcal{K} = \mathbb{Q}$  and  $\mathcal{F} = \mathbb{Q}(\alpha)$  for some fixed choice of a complex root of unity  $\alpha$ .

Let  $\mathcal{N}$  be the normal basis of  $\mathcal{F}/\mathcal{K}$  given as

$$\mathcal{N} = \{\alpha_0, \alpha_1, \dots, \alpha_{r-1}\}$$

such that  $\alpha_i = \sigma_1^i(\alpha_0)$  for  $i = 0, \dots, r - 1$ .

As with any modular setup, the method requires a bound  $s$  for the number of needed evaluation values for the interpolation stage. For this we can use the total sum of the degrees of the factors plus one, which is the degree of the product plus one as in the commutative case (since in Ore algebra,  $\deg(fg) = \deg(f) + \deg(g)$  for any two Ore polynomials  $f$  and  $g$ ), we may encounter a case where we do not have enough distinct evaluation values (i.e. enough elements in the base field belonging to different conjugacy classes). In this case, we can extend the base field to include additional elements as needed.



However, for our purposes, we can typically select a sufficiently large value that ensures there are enough elements for the evaluation map. 743  
744

Recall that our resultant formula for two bivariate skew polynomials  $f$  and  $g$  in  $\mathcal{F}[\theta_1; \sigma_1][\theta_2; \sigma_2]$  of degrees  $n$  and  $m$  respectively, is in the form 745  
746

$$\text{eval}_{(\theta_1-\sigma_1)}(\text{res}_{\theta_2}(f, g)) = \text{res}_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(f), \text{eval}_{(\theta_1-\sigma_1)}(g)). \quad (37)$$

Let us assume that the original resultant  $R$  is in the generic form with coefficients  $c_i$  as 747

$$R = \sum_{i=0}^{s-1} c_i \theta_1^i, \quad (38)$$

then our aim is to find these unknown coefficients  $c_i$  of  $R$ . 748

We fix the normal element  $\alpha_0$  and evaluate the right side of (37) at the values  $\alpha_0^j$  ( $j = 0, 1, \dots, s-1$ ) as in 749  
750

$$\begin{aligned} \text{res}_{\theta_2}(\text{eval}_{(\theta_1-\sigma_1)}(f), \text{eval}_{(\theta_1-\sigma_1)}(g))(\alpha_0^j) &= \left( \prod_{i=1}^k d_i(\sigma_1) \right) (\alpha_0^j) \\ &= (d_1(\sigma_1) d_2(\sigma_1) \cdots d_k(\sigma_1)) (\alpha_0^j) \\ &= d_1^*(d_2^*(\cdots d_k^*(\alpha_0^j))). \end{aligned} \quad (39)$$

Note that computing (39) provides a single value for each evaluation point  $\alpha_0^j$ ; let us denote this value by  $R^*(\alpha_0^j)$ . 751  
752

On the other hand, if we evaluate the generic resultant  $R$  at the same values  $\alpha_0^j$  ( $j = 0, 1, \dots, s-1$ ) in the operator evaluation manner as 753  
754

$$\begin{aligned} \text{eval}_{(\theta_1-\sigma_1)(\alpha_0^j)}(R) &= R^*(\alpha_0^j) = \sum_{i=0}^{s-1} c_i \sigma_1^i(\alpha_0^j) \\ &= \sum_{i=0}^{s-1} c_i \alpha_0^j, \quad \text{since } \alpha_i = \sigma_1^i(\alpha_0). \end{aligned} \quad (40)$$

Equalities (39) and (40) allow us to solve the following system for unknowns  $c_i$  as: 755

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_0 & \alpha_1 & \cdots & \alpha_{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_0^{s-1} & \alpha_1^{s-1} & \cdots & \alpha_{s-1}^{s-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{s-1} \end{pmatrix} = \begin{pmatrix} R^*(\alpha_0^0) \\ R^*(\alpha_0^1) \\ \vdots \\ R^*(\alpha_0^{s-1}) \end{pmatrix} \quad (41)$$

Thus, we have found the formula (38) that we were looking for. 756

#### 4.5. Examples 757

In the following example we compute the resultant of two bivariate skew polynomials over a number field  $\mathbb{Q}(\alpha)$  using both methods; the direct as well as the evaluation and interpolation methods as described in the previous sections. 758  
759  
760

**Example 4.5.1.** Let  $\mathbb{Q}(\alpha)[\theta_1; \sigma_1][\theta_2; \sigma_2]$  be a bivariate skew polynomial ring where  $\alpha = e^{2\pi i/p}$  and  $p = 101$ , endowed with two  $\mathbb{Q}$ -automorphisms  $\sigma_1, \sigma_2$  of  $\mathbb{Q}(\alpha)$  such that: 761  
762

$$\sigma_1(\alpha) = \alpha^2, \quad \sigma_2(\alpha) = \alpha^3.$$

Consider the problem of computing the resultant of the following two bivariate skew polynomials over  $\mathbb{Q}(\alpha)$  w.r.t.  $\theta_2$ : 763  
764

$$\begin{cases} f = \alpha\theta_1\theta_2^2 + \alpha\theta_1 - 1 \\ g = \alpha^2\theta_1^2\theta_2 - \alpha \end{cases} \quad (42)$$

Let the matrix  $M = \text{Sylv}_{\theta_2}(f, g)$  be the Sylvester matrix of  $f$  and  $g$  in its general form as following

$$\det(M) = \det(\text{Sylv}_{\theta_2}(f, g)) = \begin{vmatrix} f & a_2^{[0]} & a_1^{[0]} & a_0^{[0]} \\ \theta_2 g & b_1^{[1]} & b_0^{[1]} & \\ g & b_1^{[0]} & b_0^{[0]} & \end{vmatrix}$$

$$= \begin{vmatrix} a_2 & a_1 & a_0 \\ \sigma_2(b_1) & \sigma_2(b_0) & \\ & b_1 & b_0 \end{vmatrix},$$

applying it to our example it becomes

$$= \begin{vmatrix} \alpha\theta_1 & 0 & \alpha\theta_1 - 1 \\ \alpha^6\theta_1^2 & -\alpha^3 & \\ & \alpha^2\theta_1^2 & -\alpha \end{vmatrix}.$$

Now, we can use row operations to transform the above matrix to an upper triangular form (following Lemma 3.1.1 and Remark 3.1.2);

$$\det(M) = \begin{vmatrix} \alpha\theta_1 & 0 & \alpha\theta_1 - 1 \\ 0 & -\alpha^3 & -\alpha^6\theta_1^2 + \alpha^4\theta_1 \\ 0 & 0 & -\alpha^{14}\theta_1^4 + \alpha^6\theta_1^3 - \alpha \end{vmatrix}.$$

Thus, the direct resultant can be computed by multiplying the diagonal elements  $d_i$

$$\det(M) = \prod_{i=1}^3 d_i = \alpha\theta_1 (-\alpha^3)(-\alpha^{14}\theta_1^4 + \alpha^6\theta_1^3 - \alpha) \quad (43)$$

$$\begin{aligned} &= \alpha\theta_1 (\alpha^{17}\theta_1^4 - \alpha^9\theta_1^3 + \alpha^4) \\ &= \alpha^{35}\theta_1^5 - \alpha^{19}\theta_1^4 + \alpha^9\theta_1. \end{aligned} \quad (44)$$

Next, we use another method to compute (43) through an evaluation and interpolation method (by applying Theorem 4.1.5) as following:

1. From the right side of the resultant formula (37) we compute the composition product (39)

$$d_1^*(d_2^*(d_3^*(w))) \quad (45)$$

for some  $p$ -th roots of unity  $w$  (in this example,  $w$  can be  $\alpha$  to any integer power greater than 0 and less than the prime  $p$ ). Let us first compute  $d_3^*(w)$  as

$$\begin{aligned} d_3^*(w) &= (-\alpha^{14}\sigma_1^4 + \alpha^6\sigma_1^3 - \alpha)(w) \\ &= -\alpha^{14}\sigma_1^4(w) + \alpha^6\sigma_1^3(w) - \alpha w \\ &= -\alpha^{14}w^{16} + \alpha^6w^8 - \alpha w, \end{aligned}$$

then applying it to (45);

$$\begin{aligned} d_1^*(d_2^*(d_3^*(w))) &= \alpha \sigma_1(-\alpha^3(-\alpha^{14}w^{16} + \alpha^6w^8 - \alpha w)) \\ &= \alpha \sigma_1(\alpha^{17}w^{16} - \alpha^9w^8 + \alpha^4w) \\ &= \alpha \sigma_1(\alpha^{17}w^{16}) - \alpha \sigma_1(\alpha^9w^8) + \alpha \sigma_1(\alpha^4w) \\ &= \alpha^{35}w^{32} - \alpha^{19}w^{16} + \alpha^9w^2. \end{aligned}$$

We call this result the *evaluated resultant polynomial* (denoted by  $R(w)$ ) which in this case is

$$R(w) = \alpha^{35}w^{32} - \alpha^{19}w^{16} + \alpha^9w^2. \quad (46)$$

- We select evaluation values from the following normal basis that starts with the normal element  $\alpha$ :

$$\mathcal{N} = \{\sigma_1^i(\alpha) \mid i = 0, \dots, p-1\}.$$

- To perform our actual evaluations, we need a bound on the number of evaluations required to recover the original resultant which is the sum of the degrees of  $d_i$  plus one (that is 6). Therefore, we compute (46) at the first 6 values in the normal basis  $\mathcal{N}$  as

$$w_i = \sigma_1^i(\alpha), i = 0, \dots, 5$$

which are the values  $w_0 = \alpha$ ,  $w_1 = \alpha^2$ ,  $w_2 = \alpha^4$ ,  $w_3 = \alpha^8$ ,  $w_4 = \alpha^{16}$  and  $w_5 = \alpha^{32}$  (since  $\sigma_1(\alpha) = \alpha^2$ ). Note that these values satisfy  $w_{i+1} = \sigma_1(w_i)$  for  $i = 0, \dots, p-1$ .

- Let  $V$  be a vector whose entries are given by the actual evaluation of the evaluated resultant polynomial (46) at the corresponding values  $w_i$  as mentioned in the previous step. For example, the first evaluation

$$R(w_0) = R(\alpha) = \alpha^{67} - \alpha^{35} + \alpha^{11},$$

stored in the vector's first entry, and so on for the other evaluations  $R(w_i)$  for  $i = 0, \dots, 5$ .

- For the left side of the resultant formula (37), let us assume that the original resultant  $R$  is in the generic form as in (38)

$$R = c_5\theta^5 + c_4\theta^4 + c_3\theta^3 + c_2\theta^2 + c_1\theta + c_0 \quad (47)$$

and our aim is to find those coefficients  $c_i$ ,  $i = 0, \dots, 5$ . By applying our theorem, each evaluation  $R^*(w_i)$  is the same as the corresponding values that we have obtained in the previous step (i.e. the entries in  $V$ ). For example, the evaluation  $R^*(w_i)$  at the first value  $w_0 = \alpha$  is

$$R^*(\alpha) = c_5\sigma_1^5(\alpha) + c_4\sigma_1^4(\alpha) + c_3\sigma_1^3(\alpha) + c_2\sigma_1^2(\alpha) + c_1\sigma_1(\alpha) + c_0\alpha$$

which is equal to the first entry in the vector  $V$ , and so on for the other evaluations. This will allow us to solve a linear system (e.g. by using a software such as Maple) in the form

$$Mx = V, \quad (48)$$

where  $x$  is a vector of the unknowns  $c_i$  and  $M$  is a matrix of the form:

$$\begin{pmatrix} \alpha & \alpha^2 & \alpha^4 & \alpha^8 & \alpha^{16} & \alpha^{32} \\ \alpha^2 & \alpha^4 & \alpha^8 & \alpha^{16} & \alpha^{32} & \alpha^{64} \\ \alpha^4 & \alpha^8 & \alpha^{16} & \alpha^{32} & \alpha^{64} & \alpha^{27} \\ \alpha^8 & \alpha^{16} & \alpha^{32} & \alpha^{64} & \alpha^{27} & \alpha^{54} \\ \alpha^{16} & \alpha^{32} & \alpha^{64} & \alpha^{27} & \alpha^{54} & \alpha^7 \\ \alpha^{32} & \alpha^{64} & \alpha^{27} & \alpha^{54} & \alpha^7 & \alpha^{14} \end{pmatrix}.$$

From solving the linear system (48) we obtain all the coefficients  $c_i$  of the original resultant (47) as following:

$$\begin{pmatrix} 0 \\ \alpha^9 \\ 0 \\ 0 \\ -\alpha^{19} \\ \alpha^{35} \end{pmatrix},$$

which is the same as the polynomial obtained by the direct method (44).

Note that as variables are eliminated, we have reduced the computations to computing only with the elements (numbers) in the base field  $\mathbb{Q}(\alpha)$ , which is the key idea behind the method of evaluation and interpolation to improve the efficiency of the method.

Now, let us describe an example for the infinite dimensional case. For this, we can consider

$$\mathcal{F} = \bigcup_{n \geq 1} \mathbb{Q}(\alpha_n),$$

where  $\alpha_n$  is a primitive  $p^n$  for a fixed (odd) prime  $p$ , this is the union of all  $p$ -th power cyclotomic extensions of  $\mathbb{Q}$  where each of  $\mathbb{Q}(\alpha_n)$  is of finite extension. Meaning its Galois extension  $G = \text{Gal}(\mathcal{F}/\mathcal{K})$ , where  $\mathcal{K} = \mathbb{Q}$ , is a composite of finite Galois extensions. An element in  $G$  can be determined by indicating how it acts on each  $\mathbb{Q}(\alpha_n)$ . In finite Galois theory, we know an automorphism  $\sigma_n$  in  $G$  is acting by

$$\sigma_n(\alpha_n) = \alpha_n^{a_n}, \quad (49)$$

for some integer  $a_n \bmod p^n$  where  $(a_n, p) = 1$  [28, Example 3.7], with the property  $\alpha_{n+1}^p = \alpha_n$  and

$$\sigma_n|_{\mathbb{Q}(\alpha_{n-1})} = \sigma_{n-1},$$

which enables an extension of  $\sigma_n$  to an automorphism, say  $\sigma^*$ , of  $\mathcal{F}$  in  $G$ . Let us specify, for instance  $a_n = 2$  such that

$$\sigma^*(\alpha_n) = \alpha_n^2.$$

Let  $m_n$  be the order of  $\sigma_n$ , that is

$$\sigma_n^{m_n}(\alpha_n) = 1,$$

then, combined with the definition of  $\sigma_n$  (formula (49)) and with  $a_n = 2$ , we can conclude

$$\sigma_n^{m_n}(\alpha_n) = \alpha_n^d, \text{ where } d = 2^{m_n},$$

and  $d \equiv 1 \pmod{p^n}$ , since  $m_n$  is the order of  $\sigma_n$ .

Consequently, as  $n$  approaches to infinity the order  $m_n$  is also approaches to infinity, meaning our automorphism  $\sigma^*$  is of infinite order. In the following, we consider our example with two infinite order automorphisms  $\sigma_1^*$  and  $\sigma_2^*$ .

**Example 4.5.2.** Consider the problem of eliminating an indeterminate, namely  $\theta_1$ , from the algebra  $\mathcal{F}[\theta_1; \sigma_1^*][\theta_2; \sigma_2^*]$ .

In the literature, Hachenberger’s work [32], in the field of number theory, provides explicit formulation for constructing normal elements (including those that are *completely normal*, meaning the element is simultaneously normal over every intermediate field extension [32]) within cyclotomic fields of prime power order over the rational field.

Consequently, we can now follow similar steps as in Example 4.5.1 to derive the elimination process for Example 4.5.2. Note that for the evaluation and interpolation stage, it is convenient to fix a (large)  $n$  and work in a sub-algebra over  $\mathcal{F}'$ , where  $\mathcal{F}'$  is a finite extension of  $\mathcal{K}$ , then the rest will follow in the same manner as in the previous example.

## 5. Conclusion and future work

In this study, we have successfully derived the concept of the resultant for bivariate skew polynomials and applied it to eliminate indeterminates in skew polynomial systems. Our methodology covers two primary techniques; the first utilises a Sylvester-style matrix constructed from the coefficients of the polynomials, allowing for direct computation of the resultant. The second technique introduces a modular approach that utilises evaluation and interpolation methods to derive partial resultants, which are then combined to yield the original resultant.

The study’s focus on the bivariate case is essential due to its role in a recursive evaluation and interpolation technique in which this technique enables the reduction of a general  $n \times n$  system to an  $(n - 1) \times (n - 1)$  system by evaluating one indeterminate. This recursive process can be repeated until a bivariate system is reached. Subsequently, the study’s specialized bivariate techniques can be applied to solve the original system.

The contributions of this research extend beyond theoretical exploration. We have demonstrated the practical applicability of the derived resultant in combinatorial contexts, proving (or deriving) combinatorial identities, simplifying skew polynomial systems, and determining the existence of solutions by assessing the vanishing of the resultant. Our work not only contributes to the existing literature but also opens new avenues for future research in both algebraic and computational fields.

In future work, we aim to explore applications related to cryptographic schemes, including the Diffie–Hellman protocol and secret sharing among any number of participants. Prior research has already examined the use of skew polynomials, and our upcoming research intends to build on this by leveraging the resultant introduced in this paper and by properties of the Dieudonné determinant. Moreover, additional optimisation techniques can be employed, particularly by leveraging the properties of roots of unity during the evaluation stage. This approach presents opportunities for parallel computation and incorporates established techniques for handling this type of data, resulting in more efficient resource utilisation.

The techniques developed in this study provide a promising framework for tackling multivariate skew polynomial systems, thus opening avenues for further innovations in related applications.

**Author Contributions:** Conceptualisation, R.R. and A.S.S.; methodology, O.J.; software, R.R.; validation, R.R., A.S.S. and O.K.; analysis, R.R. and A.S.S.; resources, A.S.S., and O.K.; data curation, R.R.; writing—original draft preparation, R.R.; writing—review and editing, O.J., A.S.S., and O.K.; visualisation, R.R.; supervision, A.S.S., O.K.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All related data will be provided upon request.

**Acknowledgments:** The authors would like to acknowledge the support given by the Cyber Security Research Group (CSRG), Nottingham Trent University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Burger, R.; Heinle, A. A New Primitive for a Diffie-Hellman-like Key Exchange Protocol Based on Multivariate Ore Polynomials, 2015, [arxiv:cs, math/1407.1270]. <https://doi.org/10.48550/arXiv.1407.1270>.
2. Ore, O. Theory of Non-Commutative Polynomials. *Annals of Mathematics* **1933**, *34*, 480–508.
3. Chyzak, F.; Salvy, B. Non-commutative Elimination in Ore Algebras Proves Multivariate Identities. Technical Report RR-2799, INRIA, 1996.
4. Collins, G.E. Subresultant and reduced polynomial remainder sequences. *ACM Communications in Computer Algebra* **1967**, *14*, 128–142.
5. Li, Z. A subresultant theory for Ore polynomials with applications. In Proceedings of the ISSAC '98, 1998.
6. Erić, A.L. The resultant of non-commutative polynomials. *Matematički Vesnik* **2008**, *60*, 3–8.
7. Rasheed, R. Modular Methods for Solving Nonlinear Polynomial Systems. Master's thesis, University of Western Ontario, London, Ontario, 2007.
8. Caruso, X.; Borgne, J. Fast Multiplication for Skew Polynomials. In Proceedings of the Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2017, Kaiserslautern, Germany, July 25–28, 2017; Burr, M.A.; Yap, C.K.; Din, M.S.E., Eds. ACM, 2017, pp. 77–84. <https://doi.org/10.1145/3087604.3087617>.
9. Giesbrecht, M.; Huang, Q.; Schost, E. Sparse Multiplication for Skew Polynomials. In Proceedings of the Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation, New York, NY, USA, 2020; ISSAC 2020, pp. 194–201. <https://doi.org/10.1145/3373207.3404023>.
10. Rasheed, R. Resultant-based Elimination for Skew Polynomials. In Proceedings of the 2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2021, pp. 11–18. <https://doi.org/10.1109/SYNASC54541.2021.00014>.
11. Chyzak, F.; Salvy, B. Non-commutative elimination in Ore algebras proves multivariate identities. *J. Symbolic Comput* **1998**, *26*, 187–227.
12. Mora, T. *Solving Polynomial Equation Systems*; Number v. 4 in Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2003.
13. Bueso, J.L.; Gómez-Torrecillas, J.; Verschoren, A. *Algorithmic Methods in Non-Commutative Algebra: Applications to Quantum Groups*; Mathematical Modelling: Theory and Applications, Springer Netherlands, 2003.
14. Carpentier, S.; Mikhailov, A.; Wang, J. Rational recursion operators for integrable differential-difference equations. *Commun. Math. Phys.* **2019**, *370*, 807 – 851. <https://doi.org/10.1007/s00220-019-03548-8>.
15. Draxl, P. A lifting of the Dieudonné determinant and applications concerning the multiplicative group of a skew field. Translated by D.W. Morris (2019) from the original German manuscript "Eine Liftung der Dieudonné-Determinante und Anwendungen die multiplikative Gruppe eines Schiefkörpers betreffend", in Teil II of P. Draxl and M. Kneser, pages 101–116.
16. Cohn, P.M. *Free Ideal Rings and Localization in General Rings*; New Mathematical Monographs, Cambridge University Press, 2006. <https://doi.org/10.1017/CBO9780511542794>.
17. Taelman, L. Dieudonné determinants for skew polynomial rings. *Journal of Algebra and Its Applications* **2006**, *05*, 89–93. <https://doi.org/10.1142/S0219498806001600>.
18. McConnell, J.; Robson, J.; Small, L. *Noncommutative Noetherian Rings*; Graduate studies in mathematics, American Mathematical Society, 2001.
19. Khalkhali, M. *Basic Noncommutative Geometry*; EMS series of lectures in mathematics, European Mathematical Society, 2013.
20. Giesbrecht, M.; Kim, M.S. Computing the Hermite form of a matrix of Ore polynomials. *Journal of Algebra* **2013**, *376*, 341 – 362. <https://doi.org/10.1016/j.jalgebra.2012.11.033>.
21. Bell, W. *Special Functions for Scientists and Engineers*; Dover books on mathematics, Dover Publications, 2004.
22. Basu, S.; Pollack, R.; Roy, M.F.; Cohen, A.M.; Cohen, H.; Eisenbud, D.; Singer, M.F.; Sturmfels, B. *Algorithms in Real Algebraic Geometry*; Vol. 10, *Algorithms and Computation in Mathematics*, Springer: Berlin, Heidelberg, 2006. <https://doi.org/10.1007/3-540-33099-2>.
23. Mishra, B. *Algorithmic Algebra*; Springer: New York, NY, 1993. <https://doi.org/10.1007/978-1-4612-4344-1>.
24. Cox, D.; Little, J.; O'Shea, D. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*; Undergraduate Texts in Mathematics, Springer International Publishing, 2015.
25. Cohn, P.M. Free rings and their relations, 2nd Ed. *London Math. Soc. Monograph No. 19* **1985**.
26. Bucher, D.; Ulmer, F. Linear codes using skew polynomials with automorphisms and derivations. *Designs, Codes and Cryptography* **2013**, *70*. <https://doi.org/10.1007/s10623-012-9704-4>.
27. Lam, T.; Leroy, A. Vandermonde and Wronskian matrices over division rings. *Journal of Algebra* **1988**, *119*, 308–336.
28. Conrad, K. Infinite Galois Theory (Draf, CTNT 2020). Technical report, UCONN, 2020.
29. Stewart, I. *Algebraic Number Theory and Fermat's Last Theorem*; CRC Press, Boca Raton, 4th edition 2016.
30. Hachenberger, D. *Finite Fields: Normal bases and completely free elements*; The Springer International Series in Engineering and Computer Science, Springer US, 2012.

- 
31. Lenstra, H. A normal basis theorem for infinite Galois extensions. *Indagationes Mathematicae (Proceedings)* **1985**, *88*, 221–228. [https://doi.org/https://doi.org/10.1016/1385-7258\(85\)90009-5](https://doi.org/https://doi.org/10.1016/1385-7258(85)90009-5). 934  
935
32. Hachenberger, D. Universal normal bases for the abelian closure of the field of rational numbers. *Acta Arithmetica* **2000**, *93*. <https://doi.org/10.4064/aa-93-4-329-341>. 936  
937