# Hermite form computation of matrices of differential polynomials

by

Myung Sub Kim

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

# Abstract

Given a matrix $A \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ over the ring of differential polynomials, we first prove the existence of the Hermite form $H$ of $A$ over this ring. Then we determine degree bounds on $U$ and $H$ such that $UA = H$. Finally, based on the degree bounds on $U$ and $H$, we compute the Hermite form $H$ of $A$ by reducing the problem to solving a linear system of equations over $\mathsf{F}(t)$. The algorithm requires a polynomial number of operations in $\mathsf{F}$ in terms of the input sizes: $n$, $\deg_{\mathcal{D}} A$, and $\deg_t A$. When $\mathsf{F} = \mathbb{Q}$ it requires time polynomial in the bit-length of the rational coefficients as well.

# Acknowledgements

First, I would like to thank God who always leads me to the best way in my life. The LORD is my shepherd, I shall not be in want.

Second, I would like to thank my supervisor, Dr. Mark Giesbrecht. Without his passionate supervision, I could not achieve any result of this thesis. Not only his supervision but also his generous financial support allowed me to experience the latest research skills by attending international conferences. I would also like to thank Dr. George Labahn and Dr. Arne Storjohann for serving as my thesis readers.

Third, I would like to give thanks to Reinhold Burger, Wei Zhou, and other SCG members. In particular, I have had a lot of interesting discussions with Reinhold Burger. Also, his polite attitude made me comfortable while sharing office with him.

Fourth, I would like to thank my parents, Young Sul Kim and Hwa Young Park for their endless love and sacrifice. Also, I would like to thank my wife, Hye Sun Min, and kids, Isaac and Jane. Whenever I am discouraged, their existence makes me move forward.

# Dedication

*This thesis is dedicated to my wife, Hye Sun Min.*

# Contents

**1 Introduction** **1**

1.1 Applications . . . 2

1.2 Main Results . . . 4

1.3 Outline . . . 5

**2 Preliminaries** **6**

2.1 Basic Definitions and Notations . . . 6

2.2 Normal Forms . . . 9

**3 Basic Operations in $\mathsf{F}[t][\mathcal{D};\delta]$** **13**

**4 Computing GCRD and LCLM over $\mathsf{F}[t][\mathcal{D};\delta]$** **19**

4.1 Preliminaries . . . 19

4.2 On computing the GCRD over $\mathsf{F}[t][\mathcal{D};\delta]$ . . . 21

4.3 On computing the LCLM over $\mathsf{F}[t][\mathcal{D};\delta]$ . . . 27

**5 Computing the Hermite form of matrices over $\mathsf{F}(t)[\mathcal{D};\delta]$** **31**

5.1 Previous Work . . . 32

5.2 Naive Approach . . . 33

5.3 Degree bounds for $H$ and $U$ . . . 38

5.4 The linear system method for Hermite form . . . 40

5.5 Generalization to rectangular matrices . . . 47

5.6 Generalization to matrices over $\mathsf{F}(t)[\mathcal{E};\sigma]$ . . . 48

**6 Experimental Results** **50**

**7 Conclusion and Future Work** **52**

**Bibliography** **56**

# Chapter 1

# Introduction

Ore polynomials are mathematical generalizations of linear differential, difference, and q-difference polynomials. These polynomials differ from ordinary polynomials in that the ring multiplication is noncommutative. Since Ore [22], many researchers have studied the mathematical properties of these rings. The Ore polynomials share many properties with the usual polynomials. Because the Ore polynomials are a left (and right) ideal domain, they admit a unique monic greatest common right (left) divisors (GCRDs) and least common left (right) multiples (LCLMs). Those GCRD and LCLM computations are important in that the GCRD of two linear differential polynomials represents the intersection of the solution spaces of them. On the other hand, the LCLM represents the sum of the solution spaces. The GCRD and LCLM computations can be used to compute canonical forms of the Ore polynomial matrices. In commutative rings, canonical forms have played significant roles because they expose useful mathematical properties such as rank and equivalence and can be used in computing other operations. Canonical forms of matrices over the Ore polynomials are also useful because solving systems of differential and difference operators can be transformed into working with matrices over the same domain. The most well-known canonical forms for commutative rings are the Popov [24], Hermite [14], and Smith forms [25]. The Popov form is useful for representing high-order terms with respect to lower-order terms. On the other hand, the Hermite form can be used to solve a system of linear Diophantine equations over a ring because it is triangular. The Popov and Hermite forms are canonical with respect to multiplication by a unimodular matrix on one side. The Smith/Jacobson form determines equivalence with respect to left and right multiplication by a unimodular matrix. Also, we can use the Smith form to check if two given matrices are equivalent. For example, if two matrices $A$ and $B$ are equivalent, which means they have the same properties such as rank, then they have the same Smith form. Since Kannan [16] proposed the first polynomial-time algorithm to compute the Hermite and Smith forms over $\mathbb{Z}$, many researchers have developed fast algorithms for computing these canonical forms over $\mathbb{Z}$ and $\mathsf{F}[x]$. However, compared to the development of fast algorithms for computing these canonical forms over commutative rings, algorithms over non-commutative rings such as the Ore polynomial ring have only been explored recently. A main difficulty in developing fast algorithms for non-commutative rings is that many fast algorithms over a commutative ring make use of the properties such as determinant, which are not generally available in non-commutative rings. In addition, since the entries of matrices in non-commutative rings are not commutative, techniques based on the property of commutativity in linear algebra can not be directly applied. Nevertheless, we are interested in extending this algorithmic technology for matrices over $\mathbb{Z}$ and $\mathsf{F}[x]$ to matrices over differential rings.

In this thesis, we focus on developing algorithms for a differential polynomial ring though most presented algorithms can be extended to the Ore polynomial rings. In particular, we consider the differential polynomials over a rational function field $\mathsf{F}(t)$, where $\mathsf{F}$ is a field of characteristic zero, typically an extension of $\mathbb{Q}$, or some representation of $\mathbb{C}$. The main result of this thesis is to give a polynomial-time algorithm to compute the Hermite form of a matrix over differential polynomials in terms of matrix size, entry degree, and coefficient size.

## 1.1 Applications

The main benefit from the Hermite form computation is that it is very useful for solving system of linear Diophantine equations. For example, we can consider the following system of equations:

$$\begin{aligned} ty_1(t) + (t+t^2)y_2(t) + ty_2(t)' - y_3(t)' &= -t \\ ty_1(t)' + ty_2(t) + (t^2+t+1)y_2(t)' + ty_2(t)'' + y_3(t) + (t+\frac{1}{t})y_3(t)' &= 1 \end{aligned}$$

This system of equations can be represented by the matrix multiplication as follows:

$$\begin{bmatrix} t & t^2+t+t\mathcal{D} & -\mathcal{D} \\ t\mathcal{D} & t+(t^2+t+1)\mathcal{D}+t\mathcal{D}^2 & 1+(t+\frac{1}{t})\mathcal{D} \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} -t \\ 1 \end{bmatrix}.$$

Here we employ Hermite form computation in order to solve the above system. Let

$$A = \begin{bmatrix} t & t^2+t+t\mathcal{D} & -\mathcal{D} \\ t\mathcal{D} & t+(t^2+t+1)\mathcal{D}+t\mathcal{D}^2 & 1+(t+\frac{1}{t})\mathcal{D} \end{bmatrix},$$

and compute the Hermite form of $A$ such that $UA = H$ as follows:

$$\begin{bmatrix} \mathcal{D} & -1 \\ \frac{1}{t}-\mathcal{D} & 1 \end{bmatrix} \cdot A = \begin{bmatrix} 1 & t+1 & -1-(t+\frac{1}{t})\mathcal{D}-\mathcal{D}^2 \\ 0 & \mathcal{D} & 1+t\mathcal{D}+\mathcal{D}^2 \end{bmatrix}.$$

By using backward substitution we solve for

$$\begin{bmatrix} \mathcal{D} & -1 \\ \frac{1}{t}-\mathcal{D} & 1 \end{bmatrix} \cdot A \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} -t\mathcal{D}-2 \\ t\mathcal{D}+1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & t+1 & -1-(t+\frac{1}{t})\mathcal{D}-\mathcal{D}^2 \\ 0 & \mathcal{D} & 1+t\mathcal{D}+\mathcal{D}^2 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \end{bmatrix} = \begin{bmatrix} -t\mathcal{D}-2 \\ t\mathcal{D}+1 \end{bmatrix}.$$

We now have the new system of Diophantine equations, which has same solutions as the previous one:

$$\begin{aligned} y_1(t) + (t+1)y_2(t) - y_3(t) - (t+\frac{1}{t})y_3(t)' - y_3(t)'' &= -t\mathcal{D} - 2 \\ y_2(t)' + y_3(t) + ty_3(t)' + y_3(t)'' &= t\mathcal{D} + 1. \end{aligned}$$

From the equations, we see that

$$\begin{aligned} y_2(t)' + y_3(t) + ty_3(t)' + y_3(t)'' &= t\mathcal{D} + 1 \\ y_2(t)' &= -y_3(t) - ty_3(t)' - y_3(t)'' + t\mathcal{D} + 1 \\ \mathcal{D}y_2(t) &= -\mathcal{D}ty_3(t) - \mathcal{D}^2y_3(t) + \mathcal{D}t. \end{aligned}$$

Since both sides of the equation are divisible by $\mathcal{D}$ on the left, we know that the system of Diophantine equations has a solution. One possible solution is given below.

$$\begin{aligned} y_1(t) &= -2 - t - t^2 - t\mathcal{D}, \\ y_2(t) &= t, \\ y_3(t) &= 0. \end{aligned}$$

## 1.2 Main Results

In this thesis we consider matrices whose entries are differential polynomials. In the first part of the thesis we present a thorough discussion in terms of the GCRD and LCLM computations of differential polynomials. We compute the upper bound of the size of coefficients for those two operations. Then we propose an algorithm to compute the Hermite form of differential polynomial matrices by using the GCRD and LCLM operations. The main results of this thesis can be summarized as follows:

- We prove the existence of the Hermite form over the differential polynomial ring and show the uniqueness of the Hermite form.
- We compute degree bounds on $U$ and $H$ for the differential polynomial matrix

$A$, where $H$ is the Hermite form of $A$ and $U$ the corresponding unimodular matrix. That is, they have the relationship $UA = H$.

- We show how to reduce solving this differential system of equations into solving a system of (usual) linear equations over a (commutative) field, based on the degree bounds on $U$ and $H$. Based on this, we propose a polynomial-time algorithm to compute the Hermite form of differential polynomial matrices.

- We generalize our algorithm for rectangular matrices and the shift polynomial ring.

The main results of this thesis have been reported in [12].

## 1.3 Outline

This thesis is organized as follows. In Chapter 2 we define some basic properties of differential polynomial rings and notation which is used through the thesis. In Chapter 3 we consider the basic operations of differential polynomials and analyze the cost of each operation. In Chapter 4 we explore an algorithm, which is presented in [6], for computing the GCRD of differential polynomials and then analyze the cost of the algorithm. We also introduce a new algorithm for computing the LCLM of two differential polynomials. In Chapter 5 we first present a naive algorithm, using the GCRD and LCLM computations, for computing the Hermite form of a matrix over the differential polynomial ring. Then we propose a polynomial-time algorithm for the Hermite form computation. To the best of our knowledge, it is the first polynomial-time algorithm for the Hermite form computation. In Chapter 6 we compare an implementation of our polynomial-time algorithm with the naive method for the Hermite form computation of a matrix over the differential polynomial ring.

# Chapter 2

# Preliminaries

In this chapter we give some definitions used throughout the thesis and discuss normal forms of matrices over $\mathsf{F}(t)[\mathcal{D};\delta]$.

## 2.1 Basic Definitions and Notations

A differential indeterminate $\mathcal{D}$ is adjoined to a field (typically a function field) to form the ring of *differential polynomials* $\mathsf{F}(t)[\mathcal{D};\delta]$. The ring $\mathsf{F}(t)[\mathcal{D};\delta]$ consists of the polynomials in $\mathsf{F}(t)[\mathcal{D}]$ under the usual addition and a non-commutative multiplication such that $\mathcal{D}a = a\mathcal{D} + \delta(a)$, where $\delta(a) = a'$ (the usual derivative of $a$) for any $a \in \mathsf{F}(t)$. In this thesis, a polynomial in $\mathsf{F}(t)[\mathcal{D};\delta]$ is typically written with respect to the differential variable $\mathcal{D}$ as

$$f = f_0 + f_1\mathcal{D} + f_2\mathcal{D}^2 + \cdots + f_m\mathcal{D}^m, \tag{2.1}$$

where $f_0, \ldots, f_m \in \mathsf{F}(t)$, with $f_m \neq 0$. Note that we insist on writing the $f_i$ to the left of the $\mathcal{D}^i$ to make the representation unique (the ring is non-commutative). Ore [22] defines all such operators in a unified way as follows.

**Definition 2.1.** (**Ore polynomial ring**) Let $\mathbb{K}$ be a commutative field and $\sigma$ an *isomorphism* of $\mathbb{K}$. A function $\delta : \mathbb{K} \to \mathbb{K}$ is called a *pseudo-derivation* with respect to $\sigma$ if it satisfies

$$\delta\left(a+b\right)=\delta a+\delta b \text{ and } \delta\left(ab\right)=\sigma\left(a\right)\delta\left(b\right)+\delta\left(a\right)b, \text{ for any } a,b\in\mathbb{K}.$$

The set of polynomials in $\mathbb{K}\left[\mathcal{X};\sigma,\delta\right]$ is called an Ore polynomial ring if the ring multiplication obeys the following rule

$$\mathcal{X}a=\sigma\left(a\right)\mathcal{X}+\delta\left(a\right), \text{ for all } a\in\mathbb{K}.$$

Generally we will work over a function field $\mathbb{K}=\mathsf{F}(t)$. In addition, we are most interested in the differential and shift operators among Ore operators. We here define the differential and shift polynomial rings as follows.

**Definition 2.2.** (**Differential polynomial ring**) Let $\mathsf{F}$ be a commutative field of characteristic zero and $\mathcal{D}$ denote differentiation with respect to the independent variable t. We set $\delta(f(t))=\frac{d}{dt}f(t)$ and $\sigma\left(f(t)\right)=f(t)$. Then we call the set of polynomials in $\mathsf{F}(t)[\mathcal{D};\delta]$ a differential polynomial ring.

**Definition 2.3.** (**Shift polynomial ring**) Let $\mathsf{F}$ be a commutative field of characteristic zero and $\mathcal{E}$ denote a linear shift operator such that $\mathcal{E}f(t)=f(t+1)\mathcal{E}$. We set $\sigma\left(f(t)\right)=f(t+1)$ and $\delta(f(t))=f(t)$. Then we call the set of polynomials in $\mathsf{F}(t)[\mathcal{E};\sigma]$ a shift polynomial ring.

The differential and shift polynomials are only two of many examples of Ore polynomial rings. We refer readers to [22] for more detail about Ore rings.

**Definition 2.4.** For $u,v\in\mathsf{F}[t]$ relatively prime, we can define $\deg_t(u/v)=\max\{\deg_t u,$ $\deg_t v\}$. This is extended to $f\in\mathsf{F}(t)[\mathcal{D};\delta]$ as in (2.1) by letting $\deg_t f=\max_i\{\deg_t f_i\}$.

**Definition 2.5.** Let $u,v\in\mathsf{F}[t]$ and $\deg_t u,\deg_t v\leq l$. Then $M(l)$ denotes the cost of multiplying $u$ and $v$.

In this thesis, we consider a matrix over $\mathsf{F}(t)[\mathcal{D};\delta]$ where each entry is a differential polynomial. We think of $\deg_t$ as measuring coefficient size or height. Let $A\in\mathsf{F}(t)[\mathcal{D};\delta]^{m\times n}$. We use the following conventions for the thesis.

**Definition 2.6.** (**Matrix notation**)

1. $A_{ij}$ denotes the element in the $i$-th row and $j$-th column of $A$.
2. $A_{i*}$ denotes the $i$-th row of $A$; $A_{*j}$ the $j$-th column of $A$.
3. $\|A\|$ denotes the maximum entry of $A$, and $A_{i\sim j,k\sim l}$ the submatrix of $A$ consisting of the $i$ to $j$-th rows and the $k$ to $l$-th columns.
4. The $i$-th principal minor of $A$ is the submatrix of $A$ consisting of the first $i$ rows and the first $i$ columns.

We consider as input size not only matrix dimensions but also entry degrees with respect to $\mathcal{D}$ and $t$.

**Definition 2.7.** Let $\deg_{\mathcal{D}} A = d$ denote the degree of $A$ in $\mathcal{D}$ so that $d \in \mathbb{Z}_{\geq 0} \cup \{-\infty\}$ is the smallest integer such that $\deg_{\mathcal{D}} A_{ij} \leq d$ for $1 \leq i \leq m$, $1 \leq j \leq n$. Let $\deg_t A = e$ denotes the degree of $A$ in $t$, so that $\deg_t A_{ij} \leq e$ for $1 \leq i \leq m$, $1 \leq j \leq n$. Thus, each entry $A_{ij}$ can be expressed as

$$A_{ij} = A_{ij0} + A_{ij1}\mathcal{D} + \ldots + A_{ij(d-1)}\mathcal{D}^{(d-1)} + A_{ijd}\mathcal{D}^d$$

where $A_{ijk} \in \mathsf{F}(t)$.

Also, it is sometimes useful to discuss the degree of each row of a matrix instead of the degree of the whole matrix.

**Definition 2.8.** (**Row degree**) A matrix $A \in \mathsf{F}(t)[\mathcal{D};\delta]^{m\times n}$ has row degree $\overrightarrow{u} = (u_1, \ldots, u_n) \in (\mathbb{Z}_{\geq 0} \cup \{-\infty\})^m$ if the $i$-th row of $A$ has degree $u_i$ in $\mathcal{D}$. We write $rowdeg\,\overrightarrow{u}$.

The leading row coefficient matrix is used to check if a matrix is in row-reduced form.

**Definition 2.9.** (**Leading row coefficient matrix**) Let $A$ and $\overrightarrow{u}$ be same as those in Definition 2.8 . Set $N = \deg_{\mathcal{D}} A$ and $S = diag(\mathcal{D}^{N-u_1}, \ldots, \mathcal{D}^{N-u_m})$. We write

$$SA = L\mathcal{D}^N + \text{lower degree terms in } \mathcal{D},$$

where the matrix $L = LC_{row}(A) \in \mathsf{F}(t)^{m\times n}$ is called the *leading row coefficient matrix* of $A$.

The following example shows how the row degree and leading row coefficient matrix are computed from a differential polynomial matrix.

**Example 2.10.** Let

$$A = \begin{bmatrix} -2t - \mathcal{D}^2 & t^2 + 1 + t\mathcal{D} \\ t + 1 + t^2\mathcal{D} & 3t + 2t\mathcal{D} \end{bmatrix} \in \mathsf{F}(t)[\mathcal{D};\delta]^{2\times 2}.$$

Then it is clear that the row degree of $A$ is $\overrightarrow{u} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ and $\deg_{\mathcal{D}} A = 2$. So, we set $S = \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{D} \end{bmatrix}$ and then computing $SA$ gives the leading row coefficient matrix of $A$:

$$\begin{aligned} SA &= \begin{bmatrix} 1 & 0 \\ 0 & \mathcal{D} \end{bmatrix} \begin{bmatrix} -2t - \mathcal{D}^2 & t^2 + 1 + t\mathcal{D} \\ t + 1 + t^2\mathcal{D} & 3t + 2t\mathcal{D} \end{bmatrix} \\ &= \begin{bmatrix} -2t - \mathcal{D}^2 & t^2 + 1 + t\mathcal{D} \\ 1 + (3t+1)\mathcal{D} + t^2\mathcal{D}^2 & 3 + (3t+2)\mathcal{D} + 2t\mathcal{D}^2 \end{bmatrix} \\ &= \begin{bmatrix} -1 & 0 \\ t^2 & 2t \end{bmatrix} \mathcal{D}^2 + \text{lower degree terms in } \mathcal{D}. \end{aligned}$$

Thus, $LC_{row}(A) = \begin{bmatrix} -1 & 0 \\ t^2 & 2t \end{bmatrix}$.

**Definition 2.11.** For $A \in \mathsf{F}(t)[\mathcal{D};\delta]$, the leading coefficient of $A$ in $\mathcal{D}$ is denoted by $\mathrm{LC}(A)$.

## 2.2 Normal Forms

Normal forms of a matrix are a unique representation of an equivalence class of matrices in $\mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$. We consider one-sided equivalence, so the normal form of

$A$ is same as the normal form of $UA$ where $U$ is any unimodular matrix. In other words, we say that $A$ and $B$ are row equivalent if there exists a unimodular matrix $V$ such that $A = VB$. Moreover, when $A$ and $B$ are row equivalent, their rows generate the same $\mathsf{F}(t)[\mathcal{D};\delta]$-module.

**Definition 2.12.** (**Unimodular matrix**) Let $U \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ and suppose there exists a $V \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ such that $UV = I_n$, where $I_n$ is the identity matrix over $\mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$. Then $U$ is called a *unimodular* matrix over $\mathsf{F}(t)[\mathcal{D};\delta]$.

This definition is in fact symmetric, in that $V$ is also unimodular, as shown in the following lemma.

**Lemma 2.13.** *Let $U \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ be unimodular such that there exists a $V \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ with $UV = I_n$. Then $VU = I_n$ as well.*

*Proof.* We multiply $UV = I_n$ on the right by $U$, which gives $UVU = U$, or equivalently $U(VU - I_n) = 0$. Since $U$ has a right inverse, we know that the free module spanned by the columns of $U$ has row rank $n$, and that the columns of $U$ are $\mathsf{F}(t)[\mathcal{D};\delta]$-linear independent. It follows that there is no non-zero column vector $w$ such that $Uw = 0$. Thus, $VU - I_n$ must be zero and $VU = I_n$. □

**Definition 2.14.** (**Row equivalence**) Let $A, B \in \mathsf{F}(t)[\mathcal{D};\delta]$. It is said that $A$ is row equivalent to $B$ if there exists a unimodular matrix $U$ such that $UA = B$.

**Definition 2.15.** (**Row-reduced form**) A matrix $T \in \mathsf{F}(t)[\mathcal{D};\delta]^{m\times n}$ with rank $r$ is in row-reduced form if $T$ has $r$ nonzero rows and $rank\ LC_{row}(T) = r$.

The row-reduced form can be used for finding the rank and left nullspace of a matrix of differential polynomials. Also, it is shown in [3] that a row reduction algorithm can be used for computing a weak Popov form of a matrix of skew polynomials.

**Definition 2.16.** (**Row Popov form**) Let $G \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ and let $d_i$ denote the $i$-the row degree. $G$ is in Popov form if it satisfies the following properties.

1. The diagonal entries are monic and $\deg_{\mathcal{D}} G_{ii} = d_i$ for $1 \le i \le n$.

2. All entries in a column have degrees lower than that of the diagonal element.

3. The leading row coefficient is triangular.

**Example 2.17.** Let

$$G = \begin{bmatrix} 1+\mathcal{D}^2 & 2+\mathcal{D} & 1+t\mathcal{D}^3 \\ 2t+t\mathcal{D} & t+\mathcal{D}^3 & 4t+t^2 \\ 3+2t\mathcal{D} & \mathcal{D}^2 & 7+8t+\mathcal{D}^3 \end{bmatrix}.$$

$G$ is not in Row Popov form because it does not obey the second requirement. So, we multiply $G$ on the left as follows:

$$G' = \begin{bmatrix} 1 & 0 & -t \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} G = \begin{bmatrix} 1-3t-2t^2\mathcal{D}+\mathcal{D}^2 & 2+\mathcal{D}-t\mathcal{D}^2 & 1-7t-8t^2 \\ 2t+t\mathcal{D} & t+\mathcal{D}^3 & 4t+t^2 \\ 3+2t\mathcal{D} & 2+t\mathcal{D}+\mathcal{D}^2 & 7+8t+\mathcal{D}^3 \end{bmatrix}.$$

Since $G'$ satisfies all requirements, $G'$ is in Row Popov form.

One of the advantages of using the Popov form is that the highest degree of the entries of the Popov form of a matrix is no greater than that of the matrix. So, the Popov form is widely used in linear system control theory. However, since a matrix in Popov form is not triangular, it is not as directly useful for solving systems of linear Diophantine equations. The Hermite form was first introduced in 1851 for integer matrices by Hermite [14]. He used row operations to compute the triangular matrix of an integer matrix. After Hermite, many researchers [16, 15, 10, 30, 26, 27, 21] have investigated this special form not only for integer domain but also for other domains such as the ordinary polynomial domain. Here we define the Hermite form over the differential polynomial ring.

**Definition 2.18.** (**Hermite form**) Let $H \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ with full row rank. The matrix $H$ is in *Hermite form* if $H$ is upper triangular, if every diagonal entry of $H$ is monic, and if every off-diagonal entry of $H$ has degree (in $\mathcal{D}$) strictly lower than the degree of the diagonal entry below it.

**Example 2.19.** Let

$$A = \left[ \begin{array}{ccc} 1+(t+2)\mathcal{D}+\mathcal{D}^2 & 2+(2t+1)\mathcal{D} & 1+(1+t)\mathcal{D} \\ 2t+t^2+t\mathcal{D} & 2+2t+2t^2+\mathcal{D} & 4t+t^2 \\ 3+t+(3+t)\mathcal{D}+\mathcal{D}^2 & 8+4t+(5+3t)\mathcal{D}+\mathcal{D}^2 & 7+8t+(2+4t)\mathcal{D} \end{array} \right].$$

Then $A$ has the Hermite form

$$H = \left[ \begin{array}{ccc} 2+t+\mathcal{D} & 1+2t & \frac{-2+t+2t^2}{2t}-\frac{1}{2t}\mathcal{D} \\ 0 & 2+t+\mathcal{D} & 1+\frac{7t}{2}+\frac{1}{2}\mathcal{D} \\ 0 & 0 & -\frac{2}{t}+\frac{-1+2t+t^2}{t}\mathcal{D}+\mathcal{D}^2 \end{array} \right],$$

and unimodular matrix

$$U = \left[ \begin{array}{ccc} \frac{1-t}{2t} & \frac{1}{t}+\frac{1}{2t}\mathcal{D} & -\frac{1}{2t} \\ \frac{t}{2}-\frac{1}{2}\mathcal{D} & -\frac{1}{2}\mathcal{D} & \frac{1}{2} \\ \frac{1+2t^2}{t}+(t-1)\mathcal{D} & \frac{2}{t}+\frac{1-2t}{t}\mathcal{D}-\mathcal{D}^2 & -\frac{1}{t}+\mathcal{D} \end{array} \right].$$

One can check that $UA = H$.

# Chapter 3

# Basic Operations in $\mathsf{F}[t][\mathcal{D};\delta]$

In this chapter we discuss the basic operations in $\mathsf{F}(t)[\mathcal{D};\delta]$ and some straightforward algorithms for their computation. We make no claim that these algorithms are the most efficient algorithms possible, only that they are reasonable, and more importantly are completely analyzable. First, some well-known properties of $\mathsf{F}(t)[\mathcal{D};\delta]$ are worth recalling; see [4] for an algorithmic presentation of this theory. $\mathsf{F}(t)[\mathcal{D};\delta]$ is a left and right principal ideal domain. Given $f, g \in \mathsf{F}(t)[\mathcal{D};\delta]$, this implies the existence of a right (and left) division with remainder algorithm such that there exists unique $q, r \in \mathsf{F}(t)[\mathcal{D};\delta]$ such that $f = qg + r$ where $\deg_{\mathcal{D}}(r) < \deg_{\mathcal{D}}(g)$. This allows for a right (and left) euclidean-like algorithm which shows the existence of a greatest common right divisor, $h = \mathrm{GCRD}(f, g) \in \mathsf{F}(t)[\mathcal{D};\delta]$, a polynomial of maximum degree (in $\mathcal{D}$) such that $f = uh$ and $g = vh$ for $u, v \in \mathsf{F}(t)[\mathcal{D};\delta]$. The GCRD is unique up to a left multiple in $\mathsf{F}(t)\backslash\{0\}$, and there exist co-factors $a, b \in \mathsf{F}(t)[\mathcal{D};\delta]$ such that $af + bg = \mathrm{GCRD}(f, g)$. There also exists a least common left multiple $\mathrm{LCLM}(f, g) \in \mathsf{F}(t)[\mathcal{D};\delta]$. Analogously, there exists a greatest common left divisor, $\mathrm{GCLD}(f, g)$, and least common right multiple, $\mathrm{LCRM}\,(f, g)$, both of which are unique up to a right multiple in $\mathsf{F}(t)\backslash\{0\}$. The complexity and further properties of some of these operations are explored in [18].

Let

$$f = \sum_{i=0}^{n} f_i \mathcal{D}^i \in \mathsf{F}[t][\mathcal{D};\delta], \quad g = \sum_{j=0}^{m} g_j \mathcal{D}^j \in \mathsf{F}[t][\mathcal{D};\delta]$$

for $f_0, \ldots, f_n, g_0, \ldots, g_m \in \mathsf{F}[t]$. In general we will work with differential polynomials in $\mathsf{F}[t][\mathcal{D}; \delta]$, as opposed to $\mathsf{F}(t)[\mathcal{D}; \delta]$ and will manage denominators explicitly. This will make our computations and their analyses simpler. Assume that $\deg_t f \le d$, $\deg_t g \le d$, and $f_n, g_m \neq 0$. The sum $f + g$ is computed coefficient-wise, and so $\deg_{\mathcal{D}}(f+g) \le \max\{\deg_{\mathcal{D}} f, \deg_{\mathcal{D}} g\}$ and $\deg_t(f+g) = \max\{\deg_t f, \deg_t g\}$. We write the product $h = fg = h_0 + h_1\mathcal{D} + \cdots + h_{n+m}\mathcal{D}^{n+m}$. Without loss of generality we may suppose $n \ge m$. The costs of addition and subtraction are given by $O(nd)$ field operations in $\mathsf{F}$. Expanding the multiplication, we find

$$h = fg = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{d} f_i \mathcal{D}^i g_{k,j} t^k \mathcal{D}^j = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{d} f_i g_{k,j} \mathcal{D}^i t^k \mathcal{D}^j,$$

where $g_j = \sum_{k=0}^{d} g_{k,j} t^k$ for $g_{k,j} \in \mathsf{F}$. Bostan et al. [2] show, by using Leibniz's formula, the canonical form of $\mathcal{D}^i t^k$ (with coefficients in $\mathsf{F}[t]$ on the left) can be computed in $O(\min(i,k))$ field operations in $\mathsf{F}$. For example,

$$\mathcal{D}^i t^k = \sum_{l=0}^{\min(i,k)} (k)_l \binom{i}{l} t^{k-l} \mathcal{D}^{i-l},$$

where $(k)_{l+1} = (k)_l(k-l)$ and so the canonical form of $\mathcal{D}^i t^k$ is computed in $O(\min(i,k))$ operations. Thus, the total cost of computing the canonical form of the product is

$$\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{d} \min(i,k) \le \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{d} k \quad \in O(nmd^2).$$

Since if $\mathcal{D}^k g = \sum_{j=0}^{k} g_j \mathcal{D}^j$ where $g, g_j \in \mathsf{F}[t]$ then $\deg_t g_j \le d$ because $\deg_t g \le d$, the cost of polynomial multiplications in $\mathsf{F}[t]$ when computing $f \cdot g$ is bounded by $O(nmd^2)$ operations as well. We have the following.

**Lemma 3.1.** *Let $f, g \in \mathsf{F}[t][\mathcal{D}; \delta]$ with $n = \deg_{\mathcal{D}} f$ and $m = \deg_{\mathcal{D}} g$ and $\deg_t f \le d$ and $\deg_t g \le d$. The product $h = fg$ has $\deg_{\mathcal{D}} h = n + m$ and $\deg_t h \le 2d$. The cost of computing $h$ is $O(nmd^2)$ operations in $\mathsf{F}$.*

We can similarly analyze division with remainder. Suppose $f$ and $g$ are the same as above with the property $n \geq m$. We want to find $k, q,$ and $r$ such that

$$kf = q \cdot g + r$$

where $k \in \mathsf{F}[t]$, $q$ and $r \in \mathsf{F}[t][\mathcal{D};\delta]$ with $\deg_{\mathcal{D}} r < \deg_{\mathcal{D}} g$. The introduction of $k$ cancels denominators introduced by the leading coefficient of $g$.

**Lemma 3.2.** *Let $f \in \mathsf{F}[t][\mathcal{D};\delta]$ where $deg_{\mathcal{D}} f = n$ and $LC(f) = f_n$. Then the leading coefficient of the canonical form of $\mathcal{D}^i f$ is $f_n$ for $i \geq 0$.*

*Proof.* The proof follows easily by induction on $i$.

□

We construct the matrix $G$ as follows:

$$G = \begin{bmatrix} g_0^{[0]} & g_1^{[0]} & \cdots & g_{m-1}^{[0]} & g_m^{[0]} & & & \\ g_0^{[1]} & g_1^{[1]} & \cdots & \cdots & g_m^{[1]} & g_{m+1}^{[1]} & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & \\ g_0^{[n-m-1]} & g_1^{[n-m-1]} & \cdots & \cdots & \cdots & \cdots & g_{n-1}^{[n-m-1]} & \\ g_0^{[n-m]} & g_1^{[n-m]} & \cdots & \cdots & \cdots & \cdots & \cdots & g_n^{[n-m]} \end{bmatrix}.$$

The $i$-th row of the matrix $G$ corresponds to the coefficients of the canonical form of $\mathcal{D}^i g$. In other words, $\mathcal{D}^{n-m} g = g_n^{[n-m]} \mathcal{D}^n + g_{n-1}^{[n-m]} \mathcal{D}^{n-1} + \cdots + g_1^{[n-m]} \mathcal{D} + g_0^{[n-m]}$ where $g_i^{[n-m]} \in \mathsf{F}[t]$ and $\deg_{\mathcal{D}} g = m$. Then three row vectors $F$, $Q$, and $R$ are constructed as follows:

$F = \begin{pmatrix} f_0 & f_1 \cdots & f_n \end{pmatrix}$, $Q = \begin{pmatrix} q_0 & q_1 \cdots & q_{n-m} \end{pmatrix}$, and $R = \begin{pmatrix} r_0 & r_1 \cdots & r_{m-1} 0 \cdots 0 \end{pmatrix}$

where $R \in \mathsf{F}(t)^{1 \times (n+1)}$ and its last $n-m+1$ entries are all zeros. Given $G$ and $F$, we would like to find $Q$ and $R$ such that

$$F = Q \cdot G + R.$$

Moreover, we note that the row vector $Q$ can be solved for using the submatrix of

$G$ and the subvector of $F$:

$$F' = Q \cdot G',$$

where $F'$ and $G'$ are obtained from $F$ and $G$ respectively by taking the last $n-m+1$ columns. As shown above, $G'$ is a lower triangular matrix, and by Lemma 3.2 all diagonal entries are equal to $\mathrm{LC}\,(g)$. Since $G'$ is the triangular matrix, we can use backward substitution to find all entries of $Q$:

$$Q_i = \frac{F'_i - \sum_{j=i+1}^{n-m+1} G'_{j,i} Q_j}{\mathrm{LC}(g)} \text{ for } i = n-m+1, n-m, \ldots, 1.$$

**Lemma 3.3.** *Let $K_i = (\mathrm{LC}\,(g))^{n-m+2-i}\, Q_i$. Then $K_i \in \mathsf{F}[t]$ for $i = n-m+1, n-m, \ldots, 1$.*

*Proof.* We prove this by induction on $i$. For the base case $i = n-m+1$, we use backward substitution:

$$K_{n-m+1} = (\mathrm{LC}\,(g)) \frac{F'_{n-m+1}}{\mathrm{LC}\,(g)} = F'_{n-m+1} = f_n.$$

Since $f_n \in \mathsf{F}[t]$, $K_{n-m+1} \in \mathsf{F}[t]$. We assume our claim is true for $i > r$ where $r < n-m+1$ and need to show that $K_r \in \mathsf{F}[t]$. By using backward substitution, we know that

$$\begin{aligned} K_r &= (\mathrm{LC}\,(g))^{n-m+2-r}\, Q_r \\ &= (\mathrm{LC}\,(g))^{n-m+2-r}\, \frac{F'_r - \sum_{j=r+1}^{n-m+1} G'_{j,r} Q_j}{\mathrm{LC}(g)} \\ &= (\mathrm{LC}\,(g))^{n-m+1-r} \left( F'_r - \sum_{j=r+1}^{n-m+1} G'_{j,r} Q_j \right). \end{aligned}$$

By the induction hypothesis, we know that $\sum_{j=r+1}^{n-m+1} (\mathrm{LC}\,(g))^{n-m+1-r}\, Q_j \in \mathsf{F}[t]$ and hence $K_r \in \mathsf{F}[t]$.

□

By Lemma 3.3, we know that $(\mathrm{LC}\,(g))^{n-m+1}\, Q \in \mathsf{F}[t]^{1\times(n-m+1)}$ and so let $k =$

$(\mathrm{LC}(g))^{n-m+1}$, $q = (\mathrm{LC}(g))^{n-m+1}(Q_{n-m+1}\mathcal{D}^{n-m} + Q_{n-m}\mathcal{D}^{n-m-1} + \cdots + Q_2\mathcal{D} + Q_1)$, where $Q_i$ is the $i$-th entry of the matrix $Q$. Accordingly, the row vector $R$ can be computed as follows:

$$(\mathrm{LC}(g))^{n-m+1} R = (\mathrm{LC}(g))^{n-m+1} F - (\mathrm{LC}(g))^{n-m+1} Q \cdot G. \tag{3.1}$$

**Lemma 3.4.** $\deg_t (\mathrm{LC}(g))^{n-m+1} Q_i \leq d \cdot (n-m+1)$ *for* $i = n-m+1, n-m, \cdots, 1$.

*Proof.* We prove this by induction on $i$. For the base $i = n-m+1$, by using backward substitution, we get

$$(\mathrm{LC}(g))^{n-m+1} Q_{n-m+1} = (\mathrm{LC}(g))^{n-m+1} \frac{F'_{n-m+1}}{\mathrm{LC}(g)} = (\mathrm{LC}(g))^{n-m} f_n.$$

Since $\deg_t f_n \leq d$ and $\deg_t g \leq d$, $\deg_t \left((\mathrm{LC}(g))^{n-m} f_n\right) \leq d \cdot (n-m+1)$. We now assume our claim is true for $i > r$ where $r < n-m+1$ and need to show that $\deg_t \left((\mathrm{LC}(g))^{n-m+1} Q_r\right) \leq d \cdot (n-m+1)$. By using backward substitution, we know that

$$\begin{aligned} (\mathrm{LC}(g))^{n-m+1} Q_r &= (\mathrm{LC}(g))^{n-m+1} \frac{F'_r - \sum_{j=r+1}^{n-m+1} G'_{j,r} Q_j}{\mathrm{LC}(g)} \\ &= (\mathrm{LC}(g))^{n-m} \left(F'_r - \sum_{j=r+1}^{n-m+1} G'_{j,r} Q_j\right). \end{aligned}$$

By the inductive hypothesis, we know that $\deg_t \left(\sum_{j=r+1}^{n-m+1} (\mathrm{LC}(g))^{n-m+1} Q_j\right) \leq d \cdot (n-m+1)$ and so

$$\deg_t \left(\sum_{j=r+1}^{n-m+1} (\mathrm{LC}(g))^{n-m} Q_j\right) \leq d \cdot (n-m).$$

Then it follows that $\deg_t \left( (\mathrm{LC}\,(g))^{n-m} \left( F'_r - \sum_{j=r+1}^{n-m+1} G'_{j,i} Q_j \right) \right) \leq d \cdot (n-m+1)\,.$ □

Since $\deg_t \left( \mathrm{LC}\,(g)^{n-m+1} Q_i \right) \leq d \cdot (n-m+1)$ and $0 \leq (n-m)$, the cost of computing each $\mathrm{LC}\,(g)^{n-m+1} Q_i$ is dominated by the cost of computing $\sum_{j=i+1}^{n-m+1} \mathrm{LC}\,(g)^{n-m} G'_{j,i} Q_j$. In particular, the cost of the multiplication $\mathrm{LC}\,(g)^{n-m} G'_{j,i} Q_j$ is bounded by $M\,(d\,(n-m))$ because $\deg_t \left( \mathrm{LC}\,(g)^{n-m} Q_j \right) \leq d \cdot (n-m)$. When $i = 1$, we see that there exist at most $n-m$ polynomial multiplications, which are bounded by $O\,((n-m)\,M\,(d\,(n-m)))$. Since we need to compute $n-m+1$ entries of the matrix $\mathrm{LC}\,(g)^{n-m+1} Q$, the cost is $O\left( (n-m)^2 M\,(d\,(n-m)) \right)$ operations. Moreover, if $\log d \leq (n-m)^2$ then we can compute $\mathrm{LC}\,(g)^{n-m}$ with $O\left( (n-m)^2 M\,(d\,(n-m)) \right)$ operations. From equation (3.1), we see that the cost of computing the row vector $\mathrm{LC}\,(g)^{n-m+1} R$ is bounded by $O\,(m^2 M\,(d\,(n-m)))$. Thus, we have the following.

**Lemma 3.5.** *Let $f, g \in \mathsf{F}[t][\mathcal{D}; \delta]$, with $n = \deg_{\mathcal{D}} f$ and $m = \deg_{\mathcal{D}} g$ and $\deg_t f$, $\deg_t g \leq d$. Then there exists a $k \in \mathsf{F}[t]$ and $q, r \in \mathsf{F}[t][\mathcal{D}; \delta]$ such that $kf = qg + r$, and $\deg_{\mathcal{D}} r < m$, $\deg_t k$, $\deg_t q \leq d(n-m+1)$ and $\deg_t r \leq d(n-m+2)$. We can compute $k, q, r$ with $O(n^2 M(d(n-m)))$ operations in $\mathsf{F}$.*

# Chapter 4

# Computing GCRD and LCLM over $\mathsf{F}[t][\mathcal{D};\delta]$

## 4.1 Preliminaries

In this chapter we review the subresultant algorithm, which is introduced in Chardin [6], as generalized to computing the GCRD of two differential polynomials. Then we extend the idea of the algorithm to compute the LCLM of two differential polynomials. Li and Nemes [19] propose an efficient modular algorithm for the GCRD computation. The main difficulty is that evaluation mappings are not Ore ring homomorphisms. For example, let $\psi_k$ be an evaluation mapping such that $\psi_k : \mathsf{F}[t][\mathcal{D};\delta] \rightarrow \mathsf{F}[\mathcal{D};\delta]$ and then we see that $\psi_k(\mathcal{D}t) = k\mathcal{D}+1 \neq k\mathcal{D} = \psi_k(\mathcal{D})\psi_k(t)$. They overcome such a problem by applying the subresultant theory [18] for Ore polynomials. In other words, they evaluate $\psi_k(\mathrm{GCRD}(A_p, B_p))$ instead of evaluating $\mathrm{GCRD}(\psi_k(A_p), \psi_k(B_p))$ where $A_p, B_p \in \mathsf{F}_p[t][\mathcal{D};\delta]$. Since we need the GCRD and LCLM algorithms for the Hermite form computation of matrices over $\mathsf{F}[t][\mathcal{D};\delta]$, we will only go into the detail of the subresultant algorithm where we can determine an upper bound on the sizes of coefficients. Chardin defines a Sylvester-style expression for the differential subresultants. Here we redefine his expression to compute the GCRD of two differential polynomials. Let $f, g \in \mathsf{F}[t][\mathcal{D};\delta]$ where $\deg_{\mathcal{D}} f$ and $\deg_{\mathcal{D}} g$ are $n, m$ respectively with the property $m \leq n$. Then $f$ and $g$ can be expressed as $f = \sum_{i=0}^{n} f_i \mathcal{D}^i$ and $g = \sum_{i=0}^{m} g_i \mathcal{D}^i$. Since $\deg_{\mathcal{D}}(\mathcal{D}f) = \deg_{\mathcal{D}} f + 1$, we can express as $\mathcal{D}f = \sum_{i=0}^{n+1} f_i^{[1]} \mathcal{D}^i$ where each $f_i^{[1]} \in \mathsf{F}[t]$. Moreover, it can be generalized in such a way that $\mathcal{D}^j f = \sum_{i=0}^{n+j} f_i^{[j]} \mathcal{D}^i$. Now we consider the following equations:

$$\begin{aligned}
f_{n+m-1}^{[m-1]}\mathcal{D}^{n+m-1} + f_{n+m-2}^{[m-1]}\mathcal{D}^{n+m-2} + \cdots + f_0^{[m-1]} &= \mathcal{D}^{m-1} \cdot f, \\
f_{n+m-2}^{[m-2]}\mathcal{D}^{n+m-2} + f_{n+m-3}^{[m-2]}\mathcal{D}^{n+m-3} + \cdots + f_0^{[m-2]} &= \mathcal{D}^{m-2} \cdot f, \\
&\cdots\cdots \\
f_n^{[0]}\mathcal{D}^{n} + f_{n-1}^{[0]}\mathcal{D}^{n-1} + \cdots + f_0^{[0]} &= f, \\
g_{n+m-1}^{[n-1]}\mathcal{D}^{n+m-1} + g_{n+m-2}^{[n-1]}\mathcal{D}^{n+m-2} + \cdots + g_0^{[n-1]} &= \mathcal{D}^{n-1} \cdot g, \\
g_{n+m-2}^{[n-2]}\mathcal{D}^{n+m-2} + g_{n+m-3}^{[n-2]}\mathcal{D}^{n+m-3} + \cdots + g_0^{[n-2]} &= \mathcal{D}^{n-2} \cdot g, \\
&\cdots\cdots \\
g_m^{[0]}\mathcal{D}^{m} + g_{m-1}^{[0]}\mathcal{D}^{m-1} + \cdots + g_0^{[0]} &= g.
\end{aligned}$$

Based on the above equations, we form the Sylvester matrix of $f$ and $g$, where the dimension of the matrix is $(m+n) \times (m+n)$ :

$$M = \begin{bmatrix}
f_{n+m-1}^{[m-1]} & f_{n+m-2}^{[m-1]} & \cdots & \cdots & f_1^{[m-1]} & f_0^{[m-1]} \\
 & f_{n+m-2}^{[m-2]} & f_{n+m-3}^{[m-2]} & \cdots & f_1^{[m-2]} & f_0^{[m-2]} \\
 & \cdots & \cdots & \cdots & \cdots & \\
 & \cdots & f_n^{[0]} & \cdots & f_1^{[0]} & f_0^{[0]} \\
g_{n+m-1}^{[n-1]} & g_{n+m-2}^{[n-1]} & \cdots & \cdots & g_1^{[n-1]} & g_0^{[n-1]} \\
 & g_{n+m-2}^{[n-2]} & g_{n+m-3}^{[n-2]} & \cdots & g_1^{[n-2]} & g_0^{[n-2]} \\
 & & \cdots & \cdots & \cdots & \\
 & & g_m^{[0]} & \cdots & g_1^{[0]} & g_0^{[0]}
\end{bmatrix} \in \mathsf{F}[t]^{(m+n)\times(m+n)}.$$

Thus, we have the following definition.

**Definition 4.1.** Let $M$ be the Sylvester matrix of the differential polynomials $f$ and $g$. Then the submatrix $M_j^i$ is obtained from $M$ by deleting:

- rows 1 to $j$,

- rows $m+1$ to $m+j$,

- columns 1 to $j$,
- columns $n+m-j$ to $n+m$ except column $n+m-i$.

In general, the right divisor of a differential polynomial will not be a left divisor and so we need the following definition.

**Definition 4.2.** Let $f$ be a differential polynomial. If $g$ is a right divisor of $f$ then we denote by $g \mid_r f$. Accordingly, if $g$ is a left divisor of $f$ then we denote by $g \mid_l f$.

## 4.2 On computing the GCRD over $\mathsf{F}[t][\mathcal{D};\delta]$

In this thesis we do not prove correctness of the subresultant algorithm because this is established elsewhere [6, 17]. Instead, based on the subresultant algorithm we develop sharp upper bounds on the sizes of coefficients of the GCRD.

**Fact 4.3.** *(Li, 1996, Proposition 2.2.3) Let $f$,$g$, and $M$ be same as those in Definition 4.1. Then the GCRD $r$ of $f$ and $g$ can be computed as follows:*

$$r = \sum_{i=0}^{k} \det\left(M_k^i\right)\mathcal{D}^i$$

*where $k = \deg_{\mathcal{D}} r$.*

**Example 4.4.** Let $f = (t^2+t)\mathcal{D}^2 + 2t\mathcal{D} - 2$ and $g = t^3\mathcal{D}^2$. Suppose we want to find the GCRD $r$ of $f$ and $g$ by using the subresultant algorithm. First we construct the Sylvester matrix of $f$ and $g$:

$$M = \begin{bmatrix} t^2+t & 4t+1 & 0 & 0 \\ 0 & t^2+t & 2t & -2 \\ t^3 & 3t^2 & 0 & 0 \\ 0 & t^3 & 0 & 0 \end{bmatrix} \in \mathsf{F}(t)^{4\times 4}.$$

For simplicity, suppose we already know the degree of $r$, which is equal to 1. Thus, by Fact 4.3,

$$\begin{aligned} r &= \det\left(M_1^1\right)\mathcal{D} + \det\left(M_1^0\right) \\ &= -2t^4\mathcal{D} + 2t^3. \end{aligned}$$

We observe that $f = (-\frac{t+1}{2t^3}\mathcal{D} + \frac{t+3}{2t^4}) \cdot r$ and $g = (-\frac{1}{2t}\mathcal{D} + \frac{3}{2t^3}) \cdot r$. Since there is no common right factor of $(-\frac{t+1}{2t^3}\mathcal{D} + \frac{t+3}{2t^4})$ and $(-\frac{1}{2t}\mathcal{D} + \frac{3}{2t^3})$, $r$ must be the GCRD of $f$ and $g$.

By observing the subresultant algorithm, it is clear that we can compute the GCRD of two differential polynomials in polynomial-time in the matrix dimension. On the other hand, we are more concerned about the size of coefficients. In the following lemma we determine the upper bound of the size of coefficients.

**Lemma 4.5.** *Let $f$, $g$, and $M$ be same as those in Definition 4.1 and GCRD $(f, g) = r$. In addition, suppose $\deg_t f, \deg_t g \leq d$. Then $\deg_t r \leq 2dn$ and there exist $u$,$v \in \mathsf{F}[t]$ such that $uf + vg = h$ where $\deg_t u \leq 2dn$ and $\deg_t v \leq 2dn$.*

*Proof.* By Fact 4.3, the GCRD $r$ of $f$ and $g$ can be computed as follows:

$$r = \sum_{i=0}^{k} \det\left(M_k^i\right)\mathcal{D}^i,$$

where $k = \deg_{\mathcal{D}} r$. Since $M_k^i$ is a submatrix of $M$, we know that $\deg_t\left(\det\left(M_k^i\right)\right)$ is bounded by $d\,(m+n)$ and hence $\deg_t r \leq 2dn$. In [5], the author shows by induction that $\deg_{\mathcal{D}} u \leq m-k-1$ and $\deg_{\mathcal{D}} v \leq n-k-1$. Thus, we form row vectors $H$ and $R$ as follows:

$$H = [u_{m-k-1}, u_{m-k-2}, \ldots, u_1, u_0, v_{n-k-1}, v_{n-k-2}, \ldots, v_1, v_0] \in \mathsf{F}[t]^{1\times(n+m-2k)},$$

$$R = \left[0, 0, \ldots, \det\left(M_k^k\right)\right] \in \mathsf{F}[t]^{1\times(n+m-2k)}.$$

By the definition of the Sylvester matrix, we have the following:

$$H \cdot M_k^k = R.$$

Since $\det\left(M_k^k\right) \neq 0$, $M_k^k$ is invertible. By using Cramer's rule, we compute the inverse of $M_k^k$, $\left(M_k^k\right)^{-1} = \frac{\operatorname{adj}(M_k^k)}{\det(M_k^k)}$. Now we solve for $H$:

$$\begin{aligned} H &= R \cdot \left(M_k^k\right)^{-1} \\ &= R \cdot \frac{\operatorname{adj}(M_k^k)}{\det(M_k^k)} \\ &= [0, \ldots, 0, 1] \cdot \operatorname{adj}(M_k^k). \end{aligned}$$

Since $\deg_t\left(\operatorname{adj}(M_k^k)\right) \leq 2dn$, $\deg_t(H) \leq 2dn$. Therefore, $\deg_t u \leq 2dn$ and $\deg_t v \leq 2dn$.

□

**Example 4.6.** Let $f$ and $g$ be as in Example 4.4. We know from Example 4.4 that

$$M_1^1 = \begin{bmatrix} t^2+t & 2t \\ t^3 & 0 \end{bmatrix},$$

and so,

$$\operatorname{adj}(M_1^1) = \begin{bmatrix} 0 & -2t \\ -t^3 & t^2+t \end{bmatrix}.$$

Then, by Lemma 4.5, $u = -t^3$ and $v = t^2 + t$. We verify this result by computing:

$$\begin{aligned} uf + vg &= -t^3((t^2+t)\mathcal{D}^2 + 2t\mathcal{D} - 2) + (t^2+t)(t^3\mathcal{D}^2) \\ &= -2t^4\mathcal{D} + 2t^3 = r. \end{aligned}$$

We now present the subresultant algorithm which returns not only the GCRD of $f$ and $g$ but also $u$ and $v$ such that $uf + vg = r$ where $r = \operatorname{GCRD}(f, g)$.

**Lemma 4.7.** *Let $f, g \in \mathsf{F}[t][\mathcal{D}; \delta]$ where $\deg_{\mathcal{D}} f$ and $\deg_{\mathcal{D}} g$ are $n, m$ respectively with the property $m \leq n$ and $\deg_t f$, $\deg_t g \leq d$. Then the expected number of field operations of the subresultant algorithm is $O^{\sim}(mn^{\omega}d)$ where $\omega$ is the exponent of matrix multiplication over $\mathsf{F}$ and the soft-O notation $O^{\sim}$ indicates some missing logarithmic factors.*

**Algorithm 1** Subresultant algorithm for GCRD of two differential polynomials

```
Require: $f, g \in \mathsf{F}(t)[\mathcal{D}; \delta]$ with the property $\deg_{\mathcal{D}} f \geq \deg_{\mathcal{D}} g$
procedure GCRD($f, g$)
    $n \leftarrow \deg_{\mathcal{D}} f$
    $m \leftarrow \deg_{\mathcal{D}} g$
    $M \leftarrow SylvesterMatrix(f, g)$        ▷ $M$ is the Sylvester Matrix of $f$ and $g$
    for $k \leftarrow 0, m$ do
        if $\det(M_k^k) \neq 0$ then
            break
        end if
    end for
    $r \leftarrow \sum_{i=0}^{k} \det\left(M_k^i\right) \mathcal{D}^i$
    $A \leftarrow \mathrm{adj}(M_k^k)$        ▷ $A$ is the Adjoint Matrix of $M_k^k$
    $u \leftarrow \sum_{i=1}^{m-k} A_{(n+m-2k,i)} \mathcal{D}^{m-k-i}$
    $v \leftarrow \sum_{i=1}^{n-k} A_{(n+m-2k,m-k+i)} \mathcal{D}^{n-k-i}$
    return $(r, u, v)$        ▷ $r = uf + vg$
end procedure
```

*Proof.* It is clear that the cost of the subresultant algorithm is bound by the cost of finding the degree of the GCRD of $f$ and $g$. Storjohann [28] shows the determinant of a polynomial matrix with degree $d$ can be computed in $O^{\sim}(n^{\omega} d)$ field operations. Since in the worst case we have to compute determinants $(m+1)$ times, the algorithm is bounded by $O^{\sim}(mn^{\omega} d)$ field operations. □

We are also interested in computing the GCRD of several differential polynomials. The most obvious way is to compute GCRDs iteratively as follows:

$$\mathrm{GCRD}(f_1, f_2, \ldots, f_{l-1}, f_l) = \mathrm{GCRD}(f_1, \mathrm{GCRD}(f_2, \ldots, \mathrm{GCRD}(f_{l-1}, f_l) \ldots))$$

However, this iterative computation is hard to analyze because it is not clear how many division with remainder operations are used for computing the GCRD of differential polynomials. The following algorithm demonstrates that the number of division with remainder operations should be less than $n + l$, where $l$ is the number of differential polynomials, and $n = \max(\deg_{\mathcal{D}} f_1, \ldots, \deg_{\mathcal{D}} f_l)$.

For each recursive call of the algorithm either the degree or the number of dif-

**Algorithm 2** Iterative algorithm for GCRD of several differential polynomials

```
Require: $f_1, f_2, \ldots, f_{l-1}, f_l$ are in order of decreasing degree with respect to $\deg_{\mathcal{D}}$
procedure ITR-GCRD($f_1, f_2, \ldots, f_{l-1}, f_l$)
    if $l = 1$ or $\deg_{\mathcal{D}} f_1 = 0$ then
        return $\deg_{\mathcal{D}} f_1$
    end if
    $r \leftarrow \mathrm{rem}(f_1, f_l)$
    if $r = 0$ then
        return ITR-GCRD($f_2, \ldots, f_{l-1}, f_l$)
    else
        return ITR-GCRD($f_2, \ldots, f_l, r$)
    end if
end procedure
```

ferential polynomials is decreased by 1 so that there exist at most $n + l$ division with remainder operations. However, by Lemma 3.5, the degree of coefficient polynomials in $t$ increases more than double at each operation. Thus, we see that a naive approach can not run in polynomial time in terms of the size of coefficients. To the best of our knowledge, Grigor'ev [13] presents the first polynomial-time algorithm computing the GCRD of several differential polynomials. His approach is very similar to the subresultant algorithm in that he creates a Sylvester-style matrix based on input differential polynomials. Then he reduces rows and columns by using Gaussian elimination. On the other hand, we can compute the GCRD of several differential polynomials by using probabilistic approach as follows.

**Algorithm 3** Probabilistic algorithm for GCRD of several differential polynomials

```
Require: $f_1, \dots, f_l \in \mathsf{F}(t)[\mathcal{D};\delta]$ and $l \geq 2$
procedure GCRD($f_1, \dots, f_l$)
    $d \leftarrow \deg_{\mathcal{D}} f_1$
    $\hat{f}_1 \leftarrow f_1$
    randomly choose $c_3, \dots, c_l$ such that $c_i \in \mathsf{F}$
    $\hat{f}_2 \leftarrow f_2 + c_3 f_3 + \cdots + c_l f_l$
    $(\hat{g}, u, v) \leftarrow$ GCRD($\hat{f}_1, \hat{f}_2$)                ▷ $\hat{g} = u\hat{f}_1 + v\hat{f}_2$
    return $(\hat{g}, < u, v >, < c_3, \dots, c_l >)$ ;
end procedure
```

**Lemma 4.8.** *Let $f_1, f_2, \dots, f_{l-1}, f_l \in \mathsf{F}[t][\mathcal{D};\delta]$, $\deg_{\mathcal{D}} f_1 \leq d$, and $l \geq 2$. Then for randomly chosen $c_i \in \mathrm{R}, 3 \leq i \leq l$ and $\mathrm{R} \subset \mathsf{F}$ we have*

$$\Pr\left( GCRD_{1 \leq i \leq l}\left(f_i\right) = GCRD\left( f_1, f_2 + \sum_{i=3}^{l} c_i f_3 \right) \right) \geq 1 - \frac{d}{\#(\mathrm{R})}.$$

*Proof.* The proof is based on Diaz and Kaltofen's Lemma 2 [8]. Let

$$\hat{f}_1 = f_1,\ \hat{f}_2 = f_2 + \sum_{i=3}^{l} \gamma_i f_i \in \mathsf{E}[\mathcal{D}],\ \mathsf{E} = \mathsf{F}[\gamma_3, \dots, \gamma_l][t],$$

where $\gamma_3, \gamma_4, \dots, \gamma_{l-1}, \gamma_l$ are indeterminants. Let $g = \mathrm{GCR}D_{1 \leq i \leq l}\left(f_i\right)$, and so $g \mid_r \hat{f}_1$ and $g \mid_r \hat{f}_2$. Then we let $\hat{g} = \mathrm{GCRD}(\hat{f}_1, \hat{f}_2)$. Since $\hat{g} \mid_r f_1$ and since $\hat{g} \in \mathsf{F}[t][\mathcal{D};\delta]$, it follows that $g \mid_r \hat{g}$. By using the way of evaluating, we can show that $\hat{g}$ divides $f_i$ for $1 \leq i \leq l$. For example, if we set $\gamma_3 = \gamma_4 = \cdots = \gamma_l = 0$ then we have $\hat{f}_2 = f_2$ and then $\hat{g} \mid_r f_2$. Since $\hat{g} \mid_r f_i$ for $3 \leq i \leq l$, $\hat{g} \mid_r g$. Hence, we conclude that $g = \hat{g}$. By Diaz and Kaltofen's Lemma 1 [8], we know that GCRD $(\phi_{c_3,\cdots,c_l}(\hat{f}_1)$, $\phi_{c_3,\cdots,c_l}(\hat{f}_2)) = \phi_{c_3,\cdots,c_l}(\mathrm{GCRD}(\hat{f}_1, \hat{f}_2))$ if $\phi_{c_3,\cdots,c_l}(\det(M_j^j)) \neq 0$ where $M_j^j$ is a submatrix of the Sylvester matrix $M$ of $\hat{f}_1$ and $\hat{f}_2$, $j = \deg_{\mathcal{D}}\left(\mathrm{GCRD}(\hat{f}_1, \hat{f}_2)\right)$, and $\phi$ is an evaluation function, evaluating at $c_3 = \gamma_3, \cdots, c_l = \gamma_l$. Since $\det(M_j^j) \in \mathsf{F}[\gamma_3, \dots, \gamma_l][t]$ and since the total degree of $\gamma_i$ is less than and equal to $d$, by Schwartz-Zippel Theorem, $\Pr\left(\phi_{c_3,\cdots,c_l}(\det(M_j^j)) \neq 0\right) \geq 1 - \frac{d}{\#(\mathrm{R})}$. Thus, since if

$\mathrm{GCRD}(\phi_{c_3,\cdots,c_l}(\hat{f}_1), \phi_{c_3,\cdots,c_l}(\hat{f}_2)) = \phi_{c_3,\cdots,c_l}(\mathrm{GCRD}(\hat{f}_1, \hat{f}_2))$ then $g = \hat{g}$, the stated probability is established.

□

In this section we have seen algorithms for computing the GCRD of differential polynomials. However, we need not only a GCRD algorithm but also a LCLM algorithm in order to compute the Hermite form of differential polynomial matrices. Thus, in the following section we will explore an algorithm for computing the LCLM of two differential polynomials based on the result in this section.

## 4.3 On computing the LCLM over $\mathsf{F}[t][\mathcal{D}; \delta]$

Since the LCLM computation can be thought as an extension of the GCRD computation, we can compute the LCLM of two differential polynomials based on the subresultant algorithm for the GCRD computation. Let $f, g \in \mathsf{F}[t][\mathcal{D}; \delta]$ where $\deg_{\mathcal{D}} f$ and $\deg_{\mathcal{D}} g$ are $n, m$ respectively with the property $m \leq n$ and $\deg_t f$, $\deg_t g \leq d$. Then suppose $k = \deg_{\mathcal{D}} \mathrm{GCRD}\,(f, g)$ and it follows that $\deg_{\mathcal{D}} \mathrm{LCLM}\,(f, g) = n + m - k$. Moreover, we note that there exist $r$ and $s$ such that $rf + sg = 0$, where $\deg_{\mathcal{D}} r = m - k$ and $\deg_{\mathcal{D}} s = n - k$. Now, we construct a $(n + m - 2k + 2) \times (n + m - k + 1)$ matrix $A$ in such a way that

$$A = \begin{bmatrix} f^{[m-k]}_{n+m-k} & f^{[m-k]}_{n+m-k-1} & \cdots & \cdots & f^{[m-k]}_{1} & f^{[m-k]}_{0} \\ & f^{[m-k-1]}_{n+m-k-1} & \cdots & \cdots & \cdots & f^{[m-k-1]}_{0} \\ & \cdots & \cdots & \cdots & \cdots & f^{[1]}_{0} \\ & & f^{[0]}_{n} & \cdots & f^{[0]}_{1} & f^{[0]}_{0} \\ g^{[n-k]}_{n+m-k} & g^{[n-k]}_{n+m-k-1} & \cdots & \cdots & g^{[n-k]}_{1} & g^{[n-k]}_{0} \\ & g^{[n-k-1]}_{n+m-k-1} & \cdots & \cdots & \cdots & g^{[n-k-1]}_{0} \\ & & \cdots & \cdots & \cdots & g^{[1]}_{0} \\ & & g^{[0]}_{m} & \cdots & g^{[0]}_{1} & g^{[0]}_{0} \end{bmatrix}.$$

As in the case of the Sylvester matrix of $f$ and $g$, the first row corresponds to the coefficients of the canonical form of $\mathcal{D}^{m-k} f$. Accordingly, the $(m - k + 1)$-th row corresponds to the coefficients of $f$. After the first $m - k + 1$ rows, the $i$-th row corresponds to the coefficients of $\mathcal{D}^{n-k-i+1}g$. For example, the first row after the first $m-k+1$ rows corresponds to the coefficients of the canonical form of $\mathcal{D}^{n-k}g$ and the last row the coefficients of $g$. From the matrix $A$, we see that the leftmost nonzero elements of the first $m-k+1$ row are equivalent to LC $(f)$ and that the leftmost nonzero elements of the remaining rows are equivalent to LC $(g)$. In order to find the coefficients of $r$ and $s$, we have to find two row vectors $R \in \mathsf{F}[t]^{1\times(m-k+1)}$ and $S \in \mathsf{F}[t]^{1\times(n-k+1)}$ such that

$$[R\ S] \cdot A = [0] \in \mathsf{F}[t]^{1\times(n+m-k+1)}.$$

**Example 4.9.** Let $f$ and $g$ be same as those in Example 4.4. Also, we have already seen $\deg_{\mathcal{D}} r = \mathrm{GCRD}(f,g) = 1$. Now, we construct the matrix $A$ in order to find the LCLM of $f$ and $g$:

$$A = \begin{bmatrix} t^2+t & 4t+1 & 0 & 0 \\ 0 & t^2+t & 2t & -2 \\ t^3 & 3t^2 & 0 & 0 \\ 0 & t^3 & 0 & 0 \end{bmatrix}.$$

Then we find row vectors $R$ and $S$ such that

$$\begin{aligned} [R\ S] \cdot A &= [0], \\ [-t^3, 0, t^2+t, t-2] \cdot A &= [0]. \end{aligned}$$

Thus, the LCLM of $f$ and $g$ is

$$\begin{aligned} -t^3 f &= -t^3\mathcal{D}((t^2+t)\mathcal{D}^2 + 2t\mathcal{D} - 2) \\ &= (-t^5 - t^4)\mathcal{D}^3 + (-4t^4 - t^3)\mathcal{D}^2 \\ &= -((t^2+t)\mathcal{D} + (t-2))t^3\mathcal{D}^2 \\ &= -((t^2+t)\mathcal{D} + (t-2))g. \end{aligned}$$

As shown in Example 4.9, the problem of computing the LCLM of $f$ and $g$ is reduced to the problem of finding the left nullspace vector of the matrix $A$.

**Algorithm 4** Left nullspace vector algorithm for LCLM of two differential polynomials

```
Require: $\deg_{\mathcal{D}} f \geq \deg_{\mathcal{D}} g$
procedure LCLM($f, g$)
    $n \leftarrow \deg_{\mathcal{D}} f$
    $m \leftarrow \deg_{\mathcal{D}} g$
    $k \leftarrow \deg_{\mathcal{D}} \mathrm{GCRD}(f, g)$
    $A \leftarrow \begin{bmatrix} f_{n+m-k}^{[m-k]} & \cdots & \cdots & f_0^{[m-k]} \\ & \cdots & \cdots & \cdots \\ & f_n^{[0]} & \cdots & f_0^{[0]} \\ g_{n+m-k}^{[n-k]} & \cdots & \cdots & g_0^{[n-k]} \\ & \cdots & \cdots & \cdots \\ & g_m^{[0]} & \cdots & g_0^{[0]} \end{bmatrix}$    ▷ $A \in \mathsf{F}[t]^{(n+m-2k+2)\times(n+m-k+1)}$
    $[R_{m-k}, \ldots, R_0, S_{n-k}, \ldots, S_0] \leftarrow \mathrm{LeftNullVector}(A)$
    $r \leftarrow \sum_{i=0}^{m-k} R_i \mathcal{D}^i$
    $s \leftarrow \sum_{i=0}^{n-k} S_i \mathcal{D}^i$
    return $(r, s)$    ▷ $rf + sg = 0$
end procedure
```

Since we know from [29] that the cost of computing the left nullspace vector of $f$ and $g$ is bounded by the cost of computing the GCRD of $f$ and $g$, we can compute the LCLM of $f$ and $g$ with $O^{\sim}(mn^{\omega}d)$ field operations as well. The existence of $r$ and $s$ such that $rf + sg = 0$ guarantees that the rank of $A$ should be less than $(n + m - 2k + 2)$. Otherwise, $r$ and $s$ must be equal to 0. Let $l = \mathrm{rank}\,(A)$ and $[T] = [R\ S]$. Since we are interested in $\deg_t r$ and $\deg_t s$, we can apply fraction free Gaussian elimination to find the row vector $T$. After permuting rows of $A$, we can make the matrix $A$ consist of two parts in such a way that the rows of the first part are linearly independent and that the rows of the second part are linearly dependent on the rows of the first part. If we suppose $A'$ is a row echelon form of A, the fraction free Gaussian elimination algorithm [11, pp. 393-399] guarantees that $\deg_t A' \leq d \cdot (n + m - 2k + 1) \leq d \cdot (n + m + 1) \leq 2dn + d$. Let $W$ be a matrix such that $W \cdot A = A'$. Then it follows by the fraction free Gaussian elimination algorithm that $\deg_t W \leq 2dn$ since $\deg_t A' \leq 2dn + d$ and since $\deg_t A \leq d$. Now, we can get the row vector $T$ by computing

$$T = [\underbrace{0 \cdots 0}_{\iota \text{ zeros}}\ 1 \cdots 1] \cdot W.$$

Thus, we have the following lemma.

**Lemma 4.10.** *Let $f$, $g$ be same as those in Definition 4.1. Then we can find $r$ and $s$ such that $rf + sg = 0$ and that $\deg_t r, \deg_t s \leq 2dn$ with $O^{\sim}(mn^{\omega}d)$ field operations.*

# Chapter 5

# Computing the Hermite form of matrices over $\mathsf{F}(t)[\mathcal{D};\delta]$

In this chapter we will first show how to compute the Hermite form of matrices over $\mathsf{F}(t)[\mathcal{D};\delta]$ by using LCLM and GCRD computations. However, as we have seen in Chapter 4, coefficients grow very rapidly when computing the GCRD and LCLM of entries. To remedy this we convert the problem of computing the Hermite form of a matrix over $\mathsf{F}(t)[\mathcal{D};\delta]$ into a problem of solving a system of equations in $\mathsf{F}(t)$. This gives a polynomial-time bound in terms of the number of operations in $\mathsf{F}(t)$ and the size of coefficients. The linear system method was first proposed for the Hermite form computation of polynomial matrices by Kaltofen et al. [10], and then improved by Storjohann [26]. Our approach can be thought as a variant of their methods for the differential polynomial ring. The main benefit of using the linear system method is that the field $\mathsf{F}(t)$ over which we solve is the usual, commutative, field of rational functions. For convenience, we assume that our matrix is over $\mathsf{F}[t][\mathcal{D};\delta]$ instead of $\mathsf{F}(t)[\mathcal{D};\delta]$, which can easily be achieved by clearing denominators with a scalar multiple from $\mathsf{F}[t]$. For example, we construct a diagonal matrix $D$ in such a way that the $i$-th diagonal entry of $D$ is the LCLM of all denominators of coefficients of entries in the $i$-th row of an input matrix. Thus, we can clear all denominators by multiplying $D$ on the left, which is clearly a unimodular operation. We first concern ourselves with matrices in $\mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ of full rank and then generalize our algorithm to rectangular matrices.

## 5.1 Previous Work

In commutative domains such as $\mathbb{Z}$ and $\mathsf{F}[x]$, it has been more common to compute the triangular Hermite and diagonal Smith form (as well as the lower degree Popov form, especially as an intermediate computation). These forms are canonical in the sense of being a unique invariant of their class under multiplication by unimodular matrices. Polynomial-time algorithms for the Smith and Hermite forms over $\mathsf{F}[x]$ were developed by Kannan (1985) [15], with important advances by Kaltofen et al. (1987) [10], Villard (1996) [30], Mulders and Storjohann (2003) [21], and many others. One of the key features of this recent work in computing normal forms has been a careful analysis of the complexity in terms of matrix size, entry degree, and coefficient swell. Clearly identifying and analyzing the cost in terms of all these parameters has led to a dramatic drop in both theoretical and practical complexity.

Computing the classical Smith and Hermite forms of matrices over differential (and more general Ore) domains has received less attention though normal forms of differential polynomial matrices have applications in solving differential systems and control theory. Abramov and Bronstein [1] analyze the number of reduction steps necessary to compute a row-reduced form, while Beckermann et al. (2006) [3] analyze the complexity of row reduction in terms of matrix size, degree and the sizes of the coefficients of some shifts of the input matrix. Beckerman et al. [3] demonstrate tight bounds on the degree and coefficient sizes of the output, which we will employ here. For the Popov form, Cheng [7] gives an algorithm for matrices of shift polynomials. Cheng's approach involves order bases computation in order to eliminate lower order terms of Ore polynomial matrices. A main contribution of Cheng is to give an algorithm computing the row rank and a row-reduced basis of the left nullspace of a matrix of Ore polynomials in a fraction-free way. This idea is extended in Davies et al. [23], they reduce the problem of computing Popov form to a nullspace computation. However, though Popov form is useful for rewriting high order terms with respect to low order terms, we want a different normal form more suited to solving a system of linear Diophantine equations. Since the Hermite form is upper triangular, it meets this goal nicely, not to mention the fact that it is a "classical" canonical form. In a slightly different vein, Middeke [20] has recently given an algorithm for the Smith (diagonal) form of a matrix of different polynomials, which requires time polynomial in the matrix size and degree (but the coefficient size is not analyzed).

## 5.2 Naive Approach

In commutative rings the GCD and LCM computations are the most popular techniques to compute the Hermite form of an input matrix. Also, those computations are unimodular operations over commutative rings. In particular, when Kannan and Bachem [16] proposed a polynomial-time algorithm to compute the Hermite form of an integer matrix, they showed that the problem of intermediate expression swell can be controlled by careful use of the GCD algorithm. If $r = \mathrm{GCD}(a, b)$ and $a \geq b$ then there exists $p$ and $q$ such that $r = pa + qb$. In [16], the authors replaces $p$ and $q$ with $p = p + \left\lfloor \frac{q}{a} \right\rfloor b$ and $q = q - \left\lfloor \frac{q}{a} \right\rfloor a$ respectively if $|q| > |a|$. This technique assures that the intermediate expressions such as $p$ and $q$ are controlled by the size of entries. However, such a technique can not be directly applied to polynomial rings because the size of the coefficients is difficult to controll. Thus, we believe that it will require a deeper technique to solve the problem of intermediate expression swell using GCRD and LCLM computations directly. The first step to prove the existence of the Hermite form over $\mathsf{F}(t)[\mathcal{D};\delta]$ is to show that GCRD and LCLM computations are unimodular operations.

**Theorem 5.1.** *Let $a, b \in \mathsf{F}(t)\,[\mathcal{D};\delta]$ be Ore polynomials. Then we can find $u, v \in \mathsf{F}(t)\,[\mathcal{D};\delta]$ such that $ua+vb = GCRD(a,b)$. Also, there exist $s, t$ such that $sa+tb = 0$ where $sa = LCLM(a,b)$. Then $\begin{pmatrix} u & v \\ s & t \end{pmatrix}$ is invertible over $\mathsf{F}(t)\,[\mathcal{D};\delta]$ and hence is unimodular. Also, we note that $\begin{pmatrix} u & v \\ s & -t \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}$.*

*Proof.* Since $sa = tb = \mathrm{LCLM}(a, b)$ and $\mathrm{GCLD}(s, t) = 1$, it follows that there exist

$c, d \in \mathsf{F}(t)\,[\mathcal{D};\delta]$ such that $sc - td = 1$. Now, we construct the inverse of $\begin{pmatrix} u & v \\ s & t \end{pmatrix}$ as follows:

$$\begin{pmatrix} u & v \\ s & -t \end{pmatrix}\begin{pmatrix} ag^{-1} & c \\ bg^{-1} & d \end{pmatrix} = \begin{pmatrix} 1 & uc+vd \\ 0 & 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & uc+vd \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & -uc-vd \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Also, we note that $c = ag^{-1}$ and $d = bg^{-1}$ and so $uc + vd = g$. Thus, the inverse of $\begin{pmatrix} u & v \\ s & t \end{pmatrix}$ is

$$\begin{aligned}\begin{pmatrix} ag^{-1} & c \\ bg^{-1} & d \end{pmatrix}\begin{pmatrix} 1 & -uc-vd \\ 0 & 1 \end{pmatrix} &= \begin{pmatrix} ag^{-1} & ag^{-1}\,(-uc-vd)+c \\ bg^{-1} & bg^{-1}\,(-uc-vd)+d \end{pmatrix} \\ &= \begin{pmatrix} ag^{-1} & -a+c \\ bg^{-1} & -b+d \end{pmatrix} \in \mathsf{F}(t)\,[\mathcal{D};\delta]^{2\times 2}.\end{aligned}$$

□

The following theorem is very important because it shows the Hermite form of an input matrix in the differential polynomial ring is a normal form of the input matrix. In other words, if the matrix $H$ is the Hermite form of the matrix $A$ then there exists a unimodular matrix $U$ such that $H = UA$.

**Theorem 5.2.** *Let $A \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ have row rank $n$. Then there exists a matrix $H \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ with row rank $n$ in Hermite form, and a unimodular matrix $U \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$, such that $UA = H$.*

*Proof.* We show this by induction on $n$. The base case, $n = 1$, is trivial and we suppose that the theorem holds for $(n-1)\times(n-1)$ matrices. Since $A$ has row rank $n$, we can find a permutation of the rows of $A$ such that every principal submatrix of $A$ has full row rank. Since this permutation is a unimodular transformation of $A$, we assume this property about $A$. Thus, by the induction hypothesis, there exists a unimodular matrix $U_1 \in \mathsf{F}(t)[\mathcal{D};\delta]^{(n-1)\times(n-1)}$ such that

$$\begin{pmatrix} & & & 0 \\ U_1 & & & 0 \\ & & & \vdots \\ & & & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \cdot A = \bar{H} = \begin{pmatrix} \bar{H}_{1,1} & \cdots & \cdots & * & * \\ & \bar{H}_{2,2} & \cdots & * & * \\ \text{\huge 0} & & \ddots & \vdots & \vdots \\ & & & \bar{H}_{n-1,n-1} & * \\ A_{n,1} & A_{n,2} & \cdots & A_{n,n-1} & A_{n,n} \end{pmatrix} \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n},$$

where the $(n-1)$st principal minor of $\bar{H}$ is in Hermite form. By Theorem 5.1, we know that there exists a unimodular matrix

$$W = \begin{pmatrix} u_i & v_i \\ s_i & -t_i \end{pmatrix} \in \mathsf{F}(t)[\mathcal{D};\delta]^{2\times 2} \text{ such that } W \begin{pmatrix} \bar{H}_{ii} \\ A_{n,i} \end{pmatrix} = \begin{pmatrix} g_i \\ 0 \end{pmatrix} \in \mathsf{F}(t)[\mathcal{D};\delta]^{2\times 1}.$$

This allows us to reduce $A_{n,1}, \ldots, A_{n,n-1}$ to zero, and does not introduce any nonzero entries below the diagonal. Also, all off-diagonal entries can be reduced using unimodular operations modulo the diagonal entry, putting the matrix into Hermite form.

□

**Corollary 5.3.** *Let $A \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ have full row rank. Suppose $UA = H$ for unimodular $U \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ and Hermite form $H \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$. Then both $U$ and $H$ are unique.*

*Proof.* Suppose $H$ and $G$ are both Hermite forms of $A$. Then there exist unimodular matrices $U$ and $V$ such that $UA = H$ and $VA = G$, and $G = WH$ where $W = VU^{-1}$ is unimodular. Since $G$ and $H$ are upper triangular matrices, we know $W$ is as well. Moreover, since $G$ and $H$ have monic diagonal entries, the diagonal entries of $W$ equal 1. We now prove $W$ is the identity matrix. By way of contradiction, first assume that $W$ is not the identity, so there exists an entry $W_{ij}$ which is the first nonzero off-diagonal entry on the $i$th row of $W$. Since $i < j$ and since $W_{ii} = 1$, $G_{ij} = H_{ij} + W_{ij}H_{jj}$. Because $W_{ij} \neq 0$, we see $\deg_{\mathcal{D}} G_{ij} \geq \deg_{\mathcal{D}} G_{jj}$, which contradicts the definition of the Hermite form. The uniqueness of $U$ follows similarly.

□

The following corollary is a generalization of Theorem 5.2 in terms of a rectangular matrix.

**Corollary 5.4.** *Let $A \in \mathsf{F}(t)\,[\mathcal{D};\delta]^{n\times m}$ with rank $m$. There exists a unique matrix $H \in \mathsf{F}(t)\,[\mathcal{D};\delta]^{n\times m}$ with rank $m$ in Hermite form, which is left equivalent to $A$.*

*Proof.* First we permute rows of $A$ in order that the first $m$ rows of $A$ are linearly independent. Then, by Theorem 5.2, the Hermite form $H$ of $A$ with rank $m$ can be computed using unimodular operations and then the last $n-m$ rows of the Hermite form should be zero because those are dependent on the first $m$ rows. Thus, by Corollary 5.3, $A$ has the unique hermite form $H$ with rank $m$.

□

We can now give a naive algorithm to compute the Hermite form of matrices over $\mathsf{F}(t)\,[\mathcal{D};\delta]$ by using the GCRD and LCLM computations.

**Algorithm 5** Naive algorithm for Hermite form computation

```
Require: $A$ is a square matrix with full rank
procedure HERMITE($A$)
    $n \leftarrow$ row-dimension$(A)$
    $H \leftarrow A$
    $U \leftarrow I_n$
    for $i \leftarrow 1, n$ do
        if $H_{i,i} = 0$ then
            Replace $H_{i,i}$ with non-zero entry by row permutation
        end if
        for $j \leftarrow i + 1, n$ do
            Calculate $r = pH_{i,i} + qH_{j,i}$                ▷ $r = \mathrm{GCRD}(H_{i,i}, H_{j,i})$
            Calculate $l = sH_{i,i} = tH_{j,i}$                ▷ $l = \mathrm{LCLM}(H_{i,i}, H_{j,i})$
            $V \leftarrow \begin{pmatrix} p & q \\ s & -t \end{pmatrix}$
            $\begin{pmatrix} H_{i,*} \\ H_{j,*} \end{pmatrix} \leftarrow V \begin{pmatrix} H_{i,*} \\ H_{j,*} \end{pmatrix}$
            $\begin{pmatrix} U_{i,*} \\ U_{j,*} \end{pmatrix} \leftarrow V \begin{pmatrix} U_{i,*} \\ U_{j,*} \end{pmatrix}$
        end for
        $c \leftarrow \mathrm{LC}(A_{i,i})$
        $A_{i,*} \leftarrow \frac{1}{c} A_{i,*}$
        $U_{i,*} \leftarrow \frac{1}{c} U_{i,*}$
        for $j \leftarrow 1, i - 1$ do
            Calculate $A_{j,i} = qA_{i,i} + r$
            $V \leftarrow \begin{pmatrix} 1 & -q \\ 0 & 1 \end{pmatrix}$
            $\begin{pmatrix} A_{j,*} \\ A_{i,*} \end{pmatrix} \leftarrow V \begin{pmatrix} A_{j,*} \\ A_{i,*} \end{pmatrix}$
            $\begin{pmatrix} U_{j,*} \\ U_{i,*} \end{pmatrix} \leftarrow V \begin{pmatrix} U_{j,*} \\ U_{i,*} \end{pmatrix}$
        end for
    end for
    return $(U, H)$                                        ▷ $H = UA$
end procedure
```

## 5.3 Degree bounds for $H$ and $U$

The main difficulty in computing with matrices over the differential polynomial ring is that there is no determinant having the properties found in commutative linear algebra. Due to such a restriction, we can not directly compute the degree bounds of $H$ and $U$ required by techniques used in Storjohann [26] and Kaltofen et al. [10]. The first question we must then answer is, can we find degree bounds of $H$ and $U$ without using the properties of determinant? In this section, we prove sufficient degree bounds on $H$ and $U$. Also, we employ the result of Beckermann et al. [3] with respect to upper bounds of row-reduced form.

**Fact 5.5.** *(Theorem 2.2 in [3]) For any $A \in \mathsf{F}(t)\left[\mathcal{D};\delta\right]^{m\times s}$ there exists a unimodular matrix $U \in \mathsf{F}(t)\left[\mathcal{D};\delta\right]^{m\times m}$, with $T = UA$ having $r \leq \min\{m,s\}$ nonzero rows, $rowdegT \leq rowdegA$, and where the submatrix consisting of the r nonzero rows of $T$ are row-reduced. Moreover, the unimodular multiplier satisfies the degree bound*

$$rowdeg\ U \leq \overrightarrow{v} + \left(|\overrightarrow{u}| - |\overrightarrow{v}| - \min_j\{u_j\}\right)\overrightarrow{e},$$

*where $\overrightarrow{u} := \max\left(\overrightarrow{0}, rowdegA\right)$, $\overrightarrow{v} := \max\left(\overrightarrow{0}, rowdegT\right)$, and $\overrightarrow{e}$ is the column vector with all entries equal to 1.*

.

**Corollary 5.6.** *If $A \in \mathsf{F}(t)\left[\mathcal{D};\delta\right]^{n\times n}$* is a unimodular matrix then the row reduced form of $A$ is an identity matrix.

*Proof.* We prove the claim by contradiction. Let $T$ be the row-reduced form of $A$ where $A$ is a unimodular matrix. Assume, to arrive at a contradiction, that $T$ is not the identity matrix. By Fact 5.5, we know that there exists a unimodular matrix $U$ such that $T = UA$. Since $T$ is row-reduced, there exists no $\overrightarrow{v} \in \mathrm{F}(t)^{1\times n}$ with $\overrightarrow{v} \neq \overrightarrow{0}$ such that $\overrightarrow{v}L = 0$ where $L = LC_{Row}(T)$. Since $T$ is a unimodular matrix, there exist a unimodular matrix $S$ such that $I_n = ST$. Let $m := \max(\deg(s_{ij}t_{jk}))$ for $1 \leq i, j, k \leq n$ and without loss of generality we can say that $\deg(s_{ij}t_{jk})$ becomes the maximum, $m$, when $i = l_1, j = l_2$, and $k = l_3$. Then we construct a row vector $\overrightarrow{v} \in \mathsf{F}(t)^{1\times n}$ in such a way that for each $j = 1, \dots n$, if $m = \max(\deg(s_{l_1 j}t_{jk}))$ for $1 \leq k \leq n$ then $v_j := lc(s_{l_1 j})$. Otherwise, $v_j := 0$. Since $I_n = ST$, $\overrightarrow{v}L = 0$. Since $T$ is in row-reduced form, $\overrightarrow{v}$ should be a zero vector. However, based on our construction of $\overrightarrow{v}$, the row vector $\overrightarrow{v}$ should have at least one non-zero entry. By contradiction, $T$ is the identity matrix.

□

The following theorems provide degree bounds on $H$ and $U$. We first compute a degree bound of the inverse of $U$ by using the properties of the Hermite form and the idea of backward substitution, and then use the result of Beckermann et al. [3] to compute degree bound of $U$.

**Theorem 5.7.** *Let $A \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ be a matrix with $\deg_{\mathcal{D}}A_{ij} \leq d$ and full row rank. Suppose $UA = H$ for unimodular matrix $U \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ and $H \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ in Hermite form. Then there exists a unimodular matrix $V \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times n}$ such that $A = VH$ where $UV = I_n$ and $\deg_{\mathcal{D}}V_{ij} \leq d$.*

*Proof.* We prove by induction on $n$. The base case is $n = 1$. Since $H_{11} = \mathrm{GCRD}(A_{11}, \dots, A_{n1})$, $\deg_{\mathcal{D}}H_{11} \leq d$ and so $\deg_{\mathcal{D}}V_{i1} \leq d$ for $1 \leq i \leq n$. Now, we suppose that our claim is true for $k$ where $1 < k < n$. Then we have to show that $\deg_{\mathcal{D}}V_{ik+1} \leq d$. We need to consider two cases:
Case 1: $\deg_{\mathcal{D}}V_{i,k+1} > \max(\deg_{\mathcal{D}}V_{i1}, \dots, \deg_{\mathcal{D}}V_{ik})$. Since

$$\deg_{\mathcal{D}}H_{k+1,k+1} \geq \max(\deg_{\mathcal{D}}\ H_{1,k+1}, \dots, \deg_{\mathcal{D}}H_{k,k+1}),$$
$$\deg_{\mathcal{D}}A_{i,k+1} = \deg_{\mathcal{D}}(V_{i,k+1}H_{k+1,k+1}),$$

where $A_{i,k+1} = V_{i1}H_{1,k+1} + \cdots + V_{i,k+1}H_{k+1,k+1}$. Thus, $\deg_{\mathcal{D}}V_{i,k+1} \leq d$.
Case 2: $\deg_{\mathcal{D}}V_{i,k+1} \leq \max(\deg_{\mathcal{D}}V_{i1}, \dots, \deg_{\mathcal{D}}V_{ik})$. Thus, by induction hypothesis, $\deg_{\mathcal{D}}V_{i,k+1} \leq d$. □

**Corollary 5.8.** *Let $A$, $V$, and $U$ be those in Theorem 5.7. Then $\deg_{\mathcal{D}}U_{ij} \leq (n-1)d$.*

*Proof.* By Corollary 5.6, we know that the row reduced form of $V$ is $I_n$. Moreover, since $I_n = UV$, we can compute the degree bound of $U$ by using Fact 5.5. Clearly,

$$\overrightarrow{v} + (|\overrightarrow{u}| - |\overrightarrow{v}| - \min_j\{u_j\})\overrightarrow{e} \leq \overrightarrow{v} + (|\overrightarrow{u}| - \min_j\{u_j\})\overrightarrow{e},$$

where $\overrightarrow{u} := \max(\overrightarrow{0}, rowdegV)$ and $\overrightarrow{v} := \max(\overrightarrow{0}, rowdegI_n) = \overrightarrow{0}$. Since the degree of each row of $V$ is bounded by $d$, $(|\overrightarrow{u}| - \min_j\{u_j\}) \leq (n-1)d$. Then, by Fact 5.5, $rowdegU \leq (n-1)d$. Therefore, $\deg_{\mathcal{D}} U_{ij} \leq (n-1)d$. □

Thus, we can compute a degree bound of $H$.

**Corollary 5.9.** *Let $H$ be same as that in Theorem 5.7. Then $\deg_{\mathcal{D}} H_{ij} \leq nd$.*

*Proof.* Since $\deg_{\mathcal{D}} U_{ij} \leq (n-1)d$ and $\deg_{\mathcal{D}} A_{ij} \leq d$, $\deg_{\mathcal{D}} H_{ij} \leq nd$. □

## 5.4 The linear system method for Hermite form

In this section we present our polynomial-time algorithm for the Hermite form, which is a variant of the linear system method developed in Kaltofen et al. [10] and Storjohann [26].

**Theorem 5.10.** *Let $A \in \mathsf{F}[t][\mathcal{D};\delta]^{n\times n}$ have full row rank, with $\deg_{\mathcal{D}} A_{i,j} \leq d$, and $(d_1, \ldots, d_n) \in \mathbb{N}^n$ be given. Consider the system of equations $PA = G$, for $n \times n$ matrices for $P, G \in \mathsf{F}(t)[\mathcal{D};\delta]$ restricted as follows:*

- *The degree (in $\mathcal{D}$) of each entry of $P$ is bounded by $(n-1)d + \max_{1\leq i\leq n} d_i$.*
- *The matrix $G$ is upper triangular, where every diagonal entry is monic and the degree of each off-diagonal entry is less than the degree of the diagonal entry below it.*
- *The degree of the ith diagonal entry of $G$ is $d_i$.*

*Let $H$ be the Hermite form of $A$ and $(h_1, \ldots, h_n) \in \mathbb{N}^n$ be the degrees of the diagonal entries of $H$. Then the following are true:*

(a) *There exists at least one pair* $P, G$ *as above with* $PA = G$ *if and only if* $d_i \geq h_i$ *for* $1 \leq i \leq n$.

(b) *If* $d_i = h_i$ *for* $1 \leq i \leq n$ *then* $G$ *is the Hermite form of* $A$ *and* $P$ *is a unimodular matrix.*

*Proof.* The proof is similar to that of [10], Lemma 2.1. Given a degree vector $(d_1, \ldots, d_n)$, we view $PA = G$ as a system of equations in the unknown entries of $P$ and $G$. Since $H$ is the Hermite form of $A$, there exist a unimodular matrix $U$ such that $UA = H$. Thus $PU^{-1}H = G$ and the matrix $PU^{-1}$ must be upper triangular since the matrices $H$ and $G$ are upper triangular. Moreover, since the matrix $PU^{-1}$ is in $\mathsf{F}(t)[\mathcal{D}; \delta]^{n \times n}$, and $G_{ii} = (PU^{-1})_{ii} \cdot H_{ii}$ for $1 \leq i \leq n$, we know $d_i \geq h_i$ for $1 \leq i \leq n$. For the other direction, we suppose $d_i \geq h_i$ for $1 \leq i \leq n$. Let $D = diag(\mathcal{D}^{d_1 - h_1}, \ldots, \mathcal{D}^{d_n - h_n})$. Then since $(DU)A = (DH)$, we can set $P = DU$ and $G = DH$ as a solution to $PA = G$, and the $i$th diagonal of $G$ has degree $d_i$ by construction. By Corollary 5.8, we know $\deg_{\mathcal{D}} U_{i,j} \leq (n-1)d$ and so $\deg_{\mathcal{D}} P_{i,j} \leq (n-1)d + \max_{1 \leq i \leq n} d_i$. To prove (b), suppose $d_i = h_i$ for $1 \leq i \leq n$ and that, contrarily, $G$ is *not* the Hermite form of $A$. Since $PU^{-1}$ is an upper triangular matrix with ones on the diagonal, $PU^{-1}$ is a unimodular matrix. Thus $P$ is a unimodular matrix and, by Corollary 5.3, $G$ *is* the (unique) Hermite form of $A$, a contradiction.

□

**Lemma 5.11.** Let $A$, $P$, $(d_1, \ldots, d_n)$, and $G$ be as in Theorem 5.10, and $\beta := (n-1)d + \max_{1 \leq i \leq n} d_i$. Also, assume that $\deg_t A_{ij} \leq e$ for $1 \leq i, j \leq n$. Then we can express the system $PA = G$ as a linear system over $\mathsf{F}(t)$ as $\widehat{P}\widehat{A} = \widehat{G}$ where:

$$\widehat{P} \in \mathsf{F}(t)^{n \times n(\beta+1)}, \quad \widehat{A} \in \mathsf{F}[t]^{n(\beta+1) \times n(\beta+d+1)}, \quad \widehat{G} \in \mathsf{F}(t)^{n \times n(\beta+d+1)}.$$

Assuming the entries $\widehat{A}$ are known while the entries of $\widehat{P}$ and $\widehat{G}$ are indeterminates, the system of equations from $\widehat{P}\widehat{A} = \widehat{G}$ for the entries of $\widehat{P}$ and $\widehat{G}$ is linear over $\mathsf{F}(t)$ in its unknowns, and the number of equations and unknowns is $O(n^3 d)$. The entries in $\widehat{A}$ are in $\mathsf{F}[t]$ and have degree at most $e$.

*Proof.* Since $\deg_{\mathcal{D}} P_{i,j} \leq \beta$, each entry of $P$ has at most $(\beta+1)$ coefficients in $\mathsf{F}(t)$ and can be written as $P_{ij} = \sum_{0 \leq k \leq \beta} P_{ijk}\mathcal{D}^k$. We let $\widehat{P} \in \mathsf{F}(t)^{n \times n(\beta+1)}$ be the matrix formed from $P$ with $P_{ij}$ replaced by the vector $(P_{ij\beta}, \ldots, P_{ij0}) \in \mathsf{F}(t)$. Since $\deg_{\mathcal{D}} P \leq \beta$, when forming $PA$, the entries in $A$ are multiplied by $\mathcal{D}^\ell$ for $0 \leq \ell \leq \beta$, resulting in polynomials of degree in $\mathcal{D}$ of degree at most $\mu = \beta + d$. Thus, we construct $\widehat{A}$ as the matrix formed from $A$ with $A_{ij}$ replaced by the $(\beta+1) \times (\mu+1)$ matrix whose $\ell$-th row is

$$(A^{[\ell]}_{ij\mu}, A^{[\ell]}_{ij(\mu-1)}, \ldots, A^{[\ell]}_{ij0}) \text{ such that } \mathcal{D}^\ell A_{ij} = A^{[\ell]}_{ij\mu}\mathcal{D}^\mu + A^{[\ell]}_{ij(\mu-1)}\mathcal{D}^{(\mu-1)} + \cdots + A^{[\ell]}_{ij0}.$$

Note that by Lemma 3.1 we can compute $\mathcal{D}^\ell A_{i,j}$ quickly. Finally, we construct the matrix $\widehat{G}$. Each entry of $G$ has degree in $\mathcal{D}$ of degree at most $nd \leq n(\beta+d+1)$. Thus, initially $\widehat{G}$ is the matrix formed by $G$ with $G_{ij}$ replaced by

$$(G_{ij\mu}, \ldots, G_{ij0}) \quad \text{where} \quad G_{ij} = G_{ij\mu}\mathcal{D}^\mu + G_{ij(\mu-1)}\mathcal{D}^{(\mu-1)} + \cdots + G_{ij0}.$$

However, because of the structure of the system we can fix values of many of the entries of $\widehat{G}$ as follows. First, since every diagonal entry of the Hermite form is monic, we know the corresponding entry in $\widehat{G}$ is 1. Also, by the vector $(d_1, \ldots, d_n)$, the degree in $\mathcal{D}$ of the $i$-th diagonal entry of $G$ is bounded by $d_i$, and every off-diagonal has degree in $\mathcal{D}$ less than that of the $i$-th diagonal below it (and hence less than $d_i$), and we can set all coefficients of larger powers of $\mathcal{D}$ to 0 in $\widehat{G}$. The resulting system $\widehat{P}\widehat{A} = \widehat{G}$, restricted as above according to Theorem 5.10, has $O(n^3d)$ linear equations in $O(n^3d)$ unknowns. Since the coefficients in $\widehat{A}$ are all of the form $\mathcal{D}^\ell A_{ij}$, and since this does not affect their degree in $t$, the degree in $t$ of entries of $\widehat{A}$ is the same as that of $A$, namely $e$.

□

In [26], Storjohann reduces the size of the system from $O(n^3d) \times O(n^3d)$ to $O(n^2d) \times O(n^2d)$ essentially by using the fact that we do not need all unknown entries of $\widehat{G}$ in order to find all unknown entries of $\widehat{P}$ and constructing a system of equations accordingly. The following example shows how the size of the system can be reduced when removing all unknown entries of $\widehat{G}$.

**Example 5.12.** Let

$$A = \begin{bmatrix} 2t\mathcal{D} & 0 & t + (1+4t)\mathcal{D} \\ 2t^2 & 2t & 2t + 4t^2 \\ 2t + 2t\mathcal{D} & 4t + 2t\mathcal{D} & 9t - t^2 + (1+5t)\mathcal{D} \end{bmatrix}$$

For simplicity, suppose that we know the correct degrees of the diagonal entries of the Hermite form of $A$, which are $(0,0,1)$. Also, note that $\deg_{\mathcal{D}} A = 1$, $n = 3$, and $\beta = 3$ since $\beta = (n-1)d + d_{max}$. Now, we construct $\widehat{A}$, $\widehat{G}$, and $\widehat{P}$ based on Lemma 5.10. $\widehat{A} =$

$$\left[\begin{array}{ccccccccccccccc}
2t & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t & 12+t & 3 & 0 & 0 \\
0 & 2t & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t & 8+t & 2 & 0 \\
0 & 0 & 2t & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t & 4+t & 1 \\
0 & 0 & 0 & 2t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t & t \\
0 & 2t^2 & 12t & 12 & 0 & 0 & 2t & 6 & 0 & 0 & 0 & 2t+4t^2 & 6+24t & 24 & 0 \\
0 & 0 & 2t^2 & 8t & 4 & 0 & 0 & 2t & 4 & 0 & 0 & 0 & 2t+4t^2 & 4+16t & 8 \\
0 & 0 & 0 & 2t^2 & 4t & 0 & 0 & 0 & 2t & 2 & 0 & 0 & 0 & 2t+4t^2 & 2+8t \\
0 & 0 & 0 & 0 & 2t^2 & 0 & 0 & 0 & 0 & 2t & 0 & 0 & 0 & 0 & 2t+4t^2 \\
2t & 6+2t & 6 & 0 & 0 & 2t & 6+4t & 12 & 0 & 0 & 1+5t & 15+9t-t^2 & 27-6t & -6 & 0 \\
0 & 2t & 4+2t & 4 & 0 & 0 & 2t & 4+4t & 8 & 0 & 0 & 1+5t & 10+9t-t^2 & 18-4t & -2 \\
0 & 0 & 2t & 2+2t & 2 & 0 & 0 & 2t & 2+4t & 4 & 0 & 0 & 1+5t & 5+9t-t^2 & 9-2t \\
0 & 0 & 0 & 2t & 2t & 0 & 0 & 0 & 2t & 4t & 0 & 0 & 0 & 1+5t & 9t-t^2
\end{array}\right],$$

$$\widehat{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \widehat{G}_{130} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \widehat{G}_{230} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \widehat{G}_{330} \end{bmatrix},$$

and

$$\widehat{P} = \begin{bmatrix} \widehat{P}_{113} & \widehat{P}_{112} & \widehat{P}_{111} & \widehat{P}_{110} & \widehat{P}_{123} & \widehat{P}_{122} & \widehat{P}_{121} & \widehat{P}_{120} & \widehat{P}_{133} & \widehat{P}_{132} & \widehat{P}_{131} & \widehat{P}_{130} \\ \widehat{P}_{213} & \widehat{P}_{212} & \widehat{P}_{211} & \widehat{P}_{210} & \widehat{P}_{223} & \widehat{P}_{222} & \widehat{P}_{221} & \widehat{P}_{220} & \widehat{P}_{233} & \widehat{P}_{232} & \widehat{P}_{231} & \widehat{P}_{230} \\ \widehat{P}_{313} & \widehat{P}_{312} & \widehat{P}_{311} & \widehat{P}_{310} & \widehat{P}_{323} & \widehat{P}_{322} & \widehat{P}_{321} & \widehat{P}_{320} & \widehat{P}_{333} & \widehat{P}_{332} & \widehat{P}_{331} & \widehat{P}_{330} \end{bmatrix}.$$

If we remove unknown entries from $\widehat{G}$ then we have two new matrices $\widehat{G}^*$ and $\widehat{P}^*$ as follows.

$$\widehat{G}^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$\widehat{A}^* = \begin{bmatrix} 2t & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t & 12+t & 3 & 0 \\ 0 & 2t & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t & 8+t & 2 \\ 0 & 0 & 2t & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t & 4+t \\ 0 & 0 & 0 & 2t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1+4t \\ 0 & 2t^2 & 12t & 12 & 0 & 0 & 2t & 6 & 0 & 0 & 0 & 2t+4t^2 & 6+24t & 24 \\ 0 & 0 & 2t^2 & 8t & 4 & 0 & 0 & 2t & 4 & 0 & 0 & 0 & 2t+4t^2 & 4+16t \\ 0 & 0 & 0 & 2t^2 & 4t & 0 & 0 & 0 & 2t & 2 & 0 & 0 & 0 & 2t+4t^2 \\ 0 & 0 & 0 & 0 & 2t^2 & 0 & 0 & 0 & 0 & 2t & 0 & 0 & 0 & 0 \\ 2t & 6+2t & 6 & 0 & 0 & 2t & 6+4t & 12 & 0 & 0 & 1+5t & 15+9t-t^2 & 27-6t & -6 \\ 0 & 2t & 4+2t & 4 & 0 & 0 & 2t & 4+4t & 8 & 0 & 0 & 1+5t & 10+9t-t^2 & 18-4t \\ 0 & 0 & 2t & 2+2t & 2 & 0 & 0 & 2t & 2+4t & 4 & 0 & 0 & 1+5t & 5+9t-t^2 \\ 0 & 0 & 0 & 2t & 2t & 0 & 0 & 0 & 2t & 4t & 0 & 0 & 0 & 1+5t \end{bmatrix}.$$

We can construct the system of equations, $B\overrightarrow{x} = \overrightarrow{y}$, from $\widehat{P}$, $\widehat{A}$, and $\widehat{G}$ and then solving for $\overrightarrow{x}$ gives the correct values of all unknown entries in $\widehat{P}$ and $\widehat{G}$.

Now we present the polynomial-time algorithm using the linear system method for computing the Hermite form of matrices over $\mathrm{F}(t)\,[\mathcal{D};\delta]$.

**Algorithm 6** Linear System Method for Hermite form computation

```
Require: $A$ is a square matrix with full rank
procedure HERMITE($A$)
    $n \leftarrow$ row-dimension($A$)
    $d \leftarrow \deg_{\mathcal{D}} A$
    $di[1, \ldots, n] \leftarrow \{nd, \ldots, nd\}$  ▷ Initialize each entry of the vector $di$ with $nd$
    for $i \leftarrow 1, n$ do
        $start \leftarrow 0$
        $end \leftarrow nd$
        repeat
            $diff \leftarrow start + \left\lfloor \frac{start - end}{2} \right\rfloor$
            $di[i] \leftarrow start + diff$
            $\beta \leftarrow (n-1)d + \max_{1 \leq i \leq n} di[i]$
            By Lemma 5.11 construct the system of equations, $\widehat{P}\widehat{A} = \widehat{G}$
            if $\widehat{P}\widehat{A} = \widehat{G}$ is consistent then
                $end \leftarrow di[i]$
            else
                if $diff$=0 then
                    $start \leftarrow end$
                    $di[i] \leftarrow end$
                else
                    $start \leftarrow di[i]$
                end if
            end if
        until $start > end$
    end for
    Construct $U, H$ from entries of $\widehat{P}, \widehat{G}$ respectively
    return $(U, H)$                                         ▷ $H = UA$
end procedure
```

So far, we have shown how to convert the differential system over $\mathsf{F}(t)\,[\mathcal{D};\delta]$ into a linear system over $\mathsf{F}(t)$. Also, we note, by Theorem 5.10, that the correct degree of the $i$-th diagonal entry in the Hermite form of $A$ can be founded by seeking the smallest non-negative integer $k$ such that $PA = G$ is consistent when $\deg_{\mathcal{D}} G_{j,j} = nd$ for $j = 1, \ldots, i-1, i+1, \ldots, n$ and $k \leq \deg_{\mathcal{D}} G_{i,i}$. In Algorithm 6, we employ the idea of the binary search which guarantees that we can find the correct degrees of all diagonal entries by solving at most $O(n \log(nd))$ systems. We then find the correct degrees of the diagonal entries in the Hermite form of $A$, solving the system $PA = G$ with the correct diagonal degrees gives the matrices $U$ and $H$ such that $UA = H$ where $H$ is the Hermite form of $A$.

**Theorem 5.13.** *Let $A \in \mathsf{F}[t][\mathcal{D};\delta]^{n\times n}$ with $\deg_{\mathcal{D}} A_{ij} \leq d$ and $\deg_t A_{ij} \leq e$ for $1 \leq i,j \leq n$. Then we can compute the Hermite form $H \in \mathsf{F}(t)[\mathcal{D};\delta]$ of $A$, and a unimodular $U \in \mathsf{F}[t][\mathcal{D};\delta]$ such that $UA = H$, with $O((n^{10}d^3 + n^7d^2e)\log(nd))$ operations in* $\mathsf{F}$

*Proof.* Lemma 5.11 and the following discussion, above shows that computing $U$ and $H$ is reduced to solving $O(n\log(nd))$ systems of linear equations over $\mathsf{F}(t)$, each of which is $m \times m$ for $m = O(n^3 d)$ and in which the entries have degree $e$. Using standard linear algebra this can be solved with $O(m^4 e)$ operations in $\mathsf{F}$, since any solution has degree at most $me$ (see [31]). A somewhat better strategy is to use the $t$-adic lifting approach of Dixon [9], which would require $O(m^3 + m^2 e)$ operations in $\mathsf{F}$ for each system, giving a total cost of $O((n^{10}d^3 + n^7d^2e)\log(nd))$ operations in $\mathsf{F}$.

□

It is often the case that we are considering differential systems over $\mathbb{Q}[t][\mathcal{D};\delta]$ where we must contend with growth in coefficients in $\mathcal{D}$, $t$, and the size of the rational coefficients. The following lemma shows there is some small amount of extra coefficient growth when going from $A$ to $\widehat{A}$.

**Lemma 5.14.** *Let $A$ be same as that in Theorem 5.10 and $||A||$ denote the maximum coefficient length in $\mathbb{Q}$. In addition, we suppose $\deg_t A_{ijk} \leq e$ for $1 \leq i,j \leq$*

*n and* $0 \leq k \leq d$ *where* $A_{ijk} \in \mathbb{Q}[t]$ . *Then we see Size* $(A) = O\left(n^2 de \log ||A||\right)$. *Now, we claim* $\text{Size}\left(\widehat{A}\right) = O\left(n^4 d^2 e \left(\log ||A|| + e \log e\right)\right)$.

*Proof.* Since $\mathcal{D}A_{ijk} = A_{ijk}\mathcal{D} + (A_{ijk})'$, $\deg_t \mathcal{D}A_{ijk} \leq \deg_t A_{ijk}$. Then it follows $\deg_t \widehat{A}_{ij} \leq e$. However, when multiplying $A_{ijk}$ by $\mathcal{D}$ on the left side, there is a change on the size of coefficients in $\mathbb{Q}$. For example, $||\left(A_{ijk}\right)'|| \leq e||A_{ijk}||$. Moreover, we note that the $(e+1)$-th derivative of $A_{ijk}$ is equal to zero because each derivation decreases the degree of $A_{ijk}$ by 1. So, we conclude $||\left(A_{ijk}\right)^{[l]}|| \leq e!||A_{ijk}||$ for any $l \geq 0$ where $\left(A_{k,i,j}\right)^{[l]}$ denotes the $l$-th derivative of $A_{ijk}$. By Lemma 5.11, we know that the number of entries in $\widehat{A}$ is bounded by $O\left(n^4 d^2\right)$. Also, each entry of $\widehat{A}$ is a polynomial in $t$ with degree at most $e$. As mentioned above, the maximum coefficient length of $\widehat{A}$ in $\mathbb{Q}$ is bounded by $e!||A_{ijk}||$. Thus, $\text{Size}\left(\widehat{A}\right) = O\left(n^4 d^2 e \left(\log ||A|| + e \log e\right)\right)$.

□

Thus, it follows that we can find the Hermite form of $A \in \mathbb{Q}(t)[\mathcal{D};\delta]^{n \times n}$ in time polynomial in $n$, $\deg_t A_{ij}$, $\deg_{\mathcal{D}} A_{ij}$, and $\log \|A_{ij}\|$, the maximum coefficient length in an entry, for $1 \leq i, j \leq n$.

## 5.5 Generalization to rectangular matrices

In this section we generalize our algorithm for rectangular matrices. Suppose $A \in \mathsf{F}[t][\mathcal{D};\delta]^{n \times m}$, $n > m$, and that the rank of $A$ is $m$. So, we know that $A$ has $m$ linearly independent rows over $\mathsf{F}[t][\mathcal{D};\delta]$. By Corollary 5.4, we know that $A$ has a unique matrix $H \in \mathsf{F}(t)[\mathcal{D};\delta]^{n \times m}$ with rank $m$ in Hermite form. By Fact 5.5, we know that there exists a unimodular matrix $U_1$ such that

$$T = U_1 A,$$

where $T$ is a row-reduced form of $A$ and has $m$ nonzero rows. After permuting rows of $T$, we can have a new matrix $A'$, which is of the form

$$A' = \begin{pmatrix} T' \\ 0 \end{pmatrix},$$

where $A' \in \mathsf{F}(t)[\mathcal{D};\delta]^{n\times m}$ and $T' \in \mathsf{F}(t)[\mathcal{D};\delta]^{m\times m}$ consists of $m$ nonzero rows of $T$. Without loss of generality, we can say that

$$A' = U_1 A.$$

Beckermann et al. [3] introduce a polynomial time algorithm for finding such matrices $A'$ and $U_1$ by performing row reduction of a matrix of Ore polynomials in a fraction-free way. Since $T'$ consists of $m$ linearly independent rows, by applying the linear system method, we can compute $U_2$ and $H$ for $T'$ such that $U_2T' = H$ where $H$ is the Hermite form of $T'$. Also, we note that by Corollary 5.4, $H$ is the Hermite form of $A$ as well where a correspondent unimodular matrix is given by computing $U = U_2U_1$. Now, we have the following:

$$H = UA,$$

where $H$ is the Hermite form of $A$. Thus, we have the following lemma.

**Lemma 5.15.** *Let $A \in \mathsf{F}[t][\mathcal{D};\delta]^{n\times m}$ with full rank. We can find the unique Hermite form of $A$ in a polynomial time.*

## 5.6 Generalization to matrices over $\mathsf{F}(t)[\mathcal{E};\sigma]$

So far we have seen how to convert the differential system over $\mathsf{F}(t)\,[\mathcal{D};\delta]$ into a linear system over $\mathsf{F}(t)$. In order to generalize our approach to matrices over $\mathsf{F}(t)\,[\mathcal{E};\sigma]$, we also need to covert the polynomial system over $\mathsf{F}(t)[\mathcal{E};\sigma]$ into a linear system over $\mathsf{F}(t)$. Since we only use the property of the Hermite form and the idea of backward substitution when computing degree bounds on $U$ and $H$ in $\mathsf{F}(t)\,[\mathcal{D};\delta]$, we can still have same degree bounds on $U$ and $H$ in $\mathsf{F}(t)[\mathcal{E};\sigma]$. Thus, the only thing to be changed is that we have to consider not $\delta$ but $\sigma$ when constructing $\widehat{P}$, $\widehat{A}$, and $\widehat{G}$. Let

$$f = \sum_{i=0}^{n} f_i\mathcal{E}^i \in \mathsf{F}[t][\mathcal{E};\sigma] \text{ for } f_0,\ldots,f_n \in \mathsf{F}[t].$$

Then we notice that

$$\mathcal{E}f = \sum_{i=0}^{n} f_i(t+1)\mathcal{E}^{i+1}.$$

Assume $\deg_t f \leq e$ and then the cost of computing $\mathcal{E}f$ is $O(ne^2)$ operations in $\mathsf{F}$. Thus the cost of computing $\mathcal{E}^m f$ requires $O(mne^2)$ operations in $\mathsf{F}$. We have the following lemma.

**Lemma 5.16.** *Let $f \in \mathsf{F}(t)[\mathcal{E};\sigma]$ have $\deg_{\mathcal{E}} f = n$,$\deg_t f = e$, and let $m \in \mathbb{N}$. Then we can compute $\mathcal{E}^k f$, for $1 \leq k \leq m$, with $O(mne^2)$ operations in $\mathsf{F}$.*

Thus, by Lemma 5.16 we note that $\widehat{A}$ can be constructed quickly. Moreover, by Theorem 5.13 we can solve the system $\widehat{P}\widehat{A} = \widehat{G}$ with $O((n^9d^3 + n^6d^2e)$ operations in $\mathsf{F}$. Therefore, we have the following lemma.

**Lemma 5.17.** *Let $A \in \mathsf{F}[t][\mathcal{E};\sigma]^{n\times n}$ with $\deg_{\mathcal{E}} A_{ij} \leq d$ and $\deg_t A_{ij} \leq e$ for $1 \leq i,j \leq n$. Then we can compute the Hermite form $H \in \mathsf{F}[t][\mathcal{E};\sigma]$ of $A$, and a unimodular $U \in \mathsf{F}[t][\mathcal{E};\sigma]$ such that $UA = H$, with $O((n^{10}d^3 + n^7d^2e)\log(nd))$ operations in $\mathsf{F}$*

# Chapter 6

# Experimental Results

We have implemented both the linear system method and the naive method for computing the Hermite form in Maple, for $\mathsf{F} = \mathbb{Q}$. The experiments were performed on an AMD Opteron 850 CPU at 2.4 GHz with 8GB of RAM. We randomly generated matrices over $\mathsf{F}[t][\mathcal{D};\delta]$ and then calculated the elapsed time for computing the Hermite forms of input matrices. However, with the degree bound $(n-1)d+\max_{1\leq i\leq n} d_i$ for $P$, the linear system method did not surpass the naive method. Also, note that the GCRD and LCLM computations in Maple are optimized using the modular approach of Li and Nemes [19]. Thus, we make a conjecture about the degree bound of $P$.

**Conjecture 6.1.** *Let $P$, $A$, and $G$ be those in Theorem 5.10. Then the degree of $P$ is at most $(n-1)d$.*

By using the idea of Storjohann [26] where the unknown coefficients of entries in $G$ are removed, we reduce the size of the system to $O(n^2d) \times O(n^2d)$. Based on a new degree bound of $P$ and the idea of Storjohann, the size of the system is reduced by more than a factor of $n$. All experiments resulted in the correct Hermite forms for all input matrices. The following table shows how well the reduced linear system outperformed the naive method. Note that the last two columns denote the experiments for the linear system method where the sizes of the systems are $O(n^3d) \times O(n^3d)$ and $O(n^2d) \times O(n^2d)$ respectively.

Let $A$ be a matrix in $\mathsf{F}[t][\mathcal{D};\delta]$.

| Dimension($A$) | $\deg_{\mathcal{D}} A$ | $\deg_t A$ | Naive Method | Original system | Reduced system |
|---|---|---|---|---|---|
| 3 | 1 | 1 | 0.425 sec | 2.352 sec | 0.412 sec |
| 3 | 1 | 2 | 0.903 sec | 5.869 sec | 0.734 sec |
| 3 | 2 | 1 | 21.870 sec | 37.744 sec | 2.768 sec |
| 3 | 2 | 2 | 157.160 sec | 326.066 sec | 6.068 sec |
| 4 | 1 | 1 | 11.263 sec | 42.086 sec | 4.460 sec |
| 4 | 1 | 2 | 130.736 sec | 344.217 sec | 9.868 sec |
| 4 | 2 | 1 | $>$ 20 min | $>$ 20 min | 114.159 sec |
| 4 | 2 | 2 | $>$ 20 min | $>$ 20 min | 557.791 sec |
| 5 | 1 | 1 | $>$ 20 min | $>$ 20 min | 101.541 sec |
| 5 | 1 | 2 | $>$ 20 min | $>$ 20 min | 300.531 sec |
| 5 | 2 | 1 | $>$ 20 min | $>$ 20 min | $>$ 20 min |
| 5 | 2 | 2 | $>$ 20 min | $>$ 20 min | $>$ 20 min |

As shown above, when the size of the system is reduced to $O(n^2d) \times O(n^2d)$, the linear system method outperformed the naive method as increasing the dimension of $A$, $\deg_{\mathcal{D}} A$, and $\deg_t A$.

# Chapter 7

# Conclusion and Future Work

We have studied the problem of computing the Hermite form of a matrix over the differential polynomial ring. We first compute the upper bound of the size of coefficients when computing the GCRD and LCLM of two differential polynomials. Then we give the NAIVE algorithm using the GCRD and LCLM computations in order to compute the Hermite form of a matrix of differential polynomials. By using the properties of the Hermite form and the idea of backward substitution, we are able to obtain degree bounds on $U$ and $H$ where $H$ is the Hermite form of $A \in \mathsf{F}(t)\,[\mathcal{D};\delta]^{n\times n}$. Based on degree bounds on $U$ and $H$, we can also apply the linear system method for the Hermite form computation in the differential polynomial ring. In the method, we convert the differential system into a linear system in a commutative ring, which can be accomplished in a polynomial time in terms of the matrix size, the degree, and the size of coefficients. Then we generalize our algorithm for a rectangular matrix and a matrix of shift polynomials. However, from a practical point of view our method is still expensive. Thus, we have the following future research directions.

- *Reduce the degree bound of $P$ in Theorem 5.10.* In Theorem 5.10, we show the degree of $P$ is bounded by $\beta$, $\beta = (n-1)d + \max\limits_{1\le i\le n} d_i$. If it is possible to show that we can set $\beta = (n-1)d$ then we can reduce the size of the system and have a better running time. One possible solution is to use the non-commutative determinant of Dieudonné because it has a number of the desirable properties of the usual determinant.

- *Use a randomized approach to find the correct degrees of the diagonal entries.* By using the idea of binary search, we have to solve $O(n \log(nd))$ systems for finding the correct degrees of the diagonal entries. Since in general randomization gives a better expected cost, randomization approach can reduce the number of systems to be solved. For example, in general the degrees of the diagonal entries in Hermite form increase as computing the Hermite form. Also, we can employ structured matrix techniques because we solve a similar system repeatedly.

- *Compute degree bounds on $U$, $V$, and $S$ such that $UAV = S$ where $S$ is the Smith form of $A \in \mathsf{F}(t)\left[\mathcal{D};\delta\right]^{n\times n}$.* Computing the Smith form of a matrix over $\mathsf{F}(t)\left[\mathcal{D};\delta\right]$ in a polynomial time is still open problem. Even though we propose the polynomial-time algorithm for the Hermite form computation in this thesis, the Smith form computation is totally different problem because the differential polynomial ring is two-sided ideal domain. However, we believe it is possible to develop a polynomial-time algorithm for the Smith form computation. This should be fairly straightforward in the differential case (where the Smith form is always trivial) but more difficult, and interesting, in the shift case.

# Bibliography

[1] S. Abramov and M. Bronstein. On solutions of linear functional systems. In *Proc. ACM International Symposium on Symbolic and Algebraic Computation (ISSAC)*, pages 1–7, 2001. 32

[2] Nicolas Le Roux Alin Bostan, Frederic Chyzak. Products of ordinary differential operators by evaluation and interpolation. In *International Symposium on Symbolic and Algebraic Computation* , page 23 30, 2008. 14

[3] B. Beckermann, H. Cheng, and G. Labahn. Fraction-free row reduction of matrices of Ore polynomials. *Journal of Symbolic Computation* , 41(1):513–543, 2006. 10, 32, 38, 39, 48

[4] M. Bronstein and M. Petkovšek. On Ore rings, linear operators and factorisation. *Programmirovanie*, 20:27–45, 1994. 13

[5] M. Bronstein and M. Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science* , 157(1):3–33, 1996. 22

[6] M. Chardin. Differential resultants and subresultants. In *Proceedings of Fundamentals of Computation Theory* , pages 180–189. Lecture Notes in Computer Science 529, 1991. 5, 19, 21

[7] H. Cheng. *Algorithms for Normal Forms for Matrices of Polynomials and Ore Polynomials*. PhD thesis, University of Waterloo, 2003. 32

[8] A. Diaz and E. Kaltofen. On computing greatest common divisors with polynomials given by black boxes for their evaluations. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation* , pages 232–239, 1995. 26

[9] J.D. Dixon. Exact solution of linear equations using $p$-adic expansions. *Numer. Math.*, 40:137–141, 1982. 46

[10] M. S. Krishnamoorthy E. Kaltofen and B. D. Saunders. Fast parallel computation of Hermite and Smith forms of polynomial matrices. *SIAM J. Algebraic and Discrete Methods*, 8:683–690, 1987. 11, 31, 32, 38, 40, 41

[11] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms For Computer Algebra.* Kluwer Academic Publishers, 1992. 29

[12] M. Giesbrecht and M. S. Kim. On computing the Hermite form of a matrix of differential polynomials. In *Computer Algebra in Scientific Computing*, 2009. 5

[13] D. Yu. Grigor'ev. Complexity of factoring and calculating the GCD of linear ordinary differential operators. *Journal of Symbolic Computation,*, 10:7–37, 1990. 25

[14] C. Hermite. Sur l'introduction des variables continues dans la théorie des nombres. *J. Reine Angew. Math.*, 41:191–216, 1851. 1, 11

[15] R. Kannan. Polynomial-time algorithms for solving systems of linear equations over polynomials. *Theoretical Computer Science*, 39:69–88, 1985. 11, 32

[16] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979. 2, 11, 33

[17] Z. Li. *A Subresultant Theory for Linear Differential, Linear Difference and Ore Polynomials, with Applications*. PhD thesis, RISC-Linz, Johannes Kepler University, Linz, Austria, 1996. 21

[18] Z. Li. A subresultant theory for Ore polynomials with applications. In *Proc. International Symposium on Symbolic and Algebraic Computation* , pages 132–139, 1998. 13, 19

[19] Z. Li and I. Nemes. A modular algorithm for computing greatest common right divisors of Ore polynomials. In *Proc. International Symposium on Symbolic and Algebraic Computation*, pages 282–289, 1997. 19, 50

[20] J. Middeke. A polynomial-time algorithm for the Jacobson form for matrices of differential operators. Technical Report 08-13, Research Institute for Symbolic Computation (RISC), Linz, Austria, 2008. 33

[21] T. Mulders and A. Storjohann. On lattice reduction for polynomial matrices. *Journal of Symbolic Computation*, 35(4):377–401, April 2003. 11, 32

[22] O. Ore. Theory of non-commutative polynomials. *Anals of Math*, 34:480–508, 1933. 1, 6, 7

[23] H. Cheng P. Davies and G. Labahn. Computing Popov form of general Ore polynomial matrices. In *Milestones in Computer Algebra* , pages 149–156, 2008. 32

[24] V. Popov. Invariant description of linear, time-invariant controllable systems. *SIAM J. Control*, 10:252–264, 1972. 1

[25] H. J. S. Smith. On systems of linear indeterminate equations and congruences. *Phil. Trans. Roy. Soc. London* , pages 151:293–326, 1861. 1

[26] A. Storjohann. Computation of Hermite and Smith normal forms of matrices. Master's thesis, University of Waterloo, 1994. 11, 31, 38, 40, 42, 50

[27] A. Storjohann. *Algorithms for Matrix Canonical Forms* . PhD thesis, Department of Computer Science, Swiss Federal Institute of Technology—ETH, 2000. 11

[28] A. Storjohann. High-order lifting and integrality certification. *Journal of Symbolic Computation,* , 36:613–648, 2003. 24

[29] A. Storjohann and G. Villard. Computing the rank and a small nullspace basis of a polynomial matrix. In *International Symposium on Symbolic and Algebraic Computation*, volume 2005, page 309, 2005. 29

[30] G. Villard. Computing popov and hermite forms of polynomial matrices. In *International Symposium on Symbolic and Algebraic Computation* , 1996. 11, 32

[31] J. von zur Gathen and J Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999. 46