Computing Matrix Canonical Forms of Ore Polynomials

by

Mohamed Khochtali

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, 2017

Abstract

Let $\mathsf{F}[\partial;\sigma,\delta]$ be a ring of Ore polynomials over a field F, where σ is an automorphism of F and δ is a σ-derivation. In this thesis, we present algorithms to compute canonical forms of non-singular input matrix of Ore polynomials while controlling intermediate expression swell. We first give an extension of the algorithm by [Labhalla et al., 1992] to compute the Hermite form of a non-singular input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$. We also give a new fraction-free algorithm to compute the Popov form P of a non-singular $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$, accompanied by an implementation and experimental results that compare it to the best known algorithms in the literature. Our algorithm is output-sensitive, with a cost that depends on the orthogonality defect of the input matrix: the sum of the row degrees in A minus the sum of the row degrees in P. We also use the recent advances in polynomial matrix computations, including fast inversion and rank profile computation, to describe an algorithm that computes the transformation matrix U such that $UA = P$. This algorithm is asymptotically faster than the fraction-free algorithm described above, but it has not been implemented yet.

Acknowledgements

First, I would like to thank my supervisor Arne Storjohann for the countless hours and the unlimited effort he put in to make this possible. Arne was always happy to discuss different ideas and gave a lot of his personal time. I also would like to thank Johann Rosenkilde for all the input and the interesting discussions we had.

Second, I would like to thank George Labahn and Mark Giesbrecht for agreeing to be in the examining committee and for their valuable feedback. I also would like to thank Daniel Roche for always providing me with valuable advices.

Third, I would like to thank all the members of the symbolic computation group for the great environment they provided in the lab. Special thank to Joseph Haraldson as he was the first, after my supervisor, to read and give me valuable feedback on my thesis. I also would like to thank all the friends I acquired here in the past two years. Special thanks to my roommates Adam, Connor and Owen.

Finally I would like to thank all my family members and especially my parents for their support for me in whatever I am facing in life. I also would like to thank my second family in the United States for all the love they have shown me throughout the years.

Table of Contents

List of Tables

Chapter 1

Introduction

Ore polynomial rings, also known as skew polynomial rings, are non-commutative generalisations of univariate polynomial rings, introduced by Oystein Ore [Ore, 1933]. They have a variety of applications, such as modelling recurrence relations and differential equations [Ore, 1933].

Let F be a field, $\sigma : \mathsf{F} \mapsto \mathsf{F}$ be an automorphism of F, and $\delta : \mathsf{F} \mapsto \mathsf{F}$ be a σ-derivation such that

$$\delta(a+b) = \delta(a) + \delta(b)$$

and

$$\delta(ab) = \sigma(a)\delta(b) + \delta(a)b.$$

The Ore ring $\mathsf{F}[\partial;\sigma,\delta]$, also called the skew polynomial ring, is the set of all polynomials in ∂ with coefficients from F. Addition for two Ore polynomials is defined as for the ordinary polynomials from commutative domains. The premultiplication of an element $a \in \mathsf{F}$ by ∂ is defined by the following:

$$\partial a = \sigma(a)\partial + \delta(a).$$

Often $\mathsf{F} = \mathsf{k}(z)$, the field of rational functions in z, where k is a field. For $\mathsf{F} = \mathsf{k}(z)$, we highlight two important Ore rings.

i *Differential polynomials* where $\sigma(f(z)) = f(z)$ and $\delta(f(z)) = f'(z)$ is the usual derivative with respect to z. We follow [Giesbrecht and Kim, 2012] and use $\mathsf{F}[\partial, \prime]$ to denote this Ore ring.

ii The *shift case*, or time-dependence, where $\sigma(f(z)) = f(z+1)$ is the shift automorphism and $\delta = 0$.

One of the motivations to study the differential polynomial ring is that it can be used to model linear differential equations. A matrix of differential Ore polynomials can be used to solve a system of linear differential equation. For example, the following equation

$$(2z+5)\frac{\partial^2 f(z)}{\partial z^2} + 2\frac{\partial f(z)}{\partial z} + 5f(z) = 0$$

is equivalent to

$$\left((2z+5)\frac{\partial^2}{\partial z^2} + 2\frac{\partial}{\partial z} + 5\right) f(z) = 0,$$

and can be modeled by the polynomial

$$(2z+5)\partial^2 + 2\partial + 5 \in \mathbb{Q}(z)[\partial; \prime],$$

with ∂ being the differential operator with respect to z and σ and δ defined as in i. Then a system of differential equation such as the following

$$\begin{aligned}
\frac{\partial f_1(z)}{\partial z} + (2z+5)f_1(z) + 5\frac{\partial^2 f_2(z)}{\partial z^2} + z\frac{\partial f_2(z)}{\partial z} + (z^2-z)f_3(z) &= 0\\
2\frac{\partial^2 f_1(z)}{\partial z^2} + (3z^3+1)f_1(z) + z^2\frac{\partial^3 f_2(z)}{\partial z^3} + (z-2)f_2(z) + (z+1)\frac{\partial f_3(z)}{\partial z} + 3f_3(z) &= 0\\
\frac{\partial f_1(z)}{\partial z} + z^2 f_1(z) + (3z+2)\frac{\partial^2 f_2(z)}{\partial z^2} + 4zf_2(z) + z^2\frac{\partial^2 f_3(z)}{\partial z^2} + (2z^4+3z^3)f_3(z) &= 0
\end{aligned}$$

can be modeled by

$$\begin{bmatrix} \partial + (2z+5) & 5\partial^2 + z\partial & (z^2-z) \\ 2\partial^2 + (3z^3+1) & z^2\partial^3 + (z-2) & (z+1)\partial + 3 \\ \partial + z^2 & (3z+2)\partial^2 + 4z & z^2\partial^2 + (2z^4+3z^3) \end{bmatrix} \begin{bmatrix} f_1(z) \\ f_2(z) \\ f_3(z) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (1.1)$$

Similarly, let us consider $\mathcal{S}$ to be the shift operator such that

$$\mathcal{S}(zf(z)) = (z+1)f(z+1)$$

and

$$\mathcal{S}(fg)(z) = f(z+1)g(z+1).$$

The Fibonacci sequence $1, 1, 2, 3, 5, \cdots = f(1), f(2), f(3), f(4), f(5), \ldots$ satisfies the following recurrence relation (with $\mathcal{S}$ being the shift operator)

$$\mathcal{S}^2 f(z) = \mathcal{S}f(z) + f(z),$$

which can be modeled by the polynomial

$$\mathcal{S}^2 - \mathcal{S} - 1 \in \mathbb{Q}(z)[\mathcal{S}; \sigma, \delta],$$

where σ and δ are defined as in ii.

Another example is the recurrence relation

$$(z^2+1)f(z+3) + 5zf(z+1) + 3z^3 f(z) = 0,$$

which can be modeled by

$$(z^2+1)\mathcal{S}^3 + 5z\mathcal{S} + 3z^3 \in \mathbb{Q}(z)[\mathcal{S}; \sigma, \delta].$$

An Ore ring $\mathsf{F}[\partial; \sigma, \delta]$ is both a left and right Euclidean ring, with division algorithms which essentially work as for usual univariate polynomials. Ore rings also admit unique left skew field of fractions. These facts mean that matrices over Ore rings behave mostly as we are used to: the notions of *rank* and *(non)-singularity* make sense, in particular, and performing row or column operations on a matrix will not change its rank. Further, two matrices $M, M' \in \mathsf{F}[\partial; \sigma, \delta]^{n \times m}$ generate the same left row space if and only if there exists $U \in \mathrm{GL}_n(\mathsf{F}[\partial; \sigma, \delta])$ such that $M = UM'$, where $\mathrm{GL}_n(\mathsf{F}[\partial; \sigma, \delta])$ denotes the set of invertible $n \times n$ matrices over $\mathsf{F}[\partial; \sigma, \delta]$. We refer to [Draxl, 1983] for the basic skew algebra, or

[Beckermann et al., 2006, Giesbrecht and Kim, 2012] for discussions particular to the Ore polynomial case. Also, [M. Bronstein, 2001] gives a well detailed overview of pseudo linear algebra describing skew polynomials properties, difference and differential operators.

Row spaces of matrices over Ore polynomial rings arise in studying coupled systems of such equations. Two matrices A and B generate the same row space if and only if there is unimodular matrix U such that $A = UB$. Computing so called canonical forms of matrices of Ore polynomials allows comparing systems and finding small or otherwise special elements in the space. In this thesis, we focus on computing two canonical forms for a square non-singular input matrix $A \in \mathsf{k}(z)[\partial;\sigma,\delta]^{n\times n}$: the Hermite and the Popov canonical forms which are left unimodular equivalent to the input matrix.

The Hermite canonical form is upper triangular matrix with each diagonal entry monic and with degree in ∂ strictly higher than off-diagonal entries in the same column. For example, in the differential Ore ring $\mathbb{Z}_7(z)[\partial;{}']$, the matrix

$$A = \begin{bmatrix} 1+(5+z)\partial+\partial^2 & & 5+4\partial+4\partial^2 \\ 4+3\partial & 2z & 4\partial+4\partial^2 \\ 3 & & 4+4\partial \end{bmatrix} \in \mathbb{Z}_7(z)[\partial,{}']^{3\times 3} \quad (1.2)$$

is unimodular left equivalent to its Hermite form

$$H = \begin{bmatrix} 1 & & 6+6\partial \\ & 1 & \frac{2}{z}+\frac{2}{z}\partial \\ & & 6+(z+3)\partial+(z+3)\partial^2+\partial^3 \end{bmatrix} \in \mathbb{Z}_7(z)[\partial,{}']^{3\times 3}.$$

As illustrated in the above example, the Hermite form can have degrees larger than the input matrix.

The Popov form is a canonical form of A which has row degrees as small as possible. First consider a row reduced form of A: a row reduced form is not canonical but has row degrees, when sorted, that are lexicographically minimal among all possible basis of the row space of A (a formal definition is given in Section 2.1.) For example,

$$R = \begin{bmatrix} 4 + (z+2)\partial + \partial^2 & & 2 + 4\partial \\ 4 & 2z & \\ 4 & 4z & 4 + 4\partial \end{bmatrix} \in \mathbb{Z}_7(z)[\partial, \prime]^{3\times 3}$$

is a row reduced form of A (1.2). The Popov form is the unique row reduced form of A which satisfies some additional conditions. Each diagonal entry is monic and has degree in ∂ higher or equal than entries to its left and strictly higher than entries above it, below it and to its right (a formal definition is given in Section 2.2). As an example, the matrix A from (1.2) has Popov form

$$P = \begin{bmatrix} 1 + (z+2)\partial + \partial^2 & & 5 \\ \frac{2}{z} & 1 & \\ 6 & & 1 + \partial \end{bmatrix} \in \mathbb{Z}_7(z)[\partial, \prime]^{3\times 3}.$$

In coding theory, the base field is usually a finite field, and faster methods than what we present are possible. In this thesis we mainly focus on the case where F is infinite, for example $\mathsf{F} = \mathsf{k}(z)$, even $\mathsf{k} = \mathbb{Q}$. When F is infinite, the infamous problem of intermediate coefficient swell becomes a concern. We use a linearization technique [Kaltofen et al., 1990, Labhalla et al., 1992, Storjohann, 1994, Giesbrecht and Kim, 2012] which allows us to map the problem back to the commutative domain where we can use some of the known techniques (e.g., modular or fraction-free algorithms) to control coefficient growth.

[Labhalla et al., 1992] gave an algorithm to compute the Hermite form over a commutative ring of polynomials based on linearization. They show that the Hermite rows of an input matrix could be recovered from the row echelon form comutation of a large linearized matrix. We adapt their approach to the Ore case. We show how their algorithm can be extended to compute the Hermite form of a square non-singular input matrix of Ore polynomials. We note that the idea of building a larger matrix than the input and computing the row echelon form to recover the rows of the solution has also been used in [Beckermann and Labahn, 2000] and [Jeannerod et al., 2017].

The problem of computing reduced matrices and canonical forms over $\mathsf{F}[\partial;\sigma,\delta]$ where F is an infinite field has been studied in the past and different algorithms have been introduced. [Beckermann et al., 2006, Cheng, 2003] gave a fraction-free algorithm for computing row reduced matrices of Ore polynomials. Their approach was based on computing "order basis" as known from $\mathsf{F}[x]$ matrix arithmetic, and using them to retrieve the reduced form. In this thesis we will review and use some of the properties established in their paper.

A year later, [Cheng and Labahn, 2007] improved on the approach used by [Beckermann et al., 2006] and gave a modular algorithm for computing the reduced form of matrices. [Cheng and Labahn, 2007] improved the overall cost of computing the row reduced form of matrices (compared to [Beckermann et al., 2006]). [Cheng and Labahn, 2007] still used an order basis approach but it was a modular algorithm and not fraction-free.

[P. Davies, 2008] reduce the problem of computing the Popov form to that of computing the nullspace of the input matrix, a problem for which fraction-free [Beckermann et al., 2006] and modular algorithms [Cheng and Labahn, 2007] can be used.

[Giesbrecht and Kim, 2012] gave an algorithm to compute the Hermite form H of a input matrix A over the ring $\mathsf{F}[\partial;\sigma,\delta]$. Their approach consists of linearizing the input matrix A to a larger matrix over F. They can then use techniques from commutative domains such as homomorphic imaging. [Giesbrecht and Kim, 2012] derive good bounds on the degrees in ∂ of the unimodular transformation matrix U such that $UA = H$. We will appeal to some of the bounds and techniques established in [Giesbrecht and Kim, 2012].

In this thesis, we present algorithms to compute the Hermite form and Popov form of a non-singular square input matrix. We give a detailed cost analysis for these algorithms in Chapters 4 and 5 but we will summarize some of these costs here and compare, to the best of our abilities, with the best algorithms in the literature. Let $A \in \mathsf{k}(z)[\partial;\sigma,\delta]^{n\times n}$ be a non-singular matrix over the Ore ring $\mathsf{F}[\partial;\sigma,\delta]$ for $\mathsf{F} = \mathsf{k}(z)$ where k is a field. Recall that the entries in A are polynomials in ∂ with the coefficients being rational functions in $\mathsf{k}(z)$. We can assume, without loss of generality, by clearing denominators of these coeffcients in $\mathsf{k}(z)$, that A is over $\mathsf{k}[z][\partial;\sigma,\delta]$.

A running time estimate in terms of operations from k thus involves three parameters:

- n, the dimension if the square input matrix A.
- d, an upper bound on the degree in ∂ of the entries in A.
- e, an upper bound on the degree in z of the coefficients of the Ore polynomial entries in A.

Our main algorithm for Popov form computation, reported in Chapter 5, constructs from A a structured matrix of dimension $O(n^2d) \times O(n^2d)$ over $\mathsf{k}[z]$. It then performs a structured fraction-free Gaussian elimination to recover the Popov form. The cost of the algorithm is $O(n^{\omega+2}d^3\,\mathsf{M}(n^2de))$ operations from k. Here, ω is an exponent for matrix multiplication, and M is a multiplication time: two polynomials from $\mathsf{k}[z]$ of degree strictly less than t can be multiplied in $\mathsf{M}(t)$ operations from k. Assuming $\omega = 3$ and a pseudo-liner multiplication time, and ignoring logarithmic factors, the cost of our algorithm is then on the order of n^7d^4e operations from k. For comparison, the fraction-free algorithm supporting [Beckermann et al., 2006, Corollary 7.7] seems to require on the order of $n^9d^4e^2$ operations from k to produce a row reduced form of A, while the modular algorithm in [Cheng and Labahn, 2007, Theorem 6.2] requires on the order of $n^8d^4e + n^7d^3e^2$ operations from k.

Now consider the case $\mathsf{k} = \mathbb{Q}$. Like before, we assume our input matrix is over $\mathbb{Z}[z][\partial;\sigma,\delta]$. Ignoring logarithmic factors, and again assuming pseudo-linear integer arithmetic, our algorithm requires on the order of $n^9 d^5 e \log\beta$ bit operations. Here, β is a paramater that depends on the magnitude of integer coefficients in A. The modular algorithm supporting [P. Davies, 2008, Theorem 6.3] seems to require about $n^{10} d^5 e \log\beta + n^9 d^4 e^2 \log\beta$ bit operations.

On the one hand, we point out that the algorithms of [Beckermann et al., 2006, Cheng and Labahn, 2007] solve a considerably more general problem then we do in this thesis: their algorithm can be applied to an input matrices of arbitrary shape and rank. Although we hope to consider the rank deficient case in the future, our analysis currently assumes the input matrix is square nonsingular. On the other hand, the algorithms in [Beckermann et al., 2006, Cheng and Labahn, 2007] only produce a row reduced form of A and not the canonical Popov form. In many applications a row reduced form may be sufficient, but in some cases the canonical Popov form can be asymptotically smaller than a row reduced form [Jeannerod et al., 2016, Appendix B].

Beyond the improved asymptotic worst case cost estimates we have reported above, our algorithm has two additional noteworthy features. First, in the shift case, the worst case running times we have reported above are improved by a factor of n: the linearized system has a special shape in this case which the algorithm is able to exploit. Second, for inputs that are are not too far from being row reduced the running time is asymptotically faster. The *orthogonality defect* of A is the difference between the sum of the row degrees in A and the sum of the row degrees in its Popov form P, denoted by $\mathsf{OD}(A)$. Our algorithms reported in Chapter 5 are output sensitive in the parameter $\mathsf{OD}(A)$, which can be as small as 0 and as large as nd. If $n \leqslant \mathsf{OD}(A) \leqslant nd$ then the running times reported above are improved by a factor of $\mathsf{OD}(A)/(nd)$. For $\mathsf{OD}(A) < n$ further improvements are obtained. In the special case $\mathsf{OD}(A) = 0$, which means the input matrix is already reduced, the algorithm detects this and avoids the lions share of the computation, instead applying a fast normalization to transform the input to

Popov form.

Our extension of [Labhalla et al., 1992] that is described in Chapter 4, which deterministically compute the Hermite form of A, constructs a linearized matrix of size $O(n^2d) \times O(n^2d)$ over $\mathsf{k}[z]$. It then performs a Gauss-Jordan elimination on the linearized matrix and extracts the rows of the Hermite form from the transformed system. The cost of this algorithm is $\mathcal{O}(n^6d^3\mathsf{M}(n^2de))$ operations from k. For comparison, the algorithm described in [Giesbrecht and Kim, 2012] computes deterministically the Hermite form of an input matrix A of Ore polynomials in $\mathcal{O}(n^7d^3\log(nd)\mathsf{M}(n^2de))$ operations from k. We get an $n\log(nd)$ speed up as the algorithm of [Labhalla et al., 1992] does not need to know the Hermite row degrees in advance, which is necessary in the approach of [Giesbrecht and Kim, 2012]. [Giesbrecht and Kim, 2012] do n binary searches to find the row degrees of the Hermite form which costs $n\log(nd)$ operations from k (hence, the speed up). Then they solve a large linear system to recover the entries of the transformation matrix $U \mid UA = H$. [Giesbrecht and Kim, 2012] also describe a Las Vegas randomized algorithm for computing the Hermite form of an Ore polynomial matrix with cost $O\tilde{}(n^7d^3e)$ operations in k.

Let A be a $n \times n$ non-singular matrix over $\mathsf{F}[\partial;\sigma,\delta]$ such that $\deg A \leqslant d$. The main contributions of this thesis are the followings:

- For the problem of computing the Hermite normal form of A, we give an extension of the algorithm from [Labhalla et al., 1992] to the Ore case which improves on the complexity of the deterministic algorithm of [Giesbrecht and Kim, 2012] (the best algorithm to our knowledge) by a factor of $n\log(nd)$.
- For the problem of computing the Popov normal form of A, we present a fraction-free output-sensitive algorithm which improves on the modular approach of [Cheng and Labahn, 2007] (the best algorithm for reduced form computation to our knowledge) by a factor of n. An implementation

of this algorithm and timing tables to compare with other algorithms from the literature are also given.

- We use some recent work related to polynomial matrix computations: rank profile [Zhou, 2012], structured inverse [Zhou et al., 2014], and linear solving [Zhou et al., 2014], to present a fast algorithm that computes the unimodular transformation matrix U such that $UA = P$ and P is the Popov form of A.

The rest of this thesis is organized as follows.

In Chapter 2, we give some mathematical definitions and properties that will be used throughout.

In Chapter 3, we review the Mulders-Storjohann algorithm by [Mulders and Storjohann, 2003] which can be used to compute a reduced form or a Popov form, and we show how it can be applied to the Ore case. Then we discuss the intermediate coefficient swell problem which occurs when Mulders-Storjohann algorithm, or similar row reductions algorithms, are used. We then review some modular approaches which can be used to control coefficient growth.

In Chapter 4, we show how we can extend the algorithm by [Labhalla et al., 1992] to compute the Hermite form of an input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$. To do that we need to linearize the input matrix by mapping it to a larger matrix of size $\mathcal{O}(n^2d) \times \mathcal{O}(n^2d)$ with entries from the field. We then apply Jordan-Gaussian elimination on the large matrix and retrieve the Hermite form at the end.

In Chapter 5, we present a fast algorithm for Popov form computation of an input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$, F a field. First, we show how to construct an $\mathcal{O}(n^2d) \times \mathcal{O}(n^2d)$ block upper triangular matrix with entries from the field. We then explain how we can exploit the structural properties of the large matrix and retrieve the Popov form of A. We give a cost analysis in terms of operations from F and then in terms of operations from k, for $\mathsf{F} = \mathsf{k}(z)$ and k a field. We also give the bit complexity when $\mathsf{k} = \mathbb{Q}(z)$ for the differential Ore ring $\mathbb{Q}(z)[\partial; ']$ (where $\sigma(z) = z$, $\delta(z) = 1$) and the shift Ore ring $\mathbb{Q}(z)[\partial; \sigma, \delta]$ (where $\sigma(z) = z + 1$, $\delta(z) = 0$). We conclude this chapter by giving some timings and experimental results. This chapter is joint work with Arne Storjohann and Johan Rosenkilde.

In Chapter 6, We show how we can recover the unimodular transformation matrix $U \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$, after computing the Popov form $P \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$, of a non-singular input matrix $A \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ such that $UA = P$ using linear system solving. We then give an algorithm to compute the transformation matrix U directly without having to know the Popov form a priori. The Popov form can then be obtained immediately by multiplying $U \mod \partial^{d+1}$ by $A \mod \partial^{d+1}$ where d is the degree in ∂ of A.

In Chapter 7, we summarize the different algorithms described in this thesis and give some future work directions.

Chapter 2

Preliminaries

For the rest of this thesis, let $A_{i,j}$ denote the entry in $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ at the i'th row and the j'th column. Let $\mathrm{row}_i A$, $\mathrm{col}_i A$ denote the i'th row and i'th column of A respectively for $1 \leqslant i \leqslant n$.

Let $\mathsf{F}[\partial;\sigma,\delta]$ be an Ore polynomial ring. The *degree* of a row vector $\vec{v} \in \mathsf{F}[\partial;\sigma,\delta]^{1\times n}$, denoted $\deg \vec{v}$, is the maximal degree in ∂ of entries of $\vec{v}$ (we define $\deg 0 = -\infty$). For example, if

$$v = [z\partial \quad | \quad 6z^3\partial^2 \quad | \quad (3z+1)\partial^2],$$

then $\deg v = 2$.

Similarly, the *degree* of the matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$, denoted $\deg A$, is the maximal degree of entries of A. For example, if

$$A = \begin{bmatrix} z\partial & 4 & 3(z+1) \\ (4z^2+2z)\partial & 2z^2 & \partial^2 + z \\ 6\partial^2 + 2\partial & 5z^3 & (3z+1)\partial^3 \end{bmatrix} \in \mathbb{Q}(z)[\partial;\prime]^{3\times 3},$$

then $\deg A = 3$.

By rdeg$A$ we mean the list $[d_1, d_2, \ldots, d_n]$ where d_i is the degree of row$_i A$, $1 \leqslant i \leqslant n$.

Different Ore rings can be introduced based on the definition of σ and δ. Next we will list some examples of those rings. Examples i, ii, iv, v, vi, and viii are also mentioned in Example 2.2 from [Cheng, 2003]. Examples iii and vii are mentioned in Example 2.1 from [Abramov et al., 2005]. Let $f(z) \in \mathsf{k}(z)$ be a rational polynomial function in z, for $\mathsf{F} = \mathsf{k}(z)$ and k a field.

i The ring of *differential polynomials* where $\sigma(f(z)) = f(z)$ and $\delta(f(z)) = f'(z)$ is the usual derivative with respect to z. We will follow [Giesbrecht and Kim, 2012] and use $\mathsf{k}(z)[\partial; \prime]$ notation to describe it. ∂ here represents the differential operator.

ii The ring $\mathsf{F}[\partial; \sigma, \delta]$ where σ is the shift automorphism and δ is the zero map: $\sigma(f(z)) = \mathcal{S}(f(z)) = f(z+1)$ and $\delta(f(z)) = 0$. $\mathcal{S}$ here represents the shift operator.

iii The ring $\mathsf{F}[\partial; \sigma, \delta]$ where $\sigma(f(z)) = f(z+1)$ and $\delta(f(z)) = f(z+1) - f(z)$. ∂ here represents the difference operator.

iv The ring $\mathsf{F}(z,q)[\partial; \sigma, \delta]$ where $\sigma(f(z)) = f(qz)$ and $\delta(f(z)) = 0$. ∂ here represents the q-shift operator.

v The ring $\mathsf{F}(z,k)[\partial; \sigma, \delta]$ where $\sigma(f(z)) = f(z^k)$ and $\delta(f(z)) = 0$. ∂ here represents the Mahlerian operator.

vi The ring $\mathsf{F}[\partial; \sigma, \delta]$ where $\sigma(f(z)) = f(z)$ and $\delta(f(z)) = zf'(z)$. ∂ here represents the Eulerian operator.

vii The ring $\mathsf{F}(z,q)[\partial; \sigma, \delta]$ where $\sigma(f(z)) = f(qz)$ and $\delta(f(z)) = f(qz) - f(z)$. ∂ here represents the q-difference operator.

viii The ring $\mathsf{F}(z,q)[\partial; \sigma, \delta]$ where $\sigma(f(z)) = f(qz)$ and $\delta(f(z)) = \frac{f(qz)-f(z)}{(q-1)z}$. ∂ here represents the q-differentiation operator.

2.1 Rank

For a matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$, we denote by the rank the number of rows of A which are linearly left-independent (the left space spanned by A's rows). As shown by [Giesbrecht and Kim, 2012, Beckermann et al., 2006], row operations on an input matrix of Ore polynomials will not change its rank. Also, two matrices $M, M' \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ generate the same left row space if and only if there exists $U \in \mathrm{GL}_n(\mathsf{F}[\partial;\sigma,\delta])$ such that $M = UM'$, where $\mathrm{GL}_n(\mathsf{F}[\partial;\sigma,\delta])$ denotes the set of invertible $n \times n$ matrices over $\mathsf{F}[\partial;\sigma,\delta]$.

Example 1. *Consider the matrix*

$$A = \begin{bmatrix} z^3 + z^2\partial & 2 & 9+\partial \\ 5z^3 & z+1 & 3\partial \\ 6z^3 + z^2\partial & z+3 & 9+4\partial \end{bmatrix} \in \mathbb{Z}_{11}(z)[\partial, \prime]^{3\times 3}.$$

We can clearly see that the third row is just the sum of the first two rows which means it is linearly dependent. So if we apply the transformation $\mathrm{row}_3 A \leftarrow \mathrm{row}_3 A - (\mathrm{row}_1 A + \mathrm{row}_2 A)$ *on A, we get*

$$\begin{bmatrix} z^3 + z^2\partial & 2 & 9+\partial \\ 5z^3 & z+1 & 3\partial \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{Z}_{11}(z)[\partial, \prime]^{3\times 3}.$$

The first two rows are linearly independent, so the rank of A will be equal to 2.

One way to figure out the rank of a certain matrix is by computing its row reduced form. A row reduced form will have rows with degrees reduced to the minimum (i.e., no linear combinations of the matrix rows will produce anything with lower degrees).

Definition 2. *Let $A \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ be a non-singular matrix of Ore polynomials with the row degrees of A equal to* $\mathrm{rdeg} A = [d_1, d_2, \ldots, d_n]$. *The* leading matrix *of A, denoted $\mathsf{LM}(A) \in \mathsf{F}^{n\times n}$, is the matrix whose (i, j) entry is the coefficient of ∂^{d_i} of $A_{i,j}$.*

Example 3. *let us consider the matrix A such that*

$$A = \begin{bmatrix} 1 + z\partial & 5\partial + z\partial^4 & 10 + \partial^2 \\ 4\partial & (2z + 4) & 1 + \partial \\ \partial^2 & 3\partial + 2z^3\partial^2 & 1 + (5z^4 + 4z)\partial^2 \end{bmatrix} \in \mathbb{Z}_{11}(z)[\partial, \prime]^{3\times 3}$$

with $\mathrm{rdeg}(A) = (4, 1, 2)$. *The leading matrix of A is*

$$\mathsf{LM}(A) = \begin{bmatrix} & z & \\ 4 & & 1 \\ 1 & 2z^3 & (5z^4 + 4z) \end{bmatrix} \in \mathbb{Z}_{11}(z)^{3\times 3}.$$

Definition 4. *Let $A \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ be a non-singular matrix of Ore polynomials. A is said to be* row reduced *if and only if* $\mathrm{rank}(\mathsf{LM}(A)) = n$.

We consider the matrix A from Example 3. $\mathsf{LM}(A)$ has full row rank and so A is row reduced.

2.2 Popov form

In this section we define the Popov form, show its properties and establish some bounds and assumptions that will be used in later chapters.

As shown in Section 1.1, a system of differential equations can be modeled by a matrix of differential Ore polynomials. By computing the Popov form, we get a canonical row reduced basis for the row space of that matrix with minimal row degrees. Also, using Popov form we can express derivatives with high order in terms of other derivatives with lower order.

The Popov form can be defined for any matrix of arbitrary shape and rank, but in this thesis we will only focus on the case of non-singular square matrices.

Definition 5. *A non-singular matrix $P \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ is in Popov form if* $\mathrm{LM}(P)$ *is unit lower triangular and the degrees of off-diagonal entries of P are strictly less than the degree of the diagonal entry in the same column.*

For the rest of this section, we define the *pivot* of a row vector $\vec{v} \in \mathsf{F}[\partial;\sigma,\delta]^{1\times n}$, denoted $\mathrm{piv}(\vec{v})$, as right-most entry of $\vec{v}$ which has $\deg \vec{v}$. For example, for $n = 4$, if

$$\vec{v} = [2\partial^2 + 4\partial + 1 \mid 5 + 6\partial \mid \mathbf{9\partial^2} \mid 5\partial],$$

then the pivot entry would be $9\partial^2$.

Notice from the definition of Popov form (5), that the pivot entry of $\mathrm{row}_i P$ is always at column index i, with $1 \leqslant i \leqslant n$.

Example 6. *Let $A \in \mathbb{Z}_7[z][\partial;\prime]^{3\times 3}$ be defined as follows.*

$$A = \begin{bmatrix} 1+(5+z)\partial+\partial^2 & & 5+4\partial+4\partial^2 \\ 4+3\partial & 2z & 4\partial+4\partial^2 \\ 3 & & 4+4\partial \end{bmatrix} \rightarrow \overbrace{\begin{bmatrix} 2 & -\infty & \mathbf{2} \\ 1 & 0 & \mathbf{2} \\ 0 & -\infty & \mathbf{1} \end{bmatrix}}^{\textit{degree structure}}.$$

The Popov form $P \in \mathbb{Z}_7(z)[\partial;\prime]^{3\times 3}$ of A is

$$\begin{bmatrix} 1+(z+2)\partial+\partial^2 & & 5 \\ \frac{2}{z} & 1 & \\ 6 & & 1+\partial \end{bmatrix} \rightarrow \overbrace{\begin{bmatrix} \mathbf{2} & -\infty & 0 \\ 0 & \mathbf{0} & -\infty \\ 0 & -\infty & \mathbf{1} \end{bmatrix}}^{\textit{degree structure}}.$$

In the example above, notice how in the Popov form all pivots have different column index. Also, all pivots dominate their columns in terms of degree in ∂, that is every entry in that column (other than the pivot itself) will have degree strictly less than the pivot degree.

Lemma 7. *Let $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ and $P \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be the Popov form of A, then P is row reduced.*

Proof. This follows immediately as the leading matrix of P will always be lower triangular with unitary diagonal entries. □

Remark 8. *A matrix in Popov form is row reduced but the converse is not true.*

The following is classical for $\mathsf{F}[x]$ matrices, see [Kailath, 1980, Section 6.3.2]. For the extension to $\mathsf{F}[\partial;\sigma,\delta]$ matrices, see [Beckermann et al., 2006, Lemma A.1 (a)]. The last item is often called the Predictable Degree Property.

Theorem 9. *Let $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be non-singular. Then the following are equivalent:*

1. *A is row reduced.*
2. *Among all matrices that are left equivalent to A, the list of row degrees of A, when sorted in non-decreasing order, will be lexicographically minimal.*
3. *For any $\vec{v} \in \mathsf{F}[\partial;\sigma,\delta]^{1\times n}$, we have*

$$\deg(\vec{v}A) = \max_{i=1,\ldots,n} \left(\deg \operatorname{row}_i A + \deg v_i\right) .$$

Lemma 10. *If $A, U, P \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$, all non-singular and U invertible and P in Popov form such that $UA = P$, then $\deg U \leqslant (n-1)\deg A$.*

Proof. By Item 3 of Theorem 9, then $\deg U^{-1} \leqslant \deg A$ since the degree of P is non-negative. By [Giesbrecht and Kim, 2012, Corollary 3.3] then $U = (U^{-1})^{-1}$ has degree at most $(n-1)\deg A$. □

The following notion is a measure for how far A is from being row reduced:

Definition 11. *Let $A \in \mathsf{F}[\partial;\sigma,\delta]$ and non-singular. The* orthogonality defect *of A, denoted $\mathsf{OD}(A)$, is given as $\sum \operatorname{rdeg} A - \sum \operatorname{rdeg} P$, where P is the Popov form of A.*

In Example 6, the sum of the row degrees of A is equal to 5, the sum of the row degrees of P is equal to 3. So the orthogonality defect of A is: $OD(A) = 5 - 3 = 2$.

Lemma 12. *If A is row reduced then $\mathsf{OD}(A) = 0$.*

Proof. Follows immediately from Theorem 9, Item 2. $\square$

2.3 Hermite form

For the rest of this section, we define the *pivot* of a row vector $\vec{v} \in \mathsf{F}[\partial; \sigma, \delta]^{1\times n}$, denoted $\mathrm{piv}(\vec{v})$, as the left-most non-zero entry in $\vec{v}$. For example, for $n = 4$, if

$$\vec{v} = [0 \mid 0 \mid \mathbf{2\partial^2 + 4\partial} \mid 9\partial^4],$$

then the pivot entry would be $2\partial^2 + 4\partial$.

The Hermite form can be defined for any matrix of arbitrary shape and rank, but in this thesis we will only focus on the case of non-singular square matrices.

Definition 13. *A matrix $H \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ is in Hermite form if and only if it satisfies the following:*

- *i* *H is upper triangular;*
- *ii* *For $1 \leqslant i \leqslant n$, $H_{i,i}$ has degree strictly higher than all other entries in the same column;*
- *iii* *All pivot entries (i.e., diagonal entries) are monic.*

Similar to the Popov form, the Hermite form pivot entries has to dominate the column in terms of degree in ∂ (Definition 13, item ii). On the other hand the pivot entries do not have to dominate the degree in ∂ of their respective rows.

Example 14. *Let A be defined as in Example 6 then H, the Hermite form of A, is equal to*

$$H = \begin{bmatrix} 1 & & 6+6\partial \\ & 1 & \frac{2}{z}+\frac{2}{z}\partial \\ & & 6+(z+3)\partial+(z+3)\partial^2+\partial^3 \end{bmatrix} \rightarrow \overbrace{\begin{bmatrix} \mathbf{0} & -\infty & 1 \\ -\infty & \mathbf{0} & 1 \\ -\infty & -\infty & \mathbf{3} \end{bmatrix}}^{\textit{degree structure}}.$$

Notice that the pivot entries do not dominate the row degrees (i.e., you can have entries with higher degrees to the right of the pivot) but they do dominate the column degrees: Entries below the pivot are zero and entries above the pivot have degrees strictly less than that of the pivot.

Chapter 3

Techniques and challenges

In this chapter, we will start by reviewing the Mulders-Storjohann algorithm [Mulders and Storjohann, 2003] which can be used to compute a row reduced form and the Popov form of an input matrix. The complexity for the algorithm is $\mathcal{O}(n^3 d^2)$ operations from the field F, for an input matrix A of degree d filled with ordinary commutative polynomials, that is, from $\mathsf{F}[x]$. Even though there are faster methods in the literature to compute the Popov form over $\mathsf{F}[x]$ (e.g., [Gupta et al., 2012] and [Neiger, 2016]), we choose to review the Mulders-Storjohann algorithm as it extends easily to the case of matrices over an Ore ring $\mathsf{F}[\partial;\sigma,\delta]$.

The Mulders-Storjohann algorithm [Mulders and Storjohann, 2003] has a polynomial complexity in F but a more refined cost analysis should consider the growth in the size of the coefficient from the field when coefficient growth is a concern. We give an example in Section 3.2 that shows intermediate coefficient swell using the Mulders-Storjohann algorithm [Mulders and Storjohann, 2003]. In Section 3.3, we give a brief review of some well known methods in the commutative domains that are used to control this growth. Then, we discuss the possibility of applying some of these approaches to the Mulders-Storjohann algorithm when applied to the Ore case.

For the rest of this chapter, and as we did in Section 2.2, we define the *pivot* of a vector $\vec{v} \in \mathsf{F}[\partial;\sigma,\delta]^{1\times n}$, denoted $\operatorname{piv}(\vec{v})$, as right-most entry of $\vec{v}$ which has $\deg \vec{v}$.

3.1 Mulders-Storjohann algorithm

The Mulders-Storjohann algorithm [Mulders and Storjohann, 2003] is used to compute a row reduced form, and can be exploited to further compute the Popov form [Mulders and Storjohann, 2003]. The algorithm can compute the Popov form of a square non-singular input matrix $A \in \mathsf{F}[x]^{n\times n}$, F a field, without increasing the row degrees of the input throughout the computation. It finds any two pivots in A which share the same column and uses the one with lower degree to reduce the one with higher degree. The algorithm stops when all pivots have distinct column indices, in which case the leading matrix has an upper triangular shape (when rows are sorted by pivot column index), hence the matrix is row reduced by Definition 4.

Example 15. *Let $A \in \mathbb{Q}[x]^{2\times 2}$ be an input matrix. We follow Mulders-Storjohann algorithm [Mulders and Storjohann, 2003] to compute a row reduced form of A.*

$$A = \begin{bmatrix} 0 & 5+3x \\ 2 & 1+x+\frac{-4}{3}x^2 \end{bmatrix} \rightarrow \overbrace{\begin{bmatrix} -\infty & \mathbf{1} \\ 0 & \mathbf{2} \end{bmatrix}}^{\textit{degree structure}}$$

$$\xRightarrow{\text{row}_2 \leftarrow \text{row}_2 + (\frac{4}{9}x)\text{row}_1}$$

$$\begin{bmatrix} 0 & 5+3x \\ 2 & 1+\frac{29}{9}x \end{bmatrix} \rightarrow \begin{bmatrix} -\infty & \mathbf{1} \\ 0 & \mathbf{1} \end{bmatrix}$$

$$\xRightarrow{\text{row}_1 \leftarrow \text{row}_1 - \frac{27}{29}\text{row}_2}$$

$$\begin{bmatrix} \frac{-54}{29} & \frac{118}{29} \\ 2 & 1+\frac{29}{9}x \end{bmatrix} \to \begin{bmatrix} 0 & \mathbf{0} \\ 0 & \mathbf{1} \end{bmatrix}$$

$$\xRightarrow{\text{row}_2 \leftarrow \text{row}_2 - (\frac{29^2}{1062}x)\text{row}_1}$$

$$\begin{bmatrix} \frac{-54}{29} & \frac{118}{29} \\ 2+\frac{87}{59}x & 1 \end{bmatrix} \to \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{1} & 0 \end{bmatrix}$$

$$\xRightarrow{swap(\text{row}_1,\text{row}_2)}$$

$$\begin{bmatrix} 2+\frac{87}{59}x & 1 \\ \frac{-54}{29} & \frac{118}{29} \end{bmatrix} \to \begin{bmatrix} \mathbf{1} & 0 \\ 0 & \mathbf{0} \end{bmatrix}$$

Notice in the last matrix how the pivots (in bold font) do not share the same column. The leading matrix, like it was stated earlier, is upper triangular

$$\mathsf{LM}(A) = \begin{bmatrix} \frac{87}{59} & \\ \frac{-54}{29} & \frac{118}{29} \end{bmatrix}.$$

The algorithm extends naturally to the non-commutative Ore case thanks to the following Lemma.

Lemma 16. *Let us consider $s_1, s_2 \in \mathsf{F}[\partial;\sigma,\delta]$ two Ore polynomials such that $deg(s_1) \geqslant deg(s_2) \geqslant 0$, then there exists $t \in \mathsf{F}[\partial;\sigma,\delta]$ such that $deg(s_1 - ts_2) < deg(s_1)$.*

Proof. Let l_1 and l_2 be the leading coefficients of s_1 and s_2 respectively, and $\lambda \in \mathbb{Z}$ such that $\lambda = deg(s_1) - deg(s_2)$. Then the leading monomial of $\frac{l_1}{l_2}\partial^\lambda s_2$ will be equal to that of s_1. It follows that $deg(s_1 - \frac{l_1}{l_2}\partial^\lambda s_2) < deg(s_1)$. □

For an input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ of degree d, the cost of the algorithm is still $\mathcal{O}(n^3d^2)$ operations from F. The unimodular row operations used in the Ore case are the followings (analogous to the row operations used in the commutative domains):

- Swap two rows.
- Multiply one row by an element from the field F.
- Multiply a row by an element from $\mathsf{F}[\partial;\sigma,\delta]$ and add it to another row.

The Mulders-Storjohann algorithm [Mulders and Storjohann, 2003] was designed for the case when F is finite. Although the algorithm can be extended over any field F, it seems that the intermediate coefficients grow exponentially when doing reduction over an infinite field F, e.g., $\mathsf{F} = \mathsf{k}(z)$. This is known in the literature as the *intermediate coefficient growth* problem.

3.2 Intermediate coefficient growth

As stated in Section 3.1, the algorithm by [Mulders and Storjohann, 2003] can be used to compute a row reduced or a Popov form of an input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$. We consider the case $\mathsf{F} = \mathsf{k}(z)$, that is, the coefficients of the Ore polynomials are rational functions in z over a field k. A nice feature of the Mulders-Storjohann algorithm [Mulders and Storjohann, 2003] is that the row degrees (in ∂) of the entries in the matrix being reduced are monotonically non-increasing. However, the true cost of the algorithm will also depend on the size of the rational function coefficients from $\mathsf{k}(z)$. Abusing notation slightly, define the degree of a (reduced) rational function from $\mathsf{k}(z)$ as the maximum of the numerator and denominator degree. By extension, let $\deg_z f$ for an Ore polynomial $f \in \mathsf{k}(z)[\partial;\sigma,\delta]$ be the maximum degree of any coefficient of f. Although the degree in z of an input matrix A can be small, degrees in z of the intermediate entries in the matrix being reduced can grow large. In the following example, for a matrix M over $\mathsf{k}(z)[\partial;\sigma,\delta]$, let $\deg_z M$ be the integer matrix of degrees in z of entries in M, that is, each entry is replaced with the maximum degree in z of any rational function coefficient.

Example 17. *We consider an input matrix $A \in \mathbb{Z}_{11}(z)[\partial;\prime]^{5\times 5}$. Notice that the coefficients from $\mathbb{Z}$ in A are reduced modulo the prime integer 11 as we will only focus in this example on the degree growth of the coefficients from $\mathbb{Q}(z)$.*

$$\deg_z A \to \begin{bmatrix} 5 & 5 & 5 & 5 & 5 \\ 4 & 5 & 5 & 5 & 5 \\ 4 & 4 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}.$$

We follow the extension of the algorithm by [Mulders and Storjohann, 2003] described in Section 3.1 to compute a row reduced form $W \in \mathbb{Z}_{11}(z)[\partial;\prime]^{5\times 5}$,

$$\deg_z W \to \begin{bmatrix} 159 & 160 & 160 & 160 & 160 \\ 1158 & 1159 & 1159 & 1159 & 1159 \\ 1810 & 1811 & 1811 & 1811 & 1811 \\ 1917 & 1918 & 1918 & 1918 & 1917 \\ 1909 & 1909 & 1910 & 1910 & 1907 \end{bmatrix}.$$

Notice the blow-up in the degree in z of the coefficients from $\mathbb{Z}_{11}(z)$. The highest degree in z in W is 1918 versus 5 in our input matrix. We compute the Popov form $P \in \mathbb{Z}_{11}(z)[\partial;\prime]^{3\times 3}$ of our input matrix A,

$$\deg_z P \to \begin{bmatrix} 159 & 160 & 160 & 160 & 160 \\ 137 & 138 & 138 & 138 & 138 \\ 135 & 136 & 136 & 136 & 136 \\ 136 & 136 & 136 & 136 & 136 \\ 136 & 137 & 137 & 137 & 137 \end{bmatrix}.$$

Example 17 shows how the intermediate coefficient swell can be a problem when using the [Mulders and Storjohann, 2003] algorithm in the Ore case. Other examples to show intermediate coefficient swell do exist in the literature. For instance, [Hafner and McCurley, 1991] gave an example where they started with a 20×20 matrix with integer entries between 0 and 10, then after triangularization, they had an entry of size 10^{5011} in the output.

There are two major methods to contain the coefficient growth problem. The first one is to apply fraction-free algorithms (e.g., fraction-free Gaussian elimination [Geddes et al., 1992]), while the second one is using modular algorithms where we compute images of the solution then recover the result at the end.

3.3 Modular algorithms

We call modular algorithms any approach that uses modular homomorphisms to control coefficient growth. For instance, one can work modulo some primes and use the chinese remaindering theorem to reconstruct the final answer from the images of the solution. When coefficients have a polynomial structure (coefficients from $\mathsf{k}(z)$ for instance, where k is a field), an evaluation homomorphism can be employed which maps the problem from $\mathsf{k}(z)$ to k. Lifting and rational function reconstruction can then be used to recover the final result. When employing evaluation homomorphisms or computing modular homomorphisms, the primes chosen have to be non-roots of the largest invariant factor of the matrix in question, otherwise the result computed may not be a proper image of the desired solution. Also, one can work modulo a big irreducible (bigger than the entries in the final result) and use rational function reconstruction to recover the final result at the end.

To be able to use the algorithms described above, we need to have bounds on the size of the coefficients in the final result. In Section 5.3, we derive bounds on the degree and on the size of the coefficients in the final result for an Ore ring $\mathsf{F}[\partial;\sigma,\delta]$, where $\mathsf{F} = \mathsf{k}(z)$ and k a field. This suggests using a modular scheme for our extension of the Mulders-Storjohann algorithm. Unfortunately, none of the modular algorithms described above will yield a correct result as the following map does not commute with multiplication by ∂.

$$p \mapsto p \mod \Gamma$$

with $p \in \mathsf{F}[\partial;\sigma,\delta]$ and $\Gamma \in \mathsf{k}[z]$, for $\mathsf{F} = \mathsf{k}(z)$ and k a field. For example, let $p = z\partial \in \mathbb{Q}(z)[\partial;\prime]$ and $\Gamma = z - 2 \in \mathbb{Q}(z)$. If we first premultiply p by ∂ and then reduce the result modulo Γ, we obtain the following:

$$\begin{aligned}(\partial p) \bmod (z-2) &= (\partial z\partial) \bmod (z-2)\\ &= (z\partial^2 + \partial) \bmod (z-2)\\ &= 2\partial^2 + \partial.\end{aligned}$$

However if we first reduce p modulo Γ, then premultiply by ∂ then we get

$$\begin{aligned}\Big(\partial \bmod (z-2)\Big)\Big(p \bmod (z-2)\Big) &= (\partial)(2\partial)\\ &= 2\partial^2.\end{aligned}$$

We can clearly see from the above example that the modular algorithms will not yield a correct image of the result for some Ore rings.

So, to summarize, the Mulders-Storjohann algorithm can be extended to the Ore case to compute the Popov form of an input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$. When F is infinite (i.e., coefficient growth is a concern), modular approaches cannot be incorporated with the algorithm to contain the intermediate coefficient swell. We will see in later chapters how we can use some linearization techniques to map A to a larger matrix with entries from F. This will allow us to employ some modular algorithms to control coefficient growth when doing row operations.

Chapter 4

Computing the Hermite form of a matrix of Ore polynomials via linearization

In this chapter we describe an extension to the Ore case of the method of [Labhalla et al., 1992] which computes the Hermite form of a matrix with entries from $\mathsf{F}[x]$. [Giesbrecht and Kim, 2012] gives an algorithm which computes the Hermite form of an input matrix $A \in \mathsf{F}[\partial; \sigma, \delta]^{n \times n}$ of degree d in $\mathcal{O}(n^7 d^3 \log nd)$ operations from F. The algorithm by [Giesbrecht and Kim, 2012] does a binary search to find the Hermite column degrees first then solve a linear system to get the actual rows corresponding to the Hermite form.

The nice thing about the approach of [Labhalla et al., 1992] is that you can get the Hermite rows directly from the the reduced row echelon form of a big linearized matrix without the need to know the Hermite column degrees a priori. Avoiding the search for the column degrees makes the extension of [Labhalla et al., 1992] method $n \log(nd)$ times faster than the deterministic algorithm of [Giesbrecht and Kim, 2012].

In Section 4.1 we will describe how we linearize the input matrix to a larger matrix over F. In Section 4.2 we will show how we extract the Hermite rows from the row reduced echelon form of the linearized matrix. We then prove the correctness of this algorithm and give the complexity analysis.

For the rest of this thesis, we define the pivot of a vector $\vec{v} \in \mathsf{F}^{1\times(n^2d+n)}$ as the index of the left-most non-zero element of $\vec{v}$ (not to be confused with the other two pivot definitions of an $\mathsf{F}[\partial;\sigma,\delta]$ vector, see Section 2.2 and 2.3). For the rest of this chapter, and as we did in Section 2.3, we define the *pivot* of a vector $\vec{v} \in \mathsf{F}[\partial;\sigma,\delta]^{1\times n}$, denoted $\text{piv}(\vec{v})$, as the left-most non-zero entry in $\vec{v}$.

4.1 Linearize the input matrix

In this section we show how we can convert an input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ such that $\deg A = d$ to $A_{\mathsf{lin}} \in \mathsf{F}^{(n^2d+n-dn)\times(n^2d+n)}$ where all the entries of A_{lin} are from F. This conversion will be used in the next section to compute the Hermite form of A.

Based on the parameters n and d, define the polynomial linearization $\phi_H : \mathsf{F}[\partial;\sigma,\delta]^{*\times n} \mapsto \mathsf{F}^{*\times(n^2d+n)}$ by

$$\begin{aligned}\phi_H(v) &= \phi_H \left[\begin{array}{ccc} v_1 & \cdots & v_n \end{array}\right] \\ &= \left[\begin{array}{ccc|c|ccc} [v_1]_{nd} & \cdots & [v_1]_0 & \cdots & [v_n]_{nd} & \cdots & [v_1]_0 \end{array}\right] \in \mathsf{F}^{*\times(n^2d+n)},\end{aligned}$$

where $[v_i]_k$ denotes the coefficient of ∂^k of $v_i \in \mathsf{F}[\partial;\sigma,\delta]^{*\times 1}$. The function ϕ_H maps each polynomial (modulo ∂^{nd+1}) to its coefficient vector of length $nd+1$, padded with zeroes if the polynomial has degree less than nd. For example, if $n = 2$ and $d = 2$ then

$$\phi_H(\left[\begin{array}{c|c} \partial^2+3\partial+6 & 2\partial+5 \end{array}\right]) = \left[\begin{array}{ccccc|ccccc} 0 & 0 & 1 & 3 & 6 & 0 & 0 & 0 & 2 & 5 \end{array}\right]. \tag{4.1}$$

[Giesbrecht and Kim, 2012, Theorem 4.9] establishes the following bound on the transformation matrix $U \in \mathsf{k}(z)[\partial;\sigma,\delta]^{n\times n}$ such that $UA = H$ is the Hermite form of A

$$\deg U \leqslant (n-1)d. \tag{4.2}$$

Based on the bound in (4.2), we construct the following linearized matrix:

$$\phi_H\left(\begin{bmatrix} \partial^{(n-1)d}A \\ \vdots \\ \partial A \\ A \end{bmatrix}\right) \in \mathsf{F}^{(n^2d+n-dn)\times(n^2d+n)}. \tag{4.3}$$

The row space of (4.3) is equal to

$$\left\{\phi_H\left(\sum_{i=1}^{n} u_i\,\mathrm{row_i}A\right) \mid u_i \in \mathsf{F}[\partial;\sigma,\delta],\ \deg u_i \leqslant (n-1)d,\ 1 \leqslant i \leqslant n\right\}. \tag{4.4}$$

Since (4.3) includes the derivatives of all rows of A up to $(n-1)d$, then the ϕ_H-linearized rows of the Hermite form of A are contained in the row space of (4.3).

4.2 Compute the Hermite form

[Labhalla et al., 1992] showed that computing the reduced row echelon form is enough to recover the Hermite rows when $A \in \mathsf{F}[x]^{n\times n}$. Similarly, we can recover the ϕ_H-linearized rows of the Hermite form when $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$.

We review the following lemma from [Giesbrecht and Kim, 2012, Lemma 5.1] which is analogous to [Storjohann, 1994, Lemma 4] but for the Ore case.

Lemma 18. *Let $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ non-singular matrix of Ore polynomials and $H \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be its Hermite form. Let $h_i = \deg H_{ii}$ for $1 \leqslant i \leqslant n$ be the pivot degrees of H. Let $\vec{v} = [0,\ldots,v_k,v_{k+1},\ldots,v_n] \in \mathsf{F}[\partial;\sigma,\delta]^{1\times n}$ such that $\deg v_k < h_k$ and $1 \leqslant k \leqslant n$. Finally let $\mathcal{L}(A) = \{\sum_{i=1}^{n} b_i\,\mathrm{row}_i A) \mid b_i \in \mathsf{F}[\partial;\sigma,\delta]\}$.*

If $v \in \mathcal{L}(A)$ then $v_k = 0$, otherwise if $v_k \neq 0$ then $v \notin \mathcal{L}(A)$.

Theorem 19. *Let A be non-singular, and let $\mathsf{K}_{\mathsf{lin}}$ be the reduced row echelon form of (4.3). Then the Hermite form H of A is the matrix whose i'th row is the ϕ_H^{-1}-image of the row of $\mathsf{K}_{\mathsf{lin}}$ with smallest pivot l such that $(i-1)(nd+1) < l \leqslant i(nd+1)$.*

Proof. We know that the unique unimodular matrix $U \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ has $\deg U \leqslant (n-1)d$ (4.2), the matrix constructed in (4.3) contains $(n-1)d$ derivatives of every row of A, therefore the ϕ_H-linearized rows of H are contained in the row space of K_{lin}.

For $1 \leqslant i \leqslant n$, consider the i'th row $\vec{h}$ of H, which has pivot column i. Let $\vec{r}_k$ be the row of K_{lin} with the smallest pivot l such that $(i-1)(nd+1) < l \leqslant i(nd+1)$. Let k be the row index of $\vec{r}_k$.

If $\vec{w}$ is the unique vector over F satisfying $\vec{w}K_{\mathsf{lin}} = \phi_H(\vec{h})$ then using Lemma 18, it is clear to see that $w_k = 1$ and $w_j = 0$ for $j < k$.

We claim $w_j = 0$ also for $j > k$ in which case $\vec{r}_k = \phi_H(\vec{h})$ as we wanted to prove.

Suppose, to arrive at a contradiction, that $w_j \neq 0$ for some $j > k$, and let $\vec{r}_j$ be the j'th row of K_{lin} such that j' and d' are the pivot-column and the pivot-degree of $\phi_H^{-1}(\vec{r}_j)$. This means that

$$\deg H_{i,j} \geqslant d'. \tag{4.5}$$

On the other hand, since we pick the smallest pivot l such that $(i-1)(nd+1) < l \leqslant i(nd+1)$ then for $i = j'$, we know that $l \geqslant \operatorname{piv}(\vec{r}_j)$, which means $\deg H_{j',j'} \leqslant d'$. But then, using (4.5), this means that $\deg H_{i,j'} \geqslant \deg H_{j',j'}$, which contradicts that H is in Hermite form (i.e., violates item ii from Definition 13).

We conclude that $W_j = 0$ for $j > k$, and hence $\phi_H(\vec{h}) = \vec{r}_k$. □

Notice that in Example 20, A has entries from $\mathsf{F}[\partial;\prime]$ and $\mathsf{F} = \mathbb{Z}_7$. This means that the Ore polynomials will behave exactly like ordinary commutative polynomials. This was chosen just for space reasons as our algorithm works for any field F (e.g: $\mathsf{F} = \mathsf{k}(z)$ where k is a field).

Example 20. *Consider the input*

$$A = \begin{bmatrix} 3\partial^2+6\partial+6 & 5\partial^2+3\partial+3 & 6\partial^2+\partial \\ 5\partial^2+5 & 6\partial^2+\partial & 5\partial^2+2\partial+6 \\ 3\partial+5 & 4\partial+5 & 5\partial^2+2\partial+1 \end{bmatrix} \in \mathsf{F}[\partial, \prime]^{3\times 3},$$

where $\mathsf{F} = \mathbb{Z}_7$.

$$\phi_H \begin{bmatrix} \partial^4 A \\ \vdots \\ \partial A \\ A \end{bmatrix} = \left[\begin{array}{ccccccc|ccccccc|ccccccc}
3&6&6&&&&&5&3&3&&&&&6&1&0&&&& \\
5&0&5&&&&&6&1&0&&&&&5&2&6&&&& \\
0&3&5&&&&&0&4&5&&&&&5&2&1&&&& \\
&3&6&6&&&&&5&3&3&&&&&6&1&0&&& \\
&5&0&5&&&&&6&1&0&&&&&5&2&6&&& \\
&0&3&5&&&&&0&4&5&&&&&5&2&1&&& \\
&&3&6&6&&&&&5&3&3&&&&&6&1&0&& \\
&&5&0&5&&&&&6&1&0&&&&&5&2&6&& \\
&&0&3&5&&&&&0&4&5&&&&&5&2&1&& \\
&&&3&6&6&&&&&5&3&3&&&&&6&1&0& \\
&&&5&0&5&&&&&6&1&0&&&&&5&2&6& \\
&&&0&3&5&&&&&0&4&5&&&&&5&2&1& \\
&&&&3&6&6&&&&&5&3&3&&&&&6&1&0 \\
&&&&5&0&5&&&&&6&1&0&&&&&5&2&6 \\
&&&&0&3&5&&&&&0&4&5&&&&&5&2&1
\end{array}\right] \in \mathsf{F}^{15\times 21}. \tag{4.6}$$

Note that up to row permutations, the matrix (4.6) has the shape

$$\left[\begin{array}{ccccccc|ccccccc|ccccccc}
3&6&6&&&&&5&3&3&&&&&6&1&&&&& \\
&3&6&6&&&&&5&3&3&&&&&6&1&&&& \\
&&3&6&6&&&&&5&3&3&&&&&6&1&&& \\
&&&3&6&6&&&&&5&3&3&&&&&6&1&& \\
&&&&3&6&6&&&&&5&3&3&&&&&6&1& \\
\hline
5&0&5&&&&&6&1&&&&&&5&2&6&&&& \\
&5&0&5&&&&&6&1&&&&&&5&2&6&&& \\
&&5&0&5&&&&&6&1&&&&&&5&2&6&& \\
&&&5&0&5&&&&&6&1&&&&&&5&2&6& \\
&&&&5&0&5&&&&&6&1&&&&&&5&2&6 \\
\hline
&3&5&&&&&&4&5&&&&&5&2&1&&&& \\
&&3&5&&&&&&4&5&&&&&5&2&1&&& \\
&&&3&5&&&&&&4&5&&&&&5&2&1&& \\
&&&&3&5&&&&&&4&5&&&&&5&2&1& \\
&&&&&3&5&&&&&&4&5&&&&&5&2&1
\end{array}\right] \in \mathsf{F}^{15\times 21},$$

which better illustrates the Sylvester-like structure. The reduced row echelon form *of (4.6) is*

$$\left[\begin{array}{ccccccc|ccccccc|ccccccc}
1 & & & & & & & 4 & & & & & & 0 & 2 & 1 & & & & 4 & 0 \\
 & 1 & & & & & & 0 & & & & & & 3 & 6 & 1 & & & & 0 & 4 \\
 & & 1 & & & & & 0 & & & & & & 4 & 0 & 6 & & & & 5 & 3 \\
 & & & 1 & & & & 0 & & & & & & 3 & 0 & 0 & & & & 0 & 4 \\
 & & & & 1 & & & 0 & & & & & & 4 & 0 & 0 & & & & 4 & 3 \\
 & & & & & 1 & & 0 & & & & & & 3 & 0 & 0 & & & & 0 & 4 \\
 & & & & & & 1 & 0 & & & & & & 4 & 0 & 0 & & & & 0 & 0 \\
\hline
 & & & & & & & & 1 & & & & & 1 & 2 & 2 & & & & 3 & 6 \\
 & & & & & & & & & 1 & & & & 6 & 0 & 2 & & & & 4 & 1 \\
 & & & & & & & & & & 1 & & & 1 & 0 & 0 & & & & 0 & 6 \\
 & & & & & & & & & & & 1 & & 6 & 0 & 0 & & & & 6 & 1 \\
 & & & & & & & & & & & & 1 & 1 & 0 & 0 & & & & 0 & 6 \\
\hline
 & & & & & & & & & & & & & & & & 1 & & & 6 & 0 \\
 & & & & & & & & & & & & & & & & & 1 & & 6 & 0 \\
 & & & & & & & & & & & & & & & & & & 1 & 6 & 0
\end{array}\right].$$

The ϕ_H-linearized rows of the Hermite form correspond to the last nonzero row in each horizontal slice, namely rows 7, 12 and 15. Indeed,

$$\phi_H^{-1}\left[\begin{array}{c|ccc|ccccc}
1 & 0 & & 4 & 0 & 0 & & 0 & 0 \\
\hline
 & & 1 & 1 & 0 & 0 & & 0 & 6 \\
\hline
 & & & & & & 1 & 6 & 0
\end{array}\right] = \begin{bmatrix} 1 & 4 & 0 \\ & \partial + 1 & 6 \\ & & \partial^2 + 6\partial \end{bmatrix}$$

is the Hermite form of A.

Computing the reduced row echelon form of A_{lin} is equivalent to trying to find a nonzero entry in every column to use in annihilating what is above and underneath. Now let us remember that the first $nd+1$ columns in A_{lin} represent the first column in A and all the derivatives of that column. So trying to find a nonzero entry in every column within that range (the first $nd+1$ column) means that we are trying to find the smallest degree polynomial in $\mathsf{F}[\partial; \sigma, \delta]$ which can be used in annihilating all other entries in the first column of A (i.e., computing the (left) greatest common divisor of the first column of A). The same thing applies to the rest of the matrix.

Theorem 21. *Let $A \in \mathsf{F}[\partial; \sigma, \delta]^{n \times n}$ be non-singular with* $\deg A \leqslant d$*. We can compute the Hermite form of A in $\mathcal{O}(n^{2\omega} d^{\omega})$ operations in* F.

Proof. We compute the row reduced echelon form of (4.3) using the Gauss transform [Storjohann, 2000, Section 2.3] which costs $\mathcal{O}(n^{2\omega} d^{\omega})$ operations in F. □

In the next chapter we give bounds on the degree of entries and coefficients when $\mathsf{F} = \mathsf{k}(z)$ with k being a field. In addition, we give some analysis on coefficient size when $\mathsf{k} = \mathbb{Q}$ for the differential and the shift Ore rings.

Chapter 5

Computing the Popov form of a matrix of Ore polynomials via linearization

This chapter develops algorithms to deterministically compute the Popov form of an input matrix of Ore polynomials. Section 5.1 defines a linearization of an input matrix $A \in \mathsf{F}[\partial; \sigma, \delta]^{n \times n}$ to a matrix $A_{\mathsf{lin}} \in \mathsf{F}^{(n^2 d + n - \sum \mathrm{rdeg}(A)) \times (n^2 d + n)}$ such that the reduced row echelon form of A_{lin} will reveal the Popov form of A. We prove the overall correctness of the approach, then we establish some structural properties of A_{lin}. In Sections 5.2 and 5.3, we show how we can exploit those structural properties of A_{lin} to get a fast algorithm for Popov form computation. In Section 5.4 we conclude this chapter with some concrete complexity analysis for the differential and the shift case when $\mathsf{F} = \mathbb{Q}(z)$.

Since the existence and uniqueness of the Popov form of an input matrix of Ore polynomials has been shown in previous work [Cheng and Labahn, 2007, P. Davies, 2008], we omit showing that in this thesis.

For the rest of this chapter, and as we did in Section 2.2, we define the *pivot* of a vector $\vec{v} \in \mathsf{F}[\partial;\sigma,\delta]^{1\times n}$, denoted $\operatorname{piv}(\vec{v})$, as right-most entry of $\vec{v}$ which has $\deg \vec{v}$.

5.1 Popov form via linearization

In this section we apply a linearization technique to transform an input matrix with Ore polynomial entries to a big linearized matrix over the field F. We show the unique structure of our linearized matrix, then we discuss the different properties of the linearized matrix.

Define the linearization $\phi_P : \mathsf{F}[\partial;\sigma,\delta]^{*\times n} \mapsto \mathsf{F}^{*\times(n^2d+n)}$ by

$$\begin{aligned}\phi_P(\vec{v}) &= \phi_P \left[\begin{array}{ccc} v_1 & \cdots & v_n \end{array}\right] \\ &= \left[\begin{array}{ccc|c|ccc} [v_n]_{nd} & \cdots & [v_1]_{nd} & \cdots & [v_n]_0 & \cdots & [v_1]_0 \end{array}\right] \in \mathsf{F}^{*\times(n^2d+n)},\end{aligned}$$

where $[v_i]_k$ denotes the coefficient of ∂^k of $v_i \in \mathsf{F}[\partial;\sigma,\delta]^{*\times 1}$. Note that output of ϕ_P is simply a permutation of the ϕ_H in Section 4. Using the same example as in (4.1),

$$\phi_P(\left[\begin{array}{c|c} \partial^2+3\partial+6 & 2\partial+5 \end{array}\right]) = \left[\begin{array}{cc|cc|cc|cc|cc} 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 & 5 & 6 \end{array}\right].$$

Let now A_{lin} be given as the ϕ_P-image of the vectors

$$\partial^j \operatorname{row}_i(A) \qquad \text{for } i = 1,\ldots,n \text{ and } j = 0,\ldots,nd - \deg \operatorname{row}_i(A)\ ,$$

ordered by descending degrees and breaking ties by the i index. In other words, consider for $t = 0,\ldots,nd$ the matrix

$$\hat{B}_t := \operatorname{diag}(\partial^{t-\deg \operatorname{row}_1(A)} \operatorname{row}_1(A),\ldots,\partial^{t-\deg \operatorname{row}_n(A)} \operatorname{row}_n(A)).$$

Each row of $\hat{B}_t$ has degree exactly t, but some elements now have negative-degree terms. Let $B_t \in \mathsf{F}^{*\times(n^2d+n)}$ be given as the ϕ_P-image of the rows of $\hat{B}_t$ which have no negative-degree terms. Then A_{lin} consists of putting the B_t on top of each other. More precisely, we can write A_{lin} uniquely in block upper triangular form as

$$A_{\mathsf{lin}} = \left[\begin{array}{c} B_{nd} \\ \hline B_{nd-1} \\ \hline \vdots \\ \hline B_0 \end{array}\right] = \begin{bmatrix} C_{nd} & * & \cdots & * \\ & C_{nd-1} & \cdots & * \\ & & \ddots & \vdots \\ & & & C_0 \end{bmatrix}, \tag{5.1}$$

where each $C_* \in \mathsf{F}^{* \times n}$ has no zero rows. Note that the rowspace of A_{lin} is in one-to-one correspondence, through ϕ_P, with the set

$$\left\{ \sum_{i=1}^{n} u_i \operatorname{row}_i A \;\middle|\; u_i \in \mathsf{F}[\partial; \sigma, \delta],\ \deg u_i \leqslant nd - \deg \operatorname{row}_i A \right\}. \tag{5.2}$$

Lemma 22. *If A is non-singular, then A_{lin} has full row rank and row dimension $n^2d + n - \sum \operatorname{rdeg} A$. For $d \leqslant t \leqslant nd$, then B_t (and C_t) has exactly n rows.*

Proof. A_{lin} has full row rank, since any F-linear relation between rows of A_{lin} maps to an $\mathsf{F}[\partial; \sigma, \delta]$-linear relation between rows of A through ϕ_P and A is non-singular. The i'th row of A is represented in exactly $nd - \deg \operatorname{row}_i(A) + 1$ of the B_t, so the row dimension of A_{lin} becomes as claimed. This also shows that B_t has n rows when $t \geq d$ since every row of A is represented. $\square$

Recall that the pivot of a vector $\vec{v} \in \mathsf{F}^{1 \times (n^2d+n)}$ is the index of the left-most non-zero element of $\vec{v}$. Define $\eta : \{1, \ldots, n\} \times \mathbb{Z}_{\geq 0} \to \{1, \ldots, n^2d + n\}$ as the map between (pivot, degree) of $\mathsf{F}[\partial; \sigma, \delta]^{1 \times n}$ vectors and the pivot in $\mathsf{F}^{1 \times (n^2d+n)}$ vectors induced by ϕ_P, that is,

$$\eta(i, d') = n(nd - d') + n + 1 - i\,.$$

For a vector $\vec{v} \in \mathsf{F}^{1 \times (n^2d+n)}$ we say that the *P-pivot* and *P-degree* of $\vec{v}$ are the first and second components of the 2-tuple $\eta^{-1}(l)$, where l is $\vec{v}$'s pivot.

Now let R_{lin} be the reduced row echelon form of A_{lin}. Then R_{lin} can also be written uniquely in block upper triangular form as

$$R_{\mathsf{lin}} = \begin{bmatrix} T_{nd} & * & \cdots & * \\ & T_{nd-1} & \cdots & * \\ & & \ddots & \vdots \\ & & & T_0 \end{bmatrix}, \tag{5.3}$$

where each $T_* \in \mathsf{F}^{* \times n}$ has no zero rows. Note that those rows in R_{lin} with P-degree t are contained in the submatrix of R_{lin} occupied by T_t. Because R_{lin} is in echelon form, for any given degree t and pivot i, there is at most one row in R_{lin} with P-degree t and P-pivot i, and any row in the row space of R_{lin} with P-degree t and P-pivot i will be a linear combination of this row, and possibly rows below it.

Theorem 23. *Let A be non-singular, and let R_{lin} be the reduced row echelon form of A_{lin}. Then the Popov form P of A is the matrix whose i'th row is the ϕ_P^{-1}-image of the row of R_{lin} with minimal P-degree having P-pivot i.*

Proof. The unique unimodular matrix $U \in \mathsf{F}[\partial; \sigma, \delta]^{n \times n}$ with $UA = P$ has $\deg U \leqslant (n-1)d$ by Lemma 10, and therefore the ϕ_P-linearized rows of P are contained in the row space of R_{lin}. We will prove that they in fact appear directly as rows of R_{lin}. By the minimality of the row degrees of the Popov form, item 2 of Theorem 9, the rows chosen as in the theorem must therefore be exactly those rows of R_{lin}.

So for $1 \leqslant i \leqslant n$, consider the i'th row $\vec{p}$ of P, which has pivot i. Since $\phi_P(\vec{p})$ is in the row space of R_{lin}, there must be exactly one row $\vec{r}_k$ of R_{lin} with the same pivot, with row index k. If $\vec{w}$ is the unique vector over F satisfying $\vec{w}R_{\mathsf{lin}} = \phi_P(\vec{p})$ then clearly $w_k = 1$ and $w_j = 0$, for some $j < k$. We claim $w_j = 0$ also for $j > k$, in which case $\vec{r}_k = \phi_P(\vec{p})$ as we wanted to prove. Suppose, to arrive at a contradiction, that $w_j \neq 0$ for some $j > k$, and let $\vec{r}_j$ be the j'th row of R_{lin}. Since all other rows of R_{lin} are zero at the pivot position of $\vec{r}_j$, that means $\deg p_{i,j'} \geq d'$, where j', d' are the P-pivot respectively P-degree of $\vec{r}_j$. On the other hand, since the $\phi_P^{-1}(\vec{r}_j)$ is in the row space of A and has pivot j', the minimality of the degrees of the Popov form implies $d' \geq \deg p_{j',j'}$. But then $\deg p_{i,j'} \geq \deg p_{j',j'}$, which contradicts that P is in Popov form. We conclude that $w_j = 0$ for $j > k$, and hence $\phi_P(\vec{p}) = \vec{r}_k$. $\square$

Example 24. *For clarity, we exemplify the approach with a usual polynomial ring, i.e. $\sigma = \mathsf{id}$ and $\delta = 0$. Consider the input $A \in \mathsf{F}[\partial; \iota]^{3\times 3}$ from Example 20, $\mathsf{F} = \mathbb{Z}_7$. Then*

$$A_{\mathsf{lin}} = \phi_P \begin{bmatrix} \partial^4 A \\ \vdots \\ \partial A \\ A \end{bmatrix} = \left[\begin{array}{ccccccccccccccccccccc}
6&5&3&1&3&6&0&3&6&&&&&&&&&&&& \\
5&6&5&2&1&0&6&0&5&&&&&&&&&&&& \\
5&0&0&2&4&3&1&5&5&&&&&&&&&&&& \\
&&&6&5&3&1&3&6&0&3&6&&&&&&&&& \\
&&&5&6&5&2&1&0&6&0&5&&&&&&&&& \\
&&&5&0&0&2&4&3&1&5&5&&&&&&&&& \\
&&&&&&6&5&3&1&3&6&0&3&6&&&&&& \\
&&&&&&5&6&5&2&1&0&6&0&5&&&&&& \\
&&&&&&5&0&0&2&4&3&1&5&5&&&&&& \\
&&&&&&&&&6&5&3&1&3&6&0&3&6&&& \\
&&&&&&&&&5&6&5&2&1&0&6&0&5&&& \\
&&&&&&&&&5&0&0&2&4&3&1&5&5&&& \\
&&&&&&&&&&&&6&5&3&1&3&6&0&3&6 \\
&&&&&&&&&&&&5&6&5&2&1&0&6&0&5 \\
&&&&&&&&&&&&5&0&0&2&4&3&1&5&5
\end{array}\right] \in \mathsf{F}^{15\times 21}. \tag{5.4}$$

The row reduced echelon form of A_{lin} is

$$R_{\mathsf{lin}} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc|ccc}
1&&&&&6&&&6&&&&&&&2&&&0&&0 \\
&1&2&&&4&&&6&&&&&&&3&&&6&&5 \\
\hline
&&&1&&0&&&6&&&&&&&2&&&4&&1 \\
&&&&1&2&&&4&&&&&&&2&&&5&&3 \\
\hline
&&&&&&1&&0&&&&&&&6&&&0&&0 \\
&&&&&&&1&2&&&&&&&0&&&0&&0 \\
\hline
&&&&&&&&&1&&&&&&6&&&0&&0 \\
&&&&&&&&&&1&&&&&0&&&6&&5 \\
&&&&&&&&&&&1&&&&0&&&4&&1 \\
\hline
&&&&&&&&&&&&1&&&6&&&0&&0 \\
&&&&&&&&&&&&&1&&6&&&1&&2 \\
&&&&&&&&&&&&&&1&4&&&3&&6 \\
\hline
&&&&&&&&&&&&&&&&1&&6&&5 \\
&&&&&&&&&&&&&&&&&1&4&&1 \\
\hline
&&&&&&&&&&&&&&&&&&&1&2
\end{array}\right].$$

We pick out the rows of R_{lin} with minimal degree having P-pivot 1,2, and 3, respectively.

The Popov form of A is thus

$$\phi_P^{-1}\left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc|ccc} & & & & & & & & & & & & & & & & & 1 & 4 & & 1 \\ & & & & & & & & & & & & & & & & & & & 1 & 2 \\ & & & & & & & & & & & & 1 & & & 6 & & & 0 & & 0 \end{array}\right] = \begin{bmatrix} \partial+1 & 0 & 4 \\ 2 & 1 & 0 \\ 0 & 0 & \partial^2+6\partial \end{bmatrix} \in \mathsf{F}[\partial;\imath]^{3\times 3}.$$

We now consider some structural properties of A_{lin} and R_{lin}. Recall that we have written A_{lin} as a block upper triangular matrix

$$A_{\mathsf{lin}} = \left[\begin{array}{ccc|ccc} C_{nd} & \cdots & * & * & \cdots & * \\ & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & C_{k+1} & * & \cdots & * \\ \hline & & & C_k & \cdots & * \\ & & & & \ddots & \vdots \\ & & & & & C_0 \end{array}\right]$$

with each C_* of column dimension n and with no zero rows. Let $\{a_1, a_2, \ldots, a_n\}$ be the multi-set of row degrees of A. The following lemma follows from the definition of A_{lin}.

Lemma 25. *For $k = 0, 1, \ldots, nd$, the trailing submatrix*

$$\begin{bmatrix} C_k & \cdots & * \\ & \ddots & \vdots \\ & & C_0 \end{bmatrix}$$

of A_{lin} has row dimension $(k+1)n - \sum_{i=1}^{n} \min(a_i, k+1)$.

We have also written R_{lin} in a block upper triangular form as

$$R_{\mathsf{lin}} = \left[\begin{array}{ccc|ccc} T_{nd} & \cdots & * & * & \cdots & * \\ & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & T_{k+1} & * & \cdots & * \\ \hline & & & T_k & \cdots & * \\ & & & & \ddots & \vdots \\ & & & & & T_0 \end{array}\right] \tag{5.5}$$

where each $T_* \in \mathsf{F}^{*\times n}$ has no zero rows. Let $\{p_1, p_2, \ldots, p_n\}$ be the multiset of row degrees in the Popov form of A. The following lemma follows as a corollary of Theorem 23.

Lemma 26. *For $k = 0, 1, \ldots, nd$, the trailing submatrix*

$$\begin{bmatrix} T_k & \cdots & * \\ & \ddots & \vdots \\ & & T_0 \end{bmatrix} \tag{5.6}$$

of R_{lin} has row dimension at most $(k+1)n - \sum_{i=1}^{n} \min(p_i, k+1)$.

For $k = 0, 1, \ldots, nd$, define OD_k to be the nullity (row dimension of the left nullspace) of the principal submatrix

$$\begin{bmatrix} C_{nd} & \cdots & * \\ & \ddots & \vdots \\ & & C_{k+1} \end{bmatrix} \tag{5.7}$$

of A_{lin}. Recall that $\mathsf{OD}(A) := \sum \mathrm{rdeg} A - \sum \mathrm{rdeg} P$.

Theorem 27. *For $k = 0, 1, \ldots, nd$ we have $\mathsf{OD}_k \leqslant \mathsf{OD}(A)$.*

Proof. Let P be a permutation, and U be a unit lower non-singular matrix over F that such that premultiplying (5.7) by UP transforms it to echelon form

$$\left[\begin{array}{c} R_{k+1} \\ \hline \\ \end{array}\right]$$

with OD_k zero rows. Then applying $\mathrm{diag}(U, I)$ to A_{lin} yields

$$\left[\begin{array}{c|c} U & \\ \hline & I \end{array}\right] \left[\begin{array}{ccc|ccc} C_{nd} & \cdots & * & * & \cdots & * \\ & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & C_{k+1} & * & \cdots & * \\ \hline & & & C_k & \cdots & * \\ & & & & \ddots & \vdots \\ & & & & & C_0 \end{array}\right] = \left[\begin{array}{c|ccc} R_{k+1} & * & \cdots & * \\ & E_k & \cdots & * \\ \hline & C_k & \cdots & * \\ & & \ddots & \vdots \\ & & & C_0 \end{array}\right], \tag{5.8}$$

where $E_k \in \mathsf{F}^{\mathsf{OD}_k \times n}$. Considering that A_{lin} has full row rank, the row dimension of the submatrix

$$\begin{bmatrix} E_k & \cdots & * \\ C_k & \cdots & * \\ & \ddots & \vdots \\ & & C_0 \end{bmatrix} \tag{5.9}$$

of the matrix on the right of (5.8) will be equal to the row dimension of the trailing submatrix (5.6) of R_{lin}. Lemmas 25 and 26 now give

$$\begin{aligned} \mathsf{OD}_k &\leqslant \sum_{i=1}^{n} \min(a_i, k+1) - \sum_{i=1}^{n} \min(p_i, k+1) \\ &= \sum_{i=1}^{n} (\min(a_i, k+1) - \min(p_i, k+1)). \end{aligned}$$

Assume now that $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_n$ and $p_1 \leqslant p_2 \leqslant \cdots p_n$. Then $a_i - p_i \geq 0$ for $i = 1, 2, \ldots, n$ by Item 2 of Theorem 9, and

$$\min(a_i, k+1) - \min(p_i, k+1) \begin{cases} = & a_i - p_i & \text{if } a_i \leqslant k+1 \\ = & 0 & \text{if } a_i > k+1 \text{ an } p_i \geq k+1 \\ < & a_i - p_i & \text{if } a_i > k+1 \text{ and } p_i < k+1 \end{cases}.$$

Thus $\min(a_i, k+1) - \min(p_i, k+1) \leqslant a_i - p_i$ in all cases, establishing the result.

□

5.2 Block elimination of the linearized system

Let $A \in \mathsf{F}[\partial; \sigma, \delta]^{n \times n}$ be non-singular with $\deg A \leqslant d$. In this section we show how to perform a structured Gaussian elimination of the linearized system A_{lin} over F. We first consider in Section 5.2.1 the problem of transforming A to Popov form when A is already row reduced, the so called normalization problem [Sarkar and Storjohann, 2011]. Then we consider the general case in Section 5.2.2.

5.2.1 Normalization of an already row reduced matrix

We can detect if A is row reduced by testing its leading coefficient matrix for non-singularity. Suppose A is already row reduced. Let U be the unique matrix such that $UA = P$ is in Popov form. By the predictable degree property [Beckermann et al., 2006, Lemma A.1 (a)] then $\deg \mathrm{col}_i U \leqslant d - \deg \mathrm{row}_i A$, $1 \leqslant i \leqslant n$. Consider the following submatrix of A_{lin} (5.1) comprised of the last $n(d+1) - \sum \mathrm{rdeg} A$ rows:

$$\bar{A}_{\mathsf{lin}} = \begin{bmatrix} B_d \\ \vdots \\ B_0 \end{bmatrix} = \left[\begin{array}{c|ccc} \quad\quad\quad\quad & C_d & \cdots & * \\ & & \ddots & \vdots \\ & & & C_0 \end{array} \right].$$

Note that the rowspace of $\bar{A}_{\mathsf{lin}}$ is in one-to-one correspondence, through ϕ_P, with the set

$$\left\{ \sum_{i=1}^{n} u_i \,\mathrm{row}_i A \;\middle|\; u_i \in \mathsf{F}[\partial; \sigma, \delta],\ \deg u_i \leqslant d - \deg \mathrm{row}_i A \right\}.$$

As a corollary of Lemma 22 we have that $\bar{A}_{\mathsf{lin}}$ has full row rank $n(d+1) - \sum \mathrm{rdeg} A$. The next theorem is a corollary of Theorem 23.

Theorem 28. *Let A be non-singular and row reduced, and let $\bar{R}_{\mathsf{lin}}$ be the reduced row echelon form of $\bar{A}_{\mathsf{lin}}$. Then the Popov form P of A is the matrix S whose i'th row is the ϕ_P^{-1}-image of the row of $\bar{R}_{\mathsf{lin}}$ with minimal degree having P-pivot i.*

Because A is row reduced, by Lemma 12 we have $\mathsf{OD}(A) = 0$, so by Theorem 27 each block C_* in $\bar{A}_{\mathsf{lin}}$ will have full row rank. Since the right block of $\bar{A}_{\mathsf{lin}}$ has column dimension $n(d+1)$, performing standard Gauss Jordan elimination would cost $O((nd)^3)$ operations from F to produce $\bar{R}_{\mathsf{lin}}$ in its entirety. We can save a factor of d by avoiding the complete computation of $\bar{R}_{\mathsf{lin}}$. Instead, first compute an echelon form of $\bar{A}_{\mathsf{lin}}$ by applying Gaussian elimination to each full rank slice B_*. Gaussian elimination of a single B_* has cost $O(n^3 d)$, yielding a total cost for all slices of $O(n^3 d^2)$ operations in F. Then use back substitution (as it will be shown in Example 30) to reduce the n rows whose ϕ_P^{-1}-image has minimal degree and P-pivot i, $1 \leqslant i \leqslant n$. This costs an additional $O(n^3d^2)$. Finally, scale these n rows so their pivots are equal to one. We obtain the following result.

Theorem 29. *Let $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be non-singular and row reduced, with $\deg A \leqslant d$. The Popov form of A can be computed within the following cost:*

- *Computing $\partial^k \operatorname{row}_i A$ for $1 \leqslant k \leqslant d - \deg \operatorname{row}_i A$, $1 \leqslant i \leqslant n$.*
- *An additional $O(n^3d^2)$ operations from F.*

5.2.2 General case

Now assume that A is not already row reduced, so that $\mathsf{OD} := \mathsf{OD}(A) > 0$. The key observation is that since $\deg P \leqslant d$, the ϕ_P-linearization of the rows of P will be contained in the row space of the trailing submatrix (5.6) of R_{lin} for $k = d$. This implies that the rows of R_{lin} occupied by $T_{nd}, T_{nd-1}, \cdots, T_{d+1}$ are not required.

Our algorithm for performing the elimination of A_{lin} has three phases. The first phase computes the matrix (5.9) for $k = d$, whose rowspace is equal to that of (5.6) for $k = d$. The second phase transforms this matrix to row echelon form. The third phase performs back substitution to reduce the n rows whose ϕ_P^{-1}-image has minimal degree and P-pivot i, $1 \leqslant i \leqslant n$.

Our main computation tool is the Gauss transform [Storjohann, 2000, Section 2.3]. Given as input a matrix

$$\begin{bmatrix} E_k \\ C_k \end{bmatrix} \in \mathsf{F}^{O(\mathsf{OD}+n)\times n}, \tag{5.10}$$

the so called Gauss transform algorithm [Storjohann, 2000, Algorithm 2.14] can be used to produce a permutation matrix P_k and unit lower triangular matrix U_k such that

$$\overbrace{\left[\begin{array}{c|c} F_k & \\ \hline N_k & I \end{array}\right]}^{U_k} P_k \left[\begin{array}{c} E_k \\ C_k \end{array}\right] = \left[\begin{array}{c} G_k \\ \hline \\ \end{array}\right],$$

with $\left[\begin{array}{c|c} N_k & I \end{array}\right] P_k$ and G_k are the left nullspace basis and a row echelon form, respectively, of the input matrix (5.10).

Phase 1: For convenience, let E_{nd} be the $0 \times n$ matrix. We will compute a Gauss transform as described above for $k = nd, nd-1, \ldots, d+1$. At the start of stage k we are exactly in the situation shown in (5.8). The key observation is that no entries in the rows occupied by R are required, and so the computation of these rows can be avoided. To go from stage $k+1$ to k we can thus apply only the nullspace to the next slice and obtain

$$\left[\begin{array}{c|c} N_k & I \end{array}\right] P_k \left[\begin{array}{c|ccc} E_k & * & \cdots & * \\ C_k & * & \cdots & * \end{array}\right] = \left[\begin{array}{c|ccc} & E_{k-1} & \cdots & * \end{array}\right].$$

Continue this for $k = nd, nd-1, \ldots, d+1$.

Phase 2: For $k = d, d-1, \ldots, 0$, we apply the complete Gauss transform to the work matrix:

$$U_k P_k \left[\begin{array}{c|ccc} E_k & * & \cdots & * \\ C_k & * & \cdots & * \end{array}\right] = \left[\begin{array}{c|ccc} G_k & * & \cdots & * \\ & E_{k-1} & \cdots & * \end{array}\right].$$

Repeating this for $k = d, d-1, \ldots, 0$, we have computed the row echelon form

$$G = \left[\begin{array}{ccc} G_d & \cdots & * \\ & \ddots & \vdots \\ & & G_0 \end{array}\right] \in \mathsf{F}^{* \times n(d+1)}.$$

Phase 3: Identify for $i = n, n-1, \ldots, 1$, the row in G whose ϕ_P^{-1}-image has minimal degree and P-pivot i, and use back substitution to zero out the entries in this row which are above a pivot, similar to how we proceeded in Section 5.2.1. Finally, scale these n rows to make their pivots equal to one.

Example 30. *Consider the input matrix*

$$A = \begin{bmatrix} 7\partial^2+3\partial+8 & 9\partial^2+7\partial+4 & \partial^2+2\partial+2 \\ 3\partial^2+4 & 7\partial^2+6\partial+8 & 5\partial^2+10\partial \\ 3\partial^2+2\partial+5 & 7\partial^2+5\partial+1 & 4\partial^2+8\partial+5 \end{bmatrix} \in \mathbb{Z}_{11}[\partial;\prime]^{3\times 3}.$$

Then

$$\begin{aligned} A_{\mathsf{lin}} &= \begin{bmatrix} C_6 & * & * & * & * & * & * \\ & C_5 & * & * & * & * & * \\ & & C_4 & * & * & * & * \\ & & & C_3 & * & * & * \\ & & & & C_2 & * & * \end{bmatrix} \\ &= \left[\begin{array}{ccccccccccccccccccccc} 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & & & & & & & & & & \\ 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & & & & & & & & & & \\ 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & & & & & & & & & & \\ & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & & & & & & & \\ & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & & & & & & & \\ & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & & & & & & & \\ & & & & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & & & & \\ & & & & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & & & & \\ & & & & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & & & & \\ & & & & & & & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & \\ & & & & & & & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & \\ & & & & & & & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & \\ & & & & & & & & & & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 \\ & & & & & & & & & & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 \\ & & & & & & & & & & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 \end{array}\right]. \end{aligned}$$

For phase 1, step $k = 6$, we compute and apply the nullspace of C_6 to obtain

$$\left[\begin{array}{cccccc} E_5 & * & * & * & * & * \\ \hline C_5 & * & * & * & * & * \\ & C_4 & * & * & * & * \\ & & C_3 & * & * & * \\ & & & C_2 & * & * \end{array}\right] = \left[\begin{array}{ccccccccccccccccccccc} 0 & 0 & 0 & 0 & 4 & 8 & & & & & & & & & & & & & & & \\ \hline 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & & & & & & & & & & \\ 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & & & & & & & & & & \\ 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & & & & & & & & & & \\ & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & & & & & & & \\ & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & & & & & & & \\ & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & & & & & & & \\ & & & & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & & & & \\ & & & & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & & & & \\ & & & & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & & & & \\ & & & & & & & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & \\ & & & & & & & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & \\ & & & & & & & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & \end{array}\right].$$

For phase 1*, step* $k = 5$*, we compute and apply the nullspace of the* 4×3 *matrix occupied by* E_5 *and* C_5 *to obtain*

$$\left[\begin{array}{ccccc} E_4 & * & * & * & * \\ \hline C_4 & * & * & * & * \\ & C_3 & * & * & * \\ & & C_2 & * & * \end{array}\right] = \left[\begin{array}{ccccccccccccccc} 0 & 4 & 8 & & & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 4 & 8 & & & & & & & & & \\ \hline 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & & & & \\ 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & & & & \\ 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & & & & \\ & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 & & & \\ & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 & & & \\ & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 & & & \\ & & & & & & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 \\ & & & & & & 5 & 7 & 3 & 10 & 6 & 0 & 0 & 8 & 4 \\ & & & & & & 4 & 7 & 3 & 8 & 5 & 2 & 5 & 1 & 5 \end{array}\right].$$

We continue phase 1 *for steps* $k = 4, 3$ *applying only the nullspace at every step. For phase* 2*, steps* $k = 2, 1, 0$*, we apply the entire Gauss transforms, yielding the echelon form*

$$\left[\begin{array}{c|c|c|c} & G_2 & * & * \\ \hline & & G_1 & * \\ \hline & & & G_0 \end{array}\right] = \left[\begin{array}{c|ccc|ccc|ccc} & 1 & 9 & 7 & 2 & 7 & 3 & 2 & 4 & 8 \\ & & 1 & 10 & 3 & 8 & 2 & 0 & 0 & 0 \\ & & & 1 & 10 & 1 & 3 & 0 & 0 & 0 \\ \hline & & & & & 1 & 2 & 0 & 0 & 0 \\ & & & & & & 1 & 10 & 1 & 3 \\ \hline & & & & & & & & 1 & 2 \end{array}\right].$$

For phase 3, we identify the Popov rows (rows 1*,* 5 *and* 6 *in this example) and then do back substitution,*

$$\left[\begin{array}{c|ccc|ccc|ccc} & 1 & & & 2 & & & 2 & & 0 \\ & & 1 & 10 & 3 & 8 & 2 & 0 & 0 & 0 \\ & & & 1 & 10 & 1 & 3 & 0 & 0 & 0 \\ \hline & & & & & 1 & 2 & 0 & 0 & 0 \\ & & & & & & 1 & 10 & & 1 \\ \hline & & & & & & & & 1 & 2 \end{array}\right].$$

The Popov form of A *is thus*

$$\phi_P^{-1} \left[\begin{array}{c|ccc|ccc|ccc} & & & & & & 1 & 10 & & 1 \\ & & & & & & & & 1 & 2 \\ & 1 & & & 2 & & & 2 & & 0 \end{array}\right] = \begin{bmatrix} \partial + 1 & 0 & 10 \\ 2 & 1 & 0 \\ 0 & 0 & \partial^2 + 2\partial + 2 \end{bmatrix}.$$

The next theorem gives a cost analysis of the algorithm just described in terms of operations from F. The theorem gives three cost estimates. First, we give an unconditional cost estimate based only on the input parameters n and d. Second, we give a refined cost estimate in terms of OD. Third, we consider the case of special Ore rings (such as the shift case) for which A_{lin} matrix may have the shape shown in Example 24 , that is, with a large block upper triangular submatrix of zeroes in the northeast corner: the cost estimates are improved by a factor of n in this case.

Theorem 31. *Let $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be non-singular with $\deg A \leqslant d$. The Popov form of A can be computed within the following costs.*

1. *General case:*
   - *Computing $\partial^k \operatorname{row}_i A$ for $1 \leqslant k \leqslant nd - \deg \operatorname{row}_i A$, $1 \leqslant i \leqslant n$.*
   - *Additional $O(n^{\omega+2}d^3)$ field operations from F.*
2. *A more refined cost is obtained by considering the parameter OD. Assume that A is not already row reduced, so that $\mathsf{OD} := \mathsf{OD}(A) > 0$. Then the number of additional operations is reduced to:*
   - *$O(\mathsf{OD}^{\omega-2}n^4d^2)$ if $\mathsf{OD} < n$*
   - *$O(\mathsf{OD}\, n^{\omega+1}d^2)$ if $\mathsf{OD} \geq n$*
3. *Finally, suppose that the Ore ring $\mathsf{F}[\partial;\sigma,\delta]$ has the property that for any nonzero element $f \in \mathsf{F}[\partial;\sigma,\delta]$, the trailing degree of ∂f is at least one more than the trailing degree of f. Then the Big Oh estimates in parts 1 and 2 above for the additional operations are reduced by a factor of n.*

Proof. We first establish part 2 of the theorem. By Theorem 27, the row dimension of each nullspace N_* is bounded by OD. Instead of considering the three phases separately, we will partition the computational work done as follows. The nullspace N_k is applied for all k, $0 \leqslant k \leqslant nd$, but the unit lower triangular block F_k is applied only for $0 \leqslant k \leqslant d$. Also note that for $k \leqslant d$, the column dimension of the slice to which F_k is being applied to is bounded by $(d+1)n$.

The application of the permutations P_* does not dominate the cost. We can thus partition the computational work as follows.

A Gauss transform: at most $nd + 1$ times, compute a Gauss transform of a matrix bounded in dimension $O(\mathsf{OD} + n) \times n$.

B Nullspace application: at most $nd + 1$ times, multiply an $O(\mathsf{OD}) \times n$ matrix by an $n \times O(n^2 d)$ matrix.

C Computing the echelon form: at most $d + 1$ times, multiply an $O(n) \times O(n)$ matrix by a $O(n) \times O(nd)$ matrix.

D Back substitution: $O(n^3 d^2)$ operations.

Since the rank of the input matrix (5.10) is bounded by its column dimension n, the cost of computing (U_k, P_k) for a given k is bounded by $O((\mathsf{OD} + n)n^{\omega-1})$ by [Storjohann, 2000, Proposition 2.15]. This gives a total cost for (A) of $O((\mathsf{OD} + n)n^{\omega} d)$. Using an obvious block decomposition shows (C) can be done in time $O(n^{\omega} d^2)$.

It remains to bound the cost of (B). There are two cases, depending on whether $\mathsf{OD} \leqslant n$ or $\mathsf{OD} > n$. Using an obvious block decomposition shows a single nullspace application has cost $O(\mathsf{OD}^{\omega-2} n^3 d)$ if $\mathsf{OD} \leqslant n$ and $O(\mathsf{OD}\, n^{\omega} d)$ otherwise. The total cost for (B) is thus $O(\mathsf{OD}^{\omega-2} n^4 d^2)$ if $\mathsf{OD} \leqslant n$ and $O(\mathsf{OD}\, n^{\omega+1} d^2)$ if $\mathsf{OD} > n$. In both cases these upper bounds for the cost of (B) dominate the cost bounds for (A), (C) and (D).

Part 1 of the theorem follows by substituting the a priori upper bound $\mathsf{OD} \leqslant nd$ into the O-bound in part 2 for the case $\mathsf{OD} \geq n$.

For part 3, note that the (nonzero part) of the slice to which the nullspace is applied will now have dimension $O(nd)$ instead of $O(n^2 d)$. The cost estimates for the work in part (B) are thus reduced by a factor of n, but they still dominate the cost of parts (A), (C) and (D). $\square$

5.3 Fraction-free block elimination

Now consider the case when all entries in A_{lin} are coming from an integral domain, for example $\mathsf{F} = \mathsf{k}(z)$ for a field k but all entries are in $\mathsf{F}[z]$, or even $\mathbb{Z}[z]$ when $\mathsf{k} = \mathbb{Q}$. It is desirable in this setting to keep all intermediate quantities in the computation integral, while at the same time controlling their growth. The classic technology for this purpose in the linear algebra setting is fraction-free Gaussian elimination [Edmonds, 1967, Bareiss, 1968].

The Gauss transform algorithm [Storjohann, 2000, Algorithm 2.14] is actually designed to do fraction-free Gaussian elimination, and because of its column recursive formulation, is well suited to the elimination of A_{lin}.

The incorporation of fraction-free techniques into the algorithm supporting Theorem 31 is straightforward. For $k = nd, nd-1, \ldots, 0$, the fraction-free Gauss transform algorithm also computes Δ_{k+1}, the minor of R_{k+1} in (5.8) comprised of its rank profiles columns. To start the process set $\Delta_{nd+1} = 1$ since R_{nd+1} is the 0×0 matrix, and recall that E_{nd} is the $0 \times n$ matrix. At step k we have the scaled matrix

$$\Delta_{k+1} \left[\begin{array}{c|ccc} E_k & * & \cdots & * \end{array}\right]$$

from the previous step, together with Δ_{k+1}. The rows of A_{lin} occupied by C_k are premultiplied by Δ_{k+1} to form the next slice

$$\Delta_{k+1} \left[\begin{array}{c|ccc} E_k & * & \cdots & * \\ C_k & * & \cdots & * \end{array}\right], \tag{5.11}$$

which will be fraction-free, that is, all entries are minors of A_{lin} of dimension bounded by one plus the rank of R_{k+1}. (We remark that only scaling the rows of A_{lin} that will be involved in the next elimination step is important for the complexity, and similar to [Lee and Saunders, 1995].) At stage k, the fraction-free Gauss transform takes as input (5.11), together with Δ_{k+1}, and returns as output the permutation P_k and the scaled matrix

$$\bar{U}_k = \left[\begin{array}{c|c} \bar{F}_k & \\ \hline \Delta_k N_k & \Delta_k I \end{array}\right], \tag{5.12}$$

together with Δ_k. The matrix $\bar{F}_k$ is equal to the unit lower triangular F_k from before but with each row scaled by a certain minor of A_{lin} which is known a priori to clear any denominators. The output (5.12) is also fraction-free, that is, all entries are minors of A_{lin} of dimension bounded by the rank of R_k. The nullspace applications in Phase 1 can now be done in a fraction-free fashion as

$$\frac{1}{\Delta_{k+1}}\left(\left(\Delta_k \left[\begin{array}{c|c} N_k & I \end{array}\right] P_k\right)\left(\Delta_{k+1}\left[\begin{array}{c|ccc} E_k & * & \cdots & * \\ C_k & * & \cdots & * \end{array}\right]\right)\right)$$

yielding

$$\Delta_k \left[\begin{array}{c|ccc} & E_{k-1} & \cdots & * \end{array}\right].$$

In Phase 2 the entire Gauss transform is applied to obtain

$$\frac{1}{\Delta_{k+1}}\left(\bar{U}_k P_k\left(\Delta_{k+1}\left[\begin{array}{c|ccc} E_k & * & \cdots & * \\ C_k & * & \cdots & * \end{array}\right]\right)\right) = \Delta_k \left[\begin{array}{c|ccc} \bar{G}_k & * & \cdots & * \\ & E_{k-1} & \cdots & * \end{array}\right].$$

The back substitution in Phase 3 can be done iteratively in a fraction-free fashion also [Bareiss, 1968].

Next we give the detailed steps of the fraction-free block elimination algorithm.

```
Procedure FFPopov(A)
    Input: $A \in \mathsf{k}[z][\partial;\sigma,\delta]^{n\times n}$ such that $A$ is non-singular, $d = \deg A$
    Output: $P \in \mathsf{k}(z)[\partial;\sigma,\delta]^{n\times n}$ the Popov form of $A$.
1  Construct $A_{\mathsf{lin}}$ ,   $\mu, \nu \leftarrow Dimensions(A_{\mathsf{lin}})$
2  $i,\ j,\ \check{E}_1,\ \Delta_1,\ n' \leftarrow 1,\ 1,\ 0,\ 1,\ n$
3  while $i \leqslant \mu$ do
4      $n' := min(n, \mu - (i - 1 + \check{E}_1))$ ,   $\psi \leftarrow i - 1 + \check{E}_1 + n'$
5      $A_{\mathsf{lin}}[i + \check{E}_1..\psi, j..\nu] \leftarrow \Delta_1.A_{\mathsf{lin}}[i + \check{E}_1..\psi, j..\nu]$
6      $T, P, r, \Delta_2 :=$ GaussTransform($A_{\mathsf{lin}}[i..\psi, j..j - 1 + n], \Delta_1$)
7      $\check{E}_2 \leftarrow \check{E}_1 + n' - r$ ,  $\tau \leftarrow i - 1 + r + \check{E}_2$
8      Apply the permutation $P$
9      $A_{\mathsf{lin}}[i + r..\tau, j..\nu] := \frac{T[r+1..r+\check{E}_2].A_{\mathsf{lin}}[i..\tau, j..\nu]}{\Delta_1}$
10     if $j > \nu - n(d + 1)$ then
11         $A_{\mathsf{lin}}[i..i - 1 + r, j..\nu] := \frac{T[1..r].A_{\mathsf{lin}}[i..\tau, j..\nu]}{\Delta_1}$
12     $i \leftarrow i + r$ ,  $j \leftarrow j + n$ ,  $\Delta_1 \leftarrow \Delta_2$ ,  $\check{E}_1 \leftarrow \check{E}_2$
13 Identify the rows of $P$ in $A_{\mathsf{lin}}$
14 Let $\varrho \leftarrow \nu - n(d + 1) + 1$,  $\rho :=$ index of first row in $T_d$ in (5.5)
15 Let $lstPiv$ be list of first non-zero entry in $A_{\mathsf{lin}}[\rho..\mu]$
16 for $i$ looping the identified rows of $P$ do
17     for $ii$ from $i + 1$ to $\mu - \rho$ do
18         $A_{\mathsf{lin}}[i] := \frac{A_{\mathsf{lin}}[ii,j] * A_{\mathsf{lin}}[i] - A_{\mathsf{lin}}[i,j] * A_{\mathsf{lin}}[ii]}{lstPiv[ii - \rho]}$      # Back substitution
19 Make the identified Popov rows' leading entry monic
20 Retrieve the rows of $P$ from $A_{\mathsf{lin}}$ through $\phi_P^{-1}$
21 return $P$
```

Before every call to the Gauss transform algorithm at step 6, we multiply the new rows by the appropriate minor at step 5. Step 5 costs $\mathcal{O}(n^3 d)$ operations in F, repeated $\mathcal{O}(nd)$ times, so overall cost of this step is $\mathcal{O}(n^4 d^2)$.

- For $\mathsf{OD} < n$, step 5 does not dominate the cost in Theorem 31 as $\mathsf{OD}^{\omega-2}n^4d^2 > n^4d^2$ when $\mathsf{OD} < n$ (note that $\mathsf{OD} > 0$, see Section 5.2.1 for $\mathsf{OD} = 0$).
- For $\mathsf{OD} \geqslant n$, step 5 does not dominate the cost in Theorem 31 as $\mathsf{OD}\, n^{\omega+1}d^2 > n^4d^2$ when $\mathsf{OD} \geqslant n$ and $\omega > 2$.

Using the fraction-free approach, all intermediate quantities arising during the elimination (i.e., the entries of (5.11) and (5.12)) will thus be minors of A_{lin}. We recall some well known a priori bounds for the size of these minors for some common cases. We will use size and $\overline{\mathsf{size}}$ for the bounds for A_{lin} and $\bar{A}_{\mathsf{lin}}$ respectively.

- $\mathsf{F} = \mathsf{k}[z]$ with $\deg_z A_{\mathsf{lin}}, \deg_z \bar{A}_{\mathsf{lin}} \leqslant e$. Multiplying the row dimension of A_{lin} and $\bar{A}_{\mathsf{lin}}$ by e gives explicit bounds for the degrees $\mathsf{size}_{\mathsf{k}[z]}$ and $\overline{\mathsf{size}}_{\mathsf{k}[z]}$ of minors of A_{lin} and $\bar{A}_{\mathsf{lin}}$ that satisfy
$$\mathsf{size}_{\mathsf{k}[z]} \in O(n^2de) \ \text{ and } \ \overline{\mathsf{size}}_{\mathsf{k}[z]} \in O(nde).$$
- $\mathsf{F} = \mathbb{Z}$ with the magnitude of entries of A_{lin} and $\bar{A}_{\mathsf{lin}}$ bounded by β. Hadamard's inequality [Horn and Johnson, 1985, Corollary 7.82] gives an explicit bound $2^{\mathsf{size}_{\mathbb{Z}}}$ and $2^{\overline{\mathsf{size}}_{\mathbb{Z}}}$ for the magnitudes of minors of A_{lin} and $\bar{A}_{\mathsf{lin}}$ that satisfy
$$\mathsf{size}_{\mathbb{Z}} \in O(n^2d\log(nd\beta)) \ \text{ and } \ \overline{\mathsf{size}}_{\mathbb{Z}} \in O(nd\log(nd\beta)).$$
- $\mathsf{F} = \mathbb{Z}[z]$ with $\deg_z A \leqslant e$, and with the magnitude of integer coefficients of entries of A_{lin} and $\bar{A}_{\mathsf{lin}}$ bounded by β. Multiplying the determinant degree bound above with the logarithm base 2 of an explicit magnitude bound for the coefficients [Goldstein and Graham, 1974] gives
$$\mathsf{size}_{\mathbb{Z}[z]} \in O(n^4d^2e\log(nde\beta)) \ \text{ and } \ \overline{\mathsf{size}}_{\mathbb{Z}[z]} \in O(n^2d^2e\log(nde\beta)).$$

Now let M be a multiplication time for $\mathsf{k}[z]$, that is, two polynomials from $\mathsf{k}[z]$ with degree strictly less than t can be multiplied in $\mathsf{M}(t)$ operations from k. Then over $\mathsf{k}[z]$ a cost estimate in terms of operations from k is obtained by multiplying the algebraic cost estimates of Theorem 31 by $\mathsf{M}(\mathsf{size}_{\mathsf{k}[z]})$. Note that the polynomial multiplication can be done modulo z^p for $p = 2\,\mathsf{size}_{\mathsf{k}[z]} + 1$ to control degrees during the fast matrix multiplications.

If M is a multiplication time for $\mathbb{Z}$, that is, two integers with bit-length bounded by t can be multiplied with $\mathsf{M}(t)$ bit operations, then a cost estimate in terms of bit operations for the cases $\mathbb{Z}$ and $\mathbb{Z}[z]$ are obtained by multiplying the algebraic cost estimates by $\mathsf{M}(\mathsf{size}_{\mathbb{Z}})$ and $\mathsf{M}(\mathsf{size}_{\mathbb{Z}[z]})$. Note that for the $\mathbb{Z}[z]$ case we can use Kronecker substitution [Harvey, 2009] to reduce the integer polynomial multiplication to integer multiplication. Similar to the case $\mathsf{k}[z]$, the multiplication can be done modulo 2^p for an appropriate $p \in O(\mathsf{size}_{\mathbb{Z}[z]})$.

All these cost estimates can be improved (by logarithmic factors) by performing the matrix multiplications using a homomorphic imaging scheme. For example, if $\#\mathsf{k} > 2nd + 1$, then two $n \times n$ matrices over $\mathsf{k}[x]$ with degree bounded by d can be multiplied using only $O(n^{\omega}d + n^2\mathsf{M}(d))$ operations from k [Bostan and Schost, 2005], instead of $O(n^{\omega}\mathsf{M}(d))$. For integer matrix multiplication we refer to [Harvey and van der Hoeven, 2014].

5.4 Cost analysis for some common Ore rings

Let $A \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ be non-singular with degree d. Theorem 31 gave cost estimates for computing the Popov form P of A in terms of operations from F. In this section we give refined cost estimates for some specializations of F, focusing on the differential and shift cases.

First consider the case $\mathsf{F} = \mathsf{k}(z)$. We will assume that A has entries over $\mathsf{k}[z]$. This can be achieved by clearing denominators, if necessary. As in [Giesbrecht and Kim, 2012], we will assume that $\sigma(z) \in \mathsf{k}[z]$ and $\deg_z \delta(z) \leqslant 1$. Then $\partial z = \sigma(z)\partial + \delta(z) \in \mathsf{k}[z][\partial;\sigma,\delta]$ and the degree in z and in ∂ remains unchanged. The linearized systems will thus be over $\mathsf{k}[z]$, allowing application of our fraction-free algorithm.

In the following theorems recall that $\mathsf{OD} = \mathsf{OD}(A) := \sum \mathrm{rdeg} A - \sum \mathrm{rdeg} P$.

Theorem 32. *Let $A \in \mathsf{k}[z][\partial;\sigma,\delta]^{n\times n}$ be non-singular with $\deg A \le d$ and $\deg_z A \leqslant e$. If A is row reduced, this can be detected and the Popov form of A computed in $O(n^3 d^2\, \mathsf{M}(nde))$ operations from k. If A is not row reduced, the Popov form can be computed in*

1. *$O(n^{\omega+2} d^3\, \mathsf{M}(n^2 de))$ field operations from k.*
2. *A more refined cost is obtained by considering the parameter OD. Then the number of operations from k is reduced to:*
   - *$O(\mathsf{OD}^{\omega-2} n^4 d^2\, \mathsf{M}(n^2 de))$ if $\mathsf{OD} < n$*
   - *$O(\mathsf{OD}\, n^{\omega+1} d^2\, \mathsf{M}(n^2 de))$ if $\mathsf{OD} \ge n$*
3. *Finally, suppose that $\mathsf{k}[z][\partial;\sigma,\delta]$ has the property that for any nonzero element $f \in \mathsf{k}[z][\partial;\sigma,\delta]$, the trailing degree of ∂f is at least one more than the trailing degree of f. Then the Big Oh estimates in parts 1 and 2 above are reduced by a factor of n.*

Proof. To test if A is row reduced we can check if its leading coefficient matrix is non-singular. This check will not dominate the cost. The theorem now follows from Theorems 28 and 31 and the estimates for $\overline{\mathsf{size}}_{\mathsf{k}[z]}$ and $\mathsf{size}_{\mathsf{k}[z]}$. $\square$

Now consider the case $\mathsf{F} = \mathbb{Q}(z)$. As before, we will assume that A has entries over $\mathbb{Z}[z]$. Then A has entries polynomials in ∂ whose coefficients are polynomials in $\mathbb{Z}[z]$; let $||A||_\infty$ denote the largest in abolulte value of any (integer) coefficient of any of these $\mathbb{Z}[z]$ coefficients.

As mentioned in Section 5.3, we can use Kronecker substition to reduce arithmetic operations from $\mathbb{Z}[z]$ to integer arithmetic, and we get the following.

Theorem 33. *$A \in \mathbb{Z}[z][\partial;\sigma,\delta]^{n\times n}$ be non-singular with $\deg A \leqslant d$ and $\deg_z A \leqslant e$. Suppose our Ore ring is either the differential polynomials (where $\sigma(z) = z$, $\delta(z) = 1$) or the shift polynomials (where $\sigma(z) = z+1$, $\delta(z) = 0$). If A is row reduced, this can be detected and the Popov form of A computed in $O(n^3d^2)$ operations with integers bounded in length by $O\tilde{}(n^2d^2e(\log ||A||_\infty + e))$ bits. If A is not row reduced, the Popov form can be computed in*

1. *$O(n^{\omega+2}d^3)$ operations with integers bounded in length by $O\tilde{}(n^4d^2e(\log ||A||_\infty + e))$ bits.*
2. *A more refined cost is obtained by considering the parameter OD. Then the number of operations on integers is reduced to:*
   - *$O(\mathsf{OD}^{\omega-2}n^4d^2)$ if $\mathsf{OD} < n$*
   - *$O(\mathsf{OD}\, n^{\omega+1}d^2)$ if $\mathsf{OD} \geqslant n$*
3. *In the shift case the Big Oh estimates for numbers of operations in parts 1 and 2 above are reduced by a factor of n.*

Proof. As before, we can test if A is row reduced by checking if its leading coefficient matrix is non-singular using fraction-free Gaussian elimination. From the proof of [Giesbrecht and Kim, 2012, Corollary 5.9] we have that $\log\beta := \log ||A_{\mathsf{lin}}||_\infty \in O(\log ||A|| + e\log(nd))$. The theorem now follows from Theorems 28 and 31 and the estimates for $\overline{\mathsf{size}}_{\mathbb{Z}[z]}$ and $\mathsf{size}_{\mathbb{Z}[z]}$. $\square$

5.5 Experimental Results

In this section, we report on experiments and timings that compare FFPopov to other algorithms from the literature. The experiments reported in Table 5.1, Table 5.2, and Table 5.3 are all performed on square non-singular Ore polynomial matrices with entries from the differential ring (i.e., $\sigma(z) = z$ and $\delta(z) = 1$). A random input matrix is more likely to be row reduced or *almost* row reduced (i.e., couple row operations away from being row reduced), in which case FF-Popov will do a minimal amount of computation to get the Popov form. For all the experiments, we generate an input matrix $A \in \mathsf{F}[\partial; \prime]^{n \times n}$ with skewed column degrees. For example if $\deg A = 6$ and $n = 6$, we can have column degrees of A equal $[2, 4, 6]$.

We compare four different Maple implementations of four different algorithms,

- **Modular**: The modular algorithm for computing a row reduced form of an input matrix [P. Davies, 2009];
- **SparseFF**: We use the sparse fraction-free Gaussian elimination algorithm from [Lee and Saunders, 1995] to put an input matrix in reduced row echelon form then extract the Popov form at the end;
- **Muld-Stor**: The Mulders-Storjohann algorithm from [Mulders and Storjohann, 2003] which computes the Popov form of an input matrix by performing unimodular row operations from the Ore ring (described in Section 3.1) directly on the rows of the input matrix;
- **FFPopov**: The fraction-free output-sensitive algorithm described in Section 5.3.

We use the parameters n and d where n denotes the dimension of a square nonsingular input matrix $A \in \mathsf{F}[\partial; \prime]^{n \times n}$ and $d = \deg A$. For the cases where $\mathsf{F} = \mathsf{k}(z)$, k is a field, we use the parameter e to denote the degree in z of the entries of A. All experiments were run on the same machine, and the timings reported were averaged over 5 trials (for the experiments where all the algorithms run to completion). We fill the table with a dash mark whenever the algorithm does not finish after twenty four hours of execution.

We consider three different cases: when $\mathsf{F} = \mathsf{k}(z) = \mathbb{Z}_{11}(z)$, $\mathsf{F} = \mathsf{k}(z) = \mathbb{Q}(z)$, and $\mathsf{F} = \mathbb{Q}$. For the first two cases, we assume that the input matrix is fraction-free (i.e., $A \in \mathsf{k}[z][\partial;\prime]^{n\times n}$). Table 5.1 considers the first case where our non-singular input matrix with skewed column degrees is $A \in \mathbb{Z}_{11}(z)[\partial;\prime]^{n\times n}$. Notice that we use the `modp1` package from Maple for SparseFF and FFPopov, and we use the `OreTools[Modular]` package from Maple for Muld-Stor modular computations.

Parameters			**Algorithms**		
n	***e***	***d***	**SparseFF**	**Muld-Stor**	**FFPopov**
2	8	16	4.4	0.53	0.55
3	8	8	5	0.6	0.9
4	10	20	7265.4	5789	738.5
6	5	10	1672	428.1	200.1
6	10	20	-	-	20146

Table 5.1: Execution time (in seconds) over a non-singular input matrix $A \in \mathsf{F}[\partial;\prime]^{n\times n}$, with $\mathsf{F} = \mathbb{Z}_{11}(z)$, $\deg A = d$, $\deg_z A = e$, and A has skewed column degrees.

Table 5.2 considers the case where our non-singular input matrix with skewed column degrees is $A \in \mathbb{Q}(z)[\partial;\prime]^{n\times n}$.

Parameters			Algorithms			
n	**e**	**d**	**Modular**	**SparseFF**	**Muld-Stor**	**FFPopov**
2	8	16	2172	88	0.23	4.13
3	5	5	345	8.4	0.16	1.1
4	10	20	-	-	-	12200
5	5	10	-	13815	15025	1464
6	5	10	-	-	-	9616

Table 5.2: Execution time (in seconds) over a non-singular input matrix $A \in \mathsf{F}[\partial; \prime]^{n \times n}$, with $\mathsf{F} = \mathbb{Q}(z)$, $\deg A = d$, $\deg_z A = \deg_z A_{\mathrm{lin}} = e$, $\|A\|_\infty \leqslant 99$, and A has skewed column degrees.

Table 5.3 considers the case where our non-singular input matrix with skewed column degrees is $A \in \mathbb{Q}[\partial; \prime]^{n \times n}$. Since $\mathsf{F} = \mathbb{Q}$, the Ore polynomials behave exactly like the usual commutative polynomials (i.e., σ and δ have no effect). This allows us to also compare our implementations with the default Maple function `PopovForm` which is based on the algorithm of [Beckermann et al., 1999] that computes the Popov form of a univariate polynomial input matrix.

Parameters		Algorithms				
n	**d**	**PopovForm**	**Modular**	**SparseFF**	**Muld-Stor**	**FFPopov**
6	10	16.8	347.3	19.8	0.6	9.76
8	12	137.3	3446.6	120.6	1.7	55.6
10	20	3632	71162	2452	9.5	770.5

Table 5.3: Execution time (in seconds) over a non-singular input matrix $A \in \mathsf{F}[\partial; \prime]^{n \times n}$, with $\mathsf{F} = \mathbb{Q}$, $\deg A = d$, $\|A\|_\infty \leqslant 99$, and A has skewed column degrees.

Notice that in Table 5.1 and Table 5.2 the **Muld-Stor** algorithm is faster than the other algorithms for small parameters n, e, and d. But as soon as we increase the parameters values, the **Muld-Stor** algorithm suffers from intermediate coefficient growth which makes it slow, even slower than the **SparseFF** in some cases (e.g., for $n = 5$, $e = 5$, and $d = 10$).

Chapter 6

Recovering the transformation matrix via linear system solving

In Chapter 5, we saw how we can compute the Popov form P of a nonsingular matrix A of Ore polynomials using fraction-free Gaussian elimination. In this chapter, we will show how we can recover the unimodular transformation matrix U such that $UA = P$ using the linear system solving method [Storjohann, 2000, Giesbrecht and Kim, 2012]. For the rest of this chapter, and as we did in Section 2.2, we define the *pivot* of a vector $\vec{v} \in \mathsf{F}[\partial; \sigma, \delta]^{1\times n}$, denoted $\operatorname{piv}(\vec{v})$, as right-most entry of $\vec{v}$ which has $\deg \vec{v}$.

Let $A \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ be a non-singular matrix of degree d and let $P \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ be its unique Popov form computed using FFPopov from Chapter 5 with row degrees $\vec{p} = [p_1, \ldots, p_n]$. Let $U \in \mathsf{F}[\partial; \sigma, \delta]^{n\times n}$ have entries of the format $U_{i,j} = \sum_{k=0}^{(n-1)d} U_{i,j,k}\partial^k$ with unknowns $U_{i,j,k} \in \mathsf{F}$ for $1 \leqslant i, j \leqslant n$. We consider the system of equations in $U_{i,j,k}$ with the following constraints:

$$\begin{array}{llll}(UA)_{i,i,p_i} = 1, & \text{for } 1 \leqslant i \leqslant n & \rightarrow & \text{diagonal entries are monic;} \\ (UA)_{i,j,k} = 0, & \text{for } k > p_i & \rightarrow & \text{pivot entry has degree higher or equal than} \\ & & & \text{other entries in the same row or column.}\end{array} \tag{6.1}$$

By solving the system of equations with the constraints described in (6.1), we mean finding all the values of $U_{i,j,k} \in \mathsf{F}$, with $1 \leqslant i, j \leqslant n$ and $0 \leqslant k \leqslant (n-1)d$. Solving the system of equations, which will be square and non-singular, will result in the unique transformation matrix U such that $UA = P$.

Define the linearization $\phi_U : \mathsf{F}[\partial;\sigma,\delta]^{*\times n} \mapsto \mathsf{F}^{*\times(n^2d+n)}$ by

$$\begin{aligned}\phi_U(\vec{v}) &= \phi_U \left[\begin{array}{ccc} v_1 & \cdots & v_n \end{array}\right] \\ &= \left[\begin{array}{ccc|c|ccc} [v_1]_{nd} & \cdots & [v_n]_{nd} & \cdots & [v_1]_0 & \cdots & [v_n]_0 \end{array}\right] \in \mathsf{F}^{*\times(n^2d+n)},\end{aligned}$$

where $[v_i]_k$ denotes the coefficient of ∂^k of $v_i \in \mathsf{F}[\partial;\sigma,\delta]^{*\times 1}$. Let $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be an input matrix and let $P \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be its Popov form computed using FFPopov from Chapter 5. Let $U \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ be the unimodular transformation matrix such that $UA = P$. We can easily map the system of equation in (6.1) to system of equations in F by constructing A_{lin} (5.1) from A and applying ϕ_U and ϕ_P to U and P respectively (i.e., applying ϕ_U and ϕ_P to every single entry in U and P respectively):

$$\phi_U(U)A_{\mathsf{lin}} = \phi_P(P). \tag{6.2}$$

We can assume that after every call to the Gauss Transform in step 6 of FFPopov, we apply the returned T to the matrix passed in as an argument (this clearly does not dominate the cost of the algorithm). After we finish the Gaussian elimination, we can determine the column rank profile columns as any column containing a pivot. Since A_{lin} has full rank (Lemma 22), then the number of column rank profile columns is $\mu = n^2d + n - \sum \operatorname{rdeg} A$. Let $B \in \mathsf{F}^{\mu\times\mu}$ be the submatrix of A_{lin} comprised of the column rank profile columns. Let $P' \in \mathsf{F}^{n\times\mu}$ be the submatrix of $\phi_P(P)$ comprised of the columns of indices corresponding to the column rank profile columns indices in A_{lin}. We can replace the system of linear equations in the matrix equation (6.2) by

$$\phi_U(U)B = P'. \tag{6.3}$$

Solving (6.3) yields the unique transformation matrix $U \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ such that $UA = P$, the Popov form of A.

Example 34. *Let $A \in \mathbb{Q}(z)[\partial;\prime]^{2\times 2}$ be our input matrix such that*

$$A = \begin{bmatrix} (-16z-54) & 5-33\partial \\ 96 & -25-5\partial \end{bmatrix},$$

and $\deg A = 1$. *We compute the Popov form of A using FFPopov from Chapter 5:*

$$P = \begin{bmatrix} \frac{200z+475}{1719+40z} + \partial & 0 \\ \frac{-1719-40z}{425} & 1 \end{bmatrix} \in \mathbb{Q}(z)[\partial;\prime]^{2\times 2}.$$

We compute A_{lin} and set up the matrix equation as in (6.2):

$$\overbrace{\left[\begin{array}{cc|cc} U_{111} & U_{121} & U_{110} & U_{120} \\ \hline U_{211} & U_{221} & U_{210} & U_{220} \end{array}\right]}^{\phi_U(U)\in\mathbb{Q}(z)^{2\times 4}} \times \overbrace{\left[\begin{array}{cc|cc|cc} -33 & 0 & 5 & -16z-54 & 0 & -16 \\ -5 & 0 & -25 & 96 & 0 & 0 \\ 0 & 0 & -33 & 0 & 5 & -16z-54 \\ 0 & 0 & -5 & 0 & -25 & 96 \end{array}\right]}^{A_{\mathsf{lin}}\in\mathbb{Q}(z)^{4\times 6}}$$

$$= \overbrace{\left[\begin{array}{cc|cc|cc} 0 & 0 & 0 & 1 & 0 & \frac{200z+475}{1719+40z} \\ \hline 0 & 0 & 0 & 0 & 1 & \frac{-1719-40z}{425} \end{array}\right]}^{\phi_P(P)\in\mathbb{Q}(z)^{2\times 6}}.$$

We determine the column rank profile columns which are columns 1, 3, 4, *and* 5. *So we remove those columns from A_{lin} and $\phi_P(P)$ and we write the matrix equation as in (6.3):*

$$\overbrace{\left[\begin{array}{cc|cc} U_{111} & U_{121} & U_{110} & U_{120} \\ \hline U_{211} & U_{221} & U_{210} & U_{220} \end{array}\right]}^{\phi_U(U)\in\mathbb{Q}(z)^{2\times 4}} \times \overbrace{\left[\begin{array}{c|cc|c} -33 & 5 & -16z-54 & 0 \\ -5 & -25 & 96 & 0 \\ \hline 0 & -33 & 0 & 5 \\ 0 & -5 & 0 & -25 \end{array}\right]}^{B\in\mathbb{Q}(z)^{4\times 4}}$$

$$= \overbrace{\left[\begin{array}{c|cc|c} 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]}^{P'\in\mathbb{Q}(z)^{2\times 4}}.$$

The system has a unique solution for the unknown entries in $\phi_U(U)$*,*

$$\phi_U(U) = \left[\begin{array}{cc|cc} \frac{-5}{3438+80z} & \frac{33}{3438+80z} & \frac{-25}{3438+80z} & \frac{-5}{3438+80z} \\ \hline 0 & 0 & \frac{1}{170} & \frac{-33}{850} \end{array}\right].$$

We apply ϕ_U^{-1} *to obtain*

$$U = \begin{bmatrix} \frac{-25}{3438+80z} + \frac{-5}{3438+80z}\partial & \frac{-5}{3438+80z} + \frac{33}{3438+80z}\partial \\ \frac{1}{170} & \frac{-33}{850} \end{bmatrix} \in \mathbb{Q}(z)[\partial; ']^{2\times 2}$$

such that $UA = P \in \mathbb{Q}(z)[\partial; ']^{2\times 2}$*, the Popov form of* A*.*

Theorem 35. *Let* $A \in \mathsf{k}(z)[\partial; \sigma, \delta]^{n\times n}$ *be non-singular with* $\deg A \leqslant d$ *and* $\deg_z A \leqslant e$*. Let* $P \in \mathsf{k}(z)[\partial; \sigma, \delta]^{n\times n}$ *be the Popov form of* A*. We can compute the transformation matrix* $U \in \mathsf{k}(z)[\partial; \sigma, \delta]^{n\times n}$ *such that* $UA = P$ *in* $\tilde{O}(n^7 d^3 e)$ *operations from* k*.*

Proof. We set up the non-singular linear system described by the matrix equation (6.3). We use the algorithm from [Gupta et al., 2012, Algorithm RationalSystemSolve] to solve the n non-singular linear systems in (6.3). The cost of solving each system of equations is $\tilde{O}(n^{2\omega} d^{\omega} e)$ operations in k yielding a total cost of $\mathcal{O}(n^{2\omega+1} d^{\omega} e)$ operations in k. Notice that the algorithm from [Gupta et al., 2012, Algorithm RationalSystemSolve] solves the matrix equation $CX = b$ for an unknown column vector X. So to be able to use it, we first have to transpose the matrix equation in (6.3) so it becomes $\texttt{Transpose}(B)\texttt{Transpose}(\phi_U(U)) = \texttt{Transpose}(P')$. The solution to the linear system can then be transposed back to get the desired result. $\square$

6.1 An asymptotically faster algorithm

There has been a tremendous work in the recent years on algorithms for polynomial matrix computations. This work has lead to some fast algorithms in matrix rank profile computations, which we use in this Section to compute the unimodular transformation matrix $U \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ such that $UA = P$ for a non-singular matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$, P is the Popov form of A, and k is a field. Let $\mathsf{F} = \mathsf{k}(z)$, k is a field, and let $\deg A \leqslant d$ and $\deg_z A \leqslant e$. The algorithm we introduce in this section to compute the unimodular transformation matrix U, is described in the following steps:

a Construct A_{lin} as described in (5.1).

b Compute the column rank profile of A_{lin} using the algorithm from [Zhou, 2012, Algorithm rankProfile]. This costs $\tilde{O}(n^{2\omega}d^{\omega}e)$ operations in k using [Zhou, 2012, Theorem 11.2.].

c Construct the system of equation $\phi_U(U)B = P'$ as in (6.3).

d Compute a structured factorization of the inverse of B using the algorithm from [Zhou et al., 2014, Algorithm Inverse]. This step has cost $\tilde{O}(n^{2\omega}d^{\omega}e)$ operations in k by [Zhou et al., 2014, Theorem 8.].

e Solve the n non-singular linear systems described by the matrix equation in step c. This costs $\tilde{O}(n^5d^2e)$ operations in k as by [Zhou et al., 2014, Section 5.3.]).

Based on the algorithm just outlined, we obtain the following result.

Theorem 36. *Let $A \in \mathsf{k}(z)[\partial;\sigma,\delta]^{n\times n}$ be non-singular with $\deg A \leqslant d$ and $\deg_z A \leqslant e$. Let $P \in \mathsf{k}(z)[\partial;\sigma,\delta]^{n\times n}$ be the Popov form of A (unknown). We can compute the transformation matrix $U \in \mathsf{k}(z)[\partial;\sigma,\delta]^{n\times n}$ such that $UA = P$ in $\tilde{O}(n^6d^3e)$ operations from the field k.*

Since the Popov form P of A has degrees bounded by $d = \deg A$, we can now compute P as

$$(U \mod \partial^{d+1})(A \mod \partial^{d+1}).$$

We immediately get the following corollary.

Corollary 37. *Let $A \in \mathsf{k}(z)[\partial; \sigma, \delta]^{n \times n}$ be non-singular with $\deg A \leqslant d$ and $\deg_z A \leqslant e$. We can compute the Popov form of A in $O\tilde{}(n^6 d^3 e)$ operations from k.*

Notice that the cost reported in Corollary 37 is asymptotically faster than the cost reported for FFPopov in Theorem 32. Implementing the different advanced techniques used by this algorithm in step b, step d, and step e will allow us to see the effectiveness of this algorithm in practise and to compare it with the FFPopov algorithm from Chapter 5.

Chapter 7

Conclusion

In this thesis, we give an extension of the algorithm by [Labhalla et al., 1992] to compute the Hermite form of an input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$. This result is obtained by using a linearization technique that maps the input matrix A to a much larger one which has entries from the field F, then doing Gauss-Jordan elimination to recover the Hermite rows.

We also give a new algorithm FFPopov to compute the Popov form of a non-singular matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ over an Ore polynomial ring. Our approach is to construct a large linear system A_{lin} (5.1) over F whose reduced row echelon form reveals the Popov form of A. Using structural properties of A_{lin}, and since we need only a small, a priori known part of the reduced echelon form, we are able to speed up the Gaussian elimination, especially in cases where A is already close to being row reduced. This approach combines immediately with existing fraction-free techniques which utilize fast matrix multiplication. For the shift and differential Ore rings we bound the bit complexity of our algorithm when the input matrix is over $\mathbb{Z}[z]$. Our algorithm is faster than [Cheng and Labahn, 2007] which computes only a row reduced form and not the Popov form (but, however, handles the important case of non-square and singular input which we do not). FFPopov still works for rectangular and rank deficient matrices (of any dimensions) but the complexity can be much higher. Making the algorithm as efficient for rectangular and rank deficient matrices is left for future work. We present an implementation of this algorithm for the differential case and give some timings that compare our algorithm to some other known algorithms for row reduction and Popov form computation. The timings show the effectiveness of FFPopov especially for large input matrices where FFPopov clearly outperforms all other approaches.

An asymptotically faster algorithm than the one reported in Chapter 5 is given by Corollary 37. This algorithm uses some recent advances in polynomial matrix computations to compute the transformation matrix $U \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$ of a non-singular input matrix $A \in \mathsf{F}[\partial;\sigma,\delta]^{n\times n}$, for $\mathsf{F} = \mathsf{k}(z)$ and F is a field. The Popov form of A can then be obtained by multiplying $U \mod \partial^{d+1}$ by $A \mod \partial^{d+1}$ where $d = \deg A$. This asymptotically faster algorithm for computing the Popov form with transforming matrix is based on a number of recently introduced techniques which improve the deterministic worst case asymptotic complexity of polynomial matrix computations: rank profile [Zhou, 2012], structured inverse [Zhou et al., 2014], linear solving [Zhou et al., 2014]. As future work, it would be desirable to have highly optimized implementations of these new techniques available.

References

[Abramov et al., 2005] Abramov, S. A., Le, H. Q., and Li, Z. (2005). Univariate ore polynomial rings in computer algebra. *Journal of Mathematical Sciences*, 131(5):5885–5903.

[Bareiss, 1968] Bareiss, E. H. (1968). Sylvester's identity and multistep integer-preserving Gaussian elimination. *Mathematics of Computation*, 22(103):565–578.

[Beckermann et al., 2006] Beckermann, B., Cheng, H., and Labahn, G. (2006). Fraction-free row reduction of matrices of Ore polynomials. *Journal of Symbolic Computation*, 41(5):513–543.

[Beckermann and Labahn, 2000] Beckermann, B. and Labahn, G. (2000). Fraction-free computation of matrix rational interpolants and matrix GCD's. *SIAM Journal on Matrix Analysis and Applications*, 22(1):114–144.

[Beckermann et al., 1999] Beckermann, B., Labahn, G., and Villard, G. (1999). Shifted normal forms of polynomial matrices. In Dooley, S., editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'99*, pages 189—196. ACM Press, New York.

[Bostan and Schost, 2005] Bostan, A. and Schost, E. (2005). Polynomial evaluation and interpolation on special sets of points. *Journal of Complexity*, 21(4):420–446. Festschrift for the 70th Birthday of Arnold Schönhage.

[Cheng, 2003] Cheng, H. (2003). Algorithms for normal forms for matrices of polynomials and Ore polynomials. *PhD thesis, University of Waterloo*, http://www.cs.uleth.ca/ cheng/publications.html.

[Cheng and Labahn, 2007] Cheng, H. and Labahn, G. (2007). Modular computation for matrices of Ore polynomials. *Computer Algebra 2006: Latest Advances in Symbolic Algorithms*, pages 43–66.

[Draxl, 1983] Draxl, P. K. (1983). *Skew Fields*, volume 81. Cambridge Univ. Press.

[Edmonds, 1967] Edmonds, J. (1967). On systems of distinct linear representative. *J. Res. Nat. Bur. Standards*, 71B:241–245.

[Geddes et al., 1992] Geddes, K. O., Czapor, S. R., and Labahn, G. (1992). *Algorithms for Computer Algebra*. Kluwer, Boston, MA.

[Giesbrecht and Kim, 2012] Giesbrecht, M. and Kim, M. S. (2012). Computing the Hermite form of a matrix of Ore polynomials. *Journal of Algebra*, 376:341–362.

[Goldstein and Graham, 1974] Goldstein, A. J. and Graham, R. L. (1974). A Hadamard-type bound on the coefficients of a determinant of polynomials. *SIAM Review*, 16:394–395.

[Gupta et al., 2012] Gupta, S., Sarkar, S., Storjohann, A., and Valeriote, J. (2012). Triangular x-basis decompositions and derandomization of linear algebra algorithms over $\mathsf{K}[x]$. *Journal of Symbolic Computation*, 47(4). Festschrift for the 60th Birthday of Joachim von zur Gathen.

[Hafner and McCurley, 1991] Hafner, J. L. and McCurley, K. S. (1991). Asymptotically fast triangularization of matrices over rings. *SIAM Journal of Computing*, 20(6):1068–1083.

[Harvey, 2009] Harvey, D. (2009). Faster polynomial multiplication via multipoint Kronecker substitution. *Journal of Symbolic Computation*, 44(10):1502–1510.

[Harvey and van der Hoeven, 2014] Harvey, D. and van der Hoeven, J. (2014). On the complexity of integer matrix multiplication. *Journal of Symbolic Computation*. To appear.

[Horn and Johnson, 1985] Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.

[Jeannerod et al., 2016] Jeannerod, C. P., Neiger, V., Schost, E., and Villard, G. (2016). Fast computation of minimal interpolation bases in Popov form for arbitrary shifts. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'16*. ACM Press, New York.

[Jeannerod et al., 2017] Jeannerod, C. P., Neiger, V., Schost, E., and Villard, G. (2017). Computing minimal interpolation bases. *Journal of Symbolic Computation*. To appear.

[Kailath, 1980] Kailath, T. (1980). *Linear Systems*. Prentice Hall, Englewood Cliffs, N.J.

[Kaltofen et al., 1990] Kaltofen, E., Krishnamoorthy, M. S., and Saunders, B. D. (1990). Parallel algorithms for matrix normal forms. *Linear Algebra and its Applications*, 136:189–208.

[Labhalla et al., 1992] Labhalla, S. E., Lombardi, H., and Marlin, R. (1992). Algorithmes de calcul de la réduction d'Hermite d'une matrice à coefficients polynomiaux. In *Comptes-Rendus de MEGA92, Nice, France*. Birkhauser.

[Lee and Saunders, 1995] Lee, H. R. and Saunders, B. D. (1995). Fraction free Gaussian elimination for sparse matrices. *Journal of Symbolic Computation*, 19(5):393–402.

[M. Bronstein, 2001] M. Bronstein, M. P. (2001). An introduction to pseudo-linear algebra. *Theoret. Comput. Sci.*, 157(1):3–33.

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

[Neiger, 2016] Neiger, V. (2016). Fast computation of shifted popov forms of polynomial matrices via systems of modular polynomial equations. In *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'16*. ACM Press, New York.

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

[P. Davies, 2008] P. Davies, H. Cheng, G. L. (2008). Computing Popov form of general Ore polynomial matrices. In *Proceedings of Milestones in Computer Algebra*, pages 149–156.

[P. Davies, 2009] P. Davies, H. Cheng, G. L. (2009). A practical implementation of a modular algorithm for Ore polynomial matrices. *Computer Mathematics: 9th Asian Symposium (ASCM2009)*, pages 49–59.

[Sarkar and Storjohann, 2011] Sarkar, S. and Storjohann, A. (2011). Normalization of row reduced matrices. In Leykin, A., editor, *Proc. Int'l. Symp. on Symbolic and Algebraic Computation: ISSAC'11*, pages 297–303. ACM Press, New York.

[Storjohann, 1994] Storjohann, A. (1994). Computation of Hermite and Smith normal forms of matrices. Master's thesis, Dept. of Computer Science, University of Waterloo.

[Storjohann, 2000] Storjohann, A. (2000). *Algorithms for Matrix Canonical Forms*. PhD thesis, Swiss Federal Institute of Technology, ETH–Zurich.

[Zhou, 2012] Zhou, W. (2012). *Fast Order Basis and Kernel Basis Computation and Related Problems*. PhD thesis, Dept. of Computer Science, University of Waterloo.

[Zhou et al., 2014] Zhou, W., Labahn, G., and Storjohann, A. (2014). A deterministic algorithm for inverting a polynomial matrix. *Journal of Complexity*. http://dx.doi.org/10.1016/j.jco.2014.09.004.